A non-separable progressive multivariate WENO-$2r$ point value

Pep Mulet[a], Juan Ruiz-Álvarez[b], Chi-Wang Shu[c], Dionisio F. Yáñez[a]

[a] *Departamento de Matemáticas, Facultad de Matemáticas, Universidad de Valencia, Valencia, Spain*
[b] *Departamento de Matemática Aplicada y Estadística, Universidad Politécnica de Cartagena, Cartagena, Spain*
[c] *Division of Applied Mathematics, Brown University, Providence, Rhode Island, USA*

Abstract

The weighted essentially non-oscillatory technique using a stencil of $2r$ points (WENO-$2r$) is an interpolatory method that consists in obtaining a higher approximation order from the non-linear combination of interpolants of $r+1$ nodes. The result is an interpolant of order $2r$ at the smooth parts and order $r+1$ when an isolated discontinuity falls at any grid interval of the large stencil except at the central one. Recently, a new WENO method based on Aitken-Neville's algorithm has been designed for interpolation of equally spaced data at the mid-points and presents progressive order of accuracy close to discontinuities. This paper is devoted to constructing a general progressive WENO method for non-necessarily uniformly spaced data and several variables interpolating in any point of the central interval. Also, we provide explicit formulas for linear and non-linear weights and prove the order obtained. Finally, some numerical experiments are presented to check the theoretical results.

Keywords: WENO, non-linear reconstruction, order of accuracy, high accuracy interpolation

1. Introduction and review

In the last years, WENO methods have been developed and used in several applications, mainly to obtain numerical solutions of partial differential equations (PDEs) but also in other fields, such as image processing or computer-aided design (see e.g. [6, 14, 15]). The idea is to compute a non-linear combination of interpolations through polynomials of degree r, aiming to obtain the maximum possible order $2r$ when the data is free of discontinuities, and order $r+1$ at the non-smooth parts. In what follows, we briefly review the method. Let us denote as X a uniform partition of the interval $[a,b]$ in J subintervals:

$$X = \{x_i\}_{i=0}^{J}, \quad x_i = a + i \cdot h, \quad h = \frac{b-a}{J},$$

and consider the point-value discretization of a piecewise smooth function f at the nodes x_i,

$$f_i = f(x_i),\ i = 0, \dots, J, \quad f = \{f_i\}_{i=0}^{J}.$$

In this setting, the WENO method with $2r$ nodes interpolates at the mid-point of the interval (x_{i-1}, x_i), denoted by $x_{i-\frac{1}{2}}$, using the stencil $\mathcal{S}_0^{2r} = \{x_{i-r}, \cdots, x_{i+r-1}\}$. We construct this interpolation by means of the following convex combination:

$$\mathcal{I}^{2r-1}(x_{i-\frac{1}{2}}; f) = \sum_{k=0}^{r-1} \omega_k^r p_k^r(x_{i-\frac{1}{2}}), \tag{1}$$

where $\omega_k^r \geq 0$, $k = 0, \cdots, r-1$ are non-linear (data-dependent) positive weights such that $\sum_{k=0}^{r-1} \omega_k^r = 1$, and p_k^r are the Lagrange interpolants with nodes $\mathcal{S}_k^r = \{x_{i-r+k}, \cdots, x_{i+k}\}$.

[*] The second author has been supported through project 20928/PI/18 (Proyecto financiado por la Comunidad Autónoma de la Región de Murcia a través de la convocatoria de Ayudas a proyectos para el desarrollo de investigación científica y técnica por grupos competitivos, incluida en el Programa Regional de Fomento de la Investigación Científica y Técnica (Plan de Actuación 2018) de la Fundación Séneca-Agencia de Ciencia y Tecnología de la Región de Murcia) and by the national research project PID2019-108336GB-I00. The first and fourth authors have been supported through project CIAICO/2021/227 and by grant PID2020-117211GB-I00 funded by MCIN/AEI/10.13039/501100011033. The third author has been supported by NSF grant DMS-2309249.

Email addresses: pep.mulet@uv.es (Pep Mulet), juan.ruiz@upct.es (Juan Ruiz-Álvarez), chi-wang_shu@brown.edu (Chi-Wang Shu), dionisio.yanez@uv.es (Dionisio F. Yáñez)

The values of the weights ω_k^r are designed to obtain an order of accuracy $2r$ at $x_{i-\frac{1}{2}}$ when the function is smooth in the large stencil, as follows: There are optimal weights $C_k^r \geq 0$, with $k = 0, \dots, r-1$ that satisfy the following equality:

$$p_0^{2r-1}\left(x_{i-\frac{1}{2}}\right) = \sum_{k=0}^{r-1} C_k^r p_k^r \left(x_{i-\frac{1}{2}}\right).$$

A formula for these values is obtained in [6]:

$$C_k^r = \frac{1}{2^{2r-1}} \binom{2r}{2k+1}, \quad k = 0, \cdots, r-1. \tag{2}$$

As stated in [15], the weights ω_k^r satisfy

$$\omega_k^r = C_k^r + O(h^\kappa), \quad k = 0, \cdots, r-1, \tag{3}$$

at smooth zones, with $\kappa \geq r-1$, thus assuring that the interpolation in (1) attains order of accuracy $2r$

$$f(x_{i-\frac{1}{2}}) - \mathcal{I}^{2r-1}\left(x_{i-\frac{1}{2}}; f\right) = O(h^{2r}).$$

In [12, 15], (3) is achieved via the following expressions:

$$\omega_k^r = \frac{\alpha_k^r}{\sum_{j=0}^{r-1} \alpha_j^r}, \quad \text{where } \alpha_k^r = \frac{C_k^r}{(\epsilon + I_k^r)^t}, \quad k = 0, \cdots, r-1. \tag{4}$$

In the previous expression, the parameter t is an integer that assures maximum order of accuracy close to the discontinuities. The parameter $\epsilon > 0$ is introduced to avoid divisions by zero, and is usually forced to take the size of the smoothness indicators at smooth zones. In our numerical tests, we will set it to $\epsilon = h^2$. The values I_k^r are called *smoothness indicators* for $f(x)$ on each sub-stencil of r points.

Since it was first introduced in [15], several successive improvements have been proposed for WENO algorithms. They have been focused on enhancing their accuracy, efficiency, and robustness in approximating solutions to hyperbolic conservation laws, but also on extending their applicability to other contexts such as approximation and interpolation of data. Initially introduced by Liu, Osher, and Chan in 1994 [15], WENO schemes have gone through several advancements. The WENO schemes proposed by Jiang and Shu in 1996 [12], improved the original idea in [15], by proposing new smoothness indicators inspired by the measure of the total variation. These smoothness indicators were more capable of detecting discontinuities and allowed to extended the idea of WENO to higher orders of accuracy. However, the classical WENO scheme might experience a loss of accuracy when encountering critical points in the solution: for instance, a fifth-order WENO scheme may only attain third-order accuracy near smooth extrema [16]. In fact, in [16], the authors proposed the WENO-M scheme, that not only addressed the accuracy issue but also marked the first significant improvement in the solution quality near shocks and high gradients. However, the introduced mapping method proved to be computationally expensive. This study led to the publication of the article [9] where the authors proposed the WENO-Z scheme, that uses a new set of weights, derived from previously unused information within the classical WENO scheme: a higher-order global smoothness indicator constructed through a linear combination of the original smoothness indicators. This scheme achieved superior results with almost the same computational effort as the classical WENO method. After that, many variants of WENO schemes were proposed, starting from these two methodologies (see, e.g. [10, 13, 17]). In the context of data approximation, we proposed a first improvement of WENO algorithm in 2018 [2], where the algorithm attained a progressive order of accuracy close to the discontinuities using a recursive formulation of the WENO weights. Using WENO interpolation, maximum order of accuracy is obtained in smooth parts of the data but the accuracy is reduced to order $r+1$ when, at least, a discontinuity crosses a stencil. In [3], Amat, Ruiz, Shu and Yáñez present a new WENO method using the Aitken-Neville algorithm to obtain progressive order of approximation. This method is introduced for the point-value discretization in a uniform grid, and to approximate at the mid-points of the intervals. In [4], the algorithm is extended to calculate the derivative value of a function knowing its evaluation in a non-uniform grid and for any point of the considered interval.

In this paper, we extend this method to n dimensions and prove its theoretical properties. We start with dimension 1 in Section 2, following the ideas presented in [3, 4]. We provide a new recursive algorithm to obtain a non-linear scalar approximation for any point in the central interval in Section 2. Then, in Section 3, we review the multidimensional

Lagrange interpolation using tensor product, and the WENO version designed by Aràndiga et al. in [7]. Subsequently, we introduce, in Section 4, our general method for any dimension n starting with $n=2$, and give an explicit expression for the optimal weights. Afterwards, we generalize the method to any number of dimension n, and prove that the approximation attains the maximum order of accuracy when the nodes are in a region where f is smooth, and has an increasing order of accuracy depending on where the isolated discontinuity is. In Section 6, we present the smoothness indicators. The theoretical results are confirmed by the numerical examples, that are presented in Section 7. Finally, some conclusions are drawn in the last section.

2. A new univariate WENO-$2r$ algorithm with progressive order of accuracy close to discontinuities

Recently, Amat, Ruiz, Shu and Yáñez in [3] introduced a new centered WENO-$2r$ method, which consists in using all the points free of discontinuities to interpolate a value at the mid-point of an interval. In this section, we generalize this interpolation to non-uniform grids. Let $[a,b]\subset\mathbb{R}$ be an interval, we consider a non-uniform grid $a=x_0<x_1<\ldots<x_J=b$, with $h=\max_{l=1,\ldots,J}|x_l-x_{l-1}|$ and $i\in\mathbb{N}$ such that $0\leq i-r\leq i+r-1\leq J$. We consider a point, $x^*\in(x_{i-1},x_i)$; the largest stencil $\mathcal{S}_0^{2r}=\{x_{i-r},\cdots,x_{i+r-1}\}$, and construct the polynomial which interpolates at these nodes, that we denote by p_0^{2r-1}. Using the Aitken-Neville formula, we express this polynomial using the following expression

$$p_0^{2r-1}(x^*)=\sum_{j_0=0}^{1}C_{0,j_0}^{2r-2}(x^*)p_{j_0}^{2r-2}(x^*),$$

which involves the two interpolatory polynomials, $p_{j_0}^{2r-2}$, $j_0=0,1$ with respective stencils:

$$\mathcal{S}_0^{2r-1}=\{x_{i-r},\cdots,x_{i+r-2}\},\quad \mathcal{S}_1^{2r-1}=\{x_{i-r+1},\cdots,x_{i+r-1}\},$$

where $C_{0,0}^{2r-2}(x^*)$ and $C_{0,1}^{2r-2}(x^*)$ are the optimal weights. We repeat this process with each polynomial up to degree r. A representation of this process can be seen in Figure 1 (where we have removed the dependence on the value x^*), [3]. Thus, we obtain the general explicit expression:

$$\begin{aligned}p_0^{2r-1}(x^*)&=\sum_{j_0=0}^{1}C_{0,j_0}^{2r-2}(x^*)p_{j_0}^{2r-2}(x^*)\\&=\sum_{j_0=0}^{1}C_{0,j_0}^{2r-2}(x^*)\left(\sum_{j_1=j_0}^{j_0+1}C_{j_0,j_1}^{2r-3}(x^*)p_{j_1}^{2r-3}(x^*)\right)\\&=\sum_{j_0=0}^{1}C_{0,j_0}^{2r-2}(x^*)\left(\sum_{j_1=j_0}^{j_0+1}C_{j_0,j_1}^{2r-3}(x^*)\left(\sum_{j_2=j_1}^{j_1+1}C_{j_1,j_2}^{2r-4}(x^*)p_{j_2}^{2r-4}(x^*)\right)\right)\\&=\sum_{j_0=0}^{1}C_{0,j_0}^{2r-2}(x^*)\left(\sum_{j_1=j_0}^{j_0+1}C_{j_0,j_1}^{2r-3}(x^*)\left(\ldots\left(\sum_{j_{r-3}=j_{r-4}}^{j_{r-4}+1}C_{j_{r-4},j_{r-3}}^{r+1}(x^*)\left(\sum_{j_{r-2}=j_{r-3}}^{j_{r-3}+1}C_{j_{r-3},j_{r-2}}^{r}(x^*)p_{j_{r-2}}^{r}(x^*)\right)\right)\ldots\right)\right).\end{aligned}\tag{5}$$

We calculate these values through the following lemma (the version for uniform grids, and when the interpolation is at mid-points is proved in [3]).

Lemma 2.1. *Let $r\leq l\leq 2r-2$ and $0\leq j\leq(2r-2)-l$, $x^*\in(x_{i-1},x_i)$, then*

$$p_j^{l+1}(x^*)=C_{j,j}^{l}(x^*)p_j^{l}(x^*)+C_{j,j+1}^{l}(x^*)p_{j+1}^{l}(x^*),\tag{6}$$

where

$$C_{j,j}^{l}(x^*)=\frac{x^*-x_{i-r+j+l+1}}{x_{i-r+j}-x_{i-r+j+l+1}},\quad C_{j,j+1}^{l}(x^*)=1-C_{j,j}^{l}(x^*)=\frac{x_{i-r+j}-x^*}{x_{i-r+j}-x_{i-r+j+l+1}}.\tag{7}$$

Proof. The proof is direct by taking into account that the stencil for constructing p_j^{l+1} is $\{x_{i-r+j},\ldots,x_{i-r+j+l+1}\}$ and for p_j^l and p_{j+1}^l are $\{x_{i-r+j},\ldots,x_{i-r+j+l}\}$ and $\{x_{i-r+j+1},\ldots,x_{i-r+j+l+1}\}$ respectively. ■

As Corollary, we obtain the values showed in [3] for mid-point interpolation.

