EPFL

Thèse n° 8411

High-order essentially nonoscillatory methods based on radial basis functions

Présentée le 3 septembre 2020

à la Faculté des sciences de base
Chaire de mathématiques computationnelles et science de la simulation
Programme doctoral en mathématiques

pour l'obtention du grade de Docteur ès Sciences

par

Fabian MÖNKEBERG

Acceptée sur proposition du jury

Prof. D. Kressner, président du jury
Prof. J. S. Hesthaven, directeur de thèse
Prof. R. Abgrall, rapporteur
Prof. A. Iske, rapporteur
Prof. M. Picasso, rapporteur

École
polytechnique
fédérale
de Lausanne

2020

The only way you ever gonna know how strong you are as an individual is to put yourself in an extreme situation and to see how you handle it
— John D'Elia 2005

Acknowledgements

This thesis would not have been possible without all the people who surrounded me in the last years. Everyone has their own personal role in this journey, and I highly appreciate each one. There are a few people I need to mention in particular.

First of all, I must thank my advisor Prof. Jan S. Hesthaven for all the support and trust he gave me in the last years. Without him it would not have been possible to complete this thesis. Furthermore, I need to express my gratitude to Deep and Qian, who always took the time for discussions and who gave me deep insights into the problem of high-order methods for conservation laws. I must also thank Caterina, a good friend with whom I have shared the office over the past years. I could not think of anyone better to share my office with. In general, I have to thank the whole MCSS and MATHICSE group. Especially, I have to mention Babak, Boris, Mariella and Delphine, without whom the time at EPFL would not have been the same.

Everyone knows that I would not have been able to enjoy my time in Lausanne without various side activities. My deepest gratitude therefore goes to Giacomo and Giacomo, Riccardo, Eva, Edoardo and Edoardo, Gonzalo, Nicolas and the whole crew in Zürich.

Finally, I must thank my parents and my sister for always being there when I need them.

This work has been supported by the SNSF grant 513966.

Lausanne, July 28, 2020 Fabian Mönkeberg

Abstract

Essentially nonoscillatory (ENO) and weighted ENO (WENO) methods on equidistant Cartesian grids are widely employed to solve partial differential equations with discontinuous solutions. However, stable ENO/WENO methods on unstructured grids are less well studied. We propose high-order essentially nonoscillatory methods based on radial basis functions (RBFs) to solve hyperbolic conservation laws. We derive a smoothness indicator that guarantees the satisfaction of the sign property of the resulting interpolant on general one-dimensional grids. Based on this algorithm we introduce an entropy stable arbitrary high-order finite difference method (RBF-TeCNOp) and an entropy stable second order finite volume method (RBF-EFV2) for one-dimensional problems. Hence, we show that methods based on radial basis functions are as powerful as methods based on polynomial reconstruction. Next, we propose a high-order ENO method based on radial basis functions to solve hyperbolic conservation laws on general two-dimensional unstructured grids. The radial basis function reconstruction offers a flexible framework to deal with ill-conditioned cell arrangements. We define a smoothness indicator based the one-dimensional version and a stencil selection algorithm suitable for general meshes. Furthermore, we develop a stable method to evaluate the RBF reconstruction in the finite volume setting to circumvent the stagnation of the error and keep the condition number of the reconstruction bounded. To reduce the computational complexity, we develop the RBF-CWENO method. This method exhibits high-order convergence and robustness when solving challenging problems and is considerably faster. However, the resolution close to shocks and turbulent structures is lower than for the RBF-ENO method. Finally, we present a hybrid high-resolution RBF-ENO method which is based on the RBF-ENO method for unstructured patches and the standard WENO method on structured ones. Furthermore, we introduce a positivity preserving limiter for non-polynomial reconstruction methods that stabilizes the hybrid RBF-ENO method for problems with low density or pressure. We show its robustness on the scramjet inflow problem and a conical aerospike nozzle jet simulation.

Keywords: finite volume method, high-order methods, unstructured grids, radial basis functions, entropy stability, sign property, ENO reconstruction, hybrid methods, positivity preserving method, conical aerospike nozzle.

Zusammenfassung

Essentially nonoscillatory (ENO) und weighted ENO Methoden (WENO) auf äquidistanten kartesischen Gittern sind weit verbreitet, um partielle Differentialgleichungen mit unstetigen Lösungen zu lösen. Stabile ENO/WENO-Methoden auf unstrukturierten Gittern sind jedoch unzureichend untersucht. Zur Lösung hyperbolischer Erhaltungssätze schlagen wir ENO Methoden höherer Ordnung beruhend auf radialer Basisfunktionen (RBF) vor. Wir entwickeln einen Indikator zusammen mit einem Algorithmus, welcher sicherstellt, dass die Interpolationsfunktion auf eindimensionalen Gittern die sog. Sign Property erfüllt. Beruhend auf diesem Algorithmus definieren wir eine Entropie-stabile Finite-Differenzen-Methode beliebig hoher Ordnung (RBF-TeCNOp) und eine Entropie-stabile Finite-Volumen-Methode zweiter Ordnung (RBF-EFV2) für eindimensionale Gleichungen. Damit zeigen wir, dass RBF Methoden vergleichbare Eigenschaften zu polynomiellen Methoden haben. Beruhend auf den eindimensionalen Ergebnissen, konstruieren wir eine RBF-ENO Methode hoher Ordnung, um hyperbolische Erhaltungssätze auf allgemeinen zweidimensionalen unstrukturierten Gittern zu lösen. Die Rekonstruktion mit RBFs bietet eine flexible Möglichkeit, mit schlecht konditionierten Stencils umzugehen. Zur Wahl des Interpolationsstencils verwenden wir einen Algorithmus, welcher für allgemeine Gitter geeignet ist, und einen verallgemeinerten Indikator um die Glattheit der Lösung zu messen. Darüber hinaus beschreiben wir eine stabile Methode zur Auswertung der Interpolation mit RBFs für die Finite-Volumen-Methode, welche das bekannte Problem, der Stagnation des Fehlers, überwindet und die Kondition der Interpolationsmatrix beschränkt hält. Um den Rechenaufwand zu verringern, entwickeln wir eine modifizierte CWENO Methode, welche die Konvergenz hoher Ordnung und die Robustheit gegenüber komplexen Problemen erfüllt. Eine Einschränkung dieser Methode ist die Auflösung in der Nähe von Schockwellen und Turbulenzen. Im Anschluss zeigen wir eine hybride hochauflösende RBF-ENO Methode, welche strukturierte und unstrukturierte Gitter vereint. Wir entwickeln auch eine positivitätserhaltende Modifizierung, welche die Stabilität des Lösers bei Problemen mit Dichte oder Druck nahe Null gewährleistet. Zum Schluss demonstrieren wir die praktische Relevanz und Robustheit unserer Methode anhand zweier Anwendungsbeispiele: zum einen simulieren wir einen Scramjeteinlauf, zum anderen eine radialsymmetrische Aerospike-Düse.

Schlüsselwörter: Finite-Volume-Methode, Methoden hoher Ordnung, Unstrukturierte Gitter, Radiale Basisfunktionen, Entropie-Stabil, Sign Property, ENO Rekonstruktion, Hybridmethoden, Positivitätserhaltende Methode, Aerospike-Düse.

Contents

Notation

Abbreviations

PDE	partial differential equation
RBF	radial basis function
ENO	essentially nonoscillatory method
WENO	weighted ENO method
CWENO	central WENO method
SSPRK	strong stability preserving Runge-Kutta method

Mathematical notation

$\nabla_{\mathbf{u}}$	Jacobian in the variable $\mathbf{u}$
$f_x = \frac{\partial f}{\partial x}$	partial derivative
$f_{xy} = \frac{\partial^2 f}{\partial x \partial y}$	second order partial derivative
$\langle \mathbf{f}, \mathbf{g} \rangle = \sum_{i=1}^{d} \mathbf{f}_i \mathbf{g}_i$	scalar product for $\mathbf{f}, \mathbf{g} \in \mathbb{R}^d$
$C^\infty(\Omega, \hat{\Omega})$	infinitely many times differentiable functions from Ω to $\hat{\Omega}$
$C_0^\infty(\Omega, \hat{\Omega})$	infinitely many times differentiable functions from Ω to $\hat{\Omega}$ with compact support
$L^p(\Omega, \hat{\Omega})$	Lebesgue space with $p \in [1, \infty]$
$L^1_{\mathrm{loc}}(\Omega, \hat{\Omega})$	$\{u : \Omega \to \hat{\Omega} \mid u \in L^1(U) \quad \text{for all } U \subset \Omega \text{ compact}\}$
$\Pi_n(\mathbb{R}^d)$	multivariate polynomials from $\mathbb{R}^d$ to $\mathbb{R}$ of maximum degree $n \in \mathbb{N}$

$[\![f]\!]_{i+1/2} = f_{i+1} - f_i$	jump in the discrete function f at the interface $i + 1/2$
$\langle\!\langle s \rangle\!\rangle_{i+1/2} = \Big(s_{i+1} - s_i\Big)(x_{i+1/2})$	reconstructed jump at the interface $i+1/2$

1 Introduction

Conservation laws are employed in different fields of science and engineering to describe systems which conserve particular quantities, e.g., mass, momentum, and energy. Some important models describing such physical systems are the Maxwell's equations, Euler equations, magnetohydrodynamics (MHD) equations, and Einstein equations of gravitation. In the absence of source terms, a change of these quantities in a domain can be described by the flux through its boundary. The one-dimensional conservation law in differential form is given as a system of $N \in \mathbb{N}$ equations

$$\begin{aligned} u_t + f(u)_x &= 0, \quad (x,t) \in \mathbb{R} \times \mathbb{R}_+, \\ u(0) &= u_0, \end{aligned} \tag{1.1}$$

with the initial condition $u_0 : \mathbb{R} \to \mathbb{R}$, the conserved variables $u : \mathbb{R} \times \mathbb{R}_+ \to \mathbb{R}^N$ and the flux $f : \mathbb{R}^N \to \mathbb{R}^N$. Conservation laws are often expressed as hyperbolic partial differential equations. In such systems, information propagates at a finite speed which implies that the changes in time are local and do not affect all points in space. A typical feature of hyperbolic conservation laws is the spontaneous appearance of discontinuities in the solutions, even when smooth initial data is prescribed [83]. As a consequence, we need a different notion of a solution, i.e., the concept of a weak solution. The function u is called a weak solution of (1.1) if it satisfies

$$\int_{\mathbb{R}\times\mathbb{R}_+} u(x,t)\phi_t(x,t) + f(u(x,t))\phi_x(x,t)\mathrm{d}t\mathrm{d}x + \int_{\mathbb{R}} u_0(x)\phi(x,0)\mathrm{d}x = 0, \tag{1.2}$$

for all compactly supported $\phi \in C^\infty(\mathbb{R} \times \mathbb{R}_+, \mathbb{R}^N)$. However, weak solutions of (1.1) are not unique [83]. Let $\eta : \mathbb{R} \to \mathbb{R}$ be a convex scalar function (entropy function) such that, denoting $Q = \nabla_u \eta \nabla_u f$, there exists a function $q : \mathbb{R} \to \mathbb{R}$ with $\nabla_u q = Q$. We call the function q the entropy flux. The function $u : \mathbb{R} \times \mathbb{R}_+ \to \mathbb{R}^N$ is called an entropy solution of (1.1) for the entropy pair (η, q) if the inequality

$$\eta(u)_t + q(u)_x \leqslant 0, \tag{1.3}$$

is satisfied in a weak sense. For scalar conservation laws, existence and uniqueness of the weak entropy solution in $\mathbb{R}^d$ was shown by Kružkov [74]. The existence and uniqueness of solutions to general multidimensional systems of equations is still an open problem.
Solving systems of hyperbolic conservation laws with high-order methods attracts substantial interest due to their numerous applications in engineering. In particular, in aerospace engineering the availability of fast solvers is of the utmost importance.

All numerical methods for solving hyperbolic conservation laws are based on a discrete representation of the domain. In particular, finite volume methods and Discontinuous Galerkin (DG) methods are based on control volumes, finite differences are based on structured grids of point values and generalized finite difference methods on general point clouds.
The finite volume method emerges naturally from the nature of the problem. Let us consider the one dimensional conservation law (1.1). The finite volume method is based on a discretization $\{x_i\}_{i\in\mathbb{N}} \subset \mathbb{R}$ with the control volumes $C_i = (x_{i-1/2}, x_{i+1/2}]$ with $x_{i+1/2} = \frac{x_i+x_{i+1}}{2}$. Integrating (1.1) over the cell C_i, dividing by its size $|C_i|$, and applying the divergence theorem, we obtain

$$\frac{\mathrm{d}}{\mathrm{d}t}U_i(t) = -\frac{1}{|C_i|}\int_{\partial C_i} f(u)\cdot n(\mathbf{s})\,\mathrm{d}\mathbf{s} = -\frac{1}{|C_i|}(f(u(x_{i+1/2})) - f(u(x_{i-1/2}))), \tag{1.4}$$

where

$$U_i(t) = \int_{C_i} \frac{1}{|C_i|}u(x,t)\mathrm{d}x, \tag{1.5}$$

denotes the cell average value. The idea of the finite volume method is to approximate the flux $f(u(x_{i+1/2}))$ through the interface $x_{i+1/2}$ by a numerical flux

$$F_{i+1/2} = F(\dots, U_i, U_{i+1}, \dots), \tag{1.6}$$

e.g., Lax-Wendroff, Roe, or Lax-Friedrichs flux [61]. Common first order numerical fluxes only depend on the cell values of the direct neighbors of the interface, e.g., $F_{i+1/2} = F(U_i, U_{i+1})$. We recover the semi-discrete scheme

$$\frac{\mathrm{d}}{\mathrm{d}t}U_i(t) = -\frac{1}{|C_i|}(F_{i+1/2} - F_{i-1/2}), \tag{1.7}$$

which can be fully discretized by applying an appropriate time discretization technique, e.g., a strong stability preserving Runge-Kutta method [49]. A viable strategy to prove convergence to the unique solution is based on entropy stability, whenever such a solution exists and is unique. Given an entropy pair (η, q), we call a scheme entropy stable if it fulfills (1.3) at the discrete level. Some convergence results are based on

monotonicity, and total variation diminishing stability [19].

High-order accurate techniques are less diffusive or less dispersive methods with greater computational efficiency. In the case of the finite volume method, we need to combine the approximation of the surface integral of (1.4) by a high-order quadrature rule with a high-order approximation of the flux at the quadrature nodes. The resulting semi-discrete scheme can be integrated in time with a high-order Runge-Kutta method. We consider for each cell C_i a stencil of cells S_i of neighboring cells and construct a reconstruction $s_i : \mathbb{R} \to \mathbb{R}$ of the solution, that interpolates the solution in a mean value sense on the stencil S_i. To formalize the idea we introduce the averaging operator

$$\lambda_C(f) = \frac{1}{|C|} \int_C f(x) \mathrm{d}x, \tag{1.8}$$

with a function $f : \mathbb{R} \to \mathbb{R}$ and a domain $C \subset \mathbb{R}$. The interpolation problem with average values can be written as

$$\lambda_C s_i = U_C, \qquad \text{for all } C \in S_i, \tag{1.9}$$

with the average value U_C of the cell C. The high-order accurate reconstructions s_i are used to compute the high-order numerical flux based on a first order flux $F(U, V)$ as

$$F_{i+1/2} = F(s_i(x_{i+1/2}), s_{i+1}(x_{i+1/2})). \tag{1.10}$$

This interpolation procedure can introduce artificial oscillations which destabilize the scheme. Such spurious oscillations, that occur at discontinuities, are a well-known problem for high-order linear methods, referred to as the Gibbs phenomenon [77]. It can only be avoided by using nonlinear schemes. Based on the MUSCL approach, Harten et al. [57] proposed the essentially nonoscillatory (ENO) scheme. This method reduces the oscillations that occur due to the interpolation step by choosing the stencil with the least oscillatory behavior. Later, this concept was extended to multidimensional domains on general grids [56, 1]. Liu et al. [86] introduced the weighted ENO (WENO) method which allows to obtain even higher order of convergence with similar computational complexity by using convex combinations of solutions computed on different stencils of the ENO method. A further generalization is the Central WENO (CWENO) method which allows the use of stencils of different size [84]. In general, there exist multiple strategies to select the least oscillatory stencil. The method used in [57] is based on divided differences. Another well known indicator is the one introduced by Jiang and Shu [70]. The only known stability result for essentially nonoscillatory methods is based on the sign property [35]. The reconstruction satisfies the sign property if

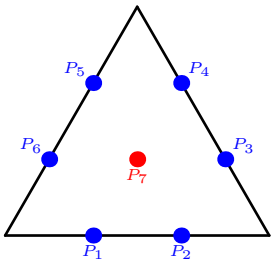


Figure 1.1 – 2D interpolation problem.

the reconstructed jump at each interface has the same sign as the average values

$$\operatorname{sgn}(s_{i+1}(x_{i+1/2}) - s_i(x_{i+1/2})) = \operatorname{sgn}(U_{i+1} - U_i), \tag{1.11}$$

for all $i \in \mathbb{N}$. With this, it is possible to construct entropy stable schemes for solving hyperbolic conservation laws. However, there exist just a few results on reconstructions that satisfy the sign property, e.g., the ENO reconstruction [36], the Minmod reconstruction [35], and a special WENO reconstruction [37].
Alternative methods to avoid stability problems and unphysical oscillations are based on adding artificial viscosity [123] or on the use of limiters [59]. A generalization of the finite volume method is the class of Discontinuous Galerkin finite element methods [18], for which it is necessary to use shock-capturing techniques to ensure nonoscillatory approximations [65].

While polynomial interpolation is well understood in one spatial dimension, it poses some challenges in higher dimensions. In the case of unstructured grids, we face the problem of (unique) solvability of the interpolation system. Let us consider the two-dimensional interpolation problem on the points $P_1, \cdots, P_6$ which lie on a equilateral triangle, see Figure 1.1. Using the second degree polynomial

$$p(\mathbf{x}) = a_0 + a_1\mathbf{x}_1 + a_2\mathbf{x}_2 + a_3\mathbf{x}_1\mathbf{x}_2 + a_4\mathbf{x}_1^2 + a_5\mathbf{x}_2^2. \tag{1.12}$$

we obtain an singular system of equations. An alternative to the exact interpolation is the least-squares method on a larger number of points, e.g., the points $P_1, \ldots, P_7$ in Figure 1.1, which does not guarantee the interpolation property to be satisfied. To overcome this issue we employ radial basis functions (RBF) in the critical interpolation step.
The use of RBFs for scattered data interpolation is not new. Their mesh-free property and flexibility for high-dimensional data makes them advantageous as compared to polynomials. Starting with the seminal work of Hardy [53] RBFs have found application in different domains. Beginning with simple interpolation in multiple space dimensions, Kansa [71, 72] considered approximations of solutions of partial differential

equations with RBFs. This launched the development of further methods, e.g., collocation techniques, variational formulations and boundary element methods based on RBFs [15] and the RBF-FD method [32, 33]. Sonar and Iske [115, 67] introduced the idea to combine RBFs and finite volume schemes. There are several other approaches that combine RBFs with finite volume methods, e.g. a high-order WENO approach based on polyharmonics [2], an adaptive ADER method using polyharmonic WENO reconstructions [3], a high-order WENO approach based on multiquadratics [11].

There exists a spectrum of methods and all of them have their advantages and disadvantages. Combining different methods is not a new idea, with several hybrid approaches readily available in literature. A common idea is to solve different parts of the equation with different schemes, e.g., different methods for viscous and inviscid fluxes [118], or to change the methods depending on the local behavior of the solutions, e.g., for shock capturing [59, 5]. Another approach splits the domain into structured and unstructured parts and solves them with different methods [131, 132, 130, 112].

1.1 Main contributions of the thesis

The main contribution of this thesis is the development of essentially nonoscillatory methods to solve general conservation laws on general unstructured grids. In particular, we introduce a sign-preserving essentially nonoscillatory reconstruction method based on infinitely smooth RBFs. We prove the sign property for the second and third order pointwise reconstruction and we conjecture that it holds for higher order reconstructions and for the mean value based version. Thus, we construct the RBF-TeCNOp method, an arbitrary high-order entropy stable finite difference scheme, and the RBF-EFV2 method, a second order entropy stable finite volume scheme, based on the RBF reconstruction for one-dimensional conservation laws with infinitely smooth RBFs. To ensure stability of the RBF interpolation we adopt the vector-valued rational approximation method [127]. These methods can be generalized to multidimensional problems on structured grids by using the principle of dimensional splitting.
To generalize the method for unstructured grids in multiple dimensions, we introduce the high-order RBF-ENO method. This method is based on the reconstruction algorithm and the smoothness indicator from the one-dimensional entropy stable method. Further, we develop a stable evaluation method of the RBF interpolation, which is based on the augmentation with polynomials, to avoid the computationally expensive vector-valued rational approximation method. This evaluation method exploits the fact that the interpolation for the high-order finite volume methods is based on small stencils of a fixed number of cells with bounded element sizes, in contrast to the general RBF interpolation problems. For one spatial dimension, we show the stability of the evaluation method. Numerical examples suggest its validity for two

dimensions. Despite this improved evaluation method, the stencil selection algorithm has still quadratic cost in the stencil size.
The RBF-CWENO method of third order reduces the computational burden of the RBF-ENO method by considering only symmetric stencils of different sizes. For the RBF-CWENO method we employ a smoothness indicator based on the work of Jiang and Shu [70], except for the stencil of size one, for which we adapt an idea developed for the multi-resolution WENO scheme [137].
Finally, we describe a hybrid high-resolution method with reduced computational cost. Our approach is based on a partition of the domain into structured and unstructured patches and on the combination of the standard WENO scheme on the former and of the RBF-ENO method on the latter. The connection between the patches is achieved with ghost cells. To enable the solution of problems with low density or pressure, we also describe a positivity preserving limiter for non-polynomial reconstructions. This combination allows us to solve a variety of non-standard challenging problems, such as the scramjet inflow problem with Mach numbers 3 and 10, and a conical aerospike nozzle simulation.

1.2 Overview of the thesis

In this thesis, we develop stable methods to solve hyperbolic conservation laws on general grids. We introduce essentially nonoscillatory methods based on an RBF reconstruction, develop a new smoothness indicator based on RBFs and propose a strategy to circumvent stability issues introduced by the RBF interpolation.
In Chapter 2, we describe the required theoretical background of hyperbolic conservation laws, finite volume/difference method, and radial basis functions.
In Chapter 3, we develop an arbitrary high-order entropy stable finite difference scheme and a second order entropy stable finite volume scheme based on the RBF reconstruction for one-dimensional conservation laws.
In Chapter 4, we introduce a high-order ENO method based on an RBF reconstruction for general grids. We develop a stable evaluation method of the interpolation problem build on polynomial augmented multiquadratic splines. This method is based on the smoothness indicator introduced in Chapter 3.
In Chapter 5, we extend the idea of the CWENO method using the RBF reconstruction. We start with the one-dimensional version and generalize it for two-dimensional problems on general grids.
In Chapter 6, we reduce the computational cost of the RBF-ENO method by developing a hybrid high-resolution RBF-ENO method. Furthermore, we develop a positivity preserving limiter for non-polynomial reconstructions that ensures the solvability for problems with density or pressure close to zero.
Chapter 7 summarizes the results and offers an outlook on future work.

2 Theoretical background

In this chapter, we present the theoretical concepts on which this thesis is based. We start by introducing hyperbolic conservation laws. Next, we describe finite volume and finite difference methods to solve partial differential equations. In the end, we discuss radial basis functions and their applications to interpolation problems.

2.1 Hyperbolic conservation laws

In this section, we introduce hyperbolic conservation laws, some concepts of its solution and some important results. Conservation laws in d space dimensions can be described through systems of partial differential equations of the form

$$\begin{aligned} \mathbf{u}_t + \sum_{i=1}^{d} f_i(\mathbf{u})_{\mathbf{x}_i} = 0, \qquad & \text{in } \mathbb{R}^d \times \mathbb{R}_+, \\ \mathbf{u}(0) = \mathbf{u}_0, \qquad & \text{in } \mathbb{R}, \end{aligned} \tag{2.1}$$

with the initial conditions $\mathbf{u}_0 : \mathbb{R}^d \to \mathbb{R}^N$, the conserved variables $\mathbf{u} : \mathbb{R}^d \times \mathbb{R}_+ \to \mathbb{R}^N$, e.g. mass, momentum, and energy, and the flux functions $f_i : \mathbb{R}^N \to \mathbb{R}^N$. These variables are conserved in the sense that for any test volume $\Omega \subset \mathbb{R}^d$ it holds

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_\Omega \mathbf{u}(\mathbf{x}, t) \mathrm{d}\mathbf{x} = - \int_{\partial\Omega} f(\mathbf{u}(\mathbf{s}, t)) \cdot \mathbf{n}(\mathbf{s}) \, \mathrm{d}\mathbf{s}, \tag{2.2}$$

with the outwards pointing normal vector $\mathbf{n}(\mathbf{s})$ and $f = (f_1, \ldots, f_d)$. Thus, the change of the conserved variables over time in the test volume can be described by the flux through its boundary.
A conservation law is called strongly *hyperbolic* if the Jacobian $\nabla_{\mathbf{u}}(f(\mathbf{u}) \cdot \mathbf{n})$ has N real eigenvalues $\lambda_1(\mathbf{u}) \leqslant \cdots \leqslant \lambda_N(\mathbf{u})$ with linearly independent eigenvectors $\mathbf{r}_1(\mathbf{u}), \ldots, \mathbf{r}_N(\mathbf{u}) \in \mathbb{R}^N$ in the each direction $\mathbf{n} \in \mathbb{R}^d$. Physically, this means that information is moving with finite speed. For example, a small perturbation in space does

not immediately affect the whole solution, but spreads over time.

2.1.1 Examples

Let us take a look at some of the most common examples of hyperbolic conservation laws.

Linear advection equation

The most basic equation is called the linear advection equation. It can be described by

$$\begin{aligned} &u_t + a u_x = 0, \\ &\quad u(x,0) = u_0(x), \quad \text{for } -\infty < x < \infty, \end{aligned} \tag{2.3}$$

with the wave speed $a \in \mathbb{R}$ [83]. The solution $u : \mathbb{R} \times \mathbb{R}_+ \to \mathbb{R}$ is a traveling wave with speed a. For a differentiable initial condition u_0 it is easy to verify that $u(x,t) = u_0(x-at)$ is its solution. So, the initial condition is propagating with speed $a > 0$ to the right (or to the left for $a < 0$).

Burgers' equation

A famous nonlinear example is Burgers' equation. Here, we have a convex flux f and therefore it belongs mathematically to the simple cases. Its equation is

$$\begin{aligned} &u_t + \frac{1}{2}\left(u^2\right)_x = 0, \\ &\quad u(x,0) = u_0(x), \quad \text{for } -\infty < x < \infty. \end{aligned} \tag{2.4}$$

Provided u is differentiable, (2.4) can be rewritten into the form

$$u_t + u u_x = 0, \tag{2.5}$$

which can be compared with the linear advection equation with an nonconstant velocity u [83].

Buckley-Leverett equation

The Buckley-Leverett equation is a simple model for a two-phase fluid flow in porous media. It finds some application in oil reservoir simulation to describe the ratio between two fluids. It is an example of a non-convex flux. In one space dimension it

can be expressed by (2.1) with

$$f(u) = \frac{u^2}{u^2 + a(1-u)^2}, \tag{2.6}$$

with $a \in \mathbb{R}$ that describes the ratio of the viscosities of the fluids [83].

Shallow water equations

A well-known system of hyperbolic conservation laws are the shallow water equations. They describe some liquid flow under the assumptions that the horizontal length scales are much larger than the vertical ones. This thin layer fluid flow assumes some bottom topography and a free surface above. And the model can be derived from the Euler equations, which model general fluid motions. In one space dimension and without bottom topography, we get the nonlinear system of equations depending on the mass flow m and the fluid height h

$$\begin{pmatrix} h \\ m \end{pmatrix}_t + \begin{pmatrix} m \\ \frac{1}{2}gh^2 + m^2/h \end{pmatrix}_x = 0, \tag{2.7}$$

with the gravitational constant g [83]. The first equation comes from the conservation of mass and the second equation describes the conservation of momentum.
The eigenvalues and eigenvectors of its Jacobian are

$$\lambda_{1,2} = \frac{m}{h} \pm \sqrt{gh}, \qquad \mathbf{r}_{1,2} = \begin{pmatrix} 1 \\ \frac{m}{h} \pm \sqrt{gh} \end{pmatrix}. \tag{2.8}$$

Euler equations

The Euler equations of gas dynamics are, when compared to the shallow water equations, a more complex system of equations. They consist of the continuity equation, the momentum equations, and the conservation of the total energy. In two space dimensions, they are described by

$$f_i(\mathbf{u}) = \begin{pmatrix} m_i \\ \frac{m_i m_1}{\rho} + p\delta_{i1} \\ \frac{m_i m_2}{\rho} + p\delta_{i2} \\ \frac{m_i}{\rho}(E+p) \end{pmatrix}, \tag{2.9}$$

with the delta function

$$\delta_{ij} = \delta_j(i) = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{otherwise} \end{cases}, \tag{2.10}$$

with $\mathbf{u} = (\rho, m_1, m_2, E)^T$ the density ρ, the mass flux m_1 and m_2 in x- and y-direction, respectively, the total energy E, and the pressure $p = \mathcal{R}\rho T = (\gamma - 1)(E - \frac{1}{2}\frac{m_1^2+m_2^2}{\rho})$ assuming an ideal gas with the ratio of specific heat γ [61]. The eigenvalues of the Jacobian

$$A(\mathbf{u}, \mathbf{n}) = \nabla_{\mathbf{u}} f_1 n_1 + \nabla_{\mathbf{u}} f_2 n_2, \tag{2.11}$$

in direction $\mathbf{n} = (n_1, n_2)^T$ are

$$\lambda_1 = u_n - c, \qquad \lambda_2 = \lambda_3 = u_n, \qquad \lambda_4 = u_n + c, \tag{2.12}$$

where $u_n = u_1 n_1 + u_2 n_2$ and $c^2 = \gamma p/\rho$ with its corresponding eigenvectors

$$R(\mathbf{u}, \mathbf{n}) = \begin{pmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \mathbf{r}_3 \\ \mathbf{r}_4 \end{pmatrix}^T = \begin{pmatrix} 1 & 1 & 0 & 1 \\ u_1 - cn_1 & u_1 & n_2 & u_1 + cn_1 \\ u_2 - cn_2 & u_2 & -n_1 & u_2 + cn_2 \\ H - cu_n & \frac{1}{2}(u_1^2 + u_2^2) & u_1 n_2 - u_2 n_1 & H + cu_n \end{pmatrix}, \tag{2.13}$$

where $H = (\gamma - 1)^{-1} c^2 + \frac{1}{2}(u_1^2 + u_2^2)$ and the velocity $u_i = m_i/\rho$.

2.1.2 Discontinuous solutions

A characteristic property of hyperbolic conservation laws is the spontaneous occurrence of discontinuous solutions. Let us consider a scalar conservation law in one space dimension. We start with the linear advection equation (2.3) with its solution $u(x,t) = u_0(x - at)$. Thus, the values $u(x,t)$ are constant along the line $x - at = x_0$ (in phase space). These lines are called characteristics and they are the trajectories of the information flow. Further, they satisfy the following differential equation

$$\begin{aligned} x'(t) &= a, \\ x(0) &= x_0. \end{aligned} \tag{2.14}$$

For Burgers' equation (2.4) the trajectories satisfy the ordinary differential equation

$$\begin{aligned} x'(t) &= u(x(t), t), \\ x(0) &= x_0. \end{aligned} \tag{2.15}$$

To verify that u is constant along them, we simply calculate

$$\frac{\mathrm{d}}{\mathrm{d}t} u(x(t), t) = 0. \tag{2.16}$$

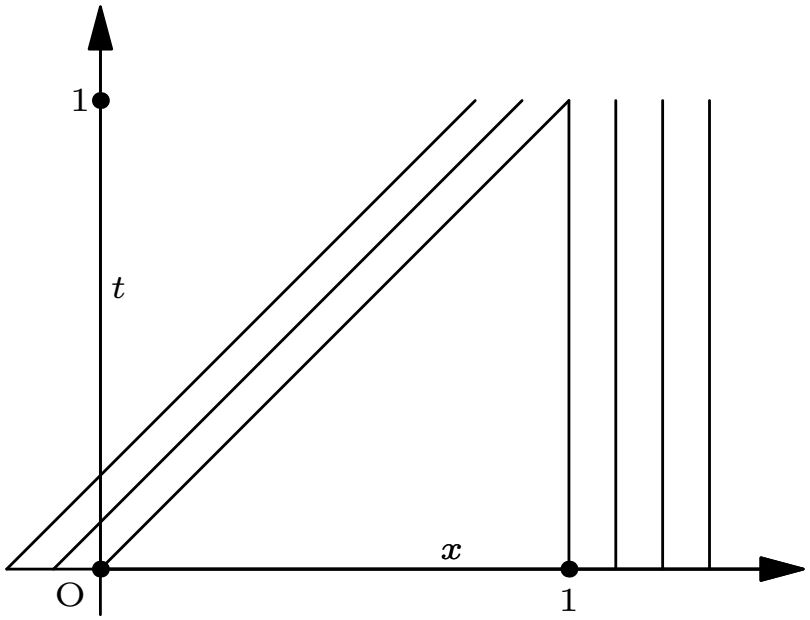


Figure 2.1 – Characteristics for Burgers' equation.

We have $x'(t) = u_0(x_0)$ which means the characteristics are straight lines of the form

$$x(t) = x_0 + u_0(x_0)t. \tag{2.17}$$

Let us now consider the initial conditions

$$u_0(x) = \begin{cases} 1 & \text{if } x < 0 \\ h(x) & 0 \leqslant x \leqslant 1 \\ 0 & \text{if } x > 1 \end{cases}, \tag{2.18}$$

with h such that u_0 is smooth. By considering the characteristics for these initial conditions we observe straight lines starting from the left side of zero with slope one, vertical lines on the right side of one and in between we have some different shapes (Figure 2.1). Since the values on the characteristics are constant, we end up with a discontinuous solution at time $t = 1$ when the characteristics intersect. The same behavior can be observed for more complicated nonlinear systems of equations.

2.1.3 Weak formulation

As discussed above, the solution to hyperbolic conservation laws can be discontinuous. Thus, they do not satisfy the smoothness requirements to make sense of (2.1) on strong form. To allow the existence of discontinuous solutions, we rewrite the equations in weak form, and interpret its solution in the weak (distributional) sense. Therefore, we

take the inner product of each term of (2.1) with a smooth and compactly supported test function $\phi \in C_0^\infty(\mathbb{R}^d \times \mathbb{R}_+, \mathbb{R}^N)$, integrate it over the whole space and time domain and apply integration by parts in time or space. This yields

$$\int_{\mathbb{R}^d \times \mathbb{R}_+} \langle \mathbf{u}, \phi_t \rangle + \sum_{i=1}^{d} \langle f_i(\mathbf{u}), \phi_{\mathbf{x}_i} \rangle \mathrm{d}t\mathrm{d}\mathbf{x} + \int_{\mathbb{R}^d} \langle \mathbf{u}_0(\mathbf{x}), \phi(\mathbf{x}, 0) \rangle \mathrm{d}\mathbf{x} = 0, \tag{2.19}$$

and is called the *weak formulation* of (2.1).

Definition 2.1. *A function* $\mathbf{u} \in L^1_{\mathrm{loc}}(\mathbb{R}^d \times \mathbb{R}_+, \mathbb{R}^N)$ *is called a* weak solution *of* (2.1) *if* (2.19) *holds for all* $\phi \in C_0^\infty(\mathbb{R}^d \times \mathbb{R}_+, \mathbb{R}^N)$.

After introducing the weak formulation and keeping in mind that solutions can become discontinuous, we consider a special kind of initial conditions

$$\mathbf{u}_0(x) = \begin{cases} \mathbf{u}_l, & \text{for } x < 0 \\ \mathbf{u}_r, & \text{for } x > 0 \end{cases}. \tag{2.20}$$

Problems with these initial conditions are known as *Riemann problems*. They help to understand the behavior of discontinuities.

Example 2.1. *Let us consider the Riemann problem based on Burgers' equation* (2.4) *with* $u_l < u_r$. *In this case, there exist infinitely many weak solutions [83]. One of them is the rarefaction wave*

$$u(x,t) = \begin{cases} u_l, & \textit{for } x < u_l t \\ x/t, & \textit{for } u_l t \leqslant x \leqslant u_r t \\ u_r, & \textit{for } x > u_r t \end{cases}. \tag{2.21}$$

A second solution is the shock wave

$$u(x,t) = \begin{cases} u_l, & \textit{for } x < st \\ u_r, & \textit{for } x > st \end{cases}, \tag{2.22}$$

with the shock speed $s = (u_l + u_r)/2$. *Thus, weak solutions are not unique and the problem is not well-posed.*

To determine the shock speed s we can apply the *Rankine-Hugoniot jump condition*

$$\sum_{i=1}^{d} n_i [\![f_i(\mathbf{u})]\!] = s [\![\mathbf{u}]\!], \tag{2.23}$$

with $[\![\mathbf{u}]\!] = \mathbf{u}^+(\mathbf{x}, t) - \mathbf{u}^-(\mathbf{x}, t)$, $\mathbf{u}^\pm(\mathbf{x}, t) = \lim_{h \to 0} \mathbf{u}(\mathbf{x} \pm \mathbf{n}h, t)$ and $\mathbf{n} = (n_1, \ldots, n_d)$.

2.1.4 Vanishing viscosity and entropy solution

Based on the knowledge that weak solutions are not unique, we need to introduce a concept that ensures uniqueness for scalar conservation laws. The goal is to pick the physical relevant solution which can be defined by the vanishing viscosity approach. Let us assume a conservation law (2.1) in one space dimension. The vanishing viscosity solution is the unique limit for $\varepsilon \to 0$ of the equations

$$\begin{aligned} &\mathbf{u}_t + f(\mathbf{u})_x = \varepsilon \mathbf{u}_{xx}, \\ &\quad \mathbf{u}(x,0) = \mathbf{u}_0(x), \quad \text{for } -\infty < x < \infty. \end{aligned} \tag{2.24}$$

The existence of the viscosity solution and its convergence to a weak solution of (2.19) for $\varepsilon \to 0$ is ensured by Oleinik [98] and Kruzkov [74] for initial data of small total variation.
However, this definition is not practical when working with numerical schemes. An other approach uses the concept of entropy functions. In physics, we have the entropy which is a physical quantity that is constant along particle paths and that jumps to higher values across discontinuities. The definition of the mathematical entropy has the opposite sign from the physical entropy such that the total entropy decreases over time. Let η be a convex scalar function such that there exists $q_i : \mathbb{R}^N \to \mathbb{R}$ for $i = 1, \ldots, d$ with $\nabla_{\mathbf{u}} q_i(\mathbf{u}) = \nabla_{\mathbf{u}} \eta(\mathbf{u}) \nabla_{\mathbf{u}} f_i(\mathbf{u})$. After multiplying $\nabla_{\mathbf{u}} \eta(\mathbf{u})$ to (2.1) we can rewrite it as

$$\eta(\mathbf{u})_t + \sum_{i=1}^{s} q_i(\mathbf{u})_{\mathbf{x}_i} = 0, \tag{2.25}$$

for smooth solutions $\mathbf{u}$. We call (η, q) the entropy pair with the *entropy function* η and its *entropy flux* $q = (q_1, \ldots, q_d)$.

Definition 2.2 (Entropy Solution)**.** *The function* $\mathbf{u} : \mathbb{R}^d \times \mathbb{R}_+ \to \mathbb{R}^N$ *is called an* entropy solution *of* (2.1)*, for the convex entropy function* η *and its corresponding entropy fluxe* $(q_1, \ldots, q_d)$*, if the inequality*

$$\eta(\mathbf{u})_t + \sum_{i=1}^{d} q_i(\mathbf{u})_{\mathbf{x}_i} \leqslant 0, \tag{2.26}$$

is satisfied in the weak sense.

In the case of scalar conservation laws, existence and uniqueness of weak entropy solutions in $\mathbb{R}^d$ was shown by Kruzkov [74] . Note that the vanishing viscosity solution satisfies the entropy inequality (2.26).
Furthermore, we can use the *entropy variable* $\mathbf{v} := \mathbf{v}(\mathbf{u}) = (\nabla_{\mathbf{u}} \eta(\mathbf{u}))^T$ to symmetrize (2.1) in the sense that $\nabla_{\mathbf{v}} \mathbf{u}(\mathbf{v})$ is symmetric positive definite and $\nabla_{\mathbf{v}} f_i(\mathbf{u}(\mathbf{v}))$ is symmetric. The function $\mathbf{v}(\mathbf{u})$ defines a change of variables that is invertible in the

case of a strongly convex η. This can be seen by introducing the *entropy potential* $\psi_i(\mathbf{v}) = \mathbf{v}^T \cdot f_i(\mathbf{u}(\mathbf{v})) - q_i(\mathbf{u}(\mathbf{v}))$ and inserting $\nabla_{\mathbf{v}} f_i(\mathbf{u}(\mathbf{v})) = \psi_i''(\mathbf{v})$, the Hessian of ψ_i.

Examples

Let us take a look at some example equations and their entropy functions.

- For the **linear advection equation** (2.3), we can choose the entropy function to be $\eta(u) = \frac{u^2}{2}$ and $q(u) = a\frac{u^2}{2}$. This gives us the following entropy variables and potential

$$v(u) = u, \qquad \psi(u) = a\frac{u^2}{2}. \tag{2.27}$$

- For the **Burgers' equation** (2.4), the entropy function $\eta(u) = \frac{u^2}{2}$ with the entropy flux $q(u) = \frac{u^3}{3}$ is a valid choice for the entropy pair. This results in

$$v(u) = u, \qquad \psi(u) = \frac{u^3}{6}. \tag{2.28}$$

- In the case of the **Shallow water equations**, we choose

$$\eta = \frac{1}{2}\Big(\frac{m^2}{h} + gh^2\Big), \qquad q = \frac{m^3}{h^2} + gmh, \tag{2.29}$$

and

$$\mathbf{v} = \begin{pmatrix} gh - \frac{m^2}{2h^2} \\ \frac{m}{h} \end{pmatrix}, \qquad \psi = \frac{1}{2}gmh. \tag{2.30}$$

More details can be found in [102].

- For the two-dimensional **Euler equations** (2.9), we have to be aware, that the thermodynamical entropy $s = \log(p) - \gamma\log(\rho)$ is different from the entropy function and entropy flux. An entropy pair is proposed in [54]

$$\eta = \frac{-\rho s}{\gamma - 1}, \qquad q_i = \frac{-m_i s}{\gamma - 1}, \qquad i = 1, 2, \tag{2.31}$$

with

$$\mathbf{v} = \begin{pmatrix} \frac{\gamma - s}{\gamma - 1} - \beta\frac{m_1^2 + m_2^2}{\rho^2} \\ 2\beta\frac{m_1}{\rho} \\ 2\beta\frac{m_2}{\rho} \\ -2\beta \end{pmatrix}, \qquad \psi_i = m_i, \qquad \beta = \frac{\rho}{2p}. \tag{2.32}$$

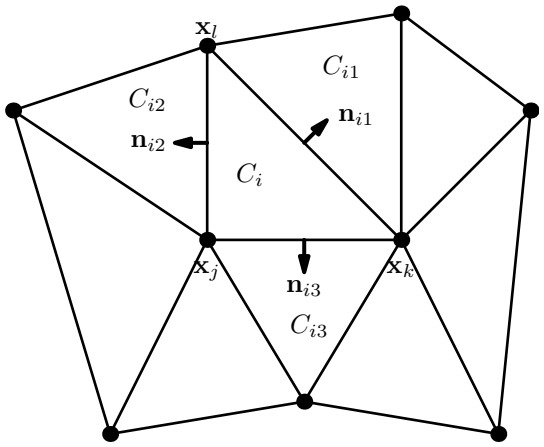


Figure 2.2 – Triangulation for the finite volume method.

2.2 Finite volume method

In this section, we introduce the basic concept of the finite volume method to construct approximate solutions to (2.1). The goal of numerical methods is the approximation of the physical correct solution. However, for hyperbolic conservation laws convergence results are mainly available for scalar and one-dimensional systems.

2.2.1 First-order finite volume method for scalar equations

In general, we assume a grid of the domain $\Omega \subset \mathbb{R}^d$, consisting of cells C_i with $i \in \mathbb{N}$, as shown in Fig. 2.2 for the two-dimensional case. The finite volume method is based on the integral form (2.2) with $\Omega = C_i$ and the cell averages $U_i = \frac{1}{|C_i|} \int_{C_i} u(\mathbf{x}) \mathrm{d}\mathbf{x}$.

The one-dimensional case

The one-dimensional grid $\{x_i\}_{i\in\mathbb{Z}}$ is composed of cells $C_i = [x_{i-1/2}, x_{i+1/2}]$ of size Δx. We can evaluate the right hand side directly at its left and right cell interfaces to obtain (1.4). After approximating $f(u(x_{i+1/2}))$ in terms of the cell averages U_i and U_{i+1} with a flux function

$$F_{i+1/2} := F(U_i, U_{i+1}) \approx f(u(x_{i+1/2})), \tag{2.33}$$

we recover the semi-discrete scheme (1.7). Note that, we have to use some quadrature rule for approximating the boundary integral in higher space dimensions. Some possible choices for flux functions are

- local Lax-Friedrichs flux $F(U,V) = \frac{f(U)+f(V)}{2} - \frac{\alpha}{2}(V-U)$,

with $\alpha = \max_{U<W<V} |f'(W)|$.

- global Lax-Friedrichs flux $F(U,V) = \frac{f(U)+f(V)}{2} - \frac{\alpha}{2}(V-U)$, with $\alpha = \max_W |f'(W)|$,
- Roe-flux

$$F(U,V) = \begin{cases} f(U) & \text{if } \frac{f(V)-f(U)}{V-U} \geqslant 0 \\ f(V) & \text{otherwise} \end{cases}. \tag{2.34}$$

In the end, we can apply an arbitrary time discretization technique to recover a fully discrete scheme from (1.7), e.g., the Euler method, a strong stability preserving Runge-Kutta method [49]. Using the explicit Euler method we receive the well-known fully discrete scheme in conservative form

$$U_i^{n+1} = U_i^n - \frac{\Delta t}{|C_i|}\Big[F_{i+1/2}^n - F_{i-1/2}^n\Big], \tag{2.35}$$

where $U_i^n \approx U_i(t^n)$, $\Delta t = t^{n+1} - t^n$ and $F_{i+1/2}^n = F(U_i^n, U_{i+1}^n)$. The discrete equation (2.35) is called conservative form since the step function

$$\tilde{u}(x,t) = U_i^n, \quad \text{for } x \in (x_{i-1/2}, x_{i+1/2}],\ t \in (t^{n-1}, t^n], \tag{2.36}$$

is *conserved* over time, e.g., $\int_{\mathbb{R}} \tilde{u}(x,t)\mathrm{d}x = \int_{\mathbb{R}} u_0(x)\mathrm{d}x$. Futher, the numerical scheme (2.35) is called *consistent* with the original conservation law if the flux function fulfills

$$F(U,U) = f(U), \tag{2.37}$$

for every $U \in \mathbb{R}$. The Roe and the Lax-Friedrichs flux are consistent.
Lax and Wendroff [79] showed that a convergent, conservative, and consistent method converges to a weak solution . However, this does not ensure convergence to the physically correct weak solution.
Let (η, q) be an entropy pair. To ensure convergence to its weak entropy solution it is enough to validate the cellwise *discrete entropy condition*

$$\frac{\mathrm{d}}{\mathrm{d}t}\eta(U_i) \leqslant -\frac{1}{|C_i|}[Q(U_i, U_{i+1}) - Q(U_{i-1}, U_i)] = -\frac{1}{|C_i|}[Q_{i+1/2} - Q_{i-1/2}], \tag{2.38}$$

with some numerical entropy flux Q that is consistent with q. Methods that satisfy the discrete entropy condition are called *entropy stable* [58, 81]. Under the assumption of convergence, this concept ensures convergence to the weak entropy solution.
Note that to demonstrate convergence we need some stronger stability properties. Convergent subsequences can be found in sequences of compact sets, e.g., sets of

bounded total variation

$$TV(v) = \limsup_{\epsilon \to 0} \frac{1}{\epsilon} \int_{-\infty}^{\infty} |v(x+\epsilon) - v(x)| \mathrm{d}x. \tag{2.39}$$

and with bounded support. The total variation of weak solutions to scalar problems is nonincreasing over time [47]. Numerical methods with this property are called *total variation dimishing (TVD).*
Similar methods are called *monotinicity preserving schemes* and *monotone schemes* [83]. Solutions of monotonicity preserving schemes are monotone if their initial data is monotone. Thus, they control oscillations close to discontinuities.
Monotone methods ensure that for two initial conditions

$$u_i^0 \leqslant v_i^0, \qquad \text{for all } i \in \mathbb{N}, \tag{2.40}$$

we get

$$u_i^n \leqslant v_i^n, \qquad \text{for all } i \in \mathbb{N}, \tag{2.41}$$

for all $n \in \mathbb{N}$. Theorem 2.1 expresses the connection between these different concepts of stability.

Theorem 2.1 (Connection: monotone, TVD and monotonicity preserving schemes, [61])**.** *Monotone schemes are TVD and TVD schemes are monotonicity preserving. Furthermore, linear monotonicity preserving schemes are monotone.*

In addition, it holds

Theorem 2.2 (Monotone schemes [83])**.** *Numerical solutions of consistent monotone methods with fixed ratio $\Delta t/\Delta x$ converge to the weak entropy solution as $\Delta t \to 0$. However, linear monotone schemes are at most first order accurate.*

These results show us that the concept introduced above works, but we are restricted to first order accurate methods for linear schemes. For higher-order methods, we need weaker conditions to ensure convergence to its entropy solution.

Finite volume method in multiple dimensions

Let us consider the two dimensional case as an example. The generalization from two dimensions to higher ones is direct. We assume a triangular grid of the domain $\Omega \subset \mathbb{R}^2$, consisting of triangular cells $C_i = \{\mathbf{x}_j, \mathbf{x}_k, \mathbf{x}_l\}$ as illustrated in Fig. 2.2. Similar to the one-dimensional case, we integrate (2.1) over the cell, divide it by the cell size $|C_i|$ and

apply the divergence theorem to recover the semi-discrete scheme

$$\frac{\mathrm{d}U_i}{\mathrm{d}t} + \frac{1}{|C_i|}\sum_{l_e=1}^{3} F_{il_e} = 0, \tag{2.42}$$

with the numerical flux $F_{il_e} = F_{il_e}(U_i, U_{il_e}, \mathbf{n}_{il_e})$ and the accuracy condition

$$\int_{S_{il_e}} f(u)\cdot \mathbf{n}_{il_e} \mathrm{d}\mathbf{s}(\mathbf{x}) = F_{il_e} + \mathcal{O}(\Delta x^p), \tag{2.43}$$

where $f = (f_1, f_2)$, $S_{il_e} = \partial C_i \cap \partial C_{il_e}$, U_{il_e} is the cell average of C_{il_e} and $\mathbf{n}_{il_e}$ is the outward pointing normal vector.
The numerical flux F_{il_e} can be expressed by using an (approximate) Riemann solver. A common choice is the Rusanov flux

$$F^R_{il_e}(U, V, \mathbf{n}_{il_e}) = \frac{|S_{il_e}|}{2}\big(f(U) + f(V)\big)\cdot \mathbf{n}_{il_e} - \frac{\alpha_{il_e}(U,V)|S_{il_e}|}{2}\big(V - U\big), \tag{2.44}$$

with

$$\alpha_{il_e}(U,V) = \max\{\lambda_{max}(\nabla_{\mathbf{u}} f(U)\cdot \mathbf{n}_{il_e}), \lambda_{max}(\nabla_{\mathbf{u}} f(V)\cdot \mathbf{n}_{il_e})\}. \tag{2.45}$$

Here, $\lambda_{max}(A)$ is the biggest eigenvalue of A and $\mathbf{n}_{il_e}$ the normal vector to the interface S_{il_e}.

2.2.2 High-order finite volume method for scalar equations

First order methods give us access to fast and stable schemes for solving hyperbolic conservation laws, but they have drawbacks. Main problems are the large diffusion of monotone schemes and the resulting poor resolution of discontinuities. A further goal of high-order methods is to compute more accurate solutions with similar computational cost. High-order accurate finite volume methods are based on the same derivation as the first order scheme, but we approximate the boundary integral in (2.43) by a quadrature rule of high order and replace the flux at each quadrature point with a high-order numerical flux with $p > 1$. Inserting this into the semi-discrete scheme (2.42), we obtain

$$\frac{\mathrm{d}U_i}{\mathrm{d}t} = -\frac{1}{|C_i|}\sum_{l_e=1}^{3} F_{il_e} = -\frac{1}{|C_i|}\sum_{l_e=1}^{3}\int_{S_{il_e}} f(u)\cdot \mathbf{n}_{il_e}\mathrm{d}\mathbf{s}(\mathbf{x}) + \mathcal{O}(\Delta x^p).$$

Using the diverence theorem we have

$$\begin{aligned}\frac{\mathrm{d}U_i}{\mathrm{d}t} &= -\frac{1}{|C_i|}\int_{\partial C_i} f(u)\cdot \mathbf{n}_{il_e}\mathrm{d}\mathbf{s}(\mathbf{x}) + \mathcal{O}(\Delta x^p),\\ &= -\frac{1}{|C_i|}\int_{C_i} f_1(u)_{\mathbf{x}_1} + f_2(u)_{\mathbf{x}_2}\mathrm{d}\mathbf{x} + \mathcal{O}(\Delta x^p),\end{aligned} \tag{2.46}$$

which ensures a local truncation error of order p. In the end, we can apply an arbitrary time discretization technique to recover a fully discrete scheme, e.g., an SSPRK method [49].

One way to generate a high-order approximation of the flux is the MUSCL approach [121]. We consider again a grid consisting of cells C_i, $i \in \mathbb{N}$ as in Figure 2.2. We define for each cell C_i a stencil S_i of neighbors and create a function $s_i : \mathbb{R}^d \to \mathbb{R}$ which is a reconstruction of the solution, that interpolates the solution in a mean value sense on the stencil. The interpolation problem with average values is formalized in (1.9). A high-order boundary integral approximation of (2.43) and the high-order accurate reconstruction s_i of the local solution are used to evaluate the first order flux $F(U, V, \mathbf{n}_{il_e})$ on the quadrature points. This high-order flux can be written as

$$F_{il_e} = \sum_{k=1}^{n_Q} \omega_k F^R_{il_e}(s_i(\mathbf{x}_k), s_{il_e}(\mathbf{x}_k), \mathbf{n}_{il_e}), \tag{2.47}$$

with the quadrature weights ω_k, the quadrature points $\mathbf{x}_k$ for $k = 1, \ldots, n_Q$ with $n_Q \in \mathbb{N}$ the number of quadrature points and the high-order accurate reconstruction s_i of the solution in cell C_i. The high-order reconstruction s_i is based on a stencil of cells which includes C_i.

A main problem with high-order accurate methods is the appearance of oscillations close to discontinuities. Figure 2.3 shows some classical behaviour of high-order methods at discontinuities illustrated by spurious oscillations of the second order Lax-Wendroff method

$$f(u, v) = \frac{f(u) + f(v)}{2} - \frac{\Delta t}{2\Delta x} f'\Big(\frac{u+v}{2}\Big)(f(v) - f(u)), \tag{2.48}$$

for the linear advection equation with a discontinuous initial condition.

Essentially nonoscillatory method

Harten et al. [57] proposed the essentially nonoscillatory method (ENO) to control spurious oscillations at discontinuities. Its principle is based on the evaluation of multiple stencils for each cell C_i, in which we reconstruct the solution for each component, see Figure 2.4. Finally, one chooses the least oscillatory reconstruction to define s_i.

For the one-dimensional polynomial reconstruction we assume a grid comprising the cells $C_i = (x_{i-1/2}, x_{i+1/2}]$ for $i \in \mathbb{N}$ with its averages U_i. The goal is to find

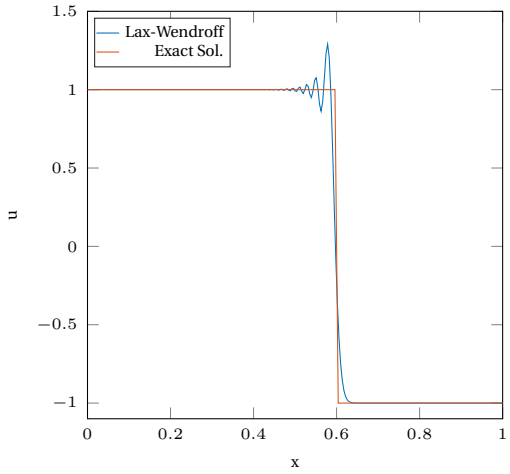


Figure 2.3 – Artificial Oscillations of the discontinuous solution for the linear advection equation with $a = 1$ at time $t = 0.1$ with $u_0(x) = -\operatorname{sgn}(x - 0.5)$ and $N = 256$ cells.

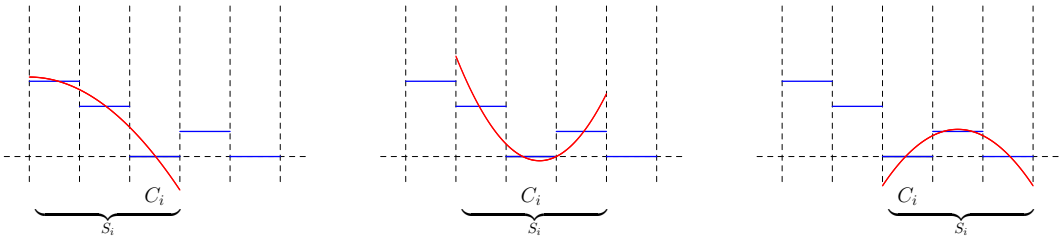


Figure 2.4 – Different stencils and its reconstruction with $n = 3$.

a function of order $n - 1$ on a stencil of size $n \in \mathbb{N}$ such that $\lambda_{C_j} s_i = U_j$ for all $j = i + r_{n-1}, \ldots, i + r_{n-1} + n - 1$, $1 - n \leqslant r_{n-1} \leqslant 0$. In the case of a polynomial reconstruction $s_i \in \Pi_{n-1}(\mathbb{R})$, with the polynomial space $\Pi_n(\mathbb{R})$ of maximum degree n, the problem can be rewritten into a pointwise interpolation. Therefore, we define the interpolation values

$$V_{i+1/2} = V_{i-1/2} + U_i |C_i|, \tag{2.49}$$

with $V_{-1/2} = C \in \mathbb{R}$ that can be interpreted as the step-wise primitive of our interpolation function. Let $S \in \Pi_n(\mathbb{R})$ be the interpolation polynomial such that

$$S(x_{i+1/2}) = V_{i+1/2}, \tag{2.50}$$

for all $i = -1, \ldots, n - 1$. Thus, the polynomial $s \in \Pi_{n-1}(\mathbb{R})$ defined by

$$s(x) = S'(x), \tag{2.51}$$

which fulfills condition (1.9).

Algorithm 2.1 Recursive Algorithm [57]

Let the interpolation points $x_{i-n+1/2}, \dots, x_{i+n-1/2}$ and its values $V_{i-n+1/2}, \dots, V_{i+n-1/2}$ be given.
Start by initializing $r_0 = -1/2$.
for $j = 1, \dots, n-1$ **do**
  if $|S[x_{i+r_{j-1}-1}, \dots, x_{i+r_{j-1}+j}]| < |S[x_{i+r_{j-1}}, \dots, x_{i+r_{j-1}+j+1}]|$ **then**
    Set $r_j = r_{j-1} - 1$
  else
    Set $r_j = r_{j-1}$
  end if
end for
Define the stencil $S_i = \{C_{i+r_{n-1}}, \dots, C_{i+r_{n-1}+n-1}\}$ and $s_j(x) = S'(x)$.

To choose the least oscillatory stencil there exist different possibilities. The stencil choice of the original method is given by Algorithm 2.1, and is based on the divided differences

$$S[x_i] = V_i, \qquad S[x_i, \dots, x_{i+j}] = \frac{S[x_{i+1}, \dots, x_{i+j}] - S[x_i, \dots, x_{i+j-1}]}{x_{i+j} - x_i}, \tag{2.52}$$

for polynomials. Note that the ENO method of order p is based on the reconstruction with stencils of size $n = p$. Harten [55] showed that for $w \in C^\infty(\mathbb{R})$ all the k-th divided differences are continuously depending on the k-th derivative. And if w has a jump discontinuity in the k-th derivative then it blows up with $\mathcal{O}(\Delta x^{-p+k})$ with $k \leqslant p$.
Multiple different methods have been proposed to measure the smoothness of polynomials, e.g., [86, 1, 70, 66]. All these reconstruction methods give reasonable results. One concept to show entropy-stability is based on a structural property which is called the sign property (1.11). The polynomial reconstruction based on Algorithm 2.1 fulfills the sign property, as shown in [36]. Furthermore, the minmod limiter is sign-stable [35].

Weighted essentially nonoscillatory method

The ENO method considers $2n-1$ cells to receive a reconstruction of degree $n-1 \in \mathbb{N}$ on a stencil of size n and a finite volume method of order $p = n$. However, by using all $2n-1$ cells the maximum degree we can hope for in the smooth case is $2n-2$ such that we end up with a finite volume method of order $p = 2n-1$. Liu et al. [86] introduced the weighted ENO method based on the idea of using a convex combination of the solutions s_i^j of each stencil $S_i^j = \{C_{i-j}, \dots, C_{i-j+n-1}\}$ for each $j = 0, \dots, n-1$ to create a stable finite volume method of order $p = 2n-1$.

Given $s_i^j : \mathbb{R} \to \mathbb{R}$ such that

$$\lambda_C s_i^j = U_C, \qquad \text{for all } C \in S_i^j, \text{ for each } j = 0, \ldots, n-1, \tag{2.53}$$

we define the reconstruction

$$s_i(x) = \sum_{j=0}^{n-1} \omega_i^j s_i^j(x), \tag{2.54}$$

such that $\omega_i^j = d_i^j + \mathcal{O}(\Delta x^{n-1})$ in smooth regions with the coefficients $d_i^j \in \mathbb{R}$ fulfilling

$$s_{i\pm1/2} = \sum_{j=0}^{n-1} d_i^j s_i^j(x_{i\pm1/2}) = u(x_{i\pm1/2}) + \mathcal{O}(\Delta x^{2n-1}). \tag{2.55}$$

The convexity property $\sum_{j=0}^{n-1} \omega_i^j = 1$ with $\omega_i^j \geqslant 0$ is needed to maintain consistency and stability. One of the most common choices for the nonlinear coefficients ω_i^j was proposed by Jiang and Shu [70]

$$\omega_i^j = \frac{\alpha_i^j}{\sum_{i_0=0}^{n} \alpha_i^{i_0}}, \qquad \alpha_i^j = \frac{d_i^j}{(\mathrm{IS}_{C_i}[s_i^j] + \bar{\varepsilon})^t}, \tag{2.56}$$

where $\bar{\varepsilon} \ll 1$ and the smoothness indicator $\mathrm{IS}_C : C^\infty(\mathbb{R}) \to \mathbb{R}$ which measures the smoothness of the reconstructions. To preserve the right order of accuracy in the smooth case we need

$$\mathrm{IS}_C[s] = C(\Delta x)(1 + \mathcal{O}(\Delta x^{n-1})). \tag{2.57}$$

In the case that the function s is not smooth in C we need

$$\mathrm{IS}_C[s] = \mathcal{O}(1). \tag{2.58}$$

In comparison with the ENO method one of the main additional challenges of the WENO method is the choice of the coefficients d_i^j, especially for unstructured grids. Concerning the sign property there exist just a few results for WENO methods, e.g., some special third order WENO methods [17, 37]. To solve multidimensional problems, there exists dimensional splitting which is a method to solve multidimensional problems with one-dimensional methods [48]. However, applying the dimensional splitting with high-order finite volume schemes does not directly result in a high-order method, but rather in a high-resolution method. To receive the right order of convergence the flux must be calculated for each quadrature point on the boundary of the quadrilateral. More information and analysis can be found in [70, 61].

