Mathematical Modelling and Numerical Analysis
Modélisation Mathématique et Analyse Numérique

Will be set by the publisher

ENTROPY STABLE ESSENTIALLY NONOSCILLATORY METHODS BASED ON RBF RECONSTRUCTION

Jan S. Hesthaven[1] and Fabian Mönkeberg[1]

Abstract. To solve hyperbolic conservation laws on general grids we propose to use high-order essentially nonoscillatory methods based on radial basis functions. 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. Thus, we show that such methods based on radial basis functions are as powerful as methods based on polynomial reconstruction. The main contribution is the construction of an algorithm and a smoothness indicator that ensures an interpolation function which fulfills the sign-property.

1991 Mathematics Subject Classification. 35L65, 65M12, 65M06, 65M08, 65D05.

August 13, 2018.

1. Introduction

Conservation laws arise in different fields of physics to describe systems with particular conserved properties, e.g. mass, momentum and energy. A change in these properties within a domain can be described by the flux through its boundaries. The one-dimensional conservation law is of the form

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

with the conserved variables $u : \mathbb{R} \times \mathbb{R}_+ \to \mathbb{R}^N$ and the flux $f : \mathbb{R}^N \to \mathbb{R}^N$. A well known and challenging property is the formation of discontinuities out of smooth initial data [23]. Thus, solutions need to be defined in the weak (distributional) sense. Since the weak solutions are not unique they need to be restricted by additional conditions. Let η be a convex scalar function (*entropy function)* such that there exists the *entropy flux* q with $\nabla_u q = \nabla_u \eta \nabla_u f$.
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

$$\eta(u)_t + q(u)_x \le 0, \tag{1.2}$$

is satisfied in the weak sense. In the case of scalar conservation laws, existence and uniqueness of the weak entropy solution in $\mathbb{R}^d$ was shown by Kružkov [20]. Furthermore, we can use the concept of the *entropy variables*

Keywords and phrases: radial basis functions, entropy stability, sign-property, finite differences, finite volume methods, high-order accuracy, ENO reconstruction

[1] SB-MATHICSE-MCSS, École Polytechnique Fédérale de Lausanne (EPFL), 1015 Lausanne, Switzerland
(e-mail: jan.hesthaven@epfl.ch, e-mail: fabian.monkeberg@epfl.ch)

$v \coloneqq (\nabla_u \eta)^T$ to symmetrize (1.1) in the sense that $\nabla_v u(v)$ is symmetric positive definite and $\nabla_v f(u(v))$ is symmetric. This can be seen by introducing the *entropy potential* $\psi(v) = v^T \cdot f(u(v)) - q(u(v))$ to end up with $\nabla_v f(u(v)) = \nabla_{vv}\psi$ [26].
Note that entropy solutions satisfy

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

in smooth regions, but satisfy (1.2) at discontinuities as entropy has to dissipate.

1.1. Finite Difference and Finite Volume Methods

One goal of numerical methods is to express the approximative behaviour of the physical correct solution. Let us assume a one-dimensional grid $\{x_i\}_{i\in\mathbb{Z}} \subset \mathbb{R}$, partitioned into cells $C_i = [x_{i-1/2}, x_{i+1/2}]$. The finite difference approach is based on approximating derivatives in space at x_i as

$$\frac{\partial f}{\partial x}(u_i) = \frac{F_{i+1/2} - F_{i-1/2}}{\Delta x} + \mathcal{O}(\Delta x^p), \tag{1.4}$$

with $p > 0$. This results in the semi-discrete scheme

$$\frac{\mathrm{d}u_i}{\mathrm{d}t} + \frac{1}{\Delta x}(F_{i+1/2} - F_{i-1/2}) = 0, \tag{1.5}$$

where the numerical flux terms $F_{i+1/2}$ depend on point values $\{u_{i-k}, \dots, u_{i+p-k}\}$ with $0 \le k \le p-1$.
On the other hand, finite volume methods work with mean values $\bar{u}_i$ for the cells C_i. By integrating (1.1) over the cells and dividing it by its size $|C_i|$ we recover (1.5) with the difference that

$$f(u(x_{i+1/2})) = F_{i+1/2} + \mathcal{O}(\Delta x^p). \tag{1.6}$$

In both cases, we can apply an arbitrary time discretization technique to receive a fully discrete scheme, e.g. Euler, SSPRK method.
There exist multiple high-order accurate methods to solve conservation laws, for example the MUSCL scheme introduced in [36], the ENO scheme [18] or the WENO scheme [34].
In [10], Fjordholm et al. proposed an entropy stable TeCNO scheme based on polynomial reconstruction. We follow the spirit of this work and introduce a scheme based on radial basis functions (RBF). The goal is to develop a scheme with the advantages of the RBFs in higher dimensions.

2. Entropy Stable Methods

The goal is to construct methods that fulfill a discrete version of (1.2), these are called *entropy stable*. As a first step, we introduce entropy conservative methods that fulfill (1.3) at the discrete level. Next, we add a dissipation term to control oscillations at discontinuities to recover an entropy stable method.

2.1. Entropy Conservative Methods

A finite difference method is *entropy conservative* if it satisfies

$$\frac{\mathrm{d}}{\mathrm{d}t}\eta(u_i) = -\frac{1}{|C_i|}[Q_{i+1/2} - Q_{i-1/2}], \tag{2.1}$$

for a consistent numerical entropy flux $Q_{i+1/2}$. To construct entropy conservative methods we use Tadmor's entropy conservation condition [35]

$$[\![v]\!]^T_{i+1/2} F_{i+1/2} = [\![\psi]\!]_{i+1/2}. \tag{2.2}$$

This condition describes a system of equations, but its solvability is not clear. For scalar conservation laws there exists a unique solution as can be summarized in the following theorem.

Theorem 2.1 (Entropy conservative schemes for scalar equations [35]). *For a given entropy pair (η, q) the numerical flux*

$$F_{i+1/2} = \begin{cases} \frac{[\![\psi]\!]_{i+1/2}}{[\![v]\!]_{i+1/2}} & \text{if } u_i \neq u_{i+1} \\ f(u_i) & \text{if } u_i = u_{i+1} \end{cases}, \tag{2.3}$$

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

Given a numerical second order two-point flux the idea of Lefloch et al. is to combine them linearly to construct a $2p$-th order accurate flux.

Theorem 2.2 (High-order entropy conservative fluxes [22]). *Let $p \in \mathbb{N}$ and assume that $\alpha_{1,p}, \dots, \alpha_{p,p}$ solve the p linear equations*

$$\sum_{i=1}^{p} i\alpha_{i,p} = 1, \qquad \sum_{i=1}^{p} i^{2s-1}\alpha_{i,p} = 0, \qquad \text{for } s = 2, \dots, p. \tag{2.4}$$

Then the flux

$$\tilde{F}^{2p}(u_{i-p+1}, \dots, u_{i+p}) = \sum_{j=1}^{p} \alpha_{j,p} \sum_{l=1}^{j} \tilde{F}^2(u_{i-j+l}, u_{i+l}), \tag{2.5}$$

is consistent, $2p$-th order accurate and entropy conservative with the second order two-point conservative flux $\tilde{F}^2$ fulfilling (2.2).

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.

2.1.1. *Entropy Conservative Methods for Shallow Water Equations*

The shallow water equations describe a flow under the assumptions that the horizontal length scales are much larger than the vertical ones. In one space dimension the system of equations depends on the mass flow m and the fluid height h

$$\begin{pmatrix} h \\ m \end{pmatrix}_t + \begin{pmatrix} m \\ \frac{1}{2}gh^2 + m^2/h \end{pmatrix}_x = 0, \tag{2.6}$$

with the gravitational constant g [23]. To apply Theorem 2.2 we need to construct a second order entropy conservative scheme by solving (2.2). One choice of an entropy pair for the one-dimensional shallow water equation is

$$\eta = \frac{1}{2}\Big(\frac{m^2}{h} + gh^2\Big), \qquad q = \frac{m^3}{h^2} + gmh, \tag{2.7}$$

which results in the entropy variables and the potential

$$v = \begin{pmatrix} gh - \frac{m^2}{2h^2} \\ \frac{m}{h} \end{pmatrix}, \qquad \psi = \frac{1}{2}gmh. \tag{2.8}$$

An alternative second order entropy conservative flux is

$$\tilde{F}_{i+1/2} = \begin{pmatrix} \bar{h}_{i+1/2}\bar{u}_{i+1/2} \\ \bar{h}_{i+1/2}(\bar{u}_{i+1/2})^2 + \frac{1}{2}g\overline{h^2}_{i+1/2} \end{pmatrix}, \tag{2.9}$$

with $u = m/h$ and $\bar{f}_{i+1/2} = \frac{1}{2}(f_i + f_{i+1})$ [9].

2.1.2. *Entropy Conservative Methods for Euler Equations*

The Euler equations can be recovered from the Navier-Stokes equations by neglecting the viscosity. They consist of the continuity equation, momentum equation and the conservation law for the total energy. In one dimension they are

$$\begin{pmatrix} \rho \\ m \\ E \end{pmatrix}_t + \begin{pmatrix} m \\ \frac{m^2}{\rho} + p \\ \frac{m}{\rho}(E+p) \end{pmatrix}_x = 0, \tag{2.10}$$

with $p = \mathcal{R}\rho T = (\gamma - 1)(E - \frac{1}{2}\frac{m^2}{\rho})$ for an ideal gas with the ratio of specific heat γ [19]. For the Euler equations the thermodynamical entropy $s = \log(p) - \gamma\log(\rho)$ is different from the entropy function and the entropy flux. One possible pair can be found in [10]

$$\eta = \frac{-\rho s}{\gamma - 1}, \qquad q = \frac{-ms}{\gamma - 1}. \tag{2.11}$$

Chandrashekar [5] proposed the kinetic energy preserving and entropy conservative (KEPEC) flux, based on the entropy variables and the potential

$$v = \begin{pmatrix} \frac{\gamma - s}{\gamma - 1} - \frac{\rho u^2}{2p} \\ \rho u / p \\ -\rho / p \end{pmatrix}, \qquad \psi = \rho u. \tag{2.12}$$

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

$$f^\rho = \hat{\rho}\bar{u}, \qquad f^m = \frac{\bar{\rho}}{2\bar{\beta}} + \bar{u}f^\rho, \qquad f^e = \Big(\frac{1}{2(\gamma-1)\hat{\beta}} - \frac{1}{2}\overline{u^2}\Big)f^\rho + \bar{u}f^m, \tag{2.13}$$

where $\bar{v} = \frac{v_{i+1} + v_i}{2}$.

2.2. **Entropy Stable Methods**

Entropy conservative methods result in 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 in the interface, controls these oscillations.
Based on an entropy conservative scheme $\tilde{F}_{j+1/2}$ of second order and a symmetric positive definite matrix $D_{i+1/2}$, Tadmor constructs the entropy stable numerical flux function [35]

$$F_{i+1/2} = \tilde{F}_{j+1/2} - \frac{1}{2}D_{i+1/2}[\![v]\!]_{i+1/2}. \tag{2.14}$$

Combining high-order conservative fluxes with dissipation terms introduces the problem that $D_{i+1/2}[\![v]\!]_{i+1/2} = \mathcal{O}(\Delta x^p)$ to maintain accuracy for smooth solutions.
For each cell C_i we define a stencil of cells S_i on which we construct an interpolation function $s_i(x)$ 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

$$F_{i+1/2} = \tilde{F}^{2p}_{i+1/2} - \frac{1}{2}D_{i+1/2}\langle\!\langle v \rangle\!\rangle_{i+1/2}, \tag{2.15}$$

with the additional condition

$$D_{i+1/2} = R_{i+1/2}\Lambda_{i+1/2}R^T_{i+1/2}, \tag{2.16}$$

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

Lemma 2.3 (Entropy stability with high-order diffusion [10])**.** *For each $i \in \mathbb{Z}$, let* (2.16) *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 $B_{i+1/2} \geq 0$ such that*

$$\langle\!\langle v \rangle\!\rangle_{i+1/2} = R_{i+1/2}^{-T} B_{i+1/2} R_{i+1/2}^{T} [\![v]\!]_{i+1/2}. \tag{2.17}$$

Then the scheme with the flux (2.15) *is entropy stable.*

By introducing the *scaled entropy variables*

$$w_i^{\pm} = R_{i\pm1/2}^{T} v_i, \qquad \tilde{w}_i^{\pm} R_{i\pm1/2}^{T} v_i^{\pm}, \tag{2.18}$$

with the reconstructed entropy variables $v_i^{\pm} = s_i(x_{i\pm1/2})$, condition (2.17) becomes

$$\langle\!\langle \tilde{w} \rangle\!\rangle_{i+1/2} = B_{i+1/2} [\![w]\!]_{i+1/2}. \tag{2.19}$$

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

$$\operatorname{sgn} \langle\!\langle \tilde{w}^l \rangle\!\rangle_{i+1/2} = \operatorname{sgn} [\![w^l]\!]_{i+1/2}, \tag{2.20}$$

for each component l. This structural property of the reconstruction is called the *sign-property*.

