Journal of Scientific Computing manuscript No.
(will be inserted by the editor)

Adaptive WENO methods based on radial basis function reconstruction

Caterina Bigoni · Jan S. Hesthaven

February 3, 2017

Abstract We explore the use of radial basis functions (RBF) in the weighted essentially non-oscillatory (WENO) reconstruction process used to solve hyperbolic conservation laws, resulting in a numerical method of arbitrarily high order to solve problems with discontinuous solutions. Thanks to the mesh-less property of the RBFs, the method is suitable for non-uniform grids and mesh adaptation. We focus on multiquadric radial basis functions and propose a simple strategy to choose the shape parameter to control the balance between achievable accuracy and the numerical stability. We also develop an original smoothness indicator which is independent of the RBF for the WENO reconstruction step. Moreover, we introduce type I and type II RBF-WENO methods by computing specific linear weights. The RBF-WENO method is used to solve linear and nonlinear problems for both scalar and systems of conservation laws, including Burgers equation, the Buckley-Leverett equation, and the Euler equations. Numerical results confirm the performance of the proposed method. We finally consider an effective conservative adaptive algorithm that captures moving shocks and rapidly varying solutions well. Numerical results on moving grids are presented for both Burgers equation and the more complex Euler equations.

Keywords High-order WENO method · Radial Basis Functions · Hyperbolic Conservation Laws · Adaptivity · Multiquadrics

Mathematics Subject Classification (2000) 35L65 · 65M08 · 65D05

C. Bigoni
E-mail: caterina.bigoni@epfl.ch

J. S. Hesthaven
E-mail: jan.hesthaven@epfl.ch

SB-MATHICSE-MCSS, École Polytechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland

1 Introduction

Finite volume methods (FVMs) are used for solving hyperbolic conservation laws across many scientific and engineering fields. Indeed, the fundamental principle of conservation applies to the conservation of mass, momentum, energy, etc. Monotone FVMs satisfy all entropy conditions and therefore converge to the unique entropy solution in a non-oscillatory manner. However, monotone methods are at most first order accurate (see [25] for a detailed description). In general, high-order linear FVMs ensure high-order accuracy in smooth regions of the solution, but introduce spurious oscillations in regions of low regularity, e.g., in the neighborhood of shocks and steep gradients. This phenomenon, known as the Gibbs phenomenon, is often unacceptable and must be eliminated, e.g., in the approximation of a density or a pressure, such oscillations may cause these to become negative and thus unphysical. The need to formulate numerical methods for conservation laws which enable arbitrary high-order accuracy without artificial oscillations has led to the development of the Essentially Non-Oscillatory (ENO) methods, introduced in 1987 by Harten and collaborators [21]. These schemes, together with the more recent Weighted ENO (WENO) methods, initiated in Liu *et al.* [26] and further developed by Jiang and Shu [24], are powerful schemes for the discretization of the spacial variable. In past years these methods, coupled with strong stability preserving (SSP) Runge-Kutta methods [17] for the time discretization, have received considerable attention as a powerful tool for solving nonlinear problems comprising both strong discontinuities and complex smooth solution features.

In the FVM the computational domain is partitioned into cells (control volumes) and the solution is expressed in terms of cell averages of the solution on the control volumes themselves. In this framework, the ENO idea is to recursively select a stencil, i.e., a collection of contiguous cells, in the neighborhood of each control volume and reconstruct the unknown solution through polynomial interpolation based on the stencil's cell average values. The stencil selection is performed using Newton divided differences in such a way that possible discontinuities of the solution are avoided and the chosen stencil is "optimal" in the sense of being the least oscillating. In the more sophisticated WENO reconstruction procedure, all candidate stencils are used and the reconstructed solution is recovered as a convex combination of all the reconstructed solutions. Provided the solution is smooth, the WENO method exhibits a superior convergence rate as compared to the ENO method. However, if the solution contains singularities, the WENO method emulates the ENO idea by essentially selecting a single stencil and avoiding stencils that contain shocks. To satisfy these two requirements, the weights associated with each stencil must be carefully defined through a suitable *smoothness indicator*.

Notwithstanding the popularity of these two methods, the use of polynomial interpolation in the reconstruction step requires the stencils to have a fixed size. Indeed, the order of the interpolating polynomials must correspond to the size of the underlying stencils, i.e., the number of cells per stencil. To overcome this restriction, which is particularly problematic for multi-dimensional unstructured meshes, least-squares methods have been applied as an alternative to interpolation. Unfortunately, standard least-squares approximation is often inaccurate as it does not satisfy the interpolation property. The need for more flexible and, at the same time, accurate alternatives becomes evident especially for mesh adaptation. This is of particular importance to guarantee computational efficiency in the solution of conservation laws, where solutions are often composed of regions of fast moving shock fronts combined with regions of high regularity.

The seminal work of [2] proposes a new type of WENO method based on non-polynomial reconstructions. More specifically, in [2] and in related work [1, 23], the mesh-free feature of Radial Basis Functions (RBFs) is successfully combined with ENO/WENO ideas for scalar conservation laws in general geometries using unstructured grids. In the past decades, interpolation techniques with RBFs have been actively studied (see [4]) to approximate functions of more than one variable. RBFs are well-known for their approximations properties [22] and are of special interest in the case of scattered data. The class of polyharmonic splines are used in [2] to define an adaptive ADER-WENO method for scalar equations in two dimensions.

Another approach to non-polynomial ENO and WENO finite volume methods studied in [18] introduces the use of adaptivity of the shape parameter to control the local error. This is done by expressing the shape parameter in terms of the cell averages of the underlying stencil. As a consequence, the local accuracy and convergence are improved with respect to the classic ENO and WENO methods if the underlying solution is smooth. For discontinuous solutions, a switching condition to the classic polynomial reconstruction is proposed to guarantee the essentially non-oscillatory property. The same idea is applied to the finite difference RBF-WENO method in [19].

In this paper we investigate several aspects connected with the RBF-WENO method and extend the method to systems of equations including the benchmark Euler equations. In particular we consider two types of RBFs: the multiquadrics, which belong to the class of infinitely smooth RBFs, and the polyharmonic splines, which are piecewise smooth RBFs. Functions in the former class are characterized by a free parameter, called a *shape parameter*, and we investigate the effect of this on the accuracy of the solution following the works of Fornberg and Wright [14] and Fasshauer [8].

The method proposed in [2] belongs to the so-called type I WENO method because the accuracy, obtained by using a convex combination of stencils, remains equal to what could be obtained using a single stencil. In this case, the WENO method is preferred to the simpler ENO method for stability reasons. By introducing suitable linear weights, the method becomes a type II WENO method and exhibits improved accuracy. Furthermore, in [2] the smoothness indicator is specific to polyharmonic splines, while the weights described in this paper are independent of the type of RBFs.

This paper is organized as follows. Section 2 gives a brief review of high-order finite volume ENO and WENO methods. Section 3 introduces the general framework of the RBF interpolation of scattered data, especially when data are cell averages. We focus our attention on multiquadric interpolation and describe a simple strategy to select a suitable shape parameter for this class of RBFs. The WENO method based on RBF interpolation is presented in Section 4. In Section 5, after briefly describing the chosen numerical flux scheme and time integration strategy, we present several one-dimensional numerical examples for both linear and nonlinear problems and for both scalar equations and systems of equations. Mesh adaptivity is presented in Section 6 together with a selection of related results. Finally, in Section 7, we provide conclusions and future directions of this line of research.

2 Finite volume ENO and WENO methods

2.1 FVM for conservation laws

We consider the hyperbolic conservation laws

$$\frac{\partial \boldsymbol{u}}{\partial t} + \nabla \cdot \boldsymbol{f}(\boldsymbol{u}) = 0, \quad (\boldsymbol{x}, t) \in \Omega_x \times \Omega_t, \tag{2.1}$$

where $\Omega_t = [0, T]$ and $\Omega_x \subset \mathbb{R}^d$. The state vector $\boldsymbol{u} = \boldsymbol{u}(\boldsymbol{x}, t) : \mathbb{R}^d \times \Omega_t \to \mathbb{R}^m$ and the flux function $\boldsymbol{f} = [f_1(\boldsymbol{u}), \ldots, f_m(\boldsymbol{u})]$. This system of equations is equipped with the initial condition $\boldsymbol{u}(\boldsymbol{x}, 0) = \boldsymbol{u}_0(\boldsymbol{x})$ along with suitable boundary conditions.

The finite volume method for the approximation of (2.1) is defined as follows: at each time step, the computational domain is partitioned into control volumes or cells C, which are possibly different in number and position at each time step. Let us for simplicity consider a scalar conservation law. Integrating (2.1) over a space-time control volume $C \times [t^n, t^{n+1}]$, we obtain

$$\bar{\boldsymbol{u}}_C(t^{n+1}) = \bar{\boldsymbol{u}}_C(t^n) - \frac{1}{|C|} \int_{t^n}^{t^{n+1}} \int_{\partial C} \hat{\boldsymbol{n}} \cdot \boldsymbol{f}(\boldsymbol{u}, t) \; d\boldsymbol{x} dt,$$

where $\hat{\boldsymbol{n}}$ represents the outward pointing normal along the boundary of C, indicated as ∂C, $|C|$ is the volume of C and

$$\bar{\boldsymbol{u}}_C(t^n) = \frac{1}{|C|} \int_C \boldsymbol{u}(\boldsymbol{x}, t^n) \, d\boldsymbol{x}$$

is the cell average of the solution $\boldsymbol{u}$ on cell C at time $t = t^n$. We recover the finite volume scheme as

$$\bar{\boldsymbol{u}}_C^{n+1} = \bar{\boldsymbol{u}}_C^n - \frac{\Delta t}{|C|} \tilde{F}_C^n,$$

where $\Delta t = t^{n+1} – t^n$. The integrals that define the numerical flux $\tilde{F}_C^n$ can be approximated using a Gaussian quadrature. The crucial step consists in the computation of the state of the solution $\boldsymbol{u}^*$ at the cell interfaces.

2.2 Reconstruction from cell averages

The goal of the ENO reconstruction is to reconstruct a function $\boldsymbol{u}(\boldsymbol{x})$ at cell interfaces with a certain accuracy. For the sake of simplicity, we consider the ENO/WENO reconstruction in 1D and refer to [33] for a detailed review of these methods.

At time $t^n = n\Delta t$, consider a computational domain $\Omega_x = [a, b]$ partitioned in N subintervals

$$C_i = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \quad \text{for} \quad i = 1, \ldots, N, \tag{2.2}$$

with $x_1 = a$ and $x_N = b$. Moreover, let $\Delta x_i = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}$ be the size of the cell and $x_i = \frac{1}{2}(x_{i-\frac{1}{2}} + x_{i+\frac{1}{2}})$ the cell center. Given the cell averages $\bar{u}_i(x)$ of a function

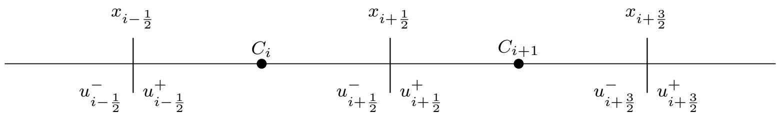

Fig. 1 Illustration of the reconstruction of the interface values

$u(x)$ for each cell C_i in the domain, we seek a polynomial $p_i(x)$ of degree at most $k{-}1$ that is a k-th order accurate approximation to a function $u(x)$ inside the cell:

$$p_i(x) = u(x) + \mathcal{O}(\Delta x^k), \quad x \in C_i, \quad \text{for } i = 1, \dots, N. \tag{2.3}$$

The evaluation of this polynomial at the cell boundaries yields the approximation of $u(x)$

$$u^-_{i+\frac{1}{2}} = p_i(x_{i+\frac{1}{2}}), \quad u^+_{i+\frac{1}{2}} = p_{i+1}(x_{i+\frac{1}{2}}), \quad i = 1, \dots, N,$$

where the superscripts $+$ and $-$ denote the right and left side of the interface, respectively, as shown in Fig. 1.

For each cell C_i, we consider k stencils, each formed by k contiguous cells

$$\mathcal{S}^r_i := \{C_{i-r+1}, \dots, C_{i+k-r}\}, \quad \text{for } r = 1, \dots, k. \tag{2.4}$$

For each one of these stencils, there exists a unique polynomial p^r_i of degree at most $k{-}1$ that satisfies (2.3), with a cell average in each cell of the stencil, corresponding to that of $u(x)$:

$$\frac{1}{\Delta x_j} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} p^r_i(\xi) d\xi = \bar{u}_j \quad \text{for } j = i - r + 1 \dots, i + k - r. \tag{2.5}$$

Since the mapping from the cell averages in the stencil $\mathcal{S}^r_i$ to the values $u^+_{i-\frac{1}{2}}$ and $u^-_{i+\frac{1}{2}}$ is linear, the reconstructed function at the interfaces is given by

$$u^{(r)-}_{i+\frac{1}{2}} := \sum_{j=1}^{k} c^k_{rj} \bar{u}_{i-r+j}, \quad u^{(r)+}_{i-\frac{1}{2}} := \sum_{j=1}^{k} \tilde{c}^{\,k}_{rj} \bar{u}_{i-r+j}. \tag{2.6}$$

The coefficients c^k_{rj} and $\tilde{c}^k_{rj}$ depend on the order of accuracy, the stencil, and the cell sizes, but not on the function u itself. Therefore, they can be pre-computed globally for uniform grids or locally in the case of non-uniform grids.

2.3 ENO and WENO schemes

To avoid polynomial reconstructions across cells that contain discontinuities in the solution, the ENO method seeks to identify the *least oscillating* stencil among the k candidates. The selection strategy is based on a recursive evaluation of Newton divided differences and guarantees high-order accuracy provided the solution is sufficiently smooth and the interpolating polynomial is of high order. It can be shown that for polynomials of order $k-1$, the ENO scheme is k-th order accurate [33].

The main restriction of this method, leading to the development of the WENO scheme, is the stencil selection process, in which k different candidate stencils are considered but only one is picked and used in the reconstruction. This results in a k-th order accurate method, even though the covered region contains $2k$–1 cells. In the WENO method this shortcoming is overcome by using a convex combination of all k stencil candidates

$$u_{i+\frac{1}{2}} = \sum_{r=1}^{k} \omega_r u^{(r)}_{i+\frac{1}{2}}. \tag{2.7}$$

Here ω_r are the weights and indicate a measure of regularity of the solution across the considered stencil $\mathcal{S}_i^r$. The method depends critically on the choice of these weights, which, for stability and consistency, must satisfy

$$\omega_r \geq 0 \;\; \forall r = 1, \dots, k, \quad \text{and} \quad \sum_{r=1}^{k} \omega_r = 1. \tag{2.8}$$

Moreover, weights are chosen such that the numerical oscillations in regions of low regularity are avoided, thus emulating the successful ENO idea, and the maximum accuracy is retrieved for smooth solutions, i.e., $(2k-1)$-th order. The final expression for the weights in the classic WENO method is

$$\omega_r = \frac{\varrho_r}{\sum_{s=1}^{k} \varrho_s} \quad \text{with} \quad \varrho_r = \frac{d_r}{(\epsilon + \mathcal{I}_r)^{\rho}}, \tag{2.9}$$

where ϵ is a small positive quantity introduced to avoid division by zero and $\rho \in \mathbb{N}$ controls the sensitivity of the weights with respect to the smoothness indicator $\mathcal{I}_r$: in the limit $\rho \to \infty$ only one stencil contributes to the reconstruction, making the WENO method similar to a ENO method. In general $\epsilon = 10^{-6}$ and $\rho = 2$ are often used. Moreover, d_r is a positive value such that in the smooth case

$$\omega_r = d_r + \mathcal{O}(\Delta x^{k-1}), \quad r = 0, \dots, k-1, \tag{2.10}$$

which suffices to ensure $(2k-1)$-th order of accuracy.