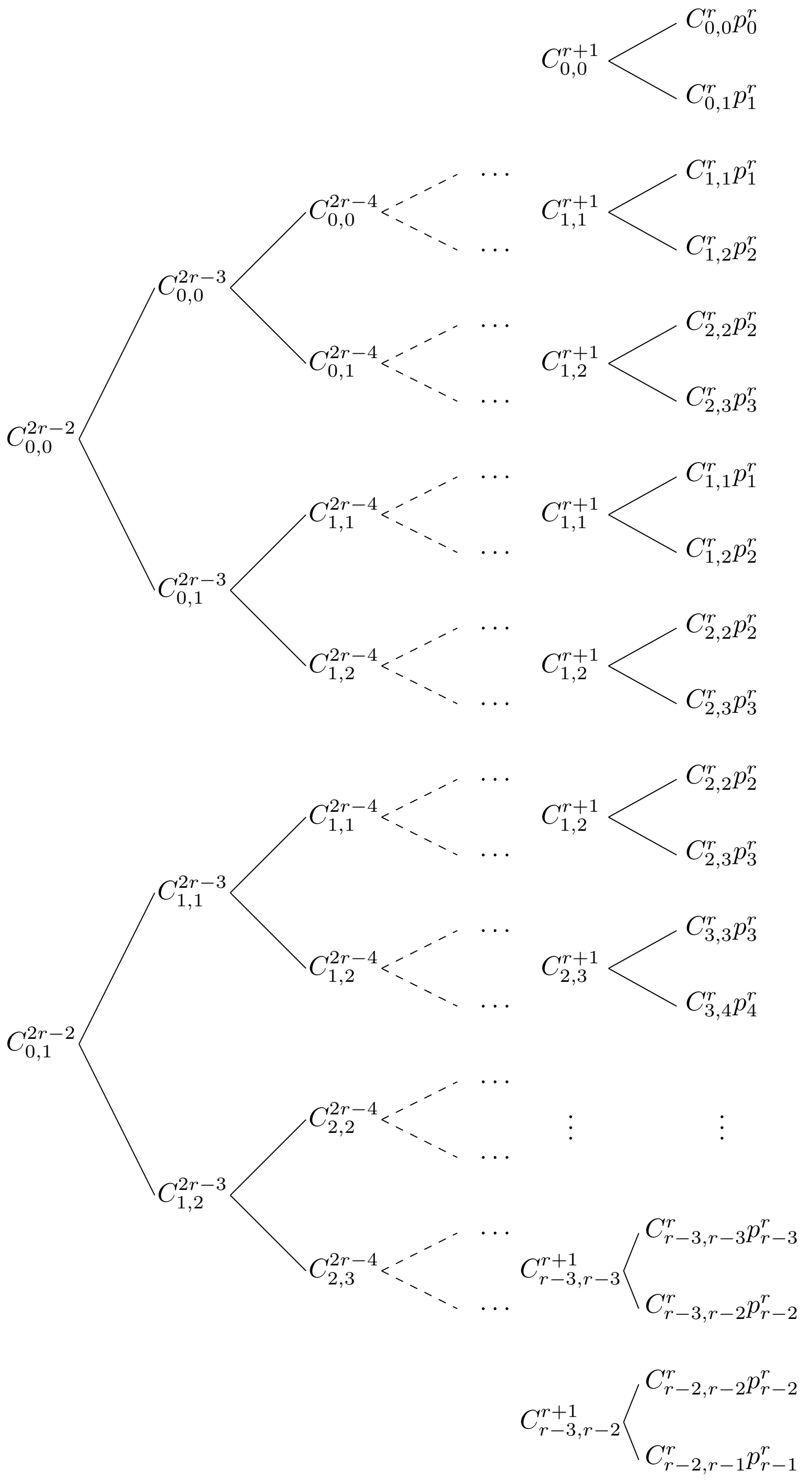


Figure 1: Diagram showing the structure of the optimal weights needed to obtain optimal order of accuracy, [3].

Corollary 2.2. *Let $r \leq l \leq 2r-2$ and $0 \leq k \leq (2r-2)-l$, if we denote as $C^l_{k,k}$ and $C^l_{k,k+1}$ the values which satisfy:*

$$p_k^{l+1}(x_{i-1/2}) = C^l_{k,k} p^l_k(x_{i-1/2}) + C^l_{k,k+1} p^l_{k+1}(x_{i-1/2}), \tag{8}$$

then

$$C^l_{k,k} = \frac{2(l-r+k+1)+1}{2(l+1)}, \quad C^l_{k,k+1} = 1 - C^l_{k,k} = \frac{2(r-k)-1}{2(l+1)}. \tag{9}$$

From Eq. (5), we can deduce the following recursive method:

$$\begin{cases} \tilde{p}^r_j(x^*) = p^r_j(x^*), & j = 0, \ldots, r-1, \\ \tilde{p}^{l+1}_j(x^*) = \tilde{\omega}^l_{j,j}(x^*)\tilde{p}^l_j(x^*) + \tilde{\omega}^l_{j,j+1}(x^*)\tilde{p}^l_{j+1}(x^*), & l = r, \ldots, 2r-2, \; j = 0, \ldots, 2r-2-l, \end{cases} \tag{10}$$

with the non-linear weights defined as:

$$\tilde{\omega}^l_{j,j_1}(x^*) = \frac{\tilde{\alpha}^l_{j,j_1}(x^*)}{\tilde{\alpha}^l_{j,j}(x^*) + \tilde{\alpha}^l_{j,j+1}(x^*)}, \quad \tilde{\alpha}^l_{j,j_1}(x^*) = \frac{C^l_{j,j_1}(x^*)}{(\epsilon + I^l_{j,j_1})^t}, \quad j_1 = j, \; j+1, \tag{11}$$

where $C^r_{j,j_1}(x^*)$, $j_1 = j, j+1$ are determined in Eq. (7). The values I^l_{j,j_1}, explained in subsection 2.1, are smoothness indicators.

With this formulation, the approximation is defined by

$$\tilde{\mathcal{I}}^{2r-1}(x^*; f) = \tilde{p}^{2r-1}_0(x^*), \tag{12}$$

being $\tilde{p}^{2r-1}_0(x^*)$ the result of the recursive process, Eq. (10).

2.1. On smoothness indicators and the accuracy of the new univariate progressive WENO method

The choice of the smoothness indicators is crucial. To design the new non-linear algorithm, we only calculate the smoothness indicators at level $l = r$, and we use them in all the levels, as we can see in this subsection. In [6] it is proved that if the smoothness indicators satisfy the following conditions:

P1 The order of a smoothness indicator that is free of discontinuities is h^2, i.e.

$$I^r_k = O(h^2) \text{ if } f \text{ is smooth in } \mathcal{S}^r_k.$$

P2 The distance between two smoothness indicators free of discontinuities is h^{r+1}, i.e. let $k, k' \in \{0, 1, \ldots, r-1\}$ be such that there does not exist any discontinuity in $\mathcal{S}^r_k$ and $\mathcal{S}^r_{k'}$, then

$$I^r_k - I^r_{k'} = O(h^{r+1}).$$

P3 When a discontinuity crosses the stencil $\mathcal{S}^r_k$ then

$$I^r_k \nrightarrow 0 \text{ as } h \to 0.$$

Then, the optimal weights satisfy Eq. (3) for determined parameters t and ϵ (Proposition 2 in [6]). We exploit this result to construct the smoothness indicators in each step of the Aitken-Neville's algorithm. We will use the following definition (Definition 4.1 in [3]).

Definition 1. *Let $l = r, \ldots, 2r-2$, and I^r_k, with $k = 0, \ldots, r-1$, be the smoothness indicators with properties **P1**,**P2** and **P3**. Then, we define the smoothness indicators at level l as:*

$$\begin{aligned} I^l_{k,k} &= I^r_k, \quad k = 0, \ldots, (2r-2)-l, \\ I^l_{k,k+1} &= I^r_{l-(r-1)+k}, \quad k = 0, \ldots, (2r-2)-l. \end{aligned} \tag{13}$$

Therefore, in each step, we will remove the part which is "contaminated" by a discontinuity. In particular, we can prove the following proposition adapting the proof of Proposition 4.4 in [3] to any point $x^* \in (x_{i-1}, x_i)$.

Proposition 2.3. *Let $r \leq l \leq 2r-2$, $0 \leq k \leq (2r-2)-l$, $x^* \in (x_{i-1}, x_i)$, and let $I^l_{k,k}$, and $I^l_{k,k+1}$ be smoothness indicators defined in Definition 1. The following weights:*

$$\tilde{\omega}^l_{k,k}(x^*) = \frac{\tilde{\alpha}^l_{k,k}(x^*)}{\tilde{\alpha}^l_{k,k}(x^*) + \tilde{\alpha}^l_{k,k+1}(x^*)}, \quad \tilde{\omega}^l_{k,k+1}(x^*) = \frac{\tilde{\alpha}^l_{k,k+1}(x^*)}{\tilde{\alpha}^l_{k,k}(x^*) + \tilde{\alpha}^l_{k,k+1}(x^*)}, \tag{14}$$

with,

$$\tilde{\alpha}^l_{k,k}(x^*) = \frac{C^l_{k,k}(x^*)}{(\epsilon + I^l_{k,k})^t}, \quad \tilde{\alpha}^l_{k,k+1}(x^*) = \frac{C^l_{k,k+1}(x^*)}{(\epsilon + I^l_{k,k+1})^t}, \tag{15}$$

and with $C^l_{k,k}(x^*) + C^l_{k,k+1}(x^*) = 1$, will fall under one of the following cases:

1. *If neither $I^l_{k,k}$ nor $I^l_{k,k+1}$ are affected by a discontinuity, then $\tilde{\omega}^l_{k,k}(x^*) = C^l_{k,k}(x^*) + O(h^{r-1})$ and $\omega^l_{k,k+1}(x^*) = C^l_{k,k+1}(x^*) + O(h^{r-1})$.*

2. *If $I^l_{k,k+1}$ is affected by a singularity, then $\tilde{\omega}^l_{k,k}(x^*) = 1 + O(h^{2t})$ and $\tilde{\omega}^l_{k,k+1}(x^*) = O(h^{2t})$.*

3. *If $I^l_{k,k}$ is affected by a singularity then $\tilde{\omega}^l_{k,k+1}(x^*) = 1 + O(h^{2t})$ and $\tilde{\omega}^l_{k,k}(x^*) = O(h^{2t})$.*

4. *If $I^l_{k,k}$ and $I^l_{k,k+1}$ are affected by a singularity then $\tilde{\omega}^l_{k,k}(x^*) \nrightarrow 0$ and $\tilde{\omega}^l_{k,k+1}(x^*) \nrightarrow 0$.*

We have the following main result:

Theorem 2.4. *Let $1 < l_0 \leq r-1$, $x^* \in (x_{i-1}, x_i)$, and $\tilde{\mathcal{I}}^{2r-1}(x^*; f)$ the result of the recursive process, Eq.* (10)*, if f is smooth in $[x_{i-r}, x_{i+r-1}] \setminus \Omega$ and f has a discontinuity at Ω then*

$$\tilde{\mathcal{I}}^{2r-1}(x^*; f) - f(x^*) = \begin{cases} O(h^{2r}), & \text{if } \Omega = \emptyset; \\ O(h^{r+l_0}), & \text{if } \Omega = [x_{i+l_0-1}, x_{i+l_0}]. \end{cases} \tag{16}$$

By symmetry, we only analyze when there exists an isolated discontinuity at an interval $[x_{i-1+l_0}, x_{i+l_0}]$, $l_0 = 1, \ldots, r-1$ (analogously, we obtain the equivalent symmetric results for $[x_{i-r+l_0}, x_{i-r+l_0+1}]$, $l_0 = 0, \ldots, r-2$).

In the next sections, we generalize this method to multi-dimensions. We also introduce some possible smoothness indicators which satisfy **P1**,**P2**, and **P3** in Section 6.

3. Comparison of multivariate linear Lagrange interpolation with the non-linear WENO method in Cartesian grids

In this section, we briefly review the Lagrange interpolation problem in several variables when the data are located in Cartesian grids (non-necessarily equally-spaced) and we construct a multivariate WENO method following the ideas presented in [7] for two variables. We present the necessary ingredients to extend the progressive WENO-$2r$ algorithm introduced in Section 2 to several variables.

3.1. Multivariate linear interpolation

Let us start supposing $a_j, b_j \in \mathbb{R}$ with $a_j < b_j$, $j = 1, \ldots, n$, and an hypercube denoted as:

$$\prod_{j=1}^{n} [a_j, b_j] = [a_1, b_1] \times \ldots \times [a_n, b_n].$$

We call the points of a grid for each interval as:

$$a_j = x_j^{(0)} < x_j^{(1)} < x_j^{(2)} < \ldots < x_j^{(J_j)} = b_j, \quad j = 1, \ldots, n,$$

and define

$$h_j = \max_{i=1,\ldots,J_j} |x_j^{(i)} - x_j^{(i-1)}|, \quad h := \max_{j=1,\ldots,n} h_j.$$

We suppose an unknown function $f : \prod_{j=1}^{n}[a_j, b_j] \to \mathbb{R}$, and consider our data as the evaluation of this function in the points of the Cartesian grid, i.e.

$$f_{(l_1,\ldots,l_n)} = f(x_1^{(l_1)}, x_2^{(l_2)}, \ldots, x_n^{(l_n)}), \quad 0 \leq l_j \leq J_j, \quad 1 \leq j \leq n.$$

Let the polynomials of n variables be denoted by:

$$\Pi_n^{\tau_1,\ldots,\tau_n} = \{p(x_1,\ldots,x_n) = \sum_{l_1=0}^{\tau_1} \ldots \sum_{l_n=0}^{\tau_n} a_{(l_1,\ldots,l_n)} x_1^{l_1} x_2^{l_2} \ldots x_n^{l_n} \mid a_{(l_1,\ldots,l_n)} \in \mathbb{R},\ 0 \leq l_j \leq \tau_j,\ j = 1,\ldots,n\},$$

when $\tau_1 = \ldots = \tau_n = \tau$ we call it $\Pi_n^{\tau} := \Pi_n^{\tau_1,\ldots,\tau_n}$. We consider $(i_1,\ldots,i_n) \in \mathbb{N}^n$ to be the reference index where the approximation is centered at, and $r \in \mathbb{N}$ such that $0 \leq i_j - r \leq i_j + r - 1 \leq J_j$, $j = 1,\ldots,n$, and the centered stencil:

$$\begin{aligned} \mathcal{S}_{\mathbf{0}}^{2r} &= \{x_1^{(i_1-r)}, \ldots, x_1^{(i_1+r-1)}\} \times \{x_2^{(i_2-r)}, \ldots, x_2^{(i_2+r-1)}\} \times \ldots \times \{x_n^{(i_n-r)}, \ldots, x_n^{(i_n+r-1)}\} \\ &=: (\mathcal{S}_0^{2r})_1 \times \ldots \times (\mathcal{S}_0^{2r})_n = \prod_{j=1}^{n} (\mathcal{S}_0^{2r})_j, \end{aligned} \tag{17}$$

then, the problem consists in calculating a polynomial of degree $2r-1$ such that:

$$p(\mathbf{x}) = f(\mathbf{x}), \quad \forall\, \mathbf{x} \in \mathcal{S}_{(0,\ldots,0)}^{2r}.$$

To do so, we will use the Lagrange base of polynomials:

$$L_p^j(x_j) = \prod_{k=i_j-r, k\neq p}^{i_j+r-1} \left(\frac{x_j - x_j^{(k)}}{x_j^{(p)} - x_j^{(k)}} \right), \quad p = i_j - r, \ldots, i_j + r - 1, \quad j = 1, \ldots, n,$$

then, the unique polynomial is

$$p_{\mathbf{0}}^{2r-1}(x_1,\ldots,x_n) = \sum_{l_1=i_1-r}^{i_1+r-1} \sum_{l_2=i_2-r}^{i_2+r-1} \cdots \sum_{l_n=i_n-r}^{i_n+r-1} f(x_1^{(l_1)}, x_2^{(l_2)}, \ldots, x_n^{(l_n)}) L_{l_1}^1(x_1) \ldots L_{l_n}^n(x_n).$$

It can be checked (see [7]) that if $\mathbf{x}^* = (x_1^*, \ldots, x_n^*) \in \prod_{j=1}^{n}[x_j^{(i_j-1)}, x_j^{(i_j)}]$, and $f \in \mathcal{C}^{2nr}(\mathbb{R})$ then the error satisfies:

$$E(\mathbf{x}^*) = f(\mathbf{x}^*) - p_{\mathbf{0}}^{2r-1}(\mathbf{x}^*) = O(h^{2r}), \tag{18}$$

and if $\mathbf{m} = (m_1, \ldots, m_n) \in \mathbb{N}^n$ with $m_j \leq 2r - 1$, $j = 1, \ldots, n$, we get

$$E^{(\mathbf{m})}(\mathbf{x}^*) = E^{(m_1,\ldots,m_n)}(\mathbf{x}^*) = O(h^{2r-\max\{m_1,\ldots,m_n\}}), \tag{19}$$

where

$$E^{(m_1,\ldots,m_n)}(\mathbf{x}) = \frac{\partial^{m_1+\ldots+m_n} E}{\partial^{m_1} x_1 \ldots \partial^{m_n} x_n}(\mathbf{x}).$$

3.2. Multivariate WENO interpolation in Cartesian grids

Using the same notation as in the previous section, the goal is to construct a non-linear interpolant in multi-dimensions in the same way as we reviewed in $1d$, i.e., an interpolant with maximum order $2r$ in the smooth parts and with order $r+1$ when there exists a discontinuity which crosses, at least, one small stencil.