2.2.3 Finite volume method for systems of equations

The derivation of the finite volume method for systems of equations is based on the same principle as the scalar case. After applying the divergence theorem, we end up with (2.42) with multiple components. The generalization of the MUSCL scheme is direct by reconstructing the solution for each component and use (2.47).
There are a number of possibilities to approximate the flux through the interfaces. Two classic numerical flux functions for one-dimensional problems include

- The Lax-Friedrichs (or Rusanov) flux that is defined by

$$F_{i+1/2} = \frac{f(\mathbf{U}_i) + f(\mathbf{U}_{i+1})}{2} - \frac{c_{i+1/2}}{2}(\mathbf{U}_{i+1} - \mathbf{U}_i), \tag{2.59}$$

  with $c_{i+1/2} = \max_{k=1,\dots,N}(|\lambda_k(\mathbf{U}_i)|, |\lambda_k(\mathbf{U}_{i+1})|)$ and $\lambda_k(\mathbf{U})$ the k-th eigenvalue of $\nabla_{\mathbf{U}} f(\mathbf{U})$.

- The Roe flux that is defined as

$$F_{i+1/2} = \frac{f(\mathbf{U}_i) + f(\mathbf{U}_{i+1})}{2} - \frac{1}{2} R_{i+1/2} |\Lambda_{i+1/2}| R^{-1}_{i+1/2}(\mathbf{U}_{i+1} - \mathbf{U}_i), \tag{2.60}$$

  with the eigenvector and eigenvalue matrices $R_{i+1/2}$, $\Lambda_{i+1/2} = \mathrm{diag}(\lambda_1, \dots, \lambda_N)$ of an approximated matrix $\tilde{A}(\mathbf{U}_i, \mathbf{U}_{i+1/2}) \approx A(\mathbf{U}) = \nabla_{\mathbf{U}} f(\mathbf{U})$. More details can be found in [105, 61].

The main difference between the finite volume method for scalar equations and systems of equations is the lack of theoretical support for systems of conservation laws, especially in multiple dimensions. Nevertheless, current methods seem to give reasonable results.
A further option to stabilize finite volume methods for systems of equations is the change of variables, e.g., entropy variables, characteristic variables.

2.3 Finite difference method

In comparison to the finite volume method the finite difference method seeks to approximate the partial derivative operator using some discrete pointwise approximation operator. For simplicity, we consider just the one-dimensional problem. Higher dimensional cases are derived by applying a dimensional splitting [48]. Unlike the finite volume scheme the finite difference method is node based and not cell based. However, the structure of the finite difference approximation of (2.1) can be written in the same way

$$\frac{\mathrm{d}\mathbf{u}_i}{\mathrm{d}t} + \frac{1}{\Delta x}(F_{i+1/2} - F_{i-1/2}) = 0, \tag{2.61}$$

where $\frac{F_{i+1/2}-F_{i-1/2}}{\Delta x} = \frac{\partial f}{\partial x}(\mathbf{u}_i) + \mathcal{O}(\Delta x^p)$ with $p > 0$. The flux terms depend on point values

$$F_{i+1/2} = F(\mathbf{u}_{i-k}, \dots, \mathbf{u}_{i+p-k-1}), \tag{2.62}$$

with $k \leqslant p$ and can be derived using Taylor expansions.
Similar to the high-order finite volume method reconstructions s_i are used to calculate high-order finite differences [114]. In this case, the reconstructions s_i are based on pointwise interpolation

$$s_i(x_j) = \mathbf{u}_j, \tag{2.63}$$

for all $j \in S_i$ with the stencil S_i.
Note that high-order for the finite difference method is slightly different from that of the finite volume case. We seek for a high-order approximation of the derivative

$$\frac{F_{i+1/2} - F_{i-1/2}}{\Delta x} = \frac{\partial f}{\partial x}(\mathbf{u}_i) + \mathcal{O}(\Delta x^p). \tag{2.64}$$

Inserting this into (2.61) we have the same result as for the finite volume method

$$\frac{\mathrm{d}\mathbf{u}_i}{\mathrm{d}t} = -\frac{1}{\Delta x}(F_{i+1/2} - F_{i-1/2}) = -\frac{\partial f}{\partial x}(\mathbf{u}_i) + \mathcal{O}(\Delta x^p),$$

in a pointwise sense. However, the idea of essentially nonoscillatory methods is working similarly with the difference that the reconstruction is based on pointwise interpolation. As in the mean-value case the reconstruction with the pointwise version of Algorithm 2.1 is sign-stable [36].

2.4 Entropy conservative and entropy stable finite difference methods

The goal is to construct methods that fulfill (2.38), referred to as *entropy stable* schemes [58, 81]. As a first step, we introduce *entropy conservative* methods that fulfill

$$\frac{\mathrm{d}}{\mathrm{d}t}\eta(\mathbf{u}_j) = -\frac{1}{|C_j|}[Q_{j+1/2} - Q_{j-1/2}]. \tag{2.65}$$

Next, we add a dissipation term to control oscillations at discontinuities to recover an entropy stable method. To construct entropy conservative methods we use Tadmor's entropy conservation condition [117]

$$[\![\mathbf{v}]\!]^T_{i+1/2} F_{i+1/2} = [\![\psi]\!]_{i+1/2}, \tag{2.66}$$

with the entropy variable $\mathbf{v}$, the entropy potential ψ and the notation $[\![f]\!]_{i+1/2} = f_{i+1} - f_i$. This condition describes a system of equations, but its solvability is generally not clear. However, for scalar conservation laws there exists a unique solution as summarized in the following theorem.

Theorem 2.3 (Entropy conservative schemes for scalar equations [117])**.** *For a given entropy pair (η, q) the numerical flux*

$$F_{i+1/2} = \begin{cases} \frac{[\![\psi]\!]_{i+1/2}}{[\![v]\!]_{i+1/2}} & \text{if } u_i \neq u_{i+1} \\ f(u_i) & \text{if } u_i = u_{i+1} \end{cases}, \tag{2.67}$$

defines an entropy conservative method for scalar equations with the entropy variable v and the conserved variable u. Furthermore, it is second-order accurate in smooth regions of u.

Given a numerical second order two-point flux the work of Lefloch et al. [82] combines these linearly to construct a $2p$-th order accurate flux on a uniform grid.

Theorem 2.4 (High-order entropy conservative fluxes [82])**.** *Let $p \in \mathbb{N}$ and assume that $\alpha_{1,p}, \dots, \alpha_{p,p}$ solve the p linear equations*

$$\sum_{i=1}^{p} i\alpha_{i,p} = 1, \qquad \sum_{i=1}^{p} i^{2s-1}\alpha_{i,p} = 0, \qquad \text{for } s = 2, \dots, p. \tag{2.68}$$

Then the flux

$$\tilde{F}^{2p}(u_{i-p+1}, \dots, u_{i+p}) = \sum_{j=1}^{p} \alpha_{j,p} \sum_{l=1}^{j} \tilde{F}^2(u_{i-j+l}, u_{i+l}), \tag{2.69}$$

is consistent, $2p$-th order accurate and entropy conservative provided the second order two-point conservative flux $\tilde{F}^2$ fulfills (2.66)*.*

The fourth order entropy conservative flux with coefficients $\alpha_2 = (\frac{4}{3}, -\frac{1}{6})$ and the sixth order scheme with $\alpha_3 = (\frac{3}{2}, -\frac{3}{10}, \frac{1}{30})$ present two explicit examples.

Remark 2.1. *In the original version of Theorem 2.4 the conditions* (2.68) *include a factor two*

$$2\sum_{i=1}^{p} i\alpha_{i,p} = 1, \tag{2.70}$$

which is wrong.
The proof is based on the Taylor expansion of $F(u_i, u_{i+j})$ and $F(u_{i-j}, u_i)$. Let us write

down the difference

$$F(u_i, u_{i+j}) - F(u_{i-j}, u_i) = jh\Big(\partial_x F(u_i, u_i) + \partial_y F(u_i, u_i)\Big) + \frac{(jh)^2}{2}\Big(\partial_{xx} F(u_i, u_i) - \partial_{yy} F(u_i, u_i)\Big) + \dots. \tag{2.71}$$

The terms with even order are zero because of the symmetry of F. The high-order odd terms are zero because

$$\sum_{i=1}^{p} i^{2s-1} \alpha_{i,p} = 0, \tag{2.72}$$

as in the original proof. To analyze the first term we apply the consistency condition to the second order numerical flux

$$\frac{\mathrm{d}}{\mathrm{d}u} f(u) = \frac{\partial}{\partial u} F(u, u) = \partial_x F(u, u) + \partial_y F(u, u). \tag{2.73}$$

Thus, we get

$$F(u_i, u_{i+j}) - F(u_{i-j}, u_i) = jh \frac{\mathrm{d}}{\mathrm{d}u} f(u_i) + 2 \sum_{k=1}^{p} \frac{(jh)^{2k+1}}{(2k+1)!} \frac{\partial^{2k+1}}{\partial x^{2k+1}} F(u_i, u_i) + \mathcal{O}(h^{2p+2}). \tag{2.74}$$

Using (2.68) *we end up with*

$$\frac{\tilde{F}^{2p}(u_{i-p+1}, \dots, u_{i+p}) - \tilde{F}^{2p}(u_{i-p}, \dots, u_{i+p-1})}{h} = \frac{\mathrm{d}}{\mathrm{d}u} f(u) + \mathcal{O}(h^{2p+1}). \tag{2.75}$$

2.4.1 Examples of 2nd-order entropy conservative schemes

Linear advection equation Let us apply Theorem 2.3 to the linear advection equation (2.3) with the entropy pair

$$\eta(u) = \frac{u^2}{2}, \qquad q(u) = a\frac{u^2}{2}. \tag{2.76}$$

For this choice we get

$$v(u) = u, \qquad \psi(u) = a\frac{u^2}{2}, \tag{2.77}$$

and we obtain the entropy conservative flux

$$\tilde{F}_{i+1/2} = \frac{u_{i+1} + u_i}{2}. \tag{2.78}$$

Burger's equation For the Burger's equation (2.4), the entropy pair

$$\eta(u) = \frac{u^2}{2}, \qquad q(u) = \frac{u^3}{3}, \tag{2.79}$$

we get

$$v(u) = u, \qquad \psi(u) = \frac{u^3}{6}. \tag{2.80}$$

This leads to the entropy conservative flux

$$\tilde{F}_{i+1/2} = \frac{u_i^2 + u_i u_{i+1} + u_{i+1}^2}{6}. \tag{2.81}$$

Shallow water equations For systems of equations it is more difficult to find entropy conservative fluxes, especially since the existence of solutions of the system (2.66) is unclear. For the one-dimensional shallow water equation the second order entropy conservative flux

$$\tilde{F}_{i+1/2} = \begin{pmatrix} \bar{h}_{i+1/2}\bar{u}_{i+1/2} \\ \bar{h}_{i+1/2}(\bar{u}_{i+1/2})^2 + \frac{1}{2}g\overline{h^2}_{i+1/2} \end{pmatrix}, \tag{2.82}$$

with $u = m/h$ and $\bar{f}_{i+1/2} = \frac{1}{2}(f_i + f_{i+1})$ is based on the entropy pair (2.29) [34].

Euler equations With the entropy pair (2.31) Chandrashekar [16] proposed the kinetic energy preserving and entropy conservative (KEPEC) flux for the one-dimensional Euler equations (2.9), based on the entropy variables and the potential

$$\mathbf{v} = \begin{pmatrix} \frac{\gamma - s}{\gamma - 1} - \frac{\rho u^2}{2p} \\ \rho u / p \\ -\rho / p \end{pmatrix}, \qquad \psi = \rho u. \tag{2.83}$$

The KEPEC flux makes use of the logarithmic averages $\hat{\rho}$ and $\hat{\beta}$ with $\beta = \frac{\rho}{2p}$ and can be written as

$$f^\rho = \hat{\rho}\bar{u}, \qquad f^m = \frac{\bar{\rho}}{2\bar{\beta}} + \bar{u}f^\rho, \qquad f^e = \Big(\frac{1}{2(\gamma-1)\hat{\beta}} - \frac{1}{2}\overline{u^2}\Big)f^\rho + \bar{u}f^m, \tag{2.84}$$

where $\bar{v} = \frac{v_{i+1}+v_i}{2}$ and the logarithmic average $\hat{v} = \frac{v_i - v_{i+1}}{\log(v_i) - \log(v_{i+1})}$.

2.4.2 Entropy stable fluxes

Entropy conservative methods yield good results in smooth regions, but it is well-known that spurious oscillations appear close to discontinuities. Introducing artificial dissipation, depending on the size of the jump over the interface, controls these oscillations. Based on an entropy conservative scheme $\tilde{F}_{i+1/2}$ and a symmetric positive definite matrix $D_{i+1/2}$, Tadmor [117] proposed the entropy stable numerical flux function

$$F_{i+1/2} = \tilde{F}_{i+1/2} - \frac{1}{2} D_{i+1/2} [\![v]\!]_{i+1/2}. \tag{2.85}$$

Combining high-order conservative fluxes with dissipation terms introduces the constraint that $D_{i+1/2}[\![v]\!]_{i+1/2} = \mathcal{O}(\Delta x^p)$ to maintain accuracy for smooth solutions.
To achieve this we define for each cell C_i a stencil of cells S_i on which we construct an interpolation function $s_i : \mathbb{R} \to \mathbb{R}$ of order p and replace the jump $[\![v]\!]_{i+1/2}$ by the jump in the reconstruction $\langle\!\langle v \rangle\!\rangle_{i+1/2} = s_{i+1}(x_{i+1/2}) - s_i(x_{i+1/2})$. Thus, the method has the form

$$F_{i+1/2} = \tilde{F}^{2p}_{i+1/2} - \frac{1}{2} D_{i+1/2} \langle\!\langle v \rangle\!\rangle_{i+1/2}, \tag{2.86}$$

with the additional condition

$$D_{i+1/2} = R_{i+1/2} \Lambda_{i+1/2} R^T_{i+1/2}, \tag{2.87}$$

where $R_{i+1/2} \in \mathbb{R}^{N \times N}$ is invertible and $\Lambda_{i+1/2} \geqslant 0$ is diagonal. Fjordholm et al. [35] recovered the following stability results.

Lemma 2.5 (Entropy stability with high-order diffusion [35])**.** *For each $i \in \mathbb{Z}$, let* (2.87) *be fulfilled. Let s_i be a reconstruction of the entropy variables in cell C_i, such that for each i, there exists a diagonal matrix $B_{i+1/2} \geqslant 0$ such that*

$$\langle\!\langle v \rangle\!\rangle_{i+1/2} = R^{-T}_{i+1/2} B_{i+1/2} R^T_{i+1/2} [\![v]\!]_{i+1/2}. \tag{2.88}$$

Then the scheme with the flux (2.86) *is entropy stable.*

By introducing the *scaled entropy variables*

$$w_i^{\pm} = R^T_{i\pm1/2} v_i, \qquad \tilde{w}_i^{\pm} R^T_{i\pm1/2} v_i^{\pm}, \tag{2.89}$$

with the reconstructed entropy variables $v_i^{\pm} = s_i(x_{i\pm1/2})$, (2.88) becomes

$$\langle\!\langle \tilde{w} \rangle\!\rangle_{i+1/2} = B_{i+1/2} [\![w]\!]_{i+1/2}. \tag{2.90}$$

Since $B_{i+1/2}$ is diagonal and semi-positive definite, this can be reformulated compo-

nentwise as

$$\operatorname{sgn}\langle\!\langle \tilde{w}^l \rangle\!\rangle_{i+1/2} = \operatorname{sgn}[\![w^l]\!]_{i+1/2}, \tag{2.91}$$

for each component l. We recognize this as the sign property (1.11).
Finally, we can combine the high-order conservative flux with the high-order ENO-based diffusion term and obtain a high-order, entropy stable, and essentially nonoscillatory class of finite difference schemes, called *TeCNO schemes* (see [35] for more details).
Two explicit examples include the combination with a high-order entropy conservative flux, based on the second order flux from Section 2.4.1, with the local Lax-Friedrichs (Rusanov) or the Roe diffusion terms:

ELLF: Let $A_{i+1/2} = \frac{1}{2} c_{i+1/2}\,\mathrm{Id}$, with $c_{i+1/2} = \max_{k=1,\dots,N}(|\lambda_k(u_i)|, |\lambda_k(u_{i+1})|)$.
This gives us the diffusion term

$$D_{i+1/2}\langle\!\langle v \rangle\!\rangle_{i+1/2} = \frac{1}{2} c_{i+1/2} R_{i+1/2} \langle\!\langle w \rangle\!\rangle_{i+1/2}. \tag{2.92}$$

ERoe: Let $A_{i+1/2} = \frac{1}{2}|\Lambda_{i+1/2}|$, with $\Lambda_{i+1/2} = \operatorname{diag}(\lambda_1(u_{i+1/2}), \dots, \lambda_N(u_{i+1/2}))$ and $u_{i+1/2} = \frac{u_i + u_{i+1}}{2}$. We obtain the diffusion term

$$D_{i+1/2}\langle\!\langle v \rangle\!\rangle_{i+1/2} = \frac{1}{2} R_{i+1/2} |\Lambda_{i+1/2}| \langle\!\langle w \rangle\!\rangle_{i+1/2}. \tag{2.93}$$

2.5 Entropy conservative and stable finite volume methods

Analogues to the finite difference method we can introduce discrete entropy conservation for the finite volume methods in one dimension. A finite volume method is called *entropy conservative* for a given entropy pair (η, q) if it satisfies (2.65) with some numerical entropy flux Q which is consistent with q. Note that for the two dimensional case Madrane et al. [87] introduced a different definition.

Definition 2.3 (Entropy conservative finite volume method [87])**.** *A numerical flux $F_{ij} = F(\mathbf{U}_i, \mathbf{U}_j, \mathbf{n}_{ij})$ is* entropy conservative *if it is of the form $F_{ij} = F^1_{ij} n^1_{ij} + F^2_{ij} n^2_{ij}$ and the components satisfy the relation*

$$[\![\mathbf{V}]\!]^T_{ij} F^k_{ij} = [\![\psi_k]\!]_{ij}, \qquad k = 1, 2, \tag{2.94}$$

where $\psi_k(\mathbf{U}) = \mathbf{V}(\mathbf{U})^T f_k(\mathbf{U}) - q_k(\mathbf{U})$ denotes the entropy potential.

In a second step, they proved the following theorem.

Theorem 2.6 (Entropy conservative finite volume method [87])**.** *Let F_{ij} be an entropy conservative flux. Then the approximate solution $\mathbf{U}_i$ satisfies the discrete entropy identity*

$$\frac{\mathrm{d}}{\mathrm{d}t}\eta(\mathbf{U}_i) + \frac{1}{|C_i|}\sum_{j\in\mathcal{N}_i} Q_{ij} = 0, \tag{2.95}$$

with the numerical entropy flux

$$Q_{ij} = \sum_{k=1}^{2} n_{ij}^k (\overline{\mathbf{V}}_{ij}^T F_{ij}^k - \overline{\psi^k}_{ij}), \tag{2.96}$$

with $\overline{w}_{ij} := \frac{1}{2}(w_i + w_j)$ *and the set of neighbouring cells* $\mathcal{N}_i$ *of cell* i*.*

Thus, the two definitions are consistent. Since the definition of entropy conservative schemes does not change for the finite volume method, we can conclude that a second order finite difference flux which fulfills (2.66) is also a second order flux for the entropy conservative finite volume method. This can be summarized as follows.

Theorem 2.7. *Every second order finite difference scheme that fulfills Tadmor's entropy conservation condition* (2.66) *in one space dimension is also a second order entropy conservative finite volume method.*

The construction of entropy stable schemes from entropy conservative schemes proceeds as for the finite difference case, the only difference being that the interpolation is based on cell averages instead of point values. Thus, Lemma 2.5 holds also for finite volume methods and we recover a second order accurate entropy stable finite volume method of the form

$$F_{i+1/2} = F^2_{i+1/2} - D_{i+1/2}\langle\!\langle v \rangle\!\rangle_{i+1/2}. \tag{2.97}$$

Remark 2.2. *The extension to higher order following Theorem 2.4 does not work in the finite volume case.*
Let us consider the approach

$$F_{i+1/2} = \sum_{l,k} \alpha_{l,k} F^2(U_{i+k}, U_{i+l}), \tag{2.98}$$

with the second order flux $F^2(U,V)$ *and a uniform grid in one dimension. The analysis*

is based on the Taylor expansions

$$\begin{aligned}U_{i+k} &= \frac{1}{h}\int_{x_{i+k-1/2}}^{x_{i+k+1/2}} u(x)\mathrm{d}x,\\ &= u(x_{i+1/2}) + \frac{u'(x_{i+1/2})}{2}\Big(k^2-(k-1)^2\Big)h + \frac{u''(x_{i+1/2})}{3!}\Big(k^3-(k-1)^3\Big)h^2\\ &\quad + \mathcal{O}(h^3),\\ &= u_{i+1/2} + \Delta u,\end{aligned}$$

and

$$\begin{aligned}F(u+\Delta u_1, u+\Delta u_2) = F(u,u) + F_x(u,u)(\Delta u_1+\Delta u_2) + \frac{F_{xx}}{2}(u,u)(\Delta u_1^2+\Delta u_2^2)\\ + F_{xy}(u,u)\Delta u_1\Delta u_2 + \mathcal{O}(\Delta u_1^3+\Delta u_2^3),\end{aligned}$$

where we use the symmetry of F*. The combination of the two expansions yields*

$$\begin{aligned}F(U_{i+k},U_{i+l}) &= f(u_{i+1/2}) + h\Big[F_x\Big|_{i+1/2}\frac{u'_{i+1/2}}{2}\Big(k^2-(k-1)^2+l^2-(l-1)^2\Big)\Big]\\ &\quad + h^2\Big[F_x\Big|_{i+1/2}\frac{u''_{i+1/2}}{3!}\Big(k^3-(k-1)^3+l^3-(l-1)^3\Big)\\ &\quad + \frac{1}{2}F_{xx}\Big|_{i+1/2}\Big(\frac{u'_{i+1/2}}{2}\Big)^2\Big((k^2-(k-1)^2)^2+(l^2-(l-1)^2)^2\Big)\\ &\quad + F_{xy}\Big|_{i+1/2}\Big(\frac{u'_{i+1/2}}{2}\Big)^2\Big((k^2-(k-1)^2)(l^2-(l-1)^2)\Big)\Big],\\ &= f(u_{i+1/2}) + h\Big[F_x\Big|_{i+1/2}\frac{u'_{i+1/2}}{2}\Big(2k+2l-2\Big)\Big]\\ &\quad + h^2\Big[3F_x\Big|_{i+1/2}\frac{u''_{i+1/2}}{3!}\Big(k^2-k+l^2-l+\frac{2}{3}\Big)\\ &\quad + 2F_{xx}\Big|_{i+1/2}\Big(\frac{u'_{i+1/2}}{2}\Big)^2\Big(k^2-k+l^2-l+\frac{1}{2}\Big)\\ &\quad + F_{xy}\Big|_{i+1/2}\Big(\frac{u'_{i+1/2}}{2}\Big)^2\Big(4kl-2k-2l+1\Big)\Big] + \mathcal{O}(h^3),\end{aligned}$$

where $u^{(j)}_{i+1/2} = u^{(j)}(x_{i+1/2})$, $F_x\Big|_{i+1/2} = F_x(u_{i+1/2},u_{i+1/2})$, $F_{xx}\Big|_{i+1/2} = F_{xx}(u_{i+1/2},u_{i+1/2})$, $F_{xy}\Big|_{i+1/2} = F_{xy}(u_{i+1/2},u_{i+1/2})$*. Using this result in* (2.98) *we obtain*

$$F_{i+1/2} = A_0 + A_1 h + (A_2^0 + A_2^1 + A_2^2)h^2 + \mathcal{O}(h^3), \tag{2.99}$$

with

$$A_0 = f(u_{i+1/2}) \sum_{l,k} \alpha_{l,k}, \tag{2.100}$$

$$A_1 = F_x\Big|_{u_{i+1/2}} u'_{i+1/2} \sum_{l,k} \alpha_{l,k}\Big(k + l - 1\Big), \tag{2.101}$$

$$A_2^0 = 3F_x\Big|_{u_{i+1/2}} \frac{u''_{i+1/2}}{3!} \sum_{l,k} \alpha_{l,k}\Big(k^2 - k + l^2 - l + \frac{2}{3}\Big), \tag{2.102}$$

$$A_2^0 = 2F_{xx}\Big|_{u_{i+1/2}} \Big(\frac{u'_{i+1/2}}{2}\Big)^2 \sum_{l,k} \alpha_{l,k}\Big(k^2 - k + l^2 - l + \frac{1}{2}\Big), \tag{2.103}$$

$$A_2^2 = F_{xy}\Big|_{u_{i+1/2}} \Big(\frac{u'_{i+1/2}}{2}\Big)^2 \sum_{l,k} \alpha_{l,k}\Big(4kl - 2k - 2l + 1\Big). \tag{2.104}$$

To receive a flux of order three we seek $\alpha_{l,k}$ *such that* $A_0 = f(u_{i+1/2})$ *and* $A_1 = A_2^i = 0$ *for* $i = 1, 2, 3$. *However,* (2.102) *and* (2.103) *can not both be zero.*

2.6 RBF-Theory

Radial basis functions (RBF) have been successfully used for scattered data interpolation. Due to their mesh-free property, they are more flexible in terms of the geometric structure of the data points. Furthermore, its application to high-dimensional problems is simple. Following the seminal work by Hardy [53], Duchon [25], and Micchelli [91], RBFs are successfully used in various domains.

2.6.1 Standard interpolation

The goal is the interpolation of a data vector $f|_X = (f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n))^T \in \mathbb{R}^n$ on the scattered set of data points $X = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^T$ with $\mathbf{x}_i \in \mathbb{R}^d$ for some function $f : \mathbb{R}^d \to \mathbb{R}$. We are seeking a function $s : \mathbb{R}^d \to \mathbb{R}$ such that

$$s(\mathbf{x}_i) = f(\mathbf{x}_i), \qquad \text{for all } i = 1, \ldots, n. \tag{2.105}$$

The idea is to use a single univariate continuous function ϕ, the *radial basis function,* composed with the Euclidean norm centered at the data points as the interpolation basis

$$\mathcal{B} = \{\phi(\varepsilon\|\mathbf{x} - \mathbf{x}_1\|), \ldots, \phi(\varepsilon\|\mathbf{x} - \mathbf{x}_n\|)\}, \tag{2.106}$$

with the shape parameter ε. To simplify the notation we use

$$\phi(\mathbf{x}-\mathbf{x}_i) := \phi(\varepsilon\|\mathbf{x}-\mathbf{x}_i\|), \qquad \phi : \mathbb{R}^d \to \mathbb{R}. \tag{2.107}$$

The general radial basis function approximation is given as

$$s(\mathbf{x}) = \sum_{i=1}^{n} a_i \phi(\mathbf{x}-\mathbf{x}_i) + p(\mathbf{x}), \tag{2.108}$$

with a polynomial $p(\mathbf{x}) = \sum_{j=1}^{m} b_j p_j(\mathbf{x})$ and the polynomial basis $\{p_1, \dots, p_m\}$ of $\Pi_{l-1}(\mathbb{R}^d)$, the space of polynomials in $\mathbb{R}^d$ of order $l-1$ with $l, m \in \mathbb{N}$, and the additional constraints

$$\sum_{i=1}^{n} a_i q(\mathbf{x}_i) = 0, \qquad \text{for all } q \in \Pi_{l-1}(\mathbb{R}^d), \tag{2.109}$$

with the coefficients $a_i \in \mathbb{R}$ for all $i = 1, \dots, n$. Conditions (2.105) and (2.109) can be expressed in the system of equations

$$\begin{pmatrix} A & P \\ P^T & 0 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} f|_X \\ 0 \end{pmatrix}, \tag{2.110}$$

with $A_{ij} = \phi(\mathbf{x}_i - \mathbf{x}_j)$, $P_{ij} = p_j(\mathbf{x}_i)$, $a = (a_1, \dots, a_n)^T$ and $b = (b_1, \dots, b_m)^T$. The choice of the radial basis function ϕ is restricted by some conditions to ensure the solvability of (2.110).

Definition 2.4 (Conditionally positive/negative definite function)**.** *A function $\phi : \mathbb{R}^d \to \mathbb{R}$ is called* conditionally positive (semi-) definite *of order l if for any pairwise distinct points $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^d$ and $c = (c_1, \dots, c_n)^T \in \mathbb{R}^n \backslash \{0\}$ such that*

$$\sum_{i=1}^{n} c_i p(\mathbf{x}_i) = 0, \tag{2.111}$$

for all $p \in \Pi_{l-1}(\mathbb{R}^d)$, the quadratic form

$$\sum_{j,k=1}^{n} c_j c_k \phi(\mathbf{x}_j - \mathbf{x}_k), \tag{2.112}$$

is positive (non-negative).
A function $\phi : \mathbb{R}^d \to \mathbb{R}$ is called conditionally negative (semi-) definite *of order l if $-\phi$ is conditionally positive (semi-) definite of order l.*

For a conditionally positive definite RBF ϕ of order r, (2.110) has a unique solution if $\mathbf{x}_1, \dots, \mathbf{x}_n$ are $\Pi_{l-1}(\mathbb{R}^d)$-unisolvent for $l \geqslant r$ [124]. A subclass of conditionally positive definite functions are the positive definite functions for which (2.112) holds but (2.111)

RBF	$\phi(r)$	Order
Infinitely smooth RBFs		
Multiquadratics	$(1+(\varepsilon r)^2)^{\nu}$	$\lceil \nu \rceil$
Inverse multiquadratics	$(1+(\varepsilon r)^2)^{-\nu}$	0
Gaussians	$\exp(-(\varepsilon r)^2)$	0
Piecewise smooth RBFs		
Polyharmonic Splines	r^{2k-d}	k
	$r^{2k-d}\log(r)$	k

Table 2.1 – Commonly used RBFs with $0 < \nu \notin \mathbb{N}$, $k \in \mathbb{N}$ and $\varepsilon > 0$.

does not.
Since the matrix A is positive definite for a positive definite function ϕ, the existence of an unique solution to (2.110) is trivial for all $l \in \mathbb{N}$, if $\mathbf{x}_1, \ldots, \mathbf{x}_n$ are pairwise disjoint. The most commonly used RBFs are listed in Table 2.1.

2.6.2 Interpolation of cell-averages

For the finite volume method we do not consider the pointwise interpolation, but the cell-averages. Let us assume a given grid of cells $C_1, \ldots, C_n$ with its average values $U_1, \ldots, U_n$ for $n \in \mathbb{N}$. Following [2, 3] we consider the interpolation function

$$s(x) = \sum_{i=1}^{n} a_i \lambda_{C_i}^{\xi} \phi(\mathbf{x} - \xi) + p(\mathbf{x}), \qquad p \in \Pi_{l-1}(\mathbb{R}^d), \tag{2.113}$$

with $\lambda_C^{\xi} f$ being the average operator of f over the cell C with respect to the variable ξ. Thus, we have the interpolation problem

$$\lambda_{C_j} s = U_j, \qquad \text{for all } j = 1, \ldots, n, \tag{2.114a}$$

$$\sum_{i=1}^{n} a_i \lambda_{C_i} q = 0, \qquad \text{for all } q \in \Pi_{l-1}(\mathbb{R}^d). \tag{2.114b}$$

Well-posedness

To show solvability of system (2.114), it suffices to assume ϕ to be conditional positive definite in a pointwise sense and $\{\lambda_{C_i}\}_{i=1}^n$ to be $\Pi_{l-1}(\mathbb{R}^d)$-unisolvent, see Theorem 2.8. We start by defining conditionally positive definiteness in terms of cell averages.

Definition 2.5 (Conditionally positive definite in the mean-value sense)**.** *A function $\phi : \mathbb{R}^d \to \mathbb{R}$ is called* conditionally positive definite *of order l in the mean-value sense if, for any stencil S of pairwise non-overlapping cells $C_1, \ldots, C_n \subset \mathbb{R}^d$ and $c =$*

$(c_1, \ldots, c_n)^T \in \mathbb{R}^n \backslash \{0\}$ *such that*

$$\sum_{i=1}^{n} c_i \lambda_{C_i} p = 0, \tag{2.115}$$

for all $p \in \Pi_{l-1}(\mathbb{R}^d)$*, the quadratic form*

$$\sum_{j,k=1}^{n} c_j c_k \lambda_{C_j}^{\mathbf{x}} \lambda_{C_k}^{\xi} \phi(\mathbf{x} - \xi), \tag{2.116}$$

is positive.
A function $\phi : \mathbb{R}^d \to \mathbb{R}$ *is called* conditionally negative definite *of order* l *in the mean-value sense if* $-\phi$ *is conditionally positive definite of order* l *in the mean-value sense.*

Remark 2.3. *An open question is if a conditionally positive definite function in the pointwise sense is also conditionally positive definite in the mean-value sense and vice versa.*

Following the result from Iske and Sonar [67], Theorem 2.8 gives us the well-posedness of (2.114) if the set $\{\lambda_{C_i}\}_{i=1}^n$ is $\Pi_{l-1}(\mathbb{R}^d)$-unisolvent.

Theorem 2.8 (Well-Posedness of RBF interpolation in the mean-value sense)**.** *Let* ϕ *be a conditionally positive definite radial basis function and let the set* $\{\lambda_{C_i}\}_{i=1}^n$ *be* $\Pi_{l-1}(\mathbb{R}^d)$*-unisolvent with* $n \in \mathbb{N}$*. Then,* (2.114) *has a unique solution.*

The proof of this theorem follows closely the one for the pointwise evaluation [124] with an additional estimate for the positive definiteness that is based on a pointwise results in [96]. Therefore, we need to introduce some new definitions. A continuous real-valued function g on $[0, \infty)$ is called *completely monotonic* on $(0, \infty)$ if it is in $C^\infty(0, \infty)$ and $(-1)^k g^{(k)}(x) \geqslant 0$ for $x \in (0, \infty), k \in \mathbb{N}_0$. The connection between completely monotonic and conditionally positive semi-definite functions is expressed in Theorem 2.9.

Theorem 2.9 (Micchelli [91])**.** *A function* $\phi(\mathbf{x}) = g(\|\mathbf{x}\|_2^2)$ *is conditionally positive semi-definite of order* m*, whenever* $g \in C[0, \infty) \cap C^\infty(0, \infty)$ *and* $(-1)^m g^{(m)}$ *is completely monotonic on* $(0, \infty)$*.*

Let us define the space $\mathcal{RP}_{m,c}^\infty$ as the space of all continuous functions $F : [0, \infty) \to \mathbb{R}$ such that $(-1)^m (\mathrm{d}^m / \mathrm{d}\sigma^m) F(\sqrt{\sigma})$ is completely monotonic on $(0, \infty)$. Note that the RBFs from the Table 2.1 belong to this class of conditionally positive semi-definite functions.
To prove Theorem 2.8 we need to find a bound from below of the quadratic form

$$Q := \sum_{j,k=1}^{n} \xi_j \xi_k \lambda_{C_j}^{\mathbf{x}} \lambda_{C_k}^{\mathbf{y}} \phi(\|\mathbf{x} - \mathbf{y}\|), \tag{2.117}$$

in the case that $\sum_{i=1}^{n} \xi_i \lambda_{C_i}^{\mathbf{x}} p(\mathbf{x}) = 0$ for all $p \in \Pi_{m-1}(\mathbb{R}^d)$. We use some results from Micchelli [91] and Narcowich and Ward [96] in terms of mean-value evaluation, stated below.

Lemma 2.10 (Mean-value version [91])**.** *If* $\sum_{i=1}^{n} a_i \lambda_{C_i}^{\mathbf{x}} p(\mathbf{x}) = 0$ *for all* $p \in \Pi_{k-1}(\mathbb{R}^d)$ *then*

$$(-1)^k \sum_{i,j=1}^{n} a_i a_j \lambda_{C_i}^{\mathbf{x}} \lambda_{C_j}^{\xi} \|\mathbf{x} - \xi\|^{2k} \geqslant 0, \tag{2.118}$$

where equality holds in (2.118) *if and only if* $\sum_{i=1}^{n} a_i \lambda_{C_i}^{\mathbf{x}} p(\mathbf{x}) = 0$ *for all* $p \in \Pi_k(\mathbb{R}^d)$.

The proof of Lemma 2.10 follows that of the original pointwise result. To find a more practical representation of the quadratic form Q (as stated in Corollary 2.12) we need Lemma 2.11.

Lemma 2.11 (Representation Lemma [96])**.** *Let* $g : [0, \infty) \to \mathbb{R}$ *be continuous on* $[0, \infty)$ *and let* $\varepsilon > 0$ *be arbitrary. If* $(-1)^m g^{(m)}$ *is completely monotonic on* $(0, \infty)$*, then on* $[0, \infty)$ *there exists a nonnegative Borel Measure* $\mathrm{d}\eta(t)$ *for which*

$$\begin{aligned} g(\sigma) = &\sum_{j=0}^{m-1} \frac{g^{(j)}(\varepsilon)(\sigma - \varepsilon)^j}{j!} \\ &+ \int_0^{\infty} \frac{1}{t^m} \left\{ e^{-\sigma t} - \Big(\sum_{j=0}^{m-1} \frac{(-1)^j (\sigma - \varepsilon)^j}{j!} t^j \Big) e^{-\varepsilon t} \right\} \mathrm{d}\eta(t). \end{aligned} \tag{2.119}$$

Based on this two results we have the following corollary.

Corollary 2.12 (Mean-value version [96])**.** *Let* $\phi \in \mathcal{RP}_{m,c}^{\infty}$ *and let* Q *be the associated quadratic form from above. Provided* $\xi = (\xi_1, \ldots, \xi_n)^T$ *satisfy* $\sum_{i=1}^{n} \xi_i \lambda_{C_i}^{\mathbf{x}} p(\mathbf{x}) = 0$ *for all* $p \in \Pi_{m-1}(\mathbb{R}^d)$*, it follows that*

$$Q = \sum_{j,k=1}^{n} \xi_j \xi_k \lambda_{C_j}^{\mathbf{x}} \lambda_{C_k}^{\mathbf{y}} \phi(\|\mathbf{x} - \mathbf{y}\|) = \int_0^{\infty} \frac{Q_t}{t^m} \mathrm{d}\eta(t), \tag{2.120}$$

where $Q_t = \sum_{j,k=1}^{n} \xi_j \xi_k \lambda_{C_j}^{\mathbf{x}} \lambda_{C_k}^{\mathbf{y}} \exp(-\|\mathbf{x} - \mathbf{y}\|^2 t)$.

Proof. From Lemma 2.11 we obtain

$$\begin{aligned}\phi(\|\mathbf{x}-\mathbf{y}\|) &= g(\|\mathbf{x}-\mathbf{y}\|^2),\\ &= \sum_{j=0}^{m-1} \frac{g^{(j)}(\varepsilon)(\|\mathbf{x}-\mathbf{y}\|^2-\varepsilon)^j}{j!}\\ &\quad + \int_0^\infty \frac{1}{t^m}\left\{ e^{-\|\mathbf{x}-\mathbf{y}\|^2 t} - \Big(\sum_{j=0}^{m-1} \frac{(-1)^j(\|\mathbf{x}-\mathbf{y}\|^2-\varepsilon)^j}{j!} t^j\Big) e^{-\varepsilon t}\right\} \mathrm{d}\eta(t).\end{aligned}$$

After exchanging the integral and the summation and applying Lemma 2.10 we recover the final result. □

Given the new representation of the quadratic form Q, Theorem 2.13 gives us a lower bound of Q_t.

Theorem 2.13 (Mean-value version [96])**.** *For every $\xi \in \mathbb{R}^n$ and every $t \geqslant 0$, there holds*

$$Q_t \geqslant \theta(t)\|\xi\|^2, \tag{2.121}$$

with

$$\theta(t) = C_s t^{-d/2} q^{-d} e^{-\delta^2 q^{-2} t^{-1}} \qquad \text{where } C_s = \frac{\delta^d}{2^{d+1}\Gamma((d+2)/2)},$$

and with a circumsribed radius of one cell $d_0^{\max}$

$$\delta = 4\left(\frac{d_0^{\max}}{q} + 3\right)\left(\frac{\pi\Gamma((d+2)/2)^2}{(3d_0^{\max}/q+9)(2\alpha_n/\chi(0)-1)}\right)^{1/(d+1)}, \tag{2.122}$$

and the positive α_n given by (2.129)*.*

We should note that compared to the pointwise case the result differs only in (2.122). By choosing $d_0^{\max} = 0$ and $\alpha_n/\chi(0) = 1$ the result coincides with the one in [96]. Before showing Theorem 2.13, we need to state some more results developed by Narcowich and Ward [96].

Lemma 2.14 (Narcowich and Ward [96])**.** *Using the Fourier transform of $e^{-r^2 t}$ over $\mathbb{R}^d$ we receive the representation*

$$\exp(-r^2 t) = \int_0^\infty \Omega_d(ur)\mathrm{d}\gamma_t(u), \tag{2.123}$$

with

$$\begin{aligned}
\mathrm{d}\gamma_t(u) &= \omega_{d-1}(4\pi)^{-d/2}t^{-d/2}e^{-u^2/4t}u^{d-1}\mathrm{d}u,\\
\Omega_d(z) &= \Gamma(d/2)(2/z)^{(d-2)/2}J_{(d-2)/2}(z),\\
\omega_{d-1} &= \frac{2\pi^{d/2}}{\Gamma(d/2)},
\end{aligned}$$

where Γ *denotes the Gamma function and* J_p *the order* p *Bessel function of first kind.*

Lemma 2.15 (Narcowich and Ward [96])**.** *Let* $\beta > 0$*. The function* $\chi : \mathbb{R}^d \to \mathbb{R}$ *defined by*

$$\chi(\mathbf{x}) = K\left(\frac{\beta}{2\pi\|\mathbf{x}\|}\right)^d J^2_{d/2}(\|\mathbf{x}\|\beta), \tag{2.124}$$

with $K = d\left(\frac{\pi}{t\beta^2}\right)^{d/2}e^{-\beta^2/t}$ *has a Fourier transform* $\hat{\chi}$ *that satisfies*

(i) $\hat{\chi} > 0$*,*

(ii) $\hat{\chi}$ *is radial function,*

(iii) $\mathrm{d}\gamma_t(u) \geqslant \hat{\chi}(u)u^{d-1}\mathrm{d}u$*.*

In particular, there holds

$$\chi(0) = s\left(\frac{\beta^2}{16\pi t}\right)^{d/2} e^{-\beta^2/t}\Gamma\left(\frac{d+2}{d}\right)^{-2}. \tag{2.125}$$

Lemma 2.16 (Narcowich and Ward [96])**.** *For* $d = 1, 2, \ldots$*, and for all* $z > 0$*, there holds*

$$J^2_{d/2}(z) \leqslant \frac{2^{d+2}}{z\pi}. \tag{2.126}$$

Next, we can prove Theorem 2.13.

Proof. (Theorem 2.13)
Given the definition of the form Q_t we use Lemma 2.14 and obtain

$$\begin{aligned}
Q_t &= \sum_{j,k=1}^{n} \xi_j\xi_k\lambda^{\mathbf{x}}_{C_j}\lambda^{\mathbf{y}}_{C_k}\exp(-\|\mathbf{x}-\mathbf{y}\|^2t),\\
&= \sum_{j,k=1}^{n} \xi_j\xi_k\lambda^{\mathbf{x}}_{C_j}\lambda^{\mathbf{y}}_{C_k}\int_0^\infty \Omega_d(u\|\mathbf{x}-\mathbf{y}\|)\mathrm{d}\gamma_t(u).
\end{aligned}$$

Next, we introduce the Fourier transform $\hat{\chi}(u)$ from Lemma 2.15 and we get

$$Q_t \geqslant \sum_{j,k=1}^{n} \xi_j \xi_k \lambda_{C_j}^{\mathbf{x}} \lambda_{C_k}^{\mathbf{y}} \int_0^\infty \Omega_d(u\|\mathbf{x}-\mathbf{y}\|)\hat{\chi}(u)u^{d-1}\mathrm{d}u.$$

From Wells and Williams [125, p.26] we know

$$\Omega_d(\mathbf{z}) = \omega_{d-1}^{-1} \int_{S_{d-1}} e^{\langle \mathbf{z},\eta\rangle}\mathrm{d}\sigma_{d-1}(\eta), \tag{2.127}$$

and we receive

$$\begin{aligned}
Q_t &\geqslant \sum_{j,k=1}^{n} \xi_j \xi_k \lambda_{C_j}^{\mathbf{x}} \lambda_{C_k}^{\mathbf{y}} \int_0^\infty \omega_{d-1}^{-1} \int_{S_{d-1}} e^{\langle \mathbf{x}-\mathbf{y},\eta\rangle}\mathrm{d}\sigma_{d-1}(\eta)\hat{\chi}(u)u^{d-1}\mathrm{d}u, \\
&= \omega_{d-1}^{-1} \sum_{j,k=1}^{n} \xi_j \xi_k \lambda_{C_j}^{\mathbf{x}} \lambda_{C_k}^{\mathbf{y}} \underbrace{\int_{\mathbb{R}^d} e^{\langle \mathbf{x}-\mathbf{y},\zeta\rangle}\hat{\chi}(\|\zeta\|)\mathrm{d}\zeta}_{(2\pi)^d\chi(\|\mathbf{x}-\mathbf{y}\|)}, \\
&= \frac{(2\pi)^d}{\omega_{d-1}}\Big[\sum_{j=1}^{n} |\xi_j|^2 \lambda_{C_j}^{\mathbf{x}} \lambda_{C_j}^{\mathbf{y}} \chi(\|\mathbf{x}-\mathbf{y}\|) + \sum_{j\neq k} \xi_j \xi_k \lambda_{C_j}^{\mathbf{x}} \lambda_{C_k}^{\mathbf{y}} \chi(\|\mathbf{x}-\mathbf{y}\|)\Big].
\end{aligned}$$

Next, we use Young's inequality and obtain the estimate

$$\begin{aligned}
Q_t &\geqslant \frac{(2\pi)^d}{\omega_{d-1}}\Big[\sum_{j=1}^{n} |\xi_j|^2 \lambda_{C_j}^{\mathbf{x}} \lambda_{C_j}^{\mathbf{y}} \chi(\|\mathbf{x}-\mathbf{y}\|) - \sum_{j\neq k} \frac{1}{2}(|\xi_j|^2 + |\xi_k|^2) \lambda_{C_j}^{\mathbf{x}} \lambda_{C_k}^{\mathbf{y}} \chi(\|\mathbf{x}-\mathbf{y}\|)\Big], \\
&= \frac{(2\pi)^d}{\omega_{d-1}}\Big[\sum_{j=1}^{n} |\xi_j|^2 \lambda_{C_j}^{\mathbf{x}} \lambda_{C_j}^{\mathbf{y}} \chi(\|\mathbf{x}-\mathbf{y}\|) - \sum_{j\neq k} |\xi_j|^2 \lambda_{C_j}^{\mathbf{x}} \lambda_{C_k}^{\mathbf{y}} \chi(\|\mathbf{x}-\mathbf{y}\|)\Big], \\
&\geqslant \frac{(2\pi)^d}{\omega_{d-1}}\Big[\alpha_n - \gamma_n\Big] \sum_{j=1}^{n} |\xi_j|^2,
\end{aligned} \tag{2.128}$$

with

$$\alpha_n = \min_j \lambda_{C_j}^{\mathbf{x}} \lambda_{C_j}^{\mathbf{y}} \chi(\|\mathbf{x}-\mathbf{y}\|), \tag{2.129}$$

$$\gamma_n = \max_k \sum_{j=1, j\neq k}^{n} \lambda_{C_j}^{\mathbf{x}} \lambda_{C_k}^{\mathbf{y}} \chi(\|\mathbf{x}-\mathbf{y}\|) = \sum_{j=1, j\neq k_0}^{n} \lambda_{C_j}^{\mathbf{x}} \lambda_{C_{k_0}}^{\mathbf{y}} \chi(\|\mathbf{x}-\mathbf{y}\|), \tag{2.130}$$

where $k_0 = \operatorname{argmax}_k \sum_{j=1, j\neq k}^{n} \lambda_{C_j}^{\mathbf{x}} \lambda_{C_k}^{\mathbf{y}} \chi(\|\mathbf{x}-\mathbf{y}\|)$. So far, the proof follows that of the pointwise method. The only difference is that in the cell average case the estimate for $\alpha_n - \gamma_n$ is more complex.
Since $\chi \in L_1(\mathbb{R})$,the following estimate holds.

For every $i \leqslant n, i \neq k_0$ there exists $\mathbf{x}_i \in C_i \backslash \partial C_i$ and $\mathbf{x}_0^i \in C_{k_0} \backslash \partial C_{k_0}$ such that

$$\lambda_{C_i}^{\mathbf{x}} \lambda_{C_{k_0}}^{\mathbf{y}} \chi(\|\mathbf{x}-\mathbf{y}\|) \leqslant \chi(\|\mathbf{x}_i - \mathbf{x}_0^i\|). \tag{2.131}$$

Let us introduce some definitions

$$q := \min\Big\{ \min_{i\neq j} \|\mathbf{x}_i - \mathbf{x}_j\|, \min_i \min_{\mathbf{x}\in C_{k_0}} \|\mathbf{x}-\mathbf{x}_i\| \Big\} \in \mathbb{R}, \tag{2.132}$$

$$\delta(\mathbf{x}_j, C_{k_0}) := \min_{\mathbf{x}\in C_{k_0}} \|\mathbf{x}-\mathbf{x}_j\| \in \mathbb{R}, \tag{2.133}$$

$$\tilde{\varepsilon}_i := \big\{ \mathbf{x}_j \mid 1 \leqslant j \leqslant n, iq \leqslant \delta(\mathbf{x}_j, C_{k_0}) \leqslant (i+1)q \big\} \subset \mathbb{R}^d, \tag{2.134}$$

$$\kappa_i := \sup\{ \sup_{\mathbf{y}\in C_{k_0}} \chi(\|\mathbf{x}-\mathbf{y}\|) \mid nq \leqslant \delta(\mathbf{x}, C_{k_0}) \leqslant (n+1)q \} \in \mathbb{R}. \tag{2.135}$$

We get an upper bound

$$\begin{aligned} \gamma_n &\leqslant \sum_{j=1, j\neq k_0}^{n} \chi(\|\mathbf{x}_j - \mathbf{x}_0^j\|) \leqslant \sum_{j=1, j\neq k_0}^{n} \sup_{\mathbf{x}\in C_{k_0}} \chi(\|\mathbf{x}_j - \mathbf{x}\|), \\ &= \sum_{i=1}^{\infty} \operatorname{card}(\tilde{\varepsilon}_i)\kappa_i. \end{aligned} \tag{2.136}$$

To bound the cardinality of $\tilde{\varepsilon}_n$ and κ_n, we inscribe a ball with center $\mathbf{x}_0$ and radius $d_0^{\min}$ into C_{k_0} and set $d_0^{\max} := \max_{\mathbf{x}\in C_{k_0}} \|\mathbf{x}-\mathbf{x}_{k_0}\|$ that can describe a cirumscribed circle. The idea is to estimate the cardinality of $\tilde{\varepsilon}_i$. It is clear that $\tilde{\varepsilon}_i$ lie between the shell around $\mathbf{x}_0$ with radius $d_0^{\max} + q(i+2)$ and the smaller one with radius $d_0^{\min} + q(i-1)$. Since all points $\mathbf{x}_j$ have at least the distance q, we can find an upper bound of the cardinality of $\tilde{\varepsilon}_i$ by using the ratio of the volume between the shells divided by the volume of the ball with diameter q. Thus, we get

$$\begin{aligned} \operatorname{card}(\tilde{\varepsilon}_i) &\leqslant \Big(\frac{d_0^{\max}}{q} + i + 2\Big)^d - \Big(\frac{d_0^{\min}}{q} + i - 1\Big)^d \leqslant \Big(\frac{d_0^{\max}}{q} + i + 2\Big)^d - (i-1)^d, \\ &\leqslant \Big(\frac{d_0^{\max}}{q} + 3\Big)^d i^{d-1}. \end{aligned}$$

With Lemma 2.15 and 2.16 we recover the upper bound of κ_i

$$\kappa_i \leqslant \frac{4d}{\beta} t^{-d/2} \pi^{-1-d/2} e^{-\beta^2/t} (qi)^{-1-d}, \tag{2.137}$$

and use this to estimate the value γ_n

$$\begin{aligned} \gamma_n &\leqslant \frac{4d}{\beta}\Big(\frac{d_0^{\max}}{q} + 3\Big)^d t^{-d/2} \pi^{-1-d/2} e^{-\beta^2/t} q^{-1-d} \sum_{i=1}^{\infty} i^{-2}, \\ &= \frac{2d}{3\beta}\Big(\frac{d_0^{\max}}{q} + 3\Big)^d t^{-d/2} \pi^{1-d/2} e^{-\beta^2/t} q^{-1-d} =: \frac{C(t)}{\beta}. \end{aligned}$$

This yields

$$Q_t \geqslant \frac{(2\pi)^d}{\omega_{d-1}} \left[\alpha_n - \frac{C(t)}{\beta} \right] \sum_{j=1}^{n} |\xi_j|^2, \tag{2.138}$$

with the free parameter $\beta > 0$. We choose β such that

$$\frac{\chi(0)}{2} = \alpha_n - \frac{C(t)}{\beta}. \tag{2.139}$$

Thus, we recover the formula from the pointwise case

$$\theta(t) = C_s t^{-d/2} q^{-d} e^{-\delta^2 q^{-2} t^{-1}}, \qquad \text{where } C_s = \frac{\delta^d}{2^{d+1}\Gamma((d+2)/2)},$$

with the only difference that

$$\delta = 4\left(\frac{d_0^{\max}}{q} + 3 \right) \left(\frac{\pi \Gamma((d+2)/2)^2}{(3d_0^{\max}/q + 9)(2\alpha_n/\chi(0) - 1)} \right)^{1/(d+1)}.$$

□

Theorem 2.17 (Lower bound of the quadratic form)**.** *Let $\phi \in \mathcal{RP}^{\infty}_{m,c}$, $\xi = (\xi_1, \ldots, \xi_n)^T$ satisfy $\sum_{i=1}^n \xi_i \lambda^{\mathbf{x}}_{C_i} p(\mathbf{x}) = 0$ for all $p \in \Pi_{m-1}(\mathbb{R}^d)$, and set*

$$\Theta := \int_0^{\infty} \frac{\theta(t)}{t^m} \mathrm{d}\eta(t), \tag{2.140}$$

where $\theta(t)$ appears in Theorem 2.13 and $\mathrm{d}\eta(t)$ is given in Lemma 2.11. Then, provided $\mathrm{d}^m/\mathrm{d}t^m \phi(\sqrt{t})$ is nonconstant, there holds

$$Q \geqslant \Theta \|\xi\|^2. \tag{2.141}$$

Proof. This theorem follows directly from Corollary 2.12 and Theorem 2.13. The condition that $\mathrm{d}^m/\mathrm{d}t^m \phi(\sqrt{t})$ is nonconstant ensures that Θ is positive. □

Finally, we can show that the mean-value interpolation problem is well-posed.

Proof. (Theorem 2.8)
After proving all pointwise results for cell averages, the proof for the unique solvability of (2.114) follows closely that of the pointwise result [124]. Let us consider (2.114) with

$U_j = 0$. Thus, we have

$$Aa + Pb = 0, \tag{2.142a}$$
$$P^T a = 0. \tag{2.142b}$$

Multiplying a^T from the left to (2.142a) gives us

$$a^T Aa = 0. \tag{2.143}$$

By Theorem 2.17 we get $a = 0$ and thus $Pb = 0$. The $\Pi_{l-1}(\mathbb{R}^d)$-unisolvency gives us $b = 0$. □

2.6.3 Ill-conditioning and VVRA-method

Despite the simple concept of RBF-interpolation in multiple dimensions, there is a major drawback, often referred to as the Uncertainty Principle [107]. It describes the trade-off between the well-known properties that flat infinitely smooth RBFs ($\varepsilon \to 0$) have increasing approximation power but decreasing numerical stability due to ill-conditioning of the interpolation matrix and a resulting stagnation (saturation) of the error under refinement [24, 78, 109].
To overcome the issue of ill-conditioning there are several propositions for choosing an optimal shape parameter [33, 104]. The approach of a continuous scaling $\varepsilon = \alpha n^{-1/d}$ causes stagnation errors [13]. However, there are multiple approaches which overcome this problem: the RBF-CP [42], the RBF-QR [41], and the RBF-GA [40]. Furthermore, there is the vector-valued rational approximation method (RBF-RA), based on the RBF-CP algorithm and introduced in [127]. A different way to overcome the stagnation error is the augmentation with polynomials [39, 9, 8]. In this case, the polynomials take over the role for the interpolation and the RBFs ensure solvability of (2.110). Note that our application to finite volume methods is of slightly different nature than most of the RBF applications, since we generally use small stencils.

Vector-valued rational approximation

The vector-valued rational approximation is not restricted to RBF-interpolation, but can be applied to approximation problems that satisfy certain conditions. Let us assume a vector-valued function $f : \mathbb{C} \to \mathbb{C}^M$, with $M > 1$. All components $f_j(\varepsilon)$ for $j = 1, \ldots, M$ are analytic in a domain Ω around the origin except for a finite number of isolated poles such that

(i) all M-components of f share the same singular points,

(ii) the direct numerical evaluation of f is possible for $|\varepsilon| \geqslant \varepsilon_R > 0$, where $|\varepsilon| \leqslant \varepsilon_R$ is in Ω,

(iii) $\varepsilon = 0$ is at most a removable singularity of f,

(iv) the function f is even.

The goal is to construct a Padé approximant $r_j(\varepsilon)$ with the same denominator for each component and its interpolation points $\varepsilon_j = \varepsilon_R e^{\pi j/K}$ for $K \in \mathbb{N}$. Condition (iv) is not mandatory, but it results in an even rational approximation

$$r_j(\varepsilon) = \frac{\sum_{i=0}^{m} a_{i,j}\varepsilon^{2i}}{1 + \sum_{i=1}^{n} b_i \varepsilon^{2i}} \approx f_j(\varepsilon), \tag{2.144}$$

for $j = 1, \ldots, M$ and $m, n \in \mathbb{N}$. Furthermore, it is fulfilled by RBFs. The interpolation problem can be described for each component by the system

$$\begin{pmatrix} 1 & \varepsilon_1^2 & \cdots & \varepsilon_1^{2m} \\ 1 & \varepsilon_2^2 & \cdots & \varepsilon_2^{2m} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \varepsilon_K^2 & \cdots & \varepsilon_K^{2m} \end{pmatrix} \begin{pmatrix} a_{0,j} \\ \vdots \\ a_{m,j} \end{pmatrix} + \operatorname{diag}(-f_j) \begin{pmatrix} \varepsilon_1^2 & \cdots & \varepsilon_1^{2n} \\ \varepsilon_2^2 & \cdots & \varepsilon_2^{2n} \\ \vdots & \ddots & \vdots \\ \varepsilon_K^2 & \cdots & \varepsilon_K^{2n} \end{pmatrix} \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix} = \begin{pmatrix} f_j(\varepsilon_1) \\ \vdots \\ f_j(\varepsilon_K) \end{pmatrix}, \tag{2.145}$$

with $(m+1)M + n$ unknown coefficients. We write (2.145) as

$$Ea_j + F_j b_j = f_j, \tag{2.146}$$

and choose $K > m + 1 + n/M$ to define an overdetermined system of equations. Algorithm 2.2 solves this system of equations. Note that there remains the choice of the parameters $n, m, K \in \mathbb{N}$ and $\varepsilon_R \in \mathbb{R}$.

RBF-RA

In the case of RBF-interpolation let $\hat{x}_1, \ldots, \hat{x}_M$ be the evaluation points and consider the approximation problem

$$f(\varepsilon) = \begin{pmatrix} s(\hat{x}_1, \varepsilon) \\ \vdots \\ s(\hat{x}_M, \varepsilon) \end{pmatrix} = \underbrace{\begin{pmatrix} \phi_\varepsilon(\|\hat{x}_1 - x_1\|) & \cdots & \phi_\varepsilon(\|\hat{x}_1 - x_n\|) \\ \vdots & \ddots & \vdots \\ \phi_\varepsilon(\|\hat{x}_M - x_1\|) & \cdots & \phi_\varepsilon(\|\hat{x}_M - x_n\|) \end{pmatrix}}_{\Phi(\varepsilon)} \begin{pmatrix} & A(\varepsilon)^{-1} & \end{pmatrix} f|_X. \tag{2.148}$$

Note that the evaluation points are fixed and the shape parameter ε is the variable of interest. Since $\phi_\varepsilon(r)$ is an analytic function in ε, all entries of $\Phi(\varepsilon)$ are analytic. In the same manner, all entries of $A(\varepsilon)$ are analytic near the origin and they have only isolated zeros. So, the entries of $A(\varepsilon)^{-1}$ are analytic with at most isolated poles and thus in

Algorithm 2.2 Vector-valued rational approximation [127]

(i) We normalize the system by dividing each row of E, F_j and f_j by $\|f(\varepsilon_k)\|_\infty$. Then, we compute a QR-decomposition of the modified E.

(ii) We multiply (2.145) with the Hermitian transpose Q^* from the left to obtain

$$\begin{pmatrix} R \\ 0 \end{pmatrix} a_j + Q^* F_j b = Q^* f_j. \tag{2.147}$$

(iii) We reorder the equations such that all $\bar{k} = \operatorname{rank} R$ equations of each component are first and all remaining put in the end. This gives us an almost upper triangular system with a full matrix block $\dot{F}$ of size $M(K-(m+1)) \times n$ and the corresponding rows of the right hand side $\dot{f}$.

(iv) We compute the least square solution of the overdetermined system of equations $\dot{F}b = \dot{f}$.

(v) By using the coefficients b we can solve the upper triangular systems to recover the remaining coefficients a_j for $j = 1, \ldots, M$.

compact domains there are at most a finite number of isolated poles. Furthermore, all entries of $s(\varepsilon)$ share the same poles since they are all dependent on $A(\varepsilon)^{-1}$. Further, the condition that $\varepsilon = 0$ is a removable singularity is typically and, in the case of the Gaussian kernel, is fulfilled [43, 109].
Condition (ii) is the only one that may not be fulfilled for large stable evaluation contours. The problem with kernels that have simple poles or branch points is that the interpolation domain may include branch points or too many poles. The case of too many poles can be handled by choosing a higher degree of the denominator. For kernels without poles and branch points, i.e., the Gaussian kernels, the problem is that the evaluation of $\phi(\epsilon)$ gets unstable since it is growing exponentially along the imaginary axis. In general, the unstable region around the origin is small enough for $n \lesssim 100$ in two dimensions and for $n \lesssim 300$ in three dimensions. More details can be found in [42, 127].

Parameter choice For the choice of the evaluation radius two different strategies have been proposed, depending on the type of the kernel [127]. For positive definite kernels without poles and branch points ε_R should be set to the approximate minimum of $\log(\tilde{\sigma}_\infty(A(\beta))$ with

$$\tilde{\sigma}_\infty(A(\beta)) = \|A(i\beta)\|_\infty \|A(\beta)^{-1}\|_\infty, \tag{2.149}$$

with the imaginary number i. For other kernels we choose ε_R smaller than the smallest distance to a singularity

$$\varepsilon_R = 0.95(\max_{i,j\leqslant N}\|x_i - x_j\|)^{-1}. \tag{2.150}$$

In some cases with small distances this seems to give too large values. Then, we can choose the minimum of (2.150) and the approximated real value ε such that $\mathrm{cond}(A(\varepsilon)) \approx 10^6$.
The two remaining parameters m, n can be chosen such that $n = \lfloor K/4 \rfloor$ and $m = K - n$. In the one-dimensional cases it is observed that $K = 16$ is a good choice.

2.6.4 Explicit formula of the RBF interpolation

Let us consider the pointwise RBF interpolation problem (2.110) with the interpolation function (2.108). Furthermore, let $x_1, \ldots, x_n$ be the grid points such that $x_i < x_{i+1}$ and $n \in \mathbb{N}$ and $y_1, \ldots, y_n \in \mathbb{R}$ its values. We are looking for an RBF interpolation function

$$s(x) = \sum_{i=1}^{n} a_i \phi(x - x_i) + \sum_{j=1}^{m} b_j L_j(x), \tag{2.151}$$

where L_j for $j = 1, \ldots, m$ are the Lagrange polynomials such that $L_j(x_i) = \delta_{ij}$ and ϕ a conditional positive definite RBF of order m. By further assuming that $m = n - 1$, it holds

Lemma 2.18 (Explicit RBF solution formula)**.** *The interpolation problem* (2.105) *and* (2.109) *can be solved using an explicit formula if we choose an RBF interpolation ansatz with a conditional positive definite RBF of order smaller than* $n - 1$

$$s(x) = \alpha \mathrm{d}\varphi(x) + \sum_{i=1}^{n-1} y_i L_i(x), \tag{2.152}$$

where $\alpha = \frac{y_n - \sum_{i=1}^{n-1} y_i L_i(x_n)}{\mathrm{d}\varphi(x_n)}$, $\mathrm{d}\varphi(x) = \varphi(x) - \sum_{i=1}^{n-1} \varphi(x_i) L_i(x)$ *and* $\varphi(x) = \phi(x - x_n) - \sum_{i=1}^{n-1} L_i(x_n)\phi(x - x_i)$.