Finally, $\mathcal{I}_r$ is the so-called *smoothness indicator* of stencil $\mathcal{S}^r$. After extensive experiments, the following smoothness indicator was proposed in [34]:

$$\mathcal{I}_r = \sum_{\ell=1}^{k-1} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \Delta x^{2\ell-1} \left(\frac{\partial^{\ell} p_r(x)}{\partial x^{\ell}} \right)^2 dx. \tag{2.11}$$

3 Radial basis functions

Radial basis functions are used in many fields to interpolate scattered data and are here proposed as an alternative to polynomials in the reconstruction step of ENO and WENO methods. We first briefly introduce the general concept of RBF interpolation and refer the readers to [4, 8, 22, 35] for a complete discussion. We then consider the RBF interpolation based on cell averages and discuss the trade-off between numerical stability and accuracy before introducing a simple strategy to compute the "optimal" shape parameters for infinitely smooth RBFs.

3.1 RBF interpolation

Given a discrete set of data points $\varXi \subset \mathbb{R}^d$, traditionally called *centers*, and data values $f_{\boldsymbol{\xi}} = f(\boldsymbol{\xi})$ with $\boldsymbol{\xi} \in \varXi$, the recovery problem consists in reconstructing the unknown function f that approximates the data values at those centers. The interpolation requirements can be expressed by $s|_{\varXi} = f|_{\varXi}$, or alternatively by

$$s(\boldsymbol{\xi}_i) = f(\boldsymbol{\xi}_i), \quad \forall\, i = 1, \ldots, n, \tag{3.1}$$

if the point set is composed of n centers, i.e., $\varXi = \{\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_n\}$, with n corresponding function values $f|_{\varXi} = [f(\boldsymbol{\xi}_1), \ldots, f(\boldsymbol{\xi}_n)]^T$. To perform this interpolation, one can consider a multivariate radial function $\varPhi : \mathbb{R}^d \to \mathbb{R}$, i.e. $\varPhi(\boldsymbol{x}) = \phi(\|\boldsymbol{x}\|)\, \forall \boldsymbol{x} \in \mathbb{R}^d$, where $\|\cdot\|$ is some norm on $\mathbb{R}^d$ – usually the Euclidean norm. Possible choices of the univariate function $\phi : [0, \infty) \to \mathbb{R}$ are shown in Table 1 together with their notion of corresponding order.

RBFs of zeroth order are called *positive definite* and the interpolant is defined as

$$s(\boldsymbol{x}) = \sum_{\boldsymbol{\xi} \in \varXi} \alpha_{\boldsymbol{\xi}} \phi(\|\boldsymbol{x} - \boldsymbol{\xi}\|), \quad \boldsymbol{x} \in \mathbb{R}^d, \tag{3.2}$$

where $\boldsymbol{\alpha} = (\alpha_{\boldsymbol{\xi}})_{\boldsymbol{\xi} \in \varXi}$ is an element of $\mathbb{R}^n$. If the radial function ϕ is *conditionally positive definite* of some order $k > 0$, the right-hand side of (3.2) needs to be augmented with a polynomial $p(\boldsymbol{x}) \in \mathcal{P}_{k-1}^d$, to ensure solvability of the interpolation problem. Here $\mathcal{P}_{k-1}^d$ denotes the space of all $d-$dimensional polynomials of degree at most $k-1$:

$$p(\boldsymbol{x}) = \sum_{|\gamma| < k} \beta_\gamma \boldsymbol{x}^\gamma.$$

Here $\gamma = (\gamma_1, \ldots, \gamma_d) \in \mathbb{N}_0^d$ is a multi-index of degree $|\gamma| = \sum_{i=1}^d \gamma_i$. In this case, the interpolant s in (3.2), is given as

$$s(\boldsymbol{x}) = \sum_{\boldsymbol{\xi} \in \varXi} \alpha_{\boldsymbol{\xi}} \phi(\|\boldsymbol{x} - \boldsymbol{\xi}\|) + p(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathbb{R}^d. \tag{3.3}$$

The centers $\boldsymbol{\xi}$ are used both for shifting the radial basis function and as interpolation points.

We limit ourselves to two conditionally positive definite RBFs: the multiquadrics of order $k := \lceil \nu \rceil$ and the polyharmonic splines of order k. Therefore, to solve the interpolation problem (3.1), we need to determine $n+q$ coefficients,

Infinitely smooth RBFs			**Piecewise smooth RBFs**		
Name	$\phi(\varepsilon, r)$	Order	Name	$\phi(r)$	Order
Multiquadrics	$(1 + (\varepsilon r)^2)^\nu$	$\lceil \nu \rceil$	Polyharmonic Splines	r^{2m-d} for d odd	$m - \lceil d/2 \rceil + 1$
Inverse Multiquadrics	$(1 + (\varepsilon r)^2)^\eta$	0		$r^{2m-d} \log(r)$ for d even	
Gaussians	$\exp(-(\varepsilon r)^2)$	0			

Table 1 Commonly used radial basis functions (RBFs) with parameters $\nu > 0$, $\nu \notin \mathbb{N}, \eta < 0, m \in \mathbb{N}$ and $\varepsilon > 0$, known as the shape parameter.

where $q = \binom{k-1+d}{d}$ is the dimension of the space $\mathcal{P}^d_{k-1}$. These are retrieved by solving (3.1) and by simultaneously requiring that the set of coefficients $\boldsymbol{\alpha} \in \mathbb{R}^n$ is orthogonal to the polynomial space $\mathcal{P}^d_{k-1}(\Xi)$, i.e.

$$\boldsymbol{\alpha} \perp \mathcal{P}^d_{k-1}(\Xi) \quad \text{with } \boldsymbol{\alpha} \in \mathbb{R}^n \iff \sum_{\boldsymbol{\xi} \in \Xi} \alpha_{\boldsymbol{\xi}} \eta(\boldsymbol{\xi}) = 0, \quad \forall \eta \in \mathcal{P}^d_{k-1}. \tag{3.4}$$

This leads to an $(n+q) \times (n+q)$ linear system $\boldsymbol{A\gamma} = \boldsymbol{\lambda}$ on the form

$$\begin{bmatrix} A & P \\ P^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix} = \begin{bmatrix} f|_{\Xi} \\ 0 \end{bmatrix}, \tag{3.5}$$

where $A = (\phi \| \boldsymbol{\xi}_i - \boldsymbol{\xi}_j \|)_{1 \leqslant i,j \leqslant n} \in \mathbb{R}^{n \times n}, P = ((\boldsymbol{\xi}_j)^\gamma)_{1 \leqslant j \leqslant n; |\gamma| < k} \in \mathbb{R}^{n \times q}$, $\boldsymbol{\alpha} \in \mathbb{R}^n$ and $\boldsymbol{\beta} = (\beta_\gamma)_{|\gamma|<k} \in \mathbb{R}^q$.

To ensure uniqueness of the linear system (3.5), the set of interpolation points Ξ has to contain $\mathcal{P}^d_{k-1}$-unisolvent subsets, i.e., the zero polynomial is the only polynomial from $\mathcal{P}^d_{k-1}$ that vanishes on all of the points of Ξ.

Depending on the application, it may be interesting to consider interpolants in which the degree of the polynomial part is arbitrarily high. It has been shown in [11] that higher accuracy can be achieved for problems on circular domains by introducing supplementary polynomials. It is also observed that small shape parameters can then be used even without employing stable algorithms. The drawback of such interpolants is the computational cost; indeed, as it will be explained in Section 4.2, the number of cells per stencil needs to be greater or equal than the degree of the polynomial. Using polynomials of degree at most $q-1$ allows us to use smaller stencils.

3.2 RBF interpolation of cell averages

RBF interpolation can be used in the FVM ENO and WENO reconstructions by considering the cells of a stencil $\mathcal{S}$ as the data points and the corresponding cell averages as the data values. We follow [2] and define the linear functional λ_C as the cell average operator over cell C:

$$\bar{u}_C = \lambda_C(u) = \frac{1}{|C|} \int_C u(\boldsymbol{x}) d\boldsymbol{x}.$$

Given the cell averages of a stencil $\{\lambda_C(u)\}_{C \in \mathcal{S}}$, in analogy with the conservation requirements for polynomial interpolation (2.5), conservation can be expressed as

$$\lambda_C(u) = \lambda_C(s) \quad \text{for all } C \in \mathcal{S}. \tag{3.6}$$

Hence, the averaged interpolant (3.3) becomes

$$s(\boldsymbol{x}) = \sum_{C \in \mathcal{S}} \alpha_C \lambda_C^{\boldsymbol{\xi}} \phi(\| \boldsymbol{x} - \boldsymbol{\xi} \|) + p(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathbb{R}^d, \tag{3.7}$$

where $\lambda_C^{\boldsymbol{\xi}} \phi(\| \boldsymbol{x} - \boldsymbol{\xi} \|) = \frac{1}{|C|} \int_C \phi(\| \boldsymbol{x} - \boldsymbol{\xi} \|) d\boldsymbol{\xi}$. Moreover, the linear constraint (3.4) can be written as

$$\sum_{C \in \mathcal{S}} \alpha_C \lambda_C \left(\eta\right) = 0, \quad \forall \eta \in \mathcal{P}^d_{k-1}. \tag{3.8}$$

Given n cell averages $\boldsymbol{\lambda} = (\lambda_C(u))_{C\in\mathcal{S}} \in \mathbb{R}^n$ over a stencil of size n to solve the interpolation problem (3.6) under the constraint (3.8), one solves a linear system analogous to (3.5):

$$\begin{bmatrix} A & P \\ P^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\lambda} \\ 0 \end{bmatrix}, \tag{3.9}$$

with the unknown parameters $\boldsymbol{\alpha} = (\alpha_C)_{C\in\mathcal{S}} \in \mathbb{R}^n$ and $\boldsymbol{\beta} = (\beta_\gamma)_{|\gamma|<m} \in \mathbb{R}^q$, matrices $A = (\lambda_C^{\boldsymbol{x}} \lambda_Z^{\boldsymbol{\xi}} \phi(\|\boldsymbol{x} - \boldsymbol{\xi}\|))_{C,Z\in\mathcal{S}} \in \mathbb{R}^{n\times n}$ and $P = (\lambda_C(\boldsymbol{x}^\gamma))_{C\in\mathcal{S},|\gamma|<k} \in \mathbb{R}^{n\times q}$.

3.3 On choosing a good shape parameter

Despite the flexibility of RBF interpolation, it is known that by using the standard approach to the RBF interpolation problem, none of the commonly used basis functions combine good approximation behavior with good numerical stability. In [31], this conflict between the theoretical achievable accuracy and the numerical stability is referred to as the *uncertainty relation*, also called the *trade-off principle*. The stability problem is due to a large condition number of the interpolation matrix in (3.9), especially when the distance between cell's centers becomes small.

In general, one radial basis function ϕ can perform better than another one depending on the function to be interpolated [8]. Without further elaboration, we refer the interested reader to [35, 32] and consider other types of solutions to the trade-off dilemma. Indeed, a preconditioning strategy for Lagrange interpolation by polyharmonic splines is developed in [2], while for infinitely smooth RBFs, it is now well-known that the accuracy of the interpolation depends strongly on the choice of the shape parameter ε. It has been shown that the interpolation error and the condition number of the interpolation matrix cannot be kept small simultaneously [31]. For small values of ε we expect superior accuracy, but at the same time the corresponding RBF is increasingly flat, resulting in the columns of $\boldsymbol{A}$ becoming more and more alike, and leading to a numerically unstable method.

This trade-off principle has interested several authors and led to a search for an "optimal" value of the shape parameter, i.e., a value that achieves maximal accuracy, while guaranteeing small condition numbers. Initially, only *ad-hoc* solutions were proposed (see, e.g., [20, 15]), followed by various algorithms and guidelines (see, e.g., [9, 30]) typically for a large number of centers. In [30], Rippa affirms that the value of the optimal ε depends on the approximated function, the number and distribution of data points, the function ϕ and on the precision of the computations.

More recentely, Fornberg and collaborators have investigated the dependency of numerical stability on the shape parameter. In e.g. [7, 12, 14, 13], the authors have introduced and further developed a series of so-called "stable algorithms" that overcome the trade-off principle. In such methods, the linear system (3.9) is not solved directly, thus allowing stable computations also when the shape parameter is very small. The two most relevant algorithms proposed in the literature are the Contour-Padé method (RBF-CP) and the RBF-QR method. While the latter is limited to the Gaussian kernel, the RBF-CP and the recent RBF-RA [3] are more flexible and can be applied to any type of smooth RBFs, including those with an additional polynomial part. However, a major drawback of these algorithms is their computational cost. Moreover, even if the extension to the cell averages of RBFs seems feasible, it remains unclear how to retrieve the coefficients needed for the WENO weights definition (see Section 4.3.1). Therefore, the possibility of applying the RBF-RA method to this context is an ongoing research topic. For the moment we seek to identify an "optimal" value of the shape parameter, which guarantees a good balance between numerical stability and accuracy.

Before choosing the most suitable strategy, note that the number of scattered points that we wish to interpolate is very small. Typically, one would use ENO methods of order at most 4, i.e., 7-th order WENO method. This corresponds to stencils of size 2 to 5 in 1D. This makes some algorithms unsuitable for our problem, as for example the leave-one-out cross validation technique proposed in [30]. Other procedures are computationally expensive and we therefore propose to use the following simple strategy, exposed in Section 17.1 of [8]. The idea is to perform a series of interpolation experiments for various values of the shape parameter and identify the one which yields the minimum interpolation error. As already discussed, the optimal shape parameter may be different for different functions (see e.g., [5, 30]), but our numerical examples show that using the same shape parameter for different problems typically yields good results. Exceptions may be needed for very small mesh sizes and high orders of the interpolants.

For the sake of simplicity, we limit ourselves to the one-dimensional case and consider the following smooth function

$$f(x) = \sin(2\pi x), \quad x \in [0, 1] \tag{3.10}$$

with periodic boundary conditions. After uniformly partitioning the domain into N cells, with $N = 2^i$ for $i = 4, \dots, 8$, we evaluate the interpolation error between the reconstructed values at the interfaces and the exact function. The reconstructed values are obtained using the WENO reconstruction procedure with multiquadrics with $\nu = \{\frac{1}{2}, \frac{3}{2}, \frac{5}{2}, \frac{7}{2}\}$, i.e., of order $k = 1, 2, 3, 4$. Note that the WENO reconstruction with RBFs is essentially equivalent to the classic WENO reconstruction except for the definition of the weights.