In this case, we introduce the method supposing that we want to obtain an approximation of the function f evaluated at any point of the hypercube $\mathbf{x}^* \in \prod_{j=1}^{n}[x_j^{(i_j-1)}, x_j^{(i_j)}]$. The construction is similar to the $1d$ case: Firstly, we make the linear combination of interpolants of lower degree, r, if $\mathbf{k} = (k_1, \ldots, k_n)$ then

$$p_0^{2r-1}(\mathbf{x}^*) = \sum_{\mathbf{k}\in\{0,1,\ldots,r-1\}^n} C_{\mathbf{k}}^r(\mathbf{x}^*) p_{\mathbf{k}}^r(\mathbf{x}^*),$$

where $C^r_{\mathbf{k}}(\mathbf{x}^*)$ are the optimal weights, and it is not difficult to prove that they have the explicit form:

$$C^r_{\mathbf{k}}(\mathbf{x}^*) = \prod_{j=1}^{n} C^r_{k_j}(x^*_j), \quad \mathbf{k} \in \{0,\dots,r-1\}^n,$$

being $C^r_{k_j}(x^*_j)$, $k_j = 0,\dots,r-1$, $j=1,\dots,n$ the optimal weights in $1d$. In the case that we want to approximate at the mid-point of the hypercube, i.e. $\mathbf{x}^* = \mathbf{x}^*_{\frac{1}{2}}$, then they are defined in Eq. (2). The polynomials $p^r_{\mathbf{k}}$ interpolate at the nodes

$$\mathcal{S}^r_{\mathbf{k}} = \{x_1^{(i_1+k_1-r)},\dots,x_1^{(i_1+k_1)}\} \times \{x_2^{(i_2+k_2-r)},\dots,x_2^{(i_2+k_2)}\} \times \dots \times \{x_n^{(i_n+k_n-r)},\dots,x_n^{(i_n+k_n)}\} = \prod_{j=1}^{n} \mathcal{S}^r_{k_j}.$$

Then, we replace the optimal weights by non-linear ones using the following formula

$$\omega^r_{\mathbf{k}}(\mathbf{x}^*) = \frac{\alpha^r_{\mathbf{k}}(\mathbf{x}^*)}{\sum_{\mathbf{l}\in\{0,\dots,r-1\}^n} \alpha^r_{\mathbf{l}}(\mathbf{x}^*)}, \text{ with } \alpha^r_{\mathbf{k}}(\mathbf{x}^*) = \frac{C^r_{\mathbf{k}}(\mathbf{x}^*)}{(\epsilon + I^r_{\mathbf{k}})^t}, \tag{20}$$

being $I^r_{\mathbf{k}}$ the smoothness indicators with the same requirements pointed out in Section 2, i.e.:

P1 The order of the smoothness indicator free of discontinuities is h^2, i.e.

$$I^r_{\mathbf{k}} = O(h^2) \text{ if } f \text{ is smooth in } \mathcal{S}^r_{\mathbf{k}}.$$

P2 The distance between two smoothness indicators free of discontinuities is h^{r+1}, i.e. let $\mathbf{k} = (k_1,\dots,k_n)$, and $\mathbf{k}' = (k'_1,\dots,k'_n)$ be such that there does not exist any discontinuity in $\mathcal{S}^r_{\mathbf{k}}$ and $\mathcal{S}^r_{\mathbf{k}'}$ then

$$I^r_{\mathbf{k}} - I^r_{\mathbf{k}'} = O(h^{r+1}).$$

P3 When a discontinuity crosses the stencil $\mathcal{S}^r_{\mathbf{k}}$ then:

$$I^r_{\mathbf{k}} \nrightarrow 0 \text{ as } h \to 0.$$

An example of smoothness indicators will be introduced in Section 6. The parameters ϵ and t are chosen to obtain maximum order. In our case, following [6] and [7] we will take $\epsilon = h^2$ and $t = \frac{1}{2}(r+1)$. Therefore, we state the next result, which is similar to Proposition 2 in [6] and Theorem 1 in [7].

Theorem 3.1. *Let* $\mathbf{x}^* \in \prod_{j=1}^n [x_j^{(i_j-1)}, x_j^{(i_j)}]$, *the multivariate WENO interpolant*

$$\mathcal{I}^{2r-1}(\mathbf{x}^*; f) = \sum_{\mathbf{k}\in\{0,1,\dots,r-1\}^n} \omega^r_{\mathbf{k}}(\mathbf{x}^*) p^r_{\mathbf{k}}(\mathbf{x}^*) \tag{21}$$

with $\omega^r_{\mathbf{k}}$, $\mathbf{k} \in \{0,1,\dots,r-1\}^n$ *defined in Eq.* (20), *with smoothness indicator* $I^r_{\mathbf{k}}$, $\mathbf{k} \in \{0,1,\dots,r-1\}^n$ *fulfilling* ***P1***, ***P2***, ***P3***; $\epsilon = h^2$ *and* $t = \frac{1}{2}(r+1)$ *satisfies:*

$$f(\mathbf{x}^*) - \mathcal{I}^{2r-1}(\mathbf{x}^*; f) = \begin{cases} O(h^{2r}), & \textit{at smooth regions,} \\ O(h^{r+1}), & \textit{if, at least, one stencil lies in a smooth region.} \end{cases}$$

4. A new progressive bivariate, $n = 2$, WENO method

The idea of this new method is to reach the maximum possible order of accuracy when one discontinuity crosses the largest stencil. For this purpose, we use the Aitken-Neville-based algorithm developed in multi-dimensions, [11]. We start with a polynomial of degree $2r-1$ and decompose it in 2^2 polynomials of degree $2r-2$ obtaining as weights polynomials of degree 1 that will be replaced by non-linear weights, depending on the location of the discontinuity. Afterwards, we continue decomposing the 2^2 polynomials of degree $2r-2$ in polynomials of degree $2r-3$, and so on. In each step, the non-linear weights determine the stencils free of discontinuity, which will be used to approximate the value. The procedure is similar to the method expounded in Section 2. For ease of reading, we start with $n = 2$ in this section, and then we extend our results to any dimension n in Section 5.

4.1. A first example: A progressive bivariate WENO method with $r = 2$

Let us start by designing a bivariate WENO method with $r = 2$. We suppose a non necessarily uniform grid in $[a_1, b_1] \times [a_2, b_2]$ defined by $\{(x_1^{(l_1)}, x_2^{(l_2)})\}_{(l_1,l_2)=(0,0)}^{(J_1,J_2)}$, with $a_1 = x_1^{(0)} < x_1^{(1)} < \ldots < x_1^{(J_1)} = b_1$, and $a_2 = x_2^{(0)} < x_2^{(1)} < \ldots < x_2^{(J_2)} = b_2$, and the data $f_{l_1,l_2} = f(x_1^{(l_1)}, x_2^{(l_2)})$, $l_j = 0, \ldots, J_j$, with $j = 1, 2$. Let $(i_1, i_2) \in \mathbb{N}^2$ such that $0 \leq i_j - 2$ and $i_j + 1 \leq J_j$, with $j = 1, 2$. We determine the largest stencil

$$\mathcal{S}^4_{(0,0)} = \{x_1^{(i_1-2)}, x_1^{(i_1-1)}, x_1^{(i_1)}, x_1^{(i_1+1)}\} \times \{x_2^{(i_2-2)}, x_2^{(i_2-1)}, x_2^{(i_2)}, x_2^{(i_2+1)}\} = (\mathcal{S}^4_0)_1 \times (\mathcal{S}^4_0)_2,$$

and we want to interpolate at any point $\mathbf{x}^* = (x_1^*, x_2^*) \in [x_1^{(i_1-1)}, x_1^{(i_1)}] \times [x_2^{(i_2-1)}, x_2^{(i_2)}]$. We compute the polynomial

$$p^3_{(0,0)}(x_1, x_2) = \sum_{l_1,l_2=0}^{3} a_{l_1,l_2} x_1^{l_1} x_2^{l_2},$$

such that $p^3_{(0,0)}(x_1^{(l_1)}, x_2^{(l_2)}) = f_{l_1,l_2}$ if $(x_1^{(l_1)}, x_2^{(l_2)}) \in \mathcal{S}^4_{(0,0)}$. Again, we can express it as the sum of 2^2 polynomials that interpolate in the stencils:

$$\begin{aligned}
\mathcal{S}^3_{(0,0)} &= \{x_1^{(i_1-2)}, x_1^{(i_1-1)}, x_1^{(i_1)}\} \times \{x_2^{(i_2-2)}, x_2^{(i_2-1)}, x_2^{(i_2)}\} = (\mathcal{S}^3_0)_1 \times (\mathcal{S}^3_0)_2,\\
\mathcal{S}^3_{(1,0)} &= \{x_1^{(i_1-1)}, x_1^{(i_1)}, x_1^{(i_1+1)}\} \times \{x_2^{(i_2-2)}, x_2^{(i_2-1)}, x_2^{(i_2)}\} = (\mathcal{S}^3_1)_1 \times (\mathcal{S}^3_0)_2,\\
\mathcal{S}^3_{(0,1)} &= \{x_1^{(i_1-2)}, x_1^{(i_1-1)}, x_1^{(i_1)}\} \times \{x_2^{(i_2-1)}, x_2^{(i_2)}, x_2^{(i_2+1)}\} = (\mathcal{S}^3_0)_1 \times (\mathcal{S}^3_1)_2,\\
\mathcal{S}^3_{(1,1)} &= \{x_1^{(i_1-1)}, x_1^{(i_1)}, x_1^{(i_1+1)}\} \times \{x_2^{(i_2-1)}, x_2^{(i_2)}, x_2^{(i_2+1)}\} = (\mathcal{S}^3_1)_1 \times (\mathcal{S}^3_1)_2,
\end{aligned}$$

called $p^2_{(0,0)}, p^2_{(1,0)}, p^2_{(0,1)}, p^2_{(1,1)}$ using, as we mentioned before, the Aitken-Neville-type formula given in [11]:

$$p^3_{(0,0)}(\mathbf{x}^*) = \sum_{\mathbf{j}_0 \in \{0,1\}^2} C^2_{(0,0),\mathbf{j}_0}(\mathbf{x}^*) p^2_{\mathbf{j}_0}(\mathbf{x}^*), \tag{22}$$

with

$$\begin{aligned}
C^2_{(0,0),(0,0)}(\mathbf{x}^*) &= \left(\frac{x_1^* - x_1^{(i_1+1)}}{x_1^{(i_1-2)} - x_1^{(i_1+1)}}\right)\left(\frac{x_2^* - x_2^{(i_2+1)}}{x_2^{(i_2-2)} - x_2^{(i_2+1)}}\right) = C^2_{0,0}(x_1^*)C^2_{0,0}(x_2^*),\\
C^2_{(0,0),(1,0)}(\mathbf{x}^*) &= \left(\frac{x_1^* - x_1^{(i_1-2)}}{x_1^{(i_1+1)} - x_1^{(i_1-2)}}\right)\left(\frac{x_2^* - x_2^{(i_2+1)}}{x_2^{(i_2-2)} - x_2^{(i_2+1)}}\right) = C^2_{0,1}(x_1^*)C^2_{0,0}(x_2^*),\\
C^2_{(0,0),(0,1)}(\mathbf{x}^*) &= \left(\frac{x_1^* - x_1^{(i_1+1)}}{x_1^{(i_1-2)} - x_1^{(i_1+1)}}\right)\left(\frac{x_2^* - x_2^{(i_2-2)}}{x_2^{(i_2+1)} - x_2^{(i_2-2)}}\right) = C^2_{0,0}(x_1^*)C^2_{0,1}(x_2^*),\\
C^2_{(0,0),(1,1)}(\mathbf{x}^*) &= \left(\frac{x_1^* - x_1^{(i_1-2)}}{x_1^{(i_1+1)} - x_1^{(i_1-2)}}\right)\left(\frac{x_2^* - x_2^{(i_2-2)}}{x_2^{(i_2+1)} - x_2^{(i_2-2)}}\right) = C^2_{0,1}(x_1^*)C^2_{0,1}(x_2^*).
\end{aligned}$$

Finally, we define the non-linear weights as:

$$\tilde{\omega}^2_{(0,0),\mathbf{k}}(\mathbf{x}^*) = \frac{\tilde{\alpha}^2_{(0,0),\mathbf{k}}(\mathbf{x}^*)}{\sum_{\mathbf{l}\in\{0,1\}^2} \tilde{\alpha}^2_{(0,0),\mathbf{l}}(\mathbf{x}^*)}, \quad \text{where} \quad \tilde{\alpha}^2_{(0,0),\mathbf{k}}(\mathbf{x}^*) = \frac{C^2_{(0,0),\mathbf{k}}(\mathbf{x}^*)}{(\epsilon + I^2_{(0,0),\mathbf{k}})^t}, \quad \mathbf{k} \in \{0,1\}^2,$$

with $I^2_{(0,0),\mathbf{k}}$ defined as:

$$I^2_{(0,0),(0,0)} = I^2_{(0,0)}, \quad I^2_{(0,0),(0,1)} = I^2_{(0,1)}, \quad I^2_{(0,0),(1,0)} = I^2_{(1,0)}, \quad I^2_{(0,0),(1,1)} = I^2_{(1,1)},$$

where $I^2_{\mathbf{k}}$, $\mathbf{k} \in \{0,1\}^2$ are smoothness indicators which satisfy the properties **P1**, **P2** and **P3**. The parameters t and ϵ are defined in Section 3.2 as h^2 and $r + 1 = 3$ respectively. Thus, we define the new interpolant as:

$$\mathcal{I}^3(\mathbf{x}^*; f) = \sum_{\mathbf{j}_0 \in \{0,1\}^2} \omega^2_{(0,0),\mathbf{j}_0}(\mathbf{x}^*) p^2_{\mathbf{j}_0}(\mathbf{x}^*).$$

If we apply it at mid-points, the interpolant presented in [7] and the method showed in Section 3.2 for $n = 2$ are similar if the same smoothness indicators are used. Therefore, when $r = 2$ there are not differences between the progressive WENO and the classical one. We show the construction of the new method for $r = 3$.