2.3. **Entropy Stable Finite Volume Methods**

The setting of one dimensional finite volume methods differs only slightly from the finite difference scheme, i.e. we are considering cell-average values $\bar{u}_i$ instead of point values and the definition of higher order methods changes to

$$F_{i+1/2} = f(u(x_{i+1/2})) + \mathcal{O}(\Delta x^p). \tag{2.21}$$

Nevertheless, given a 2-point second order finite difference flux F, it is also a second order accurate finite volume flux.

Lemma 2.4. *Assume F is a 2-point second order finite difference flux. Then, F is also a second order finite volume flux.*

Proof. We know that

$$F_{i+1/2} = \Delta x \frac{\mathrm{d} f(u(x_i))}{\mathrm{d}x} + F_{i-1/2} + \mathcal{O}(\Delta x^2). \tag{2.22}$$

By using the Taylor expansion for all the terms it follows

$$F_{i+1/2} = \Delta x \frac{\mathrm{d} f(u(x_i))}{\mathrm{d}x} + F_{i-1/2} + \mathcal{O}(\Delta x^2) = f(u(x_{i+1/2})) + \mathcal{O}(\Delta x^2). \tag{2.23}$$

□

Since the definition of entropy conservative schemes does not change for finite volume methods, Lemma 2.4 allows us to conclude that a second order finite difference flux that fulfills (2.2) is also a second order entropy conservative finite volume method. This can be summarized as follows.

Theorem 2.5. *Every second order finite difference scheme fulfilling Tadmor's entropy conservation condition* (2.2) *in one space dimension is a second order entropy conservative finite volume method.*

The construction of entropy stable schemes from entropy conservative schemes works as for the finite difference case. The only difference being that the interpolation is based on cell averages instead of point values.
Thus, Lemma 2.3 holds as well for finite volume methods and we recover an entropy stable finite volume method of the form

$$F_{i+1/2} = F_{i+1/2}^{2} - D_{i+1/2} \langle\!\langle v \rangle\!\rangle_{i+1/2}. \tag{2.24}$$

3. Radial Basis Functions

Radial basis functions (RBF) are 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 Duchon [7] and Micchelli [25], RBFs are successfully used in different domains.

3.1. **Basic Interpolation**

The goal is the interpolation of a data vector $f|_X = (f(x_1), \dots, f(x_n))^T \in \mathbb{R}^n$ at a scattered set of data points $X = (x_1, \dots, x_n)^T$ with $x_i \in \mathbb{R}^d$ for some function $f : \mathbb{R}^d \to \mathbb{R}$. The basic idea is to use one univariate continuous function ϕ, the *radial basis function*, composed with the Euclidean norm centered at the data points as the interpolation basis

$$\mathcal{B} = \{\phi(\varepsilon\|x - x_1\|), \dots, \phi(\varepsilon\|x - x_n\|)\}, \tag{3.1}$$

with the shape parameter ε. To reduce the complexity we use the notation

$$\phi(x - x_i) := \phi(\varepsilon\|x - x_i\|), \qquad \phi : \mathbb{R}^d \to \mathbb{R}. \tag{3.2}$$

The standard radial basis function approximation can be written as

$$s(x) = \sum_{i=1}^{n} a_i \phi(x - x_i) + p(x), \tag{3.3}$$

with a polynomial $p \in \Pi_{m-1}(\mathbb{R}^d)$, $m \in \mathbb{N}$, the interpolation condition

$$s(x_i) = f(x_i), \tag{3.4}$$

and the additional constraints

$$\sum_{i=1}^{n} a_i q(x_i) = 0, \qquad \text{for all } q \in \Pi_{m-1}(\mathbb{R}^d), \tag{3.5}$$

with the coefficients $a_i \in \mathbb{R}$ for all $i = 1, \dots, n$. Conditions (3.4) and (3.5) can be summarized in the system of equations

$$\begin{pmatrix} A & P \\ P^T & 0 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} f|_X \\ 0 \end{pmatrix}. \tag{3.6}$$

The choice of the radial basis function ϕ is restricted by some conditions to insure the solvability of (3.6).

Definition 3.1 (Conditionally positive function)**.** A function $\phi : \mathbb{R}^d \to \mathbb{R}$ is called conditionally positive (semi-) definite of order m if for any pairwise distinct points $x_1, \dots, x_n \in \mathbb{R}^d$ and $c = (c_1, \dots, c_n)^T \in \mathbb{R}^n \setminus \{0\}$ such that

$$\sum_{i=1}^{n} c_i p(x_i) = 0, \tag{3.7}$$

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

$$\sum_{j,k=1}^{n} c_j c_k \phi(x_j - x_k), \tag{3.8}$$

is positive (non-negative).

Wendland shows in [37] that for a conditionally positive definite RBF ϕ of order m (3.6) has a unique solution if $x_1, \dots, x_n$ are $\Pi_{m-1}(\mathbb{R}^d)$-unisolvent.

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 1. Commonly used RBFs with $\mathbb{N} \not\ni \nu > 0$, $k \in \mathbb{N}$ and $\varepsilon > 0$.

Definition 3.2 (Positive definite functions)**.** A function $\phi : \mathbb{R}^d \to \mathbb{R}$ is called positive definite if the quadratic form

$$\sum_{j,k=1}^{n} c_j c_k \phi(x_j - x_k), \tag{3.9}$$

is positive for any n pairwise different points $x_1, \dots, x_n \in \mathbb{R}^d$ and $c = (c_1, \dots, c_n)^T \in \mathbb{R}^n \setminus \{0\}$.

Note that in case of a positive definite function ϕ, the matrix A is positive definite and thus, there exists an unique solution a to the problem (3.6). Some examples of positive definite functions are the inverse multiquadratics and the Gaussians (Table 1). Other RBFs fulfill a slightly weaker condition and are conditionally positive definite of order k, e.g. multiquadratics, polyharmonic splines.

3.2. Interpolation of Cell-Averages

For the finite volume method we do not consider the pointwise interpolation, but cell-averages. Let us assume a given grid of cells $C_1, \dots, C_n$ with its average values $\bar{u}_1, \dots, \bar{u}_n$ for $n \in \mathbb{N}$. Based on [1, 2] we consider

$$s(x) = \sum_{i=1}^{n} a_i \lambda_{C_i}^{\xi} \phi(x - \xi) + p(x), \qquad p \in \Pi_{m-1}(\mathbb{R}^d), \tag{3.10}$$

with the average operator of f over the cell C, $\lambda_C^{\xi} f$, such that

$$\lambda_{C_j} s = \bar{u}_j, \qquad \text{for all } j = 1, \dots, n, \tag{3.11a}$$

$$\sum_{i=1}^{n} a_i \lambda_{C_i}(q) = 0, \qquad \text{for all } q \in \Pi_{m-1}(\mathbb{R}^d). \tag{3.11b}$$

To show solvability of system (3.11) it suffices to assume ϕ to be conditional positive definite in a pointwise sense. Aboiyar et al. claim in [1] that (3.11) has a unique solution if the set $\{\lambda_{C_i}\}_{i=1}^n$ is $\Pi_{m-1}(\mathbb{R}^d)$-unisolvent.

Theorem 3.3 (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 $\Pi_{m-1}(\mathbb{R}^d)$-unisolvent with $n \in \mathbb{N}$. Then, the problem* (3.11) *has a unique solution.*

The proof closely follows the one for the pointwise evaluation in [37] plus an estimate for the positive definiteness, based on a pointwise result in [29].

3.3. 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 [31]. It describes the trade-off between the well-known properties that flat infinitely smooth RBFs ($\varepsilon \to 0$) have an increasing approximation power but a decreasing numerical stability

due to ill-conditioning of the interpolation matrix [6, 21, 33].
To overcome the issue of ill-conditioning there are multiple propositions for choosing an 'optimal' shape parameter [8, 30]. Note that a continuous scaling $\varepsilon = \alpha n^{-1/d}$ causes stagnation errors [4]. However, there are multiple approaches which overcome this problem: the RBF-CP [15], the RBF-QR [14], and the RBF-GA [13]. Furthermore, there is the vector-valued rational approximation method (RBF-RA), based on the RBF-CP algorithm and introduced in [38].

3.3.1. *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 $j = 1, \dots, M$ are analytical 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 $|\epsilon| \geq \epsilon_R > 0$, where $|\varepsilon| \leq \varepsilon_R$ is in Ω,
(iii) $\epsilon = 0$ 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 Padé approximant

$$r_j(\varepsilon) = \frac{\sum_{i=0}^{m} a_{i,j}\varepsilon^{2i}}{1 + \sum_{i=1}^{n} b_i \varepsilon^{2i}} \approx f_j(\varepsilon), \tag{3.12}$$

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

$$\begin{bmatrix} 1 & \varepsilon_1^2 & \dots & \varepsilon_1^{2m} \\ 1 & \varepsilon_2^2 & \dots & \varepsilon_2^{2m} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \varepsilon_K^2 & \dots & \varepsilon_K^{2m} \end{bmatrix} \begin{bmatrix} a_{0,j} \\ \vdots \\ a_{m,j} \end{bmatrix} + \operatorname{diag}(-f_j) \begin{bmatrix} \varepsilon_1^2 & \dots & \varepsilon_1^{2n} \\ \varepsilon_2^2 & \dots & \varepsilon_2^{2n} \\ \vdots & \ddots & \vdots \\ \varepsilon_K^2 & \dots & \varepsilon_K^{2n} \end{bmatrix} \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix} = \begin{bmatrix} f_j(\varepsilon_1) \\ \vdots \\ f_j(\varepsilon_K) \end{bmatrix}, \tag{3.13}$$

with $(m+1)M + n$ unknown coefficients. We write (3.13) as

$$E a_j + F_j b_j = f_j, \tag{3.14}$$

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

3.3.2. *RBF-RA*

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

$$f(\varepsilon) = \begin{bmatrix} s(\hat{x}_1, \varepsilon) \\ \vdots \\ s(\hat{x}_M, \varepsilon) \end{bmatrix} = \underbrace{\begin{bmatrix} \phi_\varepsilon(\|\hat{x}_1 - x_1\|) & \cdots & \phi_\varepsilon(\|\hat{x}_1 - x_n\|) \\ \vdots & \ddots & \vdots \\ \phi_\varepsilon(\|\hat{x}_M - x_1\|) & \cdots & \phi_\varepsilon(\|\hat{x}_M - x_n\|) \end{bmatrix}}_{\Phi(\varepsilon)} \begin{bmatrix} \\ A(\varepsilon)^{-1} \\ \\ \end{bmatrix} f|_X. \tag{3.16}$$

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 around 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 compact domains there are at most a finite number of isolated poles. Furthermore, all entries

Algorithm 1 Vector-valued rational approximation [38]

(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 (3.13) with the Hermitian transpose Q^* from the left to obtain

$$\begin{pmatrix} R \\ 0 \end{pmatrix} a_j + Q^* F_j b = Q^* f_j. \tag{3.15}$$

(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 $j = 1, \dots, M$.

of $s(\varepsilon)$ share the same poles since they are all dependent on $A(\varepsilon)^{-1}$. The symmetry condition is fulfilled because of the following properties of the RBFs

$$\phi(\varepsilon r) = \phi(-\varepsilon r) = \overline{\phi(\bar{\varepsilon} r)} = \overline{\phi(-\bar{\varepsilon} r)}. \tag{3.17}$$

Further, the condition that $\varepsilon = 0$ is a removable singularity is typically and, in the case of the Gaussian kernel always, is fulfilled [16, 33].
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 an 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 on 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 [15, 38].
Parameter Choice. For the choice of the evaluation radius there are two different strategies proposed [38], depending on the type of the kernel. 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

$$\tilde{\sigma}_\infty(A(\beta)) = \|A(i\beta)\|_\infty \|A(\beta)^{-1}\|_\infty. \tag{3.18}$$

For other kernels we choose ε_R smaller than the smallest distance to a singularity

$$\varepsilon_R = 0.95(\max_{i,j \le N} \|x_i - x_j\|)^{-1}. \tag{3.19}$$

In some cases with small distances this seems to give too big values. Then, we choose the minimum of (3.19) and the approximated real value ε such that $\operatorname{cond}(A(\varepsilon)) \approx 10^6$.
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.