At each interface there are two reconstructed values, one from the left and one from the right. Thus, several interpretations of the interpolation error are possible. We follow the interpretation in [10] and evaluate the error at the interfaces as

$$\text{Err} = \|u^-_{i+\frac{1}{2}} - f(x_{i+\frac{1}{2}})\|_{L^p_h} + \|u^+_{i+\frac{1}{2}} - f(x_{i+\frac{1}{2}})\|_{L^p_h}, \tag{3.11}$$

where the discrete norm L^p_h with $h = 1/N$ is defined as

$$\|(.)_i\|_{L^p_h} = \left(\sum_{i=1}^{N-1} h|(.)_i|^p \right)^{1/p} \text{for } p < \infty, \qquad \|(.)_i\|_{L^\infty_h} = \max_i |(.)_i|. \tag{3.12}$$

Results are shown in Fig. 2 for $p = 1$. The graphs generally agree with our expectations: as the cells centers get closer, i.e., as h decreases, the numerical instability problem due to small ε becomes more severe. This phenomenon is more evident for higher-order RBFs. Indeed, one can visibly recognize a "breakdown" point $P = (\varepsilon_P, Err_P)$ that distinguishes stable solutions from unstable solutions. We require the optimal shape parameter to be outside the region of instability, i.e.

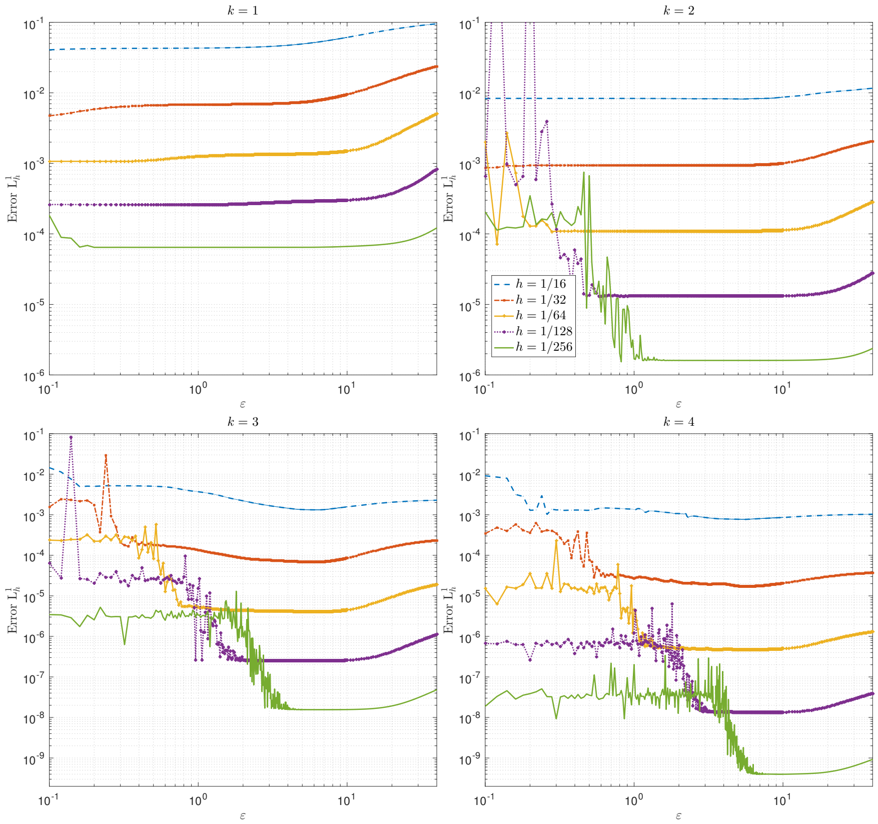

Fig. 2 Log-log plots of the RBF-WENO reconstruction errors with respect to the shape parameter ε using multiquadric interpolants $\phi = (1 + (\varepsilon r)^2)^\nu$ with $\nu = \{\frac{1}{2}, \frac{3}{2}, \frac{5}{2}, \frac{7}{2}\}$, i.e., of order $k = \lceil \nu \rceil$, on uniform grids

$\varepsilon^* \geq \varepsilon_P$. Moreover, the reconstruction becomes less accurate as ε increases further. Therefore, a value in the immediate neighborhood of ε_P represents a good candidate to be the optimal shape parameter.

Even though the optimal shape parameter clearly depends on both the order of the interpolant and the cell size, i.e. $\varepsilon^* = \varepsilon(k, h)$, in practice it is more convenient to use a single shape parameter that depends only on the order of the RBFs for every mesh refinement, i.e. $\varepsilon^* = \varepsilon(k)$ for all h. To avoid stability issues, we choose this value to be slightly bigger than the breakdown point for the smallest h we consider. For k up to 4, chosen values are indicated in Table 2. We observe that for these values, the reconstruction errors do not vary much when considering slightly smaller or bigger ε, i.e. the minima for all h lie in a region where the error lines flatten out. This suggest that the risk of decreased accuracy associated with choosing a bigger ε^* than the actual minimum one for the corresponding h is limited.

As shown in Section 5, for more complex problems breakdown points are often shifted to larger ε than those in Table 2 (see Fig. 7 and 10). This is done to overcome spurious oscillations that we observe, especially as h decreases or k increases. The choice of the optimal shape parameter in these cases is based on experimental tests and the possibility of introducing an adaptive technique that chooses the shape parameter locally is an ongoing research topic. In a recent paper by Guo and Jung [18], the authors propose a way out of this problem by suggesting a switching technique. This method first identifies the regions of sharp gradients or a discontinuity of the solution and then locally uses the standard WENO method to guarantee the essentially non-oscillatory property of the method. The drawback of this technique is that it can not be extended to general problems with non-uniform grids as it does not exploit the meshless properties of RBFs but simply builds on top of the standard ENO/WENO method.

Order of ϕ	Chosen shape parameter
k	ε
1	1
2	3
3	6
4	10

Table 2 Chosen "optimal" shape parameter for RBF-WENO reconstruction based on multiquadrics.

4 RBF-WENO method

Given these building blocks, we can now apply RBF interpolation in the FVM reconstruction step to obtain RBF-WENO methods. In addition to the different interpolating function, the RBF-WENO method distinguishes itself from the polynomial WENO method in the definition of the weights and the smoothness indicator.

We recall that an RBF-WENO was proposed in [2]. However, in that method the smoothness indicator is restricted to polyharmonic splines, whose native space is the Beppo-Levi space $\mathrm{BL}_k(\mathbb{R}^d)$ equipped with the semi-norm $|\cdot|_{\mathrm{BL}_k(\mathbb{R}^d)}$ (see [2] for definitions). Since the interpolant $s \in \mathrm{BL}_k(\mathbb{R}^d)$ in (3.7), satisfying the interpolation conditions (3.6), is the unique minimizer of $|\cdot|_{\mathrm{BL}_k(\mathbb{R}^d)}$ among all the interpolants $v \in \mathrm{BL}_k(\mathbb{R}^d)$ that satisfy (3.6), the energy functional $|\cdot|_{\mathrm{BL}_k(\mathbb{R}^d)}$ is used as the smoothness indicator for the WENO reconstruction in that work. RBFs have also been used in combination with the ENO and WENO methods in [18], where multiquadrics or the Gaussians are adopted as non-polynomial interpolation basis. In that work the coefficients (2.6) are derived by perturbing the classic polynomial coefficients and depend directly on the shape parameter, achieving a higher order of convergence than the polynomial ENO and WENO methods. The method uses the classic smoothness indicator and weights defined in [33].

4.1 RBF-WENO reconstruction

Let us consider the one-dimensional case: after partitioning the computational domain $\Omega_x = [a,b]$ into N cells C_i defined by (2.2), we consider n stencils of size m of the form (2.4). For each of these we construct an interpolant s_r with $r = 1,\ldots,n$ and define $s^{(i)}$ as a convex combination

$$s^{(i)}(x) = \sum_{r=1}^{n} \omega_r s_r(x). \tag{4.1}$$

The weights satisfy (2.8). To compute the two WENO reconstructions, the final interpolant $s^{(i)}(x)$ will be evaluated at the left and right interfaces of cell C_i, at $x_{i-\frac{1}{2}}$ and $x_{i+\frac{1}{2}}$, respectively, yielding

$$u^{+}_{i-\frac{1}{2}} = s^{(i)}(x_{i-\frac{1}{2}}), \quad \text{and} \quad u^{-}_{i+\frac{1}{2}} = s^{(i)}(x_{i+\frac{1}{2}}).$$

4.2 Stencil selection

In 1D, the number of considered stencils for each cell must be smaller than $m := \#\mathcal{S}_r \,\forall\, r = 1,\ldots,n$, to avoid considering the same group of cells more than once. We consider n stencils of size n, i.e., $m = n$. This requirement is not necessary in multiple dimensions, especially for unstructured meshes, where the stencil design is characterized by additional flexibility and the number of stencils needed in the reconstruction might be larger than the number of cells per stencil, i.e., $n > m$.

Moreover, we require $n \geq q$ (see [22] for more details). If $n = q$, the interpolant (3.7) is uniquely defined by the polynomial part. Indeed, if P in (3.9) is a square matrix, then $P^T\boldsymbol{\alpha} = 0$ implies $\boldsymbol{\alpha} = 0$ unless P is singular. For all numerical experiments in Section 5 we choose $n = k + 1$.

4.3 Weights

The weights ω_r in (4.1) assume the same form as in the classic WENO method (2.9), but they differ in the definition of the smoothness indicator $\mathcal{I}_r$. We recall that the choice of the weights ω_r is guided by the dual goal of achieving higher accuracy as compared to the ENO method and reduction of artificial oscillations near discontinuities of the solution. The first feature is attained thanks to the introduction of the linear weights d_r, while the second one depends on the choice of the smoothness indicator.

4.3.1 Linear weights

In the classic WENO method the linear weights d_r (2.9) guarantee a global $(2k-1)$-th order of accuracy in smooth regions.

We observe that for general meshes, two types of WENO methods are discussed in the literature (see e.g. [37]). The first one (type I) corresponds to WENO schemes that do not gain additional accuracy by combining small stencils. This is closer to ENO schemes as the weights ω_r are not introduced to increase the order of accuracy, but to avoid spurious oscillations and for the sake of nonlinear stability. Type II WENO schemes are designed so that the order of accuracy is higher than each of the reconstructions. Such methods are more complicated to construct, primarily because finding the linear weights may lead to negative large linear weights [27]. In general, for type I methods, the weights ω_r do not depend on the evaluation point, while this is no longer true for type II methods. In 1D, the left and right reconstructions of cell C_i are given by

$$u^+_{i-\frac{1}{2}} = \sum_{r=1}^{n} \omega_r s_r(x_{i-\frac{1}{2}}), \quad \text{and} \quad u^-_{i+\frac{1}{2}} = \sum_{r=1}^{n} \tilde{\omega}_r s_r(x_{i+\frac{1}{2}}), \tag{4.2}$$

where

$$\tilde{\omega}_r = \frac{\tilde{\varrho}_r}{\sum_{j=1}^{n} \tilde{\varrho}_j} \quad \text{with} \quad \tilde{\varrho}_r = \frac{\tilde{d}_r}{(\epsilon + \mathcal{I}_r)^\rho}, \quad \tilde{d}_r = d_{n-r+1}.$$

The classic WENO method in 1D, described in Section 2.3, and the one introduced in [18] are examples of type II WENO methods. While the RBF-WENO method designed in [2] is a type I WENO method, using $d_r = 1$ for all r, we consider both types of methods and choose type II WENO when the problem is sufficiently smooth.

To derive the coefficients, we proceed as in [33]. The interpolant s in (3.7) must be rewritten as a function of the cell averages of the underlying stencil, leading to

$$s(\boldsymbol{x}) = \sum_{C\in\mathcal{S}} \alpha_C \lambda^{\boldsymbol{\xi}}_C \phi(\|\boldsymbol{x} - \boldsymbol{\xi}\|) + \sum_{|\gamma|<k} \beta_\gamma \boldsymbol{x}^\gamma = \sum_{C\in\mathcal{S}} c_C(\boldsymbol{x}) \bar{u}_C. \tag{4.3}$$

Given a stencil $\mathcal{S}_r$, composed by n cells on a one-dimensional mesh and a RBF of order k with the polynomial part of degree up to $q-1$, we have

$$s_r(x) = \sum_{j=1}^{n} \alpha^r_j f^r_j(x) + \sum_{\ell=1}^{q} \beta^r_\ell g_\ell(x) = \sum_{j=1}^{n} c^n_{r,j}(x) \bar{u}_j, \tag{4.4}$$

where $f^r_j = \lambda^{\xi}_{C^r_j} \phi(\|x - \xi\|)$, $C^r_j \in \mathcal{S}_r$ and $g_\ell = x^{\ell-1}$. Equation (4.3) and (3.9) imply

$$\boldsymbol{c}_r = \Big[\boldsymbol{A}^{-1}[\boldsymbol{f}_r; \boldsymbol{g}]\Big]_{1:n},$$

where $\boldsymbol{c}^n_r = [c^n_{r,1}, c^n_{r,2}, \dots, c^n_{r,n}]^T \in \mathbb{R}^n$, $\boldsymbol{f}_r \in \mathbb{R}^n$ and $\boldsymbol{g} \in \mathbb{R}^q$.

To compute $c^n_{r,j}(x)$, the unknown constants $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ in (4.4) need to be expressed as functions of the cell averages in the right hand-side. We remind the reader that the above-mentioned constants $c^n_{r,j}$ depend on the evaluation point x, similar to the weights. More details on this can be found in the descriptions of the WENO methods, e.g. in [33]. To determine the $\boldsymbol{d} = [d_1, \dots, d_n]^T$coefficients such that (2.10) is satisfied, one needs to compare the constants $c^n_{r,j}$ that have been computed for all cells $C_j \in \mathcal{S}_r, \forall\, r = 1, \dots, n$ with the coefficients corresponding to the unique stencil of size $2n-1$, covering the whole region.

After computing the n coefficients for each of the n stencils and for both interfaces, one needs to recover the coefficients, indicated as $\boldsymbol{c}^{2n-1}_n$, corresponding to the stencil of size $2n-1$, including all cells in the n stencils. To determine the linear weights $\boldsymbol{d}$, we solve

$$\begin{bmatrix} c^n_{1,1} & \cdots & \cdots & c^n_{1,n} & 0 & \cdots & \cdots & 0 \\ 0 & c^n_{2,1} & c^n_{2,2} & \cdots & c^n_{2,n} & 0 & \cdots & \vdots \\ \vdots & & \ddots & & & & \ddots & \vdots \\ \vdots & & & c^n_{n-1,1} & c^n_{n-1,2} & \cdots & c^n_{n-1,n} & 0 \\ 0 & \cdots & \cdots & 0 & c^n_{n,1} & c^n_{n,2} & \cdots & c^n_{n,n} \end{bmatrix}^T \boldsymbol{d} = \boldsymbol{c}^{2n-1}_n. \tag{4.5}$$

We note that the $\boldsymbol{c}^n_r$ coefficients for 1D uniform grids obtained with RBF reconstructions are asmyptotically equal to those for the classic WENO method [33]. To prove this result one needs to analytically invert the global matrix $\boldsymbol{A}$ in (3.9). We show in the Appendix that these coefficients are equal to their classic WENO counterpart for $n = 2, 3$ up to a term $\mathcal{O}(\varepsilon\Delta x)^2$. Inserting these into (4.5) we recover, for $k=1$, the linear weights

$$d_1(x_{i-\frac{1}{2}}) = \frac{2}{3} + \mathcal{O}(\varepsilon\Delta x)^2 \quad \text{and} \quad d_2(x_{i-\frac{1}{2}}) = \frac{1}{3} + \mathcal{O}(\varepsilon\Delta x)^2, \tag{4.6}$$

which indeed satisfy (2.10). We conjecture that this result holds also for higher order schemes and thus, for any order k and uniform grids, we use the classic linear coefficients given in [33].