4.2. The case $r = 3$

In this case, we use all the points of the following largest stencil:

$$\mathcal{S}^6_{(0,0)} = \{x_1^{(i_1-3)}, x_1^{(i_1-2)}, x_1^{(i_1-1)}, x_1^{(i_1)}, x_1^{(i_1+1)}, x_1^{(i_1+2)}\} \times \{x_2^{(i_2-3)}, x_2^{(i_2-2)}, x_2^{(i_2-1)}, x_2^{(i_2)}, x_2^{(i_2+1)}, x_2^{(i_2+2)}\}.$$

We follow the recursive formula described in (10). Thus, we start calculating the evaluation at the point $\mathbf{x}^*$ of the polynomials:

$$p^3_{\mathbf{j}_1}(\mathbf{x}^*), \quad \mathbf{j}_1 = (j_1^{(1)}, j_2^{(1)}) \in \{0,1,2\}^2,$$

being the stencils used for constructing each polynomial $\mathcal{S}^3_{(l_1,l_2)}$, with $l_j = 0,1,2$, and $j = 1,2$, see Figure 2. And, by means of the Aitken-Neville formula, we represent $p^4_{\mathbf{j}_0}$, $\mathbf{j}_0 = (j_1^{(0)}, j_2^{(0)}) \in \{0,1\}^2$, whose stencils are displayed in Figure 3, as the sum of the evaluations of polynomials of degree 3 (Figure 4)

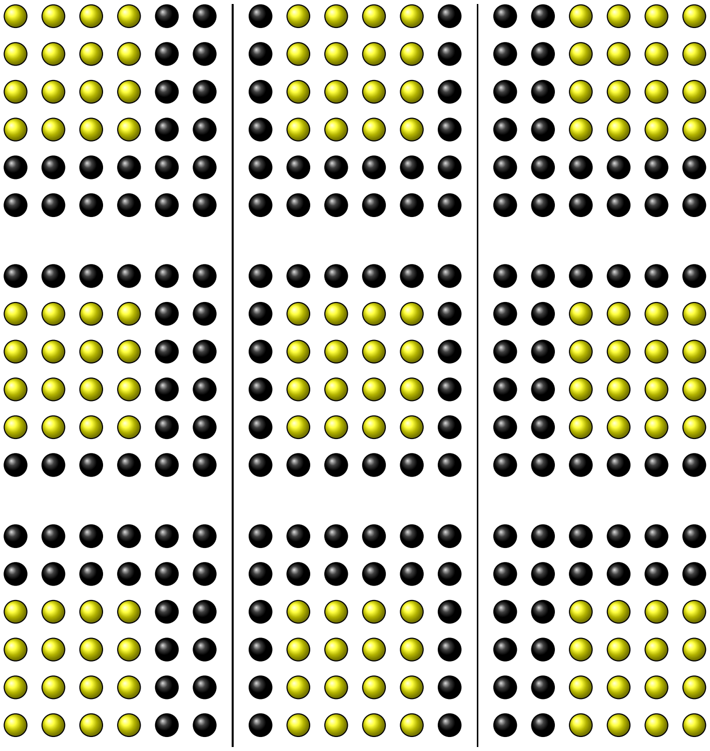


Figure 2: Stencils used to get $p^3_{\mathbf{j}_1}(\mathbf{x}^*)$, $\mathbf{j}_1 \in \{0,1,2\}^2$. They are used in classical bivariate WENO [7]

$$\begin{aligned} p^4_{(0,0)}(\mathbf{x}^*) &= \sum_{\mathbf{j}_1 \in \{0,1\}\times\{0,1\}} C^3_{(0,0),\mathbf{j}_1}(\mathbf{x}^*) p^3_{\mathbf{j}_1}(\mathbf{x}^*), \quad & p^4_{(1,0)}(\mathbf{x}^*) &= \sum_{\mathbf{j}_1 \in \{1,2\}\times\{0,1\}} C^3_{(1,0),\mathbf{j}_1}(\mathbf{x}^*) p^3_{\mathbf{j}_1}(\mathbf{x}^*), \\ p^4_{(0,1)}(\mathbf{x}^*) &= \sum_{\mathbf{j}_1 \in \{0,1\}\times\{1,2\}} C^3_{(0,1),\mathbf{j}_1}(\mathbf{x}^*) p^3_{\mathbf{j}_1}(\mathbf{x}^*), \quad & p^4_{(1,1)}(\mathbf{x}^*) &= \sum_{\mathbf{j}_1 \in \{1,2\}\times\{1,2\}} C^3_{(1,1),\mathbf{j}_1}(\mathbf{x}^*) p^3_{\mathbf{j}_1}(\mathbf{x}^*), \end{aligned} \tag{23}$$

with

$$C^3_{\mathbf{j}_0,\mathbf{j}_1} = C^3_{(j_1^{(0)},j_2^{(0)}),(j_1^{(1)},j_2^{(1)})} = C^3_{j_1^{(0)},j_1^{(1)}} C^3_{j_2^{(0)},j_2^{(1)}}, \ \mathbf{j}_0 \in \{0,1\}^2, \ \mathbf{j}_1 \in \mathbf{j}_0 + \{0,1\}^2,$$

where C^3_{k,k_1}, $k = 0,1$, $k_1 = k + \{0,1\}$ are defined in Eq. (7). Now, we replace in Eq. (23) the linear-weights for non-linear ones:

$$\tilde{\omega}^3_{\mathbf{j}_0,\mathbf{j}_1}(\mathbf{x}^*) = \frac{\tilde{\alpha}^3_{\mathbf{j}_0,\mathbf{j}_1}(\mathbf{x}^*)}{\sum_{\mathbf{l}\in\mathbf{j}_0+\{0,1\}^2} \tilde{\alpha}^3_{\mathbf{j}_0,\mathbf{l}}(\mathbf{x}^*)}, \quad \text{where} \ \ \tilde{\alpha}^3_{\mathbf{j}_0,\mathbf{j}_1}(\mathbf{x}^*) = \frac{C^3_{\mathbf{j}_0,\mathbf{j}_1}(\mathbf{x}^*)}{(\epsilon + I^3_{\mathbf{j}_0,\mathbf{j}_1})^t}, \quad \mathbf{j}_0 \in \{0,1\}^2, \mathbf{j}_1 \in \mathbf{j}_0 + \{0,1\}^2,$$

and

$$I^3_{\mathbf{j}_0,\mathbf{j}_1}=I^3_{\mathbf{j}_1},\ \mathbf{j}_0\in\{0,1\}^2,\ \mathbf{j}_1\in\mathbf{j}_0+\{0,1\}^2,$$

being $I^3_{\mathbf{j}_1}$ smoothness indicators which satisfy the properties **P1**, **P2** and **P3**. We obtain the approximation:

$$\begin{aligned}\tilde{p}^4_{(0,0)}(\mathbf{x}^*)&=\sum_{\mathbf{j}_1\in\{0,1\}\times\{0,1\}}\tilde{\omega}^3_{(0,0),\mathbf{j}_1}(\mathbf{x}^*)p^3_{\mathbf{j}_1}(\mathbf{x}^*), &\tilde{p}^4_{(1,0)}(\mathbf{x}^*)&=\sum_{\mathbf{j}_1\in\{1,2\}\times\{0,1\}}\tilde{\omega}^3_{(1,0),\mathbf{j}_1}(\mathbf{x}^*)p^3_{\mathbf{j}_1}(\mathbf{x}^*),\\ \tilde{p}^4_{(0,1)}(\mathbf{x}^*)&=\sum_{\mathbf{j}_1\in\{0,1\}\times\{1,2\}}\tilde{\omega}^3_{(0,1),\mathbf{j}_1}(\mathbf{x}^*)p^3_{\mathbf{j}_1}(\mathbf{x}^*), &\tilde{p}^4_{(1,1)}(\mathbf{x}^*)&=\sum_{\mathbf{j}_1\in\{1,2\}\times\{1,2\}}\tilde{\omega}^3_{(1,1),\mathbf{j}_1}(\mathbf{x}^*)p^3_{\mathbf{j}_1}(\mathbf{x}^*).\end{aligned}\tag{24}$$

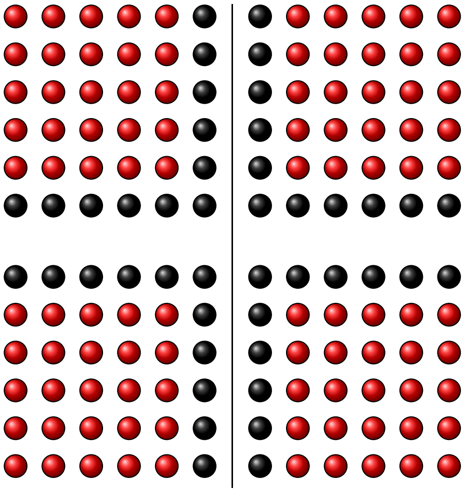


Figure 3: Stencils used to get $p^4_{\mathbf{j}_0}(\mathbf{x}^*)$, $\mathbf{j}_0\in\{0,1\}^2$.

The last step in our new algorithm for $r=3$ is to develop the polynomial of degree 5 as follows:

$$p^5_{(0,0)}(\mathbf{x}^*)=\sum_{\mathbf{j}_0\in\{0,1\}^2}C^4_{(0,0),\mathbf{j}_0}(\mathbf{x}^*)p^4_{\mathbf{j}_0}(\mathbf{x}^*),\tag{25}$$

with

$$\begin{aligned}C^4_{(0,0),(0,0)}(\mathbf{x}^*)&=\left(\frac{x_1^*-x_1^{(i_1+2)}}{x_1^{(i_1-3)}-x_1^{(i_1+2)}}\right)\left(\frac{x_2^*-x_2^{(i_2+2)}}{x_2^{(i_2-3)}-x_2^{(i_2+2)}}\right)=C^4_{0,0}(x_1^*)C^4_{0,0}(x_2^*),\\ C^4_{(0,0),(1,0)}(\mathbf{x}^*)&=\left(\frac{x_1^*-x_1^{(i_1-3)}}{x_1^{(i_1+2)}-x_1^{(i_1-3)}}\right)\left(\frac{x_2^*-x_2^{(i_2+2)}}{x_2^{(i_2-3)}-x_2^{(i_2+2)}}\right)=C^4_{0,1}(x_1^*)C^4_{0,0}(x_2^*),\\ C^4_{(0,0),(0,1)}(\mathbf{x}^*)&=\left(\frac{x_1^*-x_1^{(i_1+2)}}{x_1^{(i_1-3)}-x_1^{(i_1+2)}}\right)\left(\frac{x_2^*-x_2^{(i_2-3)}}{x_2^{(i_2+2)}-x_2^{(i_2-3)}}\right)=C^4_{0,0}(x_1^*)C^4_{0,1}(x_2^*),\\ C^4_{(0,0),(1,1)}(\mathbf{x}^*)&=\left(\frac{x_1^*-x_1^{(i_1-3)}}{x_1^{(i_1+2)}-x_1^{(i_1-3)}}\right)\left(\frac{x_2^*-x_2^{(i_2-3)}}{x_2^{(i_2+2)}-x_2^{(i_2-3)}}\right)=C^4_{0,1}(x_1^*)C^4_{0,1}(x_2^*),\end{aligned}$$

and to define the new approximation changing both the non-linear weights, and the approximation to polynomials of degree 4

$$\tilde{\mathcal{I}}^5(\mathbf{x}^*;f):=\tilde{p}^5_{(0,0)}(\mathbf{x}^*)=\sum_{\mathbf{j}_0\in\{0,1\}^2}\tilde{\omega}^4_{(0,0),\mathbf{j}_0}(\mathbf{x}^*)\tilde{p}^4_{\mathbf{j}_0}(\mathbf{x}^*),\tag{26}$$

being $\tilde{p}^4_{\mathbf{j}_0}(\mathbf{x}^*)$ obtained in Eq. (24) and

$$\tilde{\omega}^4_{(0,0),\mathbf{j}_0}(\mathbf{x}^*)=\frac{\alpha^4_{(0,0),\mathbf{j}_0}(\mathbf{x}^*)}{\sum_{\mathbf{l}\in\{0,1\}^2}\alpha^4_{(0,0),\mathbf{l}}(\mathbf{x}^*)},\quad\text{where}\quad\alpha^4_{(0,0),\mathbf{j}_0}(\mathbf{x}^*)=\frac{C^4_{(0,0),\mathbf{j}_0}(\mathbf{x}^*)}{(\epsilon+I^4_{(0,0),\mathbf{j}_0})^t},\quad\mathbf{j}_0\in\{0,1\}^2,$$

with the smoothness indicators defined in Eq. (13) as:

$$I^4_{(0,0),(0,0)} = I^3_{(0,0)}, \quad I^4_{(0,0),(0,1)} = I^3_{(0,2)}, \quad I^4_{(0,0),(1,0)} = I^3_{(2,0)}, \quad I^4_{(0,0),(1,1)} = I^3_{(2,2)},$$

where $I^3_{\mathbf{j}_0}$, $\mathbf{j}_0 \in \{0,1,2\}^2$ are smoothness indicators satisfying the properties **P1**, **P2**, and **P3**.

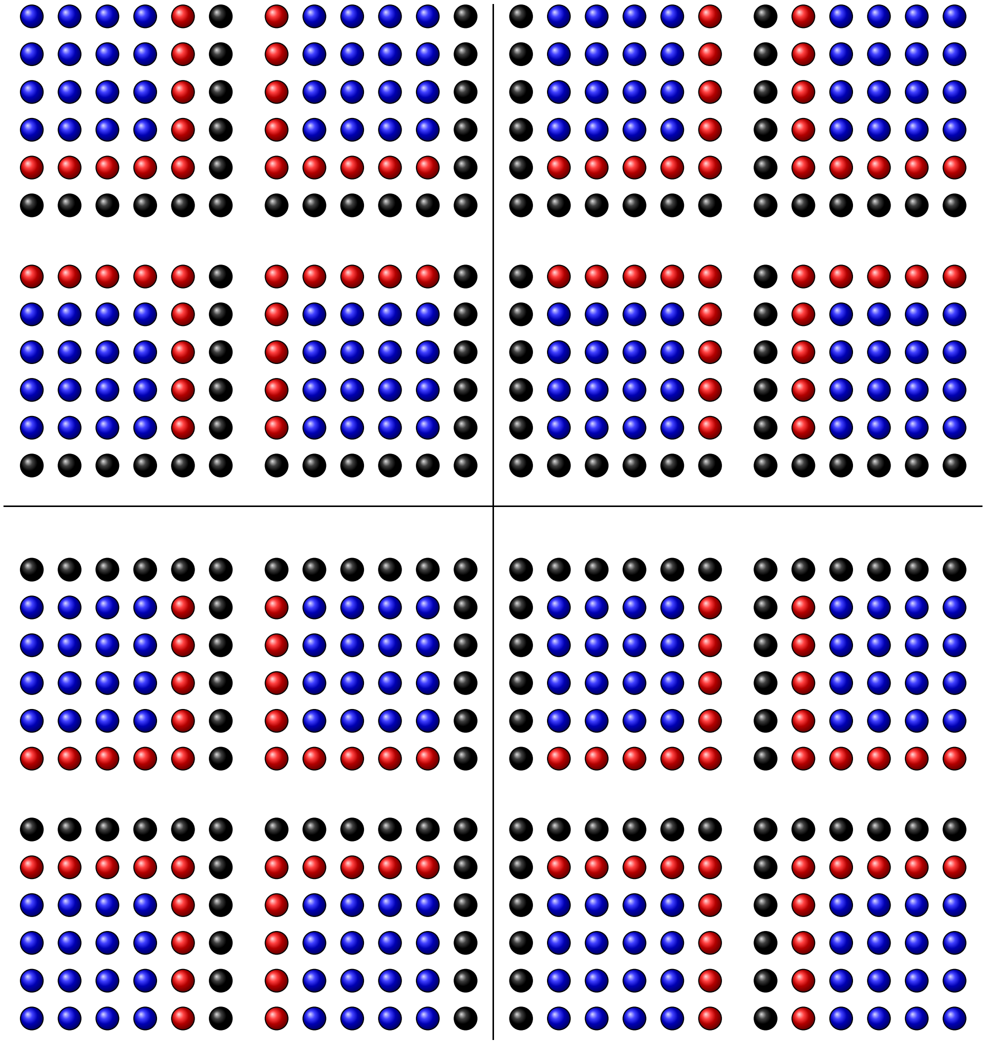


Figure 4: Contribution of each stencil of $p^3_{\mathbf{j}_1}(\mathbf{x}^*)$, $\mathbf{j}_1 \in \mathbf{j}_1 + \{0,1\}^2$ to the approximation of $p^4_{\mathbf{j}_0}(\mathbf{x}^*)$, $\mathbf{j}_0 \in \{0,1\}^2$.