3.4. **Explicit Formula of the RBF Interpolation**

Let us consider the pointwise RBF interpolation problem (3.6) with the interpolation function (3.3). Furthermore, let $x_1, \dots, x_n$ be the grid points such that $x_i < x_{i+1}$ and $n \in \mathbb{N}$ and $y_1, \dots, y_n \in \mathbb{R}$ its values. We are

looking for an RBF interpolation function

$$s(x) = \sum_{i=1}^{n} a_i \phi(x - x_i) + \sum_{j=1}^{m} b_j L_j(x), \tag{3.20}$$

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

Lemma 3.4 (Explicit RBF solution formula)**.** *The interpolation problem* (3.4) *and* (3.5) *can be solved using an explicit formula if we choose an RBF interpolation ansatz with a conditional positive definite RBF of order smaller than* $n - 1$

$$s(x) = \alpha \mathrm{d}\varphi(x) + \sum_{i=1}^{n-1} y_i L_i(x), \tag{3.21}$$

where $\alpha = \frac{y_N - \sum_{i=1}^{n-1} y_i L_i(x_n)}{\mathrm{d}\varphi(x_n)}$, $\mathrm{d}\varphi(x) = \varphi(x) - \sum_{j=1}^{n-1} \varphi(x_j) L_j(x)$ *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

$$a_j = -a_n L_j(x_n), \qquad \text{for } j = 1, \dots, n-1. \tag{3.22}$$

This yields the interpolation function

$$s(x) = \alpha \varphi(x) + \sum_{j=1}^{n-1} b_j L_j(x), \tag{3.23}$$

with $\alpha = a_n$. This interpolation function solves the reduced interpolation problem

$$\alpha \varphi(x_i) + \sum_{j=1}^{n-1} b_j L_j(x_i) = y_i, \qquad \text{for } i = 1, \dots, n. \tag{3.24}$$

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

$$\alpha = \frac{y_n - \sum_{i=1}^{n-1} y_i L_i(x_n)}{\mathrm{d}\varphi(x_n)}, \tag{3.25a}$$

$$b_j = y_j - \alpha \varphi(x_j), \qquad \text{for } j = 1, \dots, n-1. \tag{3.25b}$$

□

Remark 3.5. *We can express* $\mathrm{d}\varphi$ *in terms of projections*

$$\mathrm{d}\varphi(x) := \Psi(x, x_i) = (Id - \mathcal{P}^x)(Id - \mathcal{P}^y)[\phi(x - y))]|_{y = x_i}, \tag{3.26}$$

where the operators $\mathcal{P}^z$ *is the projection of the variable* z *on the polynomial space of dimension* $n-1$*. Schaback [32] shows that* Ψ *is positive definite on* $\mathbb{R}^d \setminus \{x_1, \dots, x_{n-1}\}$*. Thus, it is closely related to reproducing kernels and its native spaces, introduced in [32].*

Remark 3.6. *Note that this representation is independent to permutations of the indices. In general we can choose* $\tilde{y}_n = y_j$ *and* $\tilde{y}_i \in \{y_l | \, l \neq j\}$.

4. Smoothness Indicator for RBF Interpolation Functions

In essentially nonoscillatory (ENO)- and weighted ENO (WENO)-type methods it is essential 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 in an ENO method we need something similar. However, the divided differences, used in the standard Newton's interpolation formula, are valid only for polynomials.

4.1. **Generalized Divided Differences**

For non-polynomial basis functions Mühlbach [27] introduces generalized divided differences, which coincide in the monomial case with the standard one. The result of Mühlbach is based on functions $f_1, \dots, f_n$ that form a Chebyshev system, thus they satisfy

$$\begin{vmatrix} f_1(z_1) & \cdots & f_1(z_k) \\ \vdots & & \vdots \\ f_k(z_1) & \cdots & f_k(z_k) \end{vmatrix} \neq 0, \tag{4.1}$$

for all choices of distinct points $z_1, \dots, z_k$ and for $k = 0, \dots, n$.

Theorem 4.1 (Generalized Newton's interpolation formula [28])**.** *Let $f_1, \dots, f_n$ from $\mathbb{R} \to \mathbb{R}$ 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*

$$\mathrm{p}f \begin{bmatrix} f_1, \dots, f_n \\ x_1, \dots, x_n \end{bmatrix} = \sum_{k=1}^{n} \begin{bmatrix} f_1, \dots, f_k \\ x_1, \dots, x_k \end{bmatrix} \cdot g_k, \tag{4.2}$$

where

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

and

$$\begin{bmatrix} f_1, \dots, f_k \\ x_1, \dots, x_k \end{bmatrix} = \frac{\begin{bmatrix} f_1, \dots, f_{k-1} \\ x_2, \dots, x_k \end{bmatrix} - \begin{bmatrix} f_1, \dots, f_{k-1} \\ x_1, \dots, x_{k-1} \end{bmatrix}}{\begin{bmatrix} f_1, \dots, f_{k-1} \\ x_2, \dots, x_k \end{bmatrix} - \begin{bmatrix} f_1, \dots, f_{k-1} \\ x_1, \dots, x_{k-1} \end{bmatrix}}, \quad \text{for } k \geq 2, \tag{4.3}$$

and

$$r_n f = \mathrm{r}f \begin{bmatrix} f_1, \dots, f_n \\ x_1, \dots, x_n \end{bmatrix} = f - \mathrm{p}f \begin{bmatrix} f_1, \dots, f_n \\ x_1, \dots, x_n \end{bmatrix}. \tag{4.4}$$

The initial conditions are

$$\begin{bmatrix} f_1 \\ x_j \end{bmatrix} = \frac{f(x_j)}{f_1(x_j)}. \tag{4.5}$$

Based on this theorem we can express the generalized divided differences for the basis $\{1, x, \dots, x^{N-2}, \varphi\}$ for $N \in \mathbb{N}$ with φ from Lemma 3.4 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 4.2. *Let the basis be given by* $\{1, x, \dots, x^{N-2}, \varphi\}$ *for* $N \in \mathbb{N}$ *and* φ *defined in Lemma 3.4. We recover the generalized divided differences of the form*

$$\left[\begin{matrix} 1 \\ x_0 \end{matrix} \,\middle|\, f \right] = f(x_0) = y_0, \tag{4.6}$$

$$\left[\begin{matrix} 1, x, \dots, x^k \\ x_1, x_2, \dots, x_{k+1} \end{matrix} \,\middle|\, f \right] = \frac{y_{k+1} - \sum_{i=1}^{k} y_i L_i^{k+1,k-1}(x_{k+1})}{\prod_{i=1}^{k}(x_{k+1} - x_i)}, \qquad \text{for } k < N-1, \tag{4.7}$$

$$\left[\begin{matrix} 1, x, \dots, x^{N-2}, \varphi \\ x_1, x_2, \dots, x_{N-1}, x_N \end{matrix} \,\middle|\, f \right] = \frac{y_N - \sum_{i=1}^{N-1} y_i L_i^{N,N-2}(x_N)}{\varphi(x_N) - \sum_{i=1}^{N-1} \varphi(x_i) L_i^{N,N-2}(x_N)}. \tag{4.8}$$

If we compare this results with the RBF interpolation in Lemma 3.4, we see that the last divided difference can be written as

$$\left[\begin{matrix} 1, x, \dots, x^{N-2}, \varphi \\ x_1, x_2, \dots, x_{N-1}, x_N \end{matrix} \,\middle|\, f \right] = \alpha. \tag{4.9}$$

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

4.2. **Relation to Reproducing Kernel Hilbert Spaces and its Norm**

As mentioned above there is a close relation to native spaces of conditionally positive definite functions (see Schaback [32]). Indeed, the RBF-based basis function $\mathrm{d}\varphi$ can be expressed in terms of the modified kernel function $\Psi(x, y) = (Id - \mathcal{P}^x)(Id - \mathcal{P}^y)[\phi(x - y))]$.
If we further analyse the norm of the interpolation function, based on the inner product of the native space, we get

Lemma 4.3. *Let* s *be an RBF-interpolation function given by* (3.21)*. Then, it has the norm*

$$\|s\|_\phi^2 = \sum_{i=1}^{N-1} s(x_i)^2 + \alpha^2 \mathrm{d}\varphi(x_N). \tag{4.10}$$

In particular, we have

$$\|s\|_\phi \approx \frac{\beta}{\mathrm{d}\varphi(x_N)^{1/2}}, \tag{4.11}$$

with $\beta = y_N - \sum_{i=1}^{N-1} y_i L_i(x_N)$.

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

Proof. The inner product of the native space is

$$(f, g)_\phi = \sum_{i=1}^{N-1} f(x_i) g(x_i) + (f - \mathcal{P}f, g - \mathcal{P}g)_{\phi,0}, \tag{4.12}$$

with

$$(f, g)_{\phi,0} = \sum_{j=1}^{M} \sum_{k=1}^{N} \lambda_j \mu_k \phi(x_j, y_k), \tag{4.13}$$

for $f = \sum_{j=1}^{M} \lambda_j \phi(x, x_j)$ and $g = \sum_{k=1}^{N} \mu_k \phi(x, y_k)$ [32].
We have $(s - \mathcal{P}s)(x) = \beta \frac{\mathrm{d}\varphi(x)}{\mathrm{d}\varphi(x_N)}$ and

$$\|s\|_\phi^2 = \sum_{j=1}^{N-1} s(x_j)^2 + \left(\frac{\beta}{\mathrm{d}\varphi(x_N)}\right)^2 (\mathrm{d}\varphi, \mathrm{d}\varphi)_\phi. \tag{4.14}$$

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 4.4. *Assuming Lemma 4.3, we have*

$$\mathrm{d}\varphi(x_N) > 0. \tag{4.15}$$

Lemma 4.5 (Equivalent Norm)**.** *The set defined by*

$$\mathcal{B} := \Big\{\varphi\Big\} \cup \Big\{\frac{\varphi(x_j)}{L_j(x_N)}L_j \,|j = 1,\dots,N-1\Big\}, \tag{4.16}$$

is a basis of 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 $s(x) = \alpha_N\varphi(x) + \sum_{i=1}^{N-1}\alpha_i\frac{\varphi(x_j)}{L_j(x_N)}L_j(x)$.