Proof. From the representation of the polynomial part in Lagrange polynomials we recover

$$a_j = -a_n L_j(x_n), \qquad \text{for } j = 1, \ldots, n-1. \tag{2.153}$$

This yields the interpolation function

$$s(x) = \alpha\varphi(x) + \sum_{j=1}^{n-1} b_j L_j(x), \tag{2.154}$$

with $\alpha = a_n$, which solves the reduced interpolation problem

$$\alpha\varphi(x_i) + \sum_{j=1}^{n-1} b_j L_j(x_i) = y_i, \qquad \text{for } i = 1, \dots, n. \tag{2.155}$$

By the properties of the Lagrange polynomials we can write down the explicit form of α and b_j

$$\alpha = \frac{y_n - \sum_{i=1}^{n-1} y_i L_i(x_n)}{\mathrm{d}\varphi(x_n)}, \tag{2.156a}$$

$$b_j = y_j - \alpha\varphi(x_j), \qquad \text{for } j = 1, \dots, n-1. \tag{2.156b}$$

□

Remark 2.4. *We can express* $\mathrm{d}\varphi$ *in terms of projections*

$$\mathrm{d}\varphi(x) := \Psi(x, x_i) = (Id - \mathcal{P}^x)(Id - \mathcal{P}^y)[\phi(x-y))]|_{y=x_i}, \tag{2.157}$$

where the operators $\mathcal{P}^z$ *is the projection of the variable* z *on the polynomial space of dimension* $n-1$*. Schaback [108] shows that* Ψ *is positive definite on* $\mathbb{R}^d \backslash \{x_1, \dots, x_{n-1}\}$*. Thus, it is closely related to reproducing kernels and its native spaces, introduced in [108].*

Remark 2.5. *Note that this representation is independent to permutations of the indices. In general we can choose* $\tilde{y}_n = y_j$ *and* $\tilde{y}_i \in \{y_l | \ l \neq j\}$*.*

3 Entropy stable RBF-based reconstruction methods

In one space dimension there is no need to deviate from the polynomial reconstruction to construct high-order methods as introduced in Section 2.2. For unstructured grids in multiple dimensions the central problem is the construction of an interpolation function. There exist a lot of cell or point configurations such that the reconstruction problem is not well-defined. This issue can be relaxed by solving an overdetermined system of equations, but then we lose the exact interpolation property. The RBF-interpolation circumvents this problem since we do not need an unisolvent set of cells or points, but only an unisolvent subset of lower order. Thus, by adding some extra cells we significantly reduce the probability that an unsolvable configuration occurs. However, the theory of RBF-based reconstruction methods is behind that of polynomial reconstructions. In this chapter, we develop a sign-stable reconstruction method of second and third degree for one-dimensional problems. Thus, we show that the RBF reconstruction is comparable with the polynomial reconstruction with advantages in higher dimensions. The results of this chapter are published in [62].

3.1 Smoothness indicator for RBF interpolation functions

In essentially nonoscillatory (ENO)- and weighted ENO (WENO)-type methods a key component is to measure the smoothness of the interpolation function. In the polynomial ENO scheme, the highest degree divided difference plays an important role for identifying the least oscillating interpolation of a certain degree. To extend this to RBF-based interpolation we need something similar. However, the method of divided differences, used in the standard Newton's interpolation formula, is valid only for polynomials.

3.1.1 Generalized divided differences

For non-polynomial basis functions Mühlbach [92] introduces generalized divided differences, which coincide in the monomial case with the standard appearances. The result is based on functions $f_1, \dots, f_n$ that form a Chebyshev system, i.e., they satisfy

$$\begin{vmatrix} f_1(z_1) & \cdots & f_1(z_k) \\ \vdots & & \vdots \\ f_k(z_1) & \cdots & f_k(z_k) \end{vmatrix} \neq 0, \tag{3.1}$$

for all distinct points $z_1, \dots, z_k$ and for $k = 0, \dots, n$. Using Cramer's rule we recover that for any $f : \mathbb{R} \to \mathbb{R}$ and set of distinct points $x_1, \dots, x_n$ there exists a unique linear combination

$$\mathrm{p}_n f := \mathrm{p}f \begin{bmatrix} f_1, \dots, f_n \\ x_1, \dots, x_n \end{bmatrix}, \tag{3.2}$$

of $f_1, \dots, f_n$ which satisfy the interpolation condition

$$\mathrm{p}_n f(x_i) = f(x_i), \qquad \text{for all } i = 1, \dots, n. \tag{3.3}$$

Theorem 3.1 (Generalized Newton's interpolation formula [93])**.** *Let $f_1, \dots, f_n$ form a complete Chebyshev system. Then for any $f : \mathbb{R} \to \mathbb{R}$ and any subset $G_n = \{x_1, \dots, x_n\} \subset \mathbb{R}$ of cardinality n it holds*

$$\mathrm{p}f \begin{bmatrix} f_1, \dots, f_n \\ x_1, \dots, x_n \end{bmatrix} = \sum_{k=1}^{n} \left[\begin{array}{c|c} f_1, \dots, f_k & f \\ x_1, \dots, x_k & \end{array} \right] \cdot g_k, \tag{3.4}$$

where

$$\begin{aligned} g_1 &:= f_1, \\ g_k &:= r_{k-1} f_k, \qquad && \text{for } k = 2, \dots, n, \end{aligned}$$

with the interpolation error in the k-th step

$$r_k f := \mathrm{r}f \begin{bmatrix} f_1, \dots, f_k \\ x_1, \dots, x_k \end{bmatrix} := f - \mathrm{p}f \begin{bmatrix} f_1, \dots, f_k \\ x_1, \dots, x_k \end{bmatrix}, \tag{3.5}$$

and the recursively defined coefficients

$$\left[\begin{matrix} f_1,\dots,f_k \\ x_1,\dots,x_k \end{matrix}\,\middle|\, f \right] := \frac{\left[\begin{matrix} f_1,\dots,f_{k-1} \\ x_2,\dots,x_k \end{matrix}\,\middle|\, f \right] - \left[\begin{matrix} f_1,\dots,f_{k-1} \\ x_1,\dots,x_{k-1} \end{matrix}\,\middle|\, f \right]}{\left[\begin{matrix} f_1,\dots,f_{k-1} \\ x_2,\dots,x_k \end{matrix}\,\middle|\, f_k \right] - \left[\begin{matrix} f_1,\dots,f_{k-1} \\ x_1,\dots,x_{k-1} \end{matrix}\,\middle|\, f_k \right]}, \quad \textit{for } k \geqslant 2, \tag{3.6}$$

where

$$\left[\begin{matrix} f_1 \\ x_j \end{matrix}\,\middle|\, f \right] := \frac{f(x_j)}{f_1(x_j)}. \tag{3.7}$$

Based on this we can express the generalized divided differences for the basis

$$\{1, x, \dots, x^{n-2}, \varphi\}, \tag{3.8}$$

for $n \in \mathbb{N}$ and φ from Lemma 2.18 to quantify the oscillations of the interpolation function. To distinguish between the Lagrange polynomials of different degree we write $L_j^{i,d}$ for the Lagrange polynomial of degree d such that $L_j^{i,d}(x_l) = \delta_{lj}$ for $l \in \{i-d-1,\dots,i-1\}$.

Theorem 3.2. *Let the basis be given by* $\{1, x, \dots, x^{n-2}, \varphi\}$ *for* $n \in \mathbb{N}$ *and* φ *as defined in Lemma 2.18. We recover the generalized divided differences of the form*

$$\left[\begin{matrix} 1 \\ x_0 \end{matrix}\,\middle|\, f \right] = f(x_0) = y_0, \tag{3.9}$$

$$\left[\begin{matrix} 1, x, \dots, x^k \\ x_1, x_2, \dots, x_{k+1} \end{matrix}\,\middle|\, f \right] = \frac{y_{k+1} - \sum_{i=1}^{k} y_i L_i^{k+1,k-1}(x_{k+1})}{\prod_{i=1}^{k}(x_{k+1} - x_i)}, \ \textit{for } k < n-1, \tag{3.10}$$

$$\left[\begin{matrix} 1, x, \dots, x^{n-2}, \varphi \\ x_1, x_2, \dots, x_{n-1}, x_n \end{matrix}\,\middle|\, f \right] = \frac{y_n - \sum_{i=1}^{n-1} y_i L_i^{n,n-2}(x_n)}{\varphi(x_n) - \sum_{i=1}^{n-1} \varphi(x_i) L_i^{n,n-2}(x_n)}. \tag{3.11}$$

By comparing these results with the RBF interpolation in Lemma 2.18, we observe that the last divided difference can be written as

$$\left[\begin{matrix} 1, x, \dots, x^{n-2}, \varphi \\ x_1, x_2, \dots, x_{n-1}, x_n \end{matrix}\,\middle|\, f \right] = \alpha. \tag{3.12}$$

Based on the success of the classic ENO scheme this suggests that α may be a good choice as the smoothness indicator.

Proof. (Theorem 3.2) The proof is split into two steps. In the first one, we show (3.10)

and in the second one (3.11). Recall that the generalized divided differences coincide with the standard one in the case of a monomial basis.
The proof is based on induction and the base for $k = 1$ is true since $L_j^{i,0}(x) = 1$. We have

$$\left[\begin{array}{c|c} 1, x & f \\ x_1, x_2 & \end{array}\right] = \frac{y_2 - y_1}{x_2 - x_1} = \frac{y_2 - L_1^{2,0}(x_2)}{x_2 - x_1}. \tag{3.13}$$

For the inductive step, we assume (3.10) to be given for index k and we show its correctness for $k+1$. Since we only have monomials as a basis, we can use the standard divided difference method

$$\begin{aligned}\left[\begin{array}{c|c} 1, x, \ldots, x^{k+1} & f \\ x_1, x_2, \ldots, x_{k+2} & \end{array}\right] &= \frac{\left[\begin{array}{c|c} 1, x, \ldots, x^{k} & f \\ x_2, x_3, \ldots, x_{k+2} & \end{array}\right] - \left[\begin{array}{c|c} 1, x, \ldots, x^{k} & f \\ x_1, x_2, \ldots, x_{k+1} & \end{array}\right]}{x_{k+2} - x_1}, \\ &= \frac{y_{k+2} - \sum_{i=1}^{k} y_{i+1} L_{i+1}^{k+2,k-1}(x_{k+2})}{(x_{k+2} - x_1)\prod_{i=1}^{k}(x_{k+2} - x_{i+1})} - \frac{y_{k+1} - \sum_{i=1}^{k} y_i L_i^{k+1,k-1}(x_{k+1})}{(x_{k+2} - x_1)\prod_{i=1}^{k}(x_{k+1} - x_i)}.\end{aligned}$$

We rewrite it in the following form

$$\begin{aligned}\left[\begin{array}{c|c} 1, x, \ldots, x^{k+1} & f \\ x_1, x_2, \ldots, x_{k+2} & \end{array}\right] &= \Bigg(y_{k+2} + y_1 L_1^{k+1,k-1}(x_{k+1}) \prod_{j=1}^{k} \frac{x_{k+2} - x_{j+1}}{x_{k+1} - x_j} \\ &\quad + \sum_{i=2}^{k} y_i \Big(L_i^{k+1,k-1}(x_{k+1}) \prod_{j=1}^{k} \frac{x_{k+2} - x_{j+1}}{x_{k+1} - x_j} - L_i^{k+2,k-1}(x_{k+2}) \Big) \\ &\quad - y_{k+1} \Big(\prod_{j=1}^{k} \frac{x_{k+2} - x_{j+1}}{x_{k+1} - x_j} + L_{k+1}^{k+2,k-1}(x_{k+2}) \Big) \Bigg) \frac{1}{\prod_{j=1}^{k+1}(x_{k+2} - x_j)}, \\ &= y_{k+2} + \sum_{i=1}^{k+1} A_i y_i.\end{aligned}$$

To calculate the coefficients A_i we insert the definition of the Lagrange polynomial. Let us start with $i = 1$

$$\begin{aligned}A_1 &= \prod_{j=2}^{k} \frac{x_{k+1} - x_j}{x_1 - x_j} \prod_{j=1}^{k} \frac{x_{k+2} - x_{j+1}}{x_{k+1} - x_j}, \\ &= \frac{x_{k+2} - x_2}{x_{k+1} - x_1} \prod_{j=2}^{k} \frac{x_{k+2} - x_{j+1}}{x_1 - x_j} = -L_1^{k+2,k}(x_{k+2}).\end{aligned}$$

Next, we express A_i for $1 < i < k+1$

$$\begin{aligned}
A_i &= \prod_{\substack{j=1\\ j\neq i}}^{k} \frac{x_{k+1}-x_j}{x_i-x_j} \prod_{j=1}^{k} \frac{x_{k+2}-x_{j+1}}{x_{k+1}-x_j} - \prod_{\substack{j=2\\ j\neq i}}^{k+1} \frac{x_{k+2}-x_j}{x_i-x_j},\\
&= \frac{x_{k+2}-x_{i+1}}{x_{k+1}-x_i} \prod_{\substack{j=1\\ j\neq i}}^{k} \frac{x_{k+2}-x_{j+1}}{x_i-x_j} - \prod_{\substack{j=2\\ j\neq i}}^{k+1} \frac{x_{k+2}-x_j}{x_i-x_j}\\
&= \prod_{\substack{j=2\\ j\neq i}}^{k+1} \frac{x_{k+2}-x_j}{x_i-x_j}\Big(\frac{x_{k+2}-x_i}{x_1-x_i}-1\Big),\\
&= \prod_{\substack{j=2\\ j\neq i}}^{k+1} \frac{x_{k+2}-x_j}{x_i-x_j}\Big(\frac{x_{k+2}-x_1}{x_1-x_i}\Big) = -L_i^{k+2,k}(x_{k+2}).
\end{aligned}$$

The remaining term can be rewritten as

$$\begin{aligned}
A_{k+1} &= \prod_{j=1}^{k} \frac{x_{k+2}-x_{j+1}}{x_{k+1}-x_j} + \prod_{j=2}^{k} \frac{x_{k+2}-x_j}{x_{k+1}-x_j},\\
&= \prod_{j=2}^{k} \frac{x_{k+2}-x_j}{x_{k+1}-x_j}\Big(\frac{x_{k+2}-x_{k+1}}{x_{k+1}-x_1}+1\Big) = L_{k+1}^{k+2,k}(x_{k+2}).
\end{aligned}$$

This completes the proof of (3.10).
To prove (3.11) we use (3.6) with $f_{N-1} = \varphi$ and the result from above. We rewrite the generalized divided differences in terms of the standard ones

$$\begin{aligned}
\begin{bmatrix} 1,\dots,x^{N-2},\varphi \\ x_1,\dots,x_{N-1},x_N \end{bmatrix} f \Big] &= \frac{\left[\begin{matrix} 1,x,\dots,x^{N-2} \\ x_2,x_3,\dots\ x_N \end{matrix}\middle| f\right] - \left[\begin{matrix} 1,x,\dots,x^{N-2} \\ x_1,x_2,\dots,x_{N-1} \end{matrix}\middle| f\right]}{\left[\begin{matrix} 1,x,\dots,x^{N-2} \\ x_2,x_3,\dots\ x_N \end{matrix}\middle| \varphi\right] - \left[\begin{matrix} 1,x,\dots,x^{N-2} \\ x_1,x_2,\dots,x_{N-1} \end{matrix}\middle| \varphi\right]},\\
&= \frac{\left[\begin{matrix} 1,x,\dots,x^{N-1} \\ x_1,x_2,,\dots,x_N \end{matrix}\middle| f\right]}{\left[\begin{matrix} 1,x,\dots,x^{N-1} \\ x_1,x_2,\dots,x_N \end{matrix}\middle| \varphi\right]}.
\end{aligned}$$

From this representation we obtain the final result by applying (3.10). □

3.1.2 Relation to reproducing kernel Hilbert spaces and its norm

As mentioned in Remark 2.4 there is a close relation to native spaces of conditionally positive definite functions (see Schaback [108]). Indeed, the RBF-based basis function $\mathrm{d}\varphi$ can be expressed in terms of the modified kernel function

$$\Psi(x,y) = (\mathrm{Id} - \mathcal{P}^x)(\mathrm{Id} - \mathcal{P}^y)[\phi(x-y))]. \tag{3.14}$$

By analysing the norm of the interpolation function, based on the inner product of the native space, we have

Lemma 3.3. *Let s be a RBF-interpolation function given by* (2.152)*. Then, it has the norm*

$$\|s\|_\phi^2 = \sum_{i=1}^{n-1} s(x_i)^2 + \alpha^2 \mathrm{d}\varphi(x_n). \tag{3.15}$$

In particular, we have

$$\|s\|_\phi \approx \frac{\beta}{\mathrm{d}\varphi(x_n)^{1/2}}, \tag{3.16}$$

with $\beta = y_n - \sum_{i=1}^{n-1} y_i L_i(x_n)$.

This lemma suggests a scaling of $\mathrm{d}\varphi(x_n)^{1/2}$ of our smoothness indicator.

Proof. The inner product of the native space is

$$(f,g)_\phi = \sum_{i=1}^{n-1} f(x_i)g(x_i) + (f - \mathcal{P}f, g - \mathcal{P}g)_{\phi,0}, \tag{3.17}$$

with

$$(f,g)_{\phi,0} = \sum_{j=1}^{m}\sum_{k=1}^{n} \lambda_j \mu_k \phi(x_j, y_k), \tag{3.18}$$

for $f = \sum_{j=1}^m \lambda_j \phi(x, x_j)$ and $g = \sum_{k=1}^n \mu_k \phi(x, y_k)$ [108].
We have $(s - \mathcal{P}s)(x) = \beta \frac{\mathrm{d}\varphi(x)}{\mathrm{d}\varphi(x_n)}$ and

$$\|s\|_\phi^2 = \sum_{j=1}^{n-1} s(x_j)^2 + \left(\frac{\beta}{\mathrm{d}\varphi(x_n)}\right)^2 (\mathrm{d}\varphi, \mathrm{d}\varphi)_\phi. \tag{3.19}$$

Finally, we insert the definition of $\mathrm{d}\varphi$ to recover

$$\begin{aligned}(\mathrm{d}\varphi, \mathrm{d}\varphi)_\phi &= (\mathrm{d}\varphi, \mathrm{d}\varphi)_{\phi,0}, \\ &= \phi(0) - 2\sum_{j=1}^{n-1} \phi(x_n - x_j) L_j(x_n) + \sum_{j,k=1}^{n-1} \phi(x_k - x_j) L_j(x_n) L_k(x_n), \\ &= \mathrm{d}\varphi(x_n).\end{aligned}$$

□

Corollary 3.4. *Let $\mathrm{d}\varphi$ be given as in Lemma 2.18. We have*

$$\mathrm{d}\varphi(x_n) > 0. \tag{3.20}$$

Lemma 3.5 (Equivalent Norm)**.** *The set defined by*

$$\mathcal{B} := \left\{\varphi\right\} \cup \left\{ \frac{\varphi(x_i)}{L_i(x_n)} L_i \,|i = 1, \ldots, n-1 \right\}, \tag{3.21}$$

is a basis for the interpolation space. In particular, we have equivalence of the norms $||\cdot||_\phi$ and $||\cdot||_\mathcal{B}$, where

$$\|s\|_\mathcal{B}^2 = \sum_{i=1}^{n} \alpha_i^2,$$

for $s(x) = \alpha_n \varphi(x) + \sum_{i=1}^{n-1} \alpha_i \frac{\varphi(x_i)}{L_i(x_n)} L_i(x)$.

Proof. From the interpolation (2.152) we recover that $\mathcal{B}$ is a basis of the interpolation space. □

3.1.3 Smoothness indicator and stencil choice

Harten et al. [57] proposed the essentially nonoscillatory method to control spurious oscillations at discontinuities. It is based on the evaluation of multiple stencils for each cell C_i in which we need to reconstruct the solution. Finally, one chooses the least oscillatory reconstruction to define s_i. Fjordholm et al. [36] showed the sign property for the polynomial reconstruction method with the Algorithm 2.1 which utilizes the last divided difference as a local smoothness indicator [57]. A sign-preserving WENO reconstruction method was proposed by Fjordholm and Ray [37]. In the RBF reconstruction, the highest derivative is similar to the RBF-part of the reconstruction in Lemma 3.3 and Theorem 3.2. As we shall show, the recursive algorithm from the

polynomial case, combined with the smoothness indicator

$$\mathrm{IS}(s) = \frac{\beta}{\mathrm{d}\varphi(x_n)^{1/2}} \qquad \text{with } \beta = y_n - \sum_{i=1}^{n-1} y_i L_i(x_n), \tag{3.22}$$

is sign-stable for small enough grid sizes. Numerical experiments confirm this to be true for general grids. In the next section, we prove this for the second and third degree reconstructions on general grids.
Note that Corollary 3.4 ensures the definition of $\mathrm{IS}(s)$.

Algorithm 3.1 Recursive Algorithm

Let the interpolation points $x_{i-n+1}, \dots, x_{i+n-1}$ and its values $y_{i-n+1}, \dots, y_{i+n-1}$ be given.
Start by initializing $s_0 = 0$.
for $j = 0, \dots, n-2$ **do**
  if $|\,\mathrm{IS}(s(i+s_j-1, \dots, i+s_j+j))| < |\,\mathrm{IS}(s(i+s_j, \dots, i+s_j+j+1))|$ **then**
    Set $s_{j+1} = s_j - 1$
  else
    Set $s_{j+1} = s_j$
  end if
end for
Define the stencil $S_i = \{C_{i+s_n}, \dots, C_{i+s_n+n-1}\}$.

Remark 3.1. *The restriction that the sign property holds only on grids with small grid size is not a limitation. For infinitely smooth RBFs we can choose a small shape parameter to artificially decrease the computational grid size.*

Remark 3.2. *The smoothness indicator* (3.22) *requires an impractical and computationally expensive evaluation. However, from Lemma 3.5 we recover*

$$\mathrm{d}\varphi(x_n) = \|\mathrm{d}\varphi\|_{\phi}^2 \approx \|\mathrm{d}\varphi\|_{\mathcal{B}}^2 = \Big(1 + \sum_{i=1}^{n-1} L_i(x_n)^2\Big). \tag{3.23}$$

Thus, we have equivalence of the smoothness indicator (3.22) *with*

$$\mathrm{IS}(s) = \Big(\sum_{i=1}^{n} a_i^2\Big)^{1/2} = \frac{\beta}{\mathrm{d}\varphi(x_n)}\Big(1 + \sum_{i=1}^{n-1} L_i(x_n)^2\Big)^{1/2}. \tag{3.24}$$

To choose the least oscillatory stencil S_i for the i-th cell for the RBF-reconstruction we follow Algorithm 3.1 which is based on the Algorithm 2.1. We use the notation $s(j, \dots, j+k)$ that corresponds to the reconstruction on the cells $C_j, \dots, C_{j+k}$ with the interpolation points $x_j, \dots, x_{j+k}$ and its values $y_j, \dots, y_{j+k}$.

Remark 3.3. *In the general case of* $n \geqslant m+1$ *and a conditionally positive definite RBF*

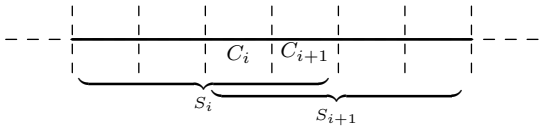


Figure 3.1 – Stencils for $r_2 = -2$, $s_2 = -1$.

of order m, we replace α by $\sqrt{\sum_{i=1}^{n} a_i^2}$ in Algorithm 3.1. In this case, it is more difficult to prove the sign property, but numerical experiments suggest that it remains valid.

3.2 Sign property for 2nd and 3rd degree reconstruction

Based on the results from the previous sections we show the sign property of the RBF interpolation for the second and third degree reconstruction, i.e. $n = 2, 3$. This means that we deal with stencils S_i of size n which represent the interpolation points for the reconstruction on cell C_i. Let us name them

$$S_i = \{C_{i+r_{n-1}}, \dots, C_{i+r_{n-1}+n-1}\}, \tag{3.25a}$$
$$S_{i+1} = \{C_{i+s_{n-1}+1}, \dots, C_{i+s_{n-1}+n}\}, \tag{3.25b}$$

where $r_{n-1} \leqslant 1 + s_{n-1}$ and C_j is the j-th cell with its mid-point x_j with $f(x_j) = y_j$ on which we apply the interpolation. Further, we define $d_{n-1} := 1 + s_{n-1} - r_{n-1} \geqslant 0$ as the shift between the stencils. The stencils are chosen by Algorithm 3.1 and there are no constraints on the stencils, see Figure 3.1 as an example.

3.2.1 Notation

For simplicity, we introduce some general notation. We assume the stencil length to be n and we name terms by the highest appearing index j that exists in the underlying stencil $C_{j-n+1}, \dots, C_j$. We also define

$$L_j^i(x) - \text{Lagrange polynomial of degree } n-1 \text{ such that} \quad L_j^i(x_l) = \delta_{lj} \text{ for } l \in \{i-n+1, \dots, i-1\}, \tag{3.26}$$

$$\varphi^j(x) := \phi(x - x_j) - \sum_{l=1}^{n-1} \phi(x - x_{j-n+l}) L_{j-n+l}^j(x_j), \tag{3.27}$$

$$\mathrm{d}\varphi^j(x) := \varphi^j(x) - \sum_{l=1}^{n-1} \varphi^j(x_{j-n+l}) L_{j-n+l}^j(x), \tag{3.28}$$

$$\beta_j := y_j - \sum_{l=1}^{n-1} y_{j-n+l} L^j_{j-n+l}(x_j), \quad \alpha_j := \frac{\beta_j}{\mathrm{d}\varphi^j(x_j)}, \qquad \gamma_j := \frac{\beta_j}{\mathrm{d}\varphi^j(x_j)^{1/2}}. \tag{3.29}$$

3.2.2 Representation of the reconstructed jumps

The idea of the proof is to give a simple representation of the reconstructed jumps

$$\mathrm{jR}_{i+1/2} := s_{i+1}(x_{i+1/2}) - s_i(x_{i+1/2}), \tag{3.30}$$

and show that each term has the same sign as the jump in its neighboring cells. Let us assume that we have given the stencils S_i and S_{i+1} for the cells i and $i+1$ from Algorithm 3.1.

Theorem 3.6 (Generalized representation)**.** *The second and third degree reconstructed jump can be written in the following form*

$$\mathrm{jR}_{i+1/2} = \sum_{j=0}^{d_{n-1}-1} C_j(\gamma_{i+r_{n-1}+n+j} - \gamma_{i+r_{n-1}+n-1+j}) + \varepsilon(\Delta x), \tag{3.31}$$

with the constants

$$\begin{aligned} C_0 &= \frac{\mathrm{d}\varphi^k(x_{i+1/2})}{\delta_k} - A_k\delta_k, \\ C_j &= C_{j-1} - A_{k+j}\delta_{k+j} = \frac{\mathrm{d}\varphi^k(x_{i+1/2})}{\delta_k} - \sum_{l=0}^{j} A_{k+l}\delta_{k+l}, \end{aligned} \tag{3.32}$$

and an error term

$$\varepsilon(\Delta x) = \gamma_{k+d_{n-1}}\Big(\frac{\mathrm{d}\varphi^{k+d_{n-1}}(x_{i+1/2})}{\delta_{k+d_{n-1}}} - C_{d_{n-1}-1}\Big), \tag{3.33}$$

where $k = i + r_{n-1} + n - 1$, $\delta_k = \mathrm{d}\varphi^k(x_k)^{1/2}$ *and*

$$A_k = \frac{L^k_{k-n+1}(x_{i+1/2})}{L^k_{k-n+1}(x_k)}. \tag{3.34}$$

The proof relies on multiple Lemmas which we now develop.

Lemma 3.7. *Given the Lagrange polynomials. For* $n = 2, 3$ *it holds*

$$-\sum_{l=1}^{n-1} y_{j-n+l} L^j_{j-n+l}(x_{i+1/2}) = A_j\beta_j - \sum_{l=1}^{n-1} y_{j-n+l+1} L^{j+1}_{j-n+l+1}(x_{i+1/2}). \tag{3.35}$$

Proof. The case $n = 2$ is immediate since the Lagrange polynomials are constant. Thus,

(3.35) is

$$-y_{j-1} = \beta_j - y_j, \tag{3.36}$$

For $n = 3$, we write the left hand side of (3.35), subtract $A_j\beta_j$ and use the definition of A_j

$$-y_{j-2}L^j_{j-2}(x_{i+1/2}) - y_{j-1}L^j_{j-1}(x_{i+1/2}) - A_j\beta_j = -y_{j-1}\Big(L^j_{j-1}(x_{i+1/2}) - A_jL^j_{j-1}(x_j)\Big) - y_jA_j.$$

Note that $A_j = L^{j+1}_j(x_{i+1/2})$ and consider

$$A_jL^j_{j-1}(x_j) - L^j_{j-1}(x_{i+1/2}) = \frac{(x_{i+1/2} - x_j)}{(x_j - x_{j-1})} = -L^{j+1}_{j-1}(x_{i+1/2}). \tag{3.37}$$

□

Lemma 3.8. *The reconstructed jump* $\mathrm{jR}_{i+1/2}$ *for the second and third degree reconstruction method can be expressed as*

$$\mathrm{jR}_{i+1/2} = \frac{\gamma_{k+d_{n-1}}}{\delta_{k+d_{n-1}}}\mathrm{d}\varphi^{k+d_{n-1}}(x_{i+1/2}) - \frac{\gamma_k}{\delta_k}\mathrm{d}\varphi^k(x_{i+1/2}) + \sum_{j=0}^{d_{n-1}-1} A_{k+j}\gamma_{k+j}\delta_{k+j}, \tag{3.38}$$

where $k = i + r_{n-1} + n - 1$, $k + d_{n-1} = i + s_{n-1} + n$.

Proof. From Lemma 2.18 and the stencils selected from (3.25) we rewrite the n-th degree reconstructed jump $\mathrm{jR}_{i+1/2}$ between cell i and $i+1$ as

$$\begin{aligned}\mathrm{jR}_{i+1/2} = {} & \alpha_{i+s_{n-1}+n}\mathrm{d}\varphi^{i+s_{n-1}+n}(x_{i+1/2}) - \alpha_{i+r_{n-1}+n-1}\mathrm{d}\varphi^{i+r_{n-1}+n-1}(x_{i+1/2}) \\ & + \sum_{j=1}^{n-1} y_{i+s_{n-1}+j}L^{i+s_{n-1}+n}_{i+s_{n-1}+j}(x_{i+1/2}) - \sum_{j=1}^{n-1} y_{i+r_{n-1}+j-1}L^{i+r_{n-1}+n-1}_{i+r_{n-1}+j-1}(x_{i+1/2}).\end{aligned}$$

The polynomial part of the reconstructed jump is

$$p_{i+1}(x_{i+1/2}) - p_i(x_{i+1/2}) = \sum_{j=0}^{d_{n-1}-1} A_{i+r_{n-1}+n-1+j}\beta_{i+r_{n-1}+n-1+j}, \tag{3.39}$$

by recursively applying Lemma 3.7. This yields

$$\mathrm{jR}_{i+1/2} = \alpha_{k+d_{n-1}}\mathrm{d}\varphi^{k+d_{n-1}}(x_{i+1/2}) - \alpha_k\mathrm{d}\varphi^k(x_{i+1/2}) + \sum_{j=0}^{d_{n-1}-1} A_{k+j}\mathrm{d}\varphi^{k+j}(x_{k+j})\alpha_{k+j}.$$

By inserting $\gamma_i = \alpha_i\mathrm{d}\varphi^i(x_i)^{1/2}$ we recover the result. □

Lemma 3.9. *We have*

$$A_j \mathrm{d}\varphi^j(x_j) - \mathrm{d}\varphi^j(x_{i+1/2}) = -(\varphi^j - \mathcal{P}_{j+1}^{n-1}\varphi^j)(x_{i+1/2}), \tag{3.40}$$

with $\mathcal{P}_{j+1}^k$ as the k-th degree polynomial approximation with respect to the interpolation points $x_j, \ldots, x_{j+1-k}$.

Proof. In the case $n = 2$, we have $A_j = 1$ and $L_j = 1$ and recover

$$\begin{aligned}\mathrm{d}\varphi^j(x_j) - \mathrm{d}\varphi^j(x_{i+1/2}) &= \phi(0) - \phi(x_j - x_{j-1}) - \phi(x_{i+1/2} - x_j) + \phi(x_{i+1/2} - x_{j-1}),\\ &= -(\varphi^j - \mathcal{P}_{j+1}^1\varphi^j)(x_{i+1/2}).\end{aligned}$$

In the case $n = 3$, we have

$$\begin{aligned}A_j \mathrm{d}\varphi^j(x_j) - \mathrm{d}\varphi^j(x_{i+1/2}) = \Big(&\varphi^j(x_j)L_{j-2}^j(x_{i+1/2}) - \varphi^j(x_{i+1/2})L_{j-2}^j(x_j)\\ &- \varphi^j(x_{j-1})L_{j-1}^j(x_j)L_{j-2}^j(x_{i+1/2}) - \varphi^j(x_{j-2})L_{j-2}^j(x_j)L_{j-2}^j(x_{i+1/2})\\ &+ \varphi^j(x_{j-1})L_{j-1}^j(x_{i+1/2})L_{j-2}^j(x_j) + \varphi^j(x_{j-2})L_{j-2}^j(x_{i+1/2})L_{j-2}^j(x_j)\Big)\frac{1}{L_{j-2}^j(x_j)},\end{aligned} \tag{3.41}$$

which can be simplified as

$$\begin{aligned}A_j \mathrm{d}\varphi^j(x_j) - \mathrm{d}\varphi^j(x_{i+1/2}) = \Big(&\varphi^j(x_j)L_{j-2}^j(x_{i+1/2}) - \varphi^j(x_{i+1/2})L_{j-2}^j(x_j)\\ &+ \varphi^j(x_{j-1})\Big(L_{j-1}^j(x_{i+1/2})L_{j-2}^j(x_j) - L_{j-1}^j(x_j)L_{j-2}^j(x_{i+1/2})\Big)\Big)\frac{1}{L_{j-2}^j(x_j)}.\end{aligned} \tag{3.42}$$

Next, we express the last term

$$L_{j-1}^j(x_{i+1/2})L_{j-2}^j(x_j) - L_{j-1}^j(x_j)L_{j-2}^j(x_{i+1/2}) = L_{j-2}^j(x_j) - L_{j-2}^j(x_{i+1/2}), \tag{3.43}$$

and insert this into (3.42)

$$\begin{aligned}A_j \mathrm{d}\varphi^j(x_j) - \mathrm{d}\varphi^j(x_{i+1/2}) &= -\varphi^j(x_{i+1/2}) + \varphi^j(x_j)L_j^{j+1}(x_{i+1/2}) + \varphi^j(x_{j-1})L_{j-1}^{j+1}(x_{i+1/2}),\\ &= -(\varphi^j - \mathcal{P}_{j+1}^{n-1}\varphi^j)(x_{i+1/2}).\end{aligned}$$

where we use that

$$L_j^{j+1}(x_{i+1/2}) = \frac{L_{j-2}^j(x_{i+1/2})}{L_{j-2}^j(x_j)}, \qquad L_{j-1}^{j+1}(x_{i+1/2}) = 1 - \frac{L_{j-2}^j(x_{i+1/2})}{L_{j-2}^j(x_j)}. \tag{3.44}$$

□

Now, we are ready to prove Theorem 3.6.

Proof. (Theorem 3.6)
The goal is to show the equivalence with the representation in Lemma 3.8. Therefore, we insert (3.33) into (3.31) to recover

$$\begin{aligned} jR_{i+1/2} &= C_{d_{n-1}-1}\gamma_{k+d_{n-1}} + \sum_{j=1}^{d_{n-1}-1} \gamma_{k+j}(C_{j-1}-C_j) - C_0\gamma_k + \varepsilon(\Delta x), \\ &= C_{d_{n-1}-1}\gamma_{k+d_{n-1}} + \sum_{j=1}^{d_{n-1}-1} \gamma_{k+j}(C_{j-1}-C_j) + A_k\delta_k\gamma_k \\ &\quad - \frac{\gamma_k}{\delta_k}\mathrm{d}\varphi^k(x_{i+1/2}) + \gamma_{k+d_{n-1}}\Big(\frac{\mathrm{d}\varphi^{k+d_{n-1}}(x_{i+1/2})}{\delta_{k+d_{n-1}}} - C_{d_{n-1}-1}\Big). \end{aligned}$$

Finally, we insert the definition of C_j to obtain

$$jR_{i+1/2} = \frac{\gamma_{k+d_{n-1}}}{\delta_{k+d_{n-1}}}\mathrm{d}\varphi^{k+d_{n-1}}(x_{i+1/2}) + \sum_{j=0}^{d_{n-1}-1} A_{k+j}\delta_{k+j}\gamma_{k+j} - \frac{\gamma_k}{\delta_k}\mathrm{d}\varphi^k(x_{i+1/2}).$$

□

Remark 3.4. *Let us define*

$$\varepsilon_j(\Delta x) := \frac{1}{\delta_{j+1}\delta_j}\Big(\mathrm{d}\varphi^{j+1}(x_{i+1/2})\frac{\delta_j}{\delta_{j+1}} - \mathrm{d}\varphi^j(x_{i+1/2}) + A_j\mathrm{d}\varphi^j(x_j)\Big). \tag{3.45}$$

Thus, the error $\varepsilon(\Delta x)$ can be written as

$$\varepsilon(\Delta x) = \beta_{k+d_{n-1}} \sum_{j=0}^{d_{n-1}-1} \varepsilon_{k+j}(\Delta x)\frac{\delta_{k+j+1}}{\delta_{k+d_{n-1}}}. \tag{3.46}$$

From Lemma 3.9 we can express $\varepsilon_j(\Delta x)$ by

$$\varepsilon_j(\Delta x) = \frac{1}{\delta_{j+1}\delta_j}\Big(\mathrm{d}\varphi^{j+1}(x_{i+1/2})\frac{\delta_j}{\delta_{j+1}} - (\varphi^j - \mathcal{P}_{j+1}^{n-1}\varphi^j)(x_{i+1/2})\Big). \tag{3.47}$$

3.2.3 Sign property for small grid size

In this section, we analyse the reconstructed jumps for infinitely smooth RBFs for small grid size $\Delta x \to 0$. From Theorem 3.6 we have a simple expression for the reconstructed jump. We first show that the error $\varepsilon(\Delta x)$ goes to zero as the grid size goes to zero. Then, we show that each term of the remaining equation has the sign of the jump $y_{i+1} - y_i$.

Remark 3.5. *The notation $\mathcal{O}(\Delta x^p)$ for $\Delta x \to 0$ should be interpreted in the way that given a grid $\hat{x}_0 < \hat{x}_1 < \cdots < \hat{x}_m$ we analyse the terms for the grid $x_0 < x_1 < \cdots < x_m$*

with $x_j = \hat{x}_j \Delta x$ *and we use*

$$\mathcal{O}(\Delta x^p) \leqslant C\Delta x^p, \quad \text{for } \Delta x \to 0. \tag{3.48}$$

Remark 3.6. *When calculating the errors* ε_j *we recall that*

$$\mathrm{d}\varphi^j(x) = \varphi^j(x) - \mathcal{P}_j^{n-1}\varphi^j(x). \tag{3.49}$$

Theorem 3.10. *Let* ϕ *be an infinitely smooth RBF of first or second order. Then, we have that* $\varepsilon(\Delta x) = \mathcal{O}(\Delta x^2)$ *for* $\Delta x \to 0$ *for* $n = 2, 3$ *and*

$$\mathrm{jR}_{i+1/2} = \sum_{j=0}^{d_{n-1}-1} C_j(\gamma_{i+r_{n-1}+n+j} - \gamma_{i+r_{n-1}+n-1+j}) + \mathcal{O}(\Delta x^2). \tag{3.50}$$

Proof. We start by analysing the different parts in the error term $\varepsilon_k(\Delta x)$. Note that as ϕ is a conditionally positive definite RBF

$$\phi(x) = h(x^2). \tag{3.51}$$

Thus, it follows by induction that $\phi^{(2k+1)}(0) = 0$ for $k \in \mathbb{N}$ and we can neglect odd terms in the Taylor expansions.
Let us start with the case $n = 2$ and a first order RBF:

$$\begin{aligned}
\mathrm{d}\varphi^k(y) &= \phi(y - x_k) - \phi(y - x_{k-1}) - \phi(x_{k-1} - x_k) + \phi(0), \\
&= \frac{\phi''(0)}{2}\Big((y - x_k)^2 - (y - x_{k-1})^2 - (x_{k-1} - x_k)^2\Big) + \mathcal{O}(\Delta x^4), \\
&= -\phi''(0)(x_{k-1} - x_k)(x_{k-1} - y) + \mathcal{O}(\Delta x^4).
\end{aligned} \tag{3.52}$$

We further have that

$$\begin{aligned}
(\varphi^k - \mathcal{P}_{k+1}^1\varphi^k)(y) &= \phi(y - x_k) - \phi(y - x_{k-1}) - \phi(0) + \phi(x_k - x_{k-1}), \\
&= \frac{\phi''(0)}{2}\Big((y - x_k)^2 - (y - x_{k-1})^2 + (x_{k-1} - x_k)^2\Big) + \mathcal{O}(\Delta x^4), \\
&= -\phi''(0)(x_{k-1} - x_k)(x_k - y) + \mathcal{O}(\Delta x^4).
\end{aligned} \tag{3.53}$$

From (3.52) we recover

$$\frac{\delta_k}{\delta_{k+1}} = \frac{x_k - x_{k-1}}{x_{k+1} - x_k} + \mathcal{O}(\Delta x^2), \tag{3.54}$$

$$\delta_k\delta_{k+1} = -\phi''(0)(x_k - x_{k-1})(x_{k+1} - x_k) + \mathcal{O}(\Delta x^4) = \mathcal{O}(\Delta x^2), \tag{3.55}$$

which allows us to conclude

$$\varepsilon_k(\Delta x) = \mathcal{O}(\Delta x^2). \tag{3.56}$$

To find a bound for $\varepsilon(\Delta x)$ we further need $\delta_k/\delta_{k+1} = \mathcal{O}(1)$ and $\beta_{k+d} = \mathcal{O}(1)$. The latter is clear since the reconstructed function is bounded and $L_j^i(x^i)$ is $\mathcal{O}(1)$. Further, $\delta_k/\delta_{k+1} = \mathcal{O}(1)$ results from (3.54). Using (3.56) in (3.46) we conclude that $\varepsilon(\Delta x) = \mathcal{O}(\Delta x^2)$.
Next, we consider the more complicated case with $n = 3$ and a second order RBF ϕ. Hence, we need to analyse the following two terms:

$$\begin{aligned}
\mathrm{d}\varphi^{k+1}(y) &= \varphi^{k+1}(y) - \varphi^{k+1}(x_k)L_k^{k+1}(y) - \varphi^{k+1}(x_{k-1})L_{k-1}^{k+1}(y),\\
&= \phi(y-x_{k+1}) - \phi(y-x_k)L_k^{k+1}(x_{k+1}) - \phi(y-x_{k-1})L_{k-1}^{k+1}(x_{k+1})\\
&\quad - \Big(\phi(x_k-x_{k+1}) - \phi(0)L_k^{k+1}(x_{k+1}) - \phi(x_k-x_{k-1})L_{k-1}^{k+1}(x_{k+1})\Big)L_k^{k+1}(y)\\
&\quad - \Big(\phi(x_{k-1}-x_{k+1}) - \phi(x_{k-1}-x_k)L_k^{k+1}(x_{k+1}) - \phi(0)L_{k-1}^{k+1}(x_{k+1})\Big)L_{k-1}^{k+1}(y),
\end{aligned} \tag{3.57}$$

$$\begin{aligned}
(\varphi^k - \mathcal{P}_{k+1}^2\varphi^k)(y) &= \varphi^k(y) - \varphi^k(x_k)L_k^{k+1}(y) - \varphi^k(x_{k-1})L_{k-1}^{k+1}(y),\\
&= \phi(y-x_k) - \phi(y-x_{k-1})L_{k-1}^k(x_k) - \phi(y-x_{k-2})L_{k-2}^k(x_k)\\
&\quad - \Big(\phi(0) - \phi(x_k-x_{k-1})L_{k-1}^k(x_k) - \phi(x_k-x_{k-2})L_{k-2}^k(x_k)\Big)L_k^{k+1}(y)\\
&\quad - \Big(\phi(x_{k-1}-x_k) - \phi(0)L_{k-1}^k(x_k) - \phi(x_{k-1}-x_{k-2})L_{k-2}^k(x_k)\Big)L_{k-1}^{k+1}(y).
\end{aligned} \tag{3.58}$$

As before, we apply the Taylor expansion

$$\begin{aligned}
\varphi^{k+1}(y) &= \phi(y-x_{k+1}) - \phi(y-x_k)L_k^{k+1}(x_{k+1}) - \phi(y-x_{k-1})L_{k-1}^{k+1}(x_{k+1}),\\
&= \frac{\phi''(0)}{2}\Big((y-x_{k+1})^2 - (y-x_k)^2L_k^{k+1}(x_{k+1}) - (y-x_{k-1})^2L_{k-1}^{k+1}(x_{k+1})\Big)\\
&\quad + \frac{\phi^{(4)}(0)}{2}\Big((y-x_k)^4 - (y-x_k)^4L_k^{k+1}(x_{k+1}) - (y-x_{k-1})^4L_{k-1}^{k+1}(x_{k+1})\Big)\\
&\quad + \mathcal{O}(\Delta x^6).
\end{aligned} \tag{3.59}$$

We write

$$\begin{aligned}
\mathrm{d}\varphi^{k+1}(y) &= a_1\frac{\phi''(0)}{2} + a_2\frac{\phi^{(4)}(0)}{4!} + \mathcal{O}(\Delta x^6),\\
(\varphi^k - \mathcal{P}_{k+1}^2\varphi^k)(y) &= b_1\frac{\phi''(0)}{2} + b_2\frac{\phi^{(4)}(0)}{4!} + \mathcal{O}(\Delta x^6).
\end{aligned}$$

Let us calculate the coefficients a_1 and a_2. From standard algebra we recover that $a_1 = 0$. The fourth order term is

$$\begin{aligned}
a_2 &= (y-x_{k+1})^4 - (y-x_k)^4L_k^{k+1}(x_{k+1}) - (y-x_{k-1})^4L_{k-1}^{k+1}(x_{k+1})\\
&\quad - \Big((x_k-x_{k+1})^4 - (x_k-x_{k-1})^4L_{k-1}^{k+1}(x_{k+1})\Big)L_k^{k+1}(y)\\
&\quad - \Big((x_{k-1}-x_{k+1})^4 - (x_{k-1}-x_k)^4L_k^{k+1}(x_{k+1})\Big)L_{k-1}^{k+1}(y),\\
&= 6(x_{k-1}-x_{k+1})(x_k-x_{k+1})(x_{k-1}-y)(x_k-y).
\end{aligned} \tag{3.60}$$

We repeat this for the coefficients b_1 and b_2. We obtain $b_1 = 0$

$$\begin{aligned} b_2 &= (y-x_k)^4 - (y-x_{k-1})^4 L_{k-1}^k(x_k) - (y-x_{k-2})^4 L_{k-2}^k(x_k) \\ &\quad - \Big(- (x_k-x_{k-1})^4 L_{k-1}^k(x_k) - (x_k-x_{k-2})^4 L_{k-2}^k(x_k)\Big) L_k^{k+1}(y) \\ &\quad - \Big((x_{k-1}-x_k)^4 - (x_{k-1}-x_{k-2})^4 L_{k-2}^k(x_k)\Big) L_{k-1}^{k+1}(y), \\ &= 6(x_{k-2}-x_k)(x_{k-1}-x_k)(x_{k-1}-y)(x_k-y). \end{aligned} \tag{3.61}$$

The results can be summarized as

$$\mathrm{d}\varphi^{k+1}(y) = \frac{\phi^{(4)}(0)}{4}(x_{k-1}-x_{k+1})(x_k-x_{k+1})(x_{k-1}-y)(x_k-y) + \mathcal{O}(\Delta x^6), \tag{3.62}$$

$$\begin{aligned} (\varphi^k - \mathcal{P}_{k+1}^2\varphi^k)(y) = &\frac{\phi^{(4)}(0)}{4}(x_{k-2}-x_k)(x_{k-1}-x_k)(x_{k-1}-y)(x_k-y) \\ &+ \mathcal{O}(\Delta x^6). \end{aligned} \tag{3.63}$$

From this we recover

$$\begin{aligned} \frac{\delta_k}{\delta_{k+1}} &= \frac{(x_k-x_{k-1})(x_k-x_{k-2})}{(x_{k+1}-x_k)(x_{k+1}-x_{k-1})} + \mathcal{O}(\Delta x^2), \\ \delta_k\delta_{k+1} &= \frac{\phi^{(4)}(0)}{4}(x_k-x_{k-1})(x_k-x_{k-2})(x_{k+1}-x_k)(x_{k+1}-x_{k-1}) + \mathcal{O}(\Delta x^6), \\ &= \mathcal{O}(\Delta x^4). \end{aligned} \tag{3.64}$$

Thus, we have

$$\mathrm{d}\varphi^{k+1}(y)\frac{\delta_k}{\delta_{k+1}} - (\varphi^k - \mathcal{P}_{k+1}^2\varphi^k)(y) = \mathcal{O}(\Delta x^6), \tag{3.65}$$

which yields

$$\varepsilon_k(\Delta x) = \mathcal{O}(\Delta x^2), \tag{3.66}$$

for $\Delta x \to 0$. Equivalent to the case $n = 2$, we can combine (3.64) and (3.66) in (3.46) and conclude $\varepsilon(\Delta x) = \mathcal{O}(\Delta x^2)$. □

Since the error term $\varepsilon(\Delta x)$ vanishes, the remaining step is to prove that each term of (3.50) has the same sign as the jump.

Theorem 3.11 (Sign property of second and third degree RBF-reconstruction)**.** *Let us assume that the stencil S_i and S_{i+1} are chosen with the Algorithm 3.1. Then, for infinitely smooth RBFs of first or second order it holds that*

$$\mathrm{sgn}(C_j(\gamma_{i+r_{n-1}+n+j} - \gamma_{i+r_{n-1}+n-1+j})) = \mathrm{sgn}(y_{i+1} - y_i), \tag{3.67}$$

for all $j = 0, \dots, d_{n-1} - 1$ *and for* $\Delta x \to 0$.

Proof. The proof is based on a study of all possible choices of stencils, that may result from Algorithm 3.1:

- $S_i = \{C_{i-1}, C_i\}$, $S_{i+1} = \{C_i, C_{i+1}\}$,
- $S_i = \{C_{i-1}, C_i\}$, $S_{i+1} = \{C_{i+1}, C_{i+2}\}$,
- $S_i = \{C_i, C_{i+1}\}$, $S_{i+1} = \{C_{i+1}, C_{i+2}\}$,
- ...

For each case we look at any inequality in Algorithm 3.1 to recover the particular stencil configuration, and show for each case that (3.67) is fulfilled.
Note that $jR_{i+1/2} = 0$, if $S_i = S_{i+1}$. Hence, we do not include such cases in the analysis.
Further, we use the notation

$$A \approx B \quad \text{if } A = B + \mathcal{O}(\Delta x), \quad \text{for } \Delta x \to 0. \tag{3.68}$$

Let us first consider $n = 2$ and assume ϕ is of first order.

Case 1. *Consider the stencils* $S_i = \{C_{i-1}, C_i\}$, $S_{i+1} = \{C_i, C_{i+1}\}$, *which require the following conditions*

$$|\gamma_i| < |\gamma_{i+1}|, \qquad |\gamma_{i+1}| < |\gamma_{i+2}|. \tag{3.69}$$

Further, we have the representation of the jump for small grid sizes

$$jR_{i+1/2} \approx C_0(\gamma_{i+1} - \gamma_i),$$

and from (3.52) *it follows that*

$$C_0 = \delta_i \left(\frac{\mathrm{d}\varphi^i(x_{i+1/2})}{\mathrm{d}\varphi^i(x_i)} - 1 \right) \approx \delta_i \frac{x_{i+1/2} - x_i}{x_i - x_{i-1}} > 0.$$

Hence

$$\operatorname{sgn}\big(C_0(\gamma_{i+1} - \gamma_i)\big) = \operatorname{sgn}(\gamma_{i+1} - \gamma_i) = \operatorname{sgn}(\gamma_{i+1}) = \operatorname{sgn}(y_{i+1} - y_i), \tag{3.70}$$

since

$$|a| > |b| \Rightarrow \operatorname{sgn}(a - b) = \operatorname{sgn}(a). \tag{3.71}$$

Case 2. *Consider the stencils* $S_i = \{C_{i-1}, C_i\}$, $S_{i+1} = \{C_{i+1}, C_{i+2}\}$, *which is equivalent to the conditions*

$$|\gamma_i| < |\gamma_{i+1}|, \qquad |\gamma_{i+1}| > |\gamma_{i+2}|. \tag{3.72}$$

The jump can be represented as

$$jR_{i+1/2} \approx C_0(\gamma_{i+1} - \gamma_i) + C_1(\gamma_{i+2} - \gamma_{i+1}).$$

As before it holds that $\operatorname{sgn}\big(C_0(\gamma_{i+1} - \gamma_i)\big) = \operatorname{sgn}(y_{i+1} - y_i)$ *and*

$$C_1 = C_0 - \delta_{i+1} \approx \delta_i \frac{x_{i+1/2} - x_{i+1}}{x_i - x_{i-1}} < 0. \tag{3.73}$$

Thus, we get for the second term

$$\operatorname{sgn}\big(C_1(\gamma_{i+2} - \gamma_{i+1})\big) = \operatorname{sgn}(\gamma_{i+1} - \gamma_{i+2}) = \operatorname{sgn}(\gamma_{i+1}) = \operatorname{sgn}(y_{i+1} - y_i),$$

where we used (3.71) *and* (3.72).

Case 3. *In the last case of the second degree reconstruction we have the stencils* $S_i = \{C_i, C_{i+1}\}$, $S_{i+1} = \{C_{i+1}, C_{i+2}\}$, *equivalent to the conditions*

$$|\gamma_i| > |\gamma_{i+1}|, \qquad |\gamma_{i+1}| > |\gamma_{i+2}|. \tag{3.74}$$

The representation of the jump is

$$jR_{i+1/2} \approx C_0(\gamma_{i+2} - \gamma_{i+1}).$$

As in the first case we recover from (3.52), *that*

$$C_0 = \delta_{i+1}\left(\frac{\mathrm{d}\varphi^{i+1}(x_{i+1/2})}{\mathrm{d}\varphi^{i+1}(x_i)} - 1\right) \approx \delta_i \frac{x_{i+1/2} - x_{i+1}}{x_{i+1} - x_i} < 0,$$

and

$$\operatorname{sgn}\big(C_0(\gamma_{i+2} - \gamma_{i+1})\big) = \operatorname{sgn}(\gamma_{i+1} - \gamma_{i+2}) = \operatorname{sgn}(\gamma_{i+1}). \tag{3.75}$$

This completes the proof of the sign property for the second degree reconstruction with infinitely smooth RBFs of first order for small enough grids.
Next, we consider $n = 3$.
All possible choices of stencils, that could result from Algorithm 3.1 are:

- $S_i = \{C_{i-2}, C_{i-1}, C_i\}$, $S_{i+1} = \{C_{i-1}, C_i, C_{i+1}\}$,
- $S_i = \{C_{i-2}, C_{i-1}, C_i\}$, $S_{i+1} = \{C_i, C_{i+1}, C_{i+2}\}$,

- $S_i = \{C_{i-2}, C_{i-1}, C_i\}$, $S_{i+1} = \{C_{i+1}, C_{i+2}, C_{i+3}\}$,
- ...

We consider $n = 3$ (third degree reconstruction) and assume ϕ is of second order. The main difference between the second and third degree is that Algorithm 3.1 now gives two conditions for each stencil that depend on different grid sizes. Therefore, we introduce the superscript l to indicate the size of the stencil

$$\delta_k^l = \mathrm{d}\varphi^k(x_k)^{1/2}, \qquad \beta_k^l = y_k - \sum_{j=1}^{l-1} y_{k-j} L_{k-j}^{k,l}(x_k), \qquad \gamma_k^l = \frac{\beta_k^l}{\delta_k^l},$$

based on the stencil $\{C_{k-l+1}, \cdots, C_k\}$. Further, we can show with simple calculations that

$$\beta_{k+1}^3 = \beta_{k+1}^2 - \frac{x_{k+1} - x_k}{x_k - x_{k-1}} \beta_k^2.$$

From (3.52) and (3.62) we recover

$$\frac{x_{k+1} - x_k}{x_k - x_{k-1}} \approx \frac{\delta_{k+1}^2}{\delta_k^2},$$

which allows us to conclude that

$$\delta_{k+1}^3 \gamma_{k+1}^3 = \beta_{k+1}^3 \approx \beta_{k+1}^2 - \frac{\delta_{k+1}^2}{\delta_k^2} \beta_k^2 = \delta_{k+1}^2 (\gamma_{k+1}^2 - \gamma_k^2),$$

and so

$$\gamma_{k+1}^3 = \frac{\delta_{k+1}^2}{\delta_{k+1}^3} (\gamma_{k+1}^2 - \gamma_k^2). \tag{3.76}$$

Note that the term $\delta_{k+1}^2 / \delta_{k+1}^3$ is always positive (Corollary 3.4). Next, we can show the sign of the constant C_l using Theorem 3.6

$$\begin{aligned} C_0 &= \delta_k \left(\frac{\mathrm{d}\varphi^k(x_{i+1/2})}{\mathrm{d}\varphi^k(x_k)} - A_k \right), \\ &\approx \delta_k \left(\frac{(x_{k-2} - x_{i+1/2})(x_{k-1} - x_{i+1/2})}{(x_{k-2} - x_k)(x_{k-1} - x_k)} - \frac{x_{i+1/2} - x_{k-1}}{x_k - x_{k-1}} \right), \\ &\approx \delta_k \frac{(x_{i+1/2} - x_{k-1})(x_{i+1/2} - x_k)}{(x_k - x_{k-2})(x_k - x_{k-1})}, \end{aligned}$$

with $k = i + r_{n-1} + n - 1$. By induction one proves that

$$C_l \approx \delta_k \frac{x_{i+1/2} - x_{k+l-1}}{x_k - x_{k-2}} \frac{x_{i+1/2} - x_{k+l}}{x_k - x_{k-1}}, \tag{3.77}$$

for $l \in \mathbb{N}$ and we recover

$$\operatorname{sgn}(C_l) = (-1)^{r_{n-1}+n-1+l}. \tag{3.78}$$

Case 1. *Consider the stencils $S_i = \{C_{i-2}, C_{i-1}, C_i\}$, $S_{i+1} = \{C_{i-1}, C_i, C_{i+1}\}$, equivalent to the conditions*

$$|\gamma_i^2| < |\gamma_{i+1}^2|, \quad |\gamma_i^3| < |\gamma_{i+1}^3|, \quad |\gamma_{i+1}^2| < |\gamma_{i+2}^2|, \quad |\gamma_{i+1}^3| < |\gamma_{i+2}^3|. \tag{3.79}$$

Note that this case can be characterized by $d_2 = 1$ and $s_2 = r_2 = -2$. From Theorem 3.10 we know

$$jR_{i+1/2} \approx C_0(\gamma_{i+1}^3 - \gamma_i^3), \tag{3.80}$$

and we recover

$$\begin{aligned} \operatorname{sgn}\left(C_0(\gamma_{i+1}^3 - \gamma_i^3)\right) &= \operatorname{sgn}(\gamma_{i+1}^3 - \gamma_i^3) = \operatorname{sgn}(\gamma_{i+1}^3) = \operatorname{sgn}(\gamma_{i+1}^2 - \gamma_i^2), \\ &= \operatorname{sgn}(\gamma_{i+1}^2) = \operatorname{sgn}(y_{i+1} - y_i), \end{aligned}$$

where we used (3.76), (3.77) *and* (3.79).