4.3. The general case

Now, we construct the new bivariate WENO-$2r$ algorithm using a larger stencil of $(2r)^2$ points

$$\mathcal{S}^{2r}_{(0,0)} = \{x_1^{(i_1-r)}, x_1^{(i_1-r+1)}, \dots, x_1^{(i_1+r-1)}\} \times \{x_2^{(i_2-r)}, x_2^{(i_2-r+1)}, \dots, x_2^{(i_2+r-1)}\} = (\mathcal{S}^{2r}_0)_1 \times (\mathcal{S}^{2r}_0)_2.$$

The goal is to obtain maximum order of approximation if there exists an isolated discontinuity. We start representing the polynomial of degree $2r-1$ as follows:

$$p^{2r-1}_{(0,0)} = \sum_{\mathbf{j}_0 \in \{0,1\}^2} C^{2r-2}_{(0,0),\mathbf{j}_0} p^{2r-2}_{\mathbf{j}_0},$$

and repeat the process up to degree r:

$$p^{2r-1}_{(0,0)} = \sum_{\mathbf{j}_0\in\{0,1\}^2} C^{2r-2}_{(0,0),\mathbf{j}_0}\left(\sum_{\mathbf{j}_1\in\Omega_0} C^{2r-3}_{\mathbf{j}_0,\mathbf{j}_1}\left(\sum_{\mathbf{j}_2\in\Omega_1} C^{2r-4}_{\mathbf{j}_1,\mathbf{j}_2}\left(\dots\left(\sum_{\mathbf{j}_{r-3}\in\Omega_{r-4}} C^{r+1}_{\mathbf{j}_{r-4},\mathbf{j}_{r-3}}\left(\sum_{\mathbf{j}_{r-2}\in\Omega_{r-3}} C^{r}_{\mathbf{j}_{r-3},\mathbf{j}_{r-2}} p^r_{\mathbf{j}_{r-2}}\right)\right)\dots\right)\right)\right), \tag{27}$$

being $\Omega_l = \mathbf{j}_l + \{0,1\}^2$ with $1 \leq l \leq r-3$. Consequently, we determine the values $C^l_{\mathbf{j},\mathbf{j}_1}$ with $r \leq l \leq 2r-2$, and $\mathbf{j}_1 \in \mathbf{j} + \{0,1\}^2$, proving the following lemma.

Lemma 4.1. *Let $r \leq l \leq 2r-2$; $\mathbf{x}^* \in [x_1^{(i_1-1)}, x_1^{(i_1)}] \times [x_2^{(i_2-1)}, x_2^{(i_2)}]$, and $0 \leq j_1, j_2 \leq (2r-2)-l$, if we denote as $C^l_{\mathbf{j},\mathbf{j}_1}$ with $\mathbf{j}_1 \in \mathbf{j}+\{0,1\}^2$ the values which satisfy:*

$$p^{l+1}_{\mathbf{j}}(\mathbf{x}^*) = \sum_{\mathbf{j}_1\in\mathbf{j}+\{0,1\}^2} C^l_{\mathbf{j},\mathbf{j}_1}(\mathbf{x}^*) p^l_{\mathbf{j}_1}(\mathbf{x}^*), \tag{28}$$

then

$$\begin{aligned}
C^l_{\mathbf{j},\mathbf{j}+(0,0)}(\mathbf{x}^*) &= \left(\frac{x_1^* - x_1^{(i_1-r+j_1+l+1)}}{x_1^{(i_1-r+j_1)} - x_1^{(i_1-r+j_1+l+1)}}\right)\left(\frac{x_2^* - x_2^{(i_2-r+j_2+l+1)}}{x_2^{(i_2-r+j_2)} - x_2^{(i_2-r+j_2+l+1)}}\right) = C^l_{j_1,j_1}(x_1^*)C^l_{j_2,j_2}(x_2^*),\\
C^l_{\mathbf{j},\mathbf{j}+(1,0)}(\mathbf{x}^*) &= \left(\frac{x_1^{(i_1-r+j_1)} - x_1^*}{x_1^{(i_1-r+j_1)} - x_1^{(i_1-r+j_1+l+1)}}\right)\left(\frac{x_2^* - x_2^{(i_2-r+j_2+l+1)}}{x_2^{(i_2-r+j_2)} - x_2^{(i_2-r+j_2+l+1)}}\right) = C^l_{j_1,j_1+1}(x_1^*)C^l_{j_2,j_2}(x_2^*),\\
C^l_{\mathbf{j},\mathbf{j}+(0,1)}(\mathbf{x}^*) &= \left(\frac{x_1^* - x_1^{(i_1-r+j_1+l+1)}}{x_1^{(i-r+j_1)} - x_1^{(i_1-r+j_1+l+1)}}\right)\left(\frac{x_2^{(i_2-r+j_2)} - x_2^*}{x_2^{(i_2-r+j_2)} - x_2^{(i_2-r+j_2+l+1)}}\right) = C^l_{j_1,j_1}(x_1^*)C^l_{j_2,j_2+1}(x_2^*),\\
C^l_{\mathbf{j},\mathbf{j}+(1,1)}(\mathbf{x}^*) &= \left(\frac{x_1^{(i-r+j_1)} - x_1^*}{x_1^{(i-r+j_1)} - x_1^{(i_1-r+j_1+l+1)}}\right)\left(\frac{x_2^{(i_2-r+j_2)} - x_2^*}{x_2^{(i_2-r+j_2)} - x_2^{(i_2-r+j_2+l+1)}}\right) = C^l_{j_1,j_1+1}(x_1^*)C^l_{j_2,j_2+1}(x_2^*).
\end{aligned}$$

Proof. The proof is direct, considering that the stencils used to obtain each interpolant are $\{x_1^{(i_1-r+j_1)}, \dots, x_1^{(i_1-r+j_1+l+1)}\} \times \{x_2^{(i_2-r+j_2)}, \dots, x_2^{(i_2-r+j_2+l+1)}\}$ for $p^{l+1}_{\mathbf{j}}$, and

$$\{x_1^{(i_1-r+(j_1)_1)}, \dots, x_1^{(i_1-r+(j_1)_1+l)}\} \times \{x_2^{(i_2-r+(j_2)_1)}, \dots, x_2^{(i_2-r+(j_2)_1+l)}\}$$

for $p^l_{\mathbf{j}_1}$, with $\mathbf{j}_1 \in \mathbf{j}+\{0,1\}^2$, and the Aitken-Neville formula, see [11]. ■

Note that if our data are equally spaced and we approximate at the mid-points, we recover the result showed in Cor. 2.2.

Corollary 4.2. *Let $r \leq l \leq 2r-2$, and $0 \leq l_1, l_2 \leq (2r-2)-l$, and $x_1^{(i_1)} = a + k_1 h_1$, $h_1 = (b_1 - a_1)/J_1$, $k_1 = 0, \dots, J_1$, and $x_2^{(k_2)} = a_2 + k_2 h_2$, $h_2 = (b_2 - a_2)/J_2$, $k_2 = 0, \dots, J_2$, if we denote as $\mathbf{x}_{\frac{1}{2}} = (x_1^{(i_1-1/2)}, x_2^{(i_2-1/2)})$, and $C^l_{\mathbf{j},\mathbf{j}_1}(\mathbf{x}_{\frac{1}{2}})$ the values which satisfy:*

$$p^{l+1}_{\mathbf{j}}(\mathbf{x}_{\frac{1}{2}}) = \sum_{\mathbf{j}_1\in\mathbf{j}+\{0,1\}^2} C^l_{\mathbf{j},\mathbf{j}_1}(\mathbf{x}_{\frac{1}{2}}) p^l_{\mathbf{j}_1}(\mathbf{x}_{\frac{1}{2}}), \tag{29}$$

then, if we denote as $C^l_{\mathbf{j},\mathbf{j}_1} := C^l_{\mathbf{j},\mathbf{j}_1}(\mathbf{x}_{\frac{1}{2}})$, we get

$$C^l_{\mathbf{j},\mathbf{j}} = C^l_{j_1,j_1}C^l_{j_2,j_2}, \quad C^l_{\mathbf{j},\mathbf{j}+(1,0)} = C^l_{j_1,j_1+1}C^l_{j_2,j_2}, \quad C^l_{\mathbf{j}+(0,1)} = C^l_{j_1,j_1}C^l_{j_2,j_2+1}, \quad C^l_{\mathbf{j}+(1,1)} = C^l_{j_1,j_1+1}C^l_{j_2,j_2+1},$$

being $C^l_{j_i,j_i}$, and $C^l_{j_i,j_i+1}$, $i = 1,2$ the weights defined in Eq. (9).

Proof. It is clear from Lemma 4.1. ■

Now, we determine the iterative process as Eq. (10)

$$\begin{aligned}
\tilde{p}^r_{\mathbf{j}_{r-2}}(\mathbf{x}^*) &= p^r_{\mathbf{j}_{r-2}}(\mathbf{x}^*), \quad \mathbf{j}_{r-2} = \{0, \dots, r-1\}^2,\\
\tilde{p}^{l+1}_{\mathbf{j}_{2r-l-3}}(\mathbf{x}^*) &= \sum_{\mathbf{j}_{2r-l-2}\in\mathbf{j}_{2r-l-3}+\{0,1\}^2} \tilde{\omega}^l_{\mathbf{j}_{2r-l-3},\mathbf{j}_{2r-l-2}}(x^*)\tilde{p}^l_{\mathbf{j}_{2r-l-2}}(x^*), \quad l = r, \dots, 2r-2, \ \mathbf{j}_{2r-l-3} \in \{0, \dots, 2r-2-l\}^2,
\end{aligned}$$

for $l=r,\dots,2r-2$, and $\mathbf{k}\in\{0,\dots,2r-2-l\}^2$, and $\mathbf{k}_1\in\mathbf{k}+\{0,1\}^2$:

$$\begin{aligned}\tilde{\omega}^l_{\mathbf{k},\mathbf{k}_1}(\mathbf{x}^*)&=\left(\frac{\tilde{\alpha}^l_{\mathbf{k},\mathbf{k}_1}}{\tilde{\alpha}^l_{\mathbf{k},\mathbf{k}}+\tilde{\alpha}^l_{\mathbf{k},\mathbf{k}+(1,0)}+\tilde{\alpha}^l_{\mathbf{k},\mathbf{k}+(0,1)}+\tilde{\alpha}^l_{\mathbf{k},\mathbf{k}+(1,1)}}\right)(\mathbf{x}^*),\\ \tilde{\alpha}^l_{\mathbf{k},\mathbf{k}_1}(\mathbf{x}^*)&=\frac{C^l_{\mathbf{k},\mathbf{k}_1}(\mathbf{x}^*)}{(\epsilon+I^l_{\mathbf{k},\mathbf{k}_1})^t},\quad \mathbf{k}_1\in\mathbf{k}+\{0,1\}^2,\end{aligned}\tag{30}$$

where $I^l_{\mathbf{k},\mathbf{k}_1}$ are the smoothness indicators determined by the following formula:

$$I^l_{\mathbf{k},\mathbf{k}_1}=\begin{cases} I^r_{\mathbf{k}}, & \text{if } \mathbf{k}_1=\mathbf{k},\\ I^r_{\mathbf{k}+(l-(r-1),0)}, & \text{if } \mathbf{k}_1=\mathbf{k}+(1,0),\\ I^r_{\mathbf{k}+(0,l-(r-1))}, & \text{if } \mathbf{k}_1=\mathbf{k}+(0,1),\\ I^r_{\mathbf{k}+(l-(r-1),l-(r-1))}, & \text{if } \mathbf{k}_1=\mathbf{k}+(1,1),\end{cases}\tag{31}$$

being $I^r_{\mathbf{k}}$, with $0\le k_1,k_2\le r-1$, smoothness indicators satisfying the properties **P1**, **P2**, and **P3**.

Finally, the new bivariate progressive WENO approximation is

$$\tilde{\mathcal{I}}^{2r-1}(\mathbf{x}^*;f)=\tilde{p}^{2r-1}_{(0,0)}(\mathbf{x}^*).$$

5. A new progressive multivariate WENO method

In this section we generalize the method designed in Section 2 for $1d$, and in Section 4.3 for $2d$. We follow the same steps: First of all, we consider the data as in Section 3.1, i.e., if $\prod_{j=1}^n[a_j,b_j]$ is an hypercube and

$$a_j=x_j^{(0)}<x_j^{(1)}<x_j^{(2)}<\dots<x_j^{(J_j)}=b_j,\quad j=1,\dots,n,$$

are the points of a non-regular grid, we suppose our data as the evaluation of a unknown function at these points

$$f_{(l_1,\dots,l_n)}=f(x_1^{(l_1)},x_2^{(l_2)},\dots,x_n^{(l_n)}),\quad 0\le l_j\le J_j,\quad 1\le j\le n.$$

Let $(i_1,\dots,i_n)\in\mathbb{N}^n$ and $r\in\mathbb{N}$ be such that $0\le i_j-r\le i_j+r-1\le J_j$, $j=1,\dots,n$, a point $\mathbf{x}^*\in\prod_{j=1}^n[x_j^{(i_j-1)},x_j^{(i_j)}]$, and the centered stencil:

$$\mathcal{S}^{2r}_{\mathbf{0}}=\prod_{j=1}^n(\mathcal{S}^{2r}_0)_j,\tag{32}$$

with $\mathbf{0}=(0,\dots,0)$. We compute the interpolatory polynomial of degree $2r-1$ with nodes $\mathcal{S}^{2r}_{\mathbf{0}}$, $p^{2r-1}_{\mathbf{0}}(\mathbf{x}^*)$, and we express it as the combination of 2^n polynomials of degree $2r-2$. Thus, we get

$$p^{2r-1}_{\mathbf{0}}(\mathbf{x}^*)=\sum_{\mathbf{j}_0\in\{0,1\}^n}C^{2r-2}_{\mathbf{0},\mathbf{j}_0}(\mathbf{x}^*)p^{2r-2}_{\mathbf{j}_0}(\mathbf{x}^*).\tag{33}$$

Now, we repeat this process up to polynomials of degree r (for simplicity remove the dependence of $\mathbf{x}^*$), we denote as $\Omega_l=\mathbf{j}_l+\{0,1\}^n$:

$$\begin{aligned}p^{2r-1}_{\mathbf{0}}&=\sum_{\mathbf{j}_0\in\{0,1\}^n}C^{2r-2}_{\mathbf{0},\mathbf{j}_0}p^{2r-2}_{\mathbf{j}_0}\\ &=\sum_{\mathbf{j}_0\in\{0,1\}^n}C^{2r-2}_{\mathbf{0},\mathbf{j}_0}\left(\sum_{\mathbf{j}_1\in\Omega_0}C^{2r-3}_{\mathbf{j}_0,\mathbf{j}_1}p^{2r-3}_{\mathbf{j}_1}\right)\\ &=\sum_{\mathbf{j}_0\in\{0,1\}^n}C^{2r-2}_{\mathbf{0},\mathbf{j}_0}\left(\sum_{\mathbf{j}_1\in\Omega_0}C^{2r-3}_{\mathbf{j}_0,\mathbf{j}_1}\left(\sum_{\mathbf{j}_2\in\Omega_1}C^{2r-4}_{\mathbf{j}_1,\mathbf{j}_2}p^{2r-4}_{\mathbf{j}_2}\right)\right)\\ &=\sum_{\mathbf{j}_0\in\{0,1\}^n}C^{2r-2}_{\mathbf{0},\mathbf{j}_0}\left(\sum_{\mathbf{j}_1\in\Omega_0}C^{2r-3}_{\mathbf{j}_0,\mathbf{j}_1}\left(\cdots\left(\sum_{\mathbf{j}_{r-2}\in\Omega_{r-3}}C^{r}_{\mathbf{j}_{r-3},\mathbf{j}_{r-2}}p^{r}_{\mathbf{j}_{r-2}}\right)\cdots\right)\right),\end{aligned}\tag{34}$$

with

$$C^l_{\mathbf{j},\mathbf{j}+\{0,1\}^n}=\prod_{i=1}^n C^r_{j_i,j_i+\{0,1\}},$$

being $C^l_{j_i,j_i}$, and $C^l_{j_i,j_i+1}$, $i=1,\dots,n$ the weights defined in Eq. (9).