Proof. From the interpolation (3.21) we directly recover that $\mathcal{B}$ is a basis of the interpolation space. □

4.3. **Smoothness Indicator and Stencil Choice**

Harten et al. proposed the *Essentially Nonoscillatory method* to reduce spurious oscillations at discontinuities [18]. Its principle is based on the evaluation of multiple stencils for each cell C_i for which we construct the reconstruction. Finally, one chooses the least oscillatory reconstruction to define s_i. Fjordholm et al. showed in [11] the sign-property for the polynomial reconstruction method with the recursive algorithm introduced by Harten et al. which utilizes the last divided difference related to the highest derivative as a local smoothness indicator. A sign preserving WENO reconstruction method was proposed by Fjordholm et al. [12]. In the RBF reconstruction the highest derivative is similar to the RBF-part of the reconstruction found in Lemma 4.3 and Theorem 4.2. As we shall show, the recursive algorithm from the polynomial case, combined with the smoothness indicator

$$IS(s) = \frac{\beta}{\mathrm{d}\varphi(x_N)^{1/2}} \qquad \text{with } \beta = y_N - \sum_{i=1}^{N-1} y_i L_i(x_N), \tag{4.17}$$

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 from Corollary 4.4 the definition of $IS(s)$ is well defined.

Remark 4.6. *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 decrease the computational grid size.*

Remark 4.7. *The smoothness indicator* (4.17) *has an impractical and computationally expensive form. However, with Lemma 4.5 we recover*

$$\mathrm{d}\varphi(x_N) = \|\mathrm{d}\varphi\|_\phi^2 \approx \|\mathrm{d}\varphi\|_{\mathcal{B}}^2 = \Big(1 + \sum_{i=1}^{N-1} L_i(x_N)^2\Big). \tag{4.18}$$

Algorithm 2 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 $|IS(s(i+s_j-1, \dots, i+s_j+j))| < |IS(s(i+s_j, \dots, i+s_j+j+1))|$ **then**
    Set $s_{j+1} = s_j - 1$
  else
    Set $s_{j+1} = s_j$
  end if
end for
Define the stencil $S_i = \{C_{i+s_N}, \dots, C_{i+s_N+N-1}\}$.

Thus, we have equivalence of the smoothness indicator IS with

$$\widetilde{IS}(s) \coloneqq \frac{\beta}{\mathrm{d}\varphi(x_N)}\Big(1 + \sum_{i=1}^{N-1} L_i(x_N)^2\Big)^{1/2} = \Big(\sum_{i=1}^{N} a_i^2\Big)^{1/2}. \tag{4.19}$$

To choose the least oscillatory stencil S_i for the i-th cell for the RBF-reconstruction we follow Algorithm 2 which is based on the one from Harten et al. [18]. 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 4.8. *In the general case $N \geq M+1$ with a conditionally positive definite RBF of order M, we replace α by $\sqrt{\sum_{i=1}^{N} a_i^2}$ in Algorithm 2. In this case it is more difficult to prove the sign-property, but numerical examples suggest that it remains valid.*

5. 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 call them

$$S_i = \{C_{i+r_{N-1}}, \dots, C_{i+r_{N-1}+N-1}\}, \tag{5.1a}$$
$$S_{i+1} = \{C_{i+s_{N-1}+1}, \dots, C_{i+s_{N-1}+N}\}, \tag{5.1b}$$

where $r_{N-1} \leq 1 + s_{N-1}$ and C_j is the j-th cell with its mid-point x_j on which we apply the interpolation. Further, we define $d_{N-1} \coloneqq 1 + s_{N-1} - r_{N-1} \geq 0$ as the shift between the stencils. Note that the stencils are chosen by Algorithm 2 and that there are no constraints on the stencils.

5.1. **Notation**

For simplicity, we introduce some 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

$$\begin{aligned} L_j^i(x) &- \text{Lagrange polynomial of degree } N-1 \text{ such that} \\ &L_j^i(x_l) = \delta_{lj} \text{ for } l \in \{i-N+1, \dots, i-1\}, \end{aligned} \tag{5.2}$$

$$\varphi^j(x) := \phi(x - x_j) - \sum_{l=1}^{N-1} \phi(x - x_{j-N+l}) L_{j-N+l}^j(x_j), \tag{5.3}$$

$$\mathrm{d}\varphi^j(x) := \varphi^j(x) - \sum_{i=1}^{N-1} \varphi^j(x_i) L_i^j(x), \tag{5.4}$$

$$\alpha_j := \frac{\beta_j}{\mathrm{d}\varphi^j(x_j)}, \qquad \gamma_j := \frac{\beta_j}{\mathrm{d}\varphi^j(x_j)^{1/2}}, \qquad \beta_j := y_j - \sum_{i=1}^{N-1} y_{j-N+i} L_{j-N+i}^j(x_j). \tag{5.5}$$

5.2. **Representation of the Reconstructed Jumps**

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

$$\mathrm{jR}_{i+1/2} := s_{i+1}(x_{i+1/2}) - s_i(x_{i+1/2}), \tag{5.6}$$

for which we can 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 2. We now prove Theorem 5.1.

Theorem 5.1 (Generalized representation)**.** *The second and third degree reconstructed jump can be written in the following form*

$$\mathrm{jR}_{i+1/2} = \sum_{j=0}^{d_{N-1}-1} C_j (\gamma_{i+r_{N-1}+N+j} - \gamma_{i+r_{N-1}+N-1+j}) + \varepsilon(\Delta x), \tag{5.7}$$

with the constants

$$\begin{aligned} C_0 &= \frac{\mathrm{d}\varphi^k(x_{i+1/2})}{\delta_k} - A_k \delta_k, \\ C_j &= C_{j-1} - A_{k+j}\delta_{k+j} = \mathrm{d}\varphi^k(x_{i+1/2}) - \sum_{l=0}^{j} A_{k+l}\delta_{k+j}, \end{aligned} \tag{5.8}$$

and an error term

$$\varepsilon(\Delta x) = \gamma_{k+d_{N-1}} \Big(\frac{\mathrm{d}\varphi^{k+d_{N-1}}(x_{i+1/2})}{\delta_{k+d_{N-1}}} - C_{d_{N-1}-1} \Big), \tag{5.9}$$

where $k = i + r_{N-1} + N - 1$.

The proof relies on multiple Lemmas which we now develop.

Lemma 5.2. *Given the Lagrange polynomials. For* $N = 2, 3$ *it holds*

$$-\sum_{l=1}^{N-1} y_{j-N+l} L_{j-N+l}^j(x_{i+1/2}) = A_j \beta_j - \sum_{l=1}^{N-1} y_{j-N+l+1} L_{j-N+l+1}^{j+1}(x_{i+1/2}), \tag{5.10}$$

where

$$A_j = \frac{L_{j-N+1}^j(x_{i+1/2})}{L_{j-N+1}^j(x_j)}. \tag{5.11}$$

Proof. The case $N = 2$ is direct since the Lagrange polynomials are constant. Thus, (5.10) is

$$-y_{j-1} = \beta_j - y_j, \tag{5.12}$$

For $N = 3$, we write the left hand side of (5.10) and subtract $A_j\beta_j$

$$-\ y_{j-2}L^j_{j-2}(x_{i+1/2})\ -\ y_{j-1}L^j_{j-1}(x_{i+1/2})\ -\ A_j\beta_j \quad = \quad -y_{j-1}\Big(L^j_{j-1}(x_{i+1/2})\ -\ A_jL^j_{j-1}(x_j)\Big)\ -\ y_jA_j. \tag{5.13}$$

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

$$A_jL^j_{j-1}(x_j) - L^j_{j-1}(x_{i+1/2}) = \frac{(x_{i+1/2} - x_j)}{(x_j - x_{j-1})} = -L^{j+1}_{j-1}(x_{i+1/2}). \tag{5.14}$$

□

Lemma 5.3. *The reconstructed jump $jR_{i+1/2}$ for the second and third degree reconstruction method can be expressed as*

$$\mathrm{jR}_{i+1/2} = \frac{\gamma_{k+d_{N-1}}}{\delta_{k+d_{N-1}}}\mathrm{d}\varphi^{k+d_{N-1}}(x_{i+1/2}) - \frac{\gamma_k}{\delta_k}\mathrm{d}\varphi^k(x_{i+1/2}) + \sum_{j=0}^{d_{N-1}-1} A_{k+j}\gamma_{k+j}\delta_{k+j}, \tag{5.15}$$

where $k = i + r_{N-1} + N - 1$, $k + d_{N-1} = i + s_{N-1} + N$ and $\delta_i = \mathrm{d}\varphi^i(x_i)^{1/2}$.

Proof. From Lemma 3.4 and the selected stencils from (5.1) we rewrite the N-th degree reconstructed jump $\mathrm{jR}_{i+1/2}$ between cell i and $i+1$ as

$$\begin{aligned}\mathrm{jR}_{i+1/2} = {} & \alpha_{i+s_{N-1}+N}\mathrm{d}\varphi^{i+s_{N-1}+N}(x_{i+1/2}) - \alpha_{i+r_{N-1}+N-1}\mathrm{d}\varphi^{i+r_{N-1}+N-1}(x_{i+1/2}) \\ & + \sum_{j=1}^{N-1} y_{i+s_{N-1}+j}L^{i+s_{N-1}+N}_{i+s_{N-1}+j}(x_{i+1/2}) - \sum_{j=1}^{N-1} y_{i+r_{N-1}+j-1}L^{i+r_{N-1}+N-1}_{i+r_{N-1}+j-1}(x_{i+1/2}).\end{aligned} \tag{5.16}$$

The polynomial part of the reconstructed jump is

$$p_{i+1}(x_{i+1/2}) - p_i(x_{i+1/2}) = \sum_{j=0}^{d_{N-1}-1} A_{i+r_{N-1}+N-1+j}\beta_{i+r_{N-1}+N-1+j}, \tag{5.17}$$

by recursively applying Lemma 5.2. This yields

$$\mathrm{jR}_{i+1/2} = \alpha_{k+d_{N-1}}\mathrm{d}\varphi^{k+d_{N-1}}(x_{i+1/2}) - \alpha_k\mathrm{d}\varphi^k(x_{i+1/2}) + \sum_{j=0}^{d_{N-1}-1} A_{k+j}\mathrm{d}\varphi^{k+j}(x_{k+j})\alpha_{k+j}. \tag{5.18}$$

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

Lemma 5.4. *We have*

$$\begin{aligned}A_j\mathrm{d}\varphi^j(x_j) - \mathrm{d}\varphi^j(x_{i+1/2}) &= -(\varphi^j - \mathcal{P}^{N-1}_{j+1}\varphi^j)(x_{i+1/2}), \\ &= -\mathrm{d}\varphi^{j+1}(x_{i+1/2}) + \varepsilon_j(\Delta x),\end{aligned} \tag{5.19}$$

with $\mathcal{P}^k_{j+1}$ as the k-th degree polynomial approximation with respect to the interpolation points $x_j, \ldots, x_{j+1-k}$ and $\varepsilon_j(\Delta x) = \mathrm{d}\varphi^{j+1}(x_{i+1/2}) - (\varphi^j - \mathcal{P}^{N-1}_{j+1}\varphi^j)(x_{i+1/2})$.