Case 2. *Assume the stencil $S_i = \{C_{i-2}, C_{i-1}, C_i\}$, $S_{i+1} = \{C_i, C_{i+1}, C_{i+2}\}$, equivalent to the conditions*

$$\begin{aligned} &|\gamma_i^2| < |\gamma_{i+1}^2|, \quad |\gamma_i^3| < |\gamma_{i+1}^3|, \\ &\begin{cases} |\gamma_{i+1}^2| < |\gamma_{i+2}^2|, \quad |\gamma_{i+1}^3| > |\gamma_{i+2}^3|, & (a) \\ |\gamma_{i+1}^2| > |\gamma_{i+2}^2|, \quad |\gamma_{i+2}^3| < |\gamma_{i+3}^3|. & (b) \end{cases} \end{aligned} \tag{3.81}$$

The jump can be written by

$$jR_{i+1/2} \approx C_0(\gamma_{i+1}^3 - \gamma_i^3) + C_1(\gamma_{i+2}^3 - \gamma_{i+1}^3). \tag{3.82}$$

For each term we calculate its sign. The first term can be understood in the same way as above and it holds for both (a) and (b) in (3.81)

$$\operatorname{sgn}\left(C_0(\gamma_{i+1}^3 - \gamma_i^3)\right) = \operatorname{sgn}(y_{i+1} - y_i),$$

For the second term we first assume that (a) *holds and compute its sign as*

$$\begin{aligned}\operatorname{sgn}\left(C_1(\gamma^3_{i+2}-\gamma^3_{i+1})\right) &= \operatorname{sgn}(\gamma^3_{i+1}-\gamma^3_{i+2}) = \operatorname{sgn}(\gamma^3_{i+1}) = \operatorname{sgn}(\gamma^2_{i+1}-\gamma^2_i),\\ &= \operatorname{sgn}(\gamma^2_{i+1}) = \operatorname{sgn}(y_{i+1}-y_i).\end{aligned}$$

For (b) *we split it in two terms using* (3.76)

$$C_1(\gamma^3_{i+2}-\gamma^3_{i+1}) = C_1\frac{\delta^2_{i+2}}{\delta^3_{i+2}}(\gamma^2_{i+2}-\gamma^2_{i+1}) - C_1\frac{\delta^2_{i+1}}{\delta^3_{i+1}}(\gamma^2_{i+1}-\gamma^2_i),$$

and calculate the sign of each one

$$\begin{aligned}\operatorname{sgn}\left(C_1\frac{\delta^2_{i+2}}{\delta^3_{i+2}}(\gamma^2_{i+2}-\gamma^2_{i+1})\right) &= \operatorname{sgn}(\gamma^2_{i+1}-\gamma^2_{i+2}) = \operatorname{sgn}(\gamma^2_{i+1}) = \operatorname{sgn}(y_{i+1}-y_i),\\ \operatorname{sgn}\left(-C_1\frac{\delta^2_{i+1}}{\delta^3_{i+1}}(\gamma^2_{i+1}-\gamma^2_i)\right) &= \operatorname{sgn}(\gamma^2_{i+1}-\gamma^2_i)\Big) = \operatorname{sgn}(\gamma^2_{i+1}) = \operatorname{sgn}(y_{i+1}-y_i).\end{aligned}$$

Case 3. *Consider the stencil* $S_i = \{C_{i-2}, C_{i-1}, C_i\}$, $S_{i+1} = \{C_{i+1}, C_{i+2}, C_{i+3}\}$, *equivalent to the conditions*

$$|\gamma^2_i| < |\gamma^2_{i+1}|, \quad |\gamma^3_i| < |\gamma^3_{i+1}|, \quad |\gamma^2_{i+1}| > |\gamma^2_{i+2}|, \quad |\gamma^3_{i+2}| > |\gamma^3_{i+3}|. \tag{3.83}$$

For the reconstructed jump we have

$$jR_{i+1/2} \approx C_0(\gamma^3_{i+1}-\gamma^3_i) + C_1(\gamma^3_{i+2}-\gamma^3_{i+1}) + C_2(\gamma^3_{i+3}-\gamma^3_{i+2}). \tag{3.84}$$

The sign of each term is

$$\begin{aligned}\operatorname{sgn}\left(C_0(\gamma^3_{i+1}-\gamma^3_i)\right) &= \operatorname{sgn}(\gamma^3_{i+1}-\gamma^3_i) = \operatorname{sgn}(\gamma^3_{i+1}) = \operatorname{sgn}(\gamma^2_{i+1}-\gamma^2_i),\\ &= \operatorname{sgn}(\gamma^2_{i+1}) = \operatorname{sgn}(y_{i+1}-y_i).\end{aligned}$$

For the second term we need

$$\begin{aligned}\operatorname{sgn}(\gamma^3_{i+1}) &= \operatorname{sgn}(\gamma^2_{i+1}-\gamma^2_i) = \operatorname{sgn}(\gamma^2_{i+1}),\\ \operatorname{sgn}(-\gamma^3_{i+2}) &= \operatorname{sgn}(\gamma^2_{i+1}-\gamma^2_{i+2}) = \operatorname{sgn}(\gamma^2_{i+1}),\end{aligned}$$

such that we can show

$$\operatorname{sgn}\left(C_1(\gamma^3_{i+2}-\gamma^3_{i+1})\right) = \operatorname{sgn}(\gamma^3_{i+1}-\gamma^3_{i+2}) = \operatorname{sgn}(\gamma^2_{i+1}) = \operatorname{sgn}(y_{i+1}-y_i).$$

The last term yields

$$\begin{aligned}\operatorname{sgn}\left(C_2(\gamma^3_{i+3}-\gamma^3_{i+2})\right) &= \operatorname{sgn}(\gamma^3_{i+3}-\gamma^3_{i+2}) = -\operatorname{sgn}(\gamma^3_{i+2}) = \operatorname{sgn}(\gamma^2_{i+1}-\gamma^2_{i+2}),\\ &= \operatorname{sgn}(\gamma^2_{i+1}) = \operatorname{sgn}(y_{i+1}-y_i).\end{aligned}$$

Case 4. *Assume the stencils* $S_i = \{C_{i-1}, C_i, C_{i+1}\}$, $S_{i+1} = \{C_i, C_{i+1}, C_{i+2}\}$, *equivalent to the conditions*

$$\begin{cases} |\gamma_i^2| < |\gamma_{i+1}^2|, \quad |\gamma_i^3| \geqslant |\gamma_{i+1}^3|, & (a1) \\ |\gamma_i^2| \geqslant |\gamma_{i+1}^2|, \quad |\gamma_{i+1}^3| < |\gamma_{i+2}^3|, & (a2) \end{cases}$$
$$\begin{cases} |\gamma_{i+1}^2| < |\gamma_{i+2}^2|, \quad |\gamma_{i+1}^3| \geqslant |\gamma_{i+2}^3|, & (b1) \\ |\gamma_{i+1}^2| \geqslant |\gamma_{i+2}^2|, \quad |\gamma_{i+2}^3| < |\gamma_{i+3}^3|. & (b2) \end{cases} \tag{3.85}$$

Here, we have the different combinations (a_1, b_1), (a_1, b_2), (a_2, b_1) *and* (a_2, b_2), *where* (a_2, b_1) *is not possible. The jump is represented as*

$$jR_{i+1/2} \approx C_0(\gamma_{i+2}^3 - \gamma_{i+1}^3). \tag{3.86}$$

Let us first consider the combination (a_1, b_1). *We have*

$$\begin{aligned} \operatorname{sgn}\left(C_0(\gamma_{i+2}^3 - \gamma_{i+1}^3)\right) &= \operatorname{sgn}(\gamma_{i+1}^3 - \gamma_{i+2}^3) = \operatorname{sgn}(\gamma_{i+1}^3) = \operatorname{sgn}(\gamma_{i+1}^2 - \gamma_i^2), \\ &= \operatorname{sgn}(\gamma_{i+1}^2) = \operatorname{sgn}(y_{i+1} - y_i). \end{aligned}$$

In the case (a_1, b_2), *we precalculate*

$$\begin{aligned} \operatorname{sgn}(\gamma_{i+1}^3) &= \operatorname{sgn}(\gamma_{i+1}^2 - \gamma_i^2) = \operatorname{sgn}(\gamma_{i+1}^2), \\ \operatorname{sgn}(-\gamma_{i+2}^3) &= \operatorname{sgn}(\gamma_{i+1}^2 - \gamma_{i+2}^2) = \operatorname{sgn}(\gamma_{i+1}^2). \end{aligned}$$

Thus,

$$\operatorname{sgn}\left(C_0(\gamma_{i+2}^3 - \gamma_{i+1}^3)\right) = \operatorname{sgn}(\gamma_{i+1}^3 - \gamma_{i+2}^3) = \operatorname{sgn}(\gamma_{i+1}^2) = \operatorname{sgn}(y_{i+1} - y_i).$$

In the last case (a_2, b_2), *we get*

$$\begin{aligned} \operatorname{sgn}\left(C_0(\gamma_{i+2}^3 - \gamma_{i+1}^3)\right) &= \operatorname{sgn}(\gamma_{i+1}^3 - \gamma_{i+2}^3) = -\operatorname{sgn}(\gamma_{i+2}^3) = \operatorname{sgn}(\gamma_{i+1}^2 - \gamma_{i+2}^2), \\ &= \operatorname{sgn}(\gamma_{i+1}^2) = \operatorname{sgn}(y_{i+1} - y_i). \end{aligned}$$

Case 5. *Assume the stencils* $S_i = \{C_{i-1}, C_i, C_{i+1}\}$, $S_{i+1} = \{C_{i+1}, C_{i+2}, C_{i+3}\}$, *equivalent to the conditions*

$$\begin{cases} |\gamma_i^2| < |\gamma_{i+1}^2|, \quad |\gamma_i^3| \geqslant |\gamma_{i+1}^3|, & (a) \\ |\gamma_i^2| \geqslant |\gamma_{i+1}^2|, \quad |\gamma_{i+1}^3| < |\gamma_{i+2}^3|, & (b) \end{cases}$$
$$|\gamma_{i+1}^2| > |\gamma_{i+2}^2|, \quad |\gamma_{i+2}^3| > |\gamma_{i+3}^3|. \tag{3.87}$$

The jump is represented as

$$jR_{i+1/2} \approx C_0(\gamma_{i+2}^3 - \gamma_{i+1}^3) + C_1(\gamma_{i+3}^3 - \gamma_{i+2}^3). \tag{3.88}$$

In the case of (a) *we precalculate*

$$\begin{aligned}\operatorname{sgn}(-\gamma^3_{i+2}) &= \operatorname{sgn}(\gamma^2_{i+1} - \gamma^2_{i+2}) = \operatorname{sgn}(\gamma^2_{i+1}),\\ \operatorname{sgn}(\gamma^3_{i+1}) &= \operatorname{sgn}(\gamma^2_{i+1} - \gamma^2_{i}) = \operatorname{sgn}(\gamma^2_{i+1}).\end{aligned}$$

With these we have

$$\begin{aligned}\operatorname{sgn}\left(C_0(\gamma^3_{i+2} - \gamma^3_{i+1})\right) &= \operatorname{sgn}(\gamma^3_{i+1} - \gamma^3_{i+2}) = \operatorname{sgn}(\gamma^2_{i+1}) = \operatorname{sgn}(y_{i+1} - y_i),\\ \operatorname{sgn}\left(C_1(\gamma^3_{i+3} - \gamma^3_{i+2})\right) &= \operatorname{sgn}(\gamma^3_{i+3} - \gamma^3_{i+2}) = -\operatorname{sgn}(\gamma^3_{i+2}) = \operatorname{sgn}(\gamma^2_{i+1} - \gamma^2_{i+2}),\\ &= \operatorname{sgn}(\gamma^2_{i+1}) = \operatorname{sgn}(y_{i+1} - y_i).\end{aligned}$$

For (b) *we can use the same calculation as above for the second term since we were not using* (a)*. The sign of the first term is*

$$\begin{aligned}\operatorname{sgn}\left(C_0(\gamma^3_{i+2} - \gamma^3_{i+1})\right) &= \operatorname{sgn}(\gamma^3_{i+1} - \gamma^3_{i+2}) = \operatorname{sgn}(\gamma^3_{i+1}) = \operatorname{sgn}(\gamma^2_{i+1} - \gamma^2_{i+2}),\\ &= \operatorname{sgn}(\gamma^2_{i+1}) = \operatorname{sgn}(y_{i+1} - y_i).\end{aligned}$$

Case 6. *The last configuration is* $S_i = \{C_i, C_{i+1}, C_{i+2}\}$, $S_{i+1} = \{C_{i+1}, C_{i+2}, C_{i+3}\}$, *equivalent to the conditions*

$$|\gamma^2_i| > |\gamma^2_{i+1}|, \quad |\gamma^3_{i+1}| > |\gamma^3_{i+2}|, \quad |\gamma^2_{i+1}| > |\gamma^2_{i+2}|, \quad |\gamma^3_{i+2}| > |\gamma^3_{i+3}|, \tag{3.89}$$

with a reconstructed jump of the form

$$jR_{i+1/2} \approx C_0(\gamma^3_{i+3} - \gamma^3_{i+2}). \tag{3.90}$$

We recover

$$\begin{aligned}\operatorname{sgn}\left(C_0(\gamma^3_{i+3} - \gamma^3_{i+2})\right) &= \operatorname{sgn}(\gamma^3_{i+3} - \gamma^3_{i+2}) = -\operatorname{sgn}(\gamma^3_{i+2}) = \operatorname{sgn}(\gamma^2_{i+1} - \gamma^2_{i+2}),\\ &= \operatorname{sgn}(\gamma^2_{i+1}) = \operatorname{sgn}(y_{i+1} - y_i).\end{aligned}$$

This completes the proof of the sign property of the reconstruction method for grids as $\Delta x \to 0$ or the shape parameter $\varepsilon \to 0$.

□

3.3 RBF-TeCNOp method

Based on the theory of entropy stable schemes and the work of Fjordholm et al. [35] we introduce the RBF-TeCNOp scheme. By using Algorithm 3.1 with (3.24) for calculating the least oscillatory stencil, Theorem 3.11 shows that the sign property holds for 2nd and 3rd degree reconstruction in the limit of $\Delta x \to 0$. We conjecture that this result

generalizes to higher order reconstructions. Thus, by combining the framework proposed in [35] with the RBF reconstruction using multiquadratics we recover an entropy stable essentially nonoscillatory RBF-based finite difference method of arbitrary high order. Furthermore, we use the RBF-RA algorithm to circumvent ill-conditioning in the reconstruction step [127].
In more detail, for constructing a p-th order RBF-TeCNOp method of the form (2.86) we use an entropy conservative flux of order $2k$ with $k = \lceil p/2 \rceil$ (see Theorem 2.4) and an ENO based RBF reconstruction (Algorithm 3.1) on the scaled entropy variables of order p with multiquadratics of order $p-1$.
Based on the Roe diffusion operator

$$R|\Lambda|R^{-1}[\![u]\!], \tag{3.91}$$

with the eigenvector matrix R and the diagonal matrix of the eigenvalues Λ, evaluated at the Roe average, we are choosing R and Λ in the same way. By Merriam [90] there is a scaling of the eigenvectors such that $RR^T = u_v = \partial_v u(v_{i+1/2})$. Thus, we get the relation

$$R|\Lambda|R^{-1}[\![u]\!] \approx R|\Lambda|R^{-1}u_v[\![v]\!] = R|\Lambda|R^T[\![v]\!], \tag{3.92}$$

that has a similar structure to that of a diffusion operator (2.87). The numerical diffusion term is

$$D_{i+1/2}\langle\!\langle v\rangle\!\rangle_{i+1/2} = R_{i+1/2}|\Lambda_{i+1/2}|\langle\!\langle w\rangle\!\rangle_{i+1/2}, \tag{3.93}$$

with the scaled entropy variables (2.89). Furthermore, we choose

$$\Lambda_{i+1/2} = \mathrm{diag}(\lambda_1(u_{i+1/2}), \ldots, \lambda_N(u_{i+1/2})) \tag{3.94}$$

and $u_{i+1/2} = \frac{u_i + u_{i+1}}{2}$ with the eigenvalues $\Lambda(u)$ of the Jacobian $\nabla_u f$.
It is important to note that the ill-conditioning of the interpolation matrix does not only affect the evaluation of the reconstruction; it also affects the calculation of the smoothness indicator which is based on the sum of the squares of the coefficients of the RBF-part of the interpolation.
From the theory we expect that the error of the interpolation with infinitely smooth RBFs decreases for smaller shape parameters. However, computations suggest that the choice of the stencil does not depend on the shape parameter. Thus, we calculate the stencil with respect to a stable shape parameter.

3.4 Entropy stable RBF-finite volume method

The combination of the RBF interpolation with finite volume methods works analogeously to the RBF-TeCNOp Method. Aboiyar et al. [2] combine in their work a

high-order WENO approach with a polyharmonic spline reconstruction and Bigoni and Hesthaven [11] apply a high-order WENO approach to multiquadratics.
We construct an entropy stable finite volume method of second order that is essentially nonoscillatory by combining (2.97) with a second order accurate RBF interpolation that acts on the scaled entropy variables. Therefore, we are using multiquadratics with the smoothness indicator (3.24), combined with Algorithm 3.1 and the vector valued rational approximation to ensure a stable evaluation of the interpolation function.
We conjecture the sign property for the RBF reconstruction on mean values that is based on Algorithm 3.1 which is fulfilled in the pointwise case for second and third degree reconstruction in the limit $\Delta x \to 0$ (Theorem 3.11). Under this assumption we recover a second order entropy stable finite volume (RBF-EFV2) method. This method can be generalized to general grids in higher dimensions [103].

3.5 Numerical results

In this section, we are evaluating the second order entropy stable finite volume (EFV2) and the TeCNOp methods based on RBF reconstructions for one-dimensional problems and compare it with its original version. Note that in one dimension we do not expect to do better than the classical methods, but also not worse.
For the polynomial reconstruction we use the original algorithm from [57] to select the stencil and in the RBF case we use Algorithm 3.1. The EFV2 and TeCNOp methods are based on the diffusion term (3.93).
The parameters for the vector-valued rational approximation are chosen as described in [127]. Further, we choose the shape parameter $\varepsilon = 0.1$ for all examples.

3.5.1 Linear advection equation

We consider the linear advection equation (2.3) with wave speed $a = 1$ and periodic boundary conditions [83]. To construct a high-order accurate scheme we choose the entropy pair (2.76) with its second order entropy conservative flux (2.78). Further, we use a 5th order SSPRK method for the time discretization in the TeCNOp method [49]. For the EFV2 method we use the second order entropy conservative flux plus a third order SSPRK method in time.
The convergence results for the smooth initial conditions are shown in Table 3.1. The L_1-errors are the same for the different reconstruction methods for grids of size smaller than $1/32$ and their convergence rates are as expected.
In Table 3.2 we present a comparison of the runtime for the 5th order TeCNO scheme with RBF reconstruction using the RBF-RA algorithm, RBF reconstruction evaluated at a stable shape parameter. The RBF method based on the RBF-RA algorithm solves multiple times the same system of equations with different shape parameters to approximate the final one, which helps to explain the difference between the first two

columns. The difference between the RBF and the polynomial reconstruction comes from the fact that the RBF algorithm in each recursive step of Algorithm 3.1 solves a system of equations and the polynomial case calculates just the next divided difference.

	N	Linear advection eq.				Burgers' eq.			
		RBF Reconstr		Poly Reconstr		RBF Reconstr		Poly Reconstr	
		error	rate	error	rate	error	rate	error	rate
TeCNO2	16	2.27e-02	-	2.26e-02	-	3.01e-02	-	2.97e-02	-
	32	7.26e-03	1.64	7.26e-03	1.64	9.01e-03	1.74	9.01e-03	1.72
	64	2.06e-03	1.82	2.06e-03	1.82	2.46e-03	1.87	2.46e-03	1.87
	128	5.44e-04	1.92	5.44e-04	1.92	6.88e-04	1.84	6.88e-04	1.84
	256	1.45e-04	1.91	1.45e-04	1.91	1.88e-04	1.87	1.88e-04	1.87
TeCNO3	16	1.48e-03	-	1.48e-03	-	5.19e-03	-	5.17e-03	-
	32	1.89e-04	2.97	1.89e-04	2.97	9.23e-04	2.49	9.23e-04	2.49
	64	2.36e-05	3.00	2.36e-05	3.00	1.47e-04	2.65	1.47e-04	2.65
	128	2.96e-06	3.00	2.96e-06	3.00	2.52e-05	2.54	2.52e-05	2.54
	256	3.70e-07	3.00	3.70e-07	3.00	4.13e-06	2.61	4.13e-06	2.61
TeCNO4	16	5.61e-04	-	5.60e-04	-	2.84e-03	-	2.84e-03	-
	32	3.98e-05	3.82	3.98e-05	3.82	5.37e-04	2.40	5.37e-04	2.40
	64	2.62e-06	3.92	2.62e-06	3.93	5.76e-05	3.22	5.76e-05	3.22
	128	1.74e-07	3.91	1.74e-07	3.90	4.97e-06	3.54	4.97e-06	3.54
	256	1.14e-08	3.93	1.14e-08	3.93	6.97e-07	2.83	6.97e-07	2.83
TeCNO5	16	4.40e-05	-	4.40e-05	-	1.17e-03	-	1.17e-03	-
	32	1.40e-06	4.98	1.40e-06	4.98	2.90e-04	2.01	2.90e-04	2.01
	64	4.43e-08	4.98	4.43e-08	4.98	1.19e-05	4.61	1.19e-05	4.61
	128	1.47e-09	4.92	1.47e-09	4.92	6.84e-07	4.12	6.84e-07	4.12
	256	5.50e-11	4.74	5.50e-11	4.74	1.81e-07	1.92	1.81e-07	1.92
EFVM2	16	2.25e-02	-	2.24e-02	-	2.68e-02	-	2.68e-02	-
	32	7.26e-03	1.63	7.25e-03	1.63	8.10e-03	1.73	8.10e-03	1.73
	64	2.06e-03	1.82	2.06e-03	1.82	2.28e-03	1.83	2.28e-03	1.83
	128	5.44e-04	1.92	5.44e-04	1.92	6.40e-04	1.84	6.40e-04	1.83
	256	1.45e-04	1.91	1.45e-04	1.91	1.78e-04	1.85	1.78e-04	1.85

Table 3.1 – Convergence rates of TeCNOp and EFV2 methods using multiquadratics and polynomials for the linear advection and Burgers' equation on $[0,1]$ at time $T = 0.1$. We use periodic boundary conditions and $u_0(x) = \sin(2\pi x)$, shape parameter $\varepsilon = 0.1$, $\mathrm{CFL} = 0.5$.

3.5.2 Burgers' equation

For the Burgers' equation (2.4) we study the order of convergence and verify that the methods handle discontinuities without introducing major oscillations. The EFV2 and TeCNOp method are based on the entropy pair (2.79) and the entropy conservative flux (2.81) which is used to construct a high-order scheme. For the time discretization we use a 5th order SSPRK method [49]. Furthermore, we choose the domain $[0,1]$ and the initial conditions $u_0(x) = \sin(2\pi x)$.

	RBF + RBF-RA	RBF	Poly
16	4.3	3.2	2.1
32	7.4	5.0	3.7
64	21.8	12.6	6.6
128	73.5	38.0	16.9
256	279.5	132.4	44.9

Table 3.2 – Runtime comparison for the 5th order method solving the linear advection equation.

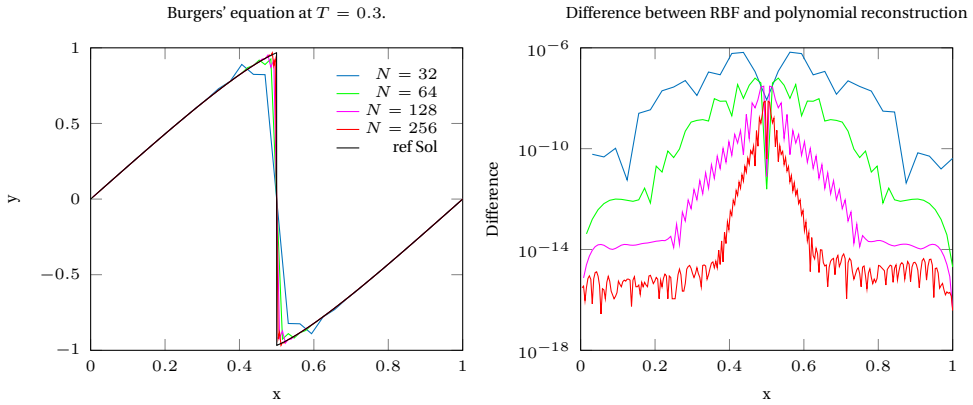


Figure 3.2 – Burgers' equation on $[0, 1]$ at time $T = 0.3$ with continuous initial condition $u_0 = \sin(2\pi x)$, shape parameter $\varepsilon = 0.1$, CFL $= 0.5$, solved by TeCNO5.

A detailed analysis of the convergence is shown in Table 3.1. The convergence rate is as expected and the errors of the two different methods (polynomial reconstruction and RBF reconstruction) coincide. At time $T = 0.3$ a discontinuity emerges at $x = 0.5$. This can be resolved accurately with vanishing oscillations (Fig. 3.2). Furthermore, we observe that the difference between the reconstruction methods vanishes in the smooth part while at the shock it stays small.

3.5.3 Shallow water equations

For the shallow water equations (2.7) we consider the dambreak problem with the initial conditions

$$(h_0, m_0) = \begin{cases} (1.5, 0) & \text{if } |x| \leqslant 0.2 \\ (1, 0) & \text{if } |x| > 0.2 \end{cases}, \tag{3.95}$$

on the domain $[-1, 1]$ and periodic boundary conditions. We use a second order entropy stable flux (2.82) to construct a high-order flux and a third order SSPRK method for the time integration.
Fjordholm et al. [35] showed that the standard TeCNO scheme behaves similar to the

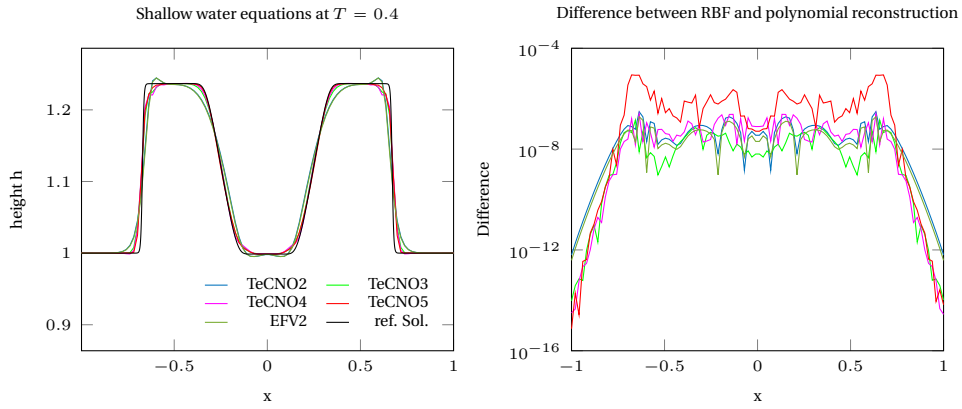


Figure 3.3 – Shallow water equations on $[-1,1]$ at time $T = 0.4$ with $N = 100$, shape parameter $\varepsilon = 0.1$, CFL $= 0.5$, solved by TeCNO and EFV2.

ENO-MUSCL scheme. The same holds for the RBF-TeCNOp scheme and the RBF-EFV2 scheme as seen in Fig. 3.3. The difference between the RBF methods and the polynomial scheme is around $\mathcal{O}(10^{-6})$ in the region where the discontinuity passed and much smaller in smooth regions.

3.5.4 Euler equations

The one-dimensional Euler equations are represented by

$$\begin{pmatrix} \rho \\ m \\ E \end{pmatrix}_t + \begin{pmatrix} m \\ \frac{m^2}{\rho} + p \\ \frac{m}{\rho}(E+p) \end{pmatrix}_x = 0. \tag{3.96}$$

After setting $m_2 = 0$ and neglecting the third component, we can take the results from Sections 2.1.1 and 2.1.4. As for the shallow water equations we use a third order SSPRK method and as a second order entropy conservative flux we use the KEPEC-flux (2.84). Further, we choose $\gamma = 1.4$ which simulates a diatomic gas such as air. Note that the reconstruction is done in the characteristic variables $\mathbf{V} = R^{-1}\mathbf{U}$, with the eigenvectors R given in (2.13).

Sod's shock tube problem

Sod's shock tube problem is a Riemann problem where two gases with different densities collide. A rarefaction wave emerges, followed by a contact and a shock discontinu-

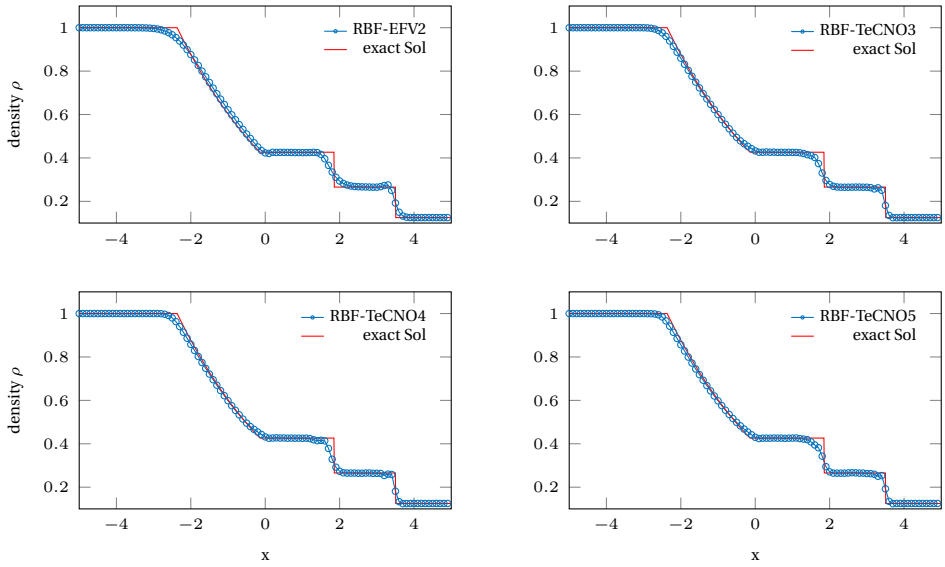


Figure 3.4 – Sod's shock tube problem on $[-5,5]$ at time $T=2$ with $N=100$, shape parameter $\varepsilon=0.1$, $\mathrm{CFL}=0.3$, solved by RBF-TeCNOp and RBF-EFV2.

ity. The initial conditions are

$$(\rho_0, m_0, p_0) = \begin{cases} (1,0,1) & \text{if } x < 0 \\ (0.125,0,0.1) & \text{if } x \geqslant 0 \end{cases}, \tag{3.97}$$

where $m = u\rho$. The results at time $T=2$ of the RBF-TeCNOp and RBF-EFV2 methods are shown in Fig. 3.4, clearly representing the rarefaction wave, the contact, and the shock discontinuity. Comparing the solutions obtained with polynomial reconstruction or with RBF reconstruction, we see in Fig. 3.5 that their difference is decreasing with the refinement of the grid.

Lax shock tube problem

The Lax shock tube problem is another Riemann problem with the initial conditions

$$(\rho_0, m_0, p_0) = \begin{cases} (0.445, 0.698, 3.528) & \text{if } x < 0 \\ (0.5,0,0.571) & \text{if } x \geqslant 0 \end{cases}, \tag{3.98}$$

where $m = u\rho$. The RBF-TeCNOp methods of order three to five represent the big shock in the density sharply with just $N=100$ points, see Fig. 3.6. The second order RBF-EFV2 method does not perform well for this case.

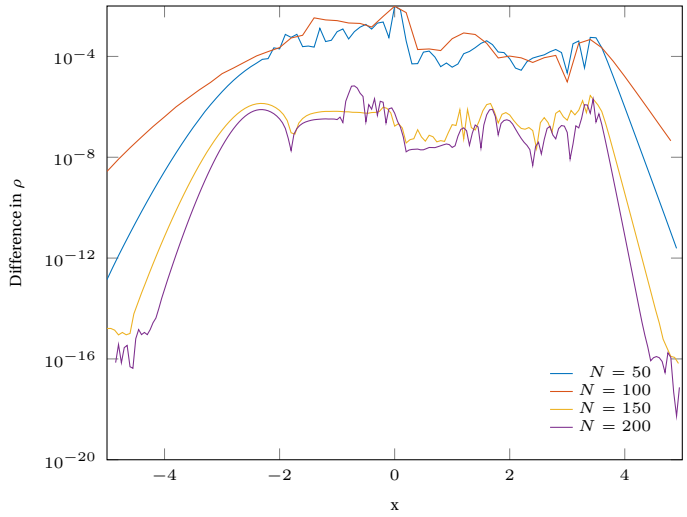


Figure 3.5 – Pointwise difference between RBF and polynomial based reconstruction EFV2 method for the Sod's shock tube problem on $[-5,5]$ at time $T = 2$ for different number of grid points N, shape parameter $\varepsilon = 0.1$, CFL $= 0.3$.

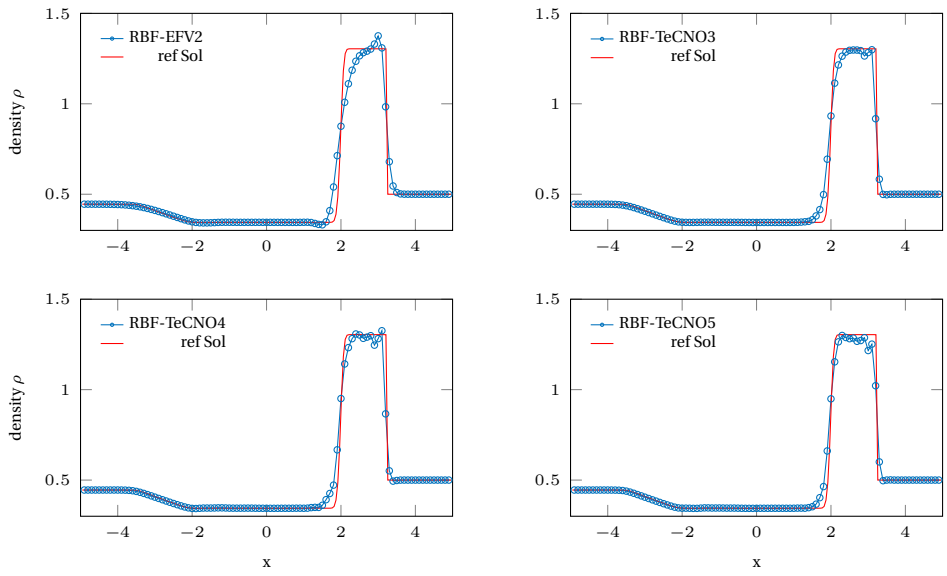


Figure 3.6 – Lax shock tube problem on $[-5,5]$ at time $T = 1.3$ with $N = 100$, shape parameter $\varepsilon = 0.1$, CFL $= 0.3$, solved by RBF-TeCNOp and RBF-EFV2.

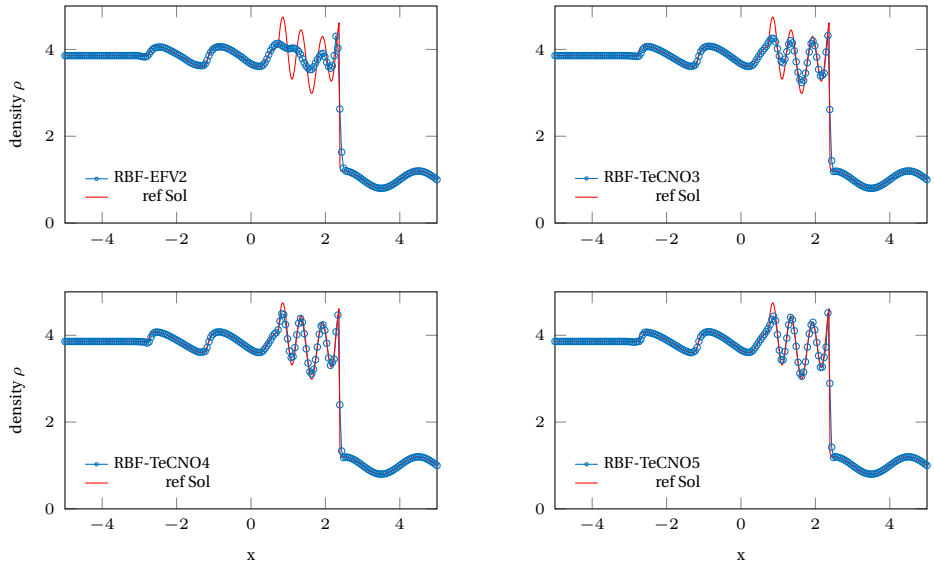


Figure 3.7 – Shu-Osher problem on $[-5,5]$ at time $T = 1.8$ with $N = 200$, shape parameter $\varepsilon = 0.1$, $\mathrm{CFL} = 0.3$ solved by RBF-TeCNOp and RBF-EFV2.

Shu-Osher shock-entropy wave interaction problem

The Shu-Osher problem models a shock-turbulence interaction in which a shock interacts with a low frequency wave. Due to this interaction, high-frequency oscillations develop over time. The initial conditions are

$$(\rho_0, m_0, p_0) = \begin{cases} (3.857143, 2.629369, 10.33333) & \text{if } x < -4 \\ (1 + 0.2\sin(5x), 0, 1) & \text{if } x \geqslant -4 \end{cases}, \tag{3.99}$$

where $m = u\rho$. The RBF-TeCNOp methods of order larger than three recover the high frequency oscillations well. The RBF-EFV2 method fits the low order oscillations and the shock, but not the high frequency wave due to excessive dissipation.

Low density problem

The low density problem on $[0,1]$ is a Riemann problem that tests the ability of preserving positive density and pressure. The initial condition are

$$(\rho_0, m_0, p_0) = \begin{cases} (1, -2, 0.4) & \text{if } x < 0.5 \\ (1, 2, 0.4) & \text{if } x \geqslant 0.5 \end{cases}, \tag{3.100}$$

with Neumann boundary conditions.
In Fig. 3.8 we observe the increasing accuracy of the RBF-TeCNOp method with increasing order p. Note that by choosing the wrong smoothness indicator, negative

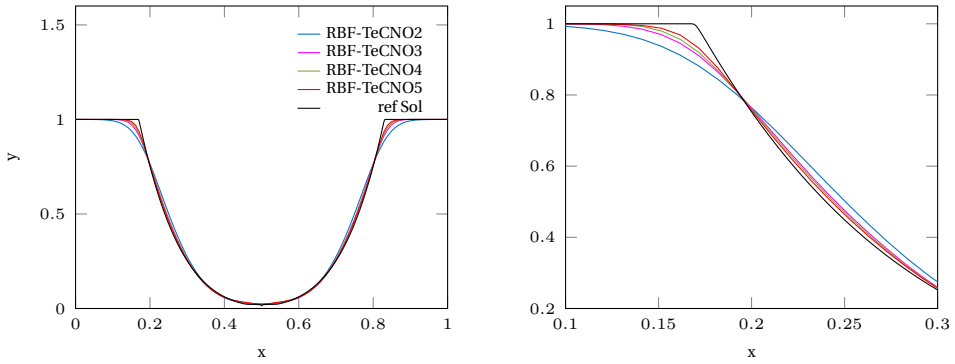


Figure 3.8 – Low density problem on $[0,1]$ at time $T = 0.12$ with $N = 100$, shape parameter $\varepsilon = 0.1$, $\mathrm{CFL} = 0.1$ solved by RBF-TeCNOp.

pressures will occur.

Two interacting blast waves

A more complex one dimensional example is the two interacting blast waves, introduced by Woodward and Colella [126]. It is based on two blast waves that interact and introduce low pressures and densities. Its initial condition is

$$(\rho_0, m_0, p_0) = \begin{cases} (1,0,1000) & \text{if } x < 0.1 \\ (1,0,0.01) & \text{if } 0.1 \leqslant x < 0.9 \\ (1,0,100) & \text{if } x \geqslant 0.9 \end{cases}. \tag{3.101}$$

Compared to the low density problem we add here an additional challenge caused by the collision of the two shocks. Note that the standard version of the TeCNO method always produces negative pressures or densities for the RBF and polynomial reconstruction.
Thus, we introduce a more complicated symmetric positive definite dissipation operator (2.87), similar to the one introduced by Derigs et al. [23]. The goal is to mimic the more dissipative Rusanov-type diffusion operator such that

$$\alpha[\![u]\!]_{i+1/2} = D_{i+1/2}[\![v]\!]_{i+1/2}. \tag{3.102}$$

See Appendix A for more details.
The results with the new dissipation matrix approximate the correct solution and we can see an improvement with increasing order (Fig. 3.9) and for increasing number of points (Fig. 3.10).

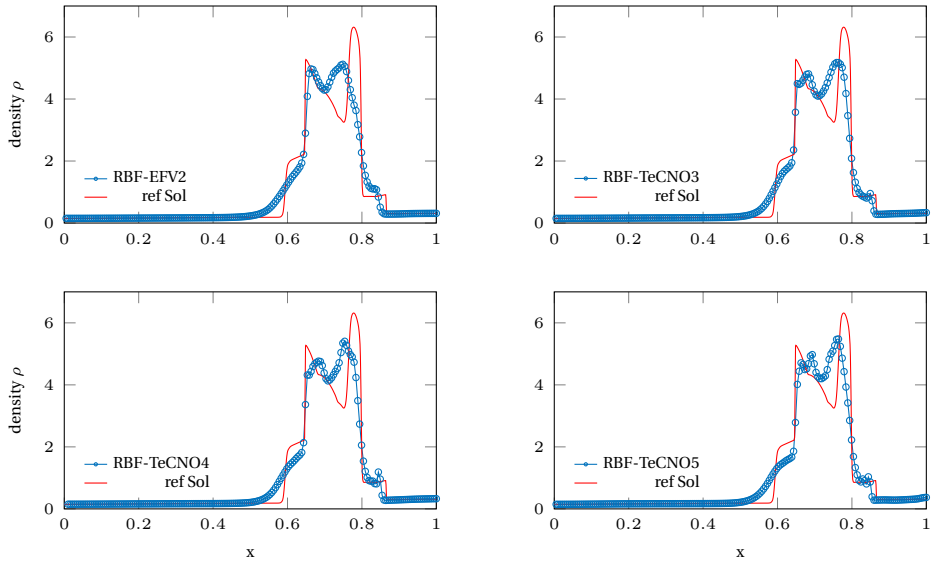


Figure 3.9 – WC blast wave problem on $[0,1]$ at time $T = 0.038$ with $N = 200$, shape parameter $\varepsilon = 0.1$, $\mathrm{CFL} = 0.1$ solved by RBF-TeCNOp and RBF-EFV2.

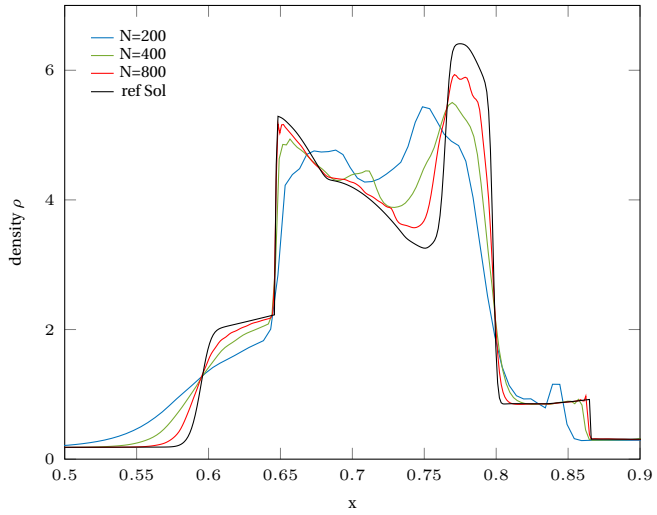


Figure 3.10 – WC blast wave problem on $[0,1]$ at time $T = 0.038$ with shape parameter $\varepsilon = 0.1$, $\mathrm{CFL} = 0.1$ solved by RBF-TeCNO4.

4 High-order RBF-based ENO method on general two-dimensional domains

After introducing entropy stable reconstruction methods for one-dimensional problems, which can be generalized to structured grids by tensor products, we now seek to take advantage of the flexibility of RBFs in multiple dimensions. In this chapter, we propose a high-order ENO method based on radial basis function to solve hyperbolic conservation laws on general two-dimensional grids. The radial basis function reconstruction offers a flexible way to deal with ill-conditioned cell constellations. We introduce a smoothness indicator based on RBFs and a stencil selection algorithm suitable for general meshes, which is based on the one introduced in the previous chapter and the one described in [56]. Furthermore, we develop a stable method to evaluate the RBF reconstruction in the finite volume setting which circumvents the stagnation of the error and keeps the condition number of the reconstruction bounded. The results in this chapter are published in [63].

4.1 Stable RBF evaluation for fixed number of nodes

As discussed in Section 2.6, the ill-conditioning of the RBF interpolation is a well-known challenge. However, RBFs in finite volume methods are of a slightly different nature. In general, the RBF approximation achieves exponential order of convergence for smooth functions by increasing the number of interpolation nodes in a certain domain. The setting for the reconstruction in finite volume methods is different since the number of interpolation points remains fixed at a rather low number of nodes and only the fill-distance is reduced.
Based on [39, 38] it is known that the combination of polyharmonic or Gaussian RBFs with polynomials overcomes the stagnation error. Bayona [7] shows that under certain assumptions the order of convergence is ensured by the polynomial part.
We propose to use multiquadratic rather than polyharmonic or Gaussian RBFs to allow the use of the smoothness indicator developed in the previous chapter. Since the RBFs

are only used to ensure solvability of the linear system, we use

$$\varepsilon = \frac{1}{\Delta x}, \tag{4.1}$$

as the shape parameter with the separation distance $\Delta x := \min_{i \neq j} \|\mathbf{x}_i - \mathbf{x}_j\|$ for the interpolation nodes $\mathbf{x}_1, \dots, \mathbf{x}_n$ with $n \in \mathbb{N}$. To control the conditioning of the polynomial part we use the basis

$$p_i(\mathbf{x}) = \tilde{p}_i(\varepsilon(\mathbf{x} - \tilde{\mathbf{x}})), \tag{4.2}$$

for $i = 1, \dots, m$ with $\tilde{p}_i \in \{\mathbb{R}^d \to \mathbb{R}, \mathbf{x} \mapsto x_1^{\alpha_1} \dots x_d^{\alpha_d} | \ \sum_{j=1}^d \alpha_j < l, \alpha_j \in \mathbb{N}\}$, $\deg(\tilde{p}_i) \leqslant \deg(\tilde{p}_{i+1})$ and $\tilde{\mathbf{x}} \in \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$. The best choice for $\tilde{\mathbf{x}}$ would be the barycenter of the stencil. However, to use the same polynomials for different stencils in the ENO scheme we choose the central node.

Remark 4.1. *The interpolation matrix is the same as the one with the interpolation basis $\tilde{p}_i$ with $i = 1, \dots, m$, the RBFs with shape parameter 1, and the nodes $\tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_n$ with $\tilde{\mathbf{x}}_i = \varepsilon(\mathbf{x}_i - \tilde{\mathbf{x}})$. This holds true for any $\Delta x \to 0$ and $\Delta \tilde{x} = 1$. Thus, the interpolation step in the finite volume method has the same condition number for all refinements as long as the interpolation nodes have a similar distribution.*

4.1.1 Stability estimate for RBF coefficients

In this section, we analyze the stability of the RBF interpolation based on (4.1) and (4.2) and show that the stability of the RBF coefficients depends only on the number of the interpolation nodes n. For the one-dimensional case we show that the stability of the polynomial coefficients depends on n and the ratio of the maximum distance between the interpolation points Dx and the minimum distance Δx. For higher dimensions we conjecture that a similar result holds.
The analysis is based on results from Schaback [107] and Wendland [124].

Lemma 4.1 (Stability estimate [107])**.** *For* (2.110) *there holds the stability estimate*

$$\frac{\|\Delta a\|_2}{\|a\|_2} \leqslant \frac{\lambda_{max}}{\lambda_{min}} \frac{\|\Delta f\|_2}{\|f - Pb\|_2}, \tag{4.3}$$

with $\lambda_{min} := \inf_{a \neq 0, P^T a = 0} \frac{a^T A a}{a^T a}$ and λ_{max} the maximal eigenvalue. Further, there exists an estimate for the polynomial coefficients

$$\frac{\|\Delta b\|_2}{\|b\|_2} \leqslant \frac{\lambda_{max, P^T P}}{\lambda_{min, P^T P}} \frac{\|P^T(\Delta f - A \Delta a)\|_2}{\|P^T(f - Aa)\|_2}, \tag{4.4}$$

$$\leqslant \left(1 + \frac{\lambda_{max}}{\lambda_{min}}\right) \frac{\lambda_{max, P^T P}}{\lambda_{min, P^T P}} \frac{\|P^T \Delta f\|_2}{\|P^T(f - Aa)\|_2}, \tag{4.5}$$

with the maximal and minimal eigenvalue of $P^T P$, $\lambda_{max,P^T P}$, $\lambda_{min,P^T P}$.

Thus, the stability of the method depends on the ratios

$$\lambda_{max}/\lambda_{min} \text{ and } \lambda_{max,P^T P}/\lambda_{min,P^T P}.$$

The maximal eigenvalues can be estimated by

$$\lambda_{max} = \sup_{a\neq 0} \frac{a^T A a}{a^T a} = \|A\|_2 \leqslant \|A\|_F \leqslant n \max_{i,j} |A_{i,j}|. \tag{4.6}$$

Note that λ_{min} is not the smallest eigenvalue of A, but its definition is similar. Schaback [107] established the following lower bound for the d-dimensional interpolation

Lemma 4.2 (Lower bound of λ_{min} [107])**.** *Given an even conditionally positive definite function* ϕ *with the positive generalized Fourier transform* $\hat{\phi}$*. It holds that*

$$\lambda_{min} \geqslant \frac{\varphi_0(M)}{2\Gamma(d/2+1)} \Big(\frac{M}{2\sqrt{\pi}}\Big)^d, \tag{4.7}$$

with the function

$$\varphi_0(r) := \inf_{\|\omega\|_2 \leqslant 2r} \hat{\phi}(\omega), \tag{4.8}$$

for $M > 0$ *satisfying*

$$M \geqslant \frac{12}{\Delta x} \Big(\frac{\pi \Gamma^2(d/2+1)}{9}\Big)^{1/(d+1)}, \tag{4.9}$$

or

$$M \geqslant \frac{6.38d}{\Delta x}, \tag{4.10}$$

and with

$$\Gamma(x) = \int_0^\infty t^{x-1} \exp(-t) \mathrm{d}t, \qquad \mathrm{Re}(x) > 0. \tag{4.11}$$

It remains to estimate $\varphi_0(M)$ depending on the RBFs. Some estimates for the examples in Table 2.1 are

Lemma 4.3 (Estimate of φ_0 for multiquadratics [107])**.** *Let* ϕ *be the multiquadratic RBF, then*

$$\varphi_0(M) \geqslant \frac{\pi^{d/2} \Gamma(d/2+\nu) M^{-d-2\nu} \exp(-2M/\varepsilon)}{\Gamma(-\nu)}. \tag{4.12}$$

Note that the lower bound of φ_0 of Lemma 4.3 is zero for $\nu \in \mathbb{N}$.

Lemma 4.4 (Estimate of φ_0 for Gaussians [124])**.** *Let ϕ be the Gaussian RBF, then*

$$\varphi_0(M) = (2\varepsilon^2)^{-d/2} \exp(-M^2/\varepsilon^2). \tag{4.13}$$

Lemma 4.5 (Estimate of φ_0 for polyharmonics [124])**.**
Let $\phi(r) = (-1)^{k+1} r^{2k} \log(r)$ be a polyharmonic RBF, then

$$\varphi_0(M) = (-1)^{k+1} 2^{2k-1+d/2} \Gamma(k + d/2) k! (2M)^{-d-2k}. \tag{4.14}$$

Given these results, we obtain the following

Corollary 4.6. *By using the shape parameter* (4.1) *we recover*

$$\frac{\|\Delta a\|_2}{\|a\|_2} \leqslant C(n,d) \|\Delta f\|_2, \tag{4.15}$$

for all $\mathbf{x}_1, \ldots, \mathbf{x}_n$, $n \in \mathbb{N}$ and a constant $C(n,d)$ which depends on the number of interpolation nodes n and the dimension d.

Proof. From Remark 4.1 we conclude

$$a := a(\mathbf{x}_1, \ldots, \mathbf{x}_n) = a(\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_n) =: \tilde{a}. \tag{4.16}$$

From the Lemmas 4.2, 4.3 and (4.6) we obtain

$$\frac{\|\Delta a\|_2}{\|a\|_2} = \frac{\|\Delta \tilde{a}\|_2}{\|\tilde{a}\|_2} \leqslant C(n,d,\Delta\tilde{x}) \|\Delta f\|_2 = C(n,d,1) \|\Delta f\|_2, \tag{4.17}$$

with a constant $C(n,d,\Delta x)$ which depends on n, d and Δx. □

Hence, the stability of the RBF coefficients depends only on the number of interpolation nodes n. This analysis is dimension independent. To utilize Lemma 4.1 it remains to estimate the ratio $\lambda_{max,P^TP}/\lambda_{min,P^TP}$.

4.1.2 Stability estimate for polynomial coefficients

The analysis of the Gram matrix $G := P^T P \in \mathbb{R}^{m \times m}$ is more challenging. For the polynomial basis (4.2) we have

$$G_{ij} = \sum_{l=1}^{n} p_i(\mathbf{x}_l) p_j(\mathbf{x}_l). \tag{4.18}$$

We note that $P = \tilde{P}$ where $(\tilde{P})_{i,j} = \tilde{p}_i(\tilde{\mathbf{x}}_j)$ with $\tilde{\mathbf{x}}_j = \varepsilon(\mathbf{x}_j - \tilde{\mathbf{x}})$. In the one-dimensional case, the following estimate of the condition number holds for the Vandermonde matrix.

Lemma 4.7 (Conditioning of the Vandermonde matrix in one dimension [45])**.** *Let V_n be the Vandermonde matrix $(V_n)_{i,j} = z_j^i$ with $z_i \neq z_j$ for $i \neq j$ and $z_j \in \mathbb{C}$. It holds that*

$$\max_j \prod_{i \neq j} \frac{\max(1, |z_i|)}{|z_j - z_i|} < \|V_n^{-1}\|_\infty \leqslant \max_j \prod_{i \neq j} \frac{1 + |z_i|}{|z_j - z_i|}. \tag{4.19}$$

Corollary 4.8.

$$\frac{\lambda_{max,P^TP}}{\lambda_{min,P^TP}} \leqslant \left(\frac{Dx}{\Delta x} + 1\right)^2 \left(\frac{Dx}{\Delta x}\right)^{2n} \frac{n^4}{\left(\lfloor n/2 - 1 \rfloor !\right)^4}, \tag{4.20}$$

with $Dx = \max_{i \neq j} |x_i - x_j|$.

Proof. We start with estimating $\|P\|_\infty$

$$\|P\|_\infty = \max_i \sum_{j=1}^n \left(\frac{x_i - x_1}{\Delta x}\right)^{j-1} \leqslant \max_i \sum_{j=1}^n \left(\frac{Dx}{\Delta x}\right)^{j-1} = \frac{\left(\frac{Dx}{\Delta x}\right)^n - 1}{\frac{Dx}{\Delta x} - 1}, \tag{4.21}$$

$$\leqslant n \left(\frac{Dx}{\Delta x}\right)^n, \tag{4.22}$$

To estimate the norm of P^{-1} we use Lemma 4.7

$$\begin{aligned}
\|P^{-1}\|_\infty &\leqslant \max_i \prod_{j \neq i} \frac{1 + |\tilde{x}_j|}{|\tilde{x}_i - \tilde{x}_j|} = \max_i \prod_{j \neq i} \frac{\Delta x + |x_j - x_1|}{|x_i - x_j|}, \\
&\leqslant \max_i \prod_{j \neq i} \frac{\Delta x + Dx}{|j - i| \Delta x} = \left(\frac{Dx}{\Delta x} + 1\right) \max_i \frac{1}{\prod_{j \neq i} |j - i|}, \\
&\leqslant \left(\frac{Dx}{\Delta x} + 1\right) \frac{1}{\prod_{j \neq \lfloor n/2 \rfloor} |j - \lfloor n/2 \rfloor|} \leqslant \left(\frac{Dx}{\Delta x} + 1\right) \frac{1}{\prod_{j < \lfloor n/2 \rfloor} |j|^2}, \\
&\leqslant \left(\frac{Dx}{\Delta x} + 1\right) \frac{1}{\left(\lfloor n/2 - 1 \rfloor !\right)^2}.
\end{aligned}$$

Furthermore, we have the standard bounds

$$\frac{1}{\sqrt{n}} \|A\|_\infty \leqslant \|A\|_2 \leqslant \|A\|_\infty \sqrt{m}, \tag{4.23}$$

for $A \in \mathbb{R}^{m \times n}$. From [120] we recover

$$\text{cond}_2 P^T P = (\text{cond}_2 P)^2, \tag{4.24}$$

when $n = m$. Combined, this yields

$$\frac{\lambda_{max,P^TP}}{\lambda_{min,P^TP}} = \|P^{-1}\|_2^2\|P\|_2^2 \leqslant n^2\|P^{-1}\|_\infty^2\|P\|_\infty^2. \tag{4.25}$$

□

Applying Corollary 4.8 to uniformly distributed nodes in $\mathbb{R}$ we obtain $Dx/\Delta x = n-1$ and the condition number of P^TP is uniformly bounded for all Δx by

$$\frac{\lambda_{max,P^TP}}{\lambda_{min,P^TP}} \leqslant \frac{(n-1)^n n^3}{\left(\lfloor n/2-1\rfloor!\right)^2}. \tag{4.26}$$

The proof of this estimate does not hold true for two-dimensional interpolation. However, we conjecture that similar bounds hold, as is confirmed in Table 4.1. Note that the reconstructions from (2.47) are based on a stencil in a grid. Thus, $Dx/\Delta x$ is bounded for these interpolation problems.

4.1.3 Approximation by RBF interpolation augmented with polynomials

Considering ansatz (2.108) for the interpolation problem (2.105), (2.109) Bayona [7] shows, under the assumption of full rank of A and P, that the order of convergence is at least $\mathcal{O}(h^{l+1})$ based on the polynomial part. With similar techniques we can relax the assumptions of full rank of A by assuming ϕ to be a conditionally positive definite RBF of order $l+1$. We write $s_{f,X}$ for the interpolation function that interpolates the function f on the scattered set of nodes X.

Theorem 4.9. *Let f be an analytic multivariate function and ϕ a conditionally positive definite RBF of order $l+1$. Further, we assume the existence of a $\Pi_l(\mathbb{R}^d)$-unisolvent subset of X. It follows*

$$\|s_{f,X} - f\|_\infty \leqslant \mathcal{O}(h^{l+1}). \tag{4.27}$$

Proof. Let us consider $\mathbf{x}_0 \in \mathbb{R}^d$ where $\mathbf{x}_0$ does not have to be a node. By assuming that f is analytic, it admits a Taylor expansion in a neighborhood of $\mathbf{x}_0$

$$f(\mathbf{x}) = \sum_{k\geqslant 1} L_k[f(\mathbf{x}_0)]p_k(\mathbf{x}-\mathbf{x}_0), \tag{4.28}$$

with $L_k[f(\mathbf{x}_0)] \in \mathbb{R}$ the coefficients for f around $\mathbf{x}_0$, e.g., $L_k[f(\mathbf{x}_0)] = \frac{1}{k!}f^{(k)}(\mathbf{x}_0)$ for univariate functions. Thus, we recover

$$f|_X = (f(\mathbf{x}_i))_{i=1}^n = \sum_{k\geqslant 1} L_k[f(\mathbf{x}_0)]\vec{p}_k, \tag{4.29}$$

with $\vec{p}_k = (p_k(\mathbf{x}_i - \mathbf{x}_0))_{i=1}^n$.
We rewrite the coefficients from (2.110)

$$\begin{pmatrix} a \\ b \end{pmatrix} = \sum_{k \geqslant 1} L_k[f(\mathbf{x}_0)] \begin{pmatrix} \vec{a}_k \\ \vec{b}_k \end{pmatrix}, \tag{4.30}$$

with $\vec{a}_k \in \mathbb{R}^n$, $\vec{b}_k \in \mathbb{R}^m$ given by

$$\begin{pmatrix} A & P \\ P^T & 0 \end{pmatrix} \begin{pmatrix} \vec{a}_k \\ \vec{b}_k \end{pmatrix} = \begin{pmatrix} \vec{p}_k \\ 0 \end{pmatrix}, \tag{4.31}$$

and m the number of basis functions of $\Pi_l(\mathbb{R}^d)$. Since there exists a $\Pi_l(\mathbb{R}^d)$-unisolvent subset, and by the well-posedness of (4.31), we have

$$a_{k,i} = 0, \quad b_{k,j} = \delta_{k,j}, \tag{4.32}$$

for $i = 1, \ldots, n$ and $j, k = 1, \ldots, m$. This allows us to write the interpolation function as

$$\begin{aligned} s_{f,X}(\mathbf{x}) &= \sum_{i=1}^n a_i \phi(\mathbf{x} - \mathbf{x}_i) + \sum_{j=1}^m b_j p_j(\mathbf{x}), && (4.33) \\ &= \sum_{k=1}^m L_k[f(\mathbf{x}_0)] p_k(\mathbf{x}) + \sum_{k>m} \sum_{i=1}^n L_k[f(\mathbf{x}_0)] a_{k,i} \phi_i(\mathbf{x}) && (4.34) \\ &\quad + \sum_{k>m} \sum_{l=1}^s L_k[f(\mathbf{x}_0)] b_{k,l} p_l(\mathbf{x}), && (4.35) \end{aligned}$$

and recover

$$\begin{aligned} f(\mathbf{x}) - s_{f,X}(\mathbf{x}) &= \sum_{k>m} L_k[f(\mathbf{x}_0)] p_k(\mathbf{x}) - \sum_{k>m} \sum_{i=1}^n L_k[f(\mathbf{x}_0)] a_{k,i} \phi_i(\mathbf{x}) \\ &\quad - \sum_{k>m} \sum_{l=1}^m L_k[f(\mathbf{x}_0)] b_{k,l} p_l(\mathbf{x}) = r_m(\mathbf{x}) - s_{r_m,X}(\mathbf{x}). \end{aligned} \tag{4.36}$$

with $r_m(\mathbf{x}) = \sum_{k>m} L_k[f(\mathbf{x}_0)] p_k(\mathbf{x})$.
Given the estimate of De Marchi and Schaback [22]

$$\|s_{f,X}\|_\infty \leqslant C(\|f\|_{\ell_\infty(X)} + \|f\|_{\ell_2(X)}), \tag{4.37}$$

we conclude

$$\|f - s_{f,X}\|_\infty = \|r_m - s_{r_m,X}\|_\infty \leqslant C h^{l+1}, \tag{4.38}$$

with $\|r_m\|_\infty \leqslant C h^{l+1}$. □

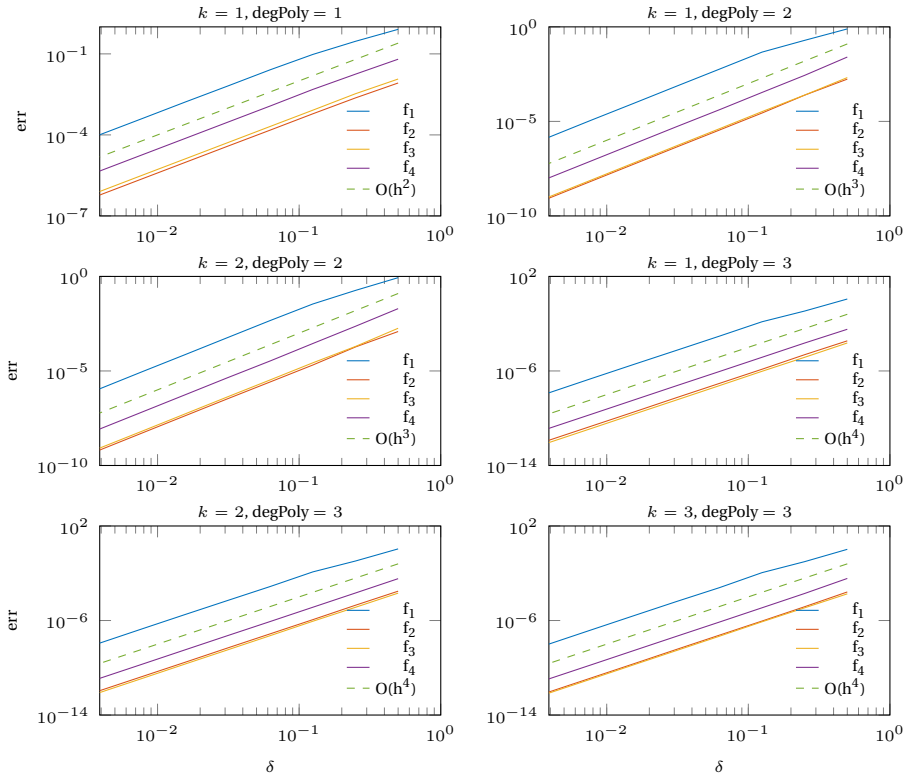


Figure 4.1 – Error for the RBF interpolation with polynomial degree $1, 2, 3$.

4.1.4 Numerical examples

In this section, we seek to verify the results in the finite volume setup (fixed number of interpolation nodes). Let $\Omega = [0,1]^2$ and $f : \Omega \to \mathbb{R}$ be a function and $\delta > 0$. We approximate f by dividing the domain into subdomains of size $\delta \times \delta$ and solve in each subdomain the interpolation problem with n nodes given from an Halton sequence [52]. Since the condition number depends on the maximal distance divided by the separation distance $Dx/\Delta x$, we use the Halton sequences with a separation distance bigger than $0.5\delta/\sqrt{n}$. We test the following functions

$$
\begin{aligned}
f_1(x,y) &= \sin(2\pi(x^2+2y^2)) - \sin(2\pi(2x^2+(y-0.5)^2)),\\
f_2(x,y) &= \exp(-(x-0.5)^2-(y-0.5)^2),\\
f_3(x,y) &= \sin(2x)+\exp(-x),\\
f_4(x,y) &= 1+\sin(4x)+\cos(3x)+\sin(2y).
\end{aligned}
$$

In the Figures 4.1 and 4.2 we show the error of the interpolation problem and confirm the correct order of convergence for the multiquadratic interpolation augmented with a polynomial of degree l of order $k \leqslant l$. For polynomial degree $l = 4$ we observe that the convergence breaks down for $\delta < 2^{-7}$. This happens at small errors $\approx 10^{-15}$ and high condition numbers $> 10^{13}$, as is shown in Table 4.1, and can be attributed to finite

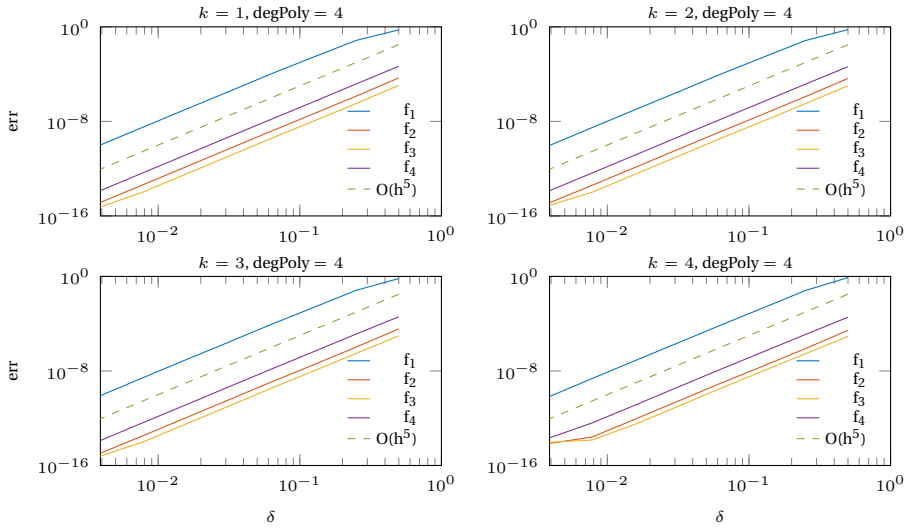


Figure 4.2 – Error for the RBF interpolation with polynomial degree 4.

precision.
Furthermore, we verify the results from Section 4.1. Table 4.1 supports the conjecture that the condition number remains constant for a fixed number of interpolation nodes n and a fixed ratio $Dx/\Delta x$.
We also observe that the condition number remains constant for the refined grids, and is considerably smaller for first order multiquadratics $k = 1$ than for the higher order ones.

deg. poly.	1	2	2	3	3	3	4	4	4	4
k	1	1	2	1	2	3	1	2	3	4
min(Cond)	$2.0e02$	$4.4e02$	$3.3e04$	$4.6e03$	$2.4e04$	$3.4e09$	$5.3e04$	$1.7e05$	$4.0e08$	$9.4e13$
max(Cond)	$2.7e02$	$7.8e02$	$1.1e05$	$8.7e03$	$7.1e04$	$6.8e09$	$4.2e05$	$2.0e06$	$1.8e09$	$9.4e14$

Table 4.1 – Comparison of maximum and minimum condition numbers for different polynomial degrees and different orders of MQs k.