We establish the recursive process as Eq. (10):

$$\begin{aligned}
&\tilde{p}^r_{\mathbf{j}_{r-2}}(\mathbf{x}^*)=p^r_{\mathbf{j}_{r-2}}(\mathbf{x}^*), \quad \mathbf{j}_{r-2}=\{0,\dots,r-1\}^n,\\
&\tilde{p}^{l+1}_{\mathbf{j}_{2r-l-3}}(\mathbf{x}^*)=\sum_{\mathbf{j}_{2r-l-2}\in\mathbf{j}_{2r-l-3}+\{0,1\}^n}\tilde{\omega}^l_{\mathbf{j}_{2r-l-3},\mathbf{j}_{2r-l-2}}(x^*)\tilde{p}^l_{\mathbf{j}_{2r-l-2}}(x^*), \quad l=r,\dots,2r-2,\ \mathbf{j}_{2r-l-3}\in\{0,\dots,2r-2-l\}^n,
\end{aligned}$$

being for $l=r,\dots,2r-2$, and $\mathbf{k}\in\{0,\dots,2r-2-l\}$, $\mathbf{k}_1\in\mathbf{k}+\{0,1\}^n$:

$$\tilde{\omega}^l_{\mathbf{k},\mathbf{k}_1}(\mathbf{x}^*)=\frac{\tilde{\alpha}^l_{\mathbf{k},\mathbf{k}_1}(\mathbf{x}^*)}{\sum_{\mathbf{l}\in\{0,1\}^n}\tilde{\alpha}^l_{\mathbf{k},\mathbf{k}+\mathbf{l}}(\mathbf{x}^*)}, \quad \tilde{\alpha}^l_{\mathbf{k},\mathbf{k}_1}(\mathbf{x}^*)=\frac{C^l_{\mathbf{k},\mathbf{k}_1}(\mathbf{x}^*)}{(\epsilon+I^l_{\mathbf{k},\mathbf{k}_1})^t}, \quad \mathbf{k}_1\in\mathbf{k}+\{0,1\}^n, \tag{35}$$

where $I^l_{\mathbf{k},\mathbf{k}_1}$ are the smoothness indicators determined by the following formula:

$$I^l_{\mathbf{k},\mathbf{k}_1}=I^r_{\mathbf{k}+(l-(r-1))\mathbf{v}}, \quad \text{if} \quad \mathbf{k}_1=\mathbf{k}+\mathbf{v}, \quad \text{with} \quad \mathbf{v}\in\{0,1\}^n, \tag{36}$$

being $I^r_{\mathbf{k}}$, with $0\leq k_j\leq r-1$, $j=1,\dots,n$, smoothness indicators satisfying the properties **P1**, **P2**, and **P3**.

Therefore, the approximation will be:

$$\tilde{\mathcal{I}}^{2r-1}(\mathbf{x}^*)=\tilde{p}^{2r-1}_{\mathbf{0}}(\mathbf{x}^*). \tag{37}$$

Finally, we calculate the order of accuracy stating the next theorem.

Theorem 5.1. *Let $\mathbf{l}_0\in\{1,\dots,r-1\}^n$, $l_0=\min_{j=1,\dots,n}(\mathbf{l}_0)_j$, $\mathbf{x}^*\in\prod_{j=1}^n[x_j^{(i_j-1)},x_j^{(i_j)}]$, and $\tilde{\mathcal{I}}^{2r-1}(\mathbf{x}^*)$ the approximation defined in Eq. (37). If f is smooth in $\prod_{j=1}^n[x_j^{(i_j-r)},x_j^{(i_j+r-1)}]\setminus\Omega$, and f has a discontinuity at Ω then*

$$\tilde{\mathcal{I}}^{2r-1}(\mathbf{x}^*)-f(\mathbf{x}^*)=\begin{cases} O(h^{2r}), & \text{if } \Omega=\emptyset;\\ O(h^{r+l_0}), & \text{if } \Omega=\prod_{j=1}^n[x_j^{(i_j+(l_0)_j-1)},x_j^{(i_j+(l_0)_j)}].\end{cases} \tag{38}$$

6. Smoothness indicators

In this section, we present some smoothness indicators which satisfy the above mentioned properties. We generalize the smoothness indicators introduced by Aràndiga et al. in [7] which are an adaptation of the ones presented in [5]. The idea is to design some functionals that fulfill **P1**, **P2**, and **P3**.

Given $\mathbf{x}^*\in\mathbb{R}^n$, $h_1,\dots,h_n>0$, $\Phi(\mathbf{s})=\mathbf{x}^*+(s_1h_1,\dots,s_nh_n)$, f sufficiently smooth

$$\begin{aligned}
\int_{\Phi([0,1]^n)}(f\circ\Phi^{-1})^{(\boldsymbol{l})}(\mathbf{x})^2d\mathbf{x}&=\int_{\Phi([0,1]^n)}\left(f^{(\boldsymbol{l})}(\Phi^{-1}(\mathbf{x}))h_1^{-l_1}\dots h_n^{-l_n}\right)^2d\mathbf{x}\\
&=h_1^{-2l_1}\dots h_n^{-2l_n}\int_{\Phi([0,1]^n)}f^{(\boldsymbol{l})}(\Phi^{-1}(\mathbf{x}))d\mathbf{x}=h_1^{-2l_1}\dots h_n^{-2l_n}\int_{[0,1]^n}f^{(\boldsymbol{l})}(\mathbf{s})h_1\dots h_nd\mathbf{s}\\
&=h_1^{-2l_1+1}\dots h_n^{-2l_n+1}\int_{[0,1]^n}f^{(\boldsymbol{l})}(\mathbf{s})d\mathbf{s}.
\end{aligned}$$

Hence, we define

$$I_\Gamma(f)=\sum_{\mathbf{l}\in\mathcal{J}}h_1^{2l_1-1}\dots h_n^{2l_n-1}\int_\Gamma(f^{\mathbf{l})}(\mathbf{x}))^2d\mathbf{x}, \tag{39}$$

where $\Gamma=\prod_{j=1}^n[x_j^{(i_j-1)},x_j^{(i_j)}]$, $\mathcal{J}=\{0,1,\dots,r\}^n\setminus\{\mathbf{0}\}$ and smoothness indicators

$$I^r_{\mathbf{k}}=I_\Gamma(p^r_{\mathbf{k}}), \quad \mathbf{k}\in\{0,1,\dots,r-1\}^n, \tag{40}$$

to get an expression which is scale independent.

Theorem 6.1. *The smoothness indicators defined in Eq. (39-40) satisfy properties **P1**, **P2**, and **P3**.*

Proof. Since $|\Gamma| = h_1 \dots h_n$, it is clear that if f is smooth in Γ, then

$$I_\Gamma(f) = O(h^2).$$

We suppose that p is a polynomial interpolant at a stencil of $(r+1)^n$ points, then

$$\begin{aligned} I_\Gamma(f) - I_\Gamma(p) &= \sum_{\mathbf{l}\in\mathcal{J}} h_1^{2l_1-1}\dots h_n^{2l_n-1}\int_\Gamma ((f^{\mathbf{l})}(\mathbf{x}))^2 - (p^{\mathbf{l})}(\mathbf{x}))^2)d\mathbf{x} \\ &= \sum_{\mathbf{l}\in\mathcal{J}} h_1^{2l_1-1}\dots h_n^{2l_n-1}\int_\Gamma E^{\mathbf{l})}(\mathbf{x})(2f^{\mathbf{l})}(\mathbf{x}) - E^{\mathbf{l})}(\mathbf{x}))d\mathbf{x} \\ &= \sum_{\mathbf{l}\in\mathcal{J}} h_1^{2l_1}\dots h_n^{2l_n} O(h^{r+1-||\mathbf{l}||_\infty}) \\ &= O(h^{r+1}). \end{aligned} \tag{41}$$

where E is defined in Eq. (18), and $||\mathbf{l}||_\infty = \max_{j=1,\dots,n}|l_j|$. Then, if $\mathbf{k},\mathbf{k}' \in \{0,\dots,r-1\}^n$, we get:

$$I_\Gamma(p^r_{\mathbf{k}}) - I_\Gamma(p^r_{\mathbf{k}'}) = I_\Gamma(p^r_{\mathbf{k}}) - I_\Gamma(f) + I_\Gamma(f) - I_\Gamma(p^r_{\mathbf{k}'}) = O(h^{r+1}).$$

Finally, if a discontinuity crosses the stencil $\mathcal{S}^r_{\mathbf{k}}$ then:

$$I^r_{\mathbf{k}} \nrightarrow 0 \text{ as } h \to 0,$$

since some of the quadratic terms of $I^r_{\mathbf{k}}$ will not converge to 0. ■

Remark 6.1. *An evaluation of these smoothness indicators for* $2d$ *in gridded data is shown in [7]. Also, some adaptation to smoothness indicators with better capabilities, and computationally more efficient than those introduced in [1], or in [8] can be performed, but the process requires an study on the order of accuracy, which exceeds the scope of this paper.*

7. Numerical experiments

In this section, we check our theoretical results through some numerical examples. We divide it into four parts, starting with $1d$ experiments for uniform and non-uniform grids and performing some tests for the multivariable case. In both cases, the order of accuracy is analyzed measuring the error in a set of points with a determined grid and refining it. For this purpose we define the error in a finite set $\Upsilon \subset \mathbb{R}^n$ as:

$$E(\Upsilon) = \max_{\mathbf{x}^*\in\Upsilon} |f(\mathbf{x}^*) - \tilde{\mathcal{I}}^{2r-1}(\mathbf{x}^*)|. \tag{42}$$

Secondly, in all cases, we study the behavior of the resulting interpolator in the zones close to the discontinuities and conclude that by employing our new non-linear algorithm, if the discontinuity is isolated, the Gibbs phenomenon is avoided.

7.1. Examples in 1D for uniform grids

We perform the first experiment using the same function studied in [3]:

$$f_1(x) = \begin{cases} x^{10} - x^9 + x^8 - 4x^7 + x^6 + x^5 + x^4 + x^3 + 5x^2 + 3x, & -\frac{\pi}{6} \le x < 0, \\ 1 - (x^{10} - 2x^9 + 3x^8 - 8x^7 - 2x^6 + x^5 - 2x^4 - 3x^3 - 5x^2 + 0.5x), & 0 \le x < 1 - \frac{\pi}{6}, \end{cases} \tag{43}$$

discretizing it with $N_\ell = 2^\ell + 1$ points in the interval $[-\frac{\pi}{6}, 1-\frac{\pi}{6}]$. To analyze the order close to the discontinuity we locate the point 0 in each level ℓ, compute the errors in the adjacent intervals, and estimate the numerical order of accuracy. Therefore, let $j_0^\ell \in \mathbb{N}$ such that $0 \in [x_{j_0^\ell - 1}, x_{j_0^\ell}]$, then we interpolate the value of the function f at 10000 points equally spaced in the interval $[x_{j_0^\ell-5}, x_{j_0^\ell+4}]$, denoted by $\mathcal{J}^\ell$, determine the error using Eq. (42), $E_s^\ell = E(\mathcal{J}^\ell \cap [x_{j_0^\ell+s-1}, x_{j_0^\ell+s}])$ and approximate the numerical order in each interval s as:

$$o_s^\ell = \log_2\left(\frac{E_s^{\ell-1}}{E_s^\ell}\right), \quad s = -4, \ldots, 4. \tag{44}$$

This allows us to analyze the progressively increasing order in the neighboring intervals to the one containing the discontinuity. For $r = 3$, we can see in Table 1 that the order of accuracy increases as we proved in Theorem 5.1. Also, for $r = 4$, Table 3, we observe the same behavior of the numerical order. However, when we apply the classical WENO algorithm, in both cases $r = 3, 4$, Tables 2 and 4, the numerical order is reduced to $r + 1$, in all the intervals where one of the small stencils is contaminated by the singularity. Note that when we use our algorithm, we interpolate in any point of the interval, but when using the classical WENO method, we can only interpolate at the mid-point.

ℓ	E_{-4}^ℓ	o_{-4}^ℓ	E_{-3}^ℓ	o_{-3}^ℓ	E_{-2}^ℓ	o_{-2}^ℓ	E_{-1}^ℓ	o_{-1}^ℓ	E_0^ℓ	o_0^ℓ	E_1^ℓ	o_1^ℓ	E_2^ℓ	o_2^ℓ	E_3^ℓ	o_3^ℓ	E_4^ℓ	o_4^ℓ
5	1.6751e-08		1.3389e-08		5.7904e-09		7.3537e-07		7.3774e-01		1.7858e-06		3.0257e-08		2.4029e-08		2.6927e-08	
6	1.4582e-10	6.84	1.2149e-10	6.78	8.5220e-10	2.76	5.1819e-08	3.82	5.1486e-01	0.51	1.1284e-07	3.98	3.5988e-10	6.39	3.4480e-10	6.12	3.3124e-10	6.34
7	1.4314e-12	6.67	1.2491e-12	6.60	3.6705e-11	4.53	3.5352e-09	3.87	9.8435e-01	-0.93	7.1647e-09	3.97	2.6416e-11	3.76	3.0225e-12	6.83	3.6214e-12	6.51
8	1.7396e-14	6.36	1.5998e-14	6.28	1.2246e-12	4.90	2.2682e-10	3.96	9.6727e-01	0.02	4.5610e-10	3.97	1.0890e-12	4.60	3.4750e-14	**6.44**	3.7415e-14	**6.59**
9	2.3939e-16	**6.18**	2.2551e-16	**6.14**	3.9328e-14	**4.96**	1.4359e-11	**3.98**	9.3163e-01	0.05	2.8800e-11	**3.98**	3.7748e-14	**4.85**	7.7716e-16	5.48	8.8818e-16	5.39

Table 1: Grid refinement analysis for the **new** WENO-6 algorithm for the function in (43).

ℓ	E_{-4}^ℓ	o_{-4}^ℓ	E_{-3}^ℓ	o_{-3}^ℓ	E_{-2}^ℓ	o_{-2}^ℓ	E_{-1}^ℓ	o_{-1}^ℓ	E_0^ℓ	o_0^ℓ	E_1^ℓ	o_1^ℓ	E_2^ℓ	o_2^ℓ	E_3^ℓ	o_3^ℓ	E_4^ℓ	o_4^ℓ
5	1.6749e-08		1.3388e-08		1.5644e-07		6.7525e-07		4.7375e-01		1.6736e-06		3.6670e-07		2.4040e-08		2.6932e-08	
6	1.4582e-10	6.84	1.2148e-10	6.78	1.1614e-08	3.75	4.8161e-08	3.80	4.7959e-01	-0.01	1.0573e-07	3.98	2.4504e-08	3.90	3.4479e-10	6.12	3.3126e-10	6.34
7	1.4314e-12	6.67	1.2491e-12	6.60	7.8165e-10	3.89	3.3045e-09	3.86	5.0394e-01	-0.07	6.7145e-09	3.97	1.5610e-09	3.97	3.0224e-12	6.83	3.6212e-12	6.51
8	1.7396e-14	6.36	1.5987e-14	6.28	4.9619e-11	3.97	2.1244e-10	3.95	5.0215e-01	0.00	4.2750e-10	3.97	9.9145e-11	3.97	3.4639e-14	6.44	3.7192e-14	6.60
9	2.3592e-16	**6.20**	2.2204e-16	**6.16**	3.1239e-12	**3.98**	1.3456e-11	**3.98**	5.0123e-01	0.00	2.6997e-11	**3.98**	6.2474e-12	**3.98**	3.3307e-16	**6.70**	5.5511e-16	**6.06**

Table 2: Grid refinement analysis for the **classical** WENO-6 algorithm for the function in (43).