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

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

In the case $N = 3$, we have

$$\begin{aligned}A_j\mathrm{d}\varphi^j(x_j) - \mathrm{d}\varphi^j(x_{i+1/2}) = \Big(&\varphi^j(x_j)L^j_{j-2}(x_{i+1/2}) - \varphi^j(x_{i+1/2})L^j_{j-2}(x_j)\\ &- \varphi^j(x_{j-1})L^j_{j-1}(x_j)L^j_{j-2}(x_{i+1/2}) - \varphi^j(x_{j-2})L^j_{j-2}(x_j)L^j_{j-2}(x_{i+1/2})\\ &+ \varphi^j(x_{j-1})L^j_{j-1}(x_{i+1/2})L^j_{j-2}(x_j) + \varphi^j(x_{j-2})L^j_{j-2}(x_{i+1/2})L^j_{j-2}(x_j)\Big)\frac{1}{L^j_{j-2}(x_j)},\end{aligned} \tag{5.20}$$

which can be simplified as

$$\begin{aligned}A_j\mathrm{d}\varphi^j(x_j) - \mathrm{d}\varphi^j(x_{i+1/2}) = \Big(&\varphi^j(x_j)L^j_{j-2}(x_{i+1/2}) - \varphi^j(x_{i+1/2})L^j_{j-2}(x_j)\\ &+ \varphi^j(x_{j-1})\Big(L^j_{j-1}(x_{i+1/2})L^j_{j-2}(x_j) - L^j_{j-1}(x_j)L^j_{j-2}(x_{i+1/2})\Big)\Big)\frac{1}{L^j_{j-2}(x_j)}.\end{aligned} \tag{5.21}$$

Next, we express the last term

$$L^j_{j-1}(x_{i+1/2})L^j_{j-2}(x_j) - L^j_{j-1}(x_j)L^j_{j-2}(x_{i+1/2}) = L^j_{j-2}(x_j) - L^j_{j-2}(x_{i+1/2}), \tag{5.22}$$

and insert this into (5.21)

$$A_j\mathrm{d}\varphi^j(x_j) - \mathrm{d}\varphi^j(x_{i+1/2}) = -\varphi^j(x_{i+1/2}) + \varphi^j(x_j)L^{j+1}_j(x_{i+1/2}) + \varphi^j(x_{j-1})L^{j+1}_{j-1}(x_{i+1/2}), \tag{5.23}$$

where we used that

$$L^{j+1}_j(x_{i+1/2}) = \frac{L^j_{j-2}(x_{i+1/2})}{L^j_{j-2}(x_j)}, \qquad L^{j+1}_{j-1}(x_{i+1/2}) = 1 - \frac{L^j_{j-2}(x_{i+1/2})}{L^j_{j-2}(x_j)}. \tag{5.24}$$

Finally, we add $\pm\mathrm{d}\varphi^{j+1}(x_{i+1/2})$ and recover

$$\begin{aligned}A_j\mathrm{d}\varphi^j(x_j) - \mathrm{d}\varphi^j(x_{i+1/2}) &= -\mathrm{d}\varphi^{j+1}(x_{i+1/2}) + \varphi^{j+1}(x_{i+1/2}) - \varphi^j(x_{i+1/2})\\ &\quad - \Big(\varphi^{j+1}(x_j) - \varphi^j(x_j)\Big)L^{j+1}_j(x_{i+1/2})\\ &\quad - \Big(\varphi^{j+1}(x_{j-1}) - \varphi^j(x_{j-1})\Big)L^{j+1}_{j-1}(x_{i+1/2}),\\ &= -\mathrm{d}\varphi^{j+1}(x_{i+1/2}) + \varepsilon_j(\Delta x).\end{aligned} \tag{5.25}$$

□

Corollary 5.5. *We have*

$$\mathrm{d}\varphi^j(x_{i+1/2}) - \sum_{k=0}^{l-1} A_{j+k}\mathrm{d}\varphi^{j+k}(x_{j+k}) = \mathrm{d}\varphi^{j+l}(x_{i+1/2}) - \sum_{k=0}^{l-1}\varepsilon_{j+k}(\Delta x). \tag{5.26}$$

Proof. This follows directly by applying Lemma 5.4 multiple times. □

Now, we are ready to prove Theorem 5.1.

Proof. (Theorem 5.1)
The goal is to show the equivalence with the representation in Lemma 5.3. Therefore, we insert (5.9) into (5.7) to have

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

Finally, we insert the definitions 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 5.6. *Note that the error $\varepsilon(\Delta x)$ can be written as*

$$\varepsilon(\Delta x) = \beta_{k+d}\sum_{i=0}^{d_{N-1}} \varepsilon_{k+i}(\Delta x)\frac{\delta_{k+i+1}}{\delta_{k+d}}, \tag{5.27}$$

with

$$\begin{aligned} \varepsilon_j(\Delta x) &= \frac{1}{\delta_{j+1}\delta_j}\Big(\mathrm{d}\varphi^{j+1}(x_{i+1/2})\frac{\delta_j}{\delta_{j+1}} - \mathrm{d}\varphi^j(x_{i+1/2}) + A_j\mathrm{d}\varphi^j(x_j)\Big), \\ &= \frac{1}{\delta_{j+1}\delta_j}\Big(\mathrm{d}\varphi^{j+1}(x_{i+1/2})\frac{\delta_j}{\delta_{j+1}} - (\varphi^j - \mathcal{P}_{j+1}^{N-1}\varphi^j)(x_{i+1/2})\Big). \end{aligned} \tag{5.28}$$

5.3. **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 5.1 we have a simple expression for the reconstructed jump to prove its sign-stability in the limit $\Delta x \to 0$. We show that the error $\varepsilon(\Delta x)$ goes to zero, if 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 5.7. *The notation $\Delta x \to 0$ should be interpreted in the way that*

$$\max\{x_{i+1} - x_i\} \to 0 \quad \textit{for a grid } x_0 < x_1 < \cdots < x_m. \tag{5.29}$$

Remark 5.8. *When calculating the errors ε_j we must be aware that*

$$\mathrm{d}\varphi^j(x) = \varphi^j(x) - \mathcal{P}_j^{N-1}\varphi^j(x). \tag{5.30}$$

Theorem 5.9. *Let ϕ be an infinitely smooth RBF of first or second order. Then, we have that $\varepsilon_j(\Delta x) = \mathcal{O}(\Delta x^2)$ for $\Delta x \to 0$ for $N = 2, 3$ and*

$$\mathrm{jR}_{i+1/2} \approx \sum_{j=0}^{d_{N-1}-1} C_j(\gamma_{i+r_{N-1}+N+j} - \gamma_{i+r_{N-1}+N-1+j}). \tag{5.31}$$

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

$$\phi(x) = h(x^2). \tag{5.32}$$

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

$$\begin{aligned} \mathrm{d}\varphi^k(y) &= \phi(y - x_k) - \phi(y - x_{k-1}) - \phi(x_{k-1} - x_k) + \phi(0), \\ &= \frac{\phi''(0)}{2}\Big((y - x_k)^2 - (y - x_{k-1})^2 - (x_{k-1} - x_k)^2\Big) + \mathcal{O}(\Delta x^4), \\ &= -\phi''(0)(x_{k-1} - x_k)(x_{k-1} - y) + \mathcal{O}(\Delta x^4). \end{aligned} \tag{5.33}$$

And we further have

$$\begin{aligned} (\varphi^k - \mathcal{P}^1_{k+1}\varphi^k)(y) &= \phi(y - x_k) - \phi(y - x_{k-1}) - \phi(0) + \phi(x_k - x_{k-1}), \\ &= \frac{\phi''(0)}{2}\Big((y - x_k)^2 - (y - x_{k-1})^2 + (x_{k-1} - x_k)^2\Big) + \mathcal{O}(\Delta x^4), \\ &= -\phi''(0)(x_{k-1} - x_k)(x_k - y) + \mathcal{O}(\Delta x^4). \end{aligned} \tag{5.34}$$

From eq. (5.33) we have

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

and we conclude

$$\varepsilon_k(\Delta x) = \mathcal{O}(\Delta x^2). \tag{5.35}$$

Next, we consider the more complicated case with $N = 3$ and a second order RBF ϕ. Therefore, we need to analyse the following two terms:

$$\begin{aligned} \mathrm{d}\varphi^{k+1}(y) &= \varphi^{k+1}(y) - \varphi^{k+1}(x_k)L^{k+1}_k(y) - \varphi^{k+1}(x_{k-1})L^{k+1}_{k-1}(y), \\ &= \phi(y - x_{k+1}) - \phi(y - x_k)L^{k+1}_k(x_{k+1}) - \phi(y - x_{k-1})L^{k+1}_{k-1}(x_{k+1}) \\ &- \Big(\phi(x_k - x_{k+1}) - \phi(0)L^{k+1}_k(x_{k+1}) - \phi(x_k - x_{k-1})L^{k+1}_{k-1}(x_{k+1})\Big)L^{k+1}_k(y) \\ &- \Big(\phi(x_{k-1} - x_{k+1}) - \phi(x_{k-1} - x_k)L^{k+1}_k(x_{k+1}) - \phi(0)L^{k+1}_{k-1}(x_{k+1})\Big)L^{k+1}_{k-1}(y), \end{aligned} \tag{5.36}$$

$$\begin{aligned} (\varphi^k - \mathcal{P}^2_{k+1}\varphi^k)(y) &= \varphi^k(y) - \varphi^k(x_k)L^{k+1}_k(y) - \varphi^k(x_{k-1})L^{k+1}_{k-1}(y), \\ &= \phi(y - x_k) - \phi(y - x_{k-1})L^k_{k-1}(x_k) - \phi(y - x_{k-2})L^k_{k-2}(x_k) \\ &- \Big(\phi(0) - \phi(x_k - x_{k-1})L^k_{k-1}(x_k) - \phi(x_k - x_{k-2})L^k_{k-2}(x_k)\Big)L^{k+1}_k(y) \\ &- \Big(\phi(x_{k-1} - x_k) - \phi(0)L^k_{k-1}(x_k) - \phi(x_{k-1} - x_{k-2})L^k_{k-2}(x_k)\Big)L^{k+1}_{k-1}(y). \end{aligned} \tag{5.37}$$

As before we apply the Taylor expansion

$$\begin{aligned}\varphi^{k+1}(y) &= \phi(y-x_{k+1}) - \phi(y-x_k)L_k^{k+1}(x_{k+1}) - \phi(y-x_{k-1})L_{k-1}^{k+1}(x_{k+1}),\\ &= \frac{\phi''(0)}{2}\Big((y-x_{k+1})^2 - (y-x_k)^2L_k^{k+1}(x_{k+1}) - (y-x_{k-1})^2L_{k-1}^{k+1}(x_{k+1})\Big)\\ &\quad + \frac{\phi^{(4)}(0)}{2}\Big((y-x_k)^4 - (y-x_k)^4L_k^{k+1}(x_{k+1}) - (y-x_{k-1})^4L_{k-1}^{k+1}(x_{k+1})\Big)\\ &\quad + \mathcal{O}(\Delta x^6).\end{aligned} \tag{5.38}$$

Thus, 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

$$a_1 = a_1^{k-1}L_{k-1}^{k+1}(x_{k+1}) + a_1^kL_k^{k+1}(x_{k+1}). \tag{5.39}$$

From standard algebra we get that $a_1^{k-1} = a_1^k = 0$. The fourth order term is

$$\begin{aligned}a_2 &= (y-x_{k+1})^4 - (y-x_k)^4L_k^{k+1}(x_{k+1}) - (y-x_{k-1})^4L_{k-1}^{k+1}(x_{k+1})\\ &\quad - \Big((x_k-x_{k+1})^4 - (x_k-x_{k-1})^4L_{k-1}^{k+1}(x_{k+1})\Big)L_k^{k+1}(y)\\ &\quad - \Big((x_{k-1}-x_{k+1})^4 - (x_{k-1}-x_k)^4L_k^{k+1}(x_{k+1})\Big)L_{k-1}^{k+1}(y),\\ &= 6(x_{k-1}-x_{k+1})(x_k-x_{k+1})(x_{k-1}-y)(x_k-y).\end{aligned} \tag{5.40}$$