4.2 RBF-ENO method

In this section, we introduce a new RBF-ENO method of order p on two-dimensional unstructured grids that can be generalized to higher dimensions. The method is based on the MUSCL approach described in Section 2.2, the RBF-ENO reconstruction introduced in Chapter 4, and the evaluation technique discussed in Section 4.1.
The finite volume method relies on the high-order flux (2.47) based on the boundary integral of the Rusanov flux (2.44) which is approximated by the Gauss-Legendre quadrature [4]. For the evaluation of the high-order flux we use the RBF reconstruction (2.113) and for the computation of the cell average we use a cubature rule for triangles [27]. The ENO reconstruction (Algorithm 4.1) is a generalization of the one-

dimensional entropy stable Algorithm 3.1 which is based on the one introduced by Harten et al. [57]. Thus, we recursively add one cell to the stencil S_i and all its neighbors to a list of possible choices N_i for the next step. In each step, we add the cell in N_i which results in the stencil that has the smallest smoothness indicator IS, indicating the smoothness of the solution on a stencil. It is well-known that this strategy comes with high costs, but is also very robust. As the smoothness indicator we choose a

Algorithm 4.1 Recursive RBF stencil selection algorithm for multiple dimensions

Let the interpolation cells $S_i = \{C_{i_1}, \dots, C_{i_k}\}$ and its mean-values $U_{i_1}, \dots, U_{i_k}$ be given.
Let $N_i = \{C_{j_0}, \dots, C_{j_l}\}$ be the direct neighbors for all $C \in S_i$ such that $N_i \cap S_i = \emptyset$.
Start by initializing $S_i := \{C_i\}$ and $N_i := \{C |\ C \text{ is neighbor of } C_i\}$.
for $j = 0, \dots, n-2$ **do**
  Set $S_{j_s} := S_i \cup \{C_{j_s}\}$ for all $s = 1, \dots, l$ and $C_{j_s} \in N_i$.
  $r := \operatorname{argmin}_s \mathrm{IS}_{RBF}(S_{j_s})$
  $S_i := S_i \cup \{C_{j_r}\}$
  $N_i := N_i \cup \{C \notin S_i |\ C \text{ is neighbor of } C_{j_r} \text{ and } d(C) \leqslant d_{max}\} \backslash \{C_{j_r}\}$
end for

generalization of the one-dimensional indicator (3.24)

$$\mathrm{IS}_{RBF}(s) := \sum_{i=1}^{n} a_i^2, \tag{4.39}$$

for the reconstruction $s(\mathbf{x}) = \sum_{i=1}^{n} a_i \lambda_{C_i}^{\xi} \phi(\mathbf{x} - \xi) + \sum_{j=1}^{m} b_j p_j(\mathbf{x})$.
It is important to choose the right degree of the polynomial for each stencil. For a polynomial of degree l we need at least $n = \frac{(l+2)(l+1)}{2}$ cells, and thus $l = -1.5 + \frac{1}{2}\sqrt{1+8n}$. To reduce the probability of $\{\lambda_{C_{i_j}}\}_{j=1}^{n}$ having no $\Pi_l(\mathbb{R}^d)$-unisolvent subset, we choose

$$l = \begin{cases} \left\lfloor -2.5 + \frac{1}{2}\sqrt{1+8(n-1)} \right\rfloor, & n \geqslant 5, \\ 0 & n < 5. \end{cases} \tag{4.40}$$

Furthermore, we use multiquadratics with a shape parameter based on (4.1) and the polynomials (4.2). Since the order of convergence is not influenced by the order of the multiquadratics and following the observations in Section 4.1.4, we choose first order multiquadratics.
We need to slightly adapt the evaluation method from Section 4.1 to use it for the RBF-ENO method. The coefficients a_i depend on the shape parameter. Thus, we must compare the smoothness indicator (4.39) with respect to the same shape parameter. By assuming approximately uniform equilateral triangles, we approximate Δx as

$$\Delta x \approx \min_j 2r_{j,inscr} \approx 2r_{i,inscr} \approx \sqrt{|C_i|}, \tag{4.41}$$

with the radius $r_{j,inscr}$ of the inscribed circle of the jth cell and $|C_i|$ is the area of the ith cell. The last estimate comes from

$$|C_i| = 3\sqrt{3}r_{i,inscr}^2 \approx 4r_{i,inscr}^2, \tag{4.42}$$

where we assume C_i to be an equilateral triangle. Hence, we choose the shape parameter as

$$\varepsilon = \frac{1}{\sqrt{|C_i|}}, \tag{4.43}$$

with the polynomial basis (4.2).
The advantage of RBFs over polynomials is the ability to deal with a stencil with a variable number of elements. The condition for RBFs to have a well-defined system of equations is the existence of a subset which is $\Pi_l(\mathbb{R}^d)$-unisolvent and l must be larger than the order of the RBF. Thus, we can use a bigger stencil than the dimension of $\Pi_l(\mathbb{R}^d)$ to circumvent cell constellations that are ill-conditioned. To keep the stencil compact we classify each cell around the central one, depending on its distance $d \in \mathbb{N}$, such that

$$\begin{aligned}
&d(C) = 0, \text{ if } C = C_i,\\
&d(C) = 1, \text{ if } C \text{ is a direct neighbor of } C_i,\\
&d(C) = 2, \text{ if } C \text{ has a neighbor } \tilde{C} \text{ with } d(\tilde{C}) = 1,\\
&\qquad \dots
\end{aligned}$$

and introduce d_{max} as the maximal distance. A stable configuration for the RBF-ENO method of order p is given in Table 4.2 with $l = p - 1$.
Note that (4.40) does not coincide with the values from Table 4.2. However, from numerical experiments this combination seems superior.

deg. poly. l	1	2	3
n	5	12	30
d_{max}	3	5	8

Table 4.2 – Stencil setting depending on the polynomial degree l.

Summary of the RBF-ENO method

- Finite volume method with a high-order flux (2.47);
- The Gauss-Legendre quadrature [4] to approximate the boundary integral of the Rusanov flux (2.44);

- Reconstruction based on the RBF approach (2.113) with the polynomial basis (4.2);
- First order multiquadratics with shape parameter (4.43);
- Size n of stencil and d_{max} from Table 4.2 depending on the order of the method;
- Stencil selection: Algorithm 4.1 and smoothness indicator (4.39) with polynomial degree (4.40).

4.2.1 Reconstruction at the boundary

In contrast to the one-dimensional version we omit the use of ghost cells. However, we need to be aware of the reduced flexibility of the stencil choice in this case. It has been shown that it is enough to use a method of order $p-1$ at the boundary to maintain the global formal accuracy [51]. Thus, to circumvent stability issues we use a polynomial of degree $l-1$ for cells at the boundary.

4.3 Numerical results

In this section, we demonstrate the robustness of the second and third order RBF-ENO method on general grids. For the time discretization we use a third order SSPRK method [49]. The grids are generated by $distmesh2d()$, which is based on the Delaunay algorithm [99].

4.3.1 Linear advection equation

We consider the linear advection equation in two dimensions

$$u_t + au_x + bu_y = 0, \tag{4.44}$$

with wave speed $a = 1$, $b = 0$ and periodic boundary conditions [83]. This results in a right moving wave given by the initial condition

$$u_0(x,y) = \cos(2\pi x)\cos(2\pi y) + 10. \tag{4.45}$$

Fig. 4.3 shows the error at $T = 0.1$. We observe a drop of the order of convergence after a certain level of refinement which is a known phenomena [116, 61]. This arises from constantly switching the stencil. For a very smooth function we recover the right order of convergence by multiplying the smoothness indicator with a penalty term D^3 which depends on the distance to the central cell

$$D := \frac{1}{|C_i|}\sum_{j\in S_i}\|x_{c,j} - x_{c,i}\|^2, \tag{4.46}$$

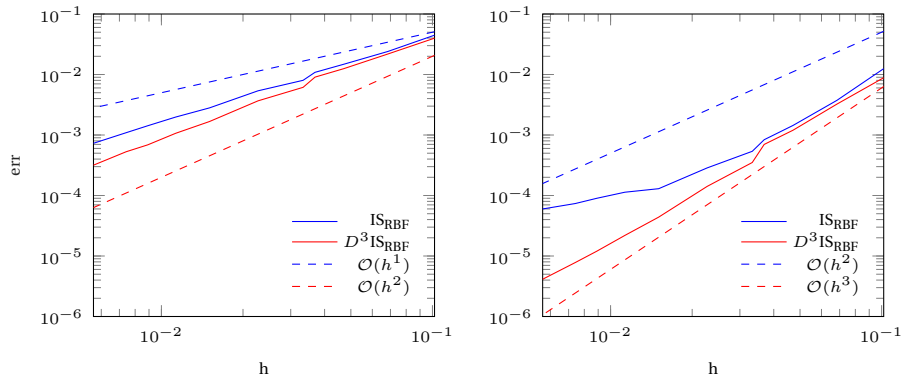


Figure 4.3 – Error for the RBF-ENO method for the linear advection equation in 2D (left 2nd order, right 3rd order).

with the center $x_{c,i}$ of cell C_i. This gives preference to the central stencil.

4.3.2 Burgers' equation

Next, we consider the two dimensional Burgers' equation

$$u_t + \frac{1}{2}(u^2)_x + \frac{1}{2}(u^2)_y = 0, \tag{4.47}$$

on the domain $\Omega = [0,1]^2$ with the initial conditions

$$u_0 = \begin{cases} -1 & \text{if } x > 0.5,\ y > 0.5, \\ -0.2 & \text{if } x < 0.5,\ y > 0.5, \\ 0.5 & \text{if } x < 0.5,\ y < 0.5, \\ 0.8 & \text{if } x > 0.5,\ y < 0.5. \end{cases} \tag{4.48}$$

The Burgers' equation illustrates the behavior of the scheme with a nonlinear flux and its ability to deal with discontinuities. Furthermore, the results can be compared with the exact solution [50]. The solution consists of shocks and rarefaction waves as its one-dimensional counterpart. To avoid boundary effects we increase the computational domain to $\Omega = [-1,2]^2$ and keep the initial conditions for the extended square, see Fig. 4.4. The solutions at time $T = 0.25$ for the 3rd and 4th order method are as expected, Fig. 4.5. There are some minor oscillations, but they remain small.

4.3.3 KPP rotating wave

We consider the two-dimensional KPP rotating wave problem

$$u_t + (\sin(u))_x + (\cos(u))_y = 0, \tag{4.49}$$

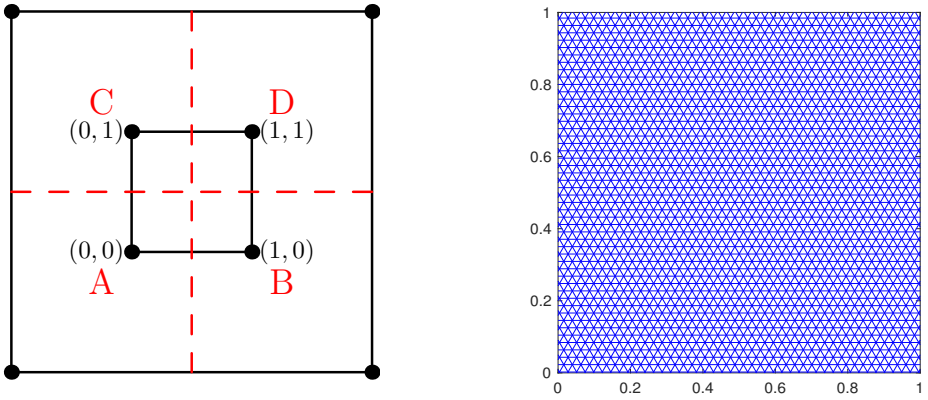
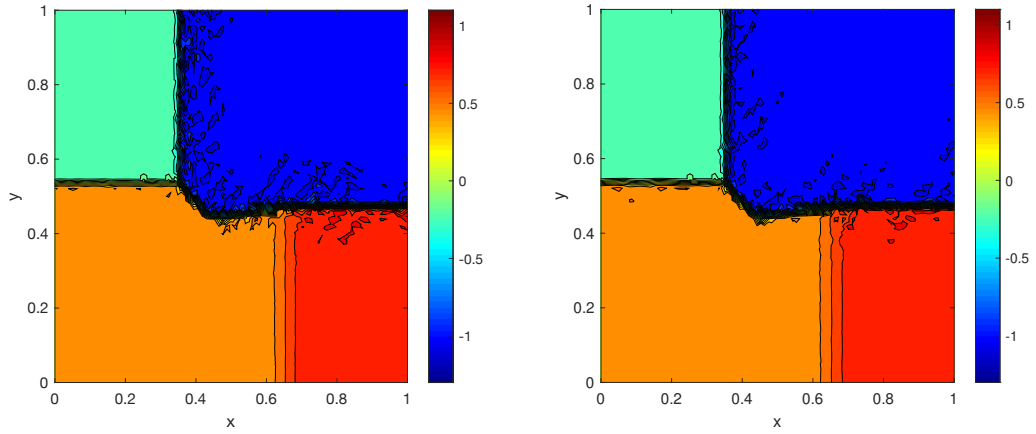


Figure 4.4 – Domain extension, initial value composition and grid.



(a) RBF-ENO of order 3.

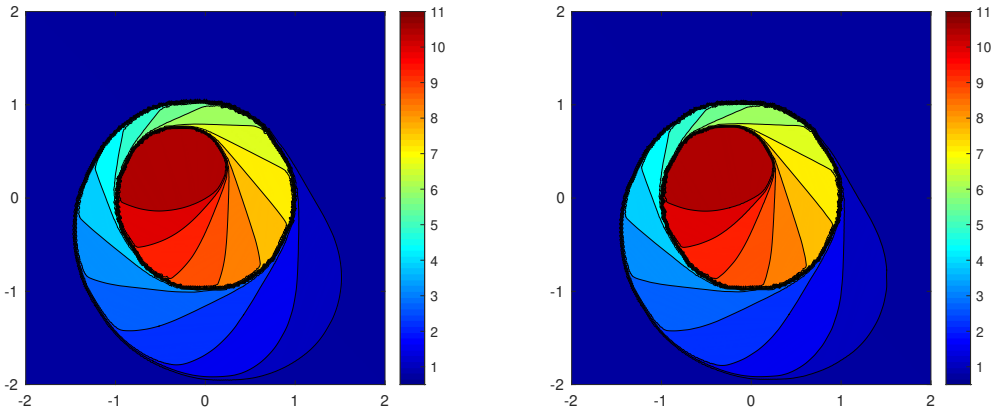
(b) RBF-ENO of order 4.

Figure 4.5 – Solution of Burgers' equation at $T = 0.25$ with $N = 37444$ cells, $\mathrm{CFL} = 0.5$.

in the domain $\Omega = [-2, 2]^2$ with periodic boundary conditions and the initial conditions

$$u_0 = \begin{cases} 3.5\pi & \text{if } x^2 + y^2 \leqslant 1, \\ 0.25\pi & \text{otherwise .} \end{cases} \tag{4.50}$$

This is a challenging non-convex scalar conservation law [76]. The KPP problem was designed to test various schemes for entropy violating solutions. At time $T = 1$ the solution forms a characteristic spiral, which is well-resolved for the second and third order method, as shown in Fig. 4.6.



(a) RBF-ENO of order 2.

(b) RBF-ENO of order 3.

Figure 4.6 – KPP problem at $T = 1$ with $N = 58646$ cells, CFL $= 0.5$.

4.3.4 Euler equations

Let us consider the two-dimensional Euler equations (2.9) with $\gamma = 1.4$ which reflects a diatomic gas such as air.

Isentropic vortex

The isentropic vortex problem describes the evolution of a inviscid isentropic vortex in a free stream on the domain $\Omega = [-5, 5]^2$. Proposed by Yee et al. [128] it is one of the few exact solutions for the Euler equations. The initial conditions are

$$\begin{aligned} \rho &= \left[1 - \frac{\beta^2(\gamma-1)}{8\gamma\pi^2}\exp(1-r^2)\right]^{\frac{1}{(\gamma-1)}},\ u_1 = M\cos(\alpha) - \frac{\beta(y-y_c)}{2\pi}\exp(\frac{1-r^2}{2}), \\ u_2 &= M\sin(\alpha) - \frac{\beta(x-x_c)}{2\pi}\exp(\frac{1-r^2}{2}),\ r = \sqrt{(x-x_c)^2+(y-y_c)^2}, \end{aligned} \tag{4.51}$$

with the initial vortex strength β, the initial vortex center (x_c, y_c) and periodic boundary conditions. The pressure is initialized by $p = \rho^\gamma$ and α prescribes the passive advection direction. The exact solution is the initial condition propagating with speed M in direction $(\cos(\alpha), \sin(\alpha))$. The parameters are chosen as $M = 0.5$, $\alpha = 0$, $\beta = 5$ and $(x_c, y_c) = (0, 0)$. We consider the order of convergence at time $T = 1$. In Fig. 4.7 we observe the same behavior as for the linear advection equation. Again, we overcome this stability issue by introducing a penalty term D^3 which depends on the distance of the cell to its central one (4.46), and recover the optimal order of convergence.

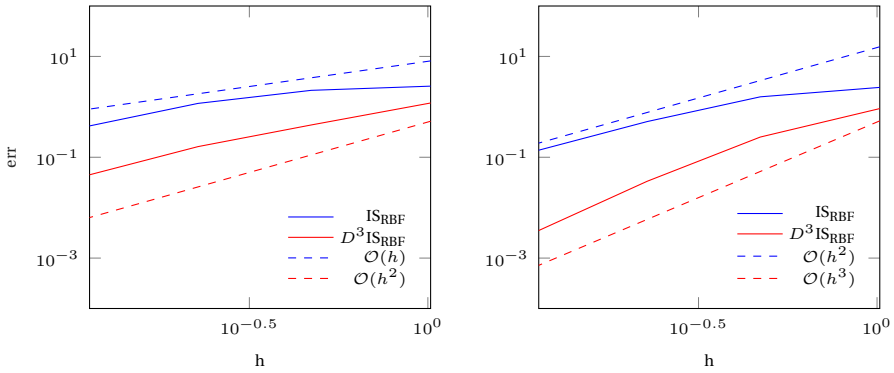


Figure 4.7 – Error for the RBF-ENO method for the isentropic vortex problem for 2nd order (left) and 3rd order (right).

Riemann problem

The initial values for Riemann problems in two dimensions are constant in each quadrant

$$u_0 = \begin{cases} (\rho_A, m_{1,A}, m_{2,A}, E_A) & \text{if } x < 0.5,\ y < 0.5, \\ (\rho_B, m_{1,B}, m_{2,B}, E_B) & \text{if } x > 0.5,\ y < 0.5, \\ (\rho_C, m_{1,C}, m_{2,C}, E_C) & \text{if } x < 0.5,\ y > 0.5, \\ (\rho_D, m_{1,D}, m_{2,D}, E_D) & \text{if } x > 0.5,\ y > 0.5, \end{cases} \tag{4.52}$$

with the physical domain $\Omega = [0,1]^2$, which is enlarged to $\Omega = [-1,2]^2$ to reduce boundary effects. The values are chosen in such a way that only a single elementary wave appears at each interface. This results in 19 genuinely different configuration for a polytropic gas [80]. We test two of them, see Table 4.3.
We solve the Riemann problems until time $T = 0.25$ on the grid shown in Fig. 4.8. In Fig. 4.9 we show that the results of the 4th configuration are well resolved with the 2nd and 3rd order methods, while keeping the oscillations small. Furthermore, Fig. 4.10 illustrates the convergence in h for the RBF-ENO method of order 3.

	Riemann Problem 4				Riemann Problem 12			
	ρ	u_1	u_2	p	ρ	u_1	u_2	p
A	1.1	0.8939	0.8939	1.1	0.8	0	0	1
B	0.5065	0	0.8939	0.35	1	0	00.7276	1
C	0.5065	0.8939	0	0.35	1	0.7276	0	1
D	1.1	0	0	1.1	0.5313	0	0	0.4

Table 4.3 – Initial values of the Riemann problem.

For the Riemann problem 12 at time $T = 0.25$ the results are of a similar quality, see Fig. 4.11 and 4.12.

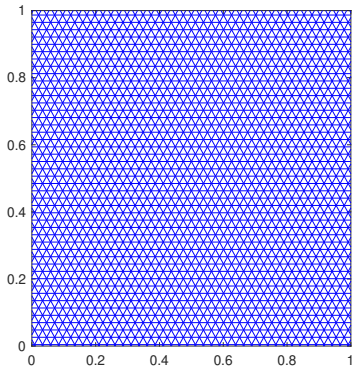
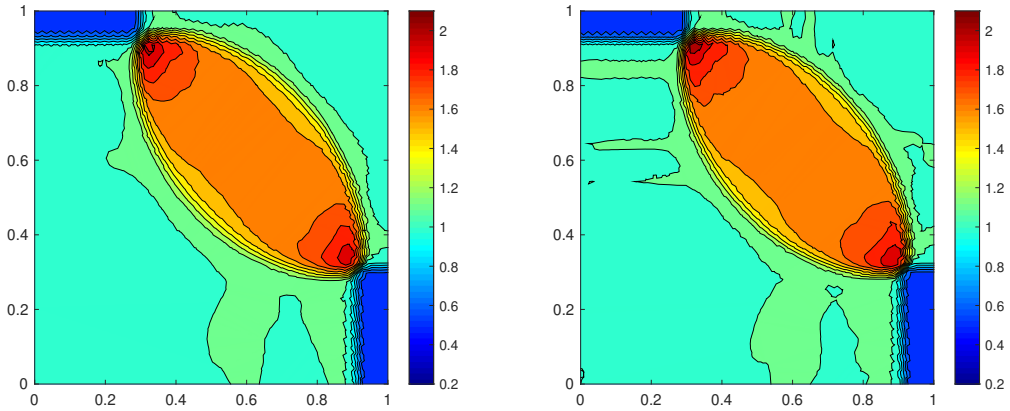


Figure 4.8 – Grid for the Riemann problems.



(a) RBF-ENO of order 2.

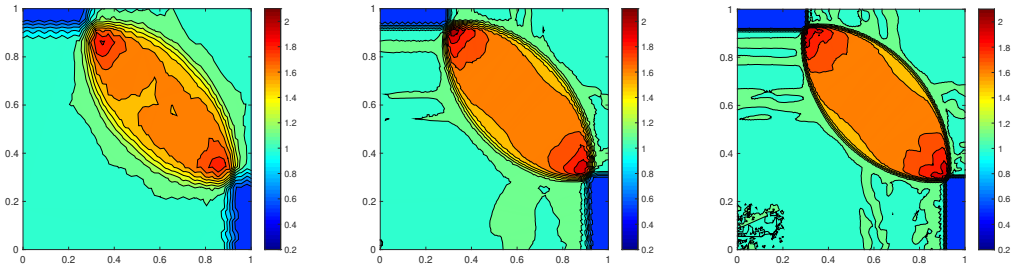
(b) RBF-ENO of order 3.

Figure 4.9 – Riemann problem 4 at $T = 0.25$ with $N = 32946$ cells in the extended domain, $\mathrm{CFL} = 0.5$ and 20 contour lines between 0.2 and 2.1.

Shock vortex interaction problem

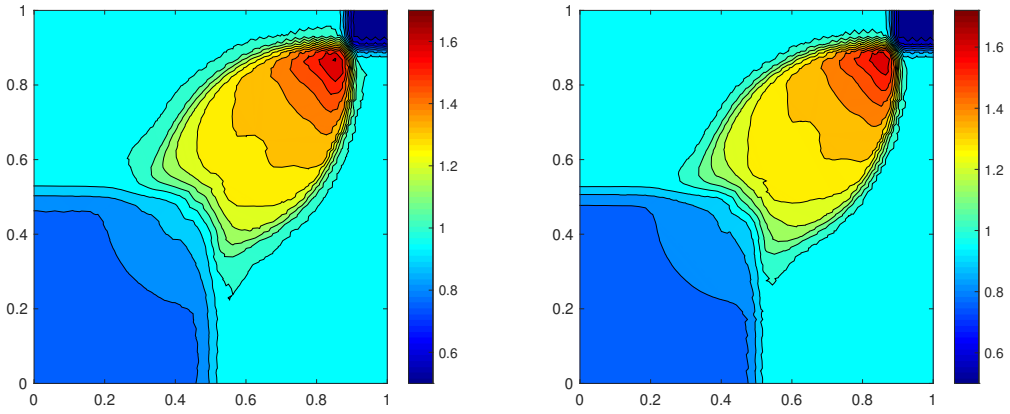
The shock vortex interaction problem was introduced to test high order methods [113]. It describes the interaction of a right-moving vortex with a left-moving shock in the domain $\Omega = [0,1]^2$. The initial condition is given by the shock discontinuity

$$(\rho, m_1, m_2, E) = \begin{cases} (\rho_L, m_{1,L}, m_{2,L}, E_L) & \text{if } x < 0.5, \\ (\rho_R, m_{1,R}, m_{2,R}, E_R) & \text{if } x \geqslant 0.5, \end{cases} \tag{4.53}$$



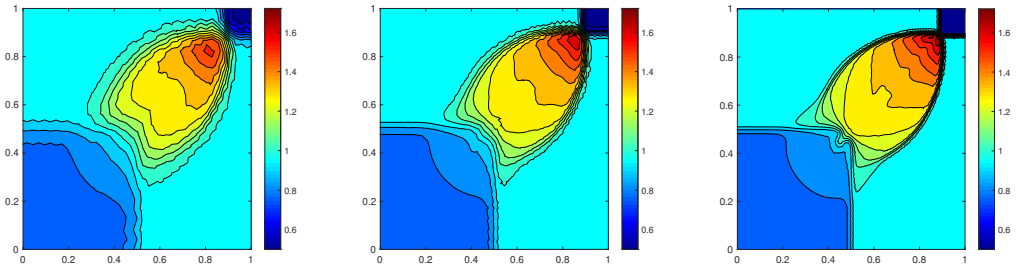
(a) RBF-ENO3, $N = 8192$. (b) RBF-ENO3, $N = 32946$. (c) RBF-ENO3, $N = 150676$.

Figure 4.10 – Convergence in h of the Riemann problem 4 at $T = 0.25$ with $\mathrm{CFL} = 0.5$ and 20 contour lines between 0.2 and 2.1.



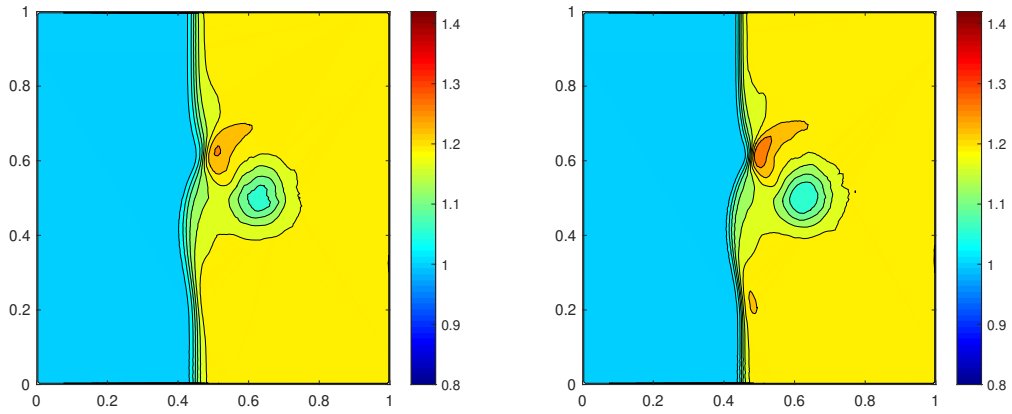
(a) RBF-ENO of order 2. (b) RBF-ENO of order 3.

Figure 4.11 – Riemann problem 12 at $T = 0.25$ with $N = 32946$ cells in the extended domain, $\mathrm{CFL} = 0.5$ and 20 contour lines between 0.5 and 1.7.



(a) RBF-ENO3, $N = 8192$. (b) RBF-ENO3, $N = 32946$. (c) RBF-ENO3, $N = 150676$.

Figure 4.12 – Convergence in h of the Riemann problem 12 at $T = 0.25$ with $\mathrm{CFL} = 0.5$ and 20 contour lines between 0.5 and 1.7.



(a) RBF-ENO of order 2.

(b) RBF-ENO of order 3.

Figure 4.13 – Shock vortex interaction problem at $T = 0.35$ with $N = 14616$ cells, $\mathrm{CFL} = 0.5$ with 20 contour lines in $[0.8, 1.42]$.

with the left state superposed by the perturbation

$$\begin{aligned} \delta u_1 &= \epsilon \frac{y - y_c}{r_c} \exp(\beta(1 - r^2)), \qquad & \delta u_2 &= -\epsilon \frac{x - x_c}{r_c} \exp(\beta(1 - r^2)), \\ \delta\theta &= -\frac{\gamma - 1}{4\beta\gamma} \epsilon^2 \exp(2\beta(1 - r^2)), & \delta s &= 0, \end{aligned} \tag{4.54}$$

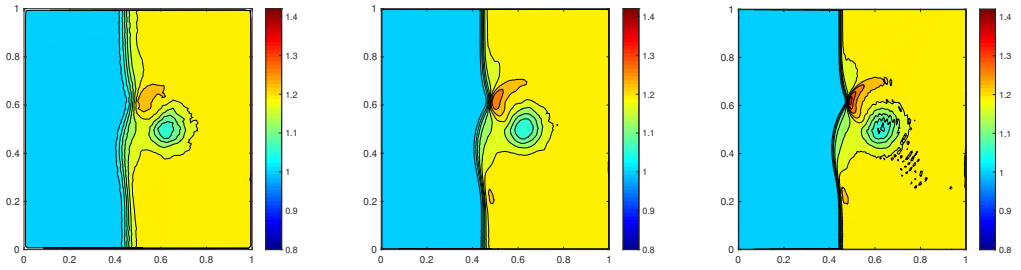
with the temperature $\theta = p/\rho$, the physical entropy $s = \log p - \gamma \log \rho$ and the distance $r^2 = ((x - x_c)^2 + (y - y_c)^2)/r_c^2$. The left state is given by

$$(\rho_L, u_{1,L}, u_{2,L}, E_L) = (1, \sqrt{\gamma}, 0, 1), \tag{4.55}$$

and the right state by

$$\begin{aligned} p_R &= 1.3, & \rho_R &= \rho_L \Big(\frac{\gamma - 1 + (\gamma + 1) p_R}{\gamma + 1 + (\gamma - 1) p_R} \Big), \\ u_{1,R} &= \sqrt{\gamma} + \sqrt{2} \Big(\frac{1 - p_R}{\sqrt{\gamma - 1 + p_R(\gamma + 1)}} \Big), & u_{2,R} &= 0. \end{aligned} \tag{4.56}$$

The parameter of the vortex are chosen as $\epsilon = 0.3$, $r_c = 0.05$, $\beta = 0.204$ with the initial center of the vortex $(x_c, y_c) = (0.25, 0.5)$. Figure 4.13 shows the result of the second and third order RBF-ENO method at the final time $T = 0.35$ for $N = 14616$ cells. The higher resolution of the third order method is clear. In Fig. 4.14 we see the convergence of the scheme for increasing number of cells. We observe minor oscillations for $N = 58646$, but they remain stable.



(a) RBF-ENO3, $N = 3552$. (b) RBF-ENO3, $N = 14616$. (c) RBF-ENO3, $N = 58646$.

Figure 4.14 – Convergence in h of the shock vortex interaction problem at $T = 0.35$ with $\mathrm{CFL} = 0.5$ with 20 contour lines in $[0.8, 1.42]$.

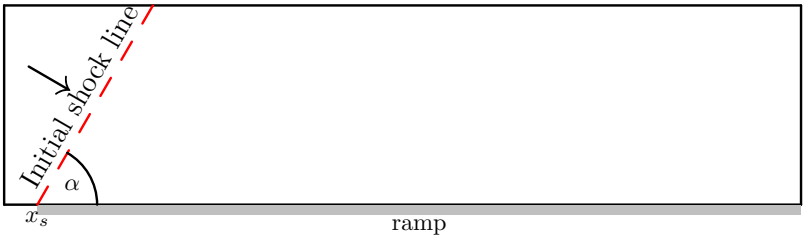


Figure 4.15 – Domain for the double Mach reflection problem.

Double Mach reflection problem

The double Mach reflection problem is a standard benchmark for Euler codes that tests its robustness in the presence of a strong shock. It was introduced by Woodward and Colella [126] and consists of a Mach 10 shock propagating at an angle of 30° ($\alpha = 60^\circ$) into the ramp, see Fig. 4.15. The domain $\Omega = [0,4] \times [0,1]$ contains a ramp starting at $x_s = 1/6$. As boundary conditions we have on the left side and on the ground in front of the ramp inflow boundary conditions with the post-shock values. On the ramp we use slip-wall conditions, on the top we apply the exact time dependent shock location and on the right we use outflow boundary conditions with the pre-shock conditions. The solution is simulated until $T = 0.2$ with the initial condition

$$(\rho, m_1, m_2, E) = \begin{cases} (8.0, 57.1597, -33.0012, 563.544) & \text{post-shock,} \\ (1.4, 0, 0, 2.5) & \text{pre-shock.} \end{cases} \tag{4.57}$$

To solve the double Mach reflection problem we must choose the multiquadratics of order l for a method of order l to get a stable solution, shown in Figure 4.16. This suggests that the proposed stencil selection algorithm from Chapter 3 is more stable than just using a first order RBF in the same algorithm. To highlight the ability to deal with fully unstructured grids, we present a solution with around a quarter of the cells refined in the lower part of the domain, Figure 4.18. The solution is based on a grid of

the form of Figure 4.17 with approximately six times more cells at each face. Note that the cells in the lower part have approximately the same size as the ones in the example in Figure 4.16.

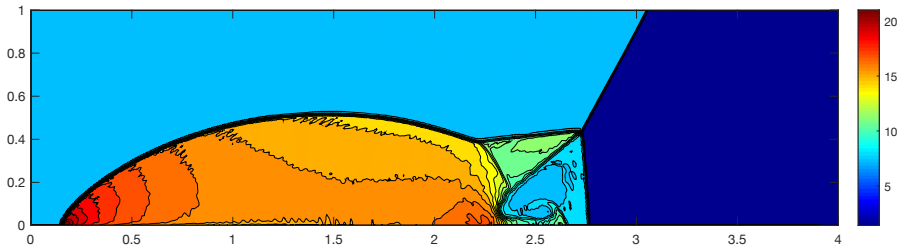


Figure 4.16 – Double Mach reflection problem at $T = 0.2$ with $N = 151216$ cells, $\mathrm{CFL} = 0.5$ solved with the RBF-ENO of order 3 and displayed with 26 contour lines in $[1.5, 21]$.

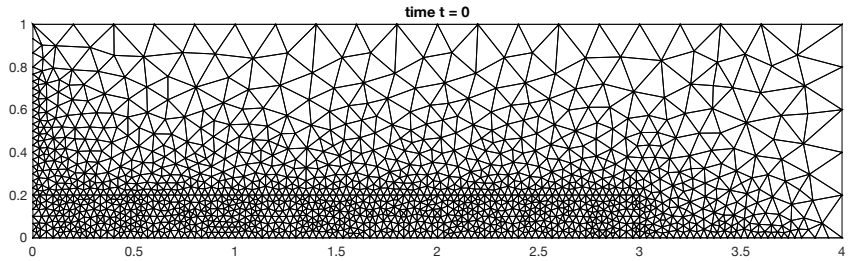


Figure 4.17 – Example of totally unstructured grid with $N = 2843$ cells.

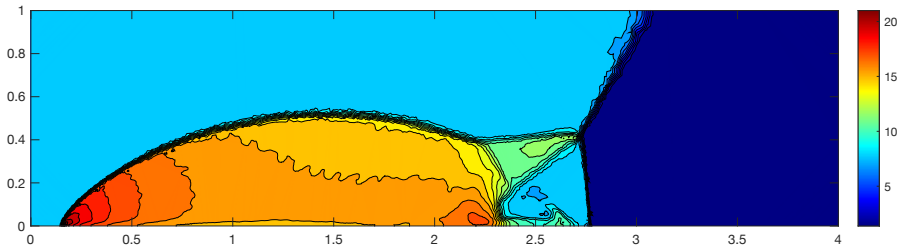


Figure 4.18 – Double Mach reflection problem at $T = 0.2$ with $N = 41140$ cells, $\mathrm{CFL} = 0.5$ solved with the RBF-ENO of order 3 on totally unstructured grid and displayed with 26 contour lines in $[1.5, 21]$.

5 RBF-based CWENO method

In the previous chapter, we introduced a new two dimensional ENO-method based on the RBF reconstruction. It is stable and flexible for general geometries. However, the choice of the stencil, based on Algorithm 4.1, is known to be expensive. Different stencil selection algorithms have been proposed to use a given number of stencils [67, 44, 73, 26]. A different way to deal with this problem is the Central WENO method (CWENO) [84] or the multi-resolution WENO method [137]. In this chapter, we introduce a CWENO method based on RBF reconstruction. The one-dimensional results are published in [64]. The two-dimensional generalization is based on a combination of the CWENO and the multi-resolution WENO method with the RBF reconstruction.

5.1 CWENO method

The CWENO method was introduced by Levy et al. [84] as a third order method. It is based on the WENO method from Section 2.2.2. Further analysis and generalizations to higher orders on unstructured grids in one dimension can be found in [20, 21].
Let us consider the one-dimensional system (2.1) with a grid $\{x_i\}_{i\in\mathbb{Z}}$ and the semi-discrete formulation (1.7) with a monotone flux function $F(\mathbf{U},\mathbf{V})$. As before, the goal is to construct for each cell $C_i = (x_{i-1/2}, x_{i+1/2}]$ a reconstruction s_i based on the stencil $S_i = \{C_{i-n+1},\dots,C_{i+n-1}\}$ for a fixed $n \in \mathbb{N}$. Unlike the ENO method we aim to make use of the whole stencil in smooth regions and the algorithm should choose a polynomial of degree $2n-2$ such that

$$\lambda_C s_i = \mathbf{U}_C, \qquad \text{for all } C \in S_i. \tag{5.1}$$

In the case of a non-smooth solution it chooses a polynomial of degree $n-1$ on the stencil $S_i^j = \{C_{i-n+j},\dots,C_{i+j-1}\}$ for one $1 \leqslant j \leqslant n$ that avoids the discontinuity. Given the reconstruction, the high-order numerical flux is

$$F_{i+1/2} = F(s_{i+1}(x_{i+1/2}), s_i(x_{i+1/2})). \tag{5.2}$$

The difference from the WENO method is found in the construction of the high-order interpolation function. Specifically, let us consider s_{opt} as the polynomial of degree $2n-2$ that interpolates all data of the stencil S_i and the polynomials s_i^j of degree $n-1$ that interpolate the data on the stencil S_i^j for $j=1,\dots,n$. Furthermore, the reconstruction depends on the choice of the positive real coefficients $d_0,\dots,d_n \in [0,1]$ such that $\sum_{j=0}^{n} d_j = 1$, $d_0 \neq 0$. Then, the reconstruction polynomial of degree $2n-2$ is

$$s_i(x) = \sum_{j=0}^{n} \omega_j s_i^j(x), \tag{5.3}$$

with

$$s_i^0(x) = \frac{1}{d_0}\Big(s_{\text{opt}}(x) - \sum_{j=1}^{n} d_j s_i^j(x)\Big), \tag{5.4}$$

and the nonlinear coefficients ω_j that are defined as

$$\omega_j = \frac{\alpha_j}{\sum_{i=0}^{n}\alpha_i}, \qquad \alpha_j = \frac{d_j}{(\text{IS}_{C_i}[s_i^j] + \bar{\varepsilon})^t}, \tag{5.5}$$

where $\text{IS}_{C_i}[s_i^j]$ indicates the smoothness of s_i^j in the cell C_i, $1 \gg \bar{\epsilon} > 0$ and $t \geqslant 1$. As with the WENO method, a classical indicator of smoothness in a cell C for a polynomial is the Jiang-Shu indicator [70]

$$\text{IS}_C[s] = \sum_{j>0} |C|^{2j-1} \int_C \Big(\frac{\mathrm{d}^j}{\mathrm{d}x^j} s(x)\Big)^2 \mathrm{d}x. \tag{5.6}$$

The choice of $\bar{\epsilon}$ is of importance, because if it is too small, it might affect the order of convergence. On the other hand if it is too big, spurious oscillations may occur. Cravero et al. [20] show that the choice $\bar{\epsilon} = \hat{\epsilon} h^p$ for $p = 1, 2$ leads to the optimal order of convergence. As proposed in [20] we define the coefficients d_j over the temporary weights

$$\hat{d}_j = \hat{d}_{n+1-j} = j, \qquad 1 \leqslant j \leqslant \frac{n+1}{2}, \tag{5.7}$$

and we choose $d_0 \in (0,1)$ for the high-order polynomial. Thus, one possible choice for the coefficients is

$$d_j = \frac{\hat{d}_j}{\sum_{i>0}\hat{d}_i}(1-d_0). \tag{5.8}$$

The main difference with respect to the classical WENO method is that for the smooth case we are not constructing s_{opt} out of the polynomials s_i^j, but we build it independently by resolving an additional system of equations. This method has the advantage

that it is easier to generalize on general grids in high dimensions, while maintaining high-order accuracy.

5.2 One-dimensional RBF-CWENO method

Methods combining RBFs and weighted essentially nonoscillatory methods have been proposed, e.g. [2, 3, 11]. The advantage of the CWENO method over the WENO method is its flexibility on general grids and its independence of the construction of a high-order interpolation function out of lower order ones. This facilitates the use of the whole grid in smooth regions and is important for non-polynomial interpolation functions which cannot be combined to a higher order function.
We propose the RBF-CWENO method which works as the classical CWENO method with the reconstruction function (5.3) and the weights (5.5), but as interpolation function we use the RBF reconstruction (2.113) instead of polynomials. For a fixed $k \in \mathbb{N}$ we choose the multiquadratic RBFs of order k with polynomials of degree $k-1$ and a stencil $S_i = \{C_{i-k}, \dots, C_{i+k}\}$. Since the problem of the ill-conditioning can be solved by using the vector-valued rational approximation method from Section 2.6.3, the main challenge for the RBF method is the choice of the smoothness indicator. For polyharmonic splines, Aboyar et al. [2] use the semi-norm of the Beppo-Levi space and Bigoni and Hesthaven [11] use a modified version of the Jiang-Shu indicator (5.6). Further, we introduced the entropy stable smoothness indicator (3.24) for the RBF-ENO method. However, for stencils with different sizes this indicator does not work.

5.2.1 Smoothness indicator

The smoothness indicator is the heart of essentially nonoscillatory methods. For the RBF-CWENO method we consider one similar to the one introduced by Bigoni and Hesthaven [11]

$$\begin{aligned} \mathrm{IS}_C[s] = &\sum_{j=1}^{n} |C|^{2j-1} \int_C \Big(\frac{\partial^j p(x)}{\partial x^j}\Big)^2 \mathrm{d}x \\ &+ |C|^{2n-1} \int_C \left(\frac{\partial^n}{\partial x^n}\Big[\sum_{i=1}^{n} a_i \lambda^{\xi}_{C_i} \phi(\|x-\xi\|)\Big]\right)^2 \mathrm{d}x, \end{aligned} \tag{5.9}$$

with n being the number of cells we are interpolating on. The first part of (5.9) is the sum of the derivatives of the polynomial part and the second term expresses the highest derivative of the RBF-part. The original Jiang-Shu indicator applied to (2.113) would include the lower derivatives of the RBF-part plus all mixed terms, but we find this to be less efficient. For simplicity the integrals can be approximated with a simple mid-point rule.

We face again the problem of ill-conditioning when recovering the coefficients a_i. Numerical examples indicate that a small shape parameter improve the accuracy, but they do not affect the choice of the stencil using the smoothness indicator (5.9). Thus, we use the stable shape parameter ε_R from (2.150) which ensures the solvability of the system of equations.

5.3 Numerical results for the 1D RBF-CWENO method

We now discuss the numerical results of the RBF-CWENO method and compare it with the RBF-WENO method [11] and the classical ENO method described in Section 2.2.2. All methods use the Lax-Friedrichs numerical flux and integration in time is done using the SSPRK-5 method [61] with time step $\Delta t = \mathrm{CFL}\,\Delta x/\lambda_{max}$ and the maximal eigenvalue λ_{max} of $\nabla_u f$. Furthermore, we use the vector-valued rational approximation approach, introduced in Section 2.6.3, to circumvent ill-conditioning of the interpolation matrix and a shape parameter $\varepsilon = 0.1$. For the nonlinear weights (5.5) we choose $\bar{\epsilon} = \hat{\epsilon}h^2$ with $\hat{\epsilon} = 0.1$.

5.3.1 Linear advection equation

Let us consider the linear advection equation (2.3) on the domain $[0,1]$ with wave speed $a = 1$, initial condition

$$u_0(x) = \sin(2\pi x), \tag{5.10}$$

and periodic boundary conditions [83]. Note that for $k = 3$ we expect the order of convergence to be 7, therefore we use the reduced time step $\Delta t = \mathrm{CFL}\cdot\Delta x^{7/5}/\lambda_{max}$ to recover the right order of convergence. The correct order of convergence of the RBF-CWENO method is almost achieved in Table 5.1 and it seems to be more accurate than the RBF-WENO method.

5.3.2 Burgers' equation

Considering the Burgers' equation (2.4) on the domain $[0,1]$ with the initial condition

$$u_0(x) = \sin(2\pi x), \tag{5.11}$$

we analyze its robustness with respect to discontinuities. In Figure 5.1, we report the results performed with $\mathrm{CFL} = 0.5$ at $T = 0.3$. We observe no oscillations around the discontinuity at $x = 0.5$ and as expected an increasing accuracy for increasing number of elements.

k	N	RBF-CWENO L_h^1 error	rate	L_h^2 error	rate	L_h^∞ error	rate	RBF-WENO L_h^2 error	rate
1	16	$5.6409e-04$	–	$2.1702e-04$	–	$1.5903e-04$	–	$1.5754e-02$	–
	32	$7.6612e-05$	2.75	$2.4817e-05$	2.99	$1.6221e-05$	3.15	$4.8924e-03$	1.69
	64	$1.0082e-05$	2.79	$2.5297e-06$	3.15	$1.3561e-06$	3.42	$1.2608e-03$	1.96
	128	$1.3812e-06$	2.74	$2.4032e-07$	3.24	$9.6982e-08$	3.63	$9.2931e-05$	3.76
	256	$2.1322e-07$	2.57	$2.3289e-08$	3.21	$6.5703e-09$	3.71	$2.3008e-06$	5.34
2	16	$2.3796e-05$	–	$7.3671e-06$	–	$4.1241e-06$	–	$5.4401e-04$	–
	32	$3.5783e-06$	2.61	$8.3093e-07$	3.01	$3.9675e-07$	3.22	$4.4938e-05$	3.60
	64	$2.8691e-07$	3.48	$5.9366e-08$	3.63	$3.6940e-08$	3.27	$3.4787e-06$	3.69
	128	$1.4563e-08$	4.11	$2.5775e-09$	4.32	$1.3965e-09$	4.51	$2.5956e-07$	3.74
	256	$6.8835e-10$	4.20	$9.6168e-11$	4.53	$4.4249e-11$	4.75	$1.9221e-08$	3.76
3	16	$3.8815e-05$	–	$1.3319e-05$	–	$7.7293e-06$	–	$2.2578e-04$	–
	32	$4.3423e-07$	6.48	$1.3452e-07$	6.63	$8.1494e-08$	6.57	$7.3483e-06$	4.94
	64	$5.1821e-09$	6.39	$1.4750e-09$	6.51	$8.8273e-10$	6.54	$1.4075e-07$	5.71
	128	$7.6636e-11$	6.08	$1.6792e-11$	6.46	$7.8655e-12$	6.81	$1.4510e-09$	6.60
	256	$1.1554e-12$	6.05	$1.5855e-13$	6.73	$6.9487e-14$	6.82	$2.0120e-11$	6.17

Table 5.1 – Convergence rates of the RBF-CWENO method using multiquadratics for the linear advection equation at time $T = 0.05$ with shape parameter $\varepsilon = 0.1$, CFL $= 0.01$.

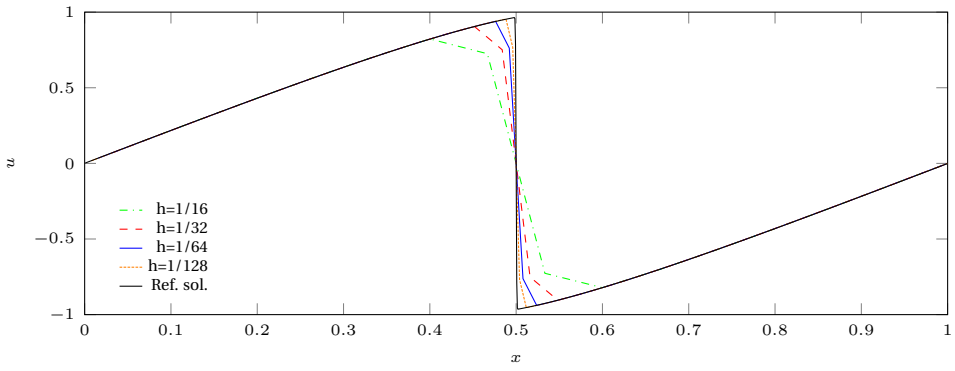


Figure 5.1 – Burgers' equation at $T = 0.3$ with $u_0 = \sin(2\pi x)$ solved by using the RBF-CWENO method with MQ interpolants of order $k = 3$.

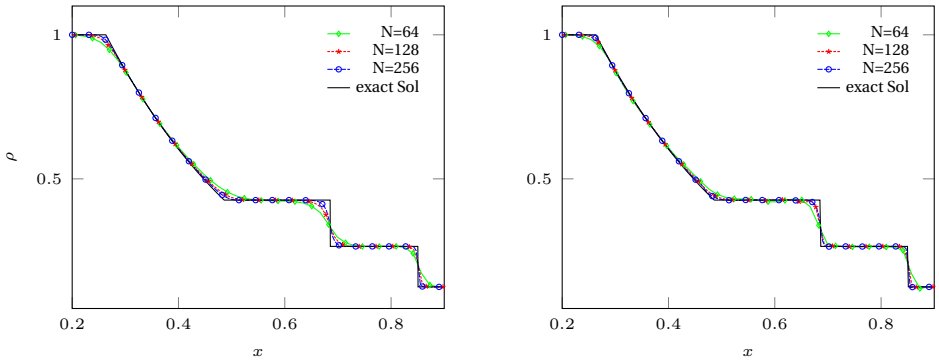


Figure 5.2 – Results for Sod's shock tube problem at $T = 0.2$ solved by using the RBF-CWENO method with MQ interpolants of order $k = 2, 3$ on characteristic variables (left: $k = 2$, right: $k = 3$) with $\mathrm{CFL} = 0.1$.

5.3.3 Euler equations

Next, we consider the one-dimensional Euler equations (3.96) with the ratio of specific heat $\gamma = 1.4$ for an ideal gas. Note that for $k = 3$ we need to change the nonlinear weights (5.5) by using $\bar{\epsilon} = \hat{\epsilon} h^2$ with $\hat{\epsilon} = 10^{-6}$ to avoid oscillations. Further, the CWENO reconstruction is done in the characteristic variables $\mathbf{V} = R^{-1}\mathbf{U}$, with the eigenvectors R from (2.13).

Sod's shock tube problem

The Sod's shock tube problem describes two colliding gases in $[0, 1]$ with different densities given by the initial conditions

$$(\rho_0, m_0, p_0) = \begin{cases} (1, 0, 1) & \text{if } x < 0.5 \\ (0.125, 0, 0.1) & \text{if } x \geqslant 0.5 \end{cases}. \tag{5.12}$$

This results in a rarefaction wave followed by a contact and a shock discontinuity which separates the domain into four domains with constant variables. The RBF-CWENO method resolves it well, see Figure 5.2. For $k = 3$, we observe minor oscillations, but their amplitude decreases for increasing number of elements. Furthermore, we observe the increasing accuracy for $k = 3$ compared to $k = 2$.

Shu-Osher shock-entropy wave interaction problem

The Shu-Osher problem describes the interaction of a discontinuity with a low frequency wave which introduces some high frequency waves. Its initial conditions are

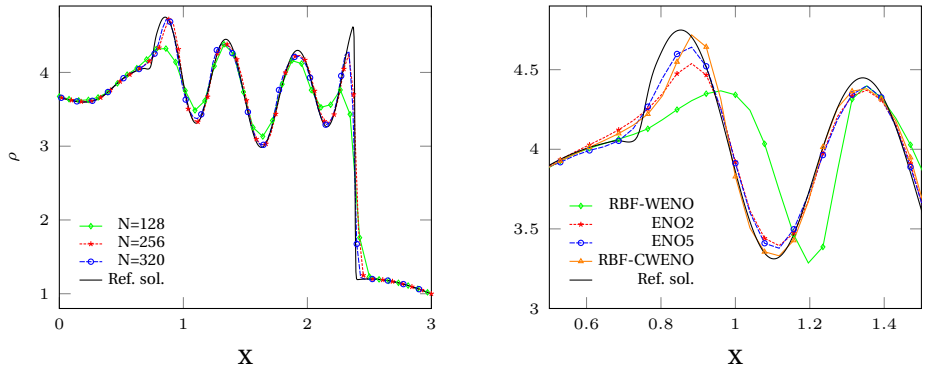


Figure 5.3 – Results for the Shu-Osher problem at $T = 1.8$ solved by using the RBF-CWENO method with MQ interpolants of order $k = 2$ on characteristic variables (left) and a comparison with WENO, ENO2 and ENO5 for $N = 256$ cells (right) with $\mathrm{CFL} = 0.2$.

$$(\rho_0, m_0, p_0) = \begin{cases} (3.857143, 2.629369, 10.33333) & \text{if } x < -4 \\ (1 + 0.2\sin(5x), 0, 1) & \text{if } x \geqslant -4 \end{cases}. \tag{5.13}$$

In Figure 5.3, we observe on the left side the increasing accuracy for increasing number of elements for $k = 2$. On the right side, we see its good behaviour compared to the existing methods ENO2, ENO5, and the RBF-WENO. In particular, we observe that the performance of the RBF-CWENO ($k = 2$) is comparable to ENO5 and superior to WENO ($k = 2$).

5.4 Two-dimensional RBF-CWENO method of third order

To generalize the CWENO method for two-dimensional problems and to introduce a competitive alternative to the RBF-ENO method for general grids we need to specify the choice of the different stencils and the smoothness indicator. A third order CWENO method on structured grids for multidimensional problems was introduced by Levy et al. [85]. Further, Semplice et al. [111] developed a generalization to non-uniform quad-tree meshes. We adapt the simple technique from the multi-resolution WENO scheme [137] such that we have three different stencils of different size for the second order reconstruction. We use the central spatial stencils starting with the single cell stencil $S_i^1 = \{C_i\}$. For the first order reconstruction, we use $S_i^2 = \{0, 1, 2, 3\}$ using the notation in Figure 5.4. The second order reconstruction is a bit different. Here, we do not use all 10 cells from $\tilde{S}_i^0$, but the first 8 such that $S_i^1 \subset S_i^0 \subset \tilde{S}_i^0$. Note that depending on the structure of the grid $|\tilde{S}_i^0|$ might be smaller than 10.
Concerning solvability we adapt the strategy from the RBF-ENO method. We choose

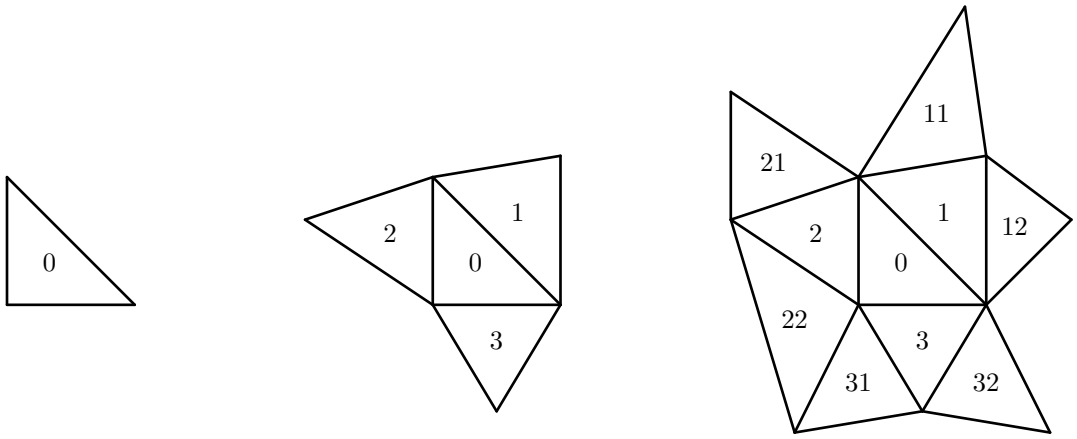


Figure 5.4 – Central stencils for the third order CWENO method. From the left to the right: S_i^1, S_i^2 and $\tilde{S}_i^0$.

(4.43) as shape parameter and

$$l = \left\lfloor -1.5 + \frac{1}{2}\sqrt{1+8n} \right\rfloor, \tag{5.14}$$

as the polynomial degree with the stencil size n. The nonlinear coefficients are not calculated directly by s_i^j as in (5.5), but we define $\tilde{s}_i^j : \mathbb{R}^2 \to \mathbb{R}$

$$\tilde{s}_i^1(\mathbf{x}) = s_i^1(\mathbf{x}), \tag{5.15}$$
$$\tilde{s}_i^2(\mathbf{x}) = s_i^2(\mathbf{x}) - s_i^1(\mathbf{x}), \tag{5.16}$$
$$\tilde{s}_i^0(\mathbf{x}) = s_i^0(\mathbf{x}) - s_i^2(\mathbf{x}). \tag{5.17}$$

Thus, we use the nonlinear weights

$$\alpha_j = \frac{d_j}{(\mathrm{IS}_{C_i}[\tilde{s}_i^j] + \bar{\epsilon})^t}, \tag{5.18}$$

instead of (5.5) with

$$\bar{\varepsilon} = \sqrt{|C_0|}, \qquad t = 1, \qquad d_j = \frac{1}{3}. \tag{5.19}$$

The smoothness indicator additionally uses all derivatives from the RBF part

$$\begin{aligned} \mathrm{IS}_C[s] = &\sum_{|j|=1}^{l} |C|^{|j|-1} \int_C \Big(\frac{\partial^{|j|} p(\mathbf{x})}{\partial x_1^{j_1} \partial x_2^{j_2}} \Big)^2 \mathrm{d}\mathbf{x} \\ &+ \sum_{|j|=1}^{l} |C|^{|j|-1} \int_C \left(\frac{\partial^{|j|}}{\partial x_1^{j_1} \partial x_2^{j_2}} \Big[\sum_{i=1}^{n} a_i \lambda_{C_i}^{\xi} \phi(\|\mathbf{x} - \xi\|) \Big] \right)^2 \mathrm{d}\mathbf{x}, \end{aligned} \tag{5.20}$$

with $p \in \Pi_l(\mathbb{R}^2)$. Note that in case of the single-cell stencil, the smoothness indicator will always be zero since the interpolation can be done exactly with

$$s_i^1(\mathbf{x}) = U_i \qquad \text{for all } \mathbf{x}. \tag{5.21}$$

Nevertheless, to be able to measure smoothness we adapt the method from [137]. Therefore, we define three polynomials $p_{1,j}(\mathbf{x}) \in \text{span}(x_1 - x_{1,j}, x_2 - x_{2,j})$ such that $p_{1,j}(\mathbf{x}_{j1}) = U_{j1}$ and $p_{1,j}(\mathbf{x}_{j2}) = U_{j2}$ for $j = 1,2,3$ with $\mathbf{x}_j = (x_{1,j}, x_{2,j})^T$ being the barycenter of cell C_j for $j = 1, 11, 12, 2, 21, 22, 3, 31, 32$ and $\mathbf{x} = (x_1, x_2)^T$. We define

$$\beta_{1,j} = \text{IS}_C[p_{1,j}], \qquad j = 1,2,3, \tag{5.22}$$

$\overline{\lambda}_{1j} = 1$ for all j and

$$\lambda_{1j} = \frac{\overline{\lambda}_{1j}}{\overline{\lambda}_{11} + \overline{\lambda}_{12} + \overline{\lambda}_{13}}, \qquad \text{for } j = 1,2, \tag{5.23}$$

and $\lambda_{13} = 1 - \lambda_{11} - \lambda_{12}$. Further, we set

$$\sigma_j = \lambda_{1j}\left(1 + \frac{(\frac{|\beta_{1,1}-\beta_{1,2}|}{3} + \frac{|\beta_{1,2}-\beta_{1,3}|}{3} + \frac{|\beta_{1,3}-\beta_{1,1}|}{3})^2}{\beta_{1,j} + \varepsilon_0}\right), \qquad \text{for } l = 1,2,3, \tag{5.24}$$

and $\sigma = \sigma_1 + \sigma_2 + \sigma_3$ with $\varepsilon_0 = 1.0e-10$. Thus, we get the smoothness indicator for the stencil of size one

$$\text{IS}_C[s_i^1] = |C| \sum_{|j|=1} \left(\frac{\partial^{|j|}}{\partial x_1^{j_1} \partial x_2^{j_2}}(\frac{\sigma_1}{\sigma} p_{1,1}(\mathbf{x}) + \frac{\sigma_2}{\sigma} p_{1,2}(\mathbf{x}) + \frac{\sigma_3}{\sigma} p_{1,3}(\mathbf{x}))\right)^2. \tag{5.25}$$

Note, we can replace the integration in (5.25) by the multiplication with $|C|$ since $\frac{\partial^{|j|}}{\partial x_1^{j_1} \partial x_2^{j_2}} \frac{\sigma_s}{\sigma} p_{1,s}(\mathbf{x}) = \text{const}$ for $|j| = 1$ and $s = 1,2,3$.

Remark 5.1. *The main difference between the RBF-CWENO and the RBF-ENO method from Chapter 4 lies in the number of evaluations of the smoothness indicator. The RBF-CWENO method needs three evaluations while the RBF-ENO method needs up to* 88 *in the worst case scenario with stencil size* 12.

5.5 Numerical results for the 2D RBF-CWENO method

In this section, we discuss the usage of the RBF-CWENO method on two-dimensional examples.

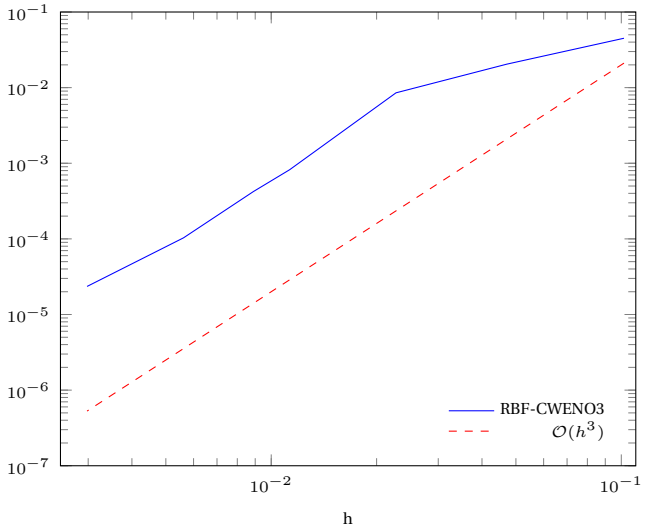


Figure 5.5 – Error for the RBF-CWENO method of 3rd order for the linear advection equation in 2D with CFL $= 0.8$.

5.5.1 Linear advection equation

We start by considering the two-dimensional linear advection equation (4.44) with wave speed $a = 1$, $b = 0$, periodic boundary conditions and the initial conditions (4.45). In Figure 5.5, we can observe that the convergence of order 3 is achieved, as for the RBF-ENO method. Note that the smoothness indicator does not recognize the smoothness of the solution for the coarse grids. However, for smaller grid sizes the method works as expected.