Note that, due to a different notation explained in detail in Section 5.1, the desired asymptotic behavior for multiquadrics interpolants is $\mathcal{O}(\Delta x^k)$ rather than $\mathcal{O}(\Delta x^{k-1})$ as in (2.10), which is still verified by (4.6).

4.3.2 Smoothness indicator

Artificial oscillations in regions of low regularity are avoided thanks to the smoothness indicator, which needs to be small when the interpolant varies slowly across the stencil and large if the solution contains a sharp gradient or a discontinuity across the stencil. We consider the following definition for each cell $C_i \in \Omega_x$

$$\mathcal{I}^r_i = \Delta x_i \left| \sum_{j=1}^{n} \alpha_j \int_{C_i} \frac{\partial^2}{\partial x^2} \left[\lambda^{\xi}_{C^r_j} \phi(\|x-\xi\|) \right] dx \right| + \sum_{\ell=1}^{k-1} \Delta x_i^{2\ell-1} \int_{C_i} \left(\frac{\partial^\ell p(x)}{\partial x^\ell} \right)^2 dx. \tag{4.7}$$

Here $r = 1, \ldots, n$ indicates one among the n possible stencils for cell C_i. This expression is composed of two parts: one depends on the RBFs and one on the polynomial part of the RBF interpolant (3.7). The first part generalizes the smoothness indicator introduced in [2] to all types of conditionally positive definite RBFs, while the second component corresponds exactly to the smoothness indicator (2.11) in the classic WENO method. Furthermore, the smoothness indicator is independent of the $\boldsymbol{x}$ argument of the interpolant s, i.e. it is the same for both the left and right reconstruction.

5 Numerical results

In the following we present numerical results to demonstrate the performance of the RBF-WENO method on uniform one-dimensional grids composed by N cells of the form (2.2) with $\Delta x := \Delta x_i$ for all $i = 1, \dots, N$.

In the first part we compare the accuracy results for the RBF-WENO reconstructions using polyharmonic splines (PHS) and multiquadrics (MQ) interpolants, while in the second part we consider evolution problems using only MQ interpolants as these generally perform better.

5.1 Reconstruction accuracy

Let us first consider the smooth function (3.10) with periodic boundary conditions. The reconstruction error at the interfaces is evaluated by (3.11).

Convergence for both MQ and PHS reconstructions is shown in Fig. 3. For the MQ interpolants, the shape parameters is given in Table 2 for all mesh refinements. We observe that, given the same order k of the RBF interpolant, reconstructions obtained with MQ converge faster than those obtained with PHS, i.e. MQ reconstructions of order k converge as $\mathcal{O}(h^{k+1})$, while PHS reconstructions of order k only converge as $\mathcal{O}(h^k)$. These results on PHS confirm the results obtained in [2] in terms of convergence rate. Because of the higher convergence rate, we focus on the multiquadrics in the following numerical examples.

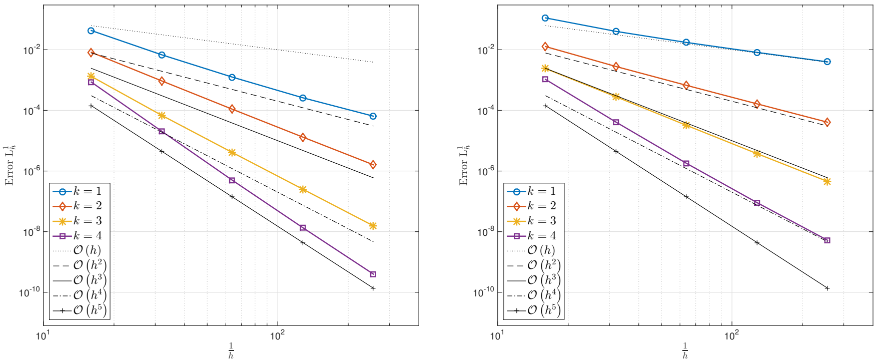

Fig. 3 Order of convergence of the RBF-WENO reconstructions using multiquadrics (left) and polyharmonic splines (right) interpolants. In both cases the RBFs are of order $k = 1, \dots, 4$

Table 3 shows the MQ reconstruction errors and rates for $k = 1, 2, 3, 4$, using the weights of type I, while errors and rates for the MQ reconstruction using type II weights are shown in Table 4 for $k = 1, 2, 3$. Note that for $k = 3$ we use the same coefficients as for the classic WENO method. Both tables indicate errors and rates computed with the discrete L^p norm with $p = 1, 2, \infty$. The results are summarized in Fig. 4, where we observe that type I reconstructions of order $2k$ have approximately the same rate of convergence of type II reconstructions of order k, even though, in general, type II reconstructions of order k are less accurate than type I reconstructions of order $2k$.

	N	L_h^1		L_h^2		L_h^∞	
		error	rate	error	rate	error	rate
$k = 1$	16	4.3098e-02	-	5.1873e-02	-	1.0626e-01	-
	32	6.8078e-03	2.66	9.6642e-03	2.42	2.5436e-02	2.06
	64	1.2518e-03	2.44	1.7410e-03	2.47	5.5854e-03	2.19
	128	2.6052e-04	2.26	3.0127e-04	2.53	6.4367e-04	3.12
	256	6.4467e-05	2.01	7.1632e-05	2.07	1.0542e-04	2.61
$k = 2$	16	8.2462e-03	-	8.6469e-03	-	1.4809e-02	-
	32	9.3694e-04	3.14	1.0107e-03	3.10	1.6580e-03	3.16
	64	1.0942e-04	3.10	1.1958e-04	3.08	1.8459e-04	3.17
	128	1.3133e-05	3.06	1.4463e-05	3.05	2.1438e-05	3.11
	256	1.6056e-06	3.03	1.7756e-06	3.03	2.5727e-06	3.06
$k = 3$	16	1.3246e-03	-	1.5593e-03	-	2.8756e-03	-
	32	6.8415e-05	4.28	7.7316e-05	4.33	1.3497e-04	4.41
	64	4.0809e-06	4.07	4.5338e-06	4.09	6.9651e-06	4.28
	128	2.5145e-07	4.02	2.7892e-07	4.02	4.0782e-07	4.09
	256	1.5565e-08	4.01	1.7298e-08	4.01	2.4816e-08	4.04
$k = 4$	16	8.5821e-04	-	1.1340e-03	-	2.4640e-03	-
	32	2.0425e-05	5.39	2.7700e-05	5.36	8.4250e-05	4.87
	64	4.9668e-07	5.36	6.1388e-07	5.50	1.7451e-06	5.59
	128	1.3486e-08	5.20	1.5495e-08	5.31	4.8074e-08	5.18
	256	3.9755e-10	5.08	4.4518e-10	5.12	1.1891e-09	5.34

Table 3 Reconstruction errors and rates of convergence for type I RBF-WENO reconstruction using multiquadrics interpolants of order $k = 1, 2, 3, 4$.

	N	L_h^1		L_h^2		L_h^∞	
		error	rate	error	rate	error	rate
$k = 1$	16	4.0428e-02	-	4.6621e-02	-	9.9786e-02	-
	32	6.0498e-03	2.74	8.1698e-03	2.51	2.3831e-02	2.07
	64	8.1178e-04	2.90	1.2165e-03	2.75	4.7461e-03	2.33
	128	6.2686e-05	3.69	7.8158e-05	3.96	2.2420e-04	4.40
	256	3.1762e-06	4.30	3.1972e-06	4.61	3.8039e-06	5.88
$k = 2$	16	1.1808e-03	-	1.5192e-03	-	3.1412e-03	-
	32	4.8086e-05	4.62	7.6201e-05	4.32	2.4483e-04	3.68
	64	1.6721e-06	4.85	3.5253e-06	4.43	1.6927e-05	3.85
	128	5.4881e-08	4.93	1.5900e-07	4.47	1.1093e-06	3.93
	256	2.9747e-09	4.21	7.3338e-09	4.44	7.0864e-08	3.97
$k = 3$	16	3.5436e-04	-	4.5772e-04	-	1.0694e-03	-
	32	5.9228e-06	5.90	8.9202e-06	5.68	3.0222e-05	5.15
	64	7.3956e-08	6.32	1.1917e-07	6.23	4.5238e-07	6.06
	128	8.6656e-10	6.42	1.3934e-09	6.42	6.4336e-09	6.14
	256	3.2076e-11	4.76	3.8268e-11	5.19	1.0676e-10	5.91

Table 4 Reconstruction errors and rates of convergence for type II RBF-WENO reconstruction using multiquadrics interpolants of orders $k = 1, 2, 3$.

Note that in a classic WENO method, the use of linear weights implies a $2k-1$ convergence order, while type II RBF-WENO method appears to yield a higher order of convergence, close to $2k$+1. However, this is caused by the use of a different

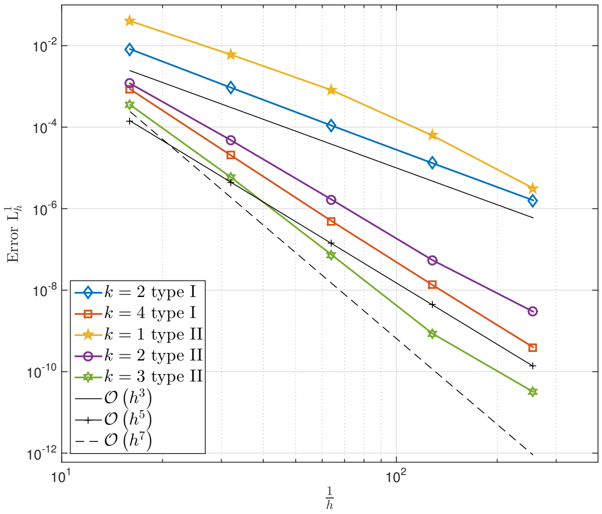

Fig. 4 Comparison of the order of convergence of the type I and type II RBF-WENO reconstructions using multiquadrics interpolants of order $k = 2, 4$ and $k = 1, 2, 3$ respectively

notation. In the classic WENO method, for an interpolant polynomial of order k–1 we consider a stencil of size k, i.e. $\#\mathcal{S} = k$. Combining k such stencils, the stencil area comprises $2k-1$ cells, yielding a global order of convergence of $\mathcal{O}(h^{2k-1})$. Conversely, in the type II RBF-WENO method, for a MQ interpolant of order k we consider n small stencils of size $\#\mathcal{S} = k+1$ and a large stencil that includes $2k$ cells. This is consistent with the observed order of convergence for multiquadrics.

5.2 Evolution Problems

We now test the performance of the method for solving one-dimensional conservation laws, starting with scalar equations such as the linear advection equation, the nonlinear Burgers equations, and the non-convex Buckely-Leverett equation. Later we consider systems of equations such as the Euler equations. Furthermore, we have observed good results also for the shallow water equations, a two-dimensional system of conservation laws. Throughout all numerical examples we use the Lax-Friedrichs numerical flux

$$F_{LF}(a, b) = \frac{f(a) + f(b)}{2} - \frac{\alpha}{2}(b - a), \tag{5.1}$$

where α is defined as $\alpha = \max_u |{}^{\partial f}/_{\partial u}|$ on a relevant set of u (see [33]). For time integration we use an SSP RK$-$3 [17, 16]. Regarding the CFL condition [6], unless otherwise specified, we use $CFL = 0.1$.

5.2.1 The linear advection equation

Consider the linear advection equation

$$u_t + u_x = 0, \quad x \in [0, 1], \tag{5.2}$$

with the exact solution $u(x,t) = u_0(x - t)$, where $u_0(x)$ is the initial condition. Let us consider a smooth initial profile

$$u_0(x) = \sin(2\pi x), \tag{5.3}$$

and assume periodic boundary conditions.

We seek to show that the orders of convergence obtained for the simple reconstruction are recovered also for time-dependent problems. Since the Runge-Kutta method is of order 3, we cannot expect a global order of convergence greater than 3 for every CFL number. We therefore limit ourselves to $k < 4$ and use smaller CFL numbers when increasing the order of the RBF interpolant, i.e., for $k=1,2$ we use $CFL = 0.2$, while for $k = 3$ we use $CFL = 0.005$. Shape parameters are given in Table 2 for every order k and refinement h. The errors and the rates of convergence for the numerical solutions obtained with type I RBF-WENO method at $t = 0.025$ are given in Table 5. These values agree with those obtained for the reconstruction problem in Table 3. Table 6 shows the errors and convergence rates for type II RBF-WENO method for $k = 1, 2, 3$ by further reducing the CFL also for $k = 2$. Results for $k = 1$ and $k = 3$ confirm the results obtained in Table 4. For $k = 2$ we observe a loss of one in the convergence rate, i.e., convergence of order 4 instead of $2k + 1 = 5$. The reason for this loss of accuracy currently remains an open question but is likely linked to the design of the weights.

	N	L_h^1		L_h^2		L_h^∞	
		error	rate	error	rate	error	rate
$k=1$	16	1.4412e-02	-	1.7131e-02	-	2.3527e-03	-
	32	3.6024e-03	2.00	5.2646e-03	1.70	4.7772e-04	2.30
	64	7.7343e-04	2.22	1.3758e-03	1.94	8.0021e-05	2.58
	128	7.1183e-05	3.44	1.0837e-04	3.67	4.2951e-06	4.22
	256	1.0614e-05	2.75	1.1536e-05	3.23	7.2103e-08	5.90
$k=2$	16	1.2482e-03	-	1.2777e-03	-	1.5839e-04	-
	32	1.3369e-04	3.22	1.5121e-04	3.08	1.0452e-05	3.92
	64	1.6456e-05	3.02	1.8675e-05	3.02	6.6014e-07	3.98
	128	2.0091e-06	3.03	2.2614e-06	3.05	4.0600e-08	4.02
	256	2.4804e-07	3.02	2.7775e-07	3.03	2.5539e-09	3.99
$k=3$	16	3.7259e-04	-	4.9728e-04	-	7.6121e-05	-
	32	1.8954e-05	4.30	2.3803e-05	4.38	1.9609e-06	5.28
	64	1.1786e-06	4.01	1.7982e-06	3.73	9.9467e-08	4.30
	128	5.7907e-08	4.35	8.2927e-08	4.44	2.9449e-09	5.08
	256	2.5429e-09	4.51	2.8495e-09	4.86	3.0941e-11	6.57

Table 5 Errors and rates of convergence for the linear advection equation with smooth initial condition solved using a type I RBF-WENO method with MQ interpolants of order $k=1,2,3$.

	N	L_h^1		L_h^2		L_h^∞	
		error	rate	error	rate	error	rate
$k=1$	16	1.2148e-02	-	1.5754e-02	-	2.4425e-03	-
	32	2.9989e-03	2.02	4.8924e-03	1.69	4.8270e-04	2.34
	64	6.1949e-04	2.28	1.2608e-03	1.96	7.8086e-05	2.63
	128	3.6460e-05	4.09	9.2931e-05	3.76	4.1230e-06	4.24
	256	5.7053e-07	6.00	2.3008e-06	5.34	6.8282e-08	5.92
$k=2$	16	4.7607e-04	-	5.4401e-04	-	7.5567e-05	-
	32	2.9179e-05	4.03	4.4938e-05	3.60	4.4938e-06	4.07
	64	1.6060e-06	4.18	3.4787e-06	3.69	2.3409e-07	4.26
	128	8.6959e-08	4.21	2.5956e-07	3.74	1.1397e-08	4.36
	256	5.1272e-09	4.08	1.9221e-08	3.76	5.5532e-10	4.36
$k=3$	16	1.9518e-04	-	2.2578e-04	-	2.7580e-05	-
	32	4.6146e-06	5.40	7.3483e-06	4.94	7.4349e-07	5.21
	64	7.2397e-08	5.99	1.4075e-07	5.71	9.4948e-09	6.29
	128	7.5322e-10	6.59	1.4510e-09	6.60	5.4878e-11	7.43
	256	1.5670e-11	5.59	2.0120e-11	6.17	3.7801e-13	7.18

Table 6 Errors and rates of convergence for the linear advection equation with smooth initial condition solved using a type II RBF-WENO method with MQ interpolants of order $k=1,2,3$.