We repeat this for the coefficients b_1 and b_2

$$b_1 = b_1^{k-1}L_{k-1}^k(x_k) + b_1^{k-2}L_{k-2}^k(x_k), \tag{5.41}$$

with $b_1^{k-1} = b_1^{k-2} = 0$. The fourth order term is

$$\begin{aligned}b_2 &= (y-x_k)^4 - (y-x_{k-1})^4L_{k-1}^k(x_k) - (y-x_{k-2})^4L_{k-2}^k(x_k)\\ &\quad - \Big(-(x_k-x_{k-1})^4L_{k-1}^k(x_k) - (x_k-x_{k-2})^4L_{k-2}^k(x_k)\Big)L_k^{k+1}(y)\\ &\quad - \Big((x_{k-1}-x_k)^4 - (x_{k-1}-x_{k-2})^4L_{k-2}^k(x_k)\Big)L_{k-1}^{k+1}(y),\\ &= 6(x_{k-2}-x_k)(x_{k-1}-x_k)(x_{k-1}-y)(x_k-y).\end{aligned} \tag{5.42}$$

We summarize the results

$$\mathrm{d}\varphi^{k+1}(y) \approx \frac{\phi^{(4)}(0)}{4}(x_{k-1}-x_{k+1})(x_k-x_{k+1})(x_{k-1}-y)(x_k-y), \tag{5.43}$$

$$(\varphi^k - \mathcal{P}_{k+1}^2\varphi^k)(y) \approx \frac{\phi^{(4)}(0)}{4}(x_{k-2}-x_k)(x_{k-1}-x_k)(x_{k-1}-y)(x_k-y), \tag{5.44}$$

plus higher order terms $\mathcal{O}(\Delta x^6)$ for $\Delta x \to 0$. From this we recover

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

Thus, we have

$$\mathrm{d}\varphi^{k+1}(y)\frac{\delta_k}{\delta_{k+1}} - (\varphi^k - \mathcal{P}^2_{k+1}\varphi^k)(y) = \mathcal{O}(\Delta x^6), \tag{5.45}$$

which yields

$$\varepsilon_k(\Delta x) = \mathcal{O}(\Delta x^2), \tag{5.46}$$

for $\Delta x \to 0$. □

Since the error term $\varepsilon(\Delta x)$ vanishes, the remaining step is to prove that each term of (5.31) has the same sign as the jump.

Theorem 5.10 (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 2. Then, for infinitely smooth RBFs of first or second order it holds that*

$$\operatorname{sgn}(C_j(\gamma_{i+r_{N-1}+N+j} - \gamma_{i+r_{N-1}+N-1+j})) = \operatorname{sgn}(y_{i+1} - y_i), \tag{5.47}$$

for all $j = 0, \dots, d_{N-1} - 1$.

Proof. The proof is based on a study of all possible choices of stencils, that may result from Algorithm 2:

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

For each case we look at any inequality due to Algorithm 2 to recover the particular stencil configuration, and show for each case that (5.47) is fulfilled.
Note that $jR_{i+1/2} = 0$, if $S_i = S_{i+1}$. So, we do not include these cases in the analysis.
Let us first consider $N = 2$ and assume ϕ is of first order.

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

$$|\gamma_i| < |\gamma_{i+1}|, \qquad |\gamma_{i+1}| < |\gamma_{i+2}|. \tag{5.48}$$

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

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

and with (5.33) 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

$$\operatorname{sgn}\big(C_0(\gamma_{i+1} - \gamma_i)\big) = \operatorname{sgn}(\gamma_{i+1} - \gamma_i) = \operatorname{sgn}(\gamma_{i+1}) = \operatorname{sgn}(y_{i+1} - y_i), \tag{5.49}$$

since

$$|a| > |b| \Rightarrow \operatorname{sgn}(a - b) = \operatorname{sgn}(a). \tag{5.50}$$

Case 2. *Let us 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*

$$|\gamma_i| < |\gamma_{i+1}|, \qquad |\gamma_{i+1}| > |\gamma_{i+2}|. \tag{5.51}$$

The jump can be represented by

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

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

$$C_1 = C_0 - \delta_{i+1} \approx \delta_i \frac{x_{i+1/2} - x_{i+1}}{x_i - x_{i-1}} < 0. \tag{5.52}$$

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 applied (5.50) *with* (5.51).

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*

$$|\gamma_i| > |\gamma_{i+1}|, \qquad |\gamma_{i+1}| > |\gamma_{i+2}|. \tag{5.53}$$

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 using (5.33), *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

$$\operatorname{sgn}\big(C_0(\gamma_{i+2} - \gamma_{i+1})\big) = \operatorname{sgn}(\gamma_{i+1} - \gamma_{i+2}) = \operatorname{sgn}(\gamma_{i+1}). \tag{5.54}$$

This finishes the proof of the sign-property for the second degree reconstruction with infinitely smooth RBFs of first order for small enough grids.
The proof for $N = 3$ can be found in Appendix A. □

6. Entropy Stable RBF-Based Methods

In one space dimension there is no need to deviate from the polynomial reconstruction. For unstructured grids in multiple dimensions the problem is the construction of an interpolation function. There exist a lot 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 can circumvent this problem since we do not need a unisolvent set of cells or points, but just a unisolvent subset of lower order. Thus, by adding some extra cells we drastically reduce the possibility that unsolvable configurations occur.

6.1. **RBF-TeCNOp Method**

Based on the theory of entropy stable schemes and the work of Fjordholm et al. [10] we introduce the RBF-TeCNOp scheme.
By using Algorithm 2 with (4.19) for calculating the least oscillatory stencil we showed in Theorem 5.10 that the sign-property holds for 2nd and 3rd degree reconstruction in the limit of $\Delta x \to 0$. We conjecture that this result holds for higher order reconstructions. Thus, by combining the framework proposed by Fjordholm et. al in [10] 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 Algorithm 1 to circumvent ill-conditioning in the reconstruction step.
In more detail, for constructing a p-th order RBF-TeCNOp method of the form (2.15) we use an entropy conservative flux of order $2k$ with $k = \lceil p/2 \rceil$ (see Theorem 2.2) and an ENO based RBF reconstruction (Algorithm 2) on the scaled entropy variables of order p with multiquadratics of order $p-1$.
Based on the Roe diffusion operator

$$R|\Lambda|R^{-1}[\![u]\!], \tag{6.1}$$

with the eigenvector matrix R and the diagonal matrix of the eigenvalues, evaluated at the Roe average, Λ we are choosing R and Λ in the same way. By Merriam [24] there is a scaling of the eigenvectors such that $RR^T = v_u = \partial_u v(u_{i+1/2})$. Thus, we get the relation

$$R|\Lambda|R^{-1}[\![u]\!] \approx R|\Lambda|R^{-1}v_u[\![v]\!] = R|\Lambda|R^T[\![v]\!], \tag{6.2}$$

that has a similar structure to that of a diffusion operator (2.16). The numerical diffusion term can be written as

$$D_{i+1/2}\langle\!\langle v \rangle\!\rangle_{i+1/2} = R_{i+1/2}\Lambda_{i+1/2}\langle\!\langle w \rangle\!\rangle_{i+1/2}, \tag{6.3}$$

with the scaled entropy variables (2.18).
Furthermore, we choose $\Lambda_{i+1/2} = \mathrm{diag}(\lambda_1(u_{i+1/2}), \dots, \lambda_N(u_{i+1/2}))$ and $u_{i+1/2} = \frac{u_i + u_{i+1}}{2}$ 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 just 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 (3.19).

6.2. **RBF-Finite Volume Method**

The combination of the RBF interpolation with finite volume methods works analogeous to the RBF-TeCNOp Method. Aboiyar et al. combine in their work [1] a high-order WENO approach with a polyharmonic spline reconstruction. Bigoni et al. apply a high-order WENO approach to multiquadratics [3].
We construct an entropy stable finite volume method of second order that is essentially nonoscillatory by combining (2.24) with a second order accurate RBF interpolation that acts on the scaled entropy variables. Therefore, we are using multiquadratics with the smoothness indicator (4.19) combined with Algorithm 2 and the vector valued rational approximation from Algorithm 1 for a stable evaluation of the interpolation function. We conjecture the sign-property for the RBF reconstruction on mean values that is based on Algorithm 2 which is fulfilled in the pointwise case for second and third degree reconstruction in the limit $\Delta x \to 0$ (Theorem 5.10). Under this assumption we recover a second order entropy stable finite volume (RBF-EFV2) method.

7. Numerical results

In this chapter, we are evaluating the second order entropy stable finite volume (EFV2) and the TeCNOp methods with RBF reconstruction 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 at least as well.
For the polynomial reconstruction we use the original algorithm from [18] to select the stencil and in the RBF case we use Algorithm 2. The EFV2 and TeCNOp methods are based on the ERoe diffusion term (6.3).
The parameters for the vector-valued rational approximation are chosen as described in Section 3.3. Further, we choose the shape parameter $\varepsilon = 0.1$ for all examples.

7.1. **Linear Advection Equation**

We consider the linear advection equation

$$u_t + a u_x = 0, \tag{7.1}$$

with wave speed $a = 1$ and periodic boundary conditions [23]. With the entropy function $\eta(u) = \frac{u^2}{2}$ we have

$$q(u) = a\frac{u^2}{2}, \qquad v(u) = u, \qquad \psi(u) = a\frac{u^2}{2}, \tag{7.2}$$

and obtain the second order entropy conservative flux

$$\tilde{F}_{i+1/2} = \bar{u}_{i+1/2}, \tag{7.3}$$

to construct a high-order accurate scheme. We use a 5th order SSPRK method for time discretization [17]. 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 2. 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 and similar to the ones found in literature.

7.2. **Burger's Equation**

For the Burger's equation

$$u_t + \frac{1}{2}(u^2)_x = 0, \tag{7.4}$$

we study the convergence and check if the methods handle discontinuities without introducing major oscillations. The EFV2 and TeCNOp method are based on the entropy $\eta(u) = u^2/2$ and

$$q(u) = \frac{u^3}{3}, \qquad v(u) = u, \qquad \psi(u) = \frac{u^3}{6}, \tag{7.5}$$

leading to an entropy conservative flux

$$\tilde{F}_{i+1/2} = \frac{u_i^2 + u_i + u_{i+1} + u_{i+1}^2}{6}, \tag{7.6}$$

which is used to construct an high-order scheme. For the time discretization we use a 5th order SSPRK method [17]. Furthermore, we choose the domain $[0, 1]$ and the initial conditions $u_0(x) = \sin(2\pi x)$.
A detailed analysis of the convergence is shown in Table 2. 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. 1). Furthermore, we observe that the difference between the reconstruction methods goes to machine precision in the smooth part and at the shock it stays small.

N		Linear Advection eq.				Burger's 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 2. Convergence rates of TeCNOp and EFV2 methods using multiquadratics and polynomials for the linear advection and Burger's equation on $[0,1]$ at time $t = 0.1$. We use periodic boundary conditions and $u_0(x) = \sin(2\pi x)$, shape parameter $\varepsilon = 0.1$, CFL = 0.5 .

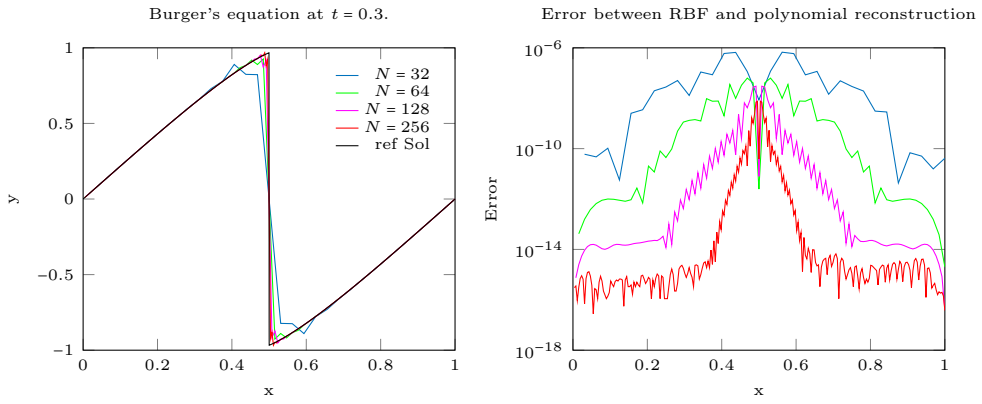

FIGURE 1. Burger's equation on $[0,1]$ at time $t = 0.3$ with continuous initial condition $u_0 = \sin(2\pi x)$, shape parameter $\varepsilon = 0.1$, CFL = 0.5, solved by TeCNO5.