ℓ	E_{-4}^ℓ	o_{-4}^ℓ	E_{-3}^ℓ	o_{-3}^ℓ	E_{-2}^ℓ	o_{-2}^ℓ	E_{-1}^ℓ	o_{-1}^ℓ	E_0^ℓ	o_0^ℓ	E_1^ℓ	o_1^ℓ	E_2^ℓ	o_2^ℓ	E_3^ℓ	o_3^ℓ	E_4^ℓ	o_4^ℓ
5	2.2152e-11		1.6644e-09		1.8454e-08		3.5396e-09		7.0227e-01		4.1523e-08		2.8408e-08		2.2035e-09		8.4167e-11	
6	9.4535e-14	**7.87**	1.2205e-11	7.09	1.6904e-10	6.77	2.2973e-09	0.62	5.3587e-01	0.39	1.2832e-09	5.01	3.1781e-10	6.48	1.9403e-11	6.82	3.3018e-13	**7.99**
7	4.9960e-16	7.56	9.1455e-14	7.06	1.7535e-12	6.59	9.5206e-11	4.59	9.8422e-01	-0.87	7.0822e-11	4.17	3.9031e-12	6.34	1.6120e-13	**6.91**	1.7764e-15	7.53
8	2.0817e-17	4.58	7.0777e-16	**7.01**	2.2673e-14	6.27	3.1514e-12	4.91	9.6632e-01	0.02	2.8244e-12	4.64	4.8961e-14	6.31	1.7764e-15	6.50	5.5511e-16	1.67
9	1.3878e-17	0.58	1.0408e-17	6.08	3.2092e-16	**6.14**	1.0091e-13	**4.96**	9.2930e-01	0.05	9.6589e-14	**4.86**	8.8818e-16	**5.78**	4.4409e-16	2.00	4.4409e-16	0.32

Table 3: Grid refinement analysis for the **new** WENO-8 algorithm for the function in (43).

ℓ	E_{-4}^ℓ	o_{-4}^ℓ	E_{-3}^ℓ	o_{-3}^ℓ	E_{-2}^ℓ	o_{-2}^ℓ	E_{-1}^ℓ	o_{-1}^ℓ	E_0^ℓ	o_0^ℓ	E_1^ℓ	o_1^ℓ	E_2^ℓ	o_2^ℓ	E_3^ℓ	o_3^ℓ	E_4^ℓ	o_4^ℓ
5	2.2458e-11		1.6502e-09		4.1669e-09		3.2826e-09		4.5560e-01		3.9398e-08		2.2108e-09		4.0566e-10		8.1793e-11	
6	9.4508e-14	**7.89**	1.5559e-10	3.40	5.6031e-10	2.89	2.0622e-09	0.67	4.7046e-01	-0.04	1.1327e-09	5.12	3.6677e-10	2.59	1.1083e-10	1.87	3.2652e-13	**7.96**
7	4.5797e-16	7.68	5.9243e-12	4.71	2.1924e-11	4.67	8.5879e-11	4.58	5.0851e-01	-0.11	6.3658e-11	4.15	1.6916e-11	4.43	4.7008e-12	4.55	1.5543e-15	7.71
8	0	-	1.9244e-13	4.94	7.1713e-13	4.93	2.8456e-12	4.91	5.0443e-01	0.01	2.5479e-12	4.64	6.4937e-13	4.70	1.7553e-13	4.74	0	-
9	3.4694e-18	-	6.1149e-15	**4.97**	2.2865e-14	**4.97**	9.1148e-14	**4.96**	5.0237e-01	0.00	8.7264e-14	**4.86**	2.1871e-14	**4.89**	5.6621e-15	**4.95**	2.2204e-16	-

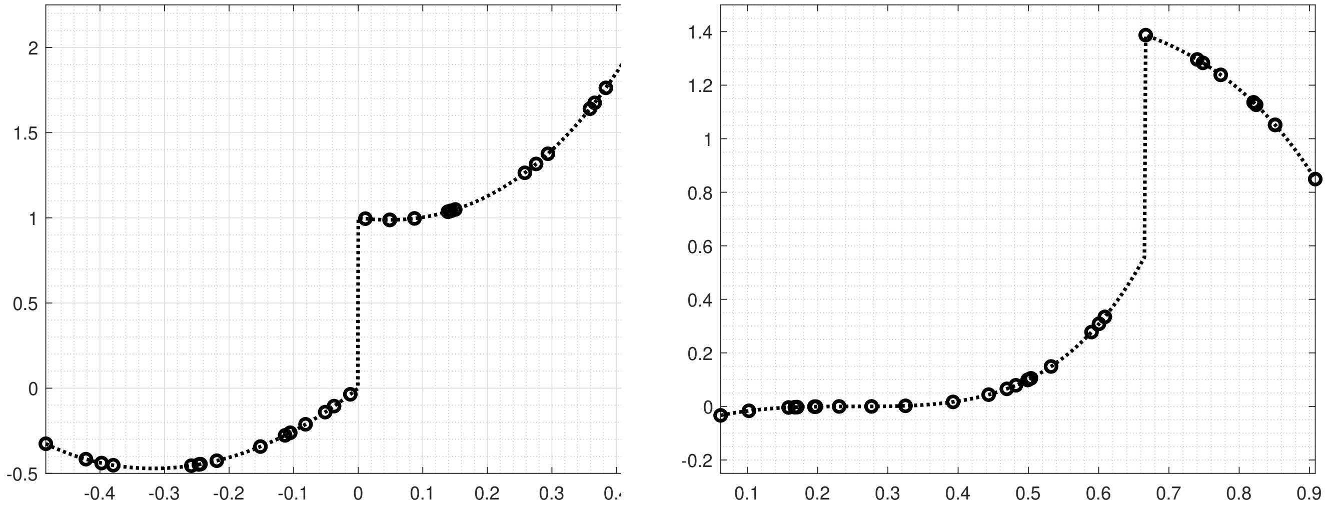
Table 4: Grid refinement analysis for the **classical** WENO-8 algorithm for the function in (43).

7.2. Examples in 1D for non-uniform grids

In this subsection, we perform two experiments: one with the function defined in (43) and, also, with the function studied in [6]:

$$f_2(x) = \begin{cases} 5(x-0.25)^3 e^{x^2}, & 0 \le x < 2/3, \\ 1.5 - (x-0.25)^3 e^{x^2}, & 2/3 \le x < 1. \end{cases} \tag{45}$$

We construct a non-uniform grid $\{x_i\}_{i=0}^{32}$ with $x_i \sim U[0,1]$, being $U[0,1]$ a uniform distribution in $[0,1]$ and interpolate the values of the function in a uniform grid of 10000 points. The result is showed in Figure 5, we can see that the two interpolants avoid Gibbs-phenomenon. We compare our method with classical WENO adapted to non-uniform grids, we change the optimal weights in (2) using the formulas presented in Lemma 2.1. In Tables 6 and 8 we can observe similar results to those obtained for uniform grids: the order of accuracy in the adjacent cells to the isolated discontinuity, o^l_2 and o^l_{-2}, is smaller than the one obtained using the new WENO algorithm. These numerical results are consistent with the theoretical



(a) (b)

Figure 5: Interpolation using the new WENO-6 algorithm. Dashed line: interpolation of the function (43) (a) and of the function (45) (b). Circles: Data points.

To analyze the numerical order we take the non-uniform mesh, $\{x^5_i\}_{i=0}^{32}$, we divide it in each level using the following formula:

$$\begin{cases} x^{\ell+1}_{2j+1} = \frac{1}{2}(x^\ell_j + x^\ell_{j+1}), & j = 0, \dots, 2^\ell - 1, \\ x^{\ell+1}_{2j} = x^\ell_j, & j = 0, \dots, 2^\ell, \end{cases} \tag{46}$$

for $\ell \geq 5$ and compute the numerical order following Eq. (44). Again, we see, in Tables 5 and 7, that the behaviour of the numerical order is the expected one, and proved in Theorem 5.1.

ℓ	E^ℓ_{-4}	o^ℓ_{-4}	E^ℓ_{-3}	o^ℓ_{-3}	E^ℓ_{-2}	o^ℓ_{-2}	E^ℓ_{-1}	o^ℓ_{-1}	E^ℓ_0	o^ℓ_0	E^ℓ_1	o^ℓ_1	E^ℓ_2	o^ℓ_2	E^ℓ_3	o^ℓ_3	E^ℓ_4	o^ℓ_4
5	1.4532e-09		1.9867e-09		1.3509e-09		2.1725e-07		5.0179e-01		9.6277e-07		5.7583e-12		1.8499e-08		2.0715e-08	
6	3.0627e-12	8.89	1.1074e-11	7.48	2.8323e-10	2.25	1.8743e-08	3.53	9.5557e-01	-0.92	1.4596e-07	2.72	1.0063e-10	-4.12	9.9920e-16	24.14	8.8818e-16	24.47
7	3.4114e-13	3.16	2.7956e-13	5.30	9.6570e-12	4.87	1.1295e-09	4.05	9.0939e-01	0.07	4.4933e-09	5.02	5.2145e-11	0.94	7.6797e-12	-12.90	6.9681e-12	-12.93
8	3.1225e-15	6.77	2.6784e-15	6.70	2.8241e-13	5.09	6.9824e-11	4.01	8.1464e-01	0.15	1.4067e-10	4.99	3.2563e-13	7.32	1.1768e-14	9.35	5.5622e-14	6.96
9	4.1633e-17	**6.22**	4.1633e-17	**6.00**	8.9997e-15	**4.97**	4.4054e-12	**3.98**	5.9349e-01	0.45	8.8614e-12	**3.98**	9.1038e-15	**5.16**	4.4409e-16	**4.72**	3.3307e-16	**7.38**

Table 5: Non-uniform grid refinement analysis for the **new** WENO-6 algorithm for the function (43)

ℓ	E^ℓ_{-4}	o^ℓ_{-4}	E^ℓ_{-3}	o^ℓ_{-3}	E^ℓ_{-2}	o^ℓ_{-2}	E^ℓ_{-1}	o^ℓ_{-1}	E^ℓ_0	o^ℓ_0	E^ℓ_1	o^ℓ_1	E^ℓ_2	o^ℓ_2	E^ℓ_3	o^ℓ_3	E^ℓ_4	o^ℓ_4
5	1.4532e-09		1.9867e-09		2.2394e-09		2.1596e-07		5.0556e-01		9.6277e-07		1.1280e-10		1.8422e-08		2.0870e-08	
6	3.0627e-12	8.89	1.1074e-11	7.48	4.4299e-09	-0.98	1.8643e-08	3.53	9.5952e-01	-0.92	1.4596e-07	2.72	9.9574e-09	-6.46	8.8818e-16	24.30	8.8818e-16	24.48
7	3.4114e-13	3.16	2.7956e-13	5.30	2.6139e-10	4.08	1.1280e-09	4.04	9.1492e-01	0.06	4.4933e-09	5.02	2.7581e-09	1.85	7.6801e-12	-13.07	6.9664e-12	-12.93
8	3.1225e-15	6.77	2.6784e-15	6.70	1.5284e-11	4.09	6.9803e-11	4.01	8.1613e-01	0.16	1.4067e-10	4.99	3.4826e-11	6.30	1.1879e-14	9.33	5.5511e-14	6.97
9	4.1633e-17	**6.22**	4.3368e-17	**5.94**	9.6033e-13	**3.99**	4.4051e-12	**3.98**	5.9392e-01	0.45	8.8614e-12	**3.98**	1.9277e-12	**4.17**	4.4409e-16	**4.74**	3.3307e-16	**7.38**

Table 6: Non-uniform grid refinement analysis for the **classical** WENO-6 algorithm for the function (43)

7.3. Examples in 2d for uniform grids

We start with a smooth function to check numerically that the order is $2r$. Thus, we consider the function

$$f_3(x_1, x_2) = \frac{1}{x_1^2 + x_2^2 + 1}, \tag{47}$$

ℓ	E^ℓ_{-4}	o^ℓ_{-4}	E^ℓ_{-3}	o^ℓ_{-3}	E^ℓ_{-2}	o^ℓ_{-2}	E^ℓ_{-1}	o^ℓ_{-1}	E^ℓ_{0}	o^ℓ_{0}	E^ℓ_{1}	o^ℓ_{1}	E^ℓ_{2}	o^ℓ_{2}	E^ℓ_{3}	o^ℓ_{3}	E^ℓ_{4}	o^ℓ_{4}
5	1.6775e-08		2.7033e-07		6.2666e-05		2.8007e-04		8.2027e-01		5.6347e-05		5.8642e-06		1.6321e-08		4.6059e-08	
6	3.5056e-12	12.22	2.1402e-12	16.94	1.9801e-06	4.98	3.5278e-06	6.31	8.1413e-01	0.01	1.0834e-05	2.37	1.3715e-07	5.41	1.3137e-11	10.27	4.4049e-12	13.35
7	2.1300e-13	4.04	1.4868e-10	-6.11	1.9147e-08	6.69	1.3796e-06	1.35	8.0577e-01	0.01	9.3678e-07	3.53	2.7173e-08	2.33	8.3370e-10	-5.98	1.8362e-10	-5.38
8	1.4874e-11	-6.12	1.5467e-11	3.26	9.9600e-10	4.26	9.4274e-08	3.87	7.9070e-01	0.02	5.5039e-08	4.08	7.7134e-10	5.13	1.6195e-11	5.68	1.7014e-11	3.43
9	2.5019e-13	**5.89**	2.5602e-13	**5.91**	3.2880e-11	**4.92**	6.1595e-09	**3.93**	7.5953e-01	0.05	3.3358e-09	**4.04**	2.3013e-11	**5.06**	2.3137e-13	**6.12**	2.3692e-13	**6.16**

Table 7: Non-uniform grid refinement analysis for the **new** WENO-6 algorithm for the function (45).