5.2.2 The Burgers equation

Now consider Burgers equation

$$u_t + \left(\frac{u^2}{2}\right)_x = 0, \quad x \in \Omega_x = [0,1], \tag{5.4}$$

with smooth initial condition (5.3). Figure 5 shows the evolution of the solution at four times $t=[0.05, 0.15, 0.25, 0, 35]$. Observe that the solution develops a discontinuity at $x={}^1\!/_2$. The dots represent the numerical solution obtained with the type I RBF-WENO method with MQ interpolants of order $k=2$ on a grid composed by $N=50$ cells, while the solid line indicates the reference solution obtained on a very fine grid. No oscillations are observed close to the discontinuity.

Let us now consider the discontinuous initial condition

$$u_0(x) = \begin{cases} u_l = 2 & \text{if} \quad x < \frac{1}{2} \\ u_r = 1 & \text{if} \quad x \geq \frac{1}{2} \end{cases} . \tag{5.5}$$

The solution is a shock wave moving at a speed $a = {}^3\!/_2$ by the Rankine-Hugoniot condition. The solution at $t = 0.1$, obtained with the type I RBF-WENO method with MQ interpolants, is shown in Fig. 6 for $k=1,2,3,4$ and for increasing refinements from $h=2^{-4}$ to $h=2^{-8}$. Figure 7 shows the solution for $k = 1$ and $h = {}^1\!/_{128}$ as ε is increased, i.e., $\varepsilon = 1, 2, 3$. For $\varepsilon = 1$ we observe small oscillations close to the discontinuity. These spurious oscillations decrease in amplitude as the shape parameter is increased, i.e., for $\varepsilon = 2, 3$. We observe a similar behavior also for higher order of interpolants and different values of h. The chosen shape parameters for every k and h are indicated in the legend of Fig. 6 and are purely experimental.

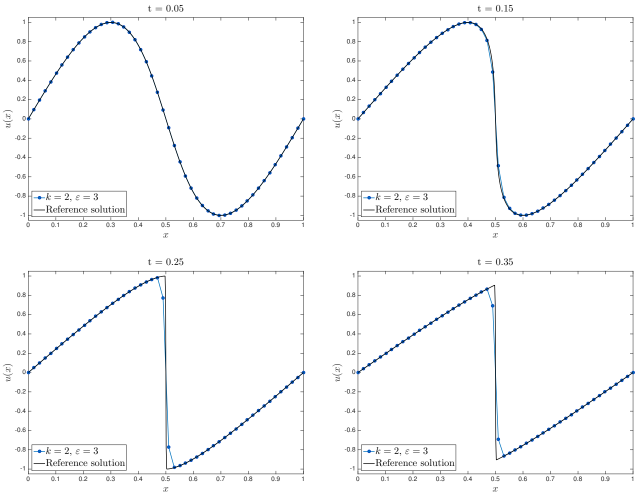

Fig. 5 Solution of Burgers equation with smooth initial condition at four times. The numerical solution (dots) is obtained using type I RBF-WENO method with MQ interpolant of order $k = 2$ on a grid of size $h = 0.02$

5.2.3 The Buckley-Leverett equation

The Buckley-Leverett equation models a two-phase fluid flow of two immiscible and incompressible fluids in a porous media. It is an example of a scalar non-convex conservation law often used in oil-reservoir simulations. For a detailed discussion of non-convex problems solved with classic WENO method and discontinuous Galerkin method we refer to [29]. Given the one-dimensional conservation law (2.1), $u(x, t)$ represents the water saturation at time t in a porous medium such as rock or sand, i.e., the fraction of fluid that is water. Therefore, $1 - u$ represents the fraction of the liquid that is oil. The flux function is

$$f(u) = \frac{u^2}{u^2 + a(1 - u)^2}, \tag{5.6}$$

where $a < 1$ is a constant that indicates the ratio of the viscosities of the two fluids; in our example we take $a = 1/2$. The following discontinuous initial condition

$$u(x, 0) = \begin{cases} 1, & x < 0 \\ 0, & x \geq 0 \end{cases} \quad x \in \Omega_x = [-5, 5], \tag{5.7}$$

indicates that there is water in the left part of the domain and oil in the right one. Oil is subsequently displaced by water being pumped from the left and, as water enters, the sharp interface between water and oil is not maintained. Behind the propagating shock wave there is a mixture of oil and water, represented by a rarefaction wave. This type of wave is called a *compound* wave [25]. Figure 8 shows the solution at time $t = 1.5$ for type I RBF-WENO method with MQ interpolation of order $k = 1, 2, 3, 4$ for several mesh sizes. The exact solution, indicated with a solid dark line, is obtained by the convex-hull construction [25].

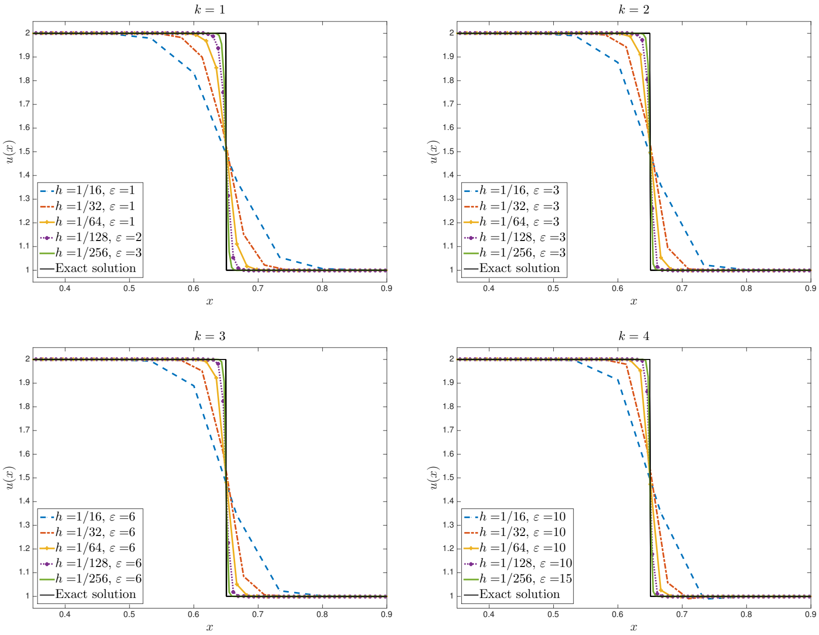

Fig. 6 Qualitative comparison of type I RBF-WENO solution of Burgers equation with a discontinuous initial condition using MQ interpolants of order $k = 1, 2, 3, 4$ at time $t = 0.1$

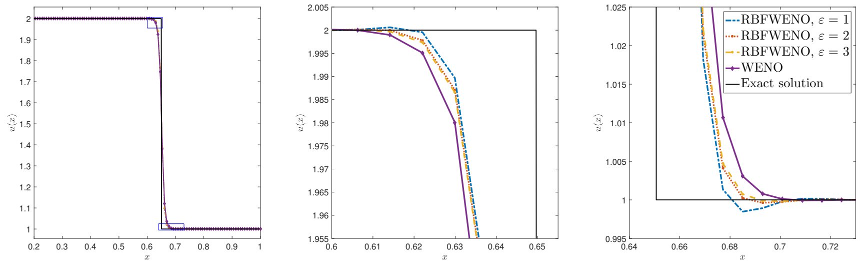

Fig. 7 Type I RBF-WENO solution of Burgers equation with a discontinuous initial condition using MQ interpolants of order $k = 1$ on a uniform grid of size $h = 1/128$ for increasing values of ε. The solution is shown on its entire domain (left) and two details of the solution are highlighted: the left side of the shock wave (center) and the right side of the shock wave (right). The classic WENO solution of corresponding order is also shown.

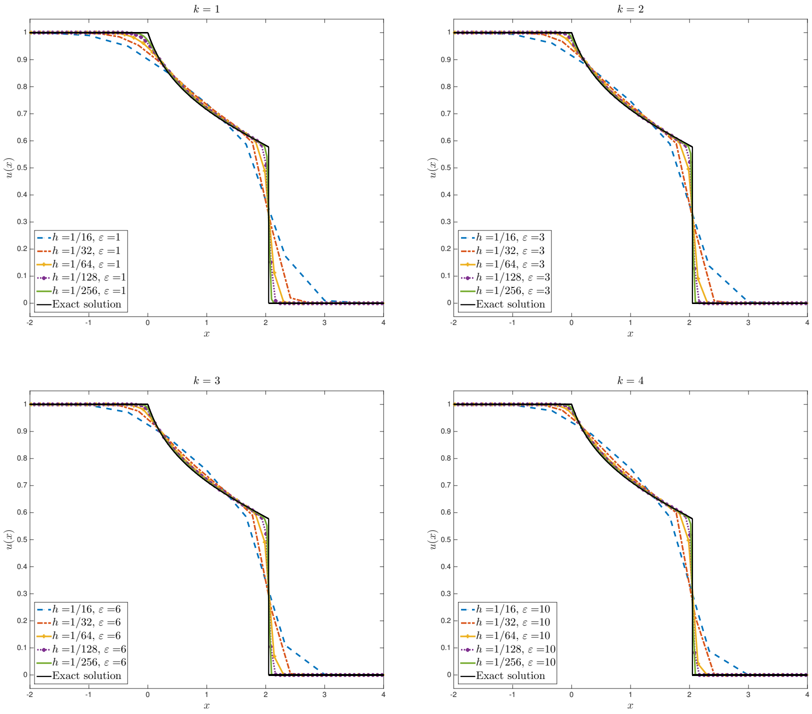

Fig. 8 Solution of the Buckley-Leverett equation with discontinuous initial condition at $t = 1.5$ obtained with type I RBF-WENO using MQ interpolants of order $k = 1, 2, 3, 4$ and ε given in Table 2. Grid refinements go from $h = 2^{-4}$ to $h = 2^{-8}$

5.2.4 The Euler equations

The one-dimensional Euler equations of gas dynamics are regarded as the classic benchmark equation for new methods for conservation laws. The Euler equations are a system of conservation laws (2.1) composed by $d + 2$ equations. For $d = 1$, the state vector and the flux function take the form

$$\boldsymbol{u}(x,t) = [\rho, \rho u, E]^T, \quad \boldsymbol{f}(\boldsymbol{u}) = [\rho u, \rho u^2 + p, (E + p)u]^T, \tag{5.8}$$

where ρ is the density, ρu the momentum, E the energy, and p the pressure. The *equation of state* for an ideal gas is

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

and closes the system of equations. Here γ is a fluid dependent constant and we take $\gamma = {}^7/_5$, as for atmospheric gasses.

To avoid spurious oscillations, we use the characteristic variables for our reconstructions. In particular, for each cell in the domain we compute two decompositions, i.e. one for each interface, by using the mean of the two cell averages sharing the interface. For more details we refer the reader to [25].

We discuss two different problems: the Sod shock tube problem and a shock-entropy wave interaction problem. We have obtained similar results for the Woodward-Colella blast wave problem [36] and the Lax shock tube problem.

We consider the Sod shock tube problem, a one-dimensional Riemann problem with an analytical solution. Given $\Omega_x = [0,1]$, the initial conditions are

$$\rho(x,0) = \begin{cases} -1 & x < 1/2 \\ 0.125 & x \geq 1/2 \end{cases}, \ \rho u(x,0) = 0, \ E(x,0) = \frac{1}{\gamma - 1} \begin{cases} 1 & x < 1/2 \\ 0.1 & x \geq 1/2 \end{cases}. \tag{5.9}$$

The boundary conditions follow directly from the initial conditions as the flow never reaches the boundary of the computational domain. This set up corresponds to a tube filled with two gases, separated by a membrane with zero initial velocity everywhere. As soon as the membrane is removed at $t = 0$, we observe a flow of gas in the direction of lower pressure, in this case the right. The flow of gas involves three distinct types of waves that separate the computational domain into four regions in which the state variables remain constant: a rarefaction wave, a contact wave, and a shock wave.

Figure 9 shows the density profile at $t = 0.2$ when the problem is solved using the type I RBF-WENO scheme with MQ interpolants of order $k = 1, 2, 3$ for several refinements. The left column shows the density profile on the whole computational domain Ω_x, while the middle and the right column show a detail of the solution, namely the rarefaction wave and the contact wave, respectively. The exact solution is shown with a solid dark line. As in the previous Burgers' example, using the shape parameters defined in Table 2 may lead to spurious oscillations as h is decreased. Figure 10 shows the solutions of the Sod problem obtained using the RBF-WENO method with multiquadric interpolants of order $k = 1$ on a uniform grid of size $h = 1/256$ for different values of the shape parameter. The graphs also show the solution obtained using the classic polynomial WENO method of corresponding order on the same uniform grid. For small values of ε, the solution obtained with the RBF-WENO method is oscillating, while the oscillations decrease in amplitude as the shape parameter is increased (for $\varepsilon = 1, 2, 3, 4$). However, in regions where the solution is smooth the solution obtained with the RBF-WENO method is less dissipative than the solution obtained with the polynomial WENO method if the shape parameter is small. The shape parameters in Fig. 9 are chosen experimentally to guarantee a balance between good accuracy and non-oscillatory behavior.