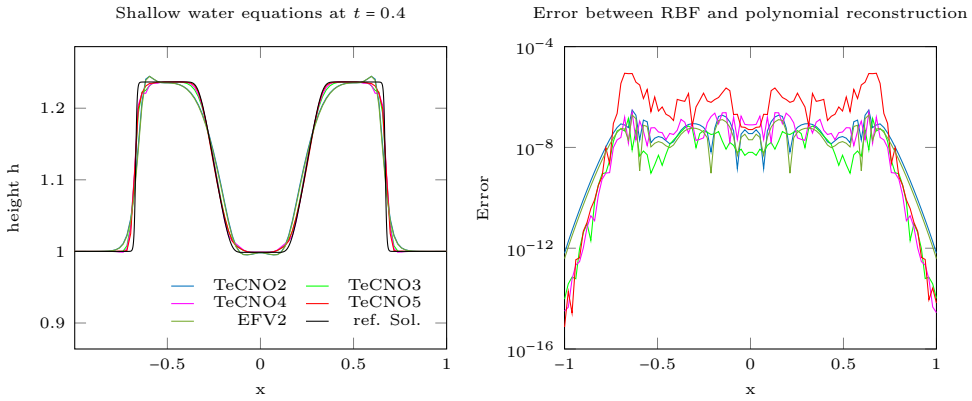

FIGURE 2. Shallow water equations on $[-1, 1]$ at time $t = 0.4$ with $N = 100$, shape parameter $\varepsilon = 0.1$, CFL = 0.5, solved by TeCNO and EFV2.

7.3. **Shallow Water Equations**

For the shallow water equations (2.6) we consider the dambreak problem with initial conditions

$$(h_0, m_0) = \begin{cases} (1.5, 0) & \text{if } |x| \le 0.2 \\ (1, 0) & \text{if } |x| > 0.2 \end{cases}, \tag{7.7}$$

on the domain $[-1, 1]$ and periodic boundary conditions. We use a second order entropy stable flux (2.9) to construct a high-order flux and a third order SSPRK method for the time integration.
Fjordholm et al. [10] showed that the standard TeCNO scheme behaves similar to the ENO-MUSCL scheme. The same holds for the RBF-TeCNOp scheme and the RBF-EFV2 scheme as seen in Fig. 2. The difference between the RBF methods and the polynomial scheme is around $1e-6$ in the region where the discontinuity passed and much smaller in smooth regions.

7.4. **Euler Equations**

The one-dimensional Euler equations (2.10) are a system of size three. 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 (2.13). Further, we choose $\gamma = 1.4$ which simulates a diatomic gas such as air.

7.4.1. *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 discontinuity.
The initial conditions are

$$(\rho_0, m_0, p_0) = \begin{cases} (1, 0, 1) & \text{if } x < 0 \\ (0.125, 0, 0.1) & \text{if } x \ge 0 \end{cases}, \tag{7.8}$$

where $m = u\rho$. The results at time $t = 2$ of the RBF-TeCNOp and RBF-EFV2 methods are shown in Fig. 3, 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. 4 that their difference is decreasing with the refinement of the grid.

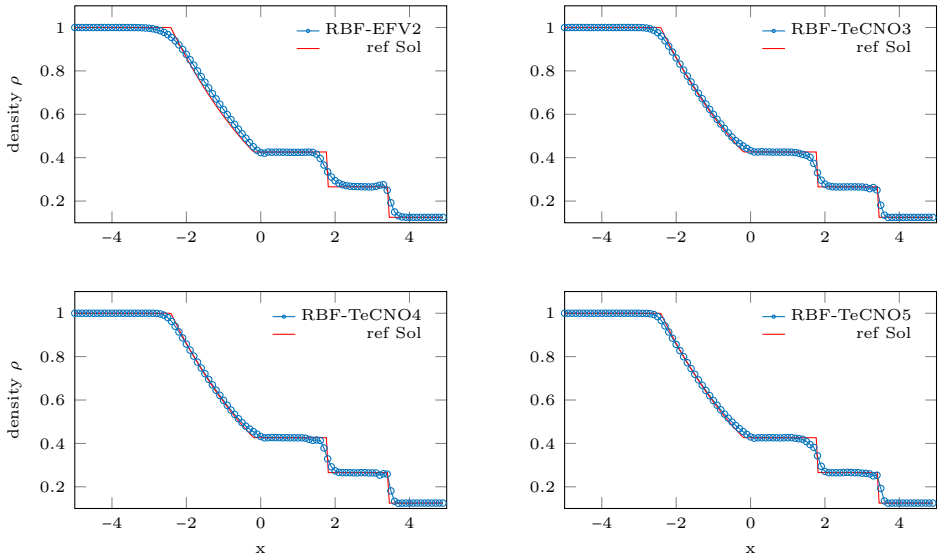

FIGURE 3. Sod's shock tube problem on $[-5,5]$ at time $t = 2$ with $N = 100$, shape parameter $\varepsilon = 0.1$, CFL = 0.3, solved by RBF-TeCNOp and RBF-EFV2.

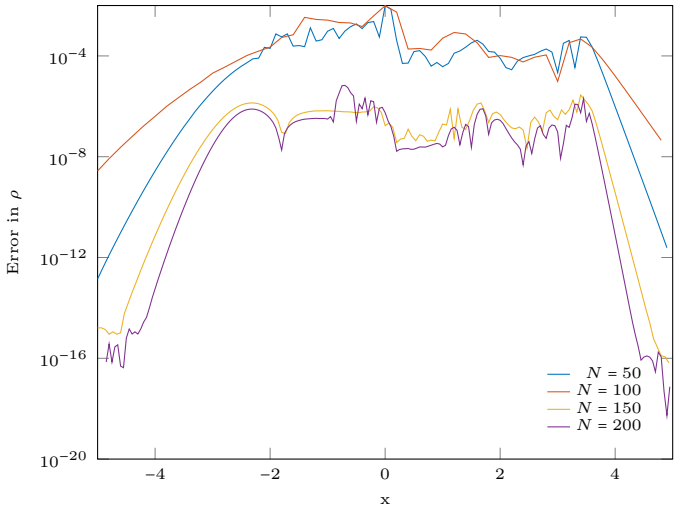

FIGURE 4. Pointwise error between RBF and polynomial based reconstruction EFV2 method for the Sod's shock tube problem on $[-5,5]$ at time $t = 2$ for different number of grid points N, shape parameter $\varepsilon = 0.1$, CFL = 0.3.

7.4.2. *Lax Shock Tube Problem*

The Lax shock tube problem is another Riemann problem defined by the initial conditions

$$(\rho_0, m_0, p_0) = \begin{cases} (0.445, 0.698, 3.528) & \text{if } x < 0 \\ (0.5, 0, 0.571) & \text{if } x \geq 0 \end{cases}, \tag{7.9}$$

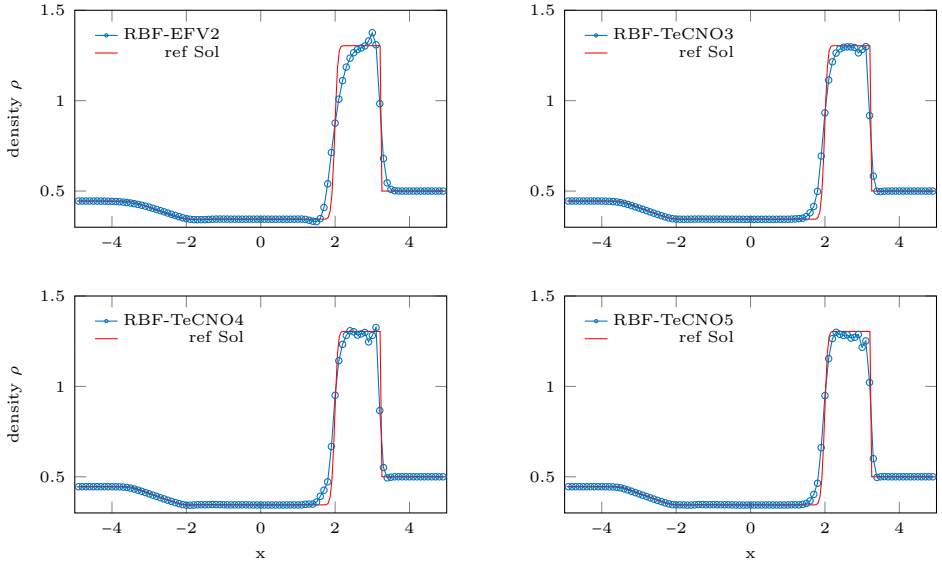

FIGURE 5. Lax shock tube problem on $[-5,5]$ at time $t = 1.3$ with $N = 100$, shape parameter $\varepsilon = 0.1$, CFL = 0.3, solved by RBF-TeCNOp and RBF-EFV2.

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. 5. The second order RBF-EFV2 method does not perform well for this case.

7.4.3. *Shu-Osher Shock-Entropy Wave Interaction Problem*

The Shu-Osher problem is a shock-turbulence interaction in which a shock propagates into a low frequency wave. Due to this interaction high-frequency oscillations develop over time. The initial conditions are

$$(\rho_0, m_0, p_0) = \begin{cases} (3.857143, 2.629369, 10.33333) & \text{if } x < -4 \\ (1 + 0.2\sin(5x), 0, 1) & \text{if } x \geq -4 \end{cases}, \tag{7.10}$$

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 one due to excessive dissipation.

8. CONCLUSIONS

We introduce a new smoothness indicator and an algorithm to choose the least oscillatory stencil based on RBF interpolation. This smoothness indicator is directly related to the RBF interpolation and it is based on the generalized divided difference method. For this ENO reconstruction we prove the sign-property in the pointwise case for the second and third order reconstruction in the limit $\Delta x \to 0$ for infinitely smooth RBFs. Further, we conjecture this property for higher order schemes and for the case of the average-based interpolation. Note that the condition $\Delta x \to 0$ can be replaced by the condition for the shape parameter $\varepsilon \to 0$.
Based on this procedure we construct a 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 minus 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 apply the vector-valued rational approximation method [38].
Thus, we introduce a method that has all the properties from the original TeCNO scheme [10]. It is entropy

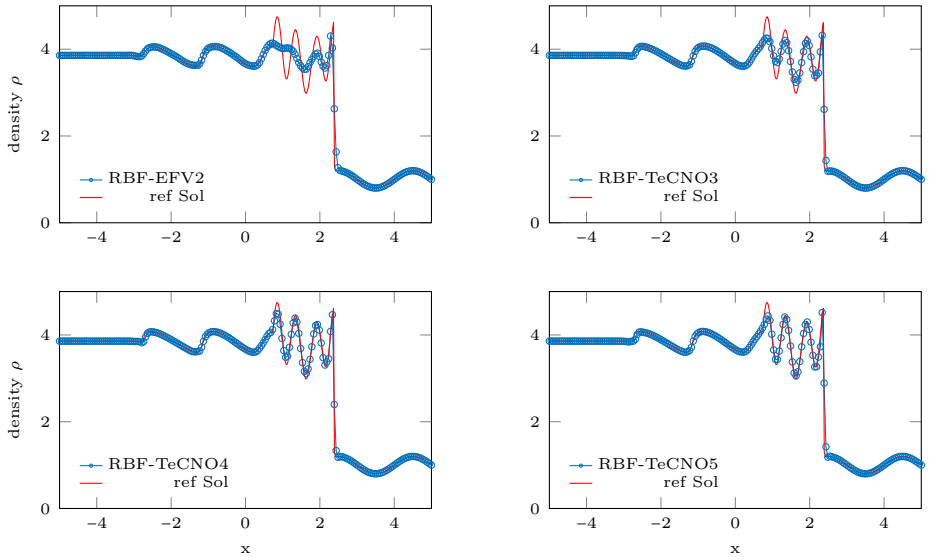

FIGURE 6. Shu-Osher problem on $[-5,5]$ at time $t = 1.8$ with $N = 200$, shape parameter $\varepsilon = 0.1$, CFL = 0.3 solved by RBF-TeCNOp and RBF-EFV2.

stable, high-order accurate for smooth solutions, essentially nonoscillatory near discontinuities, yet flexible on unstructured grids.
To show their robustness we present a range of numerical simulations in one dimension. The solutions coincide up to a small error with those obtained from the original TeCNO method.
The main drawback of the method is the expansive evaluation of the vector-valued approximation to circumvent ill-conditioning, but this problem is being considered. The advantages of RBF-based reconstructions will become much clearer for high-dimensional problems and we hope to report on this in the near future.