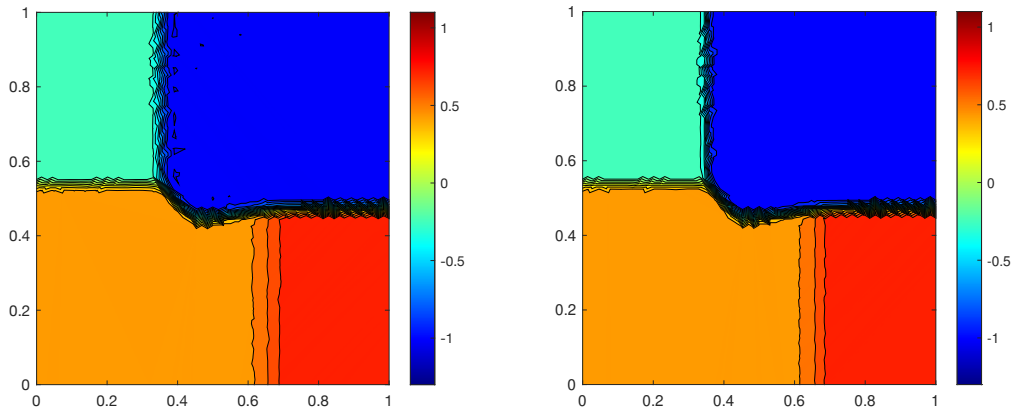
5.5.2 Burgers' equation

In this example we compare the computational cost of the RBF-CWENO method and the one of the RBF-ENO method. Let us consider Burgers' equation (4.47) with initial condition (4.48). We calculate the solution in the extended domain introduced in Figure 4.4 with $N = 1358, 5402, 12018, 21382$ cells.The comparison of the computational cost on one single core is listed in Table 5.2. The difference in the quality of the

N	RBF-CWENO	RBF-ENO	Speed-up
1358	11	96	8.7
5402	95	810	8.5
12018	338	2925	8.6
21382	776	6840	8.8

Table 5.2 – Runtime comparison for the 3th order methods solving the 2D Burgers' equation on a single core, measured in seconds.

solutions with $N = 21382$ cells is minimal, see Figure 5.6. One of the few differences is



(a) RBF-CWENO of order 3.

(b) RBF-ENO of order 3.

Figure 5.6 – Solution of the Burgers' equation at $T = 0.25$ with $N = 21382$ cells, $\mathrm{CFL} = 0.8$.

that the CWENO method introduces some small oscillations.

5.5.3 Euler equations

After ensuring the right order of convergence for a simple scalar problem, we present the results for the two-dimensional Euler equations (2.9) with $\gamma = 1.4$. These examples show the applicability of the method to complex problems. We consider the configurations discussed in Section 4.3.4.

Isentropic vortex

We start by considering the initial conditions (4.51) with the parameters $M = 0.5$, $\alpha = 0$, $\beta = 5$ and $(x_c, y_c) = (0, 0)$. Again, we analyze the order of convergence at time $T = 1$. We observe in Figure 5.7 the same behaviour as in the case of the linear advection equation. For coarse grids the smoothness indicator picks the low order reconstruction, but for the fine grids we receive the right order of convergence.

Shock vortex interaction problem

Next, we check the CWENO method when dealing with the interaction of a left-moving shock and a right-moving vortex. We consider (4.54), (4.55) and (4.56). Also, the parameter of the vortex are chosen as in Section 4.3.4 $\epsilon = 0.3$, $r_c = 0.05$, $\beta = 0.204$ with the $(x_c, y_c) = (0.25, 0.5)$. Comparing the RBF-CWENO method of order 3 with the standard first order finite volume method with the Rusanov flux, we clearly observe the

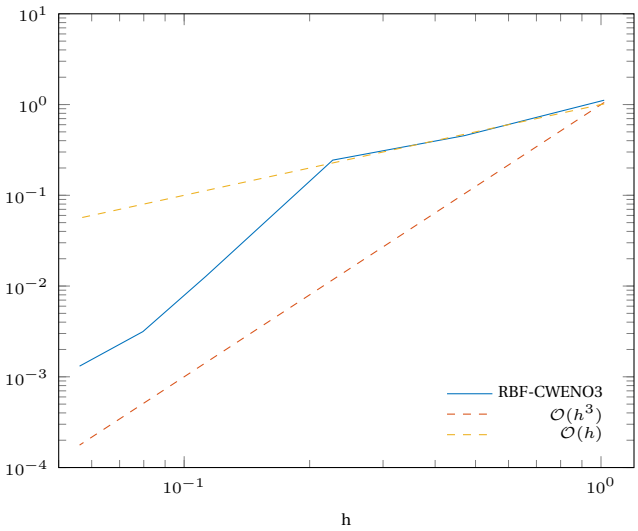
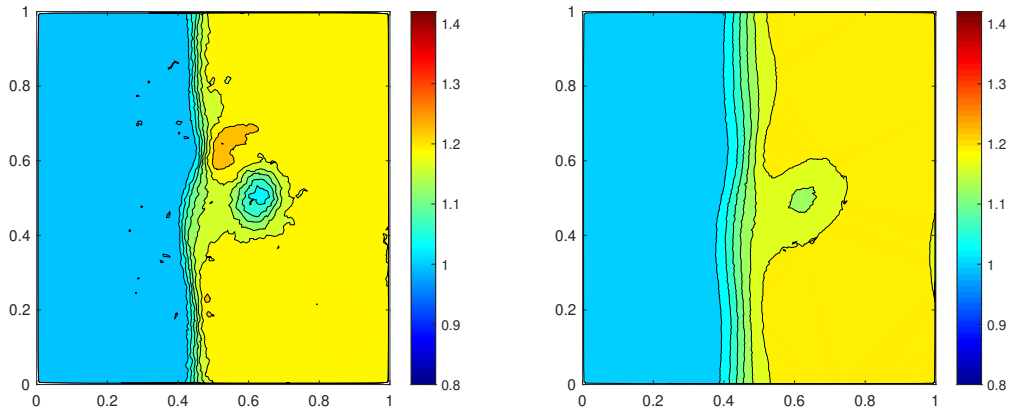


Figure 5.7 – Error for the RBF-CWENO method of 3rd order for the isentropic vortex interaction problem in 2D with $\mathrm{CFL} = 0.8$.

high-order accuracy in Figure 5.8. In comparison with the solution of the RBF-ENO method of order 3, Figure 4.13b, the solution of the RBF-CWENO method is more oscillatory and the shock is less sharp. However, the vortex seems to be of similar accuracy.

Double Mach reflection problem

Finally, we consider the double Mach reflection problem that describes a Mach 10 shock wave running into a ramp at an angle of 30°. We consider the same setting as in Figure 4.15 and we compare the results at time $T = 0.2$. Note that we do not compare exactly the same grid, but a slightly coarser one of similar type. The result is stable, but the shocks are less sharp than the ones from the RBF-ENO method in Figure 4.16. Furthermore, there are no details resolved in the turbulent part of the solution. Even in the much higher resolved case with around four times more cells, see Figure 5.10, the smooth features are less resolved than in the result of the RBF-ENO method with less cells.



(a) RBF-CWENO of order 3.

(b) Standard FVM of 1st order.

Figure 5.8 – Shock vortex interaction problem at $T = 0.35$ with $N = 16392$ cells, $\mathrm{CFL} = 0.8$ and 20 contour lines in $[0.8, 1.42]$.

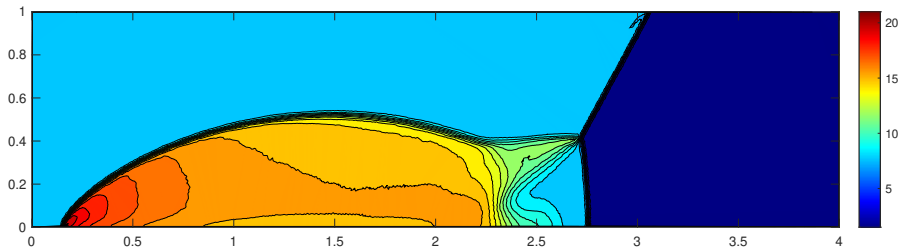


Figure 5.9 – Double Mach reflection problem at $T = 0.2$ with $N = 133594$ cells, $\mathrm{CFL} = 0.5$ solved with the RBF-CWENO of order 3 and displayed with 26 contour lines in $[1.5, 21]$.

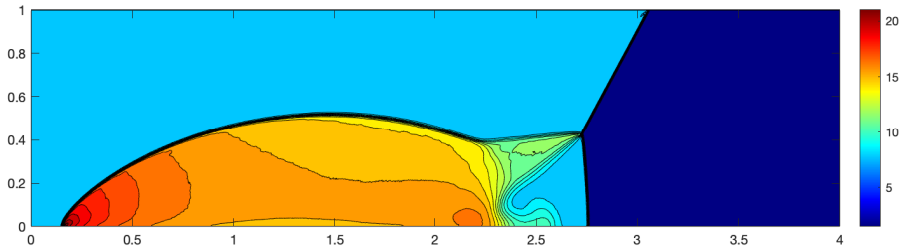


Figure 5.10 – Double Mach reflection problem at $T = 0.2$ with $N = 532222$ cells, $\mathrm{CFL} = 0.5$ solved with the RBF-CWENO of order 3 and displayed with 26 contour lines in $[1.5, 21]$.

6 Hybrid high-resolution ENO method

In Chapter 4, we presented the RBF-ENO method which is highly flexible in terms of geometry and furthermore ensures high order of accuracy. We introduced in the previous chapter the fast RBF-CWENO method. While it is substantially faster, in terms of resolution it is not a good alternative. In this chapter, we introduce a hybrid high-resolution method based on the standard WENO method on structured grids and the RBF-ENO method on the unstructured parts to reduce the overall computational cost while maintaining geometric flexibility.

6.1 Hybrid grid generation in one dimension

The basic idea is to split the domain into structured and unstructured parts. Let us take the example in Figure 6.1 with the structured part $[a,b]$ and the unstructured part $[b,c]$. In preparation for the two-dimensional case, we denote the unstructured and the structured part the triangular and the quadrilateral part, respectively. The connection between the different patches is done by using ghost cells. We divide the set of ghost cells into the structured/quadrilateral cells $\mathrm{GHOST}_{\mathrm{QUAD}}$ and the unstructured/triangular ones $\mathrm{GHOST}_{\mathrm{TRI}}$. Further, we denote the set of internal cells of the whole grid $\mathrm{INTERNAL}$, the set of all edges connected to at least one internal cell Edg, the set of edges at the boundaries such that the cells on their left are outside the patch $\mathrm{Edg}_{\mathrm{BC},L}$, and the ones such that the cells on their right are outside the patch $\mathrm{Edg}_{\mathrm{BC},R}$. The idea of the hybrid method is to enlarge the domains by $n_{\mathrm{ghost}} \in \mathbb{N}$ ghost cells on each side and create the maps

$$f_{\mathrm{TRI}} : \mathrm{GHOST}_{\mathrm{TRI}} \to \mathrm{INTERNAL}, \tag{6.1}$$

$$f_{\mathrm{QUAD}} : \mathrm{GHOST}_{\mathrm{QUAD}} \to \mathrm{INTERNAL}, \tag{6.2}$$

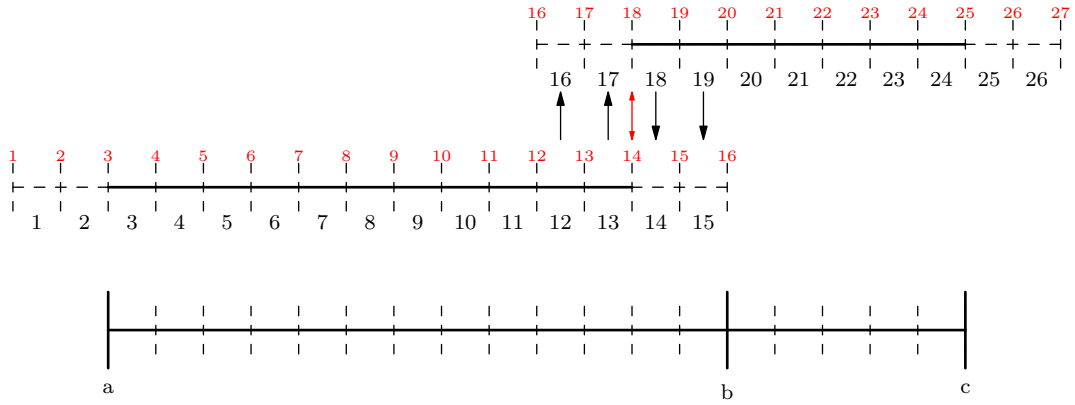


Figure 6.1 – Principle of 1D hybrid grids with $n_{\text{ghost}} = 2$, the black numbers are the labels for the cells and red ones are the labels for the edges.

to update the ghost cell values in the following way

$$U_i = U_{f_{\text{TRI}}(i)}, \qquad \text{for all } i \in \text{GHOST}_{\text{TRI}}, \tag{6.3}$$

$$U_j = U_{f_{\text{QUAD}}(j)}, \qquad \text{for all } j \in \text{GHOST}_{\text{QUAD}}\,. \tag{6.4}$$

Remark 6.1. *It is important that we are not directly using the set of structured cells in* $[a,b]$ *and the unstructured cells in* $[b,c]$*. To guarantee that the definition of the mappings make sense, we copy* n_{ghost} *cells from the structured grid to the neighboring unstructured cells.*

Now, we are able to run the WENO method on the structured parts and the RBF-ENO method on the unstructured ones and receive

$$\begin{aligned} &s_{i\pm1/2} && \text{for all } i \in \text{Edg}\setminus(\text{Edg}_{\text{BC},L} \cup \text{Edg}_{\text{BC},R}),\\ &s_{i+1/2} && \text{for all } i \in \text{Edg}_{\text{BC},L},\\ &s_{i-1/2} && \text{for all } i \in \text{Edg}_{\text{BC},R}\,. \end{aligned}$$

To set the remaining values, we define the maps

$$f_{L2R} : \text{Edg}_{\text{BC},L} \to \text{Edg}_{\text{BC},L} \cup \text{Edg}_{\text{BC},R}, \tag{6.5}$$

$$f_{R2L} : \text{Edg}_{\text{BC},R} \to \text{Edg}_{\text{BC},L} \cup \text{Edg}_{\text{BC},R}, \tag{6.6}$$

in such a way that for all $i \in \text{Edg}_{\text{BC},R}$ and $j \in \text{Edg}_{\text{BC},L}$ with $x(i) = x(j)$

$$\begin{aligned} f_{R2L}(i) &= j,\\ f_{L2R}(j) &= i, \end{aligned}$$

with the function $x : \text{Edg} \to \mathbb{R}$ that assigns each edge to its physical position. For edges

on the real boundary these functions depend on the specific boundary conditions. Since each interface i is assigned two values $s_{i\pm1/2}$, we can calculate the numerical flux through each interface and calculate the approximated solution for the next time step.

Example 6.1. *Let us take a look at the example from Figure 6.1. The sets of edges are* $\mathrm{Edg} = \{3,\dots 14\} \cup \{18,\dots 25\}$, $\mathrm{Edg}_{\mathrm{BC},L} = \{3,18\}$ *and* $\mathrm{Edg}_{\mathrm{BC},R} = \{14,25\}$. *The maps to update the ghost cells are given by*

$$\begin{aligned} f_{\mathrm{QUAD}}(14) &= 18,\\ f_{\mathrm{QUAD}}(15) &= 19,\\ f_{\mathrm{TRI}}(16) &= 12,\\ f_{\mathrm{TRI}}(17) &= 13, \end{aligned}$$

and $f_{\mathrm{QUAD}}(1), f_{\mathrm{QUAD}}(2), f_{\mathrm{TRI}}(25), f_{\mathrm{TRI}}(26) \in \{3,\dots,13\} \cup \{18,\dots,24\}$ *depending on the boundary conditions. The remaining functions are given by*

$$f_{R2L}(14) = 18, f_{L2R}(18) = 14,$$

and $f_{L2R}(3), f_{R2L}(25) \in \{3,14,18,25\}$.

6.2 Hybrid grid generation in two dimensions

The idea of the two-dimensional method follows the same idea, i.e., we split the domain into structured and unstructured parts, see Figure 6.2. At each time step, we update the ghost cells to connect the different patches and in the structured parts we use a standard two-dimensional WENO method and in the unstructured parts we apply the RBF-ENO method. Next, we update the missing left or right reconstruction values at each interface. Note that the standard WENO method is based on ghost cells on each side, but the RBF-ENO method is not, see Section 4.2.1. Let us define $\Omega \in \mathbb{R}^2$ as the interior of the computational domain such that the ghost cells from the WENO method at the boundary are outside of Ω. Similar to the one-dimensional version, we add n_{ghost} squares from the structured to the unstructured part, but we triangulate them artificially, e.g., the green structured triangulation in Figure 6.2. To ensure the connection between the domains we create the ghost cells for the green structured triangulation and define a map between the ghost cells of the triangular side and the overlapping quadrilaterals of the structured grid and vice versa. We have the following two kind of ghost cells

- ghost cells that connect two different patches (they overlap with interior cells of other patches);
- ghost cells that are outside of the boundary to apply the structured WENO

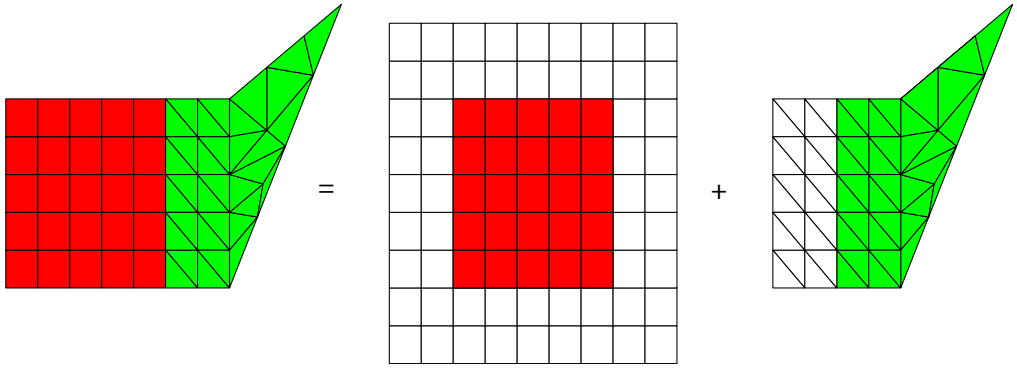


Figure 6.2 – Principle of the division into patches of structured and unstructured grids with $n_{\text{ghost}} = 2$. White cells are ghost cells, which are updated either by the boundary conditions or due to mappings in between the patches.

method (they are always quadrilaterals).

We define the three maps

$$f_{\text{TRI}} : \text{GHOST}_{\text{TRI}} \to \text{INTERNAL}, \tag{6.7}$$
$$f_{\text{QUAD},1} : \text{GHOST}_{\text{QUAD}} \to \text{INTERNAL}, \tag{6.8}$$
$$f_{\text{QUAD},2} : \text{GHOST}_{\text{QUAD}} \to \text{INTERNAL}, \tag{6.9}$$

to set the value for each ghost cell. These maps have the following properties

- For each $T \in \text{GHOST}_{\text{TRI}}$ there exists one $\tilde{T} \in \text{GHOST}_{\text{TRI}}$ such that $f_{\text{TRI}}(T) = f_{\text{TRI}}(\tilde{T})$ and $T \neq \tilde{T}$. Furthermore, it holds $T, \tilde{T} \subset f_{\text{TRI}}(T)$;
- For each $Q \in \text{GHOST}_{\text{QUAD}}$ with $Q \subset \Omega$ there exist $T, \tilde{T} \in \text{INTERNAL}_{\text{TRI}}$ or $\tilde{Q} \in \text{INTERNAL}_{\text{QUAD}}$ with $T \neq \tilde{T}$ such that $f_{\text{QUAD},1}(Q) = T$ and $f_{\text{QUAD},2}(Q) = \tilde{T}$ or $f_{\text{QUAD},1}(Q) = f_{\text{QUAD},2}(Q) = \tilde{Q}$. Again, we have the condition $T, \tilde{T}, \tilde{Q} \subset Q$;
- For each $Q \in \text{GHOST}_{\text{QUAD}}$ with $Q \not\subset \Omega$, there exists $\tilde{Q} \in \text{INTERNAL}_{\text{QUAD}}$ such that $f_{\text{QUAD},1}(Q) = f_{\text{QUAD},2}(Q) = \tilde{Q}$.

Instead of the update (6.3) and (6.4) in the two-dimensional case we use the average of the two overlapping triangles with the quadrilateral ghost cell (in case of a QUAD to QUAD map $f_{\text{QUAD},1}(Q) = f_{\text{QUAD},2}(Q)$)

$$U_T = U_{f_{\text{TRI}}(T)}, \qquad \text{for each } T \in \text{GHOST}_{\text{TRI}}, \tag{6.10}$$
$$U_Q = \frac{U_{f_{\text{QUAD},1}(Q)} + U_{f_{\text{QUAD},2}(Q)}}{2}, \qquad \text{for each } Q \in \text{GHOST}_{\text{QUAD}}\,. \tag{6.11}$$

The functions (6.5) and (6.6) can be defined in the same way as before, since every edge has a unique direction which defines a right and left cell for each edge.
To define the different patches and maps for hybrid grids in multiple dimensions, we need to introduce some more tools. Let us defined the following kind of patches

- quadrilaterals (QUAD);
- connection patches between two QUADs (Q2Q);
- connection patches between multiple Q2Qs (RQ);
- the triangular patches (TRI).

To automate the generation of the ghost cells we divide the TRI patches into

- the principle triangular part (TRI0);
- the connection patches between TRI0s and QUADs (Q2T);
- the small connection patches that connect all kind of combinations of Q2Ts and Q2Qs in the case of at least one Q2T (RT).

Figure 6.3 illustrates a way to combine quadrilateral grids. The L-shaped domain is divided into three QUADs, four Q2Qs, and one RQ patches. Note that we can use Q2Q-patches also at the boundary. The only restriction is that the grid size in each direction is uniform and we require that each side length is a multiple of its grid size. Let us take a look at Figure 6.4 to illustrate how to combine the pieces. We have a single TRI0, two Q2T, one RT, two Q2Q and three QUAD patches. The only unstructured patches are the TRI0s. The Q2T's are long patches of width $n_{\text{ghost}}\Delta x$ or $n_{\text{ghost}}\Delta y$ with a structured triangulation and ghost cells only in one direction. The RTs are of size $n_{\text{ghost}}\Delta x \times n_{\text{ghost}}\Delta y$ with a structured triangulation and its ghost cells are added just in the direction of Q2Q patches and over the corners in between two Q2Q patches.

Given the tools described above we can construct hybrid grids for general geometries. This hybrid method can be used to locally refine grids and apply a fast structured solver around this refined region. Figure 6.5 shows a possible local refinement with a central unstructured domain.

6.3 Setting of the WENO and RBF-ENO methods

In this section, we describe the specific setting of the RBF-ENO and WENO methods which are used on the hybrid grids.

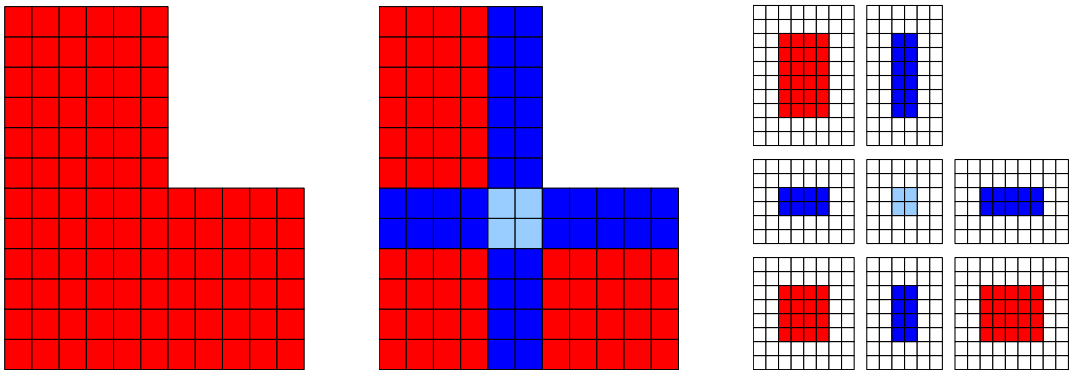


Figure 6.3 – Principle of the division into patches of just structured grids for an L-shaped domain with $n_{\text{ghost}} = 2$. White cells are ghost cells, which are updated either by the boundary conditions or due to mappings in between the patches.

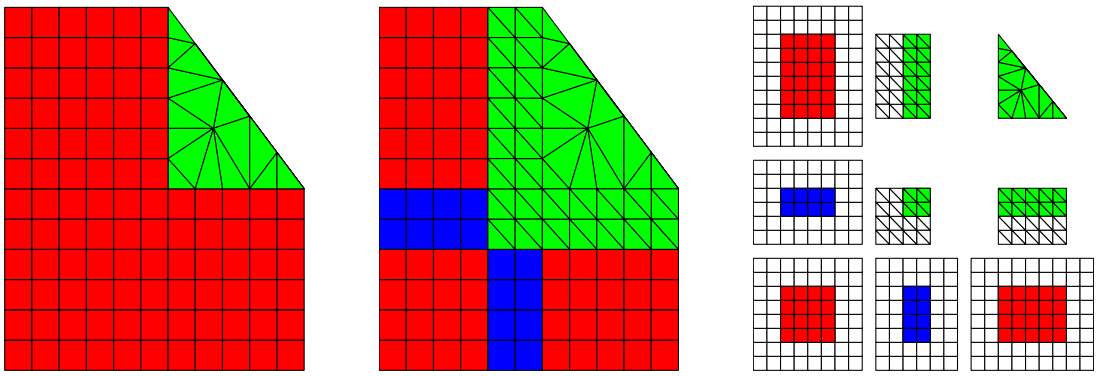
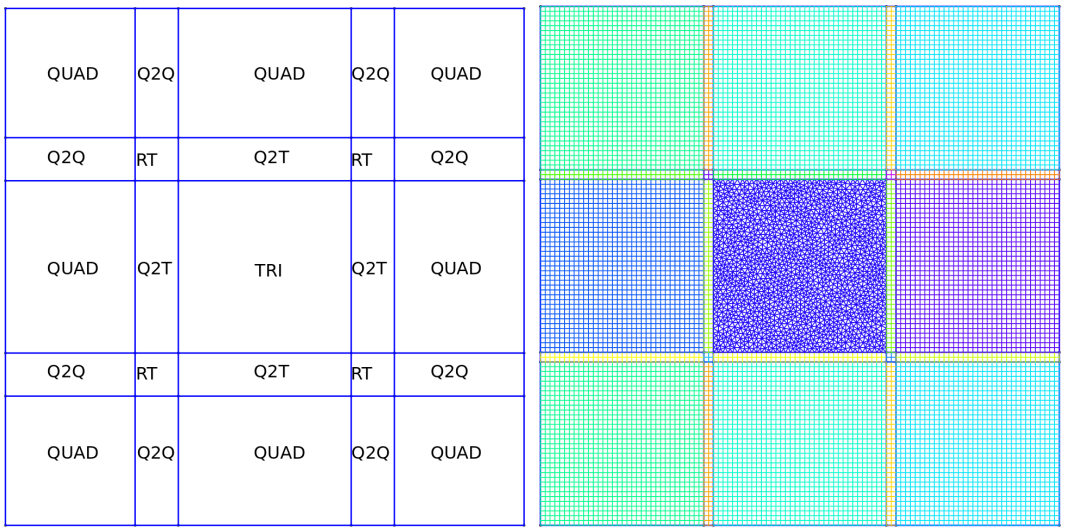


Figure 6.4 – Principle of the division into patches of structured and unstructured grids with $n_{\text{ghost}} = 2$. White cells are ghost cells, which are updated either by the boundary conditions or due to mappings in between the patches.



(a) Schematic illustration of the patches. (b) Grid with $N = 15108$ cells.

Figure 6.5 – Hybrid grid with a central unstructured part.

6.3.1 One-dimensional hybrid method

We use on the structured patches the standard WENO method of order p_{WENO} and on the unstructured patches the RBF-ENO method of order p_{ENO}. Let us consider the one-dimensional version of the RBF-ENO method, introduced in Chapter 4. It is based on the MUSCL approach from Section 2.2 with the RBF reconstruction augmented with polynomials from Section 4.1 and the shape parameter

$$\varepsilon = \frac{1}{\Delta x}. \tag{6.12}$$

Instead of the Algorithm 4.1, we use its one-dimensional version Algorithm 3.1 to choose the least oscillatory reconstruction.
To construct a method of order p the choice of $p_{\mathrm{ENO}} = p$ is given. For the WENO method there are two possibilities

$$p_{\mathrm{WENO}} = 2\left\lfloor \frac{p}{2} \right\rfloor + 1, \qquad \text{with the stencil size } n = 2\left\lfloor \frac{p}{2} \right\rfloor + 1, \tag{6.13}$$

$$p_{\mathrm{WENO}} = 2p - 1, \qquad \text{with the stencil size } n = 2p - 1, \tag{6.14}$$

with different orders of convergence. The following theorem states the stability result.

Theorem 6.1 (Stability and order of convergence)**.** *Given the hybrid RBF-ENO method with* $p_{\mathrm{ENO}} = p$ *and*

$$p_{\mathrm{WENO}} = 2\left\lfloor \frac{p}{2} \right\rfloor + 1, \tag{6.15}$$

or

$$p_{\mathrm{WENO}} = 2p - 1. \tag{6.16}$$

It provides an accuracy of order p for smooth solutions. Furthermore, the combination of the two methods is stable if Δt is the smallest time step fulfilling the CFL-condition over all patches and the number of ghost cells $n_{\mathrm{ghost}} \geqslant p - 1$.

Proof. Given the RBF-ENO method of order $p_{\mathrm{ENO}} = p$, the WENO method of order p_{WENO} by (6.13) or (6.14) and the number of ghost cells $n_{\mathrm{ghost}} = p - 1$, we get that in the smooth case each part of the method has an accuracy of order p. Thus, each flux is of order $p - 1$. Since $n_{\mathrm{ghost}} \geqslant p - 1$ the reconstruction of both the RBF-ENO and the WENO method is locally the same as for each individual reconstruction. In the end, for Δt the smallest time step fulfilling the CFL-condition, we get the same stability as for each method itself. □

Note that to our knowledge, there are no general stability results for the WENO and ENO method. The stability we conjecture states that the hybrid method is as stable as the single methods and the combination of the two does not destroy this.

6.3.2 Two-dimensional hybrid method of order three

In two space dimensions, we restrict ourselves to the case $p_{\mathrm{ENO}} = 3$ on the unstructured patches. Since the two-dimensional WENO method is based on dimensional splitting we have the same conditions (6.13) or (6.14) on the structured patches. We use the standard WENO method of order $p_{\mathrm{WENO}} = 5$ since the computational cost is similar. To receive $p_{\mathrm{WENO}} = 5$ in the smooth case we need

$$p_{\mathrm{WENO}} \leqslant 2n_{\mathrm{ghost}} + 1. \tag{6.17}$$

Furthermore, we need the number of ghost cells n_{ghost} to be large enough such that the RBF-ENO method is flexible enough to avoid oscillatory states. This results in the stability result of Theorem 6.1.

Theorem 6.2 (Stability)**.** *The high-order hybrid RBF-ENO method is stable with respect to the smallest time step Δt over all patches if the number of ghost cells n_{ghost} is large enough such that* (6.17) *is fulfilled and such that all neighbors until $d_{\max}$ are inside the ghost cell area.*

Proof. To have no restrictions for the RBF-ENO method, we need to choose n_{ghost} such that all neighbors until $d_{\max}$, the maximal distance introduced in Table 4.2, are inside the ghost cell patches. For the WENO method we need (6.17). To get the same stability

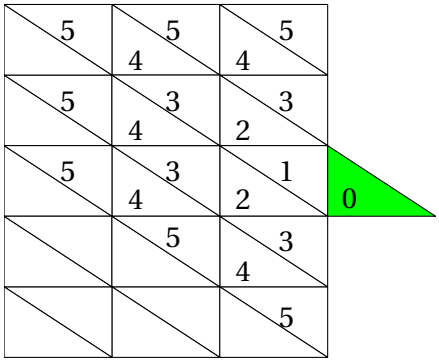


Figure 6.6 – Number of ghost cells n_{ghost} needed for the RBF-ENO method depending on $d_{\max}$ from Table 4.2.

as for each single method, it remains to satisfy the CFL condition for Δt on each patch of the computational domain. □

Remark 6.2. *In one dimension, this method is high-order accurate. However, if we implement the WENO method in two-dimensions with the standard flux splitting we recover only a high-resolution method. There is a way of evaluating the WENO reconstruction on each edge at some high-order quadrature nodes, but this is costly. Another possibility could be the accuracy correction proposed by Buchmüller and Helzel [14].*

For the RBF-ENO method of order 3 we have $d_{\max} = 5$, thus with $n_{\text{ghost}} = 3$ all neighbors can be considered, see Figure 6.6. However, except for the final example of the flow through a conical aerospike nozzle, we choose $n_{\text{ghost}} = 2$. In the last example, we must choose $n_{\text{ghost}} = 3$ to circumvent negative pressure.

6.4 Maximum preserving limiter

In this section, we show that the maximum preserving principle introduced by Perthame and Shu [100] and Zhang and Shu [133] can be generalized for non-polynomial reconstructions. Hence, we can apply it for the triangular part of the hybrid high-resolution RBF-ENO method. The structured part can be stabilized using the positivity preserving limiter for the WENO method in each direction.

6.4.1 Generalized maximum preserving limiter

The maximum principle satisfying finite volume method is based on the first order finite volume scheme (2.35)

$$U_i^{n+1} = U_i^n - \lambda[F(U_i^n, U_{i+1}^n) - F(U_{i-1}^n, U_i^n)] =: H_\lambda(U_{i-1}^n, U_i^n, U_{i+1}^n), \tag{6.18}$$

with a monotone numerical flux F and $\lambda = \Delta t/\Delta x$. For suitable numerical flux functions, e.g., the Rusanov and the Godunov scheme, H_λ is increasing in each argument under the CFL condition $\max_u |\nabla_u f(u)|\lambda \leqslant 1$. Using the consistency of the flux we have the maximum principle

$$m = H_\lambda(m, m, m) \leqslant U_i^{n+1} = H_\lambda(U_{i-1}^n, U_i^n, U_{i+1}^n) \leqslant H_\lambda(M, M, M) = M, \tag{6.19}$$

for $m \leqslant U_{i-1}^n, U_i^n, U_{i+1}^n \leqslant M$.
Let us consider the high-order MUSCL scheme

$$U^{n+1} = U^n - \lambda[F(u_{i+1/2}^-, u_{i+1/2}^+) - F(u_{i-1/2}^-, u_{i-1/2}^+)], \tag{6.20}$$

with $u_{i-1/2}^+ = p_i(x_{i-1/2})$ and $u_{i+1/2}^- = p_i(x_{i+1/2})$ of the high-order polynomial $p_i \in \Pi_k(\mathbb{R})$ interpolating on a stencil around the cell i. Note, it is enough to show the idea for the forward Euler method in time since the MUSCL scheme with a SSPRK method can be written as convex combinations of (6.20). The idea is to express the average value of each cell by the exact Gauss-Lobatto quadrature rule with nodes $\hat{x}_i^\alpha \in [x_{i-1/2}, x_{i+1/2}]$ and the weights $\hat{\omega}_\alpha$ for $\alpha = 1, \ldots, N$ with $2N - 3 \geqslant k$, i.e.,

$$U^n = \sum_{\alpha=1}^{N} \hat{\omega}_\alpha p_i(\hat{x}_i^\alpha), \tag{6.21}$$

with $\hat{x}_i^1 = x_{i-1/2}$ and $\hat{x}_i^N = x_{i+1/2}$. The maximum preserving limiter is based on the following form of (6.20)

$$\begin{aligned} U_i^{n+1} = &\sum_{\alpha=2}^{N-1} \hat{\omega}_\alpha p_i(\hat{x}_i^\alpha) + \hat{\omega}_N \Big(u_{i+1/2}^- - \frac{\lambda}{\hat{\omega}_N}[F(u_{i+1/2}^-, u_{i+1/2}^+) - F(u_{i-1/2}^+, u_{i+1/2}^-)] \Big) \\ &+ \hat{\omega}_1 \Big(u_{i-1/2}^+ - \frac{\lambda}{\hat{\omega}_1}[F(u_{i-1/2}^+, u_{i+1/2}^-) - F(u_{i-1/2}^-, u_{i-1/2}^+)] \Big), \end{aligned} \tag{6.22}$$

where we added and subtracted $F(u^+_{i-1/2}, u^-_{i+1/2})$. This can be expressed as

$$\begin{aligned} U_i^{n+1} &= \sum_{\alpha=2}^{N-1} \hat{\omega}_\alpha p_i(\hat{x}_i^\alpha) + \hat{\omega}_N H_{\lambda/\hat{\omega}_N}(u^+_{i-1/2}, u^-_{i+1/2}, u^+_{i+1/2}) + \hat{\omega}_1 H_{\lambda/\hat{\omega}_1}(u^-_{i-1/2}, u^+_{i-1/2}, u^-_{i+1/2}), \\ &= \sum_{\alpha=2}^{N-1} \hat{\omega}_\alpha p_i(\hat{x}_i^\alpha) + \hat{\omega}_N H_{\lambda/\hat{\omega}_N}(p_i(\hat{x}_i^1), p_i(\hat{x}_i^N), p_{i+1}(\hat{x}_{i+1}^1)) \\ &\quad + \hat{\omega}_1 H_{\lambda/\hat{\omega}_1}(p_{i-1}(\hat{x}_{i-1}^N), p_i(\hat{x}_i^1), p_i(\hat{x}_i^N)). \end{aligned} \tag{6.23}$$

Under the CFL condition

$$\lambda \max_u |\nabla_u f(u)| \leqslant \min_\alpha \hat{\omega}_\alpha, \tag{6.24}$$

and

$$m \leqslant p_j(\hat{x}_j^\alpha) \leqslant M, \qquad \text{for all } \alpha = 1, \ldots, N, \quad \text{and } j = i-1, i, j+1, \tag{6.25}$$

we ensure the satisfaction of the maximum principle $m \leqslant U_i^{n+1} \leqslant M$.
However, we can rewrite the method in a slightly different form

$$U_i^{n+1} = \hat{\omega}_R p_R + \hat{\omega}_N H_{\lambda/\hat{\omega}_N}(u^+_{i-1/2}, u^-_{i+1/2}, u^+_{i+1/2}) + \hat{\omega}_1 H_{\lambda/\hat{\omega}_1}(u^-_{i-1/2}, u^+_{i-1/2}, u^-_{i+1/2}), \tag{6.26}$$

with $\hat{\omega}_R = 1 - \hat{\omega}_N - \hat{\omega}_1 \geqslant 0$ and

$$p_R = \sum_{\alpha=2}^{N-1} \frac{\hat{\omega}_\alpha}{\hat{\omega}_R} p(\hat{x}_i^\alpha), \tag{6.27}$$

$$= \frac{U_i^n - \hat{\omega}_N u^-_{i+1/2} - \hat{\omega}_1 u^+_{i-1/2}}{\hat{\omega}_R}. \tag{6.28}$$

Again, we satisfy the maximum principle $m \leqslant U_i^{n+1} \leqslant M$ under the milder condition

$$m \leqslant p_R, u^-_{j+1/2}, u^+_{j-1/2} \leqslant M, \qquad \text{for } j = i-1, i, i+1. \tag{6.29}$$

Based on these results we define the limiter

$$\widetilde{u}_{\min} = \min\{p_R, u^-_{i+1/2}, u^+_{i-1/2}\}, \tag{6.30}$$

$$\widetilde{u}_{\max} = \max\{p_R, u^-_{i+1/2}, u^+_{i-1/2}\}, \tag{6.31}$$

$$\widetilde{\theta} = \min\left\{\left|\frac{U_i^n - m}{U_i^n - \widetilde{u}_{\min}}\right|, \left|\frac{U_i^n - M}{U_i^n - \widetilde{u}_{\max}}\right|, 1\right\}, \tag{6.32}$$

$$\widetilde{p}(x) = \widetilde{\theta}\big(p(x) - U_i^n\big) + U_i^n. \tag{6.33}$$

The original limiter from Zhang and Shu [133] is

$$u_{\min} = \min_{\alpha} p(\hat{x}_i^{\alpha}), \tag{6.34}$$

$$u_{\max} = \max_{\alpha} p(\hat{x}_i^{\alpha}), \tag{6.35}$$

$$\theta = \min\Big\{\Big|\frac{U_i^n - m}{U_i^n - u_{\min}}\Big|, \Big|\frac{U_i^n - M}{U_i^n - u_{\max}}\Big|, 1\Big\}, \tag{6.36}$$

$$\hat{p}(x) = \theta\big(p(x) - U_i^n\big) + U_i^n. \tag{6.37}$$

Lemma 6.3 verifies that the new limiter is conservative, maintains accuracy and

$$\begin{aligned} U_i^{n+1} = {} & \hat{\omega}_R \widetilde{p}_R + \hat{\omega}_N H_{\lambda/\hat{\omega}_N}(\widetilde{p}_i(\hat{x}_i^1), \widetilde{p}_i(\hat{x}_i^N), \widetilde{p}_{i+1}(\hat{x}_{i+1}^1)) \\ & + \hat{\omega}_1 H_{\lambda/\hat{\omega}_1}(\widetilde{p}_{i-1}(\hat{x}_{i-1}^N), \widetilde{p}_i(\hat{x}_i^1), \widetilde{p}_i(\hat{x}_i^N)), \end{aligned} \tag{6.38}$$

with

$$\widetilde{p}_R := \frac{U_i^n - \hat{\omega}_1 \widetilde{p}(\hat{x}_i^1) - \hat{\omega}_N \widetilde{p}(\hat{x}_i^N)}{\hat{\omega}_R}, \tag{6.39}$$

which satisfies the maximum condition.

Lemma 6.3. *The simplified positivity preserving limiter with*

$$\widetilde{\theta} = \min\Big\{\frac{U_i^n - \varepsilon}{U_i^n - \widetilde{u}_{\min}}, 1\Big\}, \qquad \widetilde{u}_{\min} = \min\{p_R, u_{i+1/2}^-, u_{i-1/2}^+\}, \tag{6.40}$$

is conservative, of high order and satisfies the maximum condition (6.29).

Proof. **Conservation:** Conservation is clear because p is conserved

$$\frac{1}{|C|}\int_C \widetilde{p}(x)\mathrm{d}x = \frac{\widetilde{\theta}}{|C|}\int_C p(x)\mathrm{d}x + (1-\widetilde{\theta})U_i^n = U_i^n. \tag{6.41}$$

Accuracy: Let us assume the case $\tilde{\theta} = \left|\frac{U_i^n - m}{U_i^n - \tilde{u}_{\min}}\right|$. The other case works in the same manner. From Zhang and Shu [135] we have

$$|\hat{p}(x) - p(x)| = \mathcal{O}(\Delta x^{k+1}). \tag{6.42}$$

Furthermore, we know

$$u_{\min} \leqslant \widetilde{u}_{\min}, \tag{6.43}$$

since $u_{i-1/2}^+ = p(\hat{x}_1)$, $u_{i+1/2}^- = p(\hat{x}_N)$ and p_R is a convex combination of values $p(\hat{x}_i^{\alpha})$. If we assume $\widetilde{\theta} < 1$ we obtain $u_{\min} \leqslant \widetilde{u}_{\min} < m$ and $\theta \leqslant \widetilde{\theta} \leqslant 1$. Using the definition of

the limiter and combining it with the previous results, we have

$$|\hat{p}(x) - p(x)| = |\theta(p(x) - U_i^n) + U_i^n - p(x)|, \tag{6.44}$$
$$= |\theta - 1||p(x) - U_i^n|, \tag{6.45}$$
$$\geqslant |\widetilde{\theta} - 1||p(x) - U_i^n| = |\widetilde{p}(x) - p(x)|. \tag{6.46}$$

With (6.42) we conclude

$$|\widetilde{p}(x) - p(x)| = \mathcal{O}(\Delta x^{k+1}). \tag{6.47}$$

Maximum preserving condition: By construction we have $m \leqslant \widetilde{p}(x_{i-1/2}), \widetilde{p}(x_{i+1/2}) \leqslant M$. Further, we have

$$\begin{aligned}
\widetilde{p}_R &= \frac{U_i^n - \hat{\omega}_1 \widetilde{p}(\hat{x}_i^1) - \hat{\omega}_N \widetilde{p}(\hat{x}_i^N)}{\hat{\omega}_R}, \\
&= \frac{\widetilde{\theta} U_i^n + (1-\widetilde{\theta}) U_i^n + \hat{\omega}_1(\widetilde{\theta} - 1) U_i^n - \hat{\omega}_1 \widetilde{\theta} p(\hat{x}_i^1) + \hat{\omega}_N(\widetilde{\theta} - 1) U_i^n - \hat{\omega}_N \widetilde{\theta}_p p(\hat{x}_i^N)}{\hat{\omega}_R}, \\
&= \widetilde{\theta} p_R + (1 - \widetilde{\theta}) U_i^n.
\end{aligned}$$

Thus, we have $m \leqslant \widetilde{p}_R \leqslant M$. □

6.4.2 WENO limiter

The problem with the standard WENO method compared to the ENO method is that we do not recover the high-order interpolation function, but just the values $u^{\pm}_{i\pm 1/2}$. Zhang and Shu [133] introduced a way to resolve this issue by reconstructing $p \in \Pi_k(\mathbb{R})$ using $u^+_{i-1/2}, u^-_{i+1/2}$ and surrounding cell averages U_j for $j = i - k_0, \ldots, i + k_1$ for $k_0, k_1 \in \mathbb{N}$. Given the idea from the previous section, we can create a limiter by using the extrema preserving limiter (6.33) without artificially generating a reconstruction $p \in \Pi_k(\mathbb{R})$. For the two-dimensional WENO method on structured grids, which is based on dimensional splitting [48], where we apply the one-dimensional maximum preserving limiter in each dimension.

6.4.3 Non-polynomial reconstruction

In the case of a non-polynomial reconstruction, condition (6.21) is not satisfied. However, we can make use of the concept behind (6.28). Let us consider the reconstruction $r : \mathbb{R} \to \mathbb{R}$ of order k. We define

$$p_R := \frac{U_i^n - \hat{\omega}_N u^-_{i+1/2} - \hat{\omega}_1 u^+_{i-1/2}}{\hat{\omega}_R}, \tag{6.48}$$

with the Gauss-Lobatto weights $\hat{\omega}_1, \hat{\omega}_N > 0$, $N \in \mathbb{N}$ such that $2N - 3 \geqslant k$ and

$$\hat{\omega}_R := 1 - \hat{\omega}_1 - \hat{\omega}_N. \tag{6.49}$$

As before, we can rewrite U_i^{n+1} using (6.26). The scheme is extrema preserving if (6.29) is fulfilled. If it is not fulfilled, we use $\widetilde{\theta}$ from (6.32) and define the limiter

$$\widetilde{r}(x) = \widetilde{\theta}\big(r(x) - U_i^n\big) + U_i^n. \tag{6.50}$$

As in the polynomial case, this limiter defines a high-order, conservative reconstruction fulfilling the extrema-preserving condition (6.29), see Lemma 6.4. Finally, we can define the extrema-preserving MUSCL scheme with non-polynomial reconstruction

$$U^{n+1} = U^n - \lambda[F(\tilde{r}_i(x_{i+1/2}), \tilde{r}_{i+1}(x_{i+1/2})) - F(\tilde{r}_{i-1}(x_{i-1/2}), \tilde{r}_i(x_{i-1/2}))], \tag{6.51}$$

which can be written in the following form

$$\begin{aligned} U_i^{n+1} = {} & \hat{\omega}_R \widetilde{p}_R + \hat{\omega}_N H_{\lambda/\hat{\omega}_N}(\widetilde{r}_i(\hat{x}_i^1), \widetilde{r}_i(\hat{x}_i^N), \widetilde{r}_{i+1}(\hat{x}_{i+1}^1)) \\ & + \hat{\omega}_1 H_{\lambda/\hat{\omega}_1}(\widetilde{r}_{i-1}(\hat{x}_{i-1}^N), \widetilde{r}_i(\hat{x}_i^1), \widetilde{r}_i(\hat{x}_i^N)), \end{aligned} \tag{6.52}$$

with

$$\widetilde{p}_R := \frac{U_i^n - \hat{\omega}_1 \widetilde{r}(\hat{x}_i^1) - \hat{\omega}_N \widetilde{r}(\hat{x}_i^N)}{\hat{\omega}_R}. \tag{6.53}$$

Lemma 6.4. *The simplified positivity preserving limiter* (6.50) *is conservative, of high order and satisfies the simplified maximum preserving condition* (6.29).

Proof. The proof for the consistency and the maximum preserving property works exactly the same as for Lemma 6.3. To show high-order accuracy, we introduce the polynomial $p \in \Pi_k(\mathbb{R})$ which interpolates the reconstructed values $r(\hat{x}_i^\alpha)$ at the points $\hat{x}_i^\alpha$ for $\alpha = 1, \ldots, N$ with the property

$$|r(x) - p(x)| = \mathcal{O}(\Delta x^{k+1}). \tag{6.54}$$

Further, we define the maximum preserving limiter $\widetilde{p}$ based on (6.32). Since r and p have the same values on the quadrature nodes, the scaling parameters $\widetilde{\theta}_p$ and $\widetilde{\theta}_r$ coincide.
We expand the difference between the limited and the original reconstruction

$$|\widetilde{r}(x) - r(x)| \leqslant |\widetilde{r}(x) - \widetilde{p}(x)| + |\widetilde{p}(x) - p(x)| + |p(x) - r(x)|. \tag{6.55}$$

By construction, we have $|p(x) - r(x)| = \mathcal{O}(\Delta x^{k+1})$ and from Lemma 6.3 we know

$|\widetilde{p}(x) - p(x)| = \mathcal{O}(\Delta x^{k+1})$. Since the scaling parameter $\widetilde{\theta}_r, \widetilde{\theta}_p$ coincides, we have

$$|\widetilde{r}(x) - \widetilde{p}(x)| = \widetilde{\theta}_r |r(x) - p(x)| = \mathcal{O}(\Delta x^{k+1}). \tag{6.56}$$

We conclude that $|\widetilde{r}(x) - r(x)| = \mathcal{O}(\Delta x^{k+1})$. □

6.4.4 General reconstruction on triangular grids

Zhang et al. [136] introduced a generalization of the maximum preserving limiter for triangular elements. Here, we change the method to define it for non-polynomial reconstructions. The idea is to define a quadrature rule on the triangle of the right order such that the weights are positive and all Gauss quadrature points on the interface are included. This quadrature rule is based on quadrature points on the square $[-\frac{1}{2}, \frac{1}{2}] \times [-\frac{1}{2}, \frac{1}{2}]$ defined as the product of the $k+1$ Gauss quadrature points $\{v^\beta |\ \beta = 1, \dots, k+1\}$ with its weights ω_β and the N Gauss-Lobatto quadrature points $\{\hat{u}^\alpha |\ \alpha = 1, \dots, N\}$ with its weights $\hat{\omega}_\alpha$ and $2N - 3 \geqslant k$. Thus, we have the quadrature points

$$S_k = \{(\hat{u}^\alpha, v^\beta) |\ \alpha = 1, \dots, k+1,\ \beta = 1, \dots N\}, \tag{6.57}$$

with the quadrature weights $\omega_\beta \hat{\omega}_\alpha$ on the square $[-\frac{1}{2}, \frac{1}{2}] \times [-\frac{1}{2}, \frac{1}{2}]$. Given the triangle C with the vertices $\mathbf{V}_1, \mathbf{V}_2, \mathbf{V}_3$ oriented clockwise, we define the projections

$$\mathbf{g}_1(u,v) = \Big(\frac{1}{2} + v\Big)\mathbf{V}_1 + \Big(\frac{1}{2} + u\Big)\Big(\frac{1}{2} - v\Big)\mathbf{V}_2 + \Big(\frac{1}{2} - u\Big)\Big(\frac{1}{2} - v\Big)\mathbf{V}_3, \tag{6.58}$$

$$\mathbf{g}_2(u,v) = \Big(\frac{1}{2} + v\Big)\mathbf{V}_2 + \Big(\frac{1}{2} + u\Big)\Big(\frac{1}{2} - v\Big)\mathbf{V}_3 + \Big(\frac{1}{2} - u\Big)\Big(\frac{1}{2} - v\Big)\mathbf{V}_1, \tag{6.59}$$

$$\mathbf{g}_3(u,v) = \Big(\frac{1}{2} + v\Big)\mathbf{V}_3 + \Big(\frac{1}{2} + u\Big)\Big(\frac{1}{2} - v\Big)\mathbf{V}_1 + \Big(\frac{1}{2} - u\Big)\Big(\frac{1}{2} - v\Big)\mathbf{V}_2, \tag{6.60}$$

from the square to the triangle C which map the top edge of the square to one vertex of the triangle, see Figure 6.7b. The following Lemma gives us the determinants of the gradient of the projections.

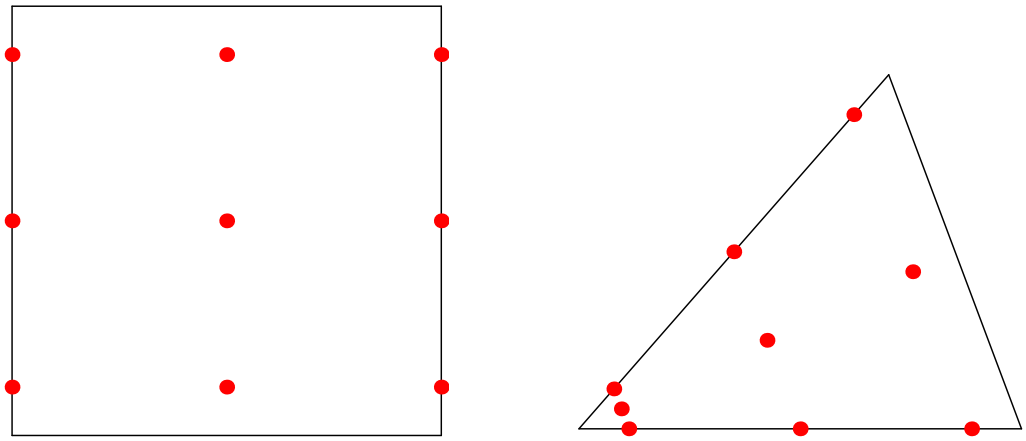
Lemma 6.5 (Jacobian of the projections [136])**.** *If the orientation of the three vertices* $\mathbf{V}_1$, $\mathbf{V}_2$ *and* $\mathbf{V}_3$ *is clockwise, then the Jacobian* $|\nabla \mathbf{g}_{l_e}(u,v)| = 2|C|(\frac{1}{2} - v)$ *for* $l_e = 1, 2, 3$.

Given the three different projections, we define the set of new quadrature nodes

$$S_k^C = \mathbf{g}_1(S_k) \cup \mathbf{g}_2(S_k) \cup \mathbf{g}_3(S_k), \tag{6.61}$$

which include all Gauss points on the cell boundary, e.g., Figure 6.8 for the case $k = 2$. Now, we can rewrite the cell average

$$U_C^n = \frac{1}{|C|}\int_C p_C(\mathbf{x})\mathrm{d}\mathbf{x} = \frac{1}{|C|}\int_{-1/2}^{1/2}\int_{-1/2}^{1/2} p_C(\mathbf{g}_{l_e}(u,v))|\nabla \mathbf{g}_{l_e}(u,v)|\mathrm{d}u\mathrm{d}v, \tag{6.62}$$



(a) Quadrature nodes on basic square.

(b) Quadrature nodes using $\mathbf{g}_1$.

Figure 6.7 – Construction of quadrature nodes for $k = 2$.

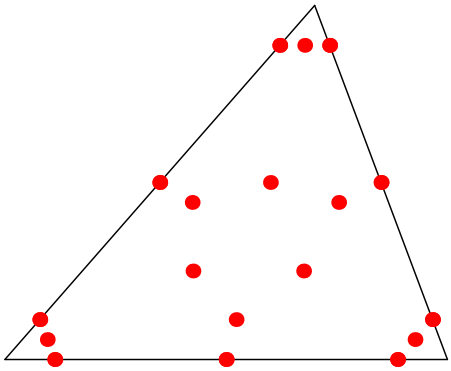


Figure 6.8 – Final set of quadrature nodes on the triangle C for $k = 2$.

for $p_C \in \Pi_k(\mathbb{R}^2)$ and $i = 1, 2, 3$. Thus, we can take the average over all $i = 1, 2, 3$

$$U_C^n = \frac{1}{3|C|} \sum_{l_e=1}^{3} \int_{-1/2}^{1/2} \int_{-1/2}^{1/2} p_C(\mathbf{g}_{l_e}(u,v)) |\nabla \mathbf{g}_{l_e}(u,v)| \mathrm{d}u\mathrm{d}v, \tag{6.63}$$

$$= \sum_{l_e=1}^{3} \sum_{\alpha=1}^{N} \sum_{\beta=1}^{k+1} p_C(\mathbf{g}_{l_e}(\hat{u}^\alpha, v^\beta)) \frac{2}{3} \Big(\frac{1}{2} - v^\beta \Big) \omega_\alpha \hat{\omega}_\beta = \sum_{\mathbf{x} \in S_k^C} p_C(\mathbf{x}) \omega_{\mathbf{x}}, \tag{6.64}$$

using the result of Lemma 6.5 with the quadrature weights $\omega_{\mathbf{x}}$ for each $\mathbf{x} \in S_k^C$. We define the set of quadrature points in the interior $S_k^{C,int}$ and the set of quadrature points on the edges $S_k^{C,edg}$. Note that in (6.64) each quadrature node on the edge is counted double with $\hat{\omega}_1 = \hat{\omega}_N$. We obtain

$$S_k^{C,edg} = \{\mathbf{x}_{1,\beta}, \mathbf{x}_{2,\beta}, \mathbf{x}_{3,\beta} |\ \beta = 1, \ldots, k+1\}, \tag{6.65}$$

with $\mathbf{x}_{1,\beta} = (0, \frac{1}{2} + v^\beta, \frac{1}{2} - v^\beta)$, $\mathbf{x}_{2,\beta} = (\frac{1}{2} - v^\beta, 0, \frac{1}{2} + v^\beta)$, $\mathbf{x}_{1,\beta} = (\frac{1}{2} + v^\beta, \frac{1}{2} - v^\beta, 0)$ written

in terms of the barycentric coordinates (ξ_1, ξ_2, ξ_3), such that $\mathbf{p} = \xi_1\mathbf{V}_1 + \xi_2\mathbf{V}_2 + \xi_3\mathbf{V}_3$. To calculate the weights on the edge $(0, \frac{1}{2}+v^\beta, \frac{1}{2}-v^\beta)$ we use that $\mathbf{g}_2(\frac{1}{2}, v^\beta) = \mathbf{g}_3(-\frac{1}{2}, -v^\beta)$ and we recover

$$\frac{2}{3}\Big(\frac{1}{2}+v^\beta\Big)\omega_\beta\hat{\omega}_1 + \frac{2}{3}\Big(\frac{1}{2}-v^\beta\Big)\omega_\beta\hat{\omega}_N = \frac{2}{3}\omega_\beta\hat{\omega}_1, \tag{6.66}$$

for the weights on the edges for the quadrature node $\mathbf{x}_{1,\beta}$. The same result can be obtained for the other edges. Analogous to the one-dimensional case, we have

$$U_C^n = p_R\omega_R + \sum_{\beta=1}^{k+1}\sum_{l_e=1}^{3}\frac{2}{3}\omega_\beta\hat{\omega}_1 u_{l_e,\beta}^C, \tag{6.67}$$

with the evaluation of the quadrature node on the l_eth edge $u_{l_e,\beta}^C = p_C(\mathbf{x}_{l_e,\beta})$,

$$\omega_R = 1 - \sum_{\beta=1}^{k+1}\sum_{l_e=1}^{3}\frac{2}{3}\omega_\beta\hat{\omega}_1, \tag{6.68}$$

and

$$p_R = \sum_{\mathbf{x}\in S_k^{C,int}} p_C(\mathbf{x})\frac{\omega_\mathbf{x}}{\omega_R} = \frac{U_C}{\omega_R} - \sum_{\beta=1}^{k+1}\sum_{l_e=1}^{3}\frac{2}{3}\frac{\omega_\beta\hat{\omega}_1}{\omega_R}u_{l_e,\beta}^C. \tag{6.69}$$

Let us rewrite the finite volume scheme (2.42) with the high order flux (2.47) and the forward Euler method in time, i.e.,

$$U_i^{n+1} = U_i^n - \frac{\Delta t}{|C_i|}\sum_{\beta=1}^{k+1}\omega_\beta\sum_{l_e=1}^{3}F_{il_e}(u_{l_e,\beta}^{C_i}, u_{l_e,\beta}^{C_{il_e}}, \mathbf{n}_{il_e}). \tag{6.70}$$

The proof of the maximum principle is based on the idea that the first order method

$$U_i^{n+1} = U_i^n - \lambda\sum_{l_e=1}^{3}F_{il_e}(U_i^n, U_{il_e}^n, \mathbf{n}_{il_e}), \tag{6.71}$$

is non-decreasing under the CFL condition

$$\max_{u,\mathbf{n}}|\nabla_u(f(u)\cdot\mathbf{n})|\lambda\sum_{l_e=1}^{3}|S_{il_e}| \leqslant 1, \tag{6.72}$$

which is satisfied for a monotone flux, e.g., the Rusanov flux (2.44).

Theorem 6.6 (Maximum principle satisfying scheme for triangular methods)**.** *Let us consider a first order finite volume method of the form* (6.71) *that is non-decreasing under the condition* (6.72).