Let us now consider a second and more complex problem for Euler equations, the shock-entropy problem, for which the initial conditions are prescribed by

$$(\rho, \rho u, E)(x,0) = \begin{cases} (3.857143, 2.629369, 10.33333) & x < -4 \\ (1 + 0.2\sin(\pi x), 0, 1) & x \geq -4 \end{cases}, \quad x \in [-5, 5] \tag{5.10}$$

and the boundary conditions are again given by the initial conditions. In contrast to the Sod's problem, there is no analytical solution and we use as reference a solution computed with the classic WENO method using a very fine grid. Figure 11 shows the solution obtained with a type I RBF-WENO method with MQ interpolants at time $t = 1.8$ for $k = 1, 2, 3$ and various refinements. The left column shows the solution on the entire domain, while the right column shows a detail where the

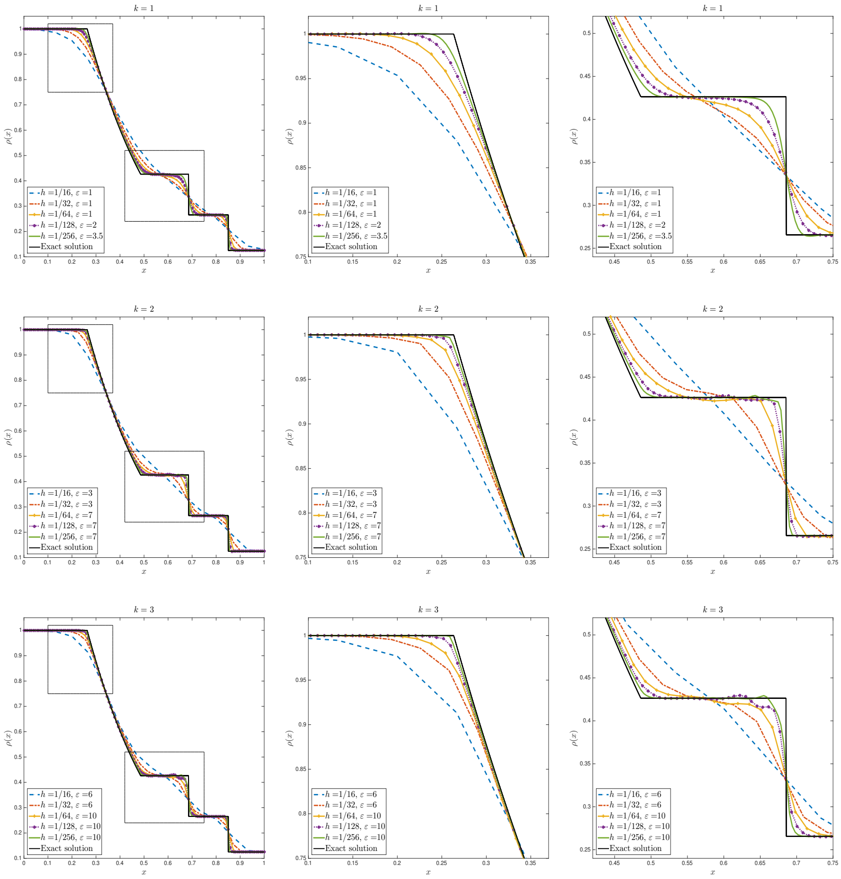

Fig. 9 Density profile for the Sod shock tube problem at $t = 0.2$ for the type I RBF-WENO method using MQ interpolants of order $k = 1$ (top row), $k = 2$ (middle row) and $k = 3$ (bottom row). For each row, the solution is shown on its entire domain (left column) and two details of the solution are highlighted: the rarefaction wave (central column) and the contact wave (right column)

solution varies rapidly. We observe convergence both for decreasing cell size and increasing order of the RBFs.

6 Adaptive meshes

A key advantage of RBF-WENO methods lies in the potential for using non-uniform grids in one dimension and unstructured grids in multiple dimensions. This feature is fundamental to define a mesh adaptation strategy that balances computational cost and quality of the solution by using different resolutions in different regions of the domain. To distinguish regions in which the solution is

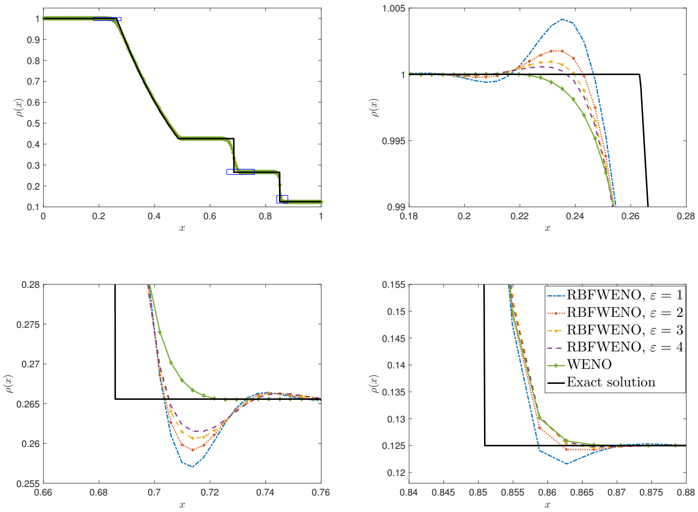

Fig. 10 Density profile for the Sod shock tube problem at $t = 0.2$ for the type I RBF-WENO method using MQ interpolants of order $k = 1$ on a uniform grid of size $h = 1/256$ for increasing values of ε. The solution is shown on its entire domain (top left) and three details of the solution are highlighted: the rarefaction wave (top right), the contact wave (bottom left) and the shock wave (bottom right). The classic WENO solution of the same order is also shown.

smooth from regions in which the solution presents sharp gradients, a suitable error indicator is needed. We use the smoothness indicator (4.7) to decided whether to increase or reduce the number of cells in a certain region.

6.1 Mesh adaptation strategy

Given a mesh $\mathcal{T}(t^n)$ at time t^n, the idea is to use a new adapted mesh $\mathcal{T}(t^{n+1})$ at time t^{n+1}. For each cell $C \in \mathcal{T}(t^n)$, we consider the corresponding type I weights ω_C^r, with $r = 1, \ldots, n$. Recall that type I weights only depend on the definition of the two constants ϵ, ρ and the smoothness indicator (4.7), as the linear weights d_r are all set to 1. In general, if these weights assume similar values, i.e. $\omega_r \approx 1/n \ \forall r = 1, \ldots, n$, one concludes that the solution is smooth. Conversely, having one or more among the n weights close to zero indicates that the solution in that region contains discontinuities or sharp gradients. In this case the WENO method emulates the ENO idea of discarding (by assigning small weights) stencils that contain the discontinuity.

By defining two thresholds $\theta_{\min}$ and $\theta_{\max}$, each cell in the domain is associated with one of the following three mutually exclusive classes: to be refined, to be coarsened, or to be left unchanged. Let ω_C^* be the maximum among the n weights associated to cell $C \in \mathcal{T}(t^n)$, i.e.

$$\omega_C^* = \max_{r=1,\ldots,n} \omega_C^r. \tag{6.1}$$

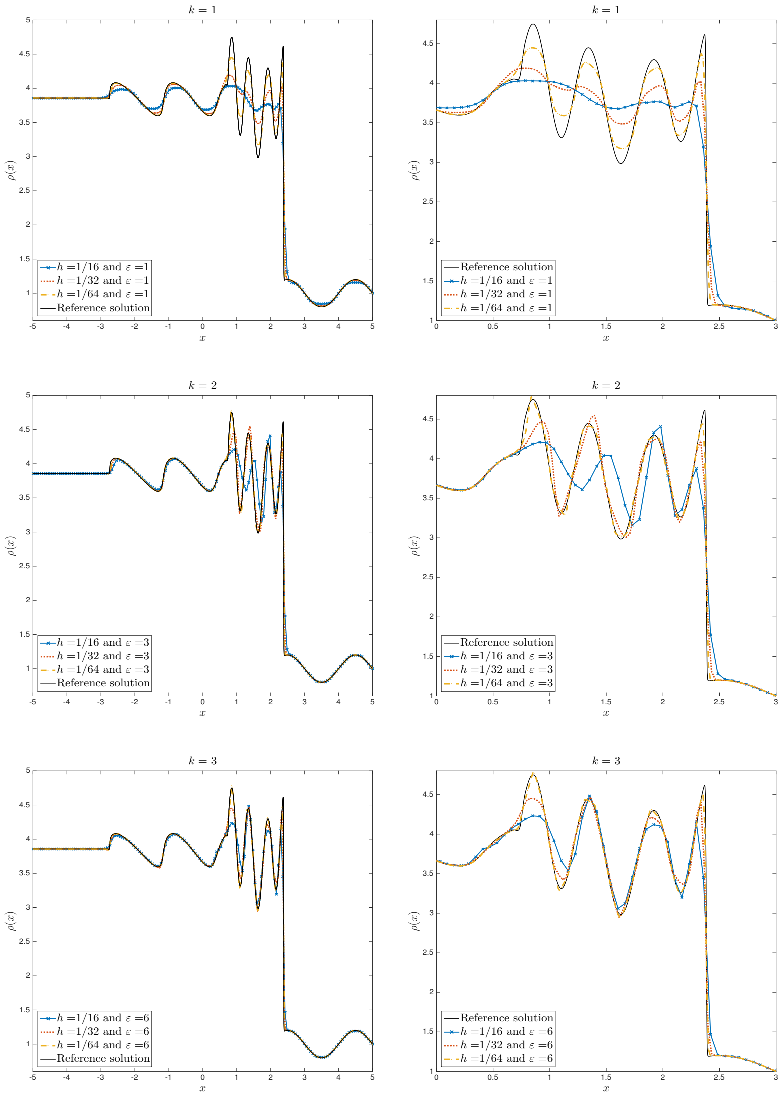

Fig. 11 Density profile for the shock entropy problem at $t=1.8$ for the type I RBF-WENO method using MQ interpolants of order $k=1$ (top row), $k=2$ (middle row) and $k=3$ (bottom row). For each row, the solution is shown on its entire domain (left column) and a detail of the solution is highlighted (right column)

If $\omega_C^* \geq \theta_{\max}$ then cell C is chosen for refinement, whereas cell C is chosen for coarsening if $\omega_C^* \leq \theta_{\min}$. If $\theta_{\min} < \omega_C^* < \theta_{\max}$, the cell is neither refined nor coarsened. In general $0 \leq \theta_{\min} < \theta_{\max} \leq 1$, and in our numerical examples we choose $\theta_{\min} = {}^1\!/\!_n + 10^{-4}$ and $\theta_{\max} = 0.88$. Note that instead of fixing these two thresholds one could store (6.1) for every $C \in \mathcal{T}(t^n)$, normalize these value over the entire domain and choose to refine a certain portion of cells and coarsen another. This strategy of relative thresholds is recommended when the solution is continuous, but one still wants to refine where it varies more rapidly. While we choose to use the maximum operator to distinguish the corresponding class of each cell, other choices are possible, e.g., using the variance of the weights. In this case, the specific values of the two thresholds need to be updated.

This strategy does not require the number of cells in the mesh to be fixed. However, it is fundamental that the cell sizes do not become neither too small nor too big, for the sake of computational cost and non-trivial recovery, respectively. If the size of a cell, marked for refinement, is smaller than a fixed threshold $d_{\min}$, then this cell will remain unchanged. Conversely, cells marked for coarsening, but with a dimension that is bigger than an upper threshold $d_{\max}$ are left unchanged. Finally, cells marked for neither refinement nor coarsening but with a size that is either smaller than $d_{\min}$ or bigger than $d_{\max}$ are marked for coarsening or refinement, respectively. Given h as the size of the initial uniform grid, we choose $d_{\min} = 0.145h$ and $d_{\max} = 3h$.

Finally, to avoid boundary issues, we mark boundary cells $C \in \partial\mathcal{T}$ for neither refinement nor coarsening.

Once each cell in the domain has been characterized, we proceed by removing cells marked for coarsening and split in two those marked for refinement. In the latter case, the cell centers become the common interface of the two new cells and the new centers are placed in the middle between this interface and the initial interfaces of the cell, i.e. given $C = [x_{\ell-1/2}, x_{\ell+1/2}]$ as the cell to be refined, we obtain $C_1 = [x_{\ell-1/2}, x_\ell]$ and $C_1 = [x_\ell, x_{\ell-1/2}]$.

6.2 Conservative mesh adaptation

Once the new grid $\mathcal{T}(t^{n+1})$ has been computed, care must be taken when assigning cell averages from the original grid to the adapted one. As we are dealing with conservative methods, we must require that any interpolation or averaging strategy maintains conservation.

For each cell $C \in \mathcal{T}(t^{n+1})$, we detect the corresponding intersecting cells in the original mesh $\mathcal{T}(t^n)$, together with their area. This step in 1D is rather simple, while in multiple dimensions one may employ the chasing algorithm [28]. Imagine that cell $C \in \mathcal{T}(t^{n+1})$ overlaps with δ cells $C_j \in \mathcal{T}(t^n)$, for $j = 1, \ldots, \delta$, such that $C = \bigcup_{j=1}^{\delta}(C \cap C_j)$ and let h_C^j be the area of C_j that contributes to cell C so that $|C| = \sum_{j=1}^{\delta} h_C^j$. Note that $h_C^j \leq |C_j|$ for all $C_j \in \mathcal{T}(t^n)$ and $h_C^j \leq |C|$. Then, the total mass of $C \in \mathcal{T}(t^{n+1})$ is given by $M_C = \sum_{j=1}^{\delta} M_{C\cap C_j}$, where each contribution is given by

$$M_{C\cap C_j} = h_C^j \bar{u}_{C_j}(t^n, x).$$

Here $\bar{u}_{C_j}(t^n, x)$ is the cell average corresponding to cell $C_j \in \mathcal{T}(t^n)$. Finally, the cell average of cell $C \in \mathcal{T}(t^{n+1})$ is given by

$$\bar{u}_C = \frac{M_C}{|C|} \quad \forall C \in \mathcal{T}(t^{n+1}).$$

6.3 Numerical results on adapted grids

Let us consider Burgers equation (5.4) with a continuous initial condition (5.3). The solution over an adapted grid at times $t=0, 0.15, 0.2, 0.25$ is shown in Fig. 12. We observe that the points cluster near the discontinuity which develops at $x=1/2$. Moreover, the left graph of Fig. 14 displays the dimension of the cells and the positions for the corresponding times. Analogously, Fig. 13 shows the evolution of Burgers equation with discontinuous initial condition (5.5) for $t=0, 0.025, 0.05, 0.1$. Qualitatively, we observe that the adaptation method captures the moving shock well. This result is confirmed by the right graph of Fig. 14, where the cell dimensions are presented and one observes that the smaller cells are confined in the neighborhood of the shock as time increases, i.e., $x = at$ where $a = 3/2$ is the speed.

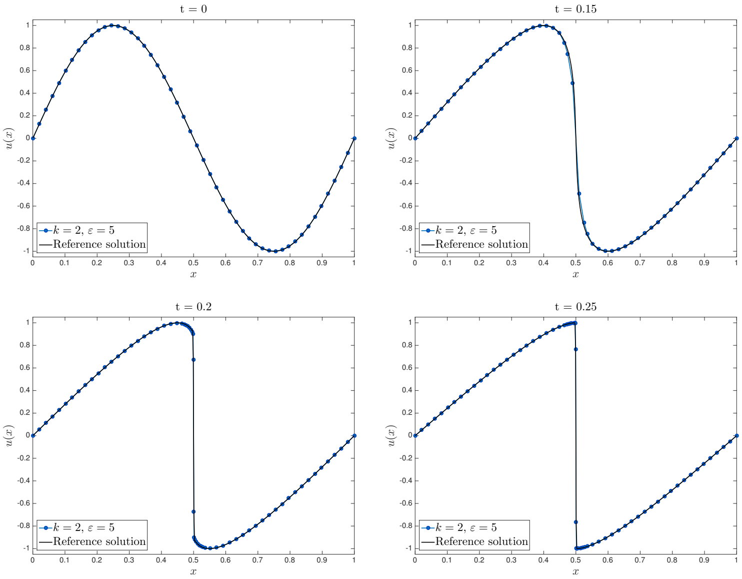

Fig. 12 Solution of Burgers equation with continuous initial condition at different times using the adaptive strategy. The type I RBF-WENO method is of order $k=2$ with $\varepsilon = 5$. The initial grid has size $h=0.02$

Finally, Fig. 15 shows the solution of Sod problem (5.10) at $t=0.2$ obtained using an adaptive type I RBF-WENO method with second order multiquadrics interpolants with $\varepsilon=5$. The left graph compares the adapted solution, using an initial uniform grid of size $h=2^{-4}$, with solutions on uniform grids of different refinements, namely $h = 2^{-4}, 2^{-5}, 2^{-6}$, as well as the exact solution. The right graph shows the corresponding cell sizes. We observe that the central region of the domain Ω_x is characterized by a fine grid. Overall we see a substantial improvement in accuracy when using the adaptive method.