APPENDIX A. PROOF OF THEOREM 5.10 FOR THE 3RD DEGREE RECONSTRUCTION

Proof. The proof is based on a study of all possible choices of stencils, that may result from Algorithm 2:

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

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 2 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

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

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

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

With (5.33) and (5.43) 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

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

Note that the term $\delta_{k+1}^2/\delta_{k+1}^3$ is always positive (Corollary 4.4). Next, we can show the sign of the constant C_l using Theorem 5.1

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

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

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

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

$$\operatorname{sgn}(C_l) = (-1)^{r_{N-1}+N-1+l}. \tag{A.3}$$

Case 1. *We 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*

$$|\gamma_i^2| < |\gamma_{i+1}^2|, \quad |\gamma_i^3| < |\gamma_{i+1}^3|, \quad |\gamma_{i+1}^2| < |\gamma_{i+2}^2|, \quad \gamma_{i+1}^3| < |\gamma_{i+2}^3|. \tag{A.4}$$

Note that this case can be characterized by $d_2 = 1$ and $s_2 = r_2 = -2$. From Theorem 5.9 we know

$$jR_{i+1/2} \approx C_0(\gamma_{i+1}^3 - \gamma_i^3), \tag{A.5}$$

and we have

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

where we used (A.1), (A.2) *and* (A.4).

Case 2. *Next, we 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*

$$\begin{aligned} &|\gamma_i^2| < |\gamma_{i+1}^2|, \quad |\gamma_i^3| < |\gamma_{i+1}^3|, \\ &\begin{cases} |\gamma_{i+1}^2| < |\gamma_{i+2}^2|, \quad |\gamma_{i+1}^3| > |\gamma_{i+2}^3|, & (a) \\ |\gamma_{i+1}^2| > |\gamma_{i+2}^2|, \quad |\gamma_{i+2}^3| < |\gamma_{i+3}^3|. & (b) \end{cases} \end{aligned} \tag{A.6}$$

The jump can be written by

$$jR_{i+1/2} \approx C_0(\gamma_{i+1}^3 - \gamma_i^3) + C_1(\gamma_{i+2}^3 - \gamma_{i+1}^3). \tag{A.7}$$

For each term we calculate its sign. The first term can be done in the same way as above and it holds for both (a) *and* (b) *in* (A.6)

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

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

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

For (b) *we split it in two terms using* (A.1)

$$C_1(\gamma_{i+2}^3-\gamma_{i+1}^3)=C_1\frac{\delta_{i+2}^2}{\delta_{i+2}^3}(\gamma_{i+2}^2-\gamma_{i+1}^2)-C_1\frac{\delta_{i+1}^2}{\delta_{i+1}^3}(\gamma_{i+1}^2-\gamma_i^2),$$

and calculate the sign of each one

$$\begin{aligned}\operatorname{sgn}\Big(C_1\frac{\delta_{i+2}^2}{\delta_{i+2}^3}(\gamma_{i+2}^2-\gamma_{i+1}^2)\Big)&=\operatorname{sgn}(\gamma_{i+1}^2-\gamma_{i+2}^2)=\operatorname{sgn}(\gamma_{i+1}^2)=\operatorname{sgn}(y_{i+1}-y_i),\\ \operatorname{sgn}\Big(-C_1\frac{\delta_{i+1}^2}{\delta_{i+1}^3}(\gamma_{i+1}^2-\gamma_i^2)\Big)&=\operatorname{sgn}(\gamma_{i+1}^2-\gamma_i^2)\Big)=\operatorname{sgn}(\gamma_{i+1}^2)=\operatorname{sgn}(y_{i+1}-y_i).\end{aligned}$$

Case 3. *Next, we 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*

$$|\gamma_i^2|<|\gamma_{i+1}^2|,\quad |\gamma_i^3|<|\gamma_{i+1}^3|,\quad |\gamma_{i+1}^2|>|\gamma_{i+2}^2|,\quad |\gamma_{i+2}^3|>|\gamma_{i+3}^3|. \tag{A.8}$$

For the reconstructed jump we have

$$jR_{i+1/2}\approx C_0(\gamma_{i+1}^3-\gamma_i^3)+C_1(\gamma_{i+2}^3-\gamma_{i+1}^3)+C_2(\gamma_{i+3}^3-\gamma_{i+2}^3), \tag{A.9}$$

and calculate the sign of each term

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

For the second term we need

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

such that we can show

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

The last term yields

$$\operatorname{sgn}\left(C_2(\gamma_{i+3}^3-\gamma_{i+2}^3)\right)=\operatorname{sgn}(\gamma_{i+3}^3-\gamma_{i+2}^3)=-\operatorname{sgn}(\gamma_{i+2}^3)=\operatorname{sgn}(\gamma_{i+1}^2-\gamma_{i+2}^2)=\operatorname{sgn}(\gamma_{i+1}^2)=\operatorname{sgn}(y_{i+1}-y_i).$$

Case 4. *Next, we 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*

$$\begin{cases}|\gamma_i^2|<|\gamma_{i+1}^2|, \quad |\gamma_i^3|\geq|\gamma_{i+1}^3|, & (a1)\\ |\gamma_i^2|\geq|\gamma_{i+1}^2|, \quad |\gamma_{i+1}^3|<|\gamma_{i+2}^3|, & (a2)\end{cases}$$
$$\begin{cases}|\gamma_{i+1}^2|<|\gamma_{i+2}^2|, \quad |\gamma_{i+1}^3|\geq|\gamma_{i+2}^3|, & (b1)\\ |\gamma_{i+1}^2|\geq|\gamma_{i+2}^2|, \quad |\gamma_{i+2}^3|<|\gamma_{i+3}^3|. & (b2)\end{cases} \tag{A.10}$$

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*

$$jR_{i+1/2} \approx C_0(\gamma_{i+2}^3 - \gamma_{i+1}^3). \tag{A.11}$$

Let us start with the combination (a_1, b_1). *We have*

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

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*

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

Case 5. *Next, we 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*

$$\begin{aligned}&\begin{cases}|\gamma_i^2| < |\gamma_{i+1}^2|, \quad |\gamma_i^3| \geq |\gamma_{i+1}^3|, & (a)\\ |\gamma_i^2| \geq |\gamma_{i+1}^2|, \quad |\gamma_{i+1}^3| < |\gamma_{i+2}^3|, & (b)\end{cases}\\ &|\gamma_{i+1}^2| > |\gamma_{i+2}^2|, \quad |\gamma_{i+2}^3| > |\gamma_{i+3}^3|.\end{aligned} \tag{A.12}$$

The jump is represented as

$$jR_{i+1/2} \approx C_0(\gamma_{i+2}^3 - \gamma_{i+1}^3) + C_1(\gamma_{i+3}^3 - \gamma_{i+2}^3). \tag{A.13}$$

In the case of (a) *we precalculate*

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

With these we have

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

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*

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

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*

$$|\gamma_i^2| > |\gamma_{i+1}^2|, \quad |\gamma_{i+1}^3| > |\gamma_{i+2}^3|, \quad |\gamma_{i+1}^2| > |\gamma_{i+2}^2|, \quad |\gamma_{i+2}^3| > |\gamma_{i+3}^3|, \tag{A.14}$$

with a reconstructed jump of the form

$$jR_{i+1/2} \approx C_0(\gamma_{i+3}^3 - \gamma_{i+2}^3). \tag{A.15}$$

We recover

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

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

Acknowledgments

The authors are grateful to Dr. Deep Ray for helpful discussions and insights. This work was partially supported by SNSF.

References

[1] 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.

[2] 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.

[3] C. Bigoni and J. S. Hesthaven. Adaptive weno methods based on radial basis function reconstruction. *Journal of Scientific Computing*, 72(3):986–1020, 2017.

[4] 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.

[5] 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.

[6] 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.

[7] J. Duchon. *Splines minimizing rotation-invariant semi-norms in Sobolev spaces*, pages 85–100. Springer, 1977.

[8] G. E. Fasshauer and J. G. Zhang. On choosing "optimal"shape parameters for rbf approximation. *Numerical Algorithms*, 45(1-4):345–368, 2007.

[9] 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.

[10] 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.

[11] U. S. Fjordholm, S. Mishra, and E. Tadmor. Eno reconstruction and eno interpolation are stable. *Foundations of Computational Mathematics*, 13(2):139–159, 2013.

[12] U. S. Fjordholm and D. Ray. A sign preserving weno reconstruction method. *Journal of Scientific Computing*, pages 1–22, 2016.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] J. Hesthaven. *Numerical Methods for Conservation Laws: From Analysis to Algorithms*. Society for Industrial and Applied Mathematics, 2018/04/06 2017.

[20] S. N. Kružkov. First order quasilinear equations in several independent variables. *Mathematics of the USSR-Sbornik*, 10(2):217, 1970.

[21] 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.
[22] 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.
[23] R. J. LeVeque. *Numerical methods for conservation laws*. Springer Science & Business Media, 1992.
[24] M. L. Merriam. *An entropy-based approach to nonlinear stability*. Stanford University, Stanford, CA, USA, 1989.
[25] C. A. Micchelli. Interpolation of scattered data: Distance matrices and conditionally positive definite functions. *Constructive Approximation*, 2(1):11–22, 1986.
[26] M. S. Mock. Systems of conservation laws of mixed type. *Journal of Differential equations*, 37(1):70–88, 1980.
[27] G. Mühlbach. A recurrence formula for generalized divided differences and some applications. *Journal of Approximation Theory*, 9(2):165–172, 1973.
[28] G. Mühlbach. The general neville-aitken-algorithm and some applications. *Numerische Mathematik*, 31(1):97–110, 1978.
[29] 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.
[30] 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.
[31] R. Schaback. Error estimates and condition numbers for radial basis function interpolation. *Advances in Computational Mathematics*, 3(3):251–264, 1995.
[32] R. Schaback. Native hilbert spaces for radial basis functions i. *New Developments in Approximation Theory*, 132:255–282, 1998.
[33] R. Schaback. Multivariate interpolation by polynomials and radial basis functions. *Constructive Approximation*, 21(3):293–317, 2005.
[34] C.-W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *Journal of computational physics*, 77(2):439–471, 1988.
[35] E. Tadmor. The numerical viscosity of entropy stable schemes for systems of conservation laws. i. *Mathematics of Computation*, 49(179):91–103, 1987.
[36] 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.
[37] H. Wendland. *Scattered Data Approximation:*. Cambridge University Press, Cambridge, 2004.
[38] G. Wright and B. Fornberg. Stable computations with flat radial basis functions using vectorvalued rational approximations. Elsevier Editorial System, 10 2016.