The scheme (6.70) *satisfies the maximum principle*

$$m \leqslant U_{C_i}^{n+1} \leqslant M, \tag{6.73}$$

under the condition that

$$m \leqslant u_{l_e,\beta}^{C_i}, p_R \leqslant M, \qquad \textit{for all } l_e = 1,2,3, \beta = 1,\ldots,k+1, \tag{6.74}$$

and the additional CFL condition

$$\max_{u,\mathbf{n}} |\nabla_u (f(u)\cdot\mathbf{n})| \frac{\Delta t}{|C_i|} \sum_{l_e=1}^{3} |S_{il_e}| \leqslant \frac{2}{3}\hat{\omega}_1, \tag{6.75}$$

Proof. The proof follows the one in [136] with the difference that we use p_R defined in (6.69). Let us decompose the flux (6.70)

$$\begin{aligned}\sum_{l_e=1}^{3} F_{il_e}(u_{i,\beta}^{C_i}, u_{i,\beta}^{C_{il_e}}, \mathbf{n}_{il_e}) &= F_{i1}(u_{1,\beta}^{C_i}, u_{1,\beta}^{C_{i1}}, \mathbf{n}_{i1}) + F_{i1}(u_{1,\beta}^{C_i}, u_{2,\beta}^{C_i}, -\mathbf{n}_{i1}) \\ &+ F_{i1}(u_{2,\beta}^{C_i}, u_{1,\beta}^{C_i}, \mathbf{n}_{i1}) + F_{i2}(u_{2,\beta}^{C_i}, u_{2,\beta}^{C_{i2}}, \mathbf{n}_{i2}) + F_{i3}(u_{2,\beta}^{C_i}, u_{3,\beta}^{C_i}, \mathbf{n}_{i3}) \\ &+ F_{i3}(u_{3,\beta}^{C_i}, u_{2,\beta}^{C_i}, -\mathbf{n}_{i3}) + F_{i3}(u_{3,\beta}^{C_i}, u_{3,\beta}^{C_{i3}}, \mathbf{n}_{i3}),\end{aligned} \tag{6.76}$$

using the conservation of the flux. Next, we combine (6.70) with (6.67) and (6.76) and obtain

$$\begin{aligned}U_i^{n+1} &= p_R\omega_R + \sum_{\beta=1}^{k+1}\sum_{l_e=1}^{3} \frac{2}{3}\omega_\beta\hat{\omega}_1 u_{l_e,\beta}^{C_{l_e}} - \frac{\Delta t}{|C_i|}\sum_{\beta=1}^{k+1}\omega_\beta \sum_{l_e=1}^{3} F_{il_e}(u_{l_e,\beta}^{C_i}, u_{l_e,\beta}^{C_{il_e}}, \mathbf{n}_{il_e}), \\ &= p_R\omega_R + \sum_{\beta=1}^{k+1}\frac{2}{3}\omega_\beta\hat{\omega}_1[H_{1,\beta} + H_{2,\beta} + H_{3,\beta}],\end{aligned}$$

with

$$\begin{aligned}H_{1,\beta} &= u_{1,\beta}^{C_i} - \frac{3\Delta t}{2\hat{\omega}_1|C_i|}[F_{i1}(u_{1,\beta}^{C_i}, u_{1,\beta}^{C_{i1}}, \mathbf{n}_{i1}) + F_{i1}(u_{1,\beta}^{C_i}, u_{2,\beta}^{C_i}, -\mathbf{n}_{i1})], \\ H_{2,\beta} &= u_{2,\beta}^{C_i} - \frac{3\Delta t}{2\hat{\omega}_1|C_i|}[F_{i1}(u_{2,\beta}^{C_i}, u_{1,\beta}^{C_i}, \mathbf{n}_{i1}) + F_{i2}(u_{2,\beta}^{C_i}, u_{2,\beta}^{C_{i2}}, \mathbf{n}_{i2}) \\ &\quad + F_{i3}(u_{2,\beta}^{C_i}, u_{3,\beta}^{C_i}, \mathbf{n}_{i3})], \\ H_{3,\beta} &= u_{3,\beta}^{C_i} - \frac{3\Delta t}{2\hat{\omega}_1|C_i|}[F_{i3}(u_{3,\beta}^{C_i}, u_{2,\beta}^{C_i}, -\mathbf{n}_{i3}) + F_{i3}(u_{3,\beta}^{C_i}, u_{3,\beta}^{C_{i3}}, \mathbf{n}_{i3})].\end{aligned}$$

Under the assumption that the first order method (6.71) is non-decreasing in each argument under the CFL condition (6.72), we have that each $H_{l_e,\beta}$ is non-decreasing under (6.75). Finally, we combine this with (6.74) and obtain the maximum principle for high-order methods on triangular grids. □

So far, we have only dealt with the polynomial case. However, the results are also true for the non-polynomial case with the definition

$$p_R = \frac{U_C}{\omega_R} - \sum_{\beta=1}^{k+1} \sum_{l_e=1}^{3} \frac{2}{3} \frac{\omega_\beta \hat{\omega}_1}{\omega_R} u^C_{l_e,\beta}. \tag{6.77}$$

Maximum principle satisfying limiter on triangular meshes

Let us consider a general reconstruction $r_{C_i} : \mathbb{R}^2 \to \mathbb{R}$ for the solution in the cell C_i. In the case that the reconstruction r_{C_i} does not satisfy (6.74), we can modify it in the same way as in one dimension. We define

$$\widetilde{u}_{\min} = \min\{p_R, u^{C_i}_{l_e,\beta} |\ l_e = 1,2,3, \beta = 1,\dots,k+1\}, \tag{6.78}$$

$$\widetilde{u}_{\max} = \max\{p_R, u^{C_i}_{l_e,\beta} |\ l_e = 1,2,3, \beta = 1,\dots,k+1\}, \tag{6.79}$$

$$\widetilde{\theta} = \min\Big\{\Big|\frac{U_i^n - m}{U_i^n - \widetilde{u}_{\min}}\Big|, \Big|\frac{U_i^n - M}{U_i^n - \widetilde{u}_{\max}}\Big|, 1\Big\}, \tag{6.80}$$

$$\widetilde{r}_{C_i}(\mathbf{x}) = \widetilde{\theta}\big(r_{C_i}(\mathbf{x}) - U_i^n\big) + U_i^n, \tag{6.81}$$

with $u^{C_i}_{l_e,\beta} = r_{C_i}(\mathbf{x}_{l_e,\beta})$. The results from one dimension can be directly transfered to two dimensions and are summarized in the following Lemma.

Lemma 6.7. *The maximum principle preserving limiter* (6.81) *with*

$$\widetilde{u}_{\min} = \min\{p_R, u^{C_i}_{l_e,\beta} |\ l_e = 1,2,3, \beta = 1,\dots,k+1\},$$
$$\widetilde{u}_{\max} = \max\{p_R, u^{C_i}_{l_e,\beta} |\ l_e = 1,2,3, \beta = 1,\dots,k+1\},$$
$$\widetilde{\theta} = \min\Big\{\Big|\frac{U_i^n - m}{U_i^n - \widetilde{u}_{\min}}\Big|, \Big|\frac{U_i^n - M}{U_i^n - \widetilde{u}_{\max}}\Big|, 1\Big\},$$

is conservative, of high order and satisfies the maximum preserving condition (6.74).

6.4.5 High-order positivity preserving scheme for the Euler equations

The maximum principle does not hold anymore for systems of equations. However, to solve the Euler equations we need to ensure positivity of the density and pressure. Let us consider a conservation law with the flux (2.9) and the pressure

$$p = \mathcal{R}\rho T = (\gamma - 1)\Big(E - \frac{1}{2}\frac{m_1^2 + m_2^2}{\rho}\Big). \tag{6.82}$$

The method is based on a positivity preserving first order method (6.70), e.g., using the Rusanov flux (2.44). Further, we use that the pressure p is concave with respect to ρ,

m_1, m_2 and E under the condition $\rho > 0$. Thus, the set of admissible states

$$G = \Big\{ (\rho, m_1, m_2, E)^T \Big| \ \rho > 0, p > 0 \Big\}, \tag{6.83}$$

is convex. We denote the cell average values at time t_n as $\mathbf{Q}_C^n = (\bar{\rho}_C^n, \bar{m}_{1,C}^n, \bar{m}_{2,C}^n, \bar{E}_C^n)^T$ and the high-order reconstructions in the cell C as

$$\mathbf{q}_C(\mathbf{x}) = (\rho_C(\mathbf{x}), m_{1C}(\mathbf{x}), m_{2C}(\mathbf{x}), E_C(\mathbf{x}))^T. \tag{6.84}$$

To preserve positivity of the density we proceed in the same way as to preserve the maximum. We define the limiter

$$\widetilde{\rho}_{\min} = \min\{\rho_R, \rho_{l_e,\beta}^C |\ l_e = 1,2,3, \beta = 1,\dots,k+1\}, \tag{6.85}$$

$$\widetilde{\theta}_1 = \min\Big\{ \Big| \frac{\bar{\rho}_i^n - \varepsilon}{\bar{\rho}_i^n - \widetilde{\rho}_{\min}} \Big|, 1 \Big\}, \tag{6.86}$$

$$\widetilde{\rho}_C(\mathbf{x}) = \widetilde{\theta}_1 \big(\rho_C(\mathbf{x}) - \bar{\rho}_i^n \big) + \bar{\rho}_i^n, \tag{6.87}$$

with the small threshold $\varepsilon > 0$, and set

$$\widetilde{\mathbf{q}}_C(\mathbf{x}) = (\widetilde{\rho}_C(\mathbf{x}), m_{1C}(\mathbf{x}), m_{2C}(\mathbf{x}), E_C(\mathbf{x}))^T. \tag{6.88}$$

To preserve positivity of the pressure p we define the function

$$\mathbf{s}_{\mathbf{x}}(t) = (1-t)\mathbf{Q}_C^n + t\widetilde{\mathbf{q}}_C(\mathbf{x}), \tag{6.89}$$

and

$$t(\mathbf{x}) = \begin{cases} 1, & \text{if } p(\widetilde{\mathbf{q}}_C(\mathbf{x})) \geqslant \varepsilon, \\ t_0 \text{ such that } p(\mathbf{s}_{\mathbf{x}}(t_0)) = \varepsilon, & \text{if } p(\widetilde{\mathbf{q}}_C(\mathbf{x})) < \varepsilon. \end{cases} \tag{6.90}$$

Further, we define the remainder $\widetilde{\mathbf{q}}_R = \mathbf{Q}_C^n - \sum_{\beta=1}^{k+1} \sum_{l_e=1}^{3} \frac{2}{3} \frac{\omega_\beta \hat{\omega}_1}{\omega_R} \widetilde{\mathbf{q}}_C(\mathbf{x}_{l_e,\beta})$

$$\mathbf{s}_R(t) = (1-t)\mathbf{Q}_C^n + t\widetilde{\mathbf{q}}_R, \tag{6.91}$$

and

$$t_R = \begin{cases} 1, & \text{if } p(\widetilde{\mathbf{q}}_R) \geqslant \varepsilon, \\ t_0 \text{ such that } p(s_R(t_0)) = \varepsilon, & \text{if } p(\widetilde{\mathbf{q}}_R) < \varepsilon. \end{cases} \tag{6.92}$$

This allows us to define the new vector of reconstruction functions

$$\widetilde{\widetilde{\mathbf{q}}}_C(\mathbf{x}) = \theta_2(\widetilde{\mathbf{q}}_C(\mathbf{x}) - \mathbf{Q}_C^n) + \mathbf{Q}_C^n, \tag{6.93}$$

$$\theta_2 = \min\left\{ \min_{\mathbf{x} \in S_k^{C,edg}} t(\mathbf{x}), t_R \right\}. \tag{6.94}$$

We have the following lemma.

Lemma 6.8. *Given the limiter* (6.87) *and* (6.93) *the reconstruction* $\widetilde{\widetilde{\mathbf{q}}}_C$ *is of high-order accuracy, conservative and preserves positivity of the density and pressure.*

Proof. For the first step, using the limiter of the density we can directly take the results from the maximum preserving limiter. Also the positivity of the pressure and the conservation property of the second limiter (6.93) are clear. The only open question is the high-order accuracy of the second step.
Let us keep in mind that the original limiter by Zhang and Shu [134] is based on the minimum over all quadrature nodes

$$\hat{\theta}_2 = \min_{\mathbf{x}\in S_k^C} t(\mathbf{x}). \tag{6.95}$$

We define the vector of polynomials $\mathbf{p}_C$ such that $\mathbf{p}_C(\mathbf{x}) = \widetilde{\widetilde{\mathbf{q}}}_C(\mathbf{x})$ for all $\mathbf{x} \in S_k^C$. Thus, the values of θ_2 and $\hat{\theta}_2$ are the same for the polynomial reconstruction $\mathbf{p}_C$ and the non-polynomial reconstruction $\widetilde{\widetilde{\mathbf{q}}}_C$. Furthermore, we know

$$\widetilde{\mathbf{q}}_R = \mathbf{Q}_C^n - \sum_{\beta=1}^{k+1}\sum_{i=1}^{3} \frac{2}{3}\frac{\omega_\beta \hat{\omega}_1}{\omega_R}\widetilde{\mathbf{q}}_C(\mathbf{x}_{i,\beta}) = \mathbf{Q}_C^n - \sum_{\beta=1}^{k+1}\sum_{i=1}^{3} \frac{2}{3}\frac{\omega_\beta \hat{\omega}_1}{\omega_R}\mathbf{p}_C(\mathbf{x}_{i,\beta}), \tag{6.96}$$

$$= \sum_{\mathbf{x}\in S_k^{C,int}} \frac{\omega_x}{\omega_R}\mathbf{p}_C(\mathbf{x}), \tag{6.97}$$

which is a convex combination of the values $\mathbf{p}_C(\mathbf{x})$ for $\mathbf{x} \in S_k^{C,int}$. Thus, we obtain

$$p((1-\hat{\theta}_2)\mathbf{Q}_C^n + \hat{\theta}_2\widetilde{\mathbf{q}}_R) = p\Big((1-\hat{\theta}_2)\mathbf{Q}_C^n + \hat{\theta}_2 \sum_{\mathbf{x}\in S_k^{C,int}} \frac{\omega_x}{\omega_R}\mathbf{p}_C(\mathbf{x})\Big), \tag{6.98}$$

$$= p\Big(\sum_{\mathbf{x}\in S_k^{C,int}} \frac{\omega_x}{\omega_R}((1-\hat{\theta}_2)\mathbf{Q}_C^n + \hat{\theta}_2\mathbf{p}_C(\mathbf{x}))\Big), \tag{6.99}$$

$$\geqslant \sum_{\mathbf{x}\in S_k^{C,int}} \frac{\omega_x}{\omega_R} p\Big((1-\hat{\theta}_2)\mathbf{Q}_C^n + \hat{\theta}_2\mathbf{p}_C(\mathbf{x})\Big) \geqslant \varepsilon, \tag{6.100}$$

since p is concave. We conclude that $\theta_2 \geqslant \hat{\theta}_2$. Finally, we use the estimate in (6.55)

$$|\widetilde{\mathbf{q}}_C(\mathbf{x}) - \mathbf{q}_C(\mathbf{x})| \leqslant |\widetilde{\mathbf{q}}_C(\mathbf{x}) - \widetilde{\mathbf{p}}_C(\mathbf{x}))| + |\widetilde{\mathbf{p}}_C(\mathbf{x})) - \mathbf{p}_C(\mathbf{x}))| + |\mathbf{p}_C(\mathbf{x})) - \mathbf{q}_C(\mathbf{x})|,$$

with

$$|\mathbf{q}_C(\mathbf{x}) - \mathbf{p}_C(\mathbf{x}))| = \mathcal{O}(\Delta\mathbf{x}^{k+1}), \tag{6.101}$$
$$|\tilde{\mathbf{q}}_C(\mathbf{x}) - \tilde{\mathbf{p}}_C(\mathbf{x}))| = \theta_2|\mathbf{q}_C(\mathbf{x}) - \mathbf{p}_C(\mathbf{x}))| = \mathcal{O}(\Delta\mathbf{x}^{k+1}), \tag{6.102}$$
$$|\tilde{\mathbf{p}}_C(\mathbf{x}) - \mathbf{p}_C(\mathbf{x}))| = |1-\theta_2||\mathbf{p}_C(\mathbf{x})) - \mathbf{Q}_C^n))| \leqslant |1-\hat{\theta}_2||\mathbf{p}_C(\mathbf{x})) - \mathbf{Q}_C^n))|, \tag{6.103}$$
$$= |\hat{\mathbf{p}}_C(\mathbf{x}) - \mathbf{p}_C(\mathbf{x}))| = \mathcal{O}(\Delta\mathbf{x}^{k+1}), \tag{6.104}$$

with the original limiter $\hat{\mathbf{p}}_C$ based on $\hat{\theta}_2$ from [136]. □

Remark 6.3. *The positivity preserving limiter works exactly the same for the one-dimensional schemes with* $S_k^{C_i,edg} = \{x_{i-1/2}, x_{i+1/2}\}$.

6.5 Numerical results for one-dimensional problems

Let us take a look at some one-dimensional examples. We verify the order of convergence and show the ability to deal with challenging one-dimensional examples.

6.5.1 Linear advection equation

To confirm the order of convergence given in Theorem 6.1, we consider the linear advection equation (2.3). We assume periodic boundary conditions on the domain $[-1,1]$ and a wave speed $a = 1$.
Next, we consider two different hybrid grids. The structured hybrid grid consists of N cells split equally into two grids $\{x_0,\dots,x_{N/2}\}$ and $\{x_{N/2},\dots,x_N\}$ with $x_i = \frac{2i}{N} - 1$. The unstructured hybrid grid consists of the unstructured part $\{\tilde{x}_0,\dots,\tilde{x}_{N/2}\}$ with $\tilde{x}_i = x_i + \varepsilon_i$, $\varepsilon_i \in \mathcal{U}(-\frac{0.1}{N}, \frac{0.1}{N})$ uniformly distributed between $[-\frac{0.1}{N}, \frac{0.1}{N}]$ and the structured one $\{x_{N/2},\dots,x_N\}$.
The convergence of the hybrid method is generally as expected, see Table 6.1. We compare the accuracy using the hybrid method with $p_{\text{WENO}} = 2\left\lfloor\frac{p}{2}\right\rfloor + 1$ and $p_{\text{WENO}} = 2p-1$. For the 3rd order method we observe a reduced error by around a factor 10 in the case $p_{\text{WENO}} = 2p-1$. Table 6.2 shows the runtime for the different 3rd order methods. The hybrid methods with $p_{\text{WENO}} = 2\left\lfloor\frac{p}{2}\right\rfloor + 1$ and $p_{\text{WENO}} = 2p-1$ have a similar computational complexity and they are a bit faster than the RBF-ENO method. Note that the costs of the stencil selection in the RBF-ENO method in one space dimension is not much more expensive than the WENO method. However, the cost of the two-dimensional stencil selection algorithm is quadratic in the size of the stencil. Comparing these results with the ones from Table 3.2 we should keep in mind that the methods are different.

N		Unstructured grid						Structured grid	
		Hybrid $2p-1$		Hybrid $2\lfloor p/2 \rfloor + 1$		RBF-ENO p		Hybrid $2p-1$	
		error	rate	error	rate	error	rate	error	rate
$p=3$	16	2.32e-03	-	1.79e-02	-	2.99e-03	-	2.54e-03	-
	32	2.46e-04	3.25	3.90e-03	2.16	5.25e-04	2.55	5.63e-04	2.17
	64	6.07e-05	1.95	9.70e-04	2.01	8.20e-05	2.61	4.69e-05	3.58
	128	7.78e-06	3.16	2.26e-04	2.09	1.28e-05	2.68	5.88e-06	2.99
	256	9.38e-07	2.86	3.08e-05	2.84	1.61e-06	3.00	9.10e-07	2.69
	512	1.44e-07	2.7	2.14e-06	3.85	2.64e-07	2.60	1.26e-07	2.85
$p=4$	16	2.41e-04	-	6.49e-04	-	9.77e-04	-	4.96e-04	-
	32	3.86e-05	2.69	5.37e-05	3.63	6.12e-05	4.09	4.52e-05	3.46
	64	2.71e-06	3.71	2.58e-06	4.31	5.36e-06	3.47	2.20e-06	4.36
	128	1.47e-07	4.21	2.06e-07	3.66	4.31e-07	3.57	1.70e-07	3.69
	256	1.70e-08	3.1	1.57e-08	3.71	2.89e-08	3.89	1.66e-08	3.35
	512	1.48e-09	3.53	1.22e-09	3.68	2.97e-09	3.27	1.51e-09	3.47
$p=5$	16	1.15e-04	-	3.71e-04	-	2.38e-04	-	2.43e-04	-
	32	6.04e-06	4.10	1.75e-05	4.60	9.96e-06	4.66	7.12e-06	5.09
	64	1.71e-07	5.16	5.98e-07	4.32	2.96e-07	4.93	2.62e-07	4.76
	128	8.19e-09	4.33	1.90e-08	4.50	1.34e-08	4.43	7.19e-09	5.19
	256	4.76e-10	4.14	6.70e-10	4.72	7.36e-10	4.16	3.23e-10	4.45
	512	1.52e-11	4.96	3.18e-11	4.22	3.98e-11	4.21	2.83e-11	4.54

Table 6.1 – Convergence rates of the Hybrid ENO method for different grid sizes compared with the RBF-ENO method for the linear advection equation on $[-1,1]$ at time $T = 0.1$. We use periodic boundary conditions and $u_0(x) = \sin(\pi x)$, CFL $= 0.5$.

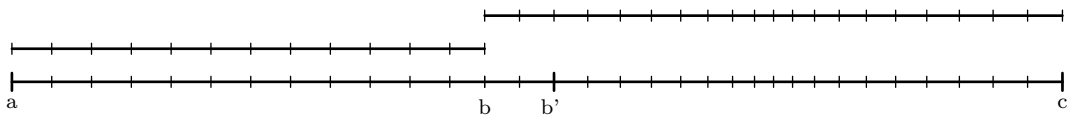


Figure 6.9 – 1D hybrid grid for Euler equations .

6.5.2 Euler equations

Next, we present numerical results for the one-dimensional Euler equations (3.96) with the ratio of specific heat $\gamma = 1.4$. We test the behavior of the hybrid RBF-ENO method with shocks, contact discontinuities, and rarefaction waves. The computational grid for the Euler equations is shown in Figure 6.9. The idea is to run it on the left half with the structured WENO method and on the right one with a continuously refined grid such that the middle cell is half the size of the outer ones. Note that we have to change the two original meshes in $[a,b']$ and $[b',c]$ by adding the n_{ghost} last cells of the structured mesh to the unstructured one. The following results are based on a grid with $N_1 = 78$ cells in $[a,b]$ and $N_2 = 101$ cells in $[b,c]$. Further, the reconstruction is performed in the characteristic variables $\mathbf{V} = R^{-1}\mathbf{U}$, with the eigenvectors R from (2.13).

	Hybrid $2p-1$	Hybrid $2\lfloor p/2 \rfloor + 1$	RBF-ENO p
16	2.7	1.4	0.8
32	4.4	2.5	1.6
64	4.6	3.2	3.2
128	8.4	7.4	8.4
256	18.8	15.8	19.8
512	45.5	44.0	62.6

Table 6.2 – Runtime comparison for the 3rd order methods solving the linear advection equation.

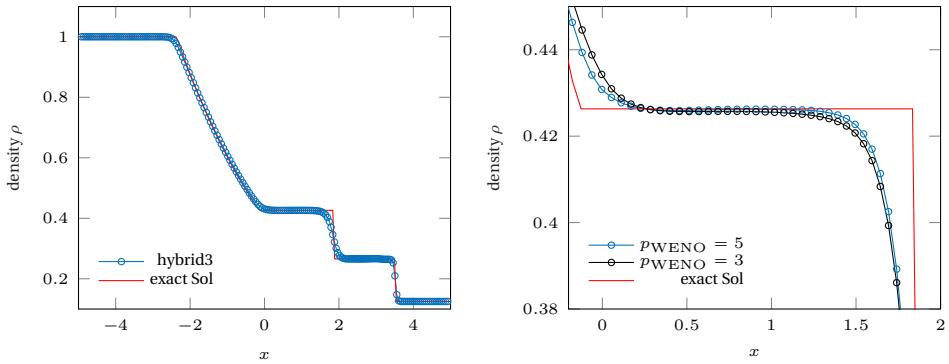


Figure 6.10 – Sod's shock tube problem on $[-5,5]$ at time $T = 2$ with $\text{CFL} = 0.8$ solved by the hybrid RBF-ENO method of order 3.

Sod's shock tube problem

We consider Sod's shock tube problem based on the initial conditions (3.97) on the domain $[-5,5]$. Figure 6.10 shows the results for the hybrid RBF-ENO method of order 3. We observe that the rarefaction wave, the contact discontinuity, and the shock are well resolved. Furthermore, it is clear that the 3rd order method with $p_{\text{WENO}} = 5$ resolves the rarefaction wave and the contact discontinuity better. In the case of the 5th order method the differences between $p_{\text{WENO}} = 5$ and $p_{\text{WENO}} = 9$ are not obvious anymore, Figure 6.11. However, we observe the increased resolution of the solution from the 5th order method compared to the 3rd order scheme.

Lax shock tube problem

The Lax shock tube problem is based on the initial conditions (3.98) and solved on the domain $[-5,5]$. From now on, we use the hybrid RBF-ENO method with $p_{\text{WENO}} = 2p-1$. The 3rd and 5th order method present the contact discontinuity and the shock with a high resolution, Figure 6.12. Furthermore, we observe in Figure 6.13 the superior behavior of the method of order 5 to the one of order 3 and 4.

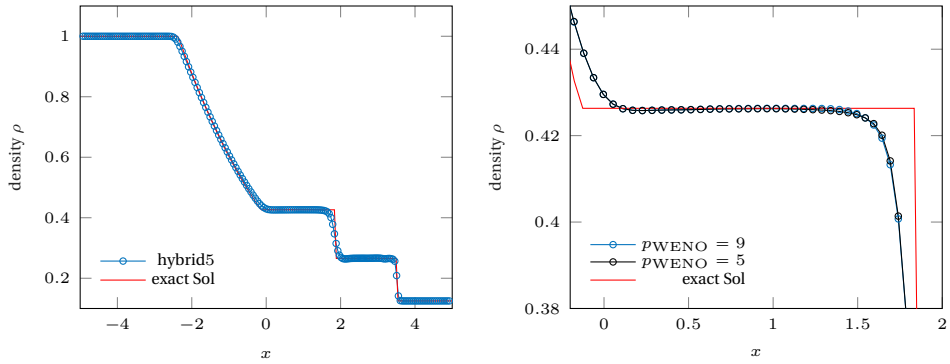


Figure 6.11 – Sod's shock tube problem on $[-5, 5]$ at time $T = 2$ with $\mathrm{CFL} = 0.8$ solved by the hybrid RBF-ENO method of order 5.

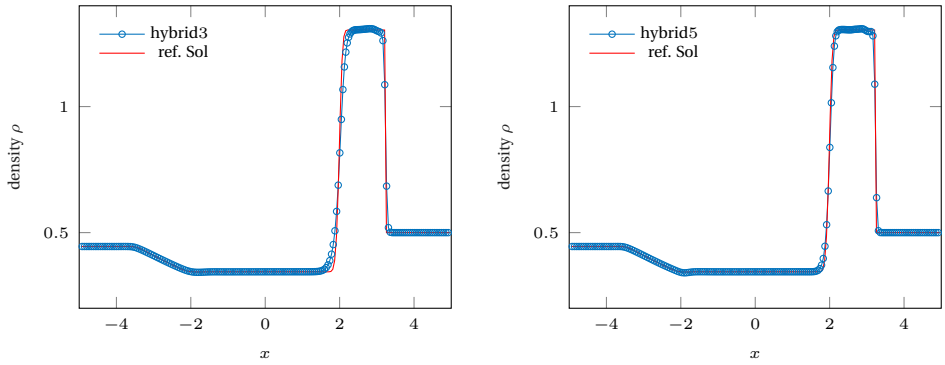


Figure 6.12 – Lax shock tube problem on $[-5, 5]$ at time $T = 1.3$ with $\mathrm{CFL} = 0.8$ solved by the hybrid RBF-ENO method of order 3 and 5 with $p_{\mathrm{WENO}} = 2p - 1$.

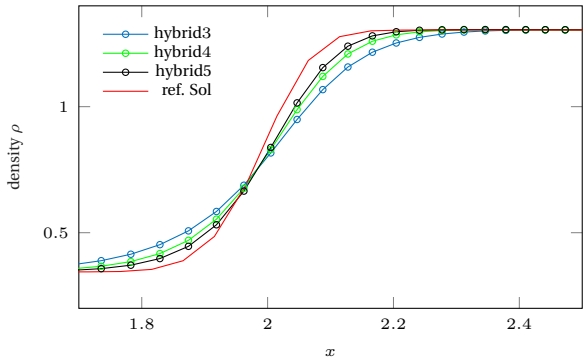


Figure 6.13 – Comparison of the accuracy of the hybrid RBF-ENO method of order 3, 4 and 5 at the contact discontinuity of the Lax shock tube problem on $[-5, 5]$ at time $T = 1.3$ for with $\mathrm{CFL} = 0.8$ with $p_{\mathrm{WENO}} = 2p - 1$.

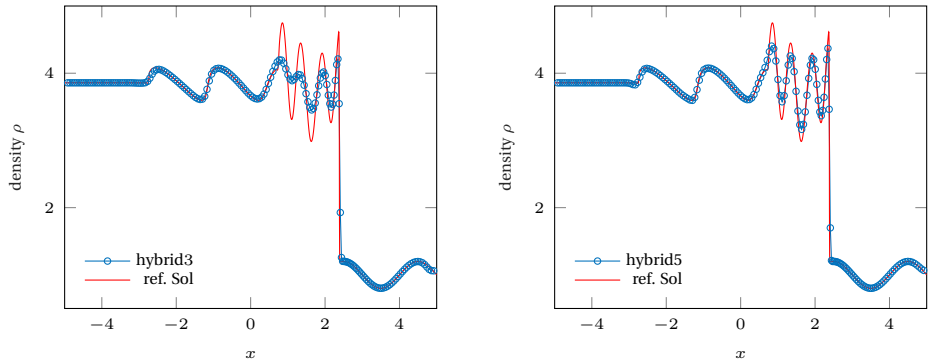


Figure 6.14 – Shu-Osher problem on $[-5,5]$ at time $T = 1.8$ with CFL $= 0.8$ solved by the hybrid RBF-ENO method of order 3 and 5 with $p_{\text{WENO}} = 2p-1$.

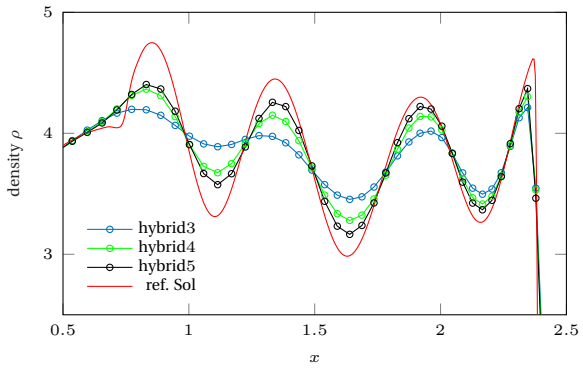


Figure 6.15 – Comparison of the hybrid RBF-ENO method of order 3, 4 and 5 for the Shu-Osher problem on $[-5,5]$ at time $T = 1.8$ with CFL $= 0.8$ with $p_{\text{WENO}} = 2p-1$.

Shu-Osher shock-entropy wave interaction problem

We consider the Shu-Osher shock-entropy wave interaction problem. This Riemann problem has the initial conditions (3.99) and the computational domain $[-5,5]$. As before, we obtain the correct solution with the 3rd and 5th order method, Figure 6.14. In this example, we see a substantial advantage of the high-order methods. There are evident differences in the resolution of the waves, Figure 6.15.

Two interacting blast waves

As the last one-dimensional example, we test the method on the two interacting blast waves based on the initial conditions (3.101). In Chapter 3, we had already some difficulties to solve this problem with the RBF-TeCNOp method. There, we introduced a new symmetric positive definite dissipation operator (2.87), which mimics the more dissipative Rusanov-type diffusion operator. Here, we calculate the two interacting blast waves with the hybrid RBF-ENO method of order 5 based on the same grid as

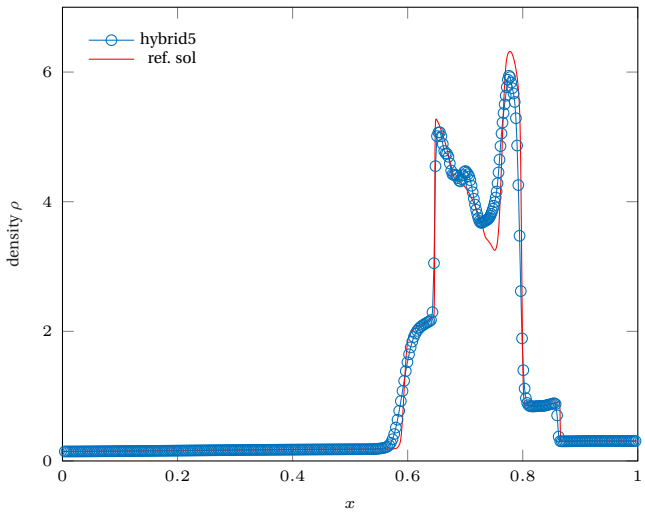


Figure 6.16 – WC blast wave problem on $[0,1]$ at time $T = 0.038$ with $N = 200$, CFL $= 0.5$ solved by the hybrid RBF-ENO method of order 5.

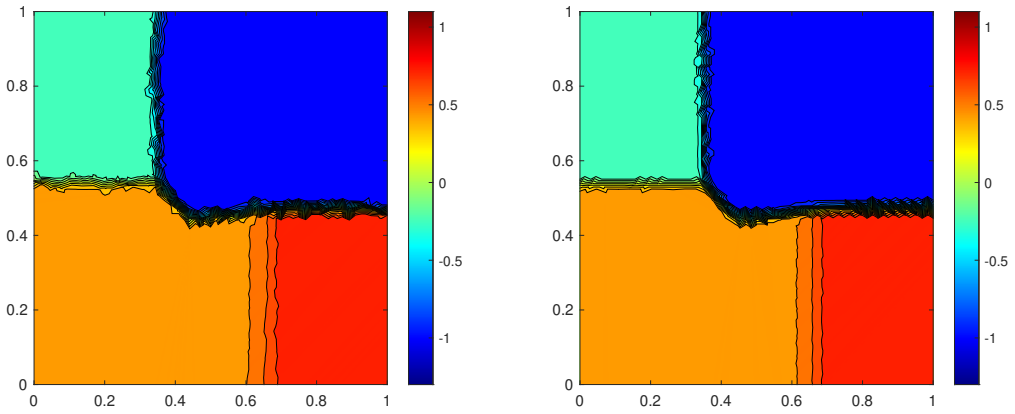
before with $N_1 = 158$ and $N_2 = 205$. If we use the original version, we obtain negative density or pressure. By using the positivity preserving limiter from Section 6.4, we can stabilize the method and calculate the solution at time $T = 0.038$. The results of the fifth order hybrid RBF-ENO method combined with the positivity preserving limiter are shown in Figure 6.16.

6.6 Numerical results for two-dimensional problems

In this section, we demonstrate the hybrid RBF-ENO method with all its features on a couple of numerical examples. First, we solve Burgers' equation to compare the complexity of the hybrid and the non-hybrid method. To show the robustness of the method in two dimensions, we conclude with several numerical examples of the Euler equations (2.9) in complex geometries. We start with some already known examples to show that the solutions are comparable to the ones generated with the other methods. We conclude the section with the simulation of the scramjet and a model of a conical aerospike nozzle. The grids in this section are generated using Gmsh [46] and to specify the grids we introduce the number of triangular cells N_{TRI} and the number of quadrilateral cells N_{QUAD}.

6.6.1 Burgers' equation

With the Burgers' equation we demonstrate the difference in the computational cost of the hybrid and the non-hybrid RBF-ENO method. We assume Burgers' equation (4.47)



(a) Hybrid RBF-ENO of order 3.

(b) RBF-ENO of order 3.

Figure 6.17 – Solution of the Burgers' equation at $T = 0.25$ with $N_{\mathrm{TRI}} = 2390$ cells, $\mathrm{CFL} = 0.8$.

with initial condition (4.48) on the extended domain $[-1,2] \times [-1,2]$, see Figure 4.4. The hybrid method is based on the grid from Figure 6.5 and the non-hybrid scheme is based on a uniform triangulation. We compare the computational cost depending on the number of triangles N_{TRI} in the target area $[0,1] \times [0,1]$. Note that the triangulation for the non-hybrid method has in total around nine times the number of cells from the target area. The hybrid method has around one-ninth of triangular cells plus the quadrilateral cells around them. Thus, the upper bound for the speed-up is nine. From Table 6.3, we get a speed-up of around 7.2 for a fine enough grid. Compared to the

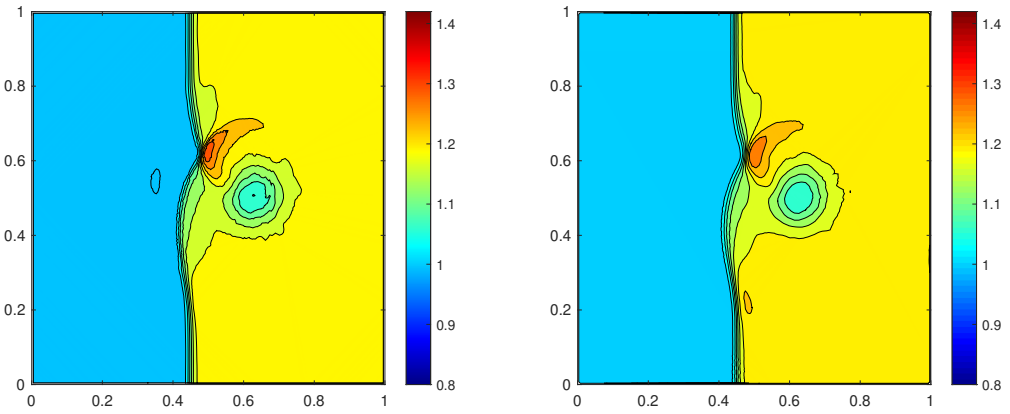
N_{TRI}	Hybrid RBF-ENO	RBF-ENO	Speed-up
155	24	96	4
610	143	810	5.6
1348	411	2925	7.1
2390	947	6840	7.2

Table 6.3 – Runtime comparison for the 3th order methods solving the 2D Burgers' equation on a single core, measured in seconds.

one-dimensional linear advection equation, Table 6.2, we observe the computational advantage of the hybrid RBF-ENO method in multiple dimensions. Similar to the comparison with the CWENO method we do not see any marginal difference between the solutions of the different methods with $N_{\mathrm{TRI}} = 2390$, Figure 6.17.

6.6.2 Shock vortex interaction problem

The already considered shock vortex interaction problem is based on (4.54), (4.55) and (4.56). To compare the solutions with the ones from before, we choose the parameters



(a) Hybrid RBF-ENO of order 3 with $N = 18368$ and $\mathrm{CFL} = 0.8$.

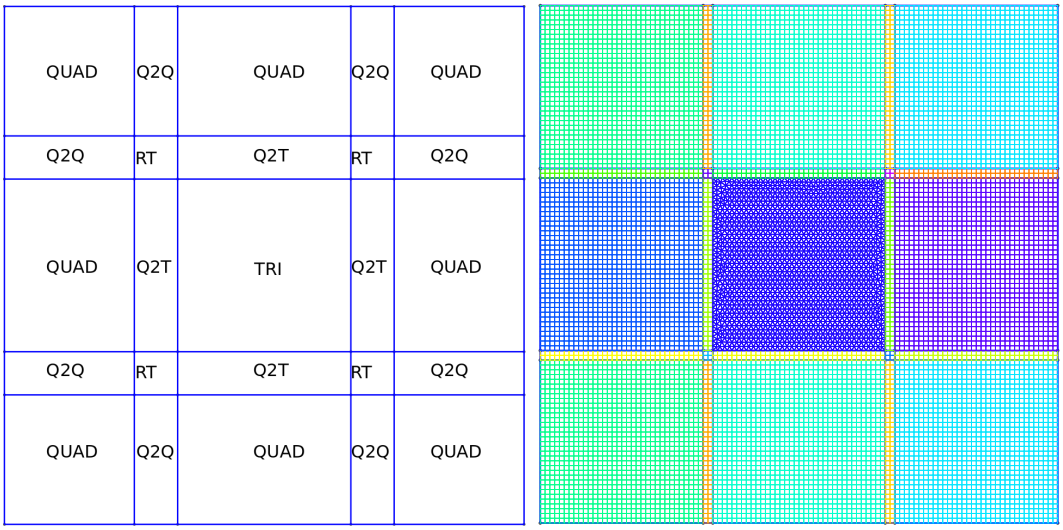
(b) RBF-ENO of order 3 with $N = 14616$ and $\mathrm{CFL} = 0.5$.

Figure 6.18 – Shock vortex interaction problem at $T = 0.35$ with 20 contour lines in $[0.8, 1.42]$.

of the vortex as in Section 4.3.4 $\epsilon = 0.3$, $r_c = 0.05$, $\beta = 0.204$ with the $(x_c, y_c) = (0.25, 0.5)$. The computational grid is shown in Figure 6.5 with $n_{\text{ghost}} = 2$ in $[0,1] \times [0,1]$. The comparison between the hybrid RBF-ENO method and the RBF-ENO method of order 3 shows a similar behavior, Figure 6.18. Both the shock and the vortex are resolved in a similar way. We observe a more oscillatory solution in the hybrid case than in the non-hybrid one. This might come from the differences of the grid. The triangular part of the hybrid grid is highly unstructured, while the triangulation from Figure 6.18b is similar to the one from Figure 4.4.

6.6.3 Riemann problem 12

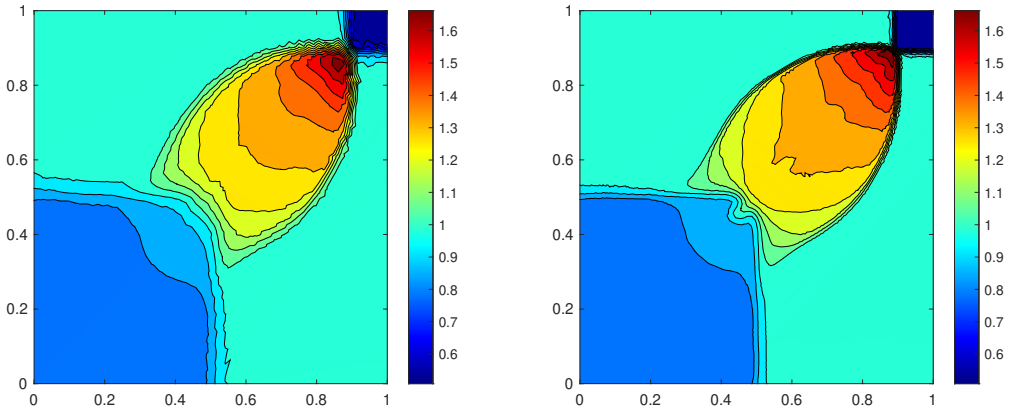
The Riemann problem 12 is based on the initial conditions (4.52) with values from Table 4.3. The hybrid grid is constructed in the same manner as the one for the Burgers' equation with a more structured triangular part, Figure 6.19. As for the Burgers' equation, the triangular part is in the square $[0,1] \times [0,1]$. The results of the hybrid RBF-ENO method of order 3, see Figure 6.20, are similar to the one calculated with the RBF-ENO3 method, see Figure 4.11. This example is just constructed to show that we get comparable results to the ones calculated with the RBF-ENO method. In practice, the division in triangles is not required for these examples.



(a) Schematic illustration of the patches.

(b) Grid with $N_{\mathrm{TRI}} = 3046$ and $N_{\mathrm{QUAD}} = 11700$ cells.

Figure 6.19 – Hybrid grid for the Riemann problem 12.



(a) $N_{\mathrm{TRI}} = 3046$ and $N_{\mathrm{QUAD}} = 11700$.

(b) $N_{\mathrm{TRI}} = 14784$ and $N_{\mathrm{QUAD}} = 51200$.

Figure 6.20 – Riemann problem 12 at $T = 0.25$, $\mathrm{CFL} = 0.8$ and 18 contour lines in $[0.51, 1.66]$.

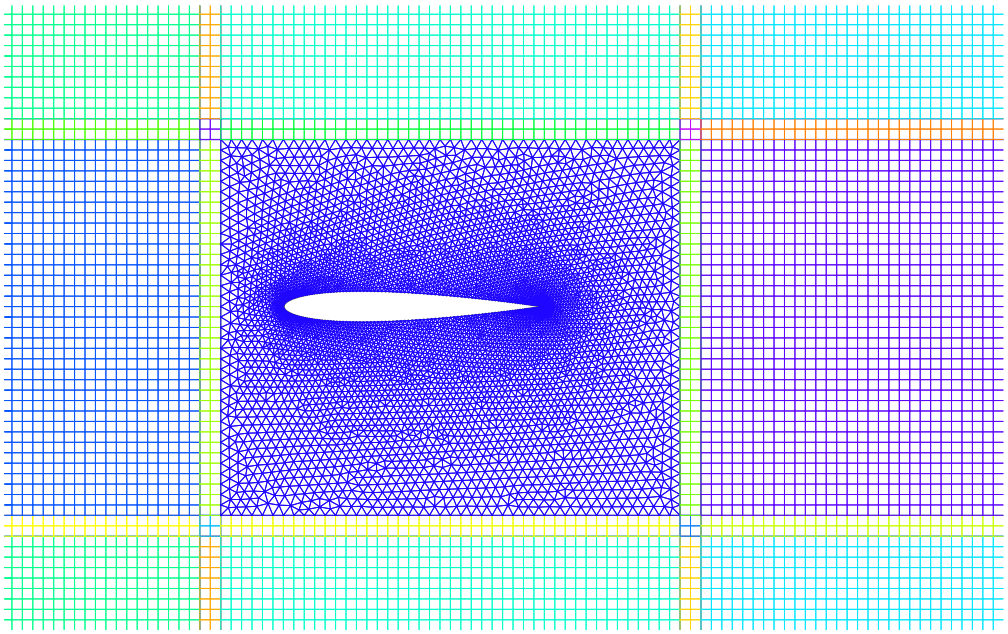


Figure 6.21 – Hybrid grid for the Airfoil NACA-0012.

6.6.4 Transonic flow past NACA-0012 airfoil

The NACA-0012 is a two-dimensional cross section of an airfoil for aircraft wing. It is based on the first systematic tests of airfoils in a wind tunnel. The NACA-0012 is a common test case for numerical solvers. It has no camber and a ratio of profile thickness to chord length of 0.12. The transonic simulation of a NACA-0012 airfoil in a freestream of Mach number $M_\infty = 0.85$ with an angle of attack α builds one shock at the top and one at the bottom of the airfoil. The hybrid grid is build in the same manner as the one from the problems before. We have eight QUAD patches, a TRI one in the center of the grid and some Q2Q, Q2T and RT connection patches. The whole grid is of the size $[-2, 15] \times [-8, 8]$ with the triangular grid inside $[-0.2, 1.5] \times [-0.8, 0.8]$ and it consists of 199 points on the airfoil, $N_{\mathrm{QUAD}} = 169036$ quadrilaterals and $N_{\mathrm{TRI}} = 10012$ triangles . The central part of the grid with its triangulation is shown in Figure 6.21. The solution with an angle of attack $\alpha = 0^\circ$ of the hybrid RBF-ENO method of order 3 is shown in Figure 6.22. We observe the characteristic steady shock waves at the top and the bottom of the surface of the airfoil, which are comparable to the results in [12]. For an angle of attack $\alpha = 2^\circ$ we show the Mach number of the solution in Figure 6.24. The dimensionless pressure coefficient

$$C_p = \frac{2}{\gamma M_\infty^2}\Big(\frac{p}{p_\infty} - 1\Big), \tag{6.105}$$

with the farfield pressure p_∞ and the pressure $p = p(x)$ is often used in aerodynamics and hydrodynamics to test engineering models. The pressure coefficient at the surface of the airfoil is shown in Figure 6.23. For zero degree angle of attack, we observe a

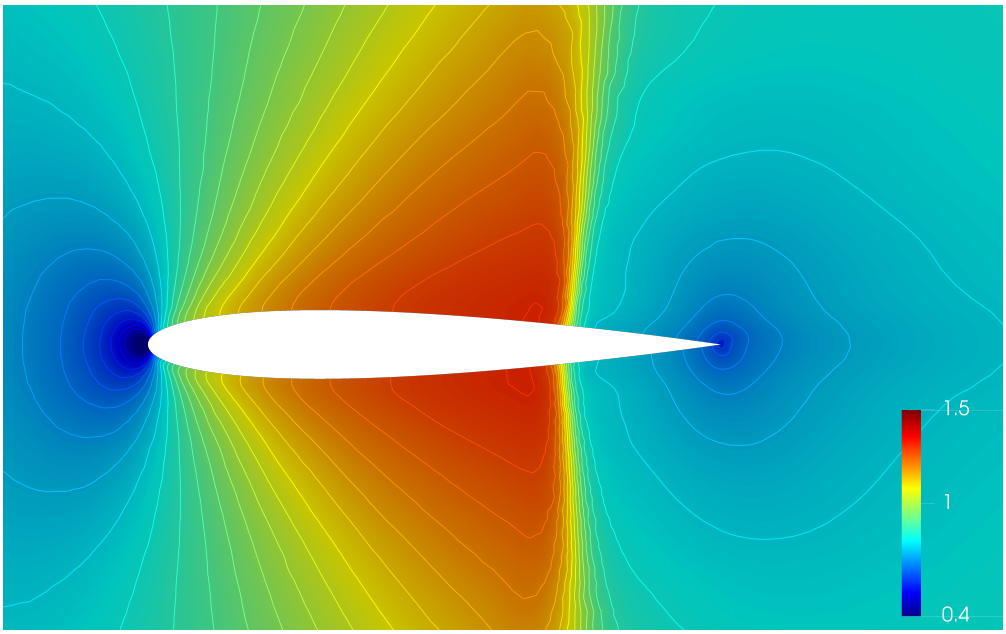


Figure 6.22 – Mach number of the Airfoil NACA-0012 problem with $\alpha = 0^\circ$ by the hybrid RBF-ENO method of order 3 with $\mathrm{CFL} = 0.8$ and 30 contour lines between 0.4 and 1.5.

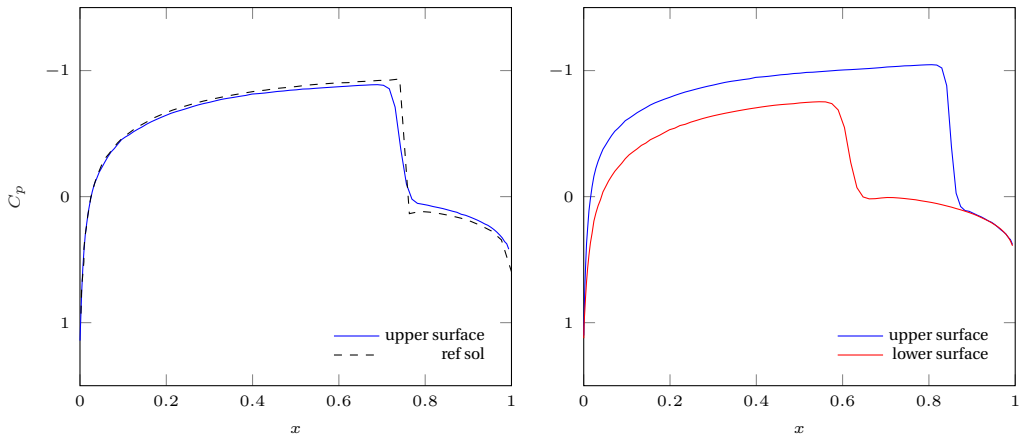
qualitatively similar solution to that in [12], see Figure 6.23a.

6.6.5 Flow around a cylinder

The airfoil is designed to be in a steady air stream. Thus, it might be that the flow around a cylinder is more delicate. We simulate the flow around a cylinder of radius $7/16$ centered at $(1.5, 0)$ in a domain $[-1, 6] \times [-2.5, 2.5]$. The grid is based on the ones from before with the triangular part in $[1, 2] \times [-0.5, 0.5]$ with $N_{\mathrm{TRI}} = 12422$ and $N_{\mathrm{QUAD}} = 434884$ with 332 cells on the cylinder surface.
The initial conditions are

$$\rho_\infty = 1, \qquad u_{1,\infty} = \sqrt{\gamma} M_\infty, \qquad u_{2,\infty} = 0, \qquad p_\infty = 1, \tag{6.106}$$

with the energy $E_\infty = \frac{p_\infty}{\gamma - 1} + \frac{1}{2}(u_{1,\infty}^2 + u_{2,\infty}^2)\rho_\infty$. The density plots for the Mach numbers $M_\infty = 0.5, 1$ are shown in the Figures 6.25 and 6.26. In both cases, we are not at steady state, but the position of the bow shock is almost steady and well resolved. Similar to flow around the airfoil, it develops shock waves on the surface, where we pass from a supersonic regime to a subsonic regime.



(a) $\alpha = 0^\circ$ with reference solution [12]. (b) $\alpha = 2^\circ$.

Figure 6.23 – Pressure coefficient at the surface of the airfoil.

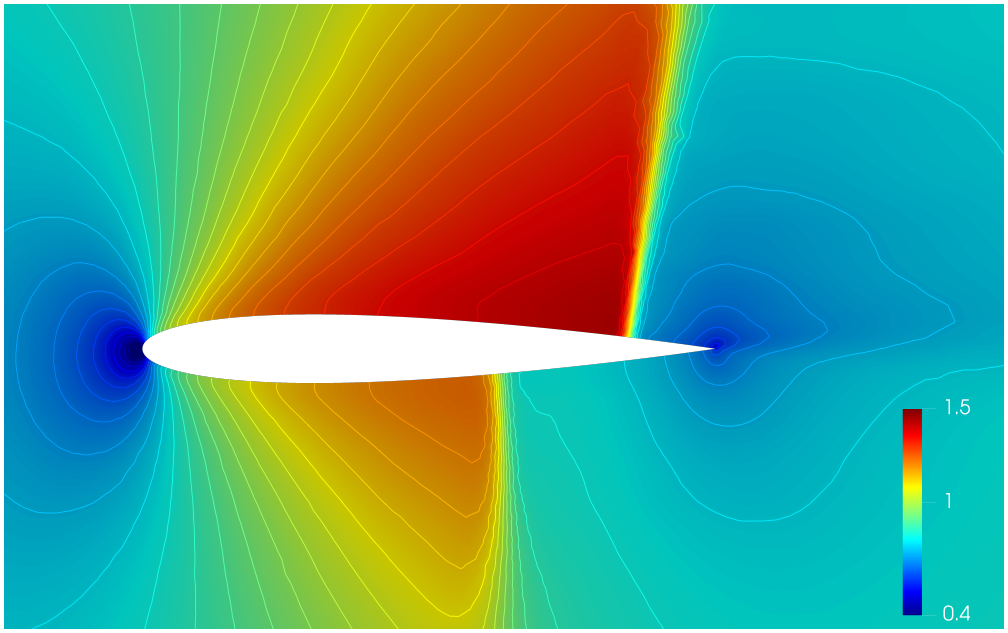


Figure 6.24 – Mach number of the Airfoil NACA-0012 problem with $\alpha = 2^\circ$ by the hybrid RBF-ENO method of order 3 with $\mathrm{CFL} = 0.8$ and 30 contour lines between 0.4 and 1.5.

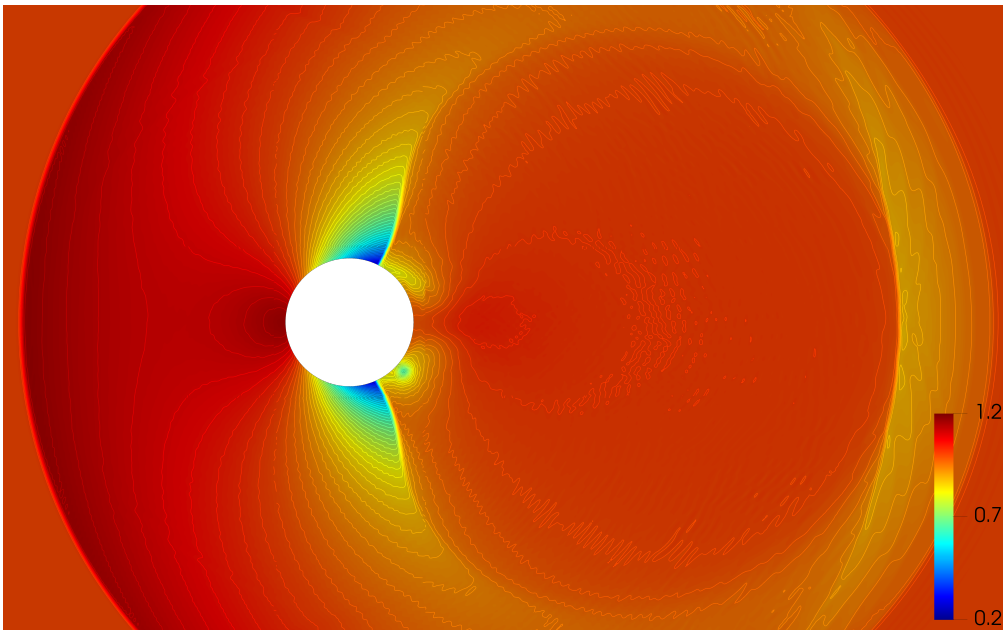


Figure 6.25 – Density plot of flow around a cylinder with $M_\infty = 0.5$ at time $T = 3$ solved with the hybrid RBF-ENO method of order 3 with $\mathrm{CFL} = 0.8$ and 50 contour lines between 0.2 and 1.2.

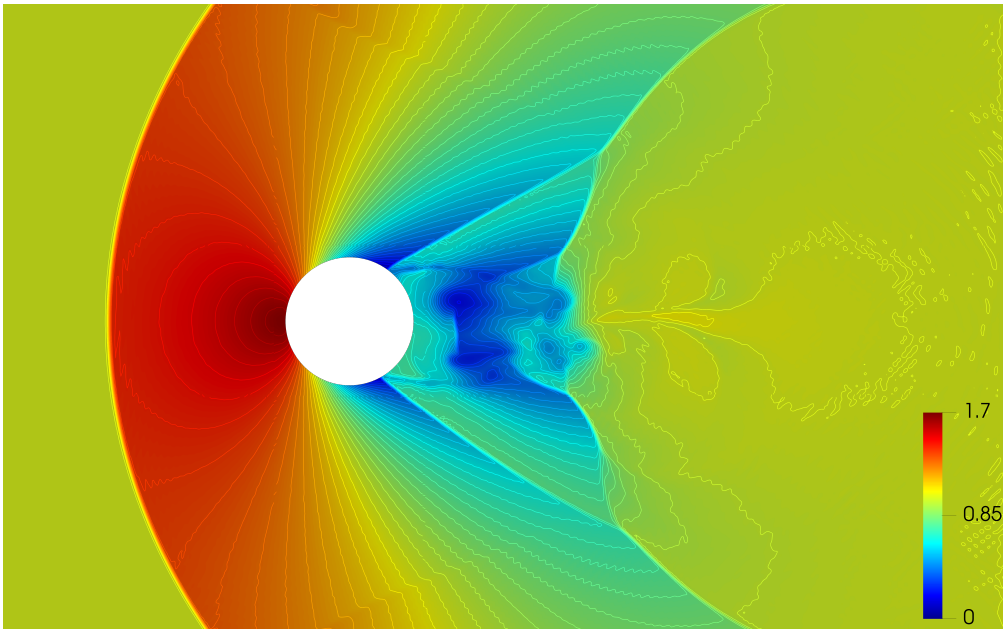


Figure 6.26 – Density plot of flow around a cylinder with $M_\infty = 1$ at time $T = 3$ solved with the hybrid RBF-ENO method of order 3 with $\mathrm{CFL} = 0.8$ and 50 contour lines between 0 and 1.7.

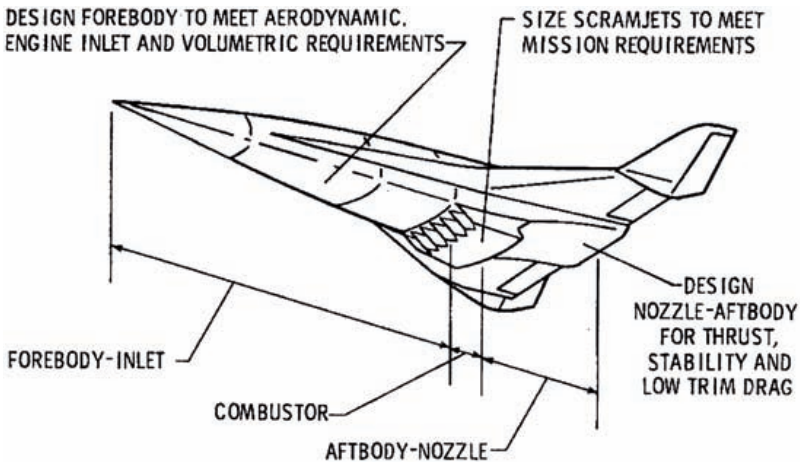


Figure 6.27 – Design features for efficient engine / airframe integration [29].

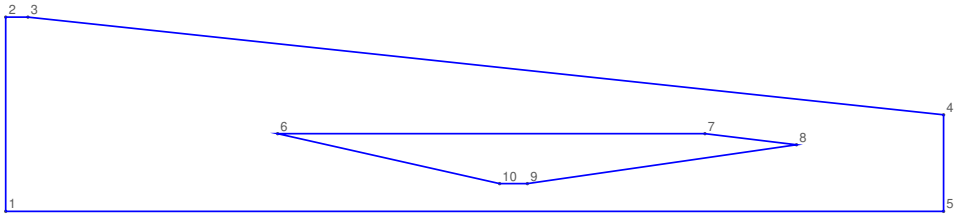


Figure 6.28 – Geometry of the scramjet model from [88].

6.6.6 Scramjet flow model

The supersonic combustion ramjet (scramjet) is based on the ramjet engine. The idea is to avoid the deceleration before the combustion to increase its efficiency at high speeds. Similar to the ramjet it requires hypersonic initial speed and must therefore be accelerated by other jet engines. Figure 6.27 shows the design of an airbeathing cruise or acceleration vehicle with a scramjet engine at the bottom composed out of six modules [29]. More details can be found in [110, 31].
We are simulating the two-strut scramjet [75, 28, 95, 6, 65] with the geometrical details from [88]. However, due to the symmetry we use just the upper half with symmetric boundary conditions, Figure 6.28 with the coordinates in Table 6.4. In the first experiment, we enforce a Mach 3 inflow at the inlet between the points 1 and 2 and outflow boundary conditions at the outlet between the points 4 and 5. At the real walls we apply slip wall boundary conditions [89] and symmetric boundary conditions between the points 1 and 5. Kumar [75] simulates the scramjet engine solving the full Navier-Stokes equations. We are interested in the shock capturing of the method and therefore consider the inviscid Euler equations (2.9). The simulation is performed on a grid based on the division into patches shown in Figure 6.29 with $N_{\mathrm{TRI}} = 5036$ and $N_{\mathrm{QUAD}} = 14939$. It is generated using the frontal Delaunay option in Gmsh. The

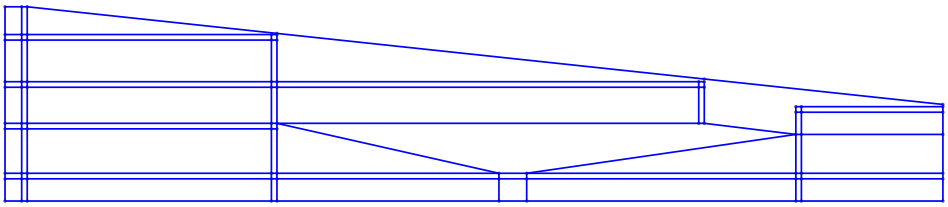


Figure 6.29 – Scheme for hybrid grid for scramjet model.

solution of the hybrid RBF-ENO method, shown in Figure 6.30, is comparable with the reference solution from [88] in Figure 6.31. Following Eberle et al. [28] we also model

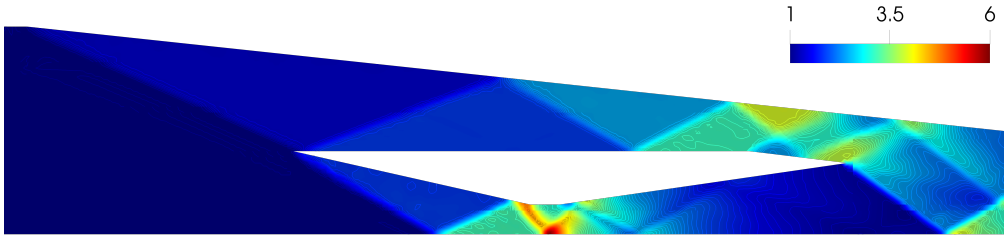
Points	1	2	3	4	5	6	7	8	9	10
x-Coord	0	0	0.4	16.9	16.9	4.9	12.6	14.25	9.4	8.9
y-Coord	0	3.5	3.5	1.74	0	1.4	1.4	1.2	0.5	0.5

Table 6.4 – Coordinates defining the geometry of the scramjet model from [88].

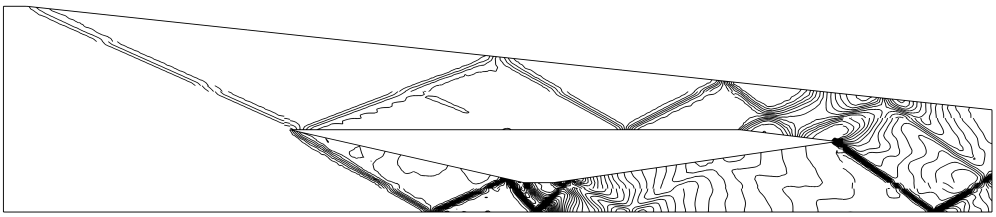
the more difficult problem with a Mach 10 inflow. Due to the strong shock waves, we get negative density and pressure with the original hybrid RBF-ENO method. Hence, we calculate the solution with the positivity preserving limiter described in Section 6.4. The solution of the Mach 10 inflow problem, Figure 6.32, is comparable with the reference solution from [28] in Figure 6.33.

6.6.7 Flow through conical aerospike nozzle

One approach to create thrust is the conical aerospike nozzle [119]. In this section, we consider some nozzle jet flow simulations using a conical aerospike nozzle. Different from the bell nozzle the aerospike nozzle is an annular nozzle and it develops the thrust against the outer surface of the conical plug at its center. At design pressure the efficiency of the aerospike nozzle is the same as for the bell nozzle. However, in the case of lower and higher outer pressure the aerospike nozzle is more efficient. This advantage comes with a high price of construction complexity. An improvement of this concept is the aerospike nozzle with a truncated plug. The lost thrust is compensated by additional cold air injected at the truncation face, called base bleed. More precisely, the pressure in this cold gas area, which is acting on the truncated face, adds the additional thrust. This concept appears to give promising results for the development of reusable launch vehicles as Single-Stage-To-Orbit or Two-Stage-To-Orbit systems. In recent years, multiple studies of conical aerospike nozzles have been done [68, 97, 129, 30, 106, 101]. This numerical example is based on an experiment by Verma [122] with a linear plug geometry, Figure 6.34. Based on the experiments of Verma, multiple numerical studies were carried out [60, 69, 94]. These studies analyze the shock-boundary layer interaction and use the axisymmetric Navier-Stokes equations.