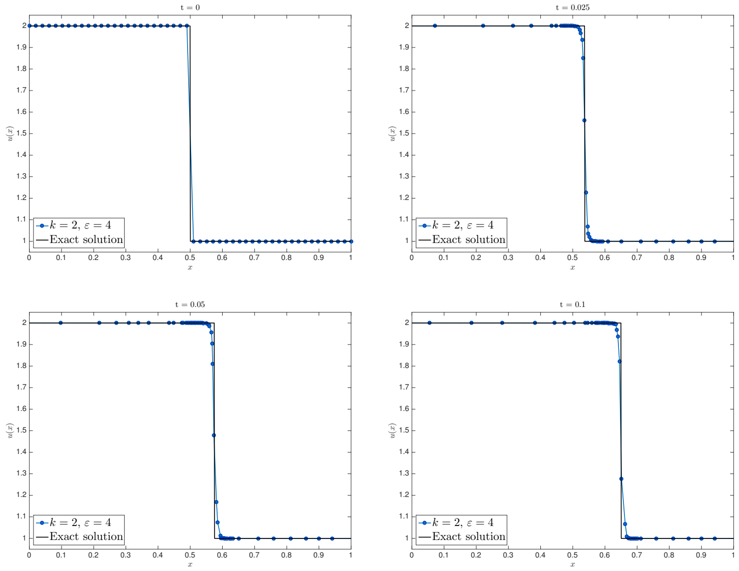

Fig. 13 Solution of Burgers equation with a discontinuous initial condition at different times using the adaptive strategy. The type I RBF-WENO method is of order $k=2$ with $\varepsilon=4$. The initial grid has size $h=0.02$

7 Conclusions

We have developed a RBF-WENO method that uses high-order RBF interpolants in the reconstruction step. Exploiting the mesh-less property of RBFs, these schemes are very flexible as the order of the RBF is not required to match the stencil size. This feature is particularly important for multidimensional problems and when the mesh is unstructured. Particular attention is given to the definition of the weights, both in terms of the linear weights and the smoothness indicator. Moreover, the latter plays an important role in the implementation of a conservative mesh adaptation strategy.

We test two types of RBFs for the reconstruction step and show that using multiquadrics yields more accurate results than when using polyharmonic splines. The multiquadrics belong to the so-called class of infinitely smooth RBFs and the shape parameter is a scaling parameter peculiar to this class of functions. Choosing this parameter is a delicate task as too small values lead to numerically unstable solutions while too large values result in inaccurate solutions. The optimal choice of the shape parameter depends upon a set of conditions, but in particular upon the order of the RBF and the refinement of the mesh. A simple selection strategy is described and supported by several numerical results.

The numerical results demonstrate good performance of the method both for scalar conservation laws and system of equations. Examples include Burgers equations, the Buckley-Leverett equation, and the Euler equations; we have obtained

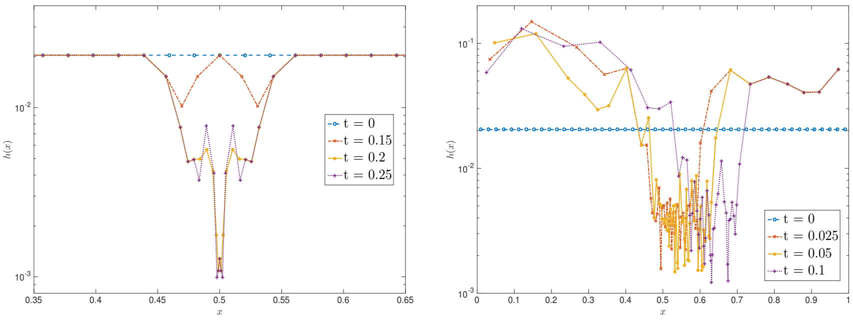

Fig. 14 Cell dimensions of the solution of Burgers equation at different time steps using adaptive grids with continuous initial condition (left) and discontinuous initial conditions (right). The type I RBF-WENO method is of order $k = 2$ with $\varepsilon = 5$ and $\varepsilon = 4$, respectively

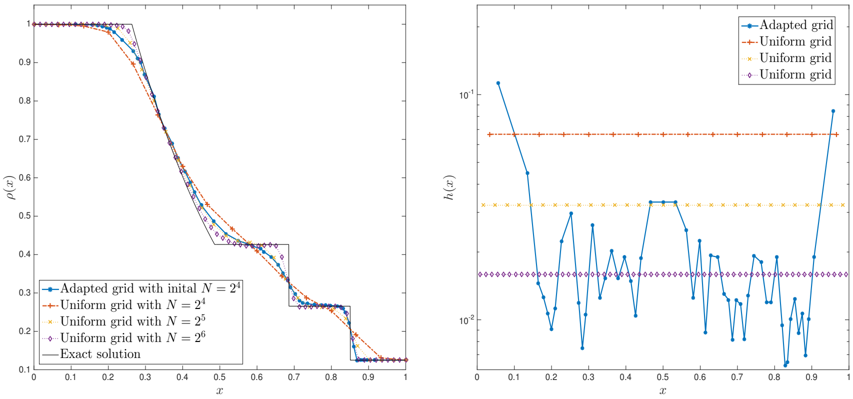

Fig. 15 Comparison of the solution of the Sod problem obtained using and adaptive mesh and uniform meshes of different refinements (left). Corresponding cell dimensions on a semilogarithmic axis (right). The type I RBF-WENO method is of order $k = 2$ with $\varepsilon = 5$

similar results for several other problems. Good results are also obtained on adaptive grids.

The possibility of extending the proposed adapted method to multiple dimensions on unstructured meshes is currently under investigation.

References

1. Aboiyar, T., Georgoulis, E.H., Iske, A.: 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)
2. Aboiyar, T., Georgoulis, E.H., Iske, A.: Adaptive ADER methods using kernel-based polyharmonic spline WENO reconstruction. SIAM Journal on Scientific Computing **32**(6), 3251–3277 (2010)
3. Wright, G. B. and Fornberg, B.: Stable computations with flat radial basis functions using vector-valued rational approximations. Journal of Computational Physics **331**, 137–156 (2017)
4. Buhmann, M.D.: Radial basis functions. Acta Numerica 2000 **9**, 1–38 (2000)
5. Carlson, R.E., Foley, T.A.: The parameter R^2 in multiquadric interpolation. Computers & Mathematics with Applications **21**(9), 29–42 (1991)
6. Courant, R., Friedrichs, K., Lewy, H.: On the partial difference equations of mathematical physics. IBM journal of Research and Development **11**(2), 215–234 (1967)
7. Driscoll, T.A., Fornberg, B.: Interpolation in the limit of increasingly flat radial basis functions. Computers & Mathematics with Applications **43**(3), 413–422 (2002)
8. Fasshauer, G.E.: Meshfree approximation methods with MATLAB, vol. 6. World Scientific (2007)
9. Fasshauer, G.E., Zhang, J.G.: On choosing "optimal" shape parameters for RBF approximation. Numerical Algorithms **45**(1-4), 345–368 (2007)
10. Fjordholm, U.S., Ray, D.: A sign preserving WENO reconstruction method. Journal of Scientific Computing pp. 1–22 (2015)
11. Flyer, N., Fornberg, B., Bayona, V., Barnett, G.A.: On the role of polynomials in RBF-FD approximations: I. interpolation and accuracy. Journal of Computational Physics **321**, 21–38 (2016)
12. Fornberg, B., Larsson, E., Flyer, N.: Stable computations with gaussian radial basis functions. SIAM Journal on Scientific Computing **33**(2), 869–892 (2011)
13. Fornberg, B., Piret, C.: A stable algorithm for flat radial basis functions on a sphere. SIAM Journal on Scientific Computing **30**(1), 60–80 (2007)
14. Fornberg, B., Wright, G.: Stable computation of multiquadric interpolants for all values of the shape parameter. Computers & Mathematics with Applications **48**(5), 853–867 (2004)
15. Franke, R.: Scattered data interpolation: Tests of some methods. Mathematics of computation **38**(157), 181–200 (1982)
16. Gottlieb, S., Ketcheson, D.I., Shu, C.W.: High order strong stability preserving time discretizations. Journal of Scientific Computing **38**(3), 251–289 (2009)
17. Gottlieb, S., Shu, C.W., Tadmor, E.: Strong stability-preserving high-order time discretization methods. SIAM review **43**(1), 89–112 (2001)
18. Guo, J., Jung, J.H.: A RBF-WENO finite volume method for hyperbolic conservation laws with the monotone polynomial interpolation method. Applied Numerical Mathematics **112**, 27–50 (2017)
19. Guo, J., Jung, J.H.: Radial basis function ENO and WENO finite difference methods based on the optimization of shape parameters. Journal of Scientific Computing **70**(2), 551–575 (2017)
20. Hardy, R.L.: Multiquadric equations of topography and other irregular surfaces. Journal of geophysical research **76**(8), 1905–1915 (1971)
21. Harten, A., Engquist, B., Osher, S., Chakravarthy, S.R.: Uniformly high order accurate essentially non-oscillatory schemes, III. Journal of Computational Physics **71**(2), 231–303 (1987)
22. Iske, A.: Multiresolution Methods in Scattered Data Modelling. SpringerVerlag (2004)
23. Iske, A., Sonar, T.: On the structure of function spaces in optimal recovery of point functionals for ENO-schemes by radial basis functions. Numerische Mathematik **74**(2), 177–201 (1996)
24. Jiang, G.S., Shu, C.W.: Efficient implementation of weighted ENO schemes. Journal of Computational Physics **126**(1), 202 – 228 (1996)
25. LeVeque, R.J.: Numerical Methods for Conservation Laws. Lectures in Mathematics. ETH Zürich. Birkhäuser Basel (1992)
26. Liu, X.D., Osher, S., Chan, T.: Weighted essentially non-oscillatory schemes. Journal of computational physics **115**(1), 200–212 (1994)

27. Liu, Y., Zhang, Y.T.: A robust reconstruction for unstructured WENO schemes. Journal of Scientific Computing **54**(2-3), 603–621 (2013)
28. O'Rourke, J.: Computational geometry in C. Cambridge university press (1998)
29. Qiu, J.M., Shu, C.W.: Convergence of high order finite volume weighted essentially nonoscillatory scheme and discontinuous galerkin method for nonconvex conservation laws. SIAM Journal on Scientific Computing **31**(1), 584–607 (2008)
30. Rippa, S.: 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. Schaback, R.: Error estimates and condition numbers for radial basis function interpolation. Advances in Computational Mathematics **3**(3), 251–264 (1995)
32. Schaback, R., Wendland, H.: Kernel techniques: from machine learning to meshless methods. Acta Numerica **15**, 543–639 (2006)
33. Shu, C.W.: Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws. In: A. Quarteroni (ed.) Advanced Numerical Approximation of Nonlinear Hyperbolic Equations, *Lecture Notes in Mathematics*, vol. 1697, pp. 325–432. Springer Berlin Heidelberg (1998)
34. Shu, C.W., Osher, S.: Efficient implementation of essentially non-oscillatory shock-capturing schemes. Journal of Computational Physics **77**(2), 439–471 (1988)
35. Wendland, H.: Scattered Data Approximation. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press (2004)
36. Woodward, P., Colella, P.: The numerical simulation of two-dimensional fluid flow with strong shocks. Journal of computational physics **54**(1), 115–173 (1984)
37. Zhang, Y.T., Shu, C.W.: Third order WENO scheme on three dimensional tetrahedral meshes. Communications in Computational Physics **5**(2-4), 836–848 (2009)

black

Appendix A Derivation of the expansion coefficients of the RBF interpolant s of order $k=1,2$

In the following we show that, for uniform one-dimensional meshes, the c_C coefficients in (4.3) for a stencil of size $n=k+1$ for $k=1$ or $k=2$ agree with the coefficients used in the classic polynomial WENO reconstructions for the same stencils sizes (see [33]) up to a term that depends on ε and Δx. We conjecture that these results extend to $k>2$. Clearly, if the expansion coefficients are the same for the RBF interpolation and the polynomial interpolation, so are the linear weights $\boldsymbol{d}$ in the definition of the weights ω, since (4.5) is the same in the two cases.

Given a cell $C_i=[x_{i-\frac{1}{2}},x_{i+\frac{1}{2}}]$ consider the three stencils $\mathcal{S}_1^{k=1}=\{C_{i-1},C_i\}$, $\mathcal{S}_2^{k=1}=\{C_i,C_{i+1}\}$ and $\mathcal{S}_2^{k=2}=\{C_{i-1},C_i,C_{i+1}\}$ as illustrated in Fig. 16. For each of these stencils we construct the relative interpolant $s(x)$ in terms of the coefficients $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ as in (3.7) and then evaluate s at the left and the right interface of cell C_i: $x_L=x_{i-\frac{1}{2}}$ and $x_R=x_{i+\frac{1}{2}}$, respectively. To derive the c_C coefficients for all the cells in the stencil, we express $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ in terms of the cell averages of the solution corresponding to the underlying stencil by solving the linear system (3.9) and use (4.3) for direct comparison of the terms. Similar calculations, albeit with a different goal, can be found in [7], where the authors show that in 1D, under simple assumptions on the basis function, the interpolants converge to the Lagrange interpolating polynomial as $\varepsilon\to 0$.

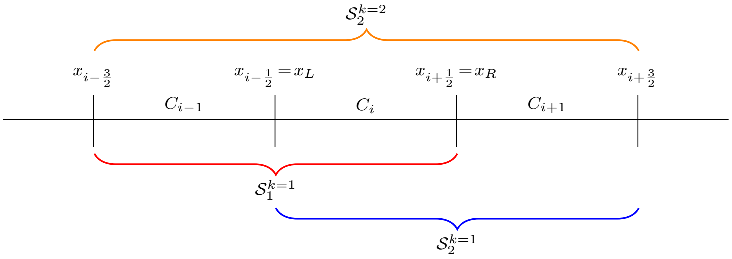

Fig. 16 Stencils of size 2 and size 3 associated with cell C_i

Stencil $\mathcal{S}_1^{k=1}=\{C_{i-1},C_i\}$. Let us first consider the left stencil of size $n=2$ in Fig. 16 and notice that the problem will be very similar for the right stencil. Given the general expression of the interpolants (3.7), they are constructed over these two cells as

$$\begin{aligned}
s_1^{k=1}(x) =& \sum_{j=1}^{n=2} \alpha_{1,j}\, \lambda^{\xi}_{C_{i-2+j}} \phi_{k=1}(\|x-\xi\|) + \sum_{\ell=1}^{q=1} \beta_{1,\ell} x^{\ell-1} \\
=&\alpha_{11} \frac{1}{\Delta x} \int_{x_{i-\frac{3}{2}}}^{x_{i-\frac{1}{2}}} \phi_1(\|x-\xi\|) d\xi + \alpha_{12} \frac{1}{\Delta x} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \phi_1(\|x-\xi\|) d\xi + \beta_{11} \\
=&\alpha_{11} f_1(x, x_{i-\frac{3}{2}}, x_{i-\frac{1}{2}}) + \alpha_{12} f_1(x, x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}) + \beta_{11},
\end{aligned} \tag{A.1}$$

where $\boldsymbol{\alpha}_1 = [\alpha_{11}, \alpha_{12}]^T$ and $\boldsymbol{\beta}_1 = [\beta_{11}]$ are the unknown coefficients to be determined. The first subscripts indicate the "first" stencil, i.e. the left one. Moreover,

$$f_k(x, a, b) = \frac{1}{\Delta x} \int_a^b \phi_k(\|x-\xi\|) d\xi. \tag{A.2}$$

Given the general expression of the linear system (3.9), α_{11}, α_{12} and β_{11} are obtained by solving

$$\begin{bmatrix} A_{11} & A_{12} & P_{11} \\ A_{21} & A_{22} & P_{21} \\ P_{11} & P_{21} & 0 \end{bmatrix} \begin{bmatrix} \alpha_{11} \\ \alpha_{12} \\ \beta_{11} \end{bmatrix} = \begin{bmatrix} \overline{v}_{i-1} \\ \overline{v}_i \\ 0 \end{bmatrix},$$

where the rectangular matrix $\boldsymbol{P} = [P_{11}, P_{21}]^T$ is a vector composed by ones, i.e. $P_{11} = P_{21} = 1$ (see the definition of general matrix P in (3.9)). Moreover, matrix $\boldsymbol{A}$ is symmetric by construction, i.e., $A_2 := A_{12} = A_{21}$. Since the grid is uniform the elements on the diagonal are all the same, i.e., $A_1 := A_{11} = A_{22}$. Matrix $\boldsymbol{A}$ is therefore a symmetric Toeplitz matrix of size 2. Using these simplifications, the system becomes

$$\begin{bmatrix} A_1 & A_2 & 1 \\ A_2 & A_1 & 1 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} \alpha_{11} \\ \alpha_{12} \\ \beta_{11} \end{bmatrix} = \begin{bmatrix} \overline{v}_{i-1} \\ \overline{v}_i \\ 0 \end{bmatrix}. \tag{A.3}$$

For the following steps, we choose ϕ to be a multiquadrics function, i.e., for $k=1$

$$\phi_1(\varepsilon, \|x-\xi\|) \overset{k=1\to\nu=1/2}{=} \sqrt{1+\varepsilon^2(x-\xi)^2}. \tag{A.4}$$

Therefore, the interpolation matrix in (A.3) is composed of the following elements

$$\begin{aligned}
A_1 =& \frac{1}{\Delta x^2} \int_{x_{i-\frac{3}{2}}}^{x_{i-\frac{1}{2}}} \int_{x_{i-\frac{3}{2}}}^{x_{i-\frac{1}{2}}} \phi_1(\|x-\xi\|) d\xi dx = 2m(0) - 2m(\Delta x), \\
A_2 =& \frac{1}{\Delta x^2} \int_{x_{i-\frac{3}{2}}}^{x_{i-\frac{1}{2}}} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \phi_1(\|x-\xi\|) d\xi dx = -m(0) + 2m(\Delta x) - m(2\Delta x),
\end{aligned} \tag{A.5}$$

where

$$m(y) = -\frac{1}{6\varepsilon^2 \Delta x^2} \left[\sqrt{1+\varepsilon^2 y^2}(\varepsilon^2 y^2 - 2) + 3\varepsilon y \sinh^{-1}(\varepsilon y) \right]. \tag{A.6}$$

Our goal is to derive the coefficients c_{1j} with $j = 1, 2$, such that

$$s_1^{k=1}(x) = c_{11}(x) \overline{v}_{i-1} + c_{12}(x) \overline{v}_i. \tag{A.7}$$

To do so, we need to compare (A.7) and (A.1) by expressing the coefficients α_{11}, α_{12} and β_{11} as a function of the cell averages. Inverting the 3×3 matrix in (A.3), we obtain

$$\begin{bmatrix} A_1 & A_2 & 1 \\ A_2 & A_1 & 1 \\ 1 & 1 & 0 \end{bmatrix}^{-1} = \frac{1}{2} \begin{bmatrix} (A_1 - A_2)^{-1} & (A_2 - A_1)^{-1} & 1 \\ (A_2 - A_1)^{-1} & (A_1 - A_2)^{-1} & 1 \\ 1 & 1 & -(A_1 + A_2) \end{bmatrix}.$$

Note that only the first 2 columns of the inverse matrix are needed to compute the coefficients $\boldsymbol{\alpha}_1$ and $\boldsymbol{\beta}_1$ because only the first 2 elements of the right hand side are nonzero by definition. Therefore we obtain

$$\begin{bmatrix} \alpha_{11} \\ \alpha_{12} \\ \beta_{11} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} (A_1 - A_2)^{-1} & (A_2 - A_1)^{-1} \\ (A_2 - A_1)^{-1} & (A_1 - A_2)^{-1} \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \overline{v}_{i-1} \\ \overline{v}_i \end{bmatrix},$$

or

$$\begin{cases} \alpha_{11} = \frac{1}{2}(A_2 - A_1)^{-1}(-\overline{v}_{i-1} + \overline{v}_i) \\ \alpha_{12} = -\alpha_{11} \\ \beta_{11} = \frac{1}{2}(\overline{v}_{i-1} + \overline{v}_i) \end{cases}.$$

Inserting these coefficients into the last equality of (A.1), we obtain

$$\begin{aligned} s_1^{k=1}(x) = &\frac{1}{2}(A_2 - A_1)^{-1}(-\overline{v}_{i-1} + \overline{v}_i) \left[f_1(x, x_{i-\frac{3}{2}}, x_{i-\frac{1}{2}}) - f_1(x, x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}) \right] \\ &+ \frac{1}{2}(\overline{v}_{i-1} + \overline{v}_i). \end{aligned}$$

By factoring out the cell averages, we recover an explicit expression of the two coefficients in (A.7) on the form

$$\begin{cases} c_{11}(x) = -\frac{1}{2(A_2 - A_1)} \left[f_1(x, x_{i-\frac{3}{2}}, x_{i-\frac{1}{2}}) - f_1(x, x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}) \right] + \frac{1}{2} \\ c_{12}(x) = \frac{1}{2(A_2 - A_1)} \left[f_1(x, x_{i-\frac{3}{2}}, x_{i-\frac{1}{2}}) - f_1(x, x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}) \right] + \frac{1}{2} \end{cases} \tag{A.8}$$

Each of these two coefficients need to be evaluated at the left and the right interface of the central cell C_i, i.e., $x_L = x_{i-\frac{1}{2}}$ and $x_R = x_{i+\frac{1}{2}}$. Therefore, we need to evaluate $f_1(x_{i-\frac{1}{2}}, x_{i-\frac{3}{2}}, x_{i-\frac{1}{2}})$ and $f_1(x_{i-\frac{1}{2}}, x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}})$ for the left coefficients and $f_1(x_{i+\frac{1}{2}}, x_{i-\frac{3}{2}}, x_{i-\frac{1}{2}})$ and $f_1(x_{i+\frac{1}{2}}, x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}})$ for the right coefficients. If we again consider first order multiquadrics (A.4), then $f_1(x, a, b)$, defined in (A.2), becomes

$$\begin{aligned} f_1(x, a, b) &= \frac{1}{\Delta x} \int_a^b \sqrt{1 + \varepsilon^2 (x - \xi)^2} d\xi \\ &= \frac{1}{2\varepsilon \Delta x} \left[\varepsilon(\xi - x)\sqrt{1 + \varepsilon^2 (x - \xi)^2} - \sinh^{-1}(\varepsilon(x - \xi)) \right]_a^b. \end{aligned}$$

Using this expression for the left interface we obtain

$$f_1(x_{i-\frac{1}{2}}, x_{i-\frac{3}{2}}, x_{i-\frac{1}{2}}) = \frac{1}{2\varepsilon \Delta x} \left[\varepsilon \Delta x \sqrt{1 + \varepsilon^2 \Delta x^2} + \sinh^{-1}(\varepsilon \Delta x) \right],$$

and

$$f_1(x_{i-\frac{1}{2}}, x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}) = \frac{1}{2\varepsilon\Delta x}\left[\varepsilon\Delta x\sqrt{1+\varepsilon^2\Delta x^2} + \sinh^{-1}(\varepsilon\Delta x)\right]. \tag{A.9}$$

It is clear that

$$f_1(x_{i-\frac{1}{2}}, x_{i-\frac{3}{2}}, x_{i-\frac{1}{2}}) = f_1(x_{i-\frac{1}{2}}, x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}),$$

which, when inserted into (A.8), implies

$$c_{11}(x_L) = \frac{1}{2} \quad \text{and} \quad c_{12}(x_L) = \frac{1}{2}.$$

Proceeding in the same manner for the right interface, we recover

$$\begin{aligned} f_1(x_{i+\frac{1}{2}}, x_{i-\frac{3}{2}}, x_{i-\frac{1}{2}}) = \frac{1}{2\varepsilon\Delta x}\Big[&-\varepsilon\Delta x\sqrt{1+\varepsilon^2\Delta x^2} - \sinh^{-1}(\varepsilon\Delta x) + \\ &2\varepsilon\Delta x\sqrt{1+4\varepsilon^2\Delta x^2} + \sinh^{-1}(2\varepsilon\Delta x)\Big], \end{aligned}$$

and

$$f_1(x_{i+\frac{1}{2}}, x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}) = \frac{1}{2\varepsilon\Delta x}\left[\varepsilon\Delta x\sqrt{1+\varepsilon^2(\Delta x)^2} + \sinh^{-1}(\varepsilon\Delta x)\right]. \tag{A.10}$$

Since these two functions are no longer equivalent, this case is more laborious than the previous one. To compute the constants $c_{11}(x_R)$ and $c_{12}(x_R)$, we first consider the Taylor expansion of the difference of these two expressions

$$\begin{aligned} &f_1(x_{i+\frac{1}{2}}, x_{i-\frac{3}{2}}, x_{i-\frac{1}{2}}) - f_1(x_{i+\frac{1}{2}}, x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}) \\ &= \frac{1}{2\varepsilon\Delta x}\Big[-2\varepsilon\Delta x\sqrt{1+\varepsilon^2\Delta x^2} - 2\sinh^{-1}(\varepsilon\Delta x) \\ &\quad +2\varepsilon\Delta x\sqrt{1+4\varepsilon^2(\Delta x)^2} + \sinh^{-1}(2\varepsilon\Delta x)\Big] \\ &= (\varepsilon\Delta x)^2 - \frac{3}{4}(\varepsilon\Delta x)^4 + \frac{9}{8}(\varepsilon\Delta x)^6 + \mathcal{O}(\Delta x^8). \end{aligned} \tag{A.11}$$

We evaluate $2(A_2 - A_1)$ by the definitions in (A.5)

$$2(A_2 - A_1) = 2\left(-3m(0) + 4m(\Delta x) - m(2\Delta x)\right). \tag{A.12}$$

To compute the Taylor expansion of (A.12), we consider the Taylor expansion of each term using (A.6). Therefore,

$$\begin{aligned} m(0) &= -\frac{1}{6\varepsilon^2\Delta x^2}[-2] = \frac{1}{3}(\varepsilon\Delta x)^{-2} \\ m(\Delta x) &= -\frac{1}{6\varepsilon^2\Delta x^2}\left[\sqrt{1+\varepsilon^2\Delta x^2}(\varepsilon^2\Delta x^2 - 2) + 3\varepsilon\Delta x\sinh^{-1}(\varepsilon\Delta x)\right] \\ &= \frac{1}{3}(\varepsilon\Delta x)^{-2} - \frac{1}{2} - \frac{1}{24}(\varepsilon\Delta x)^2 - \frac{73}{2480}(\varepsilon\Delta x)^4 - \frac{1}{112}(\varepsilon\Delta x)^6 + \mathcal{O}(\Delta x^8) \\ m(2\Delta x) &= -\frac{1}{6\varepsilon^2\Delta x^2}\left[\sqrt{1+4\varepsilon^2\Delta x^2}(4\varepsilon^2\Delta x^2 - 2) + 6\varepsilon\Delta x\sinh^{-1}(2\varepsilon\Delta x)\right] \\ &= \frac{1}{3}(\varepsilon\Delta x)^{-2} - 2 - \frac{2}{3}(\varepsilon\Delta x)^2 + \frac{4}{15}(\varepsilon\Delta x)^4 - \frac{2}{7}(\varepsilon\Delta x)^6 + \mathcal{O}(\Delta x^8) \end{aligned}$$

Inserting these results into (A.12), we obtain

$$2(A_2 - A_1) = (\varepsilon\Delta x)^2 - \frac{1}{2}(\varepsilon\Delta x)^4 + \frac{65}{112}(\varepsilon\Delta x)^6 + \mathcal{O}(\Delta x^8) \tag{A.13}$$

Finally, combining the two expressions (A.11), we obtain

$$\begin{aligned} &\frac{f_1(x_{i+\frac{1}{2}}, x_{i-\frac{3}{2}}, x_{i-\frac{1}{2}}) - f_1(x_{i+\frac{1}{2}}, x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}})}{2(A_2 - A_1)} \\ &= \frac{(\varepsilon\Delta x)^2 - \frac{3}{4}(\varepsilon\Delta x)^4 + \frac{9}{8}(\varepsilon\Delta x)^6 + \mathcal{O}(\Delta x^8)}{(\varepsilon\Delta x)^2 - \frac{1}{2}(\varepsilon\Delta x)^4 + \frac{65}{112}(\varepsilon\Delta x)^6 + \mathcal{O}(\Delta x^8)} \\ &= 1 - \frac{(\varepsilon\Delta x)^2}{4} + \mathcal{O}(\varepsilon\Delta x)^4 \end{aligned} \tag{A.14}$$

which, when inserted into (A.8), implies

$$c_{11}(x_R) = -\frac{1}{2} + \mathcal{O}(\varepsilon\Delta x)^2 \quad \text{and} \quad c_{12}(x_R) = -\frac{3}{2} + \mathcal{O}(\varepsilon\Delta x)^2$$

Stencil $\mathcal{S}_2^{k=1} = \{C_i, C_{i+1}\}$ The coefficients for the second stencil in Fig. 16 are obtained by symmetry. For the left and right interface they are, respectively,

$$c_{21}(x_L) = \frac{3}{2} + \mathcal{O}(\varepsilon\Delta x)^2, \quad c_{22}(x_L) = -\frac{1}{2} + \mathcal{O}(\varepsilon\Delta x)^2 \quad \text{and} \quad c_{21}(x_R) = c_{22}(x_R) = \frac{1}{2}.$$

Stencil $\mathcal{S}_2^{k=2} = \{C_{i-1}, C_i, C_{i+1}\}$ Among the three stencils of size 3 associated to cell C_i, we start by considering the second one $\mathcal{S}_2$, i.e. the one drawn in Fig. 16, as it covers the union of the two stencils of size 2 for which we derived the constants in the previous paragraphs. Computing the coefficients of $\mathcal{S}_2^{k=2}$ allows the derivation of the linear weights $\boldsymbol{d}$ by solving the linear system (4.5).

Applying the previous steps to a stencil of size three leads to similar but more complex calculations. However, it can be shown that

$$c_{21}(x_L) = \frac{1}{3} + \mathcal{O}(\varepsilon\Delta x)^2, \quad c_{22}(x_L) = \frac{5}{6} + \mathcal{O}(\varepsilon\Delta x)^2, \quad c_{23}(x_L) = -\frac{1}{6} + \mathcal{O}(\varepsilon\Delta x)^2,$$

$$c_{21}(x_R) = -\frac{1}{6} + \mathcal{O}(\varepsilon\Delta x)^2, \quad c_{22}(x_R) = \frac{5}{6} + \mathcal{O}(\varepsilon\Delta x)^2, \quad c_{23}(x_R) = \frac{1}{3} + \mathcal{O}(\varepsilon\Delta x)^2.$$

These are again equivalent to the linear weights of the WENO scheme up to $\mathcal{O}(\varepsilon\Delta x)^2$. We conjecture that hold for any value of the order of the radial basis function.