(a) Density plot.



(b) 50 Contour lines between 1.0 and 6.0.

Figure 6.30 – Density in the scramjet engine with Mach 3 inflow by the hybrid RBF-ENO method of order 3 with $\mathrm{CFL} = 0.8$.

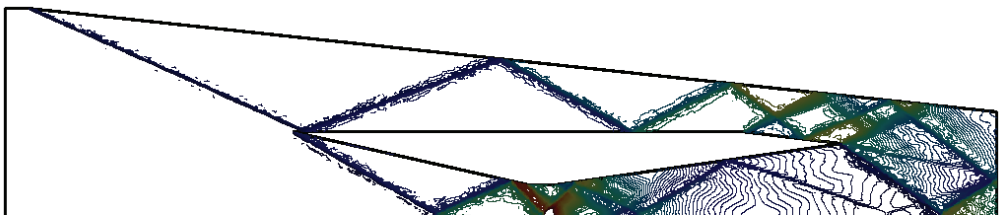
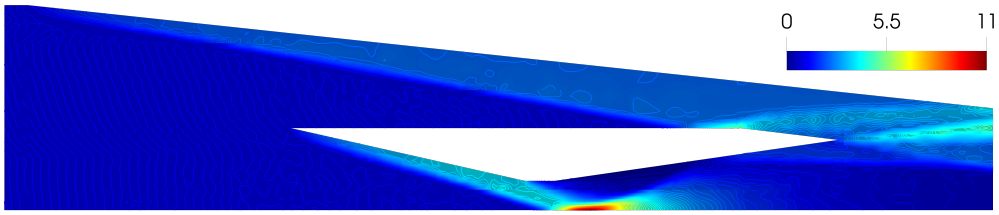
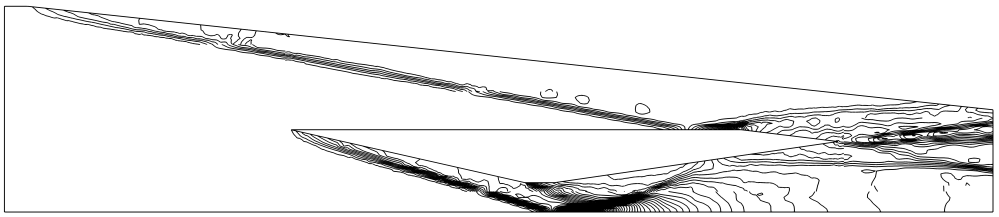


Figure 6.31 – Reference solution of the density in the scramjet engine with Mach 3 inflow [88].



(a) Density plot.



(b) 50 Contour lines between 0.0 and 11.0.

Figure 6.32 – Density in the scramjet engine with Mach 10 inflow by the hybrid RBF-ENO method of order 3 with $\mathrm{CFL} = 0.8$.

Figure 6.33 – Reference solution of the density in the scramjet engine with Mach 10 inflow [28].

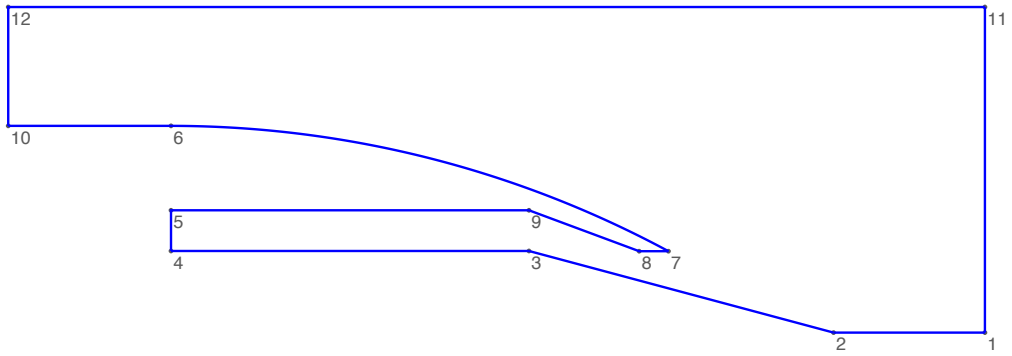


Figure 6.34 – Geometry of the nozzle model with the details from Table 6.5 [122] .

As for the scramjet simulation we consider the 3rd order method with the shock waves and use the axisymmetric Euler equations.

Points	1	2	3	4	5	6	7	8	9	10	11	12
x-Coord	$25D$	203.51	109.95	0	0	0	152.8	143.8	109.95	$-5D$	$25D$	$-5D$
y-Coord	0	0	4.5	A	$A+12.5$	63.5	25	25	$A+12.5$	63.5	$5D$	$5D$

Table 6.5 – Coordinates defining the geometry of the nozzle from [94] with $A = 25.0705074$ and $D = 50$. The curved line is a circular segment with radius $R = \frac{152.8^2+38.5^2}{2\times 38.5}$.

Axisymmetric Euler equations

The axisymmetric Euler equations are using cylindrical coordinates and they can be written with or without swirling flows [10]. Let us assume the Euler equations in three space dimensions with the variables $\mathbf{x} = (x_1, x_2, x_3)^T \in \mathbb{R}^3$

$$\frac{\partial \mathbf{u}}{\partial t} + \sum_{i=1}^{3} \frac{\partial}{\partial x_i} f_i(\mathbf{u}), \tag{6.107}$$

with $\mathbf{u} = (\rho, m_1, m_2, m_3, E)^T$ and

$$f_i(\mathbf{u}) = \begin{pmatrix} m_i \\ \frac{m_i m_1}{\rho} + p\delta_{i1} \\ \frac{m_i m_2}{\rho} + p\delta_{i2} \\ \frac{m_i m_3}{\rho} + p\delta_{i3} \\ \frac{m_i}{\rho}(E+p) \end{pmatrix}. \tag{6.108}$$

The axisymmetric Euler equations are based on the cylindrical coordinates (x, r, θ) and the relation

$$(x_1, x_2, x_3) = (x, r\cos\theta, r\sin\theta), \tag{6.109}$$

with the symmetry assumption around the x-axis all terms with partial derivative in θ are zero. This gives the following system of equations

$$\frac{\partial \hat{\mathbf{u}}}{\partial t} + \frac{\partial}{\partial x} f_x(\hat{\mathbf{u}}) + \frac{\partial}{\partial r} f_r(\hat{\mathbf{u}}) = H(\hat{\mathbf{u}}), \tag{6.110}$$

with $\hat{\mathbf{u}} = (\rho, m_x, m_r, m_\theta, E_S)^T$ and

$$f_x(\hat{\mathbf{u}}) = \begin{pmatrix} m_x \\ \frac{m_x^2}{\rho} + p \\ \frac{m_x m_r}{\rho} \\ \frac{m_x m_\theta}{\rho} \\ \frac{m_x}{\rho}(E_S + p) \end{pmatrix}, \quad f_r(\hat{\mathbf{u}}) = \begin{pmatrix} m_r \\ \frac{m_r m_x}{\rho} \\ \frac{m_r^2}{\rho} + p \\ \frac{m_r m_\theta}{\rho} \\ \frac{m_r}{\rho}(E_S + p) \end{pmatrix}, \quad H(\tilde{\mathbf{u}}) = -\frac{1}{r} \begin{pmatrix} m_r \\ \frac{m_r m_x}{\rho} \\ \frac{m_r^2 - m_\theta^2}{\rho} \\ 2\frac{m_r m_\theta}{\rho} \\ \frac{m_r}{\rho}(E_S + p) \end{pmatrix},$$

with $E_S = E - \frac{1}{2}u_\theta^2$ [10].
Note that even if we assume the derivatives in θ to be zero the axisymmetric Euler equations (6.110) include swirling flows $u_\theta \neq 0$. Thus, we have a two-dimensional system of equations of size five.
In the simplified case without swirling flows, we assume $u_\theta = 0$ and we obtain

$$\frac{\partial \tilde{\mathbf{u}}}{\partial t} + \frac{\partial}{\partial x} \tilde{f}_x(\tilde{\mathbf{u}}) + \frac{\partial}{\partial r} \tilde{f}_r(\tilde{\mathbf{u}}) = \tilde{H}(\tilde{\mathbf{u}}), \tag{6.111}$$

with $\tilde{\mathbf{u}} = (\rho, m_x, m_r, E_S)^T$ and

$$\tilde{f}_x(\tilde{\mathbf{u}}) = \begin{pmatrix} m_x \\ \frac{m_x^2}{\rho} + p \\ \frac{m_x m_r}{\rho} \\ \frac{m_x}{\rho}(E_S + p) \end{pmatrix}, \quad \tilde{f}_r(\tilde{\mathbf{u}}) = \begin{pmatrix} m_r \\ \frac{m_r m_x}{\rho} \\ \frac{m_r^2}{\rho} + p \\ \frac{m_r}{\rho}(E_S + p) \end{pmatrix}, \quad \tilde{H}(\tilde{\mathbf{u}}) = -\frac{1}{r} \begin{pmatrix} m_r \\ \frac{m_r m_x}{\rho} \\ \frac{m_r^2}{\rho} \\ \frac{m_r}{\rho}(E_S + p) \end{pmatrix}.$$

This system of equations is equivalent to the two-dimensional Euler equations (2.9) with the additional source term $\tilde{H}$.

High-order source term

To solve hyperbolic conservation laws with source term

$$\begin{aligned} \mathbf{u}_t + \sum_{i=1}^{d} f_i(\mathbf{u})_{x_i} &= S(\mathbf{u}, \mathbf{x}, t), \quad (\mathbf{x}, t) \in \mathbb{R}^d \times \mathbb{R}_+, \\ \mathbf{u}(0) &= \mathbf{u}_0, \end{aligned} \tag{6.112}$$

with the source term S, we can use the finite volume method (2.42) and add an approximation of the average source term over the cell C_i

$$\frac{\mathrm{d}U_i}{\mathrm{d}t} + \frac{1}{|C_i|}\sum_{l_e=1}^{3} F_{il_e} = S_i, \tag{6.113}$$

with

$$S_i = \frac{1}{|C_i|}\int_{C_i} S(\mathbf{u}, \mathbf{x}, t)\mathrm{d}\mathbf{x} + \mathcal{O}(\Delta\mathbf{x}^p). \tag{6.114}$$

For a first order method we use

$$S_i = S(\mathbf{U}_i, \mathbf{x}_{M,i}, t), \tag{6.115}$$

with the midpoint $\mathbf{x}_{M,i}$ of the cell C_i. In the case of higher order methods, we need to distinguish between triangular cells and the quadrilaterals. For triangular cells using the RBF-ENO reconstruction, we make use of the high-order reconstruction s_i from the ENO step and evaluate it at the two-dimensional symmetric Gaussian quadrature points $\mathbf{x}_k$ of order p for triangles

$$S_i = \sum_{k=1}^{n_Q} \omega_k S(s_i(\tilde{\mathbf{x}}_k), \mathbf{x}_k, t), \tag{6.116}$$

with its quadrature weights ω_k and points $\tilde{\mathbf{x}}_k$ [27]. The additional evaluations of the reconstruction increase the cost only marginal.
In the case of quadrilateral cells, we can not use the same technique since we never construct the explicit polynomial. However, we can adapt the technique introduced by Buchmüller and Helzel [14].
Let us assume the cell

$$C_{ij} = \{(x_{i-1/2}, y_{j-1/2}), (x_{i-1/2}, y_{j+1/2}), (x_{i+1/2}, y_{j-1/2}), (x_{i+1/2}, y_{j+1/2})\},$$

with $i, j \in \mathbb{N}$. We seek for a high-order approximation of the integral of the source term based on a quadrature rule in one dimension

$$S_{ij} = \frac{1}{\Delta x \Delta y}\int_{x_{i-1/2}}^{x_{i+1/2}}\int_{y_{i-1/2}}^{y_{i+1/2}} S(\mathbf{u}, (x, y), t)\mathrm{d}y\mathrm{d}x \approx \sum_{k=1}^{n_Q}\sum_{l=1}^{n_Q} \omega_k\omega_l S(\mathbf{u}(\tilde{\mathbf{x}}_{kl}^{ij}), \tilde{\mathbf{x}}_{kl}^{ij}, t), \tag{6.117}$$

with the quadrature nodes $\tilde{\mathbf{x}}_{kl}^{ij} = (x_{ik}, y_{jl})$ for $k, l = 1, \ldots, n_Q$. Thus, the goal is to find a high-order approximation of $\mathbf{u}(\tilde{\mathbf{x}}_{kl}^{ij})$ in terms of the average cell values $\mathbf{U}_{i,j}$, see Figure 6.35. In a first step, we express the edge averages at the quadrature nodes x_{ik}

$$\tilde{\mathbf{U}}_k^{ij} = \frac{1}{\Delta y}\int_{y_{j-1/2}}^{y_{j+1/2}} \mathbf{u}(x_{ik}, y)\mathrm{d}y = \tilde{\mathbf{U}}_k(\ldots, \mathbf{U}_{i,j}, \mathbf{U}_{i+1,j}, \ldots) + \mathcal{O}(\Delta x^p), \tag{6.118}$$

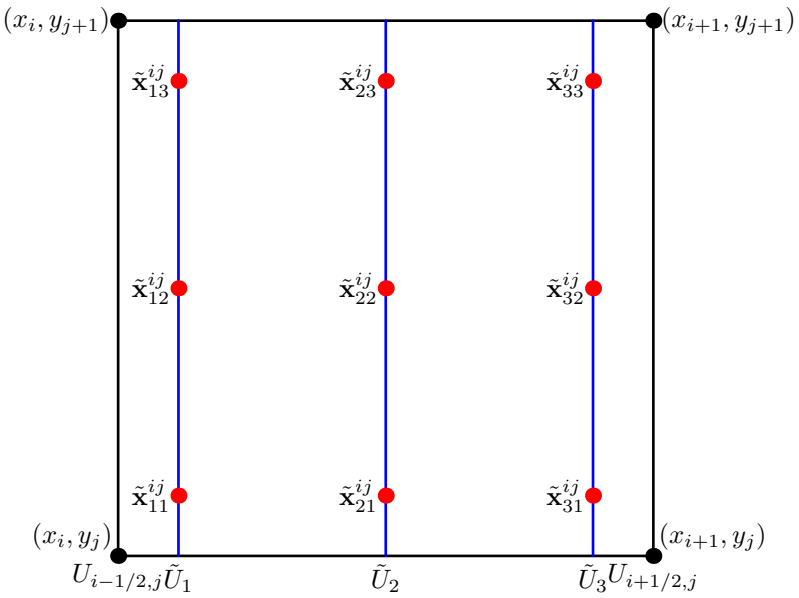


Figure 6.35 – Principle of 2D quadrilateral quadrature for $n_Q = 3$.

in terms of cell averages $\mathbf{U}_{i,j}$. In the second step, we estimate

$$\mathbf{u}(\tilde{\mathbf{x}}_{kl}^{ij}) = \tilde{\mathbf{u}}_l(\dots, \tilde{\mathbf{U}}_k^{ij}, \dots) + \mathcal{O}(\Delta y^p), \tag{6.119}$$

with the edge averages $\tilde{\mathbf{U}}_k^{ij}$ for $k \in \mathbb{N}$.

Example 6.2. *In the case of Gauss-Legendre integration with $n_Q = 3$, we receive the following approximations*

$$\begin{aligned}
\tilde{\mathbf{U}}_1^{ij} &= \frac{\mathbf{U}_{i-1,j}}{4\sqrt{3}} + \mathbf{U}_{i,j} - \frac{\mathbf{U}_{i+1,j}}{4\sqrt{3}} + \mathcal{O}(\Delta x^p), \\
\tilde{\mathbf{U}}_2^{ij} &= \frac{-\mathbf{U}_{i-1,j}}{24} + \frac{26\mathbf{U}_{i,j}}{24} - \frac{\mathbf{U}_{i+1,j}}{24} + \mathcal{O}(\Delta x^p), \\
\tilde{\mathbf{U}}_3^{ij} &= \frac{-\mathbf{U}_{i-1,j}}{4\sqrt{3}} + \mathbf{U}_{i,j} + \frac{\mathbf{U}_{i+1,j}}{4\sqrt{3}} + \mathcal{O}(\Delta x^p),
\end{aligned}$$

for the edge averages. For the evaluation at the quadrature points we receive

$$\begin{aligned}
\mathbf{u}(\tilde{\mathbf{x}}_{k1}^{ij}) &= \frac{\tilde{\mathbf{U}}_k^{ij-1}}{4\sqrt{3}} + \tilde{\mathbf{U}}_k^{ij} - \frac{\tilde{\mathbf{U}}_k^{ij+1}}{4\sqrt{3}} + \mathcal{O}(\Delta x^p) + \mathcal{O}(\Delta y^p), \\
\mathbf{u}(\tilde{\mathbf{x}}_{k2}^{ij}) &= \frac{-\tilde{\mathbf{U}}_k^{ij-1}}{24} + \frac{26\tilde{\mathbf{U}}_k^{ij}}{24} - \frac{\tilde{\mathbf{U}}_k^{ij+1}}{24} + \mathcal{O}(\Delta x^p) + \mathcal{O}(\Delta y^p), \\
\mathbf{u}(\tilde{\mathbf{x}}_{k3}^{ij}) &= \frac{-\tilde{\mathbf{U}}_k^{ij-1}}{4\sqrt{3}} + \tilde{\mathbf{U}}_k^{ij} + \frac{\tilde{\mathbf{U}}_k^{ij+1}}{4\sqrt{3}} + \mathcal{O}(\Delta x^p) + \mathcal{O}(\Delta y^p),
\end{aligned}$$

with $p = 5$.

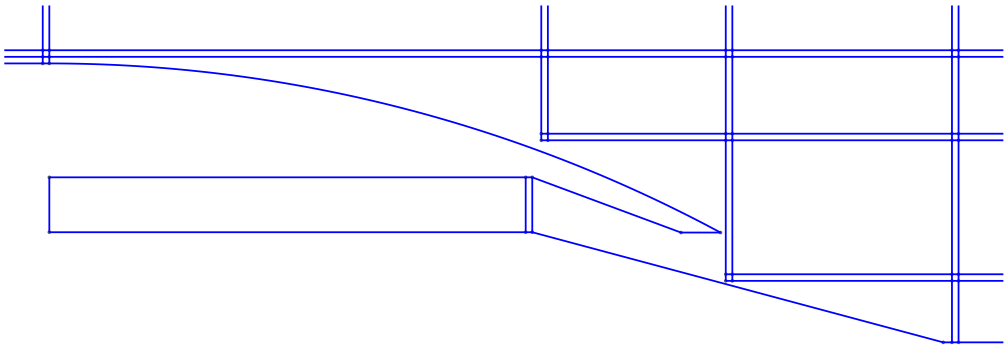


Figure 6.36 – Hybrid mesh of the aerospike nozzle.

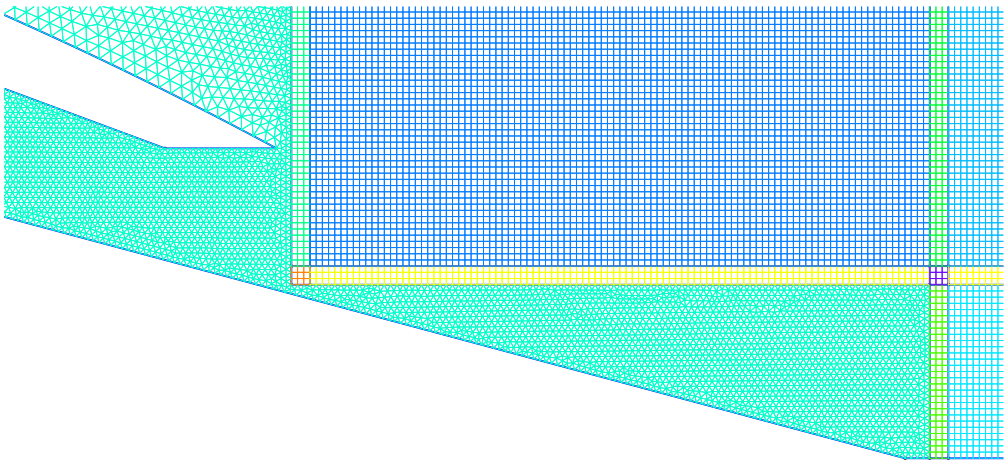


Figure 6.37 – Hybrid mesh at the nozzle exit.

Combing the hybrid high-resolution RBF-ENO method with the high-order source term, we can solve the axisymmetric Euler equations (6.111).

Remark 6.4. *The high-order source term evaluation does not reduce oscillations. In principle, this could cause problems like negative density or pressure. However, we never faced this issue in our case.*

Hybrid grid and numerical results

We discretize the geometry of the conical aerospike nozzle given by Figure 6.34. The triangular part of the hybrid grid includes the nozzle exit and its outer curved surface. The remaining domain is divided into quadrilateral patches. Figure 6.36 shows the domain division into structured and unstructured patches. The final mesh at the nozzle exit can be found in Figure 6.37. Note that we use $n_{\text{ghost}} = 3$ in this example. This gives us a grid with $N_{\text{TRI}} = 18872$ and $N_{\text{QUAD}} = 1400543$.

The boundary conditions are inflow boundary conditions at the inlet between the

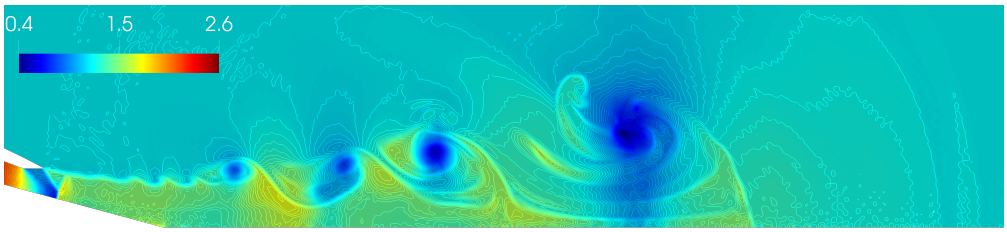


Figure 6.38 – Density of conical aerospike nozzle with $\mathrm{NPR} = 2.1$.

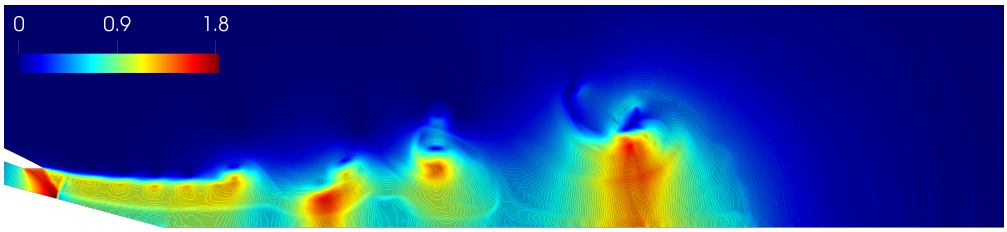


Figure 6.39 – Mach number of conical aerospike nozzle with $\mathrm{NPR} = 2.1$.

points 4 and 5, slip wall boundary conditions for the nozzle and symmetric boundary conditions at the origin $r = 0$. The outside is modeled with far-field boundary conditions with the ambient pressure $p_\infty = 101325\,\mathrm{Pa}$, temperature $T_\infty = 300\,\mathrm{K}$ and zeros speed. The ideal gas law $p = \rho \mathcal{R} T$ with the gas constant $\mathcal{R} = 287.14\,\mathrm{J/kg/K}$ is used to calculate the density. At the inlet we choose the pressure $p_{in} = \mathrm{NPR}\, p_\infty$, temperature $T_{in} = T_\infty$, $u_x = 100\,\mathrm{m/s}$ and $u_r = 0$ with the nozzle pressure ratio NPR.
Based on the results from Nair et al. we simulate the conical aerospike nozzle with $\mathrm{NPR} = 2.1$ and 3.82 [94]. The results for the nozzle pressure ratio $\mathrm{NPR} = 2.1$ at time $T = 2\,\mathrm{s}$ can be found in Figures 6.38 and 6.39 with close up view in Figure 6.42a. The Figures 6.40 and 6.41 show the density and the Mach number distribution with $\mathrm{NPR} = 3.82$ at time $T = 2\,\mathrm{s}$. We need to be aware of the difference between our model and the one from [94] and the uncertainties in the setting of the boundary conditions. However, the shock patterns of our results, Figure 6.42, are comparable to the reference solution in Figure 6.43. In Figure 6.42, we observe a discontinuous behavior of the contour lines. This is a rendering artifact coming from the transition between triangular and rectangular patches and does not influence the simulation.

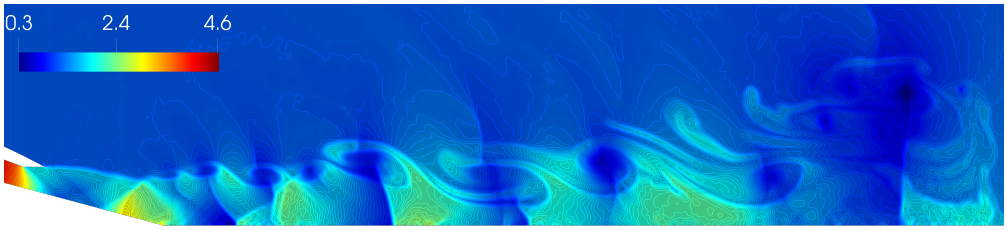


Figure 6.40 – Density of conical aerospike nozzle with $\mathrm{NPR} = 3.82$.

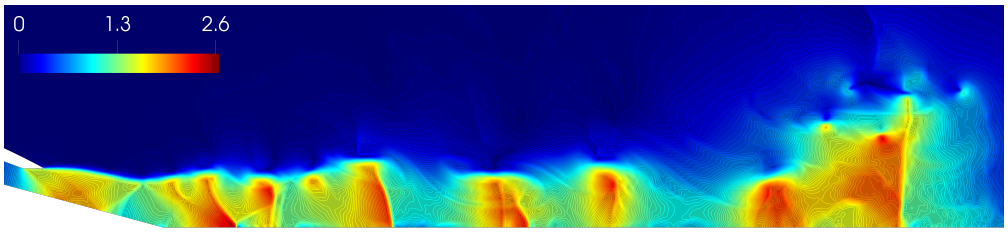
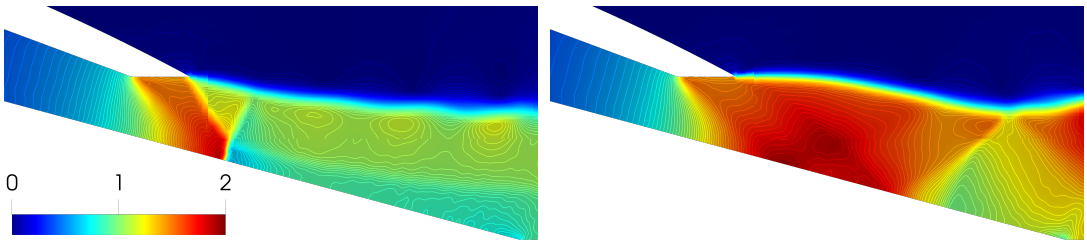
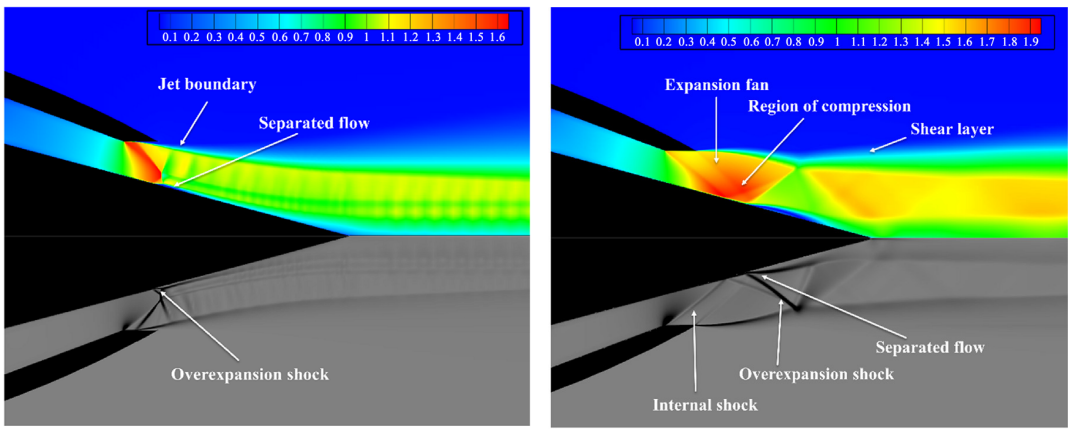


Figure 6.41 – Mach number of conical aerospike nozzle with $\mathrm{NPR} = 3.82$.



(a) $\mathrm{NPR} = 2.1$. (b) $\mathrm{NPR} = 3.82$.

Figure 6.42 – Mach number at nozzle exit at $T = 2\,\mathrm{s}$, $\mathrm{CFL} = 0.8$.



(a) $\mathrm{NPR} = 2.1$. (b) $\mathrm{NPR} = 3.82$.

Figure 6.43 – Reference computation of Mach number from Nair et al. [94].

7 Summary and Outlook

In this thesis, we have introduced a family of schemes based on radial basis functions with the goal to solve conservation laws in complex geometries. We began by introducing a new smoothness indicator based on infinitely smooth RBFs that satisfies the sign property in the limit $\Delta x \to 0$ or $\varepsilon \to 0$. In one dimension, we proved this property in the pointwise case for the 2nd and 3rd order method and conjectured it for higher orders and for the mean value interpolation. Thus, we were able to show equality in terms of stability of RBF and polynomial reconstruction methods. We formulated the RBF-TeCNOp method as an arbitrary high-order entropy stable finite difference method and the RBF-EFV2 method as a second order entropy stable finite volume method. Both are based on high-order entropy conservative schemes and a diffusion term which depends on the RBF reconstruction in the scaled entropy variables. To circumvent the ill-conditioning of the local interpolation problems we applied the vector-valued rational approximation method.

Next, we introduced the high-order RBF-ENO method on general multidimensional grids. We built on the previously introduced smoothness indicator. To reduce the computational complexity, we developed a stable evaluation method for RBFs, augmented with polynomials and a stencil selection algorithm based on the one-dimensional version. We showed that the algorithm preserves the expected accuracy and we demonstrated its robustness for challenging test cases, including two classic Riemann problems, the shock-vortex interaction and the double Mach reflection problem.

However, it is well-known that the strategy of this stencil selection algorithm is accompanied by large computational costs. As shown for the Burgers' equation, the method also works in the 4th order setup, but due to the high number of cells in the stencil it is extremely costly.

To reduce the computational complexity, we investigated a reconstruction method with fewer evaluations. The one-dimensional RBF-CWENO method is directly based on its original version [84]. The generalization into multiple dimensions is not straight forward. We combined the one-dimensional idea with the idea from the multi-resolution WENO scheme using central stencils of different sizes [137]. This method gives us the

desired speed-up along with the right order of convergence. However, the resolution close to discontinuities and multi-scale regimes is not comparable to the one from the RBF-ENO method.
Finally, we introduced the hybrid high-resolution RBF-ENO method to reduce the overall computational complexity. In the one-dimensional case, this method achieves the right order of convergence. We demonstrated the robustness of the two-dimensional hybrid high-resolution RBF-ENO method with two complex non-classical problems: the scramjet inflow problem and a conical aerospike nozzle jet simulation. To solve the conical aerospike nozzle simulation with the axisymmetric Euler equations, we described a method to evaluate the source term with high-order accuracy.

7.1 Outlook

We proved the sign property for the new smoothness indicator for first and second order RBF-reconstructions in the pointwise case. The proof of the general case is still open and requires a different technique.
A current restriction of the hybrid RBF-ENO method is the reduction of the time steps due to the difference in size of the quadrilateral and the triangular cells. This can be addressed with local time stepping. In this work, most of the two-dimensional examples are performed using a third order method. As proven with the Burgers' equation, this concept can be directly generalized to fourth and fifth order. However, it would be interesting to consider more results for higher orders for the two-dimensional hybrid RBF-ENO method.
Also we presented only the third order RBF-CWENO method. The generalization to fourth and fifth order is direct. Thus, it would be interesting to introduce a fifth order hybrid RBF-CWENO method and compare it with the hybrid RBF-ENO methods.
To simulate problems close to real applications, it would be necessary to generalize the method for viscous fluid flows. Further, to take full advantage of the method and show its power, it would be interesting to consider some full three dimensional simulations, e.g., a full simulation of a wing or a rocket.

A New diffusion matrix

The goal is to recover a new diffusion matrix that mimics the first order Rusanov-type diffusion operator. We combine results from Chandrashekar [16]

$$\Delta v_1 = \frac{\Delta\rho}{\hat{\rho}} + \Big[\frac{1}{(\gamma-1)\hat{\beta}} - \overline{u^2}\Big] - 2\overline{u}\overline{\beta}\Delta u, \tag{A.1}$$

$$\Delta v_2 = 2\overline{\beta}\Delta u + 2\overline{u}\Delta\beta, \tag{A.2}$$

$$\Delta v_3 = -2\Delta\beta, \tag{A.3}$$

with the following

$$\begin{aligned}
\Delta E &= \frac{\Delta p}{\gamma-1} + \frac{1}{2}\Delta(u^2\rho),\\
\frac{\Delta\rho}{2} &= \Delta(\beta p) = \bar{p}\Delta\beta + \bar{\beta}\Delta p,\\
\Delta(u^2\rho) &= 2\bar{\rho}\bar{u}\Delta u + \overline{u^2}\Delta\rho.
\end{aligned}$$

This can be summarized as

$$\begin{pmatrix}\Delta\rho\\ \Delta m\\ \Delta E\end{pmatrix} = \tilde{D}\begin{pmatrix}\Delta v_1\\ \Delta v_2\\ \Delta v_3\end{pmatrix}, \tag{A.4}$$

with

$$\tilde{D} = \begin{pmatrix}\hat{\rho} & \overline{u}\hat{\rho} & E_1\\ \overline{u}\hat{\rho} & \overline{u}^2\hat{\rho} + \frac{\bar{\rho}}{2\bar{\beta}} & \overline{u}E_1 + \frac{\overline{u}\,\bar{\rho}}{2\bar{\beta}}\\ E_2 & \overline{u}E_2 + \frac{\overline{u}\,\bar{\rho}}{2\bar{\beta}} & \tilde{R}\end{pmatrix}, \tag{A.5}$$

with $E_1 = \hat{\rho}\overline{u}^2 + \frac{\hat{\rho}}{2}\Big[\frac{1}{(\gamma-1)\hat{\beta}} - \overline{u^2}\Big]$, $E_2 = \frac{\hat{\rho}}{2}\Big[\frac{1}{(\gamma-1)\overline{\beta}} + \overline{u^2}\Big]$, $\tilde{R} = \frac{E_1E_2}{\hat{\rho}} + \frac{\overline{u}^2\,\bar{\rho}}{2\bar{\beta}} + \frac{\bar{p}}{2(\gamma-1)\bar{\beta}}$.
To use this in the entropy stable framework we need to symmetrize it. By assuming

$\tilde{D}_{3,1} = E_1$ and $\tilde{D}_{3,2} = \overline{u}E_1 + \frac{\overline{u}\,\overline{\rho}}{2\overline{\beta}}$ we get at least the exact jump for the density and mass flow.
We recover the matrix

$$D = \alpha \begin{pmatrix} \hat{\rho} & \overline{u}\hat{\rho} & E_1 \\ \overline{u}\hat{\rho} & \overline{u}^2\hat{\rho} + \frac{\overline{\rho}}{2\overline{\beta}} & \overline{u}E_1 + \frac{\overline{u}\,\overline{\rho}}{2\overline{\beta}} \\ E_1 & \overline{u}E_1 + \frac{\overline{u}\,\overline{\rho}}{2\overline{\beta}} & R \end{pmatrix}, \tag{A.6}$$

with $R = \frac{E_1^2}{\hat{\rho}} + \frac{\overline{u}^2\,\overline{\rho}}{2\overline{\beta}} + \frac{\overline{p}}{2(\gamma-1)\overline{\beta}}$, which mimics the Rusanov-type diffusion operator for the density and mass flow. By showing that the leading principal minors are positive we get the positive definiteness for D in the case $\gamma > 1$. However, we are not aware of an exact and stable decomposition $D = LBL^T$ for an invertible matrix L and a diagonal one B. Thus, this needs to be done numerically in each step.
Note that Derigs et al. [23] obtained a related result with a different value R.

Bibliography

[1] R. Abgrall. On essentially non-oscillatory schemes on unstructured meshes: Analysis and implementation. *Journal of Computational Physics*, 114(1):45–58, 1994.

[2] T. Aboiyar, E. H. Georgoulis, and A. Iske. High order WENO finite volume schemes using polyharmonic spline reconstruction. In *Proceedings of the international conference on numerical analysis and approximation theory NAAT2006*, Cluj-Napoca (Romania), 2006. Dept. of Mathematics. University of Leicester.

[3] T. Aboiyar, E. H. Georgoulis, and A. Iske. Adaptive ADER methods using kernel-based polyharmonic spline WENO reconstruction. *SIAM Journal on Scientific Computing*, 32(6):3251–3277, 2010.

[4] M. Abramowitz and I. A. Stegun. *Handbook of mathematical functions: with formulas, graphs, and mathematical tables*, volume 55. Courier Corporation, 1965.

[5] N. A. Adams and K. Shariff. A high-resolution hybrid compact-eno scheme for shock-turbulence interaction problems. *Journal of Computational Physics*, 127(1):27–51, 1996.

[6] F. Alauzet and P. J. Frey. Estimateur d'erreur géométrique et métriques anisotropes pour l'adaptation de maillage. partie ii : exemples d'applications. Research Report RR-4789, INRIA, 01 2003.

[7] V. Bayona. An insight into rbf-fd approximations augmented with polynomials. *Computers & Mathematics with Applications*, 2019.

[8] V. Bayona, N. Flyer, and B. Fornberg. On the role of polynomials in rbf-fd approximations: Iii. behavior near domain boundaries. *Journal of Computational Physics*, 380:378–399, 2019.

[9] V. Bayona, N. Flyer, B. Fornberg, and G. A. Barnett. On the role of polynomials in rbf-fd approximations: Ii. numerical solution of elliptic pdes. *Journal of Computational Physics*, 332:257–273, 2017.

[10] A. Bernard-Champmartin, J.-P. Braeunig, and J.-M. Ghidaglia. An Eulerian finite volume solver for multi-material fluid flows with cylindrical symmetry. *Computers & Fluids*, 83:170–176, 2013.

[11] C. Bigoni and J. S. Hesthaven. Adaptive WENO methods based on radial basis function reconstruction. *Journal of Scientific Computing*, 72(3):986–1020, 2017.

[12] F. Bisson, S. Nadarajah, and D. Shi-Dong. Adjoint-based aerodynamic optimization of benchmark problems. In *52nd Aerospace Sciences Meeting*, page 1948, 01 2014.

[13] J. P. Boyd. Error saturation in gaussian radial basis functions on a finite interval. *Journal of Computational and Applied Mathematics*, 234(5):1435–1441, 7 2010.

[14] P. Buchmüller and C. Helzel. Improved accuracy of high-order weno finite volume methods on cartesian grids. *Journal of Scientific Computing*, 61(2):343–368, 2014.

[15] M. D. Buhmann. Radial basis functions. *Acta Numerica 2000*, 9:1–38, 2000.

[16] P. Chandrashekar. Kinetic energy preserving and entropy stable finite volume schemes for compressible euler and navier-stokes equations. *Communications in Computational Physics*, 14(5):1252–1286, 2013.

[17] X. Cheng and Y. Nie. A third-order entropy stable scheme for hyperbolic conservation laws. *Journal of Hyperbolic Differential Equations*, 13(01):129–145, 2016.

[18] B. Cockburn and C.-W. Shu. Tvb runge-kutta local projection discontinuous galerkin finite element method for conservation laws. ii. general framework. *Mathematics of computation*, 52(186):411–435, 1989.

[19] M. G. Crandall and A. Majda. Monotone difference approximations for scalar conservation laws. *Mathematics of Computation*, 34(149):1–21, 1980.

[20] I. Cravero, G. Puppo, M. Semplice, and G. Visconti. Cweno: uniformly accurate reconstructions for balance laws. *arXiv preprint arXiv:1607.07319*, 2016.

[21] I. Cravero and M. Semplice. On the accuracy of weno and cweno reconstructions of third order on nonuniform meshes. *Journal of Scientific Computing*, 67(3):1219–1246, 2016.

[22] S. De Marchi and R. Schaback. Stability of kernel-based interpolation. *Advances in Computational Mathematics*, 32(2):155–161, 2010.

[23] D. Derigs, A. R. Winters, G. J. Gassner, and S. Walch. A novel averaging technique for discrete entropy-stable dissipation operators for ideal mhd. *Journal of Computational Physics*, 330:624–632, 2017.

[24] T. A. Driscoll and B. Fornberg. Interpolation in the limit of increasingly flat radial basis functions. *Computers & Mathematics with Applications*, 43(3):413–422, 2002.

[25] J. Duchon. *Splines minimizing rotation-invariant semi-norms in Sobolev spaces*, pages 85–100. Springer, 1977.

[26] M. Dumbser and M. Käser. Arbitrary high order non-oscillatory finite volume schemes on unstructured meshes for linear hyperbolic systems. *Journal of Computational Physics*, 221(2):693–723, 2007.

[27] D. Dunavant. High degree efficient symmetrical Gaussian quadrature rules for the triangle. *International journal for numerical methods in engineering*, 21(6):1129–1148, 1985.

[28] A. Eberle, M. Schmatz, and N. Bissinger. Generalized fluxvectors for hypersonic shock-capturing. In *28th Aerospace Sciences Meeting*, page 390, 1990.

[29] C. Edwards, W. Small, J. Weidner, and P. Johnston. Studies of scramjet/airframe integration techniques for hypersonic aircraft. In *13th Aerospace Sciences Meeting*, page 58, 1975.

[30] S. Eilers, W. Matthew, and S. Whitmore. Analytical and experimental evaluation of aerodynamic thrust vectoring on an aerospike nozzle. In *46th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*, page 6964, 2010.

[31] A. F. El-Sayed. *Fundamentals of aircraft and rocket propulsion*. Springer, 2016.

[32] G. E. Fasshauer. *Meshfree approximation methods with MATLAB*, volume 6. World Scientific, 2007.

[33] G. E. Fasshauer and J. G. Zhang. On choosing "optimal"shape parameters for rbf approximation. *Numerical Algorithms*, 45(1-4):345–368, 2007.

[34] U. S. Fjordholm, S. Mishra, and E. Tadmor. Well-balanced and energy stable schemes for the shallow water equations with discontinuous topography. *Journal of Computational Physics*, 230(14):5587–5609, 2011.

[35] U. S. Fjordholm, S. Mishra, and E. Tadmor. Arbitrarily high-order accurate entropy stable essentially nonoscillatory schemes for systems of conservation laws. *SIAM Journal on Numerical Analysis*, 50(2):544–573, 2012.

[36] U. S. Fjordholm, S. Mishra, and E. Tadmor. Eno reconstruction and eno interpolation are stable. *Foundations of Computational Mathematics*, 13(2):139–159, 2013.

[37] U. S. Fjordholm and D. Ray. A sign preserving WENO reconstruction method. *Journal of Scientific Computing*, pages 1–22, 2016.

[38] N. Flyer, G. A. Barnett, and L. J. Wicker. Enhancing finite differences with radial basis functions: Experiments on the navier–stokes equations. *Journal of Computational Physics*, 316:39–62, 2016.

[39] N. Flyer, B. Fornberg, V. Bayona, and G. A. Barnett. On the role of polynomials in rbf-fd approximations: I. interpolation and accuracy. *Journal of Computational Physics*, 321:21–38, 2016.

[40] B. Fornberg, E. Larsson, and N. Flyer. Stable computations with gaussian radial basis functions in 2-d. Technical report, Department of Information Technology, Uppsala University, 2009.

[41] B. Fornberg and C. Piret. A stable algorithm for flat radial basis functions on a sphere. *SIAM Journal on Scientific Computing*, 30(1):60–80, 2007.

[42] B. Fornberg and G. Wright. Stable computation of multiquadric interpolants for all values of the shape parameter. *Computers & Mathematics with Applications*, 48(5):853–867, 2004.

[43] B. Fornberg, G. Wright, and E. Larsson. Some observations regarding interpolants in the limit of flat radial basis functions. *Computers & Mathematics with Applications*, 47(1):37–55, 2004.

[44] O. Friedrich. Weighted essentially non-oscillatory schemes for the interpolation of mean values on unstructured grids. *Journal of computational physics*, 144(1):194–212, 1998.

[45] W. Gautschi. How (un) stable are vandermonde systems. *Asymptotic and computational analysis*, 124:193–210, 1990.

[46] C. Geuzaine and J. Remacle. Gmsh: A 3-d finite element mesh generator with built-in pre-and post-processing facilities. *International journal for numerical methods in engineering*, 79(11):1309–1331, 2009.

[47] E. Godlewski and P.-A. Raviart. *Hyperbolic systems of conservation laws*. Ellipses, 1991.

[48] S. K. Godunov. A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics. *Matematicheskii Sbornik*, 89(3):271–306, 1959.

[49] S. Gottlieb, D. I. Ketcheson, and C.-W. Shu. High order strong stability preserving time discretizations. *Journal of Scientific Computing*, 38(3):251–289, 2009.

[50] J.-L. Guermond, R. Pasquetti, and B. Popov. Entropy viscosity method for nonlinear conservation laws. *Journal of Computational Physics*, 230(11):4248–4267, 2011.

[51] B. Gustafsson. The convergence rate for difference approximations to mixed initial boundary value problems. *Mathematics of Computation*, 29(130):396–406, 1975.

[52] J. H. Halton. On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numerische Mathematik*, 2(1):84–90, 1960.

[53] R. L. Hardy. Multiquadric equations of topography and other irregular surfaces. *Journal of geophysical research*, 76(8):1905–1915, 1971.

[54] A. Harten. On the symmetric form of systems of conservation laws with entropy. *Journal of computational physics*, 49:151–164, 1983.

[55] A. Harten. *On high-order accurate interpolation for non-oscillatory shock capturing schemes*, pages 71–105. Springer, 1986.

[56] A. Harten and S. R. Chakravarthy. Multi-dimensional eno schemes for general geometries. Technical Report 91-76, ICASE, 1991.

[57] A. Harten, B. Engquist, S. Osher, and S. R. Chakravarthy. Uniformly high order accurate essentially non-oscillatory schemes, iii. *Journal of Computational Physics*, 71(2):231–303, 1987.

[58] A. Harten, J. M. Hyman, P. D. Lax, and B. Keyfitz. On finite-difference approximations and entropy conditions for shocks. *Communications on pure and applied mathematics*, 29(3):297–322, 1976.

[59] A. Harten and G. Zwas. Self-adjusting hybrid schemes for shock computations. *Journal of Computational Physics*, 9(3):568–583, 1972.

[60] M. He, L. Qin, and Y. Liu. Numerical investigation of flow separation behavior in an over-expanded annular conical aerospike nozzle. *Chinese Journal of Aeronautics*, 28(4):983–1002, 2015.

[61] J. S. Hesthaven. *Numerical Methods for Conservation Laws: From Analysis to Algorithms*. Society for Industrial and Applied Mathematics, 2018/04/06 2017.

[62] J. S. Hesthaven and F. Mönkeberg. Entropy stable essentially nonoscillatory methods based on RBF reconstruction. *ESAIM: M2AN*, 53(3):925–958, 2019.

[63] J. S. Hesthaven and F. Mönkeberg. Two-dimensional RBF-ENO method on unstructured grids. *Journal of Scientific Computing*, 82(3):1–24, 2020.

[64] J. S. Hesthaven, F. Mönkeberg, and S. Zaninelli. *RBF Based CWENO Method*. Springer, 2018. Accepted.

[65] J. S. Hesthaven and T. Warburton. *Nodal discontinuous Galerkin methods: algorithms, analysis, and applications*. Springer Science & Business Media, 2007.

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

[67] A. Iske and T. Sonar. On the structure of function spaces in optimal recovery of point functionals for ENO-schemes by radial basis functions. *Numerische Mathematik*, 74(2):177–201, 1996.

[68] T. Ito and K. Fujii. Numerical analysis of the base bleed effect on the aerospike nozzles. In *40th AIAA Aerospace Sciences Meeting & Exhibit*, page 512, 2002.

[69] Y. Ji, M. He, and H. Liu. Reynolds number influence on the backpressure-induced shock–boundary layer interaction in an asymmetric supersonic expansion flow. *Aerospace Systems*, pages 1–8, 2019.

[70] G.-S. Jiang and C.-W. Shu. Efficient implementation of weighted ENO schemes. *Journal of Computational Physics*, 126(1):202–228, 1996.

[71] E. Kansa. A scattered data approximation scheme with applications to computational fluid dynamics. i. surface approximations and partial derivative estimates. *Comput. Math. Appl*, 19(8):9, 1990.

[72] E. Kansa. A scattered data approximation scheme with applications to computational fluid dynamics: Ii. solutions to parabolic, hyperbolic and elliptic partial differential equations. *Comput. Math. Appl*, 19:147–61, 1990.

[73] M. Käser and A. Iske. ADER schemes on adaptive triangular meshes for scalar conservation laws. *Journal of Computational Physics*, 205(2):486–508, 5 2005.

[74] S. N. Kružkov. First order quasilinear equations in several independent variables. *Mathematics of the USSR-Sbornik*, 10(2):217, 1970.

[75] A. Kumar. Numerical analysis of the scramjet inlet flow field using two-dimensional Navier-Stokes equations. In *19th Aerospace Sciences Meeting*, page 185, 1981.

[76] A. Kurganov, G. Petrova, and B. Popov. Adaptive semidiscrete central-upwind schemes for nonconvex hyperbolic conservation laws. *SIAM Journal on Scientific Computing*, 29(6):2381–2401, 2007.

[77] C. Lanczos. *Discourse on Fourier series*, volume 76. SIAM, 2016.

[78] E. Larsson and B. Fornberg. Theoretical and computational aspects of multivariate interpolation with increasingly flat radial basis functions. *Computers & Mathematics with Applications*, 49(1):103–130, 2005.

[79] P. Lax and B. Wendroff. Systems of conservation laws. *Communications on Pure and Applied mathematics*, 13(2):217–237, 1960.

[80] P. D. Lax and X.-D. Liu. Solution of two-dimensional Riemann problems of gas dynamics by positive schemes. *SIAM Journal on Scientific Computing*, 19(2):319–340, 1998.

[81] A.-Y. le Roux. A numerical conception of entropy for quasi-linear equations. *Mathematics of Computation*, 31(140):848–872, 1977.

[82] P. G. Lefloch, J.-M. Mercier, and C. Rohde. Fully discrete, entropy conservative schemes of arbitraryorder. *SIAM Journal on Numerical Analysis*, 40(5):1968–1992, 2002.

[83] R. J. LeVeque. *Numerical methods for conservation laws*. Springer Science & Business Media, 1992.

[84] D. Levy, G. Puppo, and G. Russo. Central weno schemes for hyperbolic systems of conservation laws. *ESAIM: Mathematical Modelling and Numerical Analysis*, 33(3):547–571, 1999.

[85] D. Levy, G. Puppo, and G. Russo. Compact central weno schemes for multidimensional conservation laws. *SIAM Journal on Scientific Computing*, 22(2):656–672, 2000.

[86] X.-D. Liu, S. Osher, and T. Chan. Weighted essentially non-oscillatory schemes. *Journal of computational physics*, 115(1):200–212, 1994.

[87] A. Madrane, U. Fjordholm, S. Mishra, and E. Tadmor. Entropy conservative and entropy stable finite volume schemes for multi-dimensional conservation laws on unstructured meshes. In *European Congress Computational Methods Applied Sciences and Engineering, Proceedings of ECCOMAS 2012, held in Vienna*, 2012.

[88] A. Mazaheri, C.-W. Shu, and V. Perrier. Bounded and compact weighted essentially nonoscillatory limiters for discontinuous Galerkin schemes: Triangular elements. *Journal of Computational Physics*, 2019.

[89] G. Mengaldo, D. De Grazia, F. Witherden, A. Farrington, P. Vincent, S. Sherwin, and J. Peiro. A guide to the implementation of boundary conditions in compact high-order methods for compressible aerodynamics. In *7th AIAA Theoretical Fluid Mechanics Conference*, page 2923, 2014.

[90] M. L. Merriam. *An entropy-based approach to nonlinear stability*. Stanford University, Stanford, CA, USA, 1989.

[91] C. A. Micchelli. Interpolation of scattered data: Distance matrices and conditionally positive definite functions. *Constructive Approximation*, 2(1):11–22, 1986.

[92] G. Mühlbach. A recurrence formula for generalized divided differences and some applications. *Journal of Approximation Theory*, 9(2):165–172, 1973.

[93] G. Mühlbach. The general neville-aitken-algorithm and some applications. *Numerische Mathematik*, 31(1):97–110, 1978.

[94] P. P. Nair, A. Suryan, and H. D. Kim. Computational study on flow through truncated conical plug nozzle with base bleed. *Propulsion and Power Research*, 2019.

[95] K. Nakahashi and E. Saitoh. Space-marching method on unstructured grid for supersonic flows with embedded subsonic regions. *AIAA journal*, 35(8):1280–1285, 1997.

[96] F. J. Narcowich and J. D. Ward. Norm estimates for the inverses of a general class of scattered-data radial-function interpolation matrices. *Journal of Approximation Theory*, 69(1):84–109, 1992.

[97] M. Nazarinia, A. Naghib-Lahouti, and E. Tolouei. Design and numerical analysis of aerospike nozzles with different plug shapes to compare their performance with a conventional nozzle. In *AIAC-11 Eleventh Australian International Aerospace Congress*, 2005.

[98] O. A. Oleinik. Discontinuous solutions of non-linear differential equations. *Uspekhi Matematicheskikh Nauk*, 12(3):3–73, 1957.

[99] P.-O. Persson and G. Strang. A simple mesh generator in matlab. *SIAM review*, 46(2):329–345, 2004.

[100] B. Perthame and C.-W. Shu. On positivity preserving finite volume schemes for Euler equations. *Numerische Mathematik*, 73(1):119–130, 1996.

[101] M. Propst, J. Sieder, C. Bach, and M. Tajmar. Numerical analysis on an aerodynamically thrust-vectored aerospike nozzle. In *Proceedings of the 63rd German Aerospace Congress (DGLR), Augsburg*, 2014.

[102] H. Ranocha. Shallow water equations: split-form, entropy stable, well-balanced, and positivity preserving numerical methods. *GEM-International Journal on Geomathematics*, 8(1):85–133, 2017.

[103] D. Ray, P. Chandrashekar, U. S. Fjordholm, and S. Mishra. Entropy stable scheme on two-dimensional unstructured grids for euler equations. *Communications in Computational Physics*, 19(5):1111–1140, 2016.

[104] S. Rippa. An algorithm for selecting a good value for the parameter c in radial basis function interpolation. *Advances in Computational Mathematics*, 11(2-3):193–210, 1999.

[105] P. L. Roe. Approximate riemann solvers, parameter vectors, and difference schemes. *Journal of computational physics*, 43(2):357–372, 1981.

[106] S. Sanoob, M. Prince, and B. Sundar. Numerical analysis of aero-spike nozzle for spike length optimization. *International Journal of Research in Engineering & Technology*, 1(6):1–14, 2013.

[107] R. Schaback. Error estimates and condition numbers for radial basis function interpolation. *Advances in Computational Mathematics*, 3(3):251–264, 1995.

[108] R. Schaback. Native hilbert spaces for radial basis functions i. *New Developments in Approximation Theory*, 132:255–282, 1998.

[109] R. Schaback. Multivariate interpolation by polynomials and radial basis functions. *Constructive Approximation*, 21(3):293–317, 2005.

[110] C. Segal. *The scramjet engine: processes and characteristics*, volume 25. Cambridge University Press, 2009.

[111] M. Semplice, A. Coco, and G. Russo. Adaptive mesh refinement for hyperbolic systems based on third-order compact weno reconstruction. *Journal of Scientific Computing*, 66(2):692–724, 2016.

[112] H. Shen and M. Parsani. Positivity-preserving ce/se schemes for solving the compressible euler and navier–stokes equations on hybrid unstructured meshes. *Computer Physics Communications*, 232:165–176, 2018.

[113] C.-W. Shu. *High order ENO and WENO schemes for computational fluid dynamics*, pages 439–582. Springer, 1999.

[114] C.-W. Shu, B. Cockburn, C. Johnson, and E. Tadmor. *Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws*, pages 325–432. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998.

[115] T. Sonar. Optimal recovery using thin plate splines in finite volume methods for the numerical solution of hyperbolic conservation laws. *IMA Journal of Numerical Analysis*, 16(4):549–581, 1996.

[116] G. Strang. Accurate partial difference methods i: Linear cauchy problems. *Archive for Rational Mechanics and Analysis*, 12(1):392–402, 1963.

[117] E. Tadmor. The numerical viscosity of entropy stable schemes for systems of conservation laws. i. *Mathematics of Computation*, 49(179):91–103, 1987.

[118] R. R. Thareja, J. R. Stewart, O. Hassan, K. Morgan, and J. Peraire. A point implicit unstructured grid solver for the Euler and Navier–Stokes equations. *International journal for numerical methods in fluids*, 9(4):405–425, 1989.

[119] M. J. Turner. *Rocket and spacecraft propulsion: principles, practice and new developments*. Springer Science & Business Media, 2008.

[120] E. E. Tyrtyshnikov. How bad are hankel matrices? *Numerische Mathematik*, 67(2):261–269, 1994.

[121] B. Van Leer. Towards the ultimate conservative difference scheme. v. a second-order sequel to Godunov's method. *Journal of computational Physics*, 32(1):101–136, 1979.

[122] S. B. Verma. Performance characteristics of an annular conical aerospike nozzle with freestream effect. *Journal of Propulsion and Power*, 25(3):783–791, 2009.

[123] J. VonNeumann and R. D. Richtmyer. A method for the numerical calculation of hydrodynamic shocks. *Journal of applied physics*, 21(3):232–237, 1950.

[124] H. Wendland. *Scattered Data Approximation:*. Cambridge University Press, Cambridge, 2004.

[125] L. R. Williams and J. H. Wells. *Embeddings and Extensions in Analysis*. Springer Berlin Heidelberg, 1975.

[126] P. Woodward and P. Colella. The numerical simulation of two-dimensional fluid flow with strong shocks. *Journal of computational physics*, 54(1):115–173, 1984.

[127] G. B. Wright and B. Fornberg. Stable computations with flat radial basis functions using vector-valued rational approximations. *Journal of Computational Physics*, 331:137–156, 2017.

[128] H. C. Yee, N. D. Sandham, and M. J. Djomehri. Low-dissipative high-order shock-capturing methods using characteristic-based filters. *Journal of computational physics*, 150(1):199–238, 1999.

[129] N. Zeoli and S. Gu. Computational validation of an isentropic plug nozzle design for gas atomisation. *Computational Materials Science*, 42(2):245–258, 2008.

[130] L. Zhang, W. Liu, L. He, and X. Deng. A class of hybrid dg/fv methods for conservation laws iii: Two-dimensional euler equations. *Communications in Computational Physics*, 12(1):284–314, 2012.

[131] L. Zhang, L. Wei, H. Lixin, D. Xiaogang, and Z. Hanxin. A class of hybrid dg/fv methods for conservation laws i: Basic formulation and one-dimensional systems. *Journal of Computational Physics*, 231(4):1081–1103, 2012.

[132] L. Zhang, L. Wei, H. Lixin, D. Xiaogang, and Z. Hanxin. A class of hybrid dg/fv methods for conservation laws ii: Two-dimensional cases. *Journal of Computational Physics*, 231(4):1104–1120, 2012.

[133] X. Zhang and C.-W. Shu. On maximum-principle-satisfying high order schemes for scalar conservation laws. *Journal of Computational Physics*, 229(9):3091–3120, 2010.

[134] X. Zhang and C.-W. Shu. On positivity-preserving high order discontinuous Galerkin schemes for compressible Euler equations on rectangular meshes. *Journal of Computational Physics*, 229(23):8918–8934, 11 2010.

[135] X. Zhang and C.-W. Shu. Maximum-principle-satisfying and positivity-preserving high-order schemes for conservation laws: survey and new developments. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 467:2752–2776, 2011.

[136] X. Zhang, Y. Xia, and C.-W. Shu. Maximum-principle-satisfying and positivity-preserving high order discontinuous Galerkin schemes for conservation laws on triangular meshes. *Journal of Scientific Computing*, 50(1):29–62, 2012.

[137] J. Zhu and C.-W. Shu. A new type of multi-resolution weno schemes with increasingly higher order of accuracy on triangular meshes. *Journal of Computational Physics*, 392:19–33, 2019.

Fabian Mönkeberg
Chemin de Bel-Orne 30A
1008 Prilly
Switzerland
fabian@moenkeberg.ch
+41 (0)76 723 05 49

Citizenship: Swiss & German
Date of birth: May 30, 1990

Curriculum Vitae

Education

PhD in Mathematics, EPFL Lausanne (CH) Development of high-order methods for conservation laws and its implementation for two-dimensional problems	**2016 - 2020**
Master of Science in Scientific Computing, TU Berlin (DE) Focus on Numerical Methods for Stochastic Processes, Partial Differential Equations and Asymptotic Analysis Final grade: Excellent	**2013 - 2016**
Master's thesis: Asymptotic and Numerical Methods for the Two-Dimensional Narrow Escape Problem for Finite-Size Particles, Grade: Excellent	**Nov. 2015**
Bachelor of Science in Mathematics, ETH Zürich (CH) Focus on Numerical Mathematics (finite element methods, finite volume methods, finite difference methods, parallel numerical computing) Final grade: 5.17 (Swiss grading system)	**2009 - 2013**
Bachelor's thesis: Finite Volume Methods for Fluid Flow in Porous Media Grade: 5.00 (Swiss grading system)	**May 2012**

Professional Experience

SCOR Switzerland AG, Trainee (7 months)	**2016**
Bombardier Transportation AG, internship (6 months)	**2013**

Publications

J. S. Hesthaven, F. Mönkeberg, **Entropy stable essentially nonoscillatory methods based on RBF reconstructions,** ESAIM: M2AN, 53(3):925–958, 2019

J. S. Hesthaven, F. Mönkeberg, S. Zaninelli, **Rbf based cweno method,** Springer, Spectral and High Order Methods for PDEs ICOSAHOM 2018, 2018, Accepted

J. S. Hesthaven, F. Mönkeberg, **Two-dimensional RBF-ENO method on unstructured grids,** Journal of Scientific Computing, 82(3):1–24, 2020

J. S. Hesthaven, F. Mönkeberg, **Hybrid high-resolution RBF-ENO method SUBMITTED**

Conferences, workshops and invited talks

ICOSAHOM 2018 - "ENO Reconstruction Methods based on Radial Basis Functions", July, 2018, Imperial College London, London, UK
Seminar Groupe de Travail des Thésardes - "ENO Reconstruction Methods based on Radial Basis Functions", November, 2018, Sorbonne Université, Paris, FR
DRWA 2019 - "On RBF based methods for conservation laws", September, 2019, University of Verona, Trento, IT
AIM Week 2019 - Academia Industry Modeling Week, October, 2019, Computational Science Zurich PhD Program, Zurich CH

Personal Activities

Volleyball	**1998 - 2016**
- Berlin league VC Rotation Mitte, Germany	
- 1. league Volero Zürich, Switzerland	
- Referee (3. - 5. league)	
- Coaching children and adults (Jugend+Sport coaching certificate)	
Bicycle touring	**since 2011**

Language Skills

German	**Native**
English	**Professional**
French	**Advanced**

Computer Skills

Matlab/Simulink	**10+ years**
C++	**10+ years**
C#	**7 months**
Parallel computing	
- Message Passing Interface (MPI)	
- Open Multi-Processing (OpenMP)	
Python	**4+ years**
Linux, Windows	
MS Office	

References

Prof. Dr. Jan S. Hesthaven	jan.hesthaven@epfl.ch
Thesis Director	+41 (0)79 703 48 36
EPFL Lausanne	
Ake Wallin	a.wallin@hispeed.ch
Head of IT Development	+41 (0)44 701 11 33
SCOR Switzerland AG	

Lausanne, July 28, 2020 **Fabian Mönkeberg**