ℓ	E^ℓ_{-4}	o^ℓ_{-4}	E^ℓ_{-3}	o^ℓ_{-3}	E^ℓ_{-2}	o^ℓ_{-2}	E^ℓ_{-1}	o^ℓ_{-1}	E^ℓ_{0}	o^ℓ_{0}	E^ℓ_{1}	o^ℓ_{1}	E^ℓ_{2}	o^ℓ_{2}	E^ℓ_{3}	o^ℓ_{3}	E^ℓ_{4}	o^ℓ_{4}
5	1.6383e-08		2.8758e-07		1.0398e-04		3.9364e-04		8.1970e-01		5.6347e-05		1.8207e-05		1.6304e-08		4.7433e-08	
6	3.5061e-12	12.19	2.1401e-12	17.03	5.2570e-06	4.30	3.5206e-06	6.80	8.1338e-01	0.01	1.0834e-05	2.37	8.1099e-07	4.48	1.3140e-11	24.30	4.4047e-12	13.39
7	2.1300e-13	4.04	1.4842e-10	-6.11	2.3862e-07	4.46	1.3735e-06	1.35	8.0534e-01	0.01	9.3678e-07	3.53	1.9806e-07	2.03	8.3530e-10	-13.07	1.8435e-10	-5.38
8	1.4875e-11	-6.12	1.5467e-11	3.26	2.0923e-08	3.51	9.4127e-08	3.86	7.8938e-01	0.02	5.5039e-08	4.08	1.2003e-08	4.04	1.6196e-11	9.33	1.7015e-11	3.43
9	2.5013e-13	**5.89**	2.5613e-13	**5.91**	1.3520e-09	**3.95**	6.1566e-09	**3.93**	7.5647e-01	0.06	3.3358e-09	**4.04**	7.3780e-10	**4.02**	2.3115e-13	**4.74**	2.3714e-13	**6.16**

Table 8: Non-uniform grid refinement analysis for the **classical** WENO-6 algorithm for the function (45)

and interpolate it using a Cartesian grid in the square $[-1,1]^2$:

$$X^\ell = \{(x_1^{\ell,(i)}, x_2^{\ell,(j)})\}_{i,j=0}^{2^\ell}, \quad x_1^{\ell,(i)} = -1 + ih_\ell,\ x_2^{\ell,(j)} = -1 + jh_\ell,\ h_\ell = 2^{-\ell+1}, \tag{48}$$

at the points

$$B^\ell = \{(\tau(k_1)x_1 + (1-\tau(k_1))y_1, \tau(k_2)x_2 + (1-\tau(k_2))y_2) : (x_1,x_2),(y_1,y_2) \in X^\ell,\ \tau(k) = 0.3 + 0.1k,\ k = 0,4\},$$

applying the new WENO algorithm for $r = 3$. Note that the set of points B^ℓ is conformed by a convex combination of points of the grid. We have chosen this collection of points, but we could have selected any other. The result is shown in Fig. 6, and we observe that it is similar to the original function f_3. In order to compute the numerical order, we use the same strategy presented in Section 7.1, we calculate $E^\ell = E(B^\ell)$ and approximate the order as in Eq. (44). Regarding the table presented in Fig. 6 it is clear that the order of accuracy is the expected one.

	$r=3$		$r=4$	
ℓ	E^ℓ	o^ℓ	E^ℓ	o^ℓ
4	7.9879e-07		1.6882e-08	
5	1.3037e-08	5.94	6.4910e-11	8.02
6	2.0584e-10	5.98	2.5102e-13	**8.01**
7	3.2245e-12	**6.00**	1.4433e-15	7.44

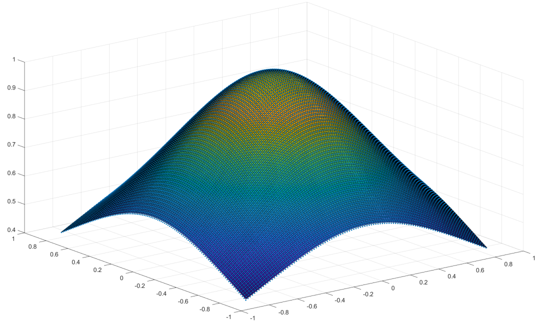


Figure 6: Left: Grid refinement analysis for the new WENO-6 algorithm for the function (47). Right: Interpolation using the new WENO-6 algorithm.

In order to analyze the order close to the discontinuities, we perform an experiment with the function:

$$f_3(x_1,x_2) = \begin{cases} e^{x_1+x_2}\cos(x_1-x_2), & x_1+x_2 \leq 0, \\ e^{x_1+x_2}\cos(x_1-x_2)+1, & x_1+x_2 > 0. \end{cases} \tag{49}$$

We take the mid-point $\mathbf{x}_0 = (x_1^{\ell,(2^\ell-1)}, x_2^{\ell,(2^\ell-1)}) = (0,0)$ of the cartesian grid X^ℓ; the set

$$Y^\ell = \{(s_1,s_2)h_\ell : -5 \leq s_1 \leq 6,\ -4 \leq s_2 \leq 5\},$$

(see Figure 7.(a) red big points), and calculate an approximation to the function in the points

$$\mathcal{X}^\ell = \{(\tau(k_1)x_1 + (1-\tau(k_1))y_1, \tau(k_2)x_2 + (1-\tau(k_2))y_2) : (x_1,x_2),(y_1,y_2) \in Y^\ell,\ \tau(k) = 0.3 + 0.1k,\ k = 0,\dots,4\},$$

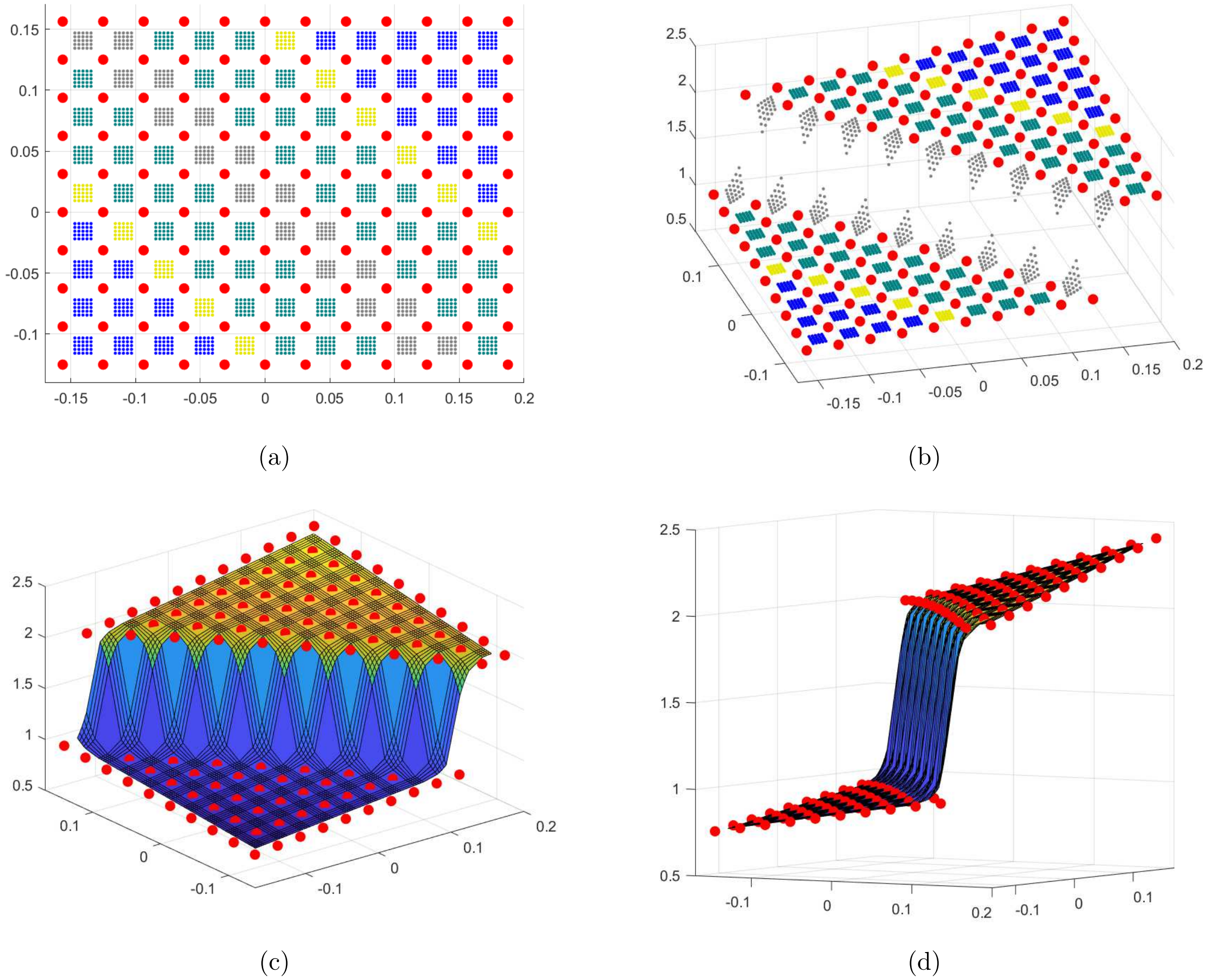
(see Figure 7.(a) gray, green, yellow and blue small points). We chose this set for the convenience of analyzing the order, but any other random sample of points could have been selected. A region between four data points is defined as:

$$\mathcal{X}^{\ell}_{(s_1,s_2)} = \mathcal{X}^{\ell} \cap [s_1 h_\ell, (s_1+1)h_\ell] \times [s_2 h_\ell, (s_2+1)h_\ell], \quad -5 \leq s_1 \leq 5,\, -4 \leq s_2 \leq 4,$$

and the errors and the numerical orders in each region as:

$$E^{\ell}_{(s_1,s_2)} = E(\mathcal{X}^{\ell}_{(s_1,s_2)}), \quad o^{\ell}_{(s_1,s_2)} = \log_2\left(\frac{E^{\ell-1}_{(s_1,s_2)}}{E^{\ell}_{(s_1,s_2)}}\right), \quad -5 \leq s_1 \leq 5,\, -4 \leq s_2 \leq 4.$$

Now, we compare with a modification of the classical WENO method adapted to work using tensor products. Note that the WENO-2d method defined in [7] is constructed for uniform-grids. In this case, we reformulate the classical WENO to approximate at any point in the considered interval and for non-uniform-grids. Theoretically, we have proved that the order of accuracy obtained at the green points in Figure 7.(a) is equal to 4, as there is one unique square stencil that is not contaminated by the discontinuity. For the yellow ones, 5 is the theoretical order of accuracy, and 6 for the rest of the points. Using the classical WENO algorithm, we only distinguish two zones, order 4 (yellow and green zones) and order 6 for the rest of the points.



(a)

(b)

(c)

(d)

Figure 7: Interpolation using new WENO-6 algorithm to the function (49). (a): Nodes (red points) and points of interpolation (color points). (b), (c) and (d): Result of the interpolation in these points.

We can check this fact in Tables 9 and 10, where we determine the order, $o^7_{(s_1,s_2)}$ in each region. In this case, in the smooth part, we obtain numerical order 7. The interpolation is shown in Figure 7.(c). We can see again that the resulting intepolator avoids Gibbs phenomenon (Figure 7.(d)).

$o^7_{(s_1,s_2)}$	$s_1=-5$	-4	-3	-2	-1	0	1	2	3	4	5
$s_2=-4$	-0.00	0.00	4.03	4.07	3.99	**5.10**	7.07	7.10	7.32	6.81	7.09
-3	3.93	-0.00	0.00	4.05	4.08	3.99	**5.11**	7.23	7.09	7.00	7.08
-2	3.90	3.94	-0.00	0.00	4.06	4.09	4.00	**5.11**	7.06	7.21	7.25
-1	3.99	3.92	3.95	-0.00	0.00	4.06	4.09	4.00	**5.11**	7.07	7.20
0	**4.90**	4.00	3.92	3.96	-0.00	0.00	4.06	4.09	3.99	**5.10**	7.09
1	6.89	**4.91**	4.01	3.93	3.96	-0.00	0.00	4.06	4.08	3.99	**5.09**
2	6.86	6.97	**4.91**	4.01	3.93	3.96	-0.00	0.00	4.05	4.07	3.97
3	6.88	6.93	6.96	**4.91**	4.01	3.92	3.95	-0.00	0.00	4.03	4.06
4	6.84	6.80	6.88	6.97	**4.91**	4.00	3.92	3.94	-0.00	0.00	4.02

Table 9: Grid refinement analysis for the **new** WENO-6 algorithm for the function (49) with $\ell = 7$

$o^7_{(s_1,s_2)}$	$s_1=-5$	-4	-3	-2	-1	0	1	2	3	4	5
$s_2=-4$	-0.00	0.00	4.03	4.07	4.04	**4.08**	7.15	7.10	7.15	6.98	7.09
-3	3.92	-0.00	0.00	4.04	4.08	4.05	**4.08**	7.06	6.89	7.00	6.98
-2	3.90	3.94	0.00	0.00	4.05	4.08	4.05	**4.09**	6.91	7.21	7.25
-1	3.94	3.91	3.95	-0.00	0.00	4.05	4.09	4.05	**4.08**	7.15	7.19
0	**3.91**	3.95	3.92	3.95	-0.00	0.00	4.05	4.08	4.05	**4.08**	7.14
1	6.89	**3.92**	3.96	3.92	3.96	-0.00	0.00	4.05	4.08	4.04	**4.07**
2	6.91	6.97	**3.92**	3.96	3.92	3.95	-0.00	0.00	4.04	4.07	4.02
3	6.92	6.93	6.96	**3.92**	3.96	3.92	3.95	-0.00	0.00	4.03	4.05
4	6.90	6.94	6.93	6.90	**3.92**	3.95	3.91	3.94	-0.00	0.00	4.01

Table 10: Grid refinement analysis for the **classical** WENO-6 algorithm for the function (49) with $\ell = 7$

7.4. Examples in 2d for non-uniform grids

Finally, in this subsection, we will present two examples to corroborate that our new algorithm is also valid for non-uniform grids. To study the order we design a non-regular grid for $\ell = 4$ applying the following formula

$$\tilde{X}^\ell = \{(\tilde{x}_1^{\ell,(i)}, \tilde{x}_2^{\ell,(j)})\}_{i,j=0}^{2^\ell}, \quad \tilde{x}_1^{\ell,(i)} = -1 + ih_\ell + \varepsilon_i^\ell,\ \tilde{x}_2^{\ell,(j)} = -1 + jh_\ell + \varepsilon_j^\ell,\ h_\ell = 2^{-\ell+1},\ \varepsilon_i^\ell, \varepsilon_j^\ell \sim U\left[-\frac{h_\ell}{2}, \frac{h_\ell}{2}\right] \tag{50}$$

being $U\left[-\frac{h_\ell}{2}, \frac{h_\ell}{2}\right]$ the uniform distribution in the interval $[-h_\ell/2, h_\ell/2]$. And for $\ell \geq 4$ we use the same strategy as in 1d, Eq. (46), i.e., we define $\tilde{X}^{\ell+1} = \{(\tilde{x}_1^{\ell+1,(i)}, \tilde{x}_2^{\ell+1,(j)})\}_{i,j=0}^{2^{\ell+1}}$ as

$$\tilde{x}_1^{\ell+1,(l_1)} = \begin{cases} \frac{1}{2}(\tilde{x}_1^{\ell,(i)} + \tilde{x}_1^{\ell,(i+1)}), & l_1 = 2i+1,\ 1 \leq i \leq 2^{\ell+1}-1, \\ \tilde{x}_1^{\ell,(i)}, & l_1 = 2i,\ 0 \leq i \leq 2^{\ell+1}, \end{cases}$$

$$\tilde{x}_2^{\ell+1,(l_2)} = \begin{cases} \frac{1}{2}(\tilde{x}_2^{\ell,(j)} + \tilde{x}_2^{\ell,(j+1)}), & l_2 = 2j+1,\ 0 \leq j \leq 2^{\ell+1}-1, \\ \tilde{x}_2^{\ell,(j)}, & l_2 = 2j,\ 0 \leq j \leq 2^{\ell+1}, \end{cases}$$

and interpolate at 25 points contained in each region formed by 4 points:

$$\tilde{\mathcal{X}}^\ell = \{(\tau(k_1)x_1 + (1-\tau(k_1))y_1, \tau(k_2)x_2 + (1-\tau(k_2))y_2) : (x_1,x_2),(y_1,y_2) \in \tilde{X}^\ell,\ \tau(k) = 0.3 + 0.1k,\ k = 0,1,2,3,4\}.$$

Then we calculate the errors and numerical orders in $\Omega_\ell = \mathcal{X}^\ell \cap [-5.5h_\ell, 5.5h_\ell]^2$, as in previous subsections, with $E^\ell = E(\Omega_\ell)$. We can observe again in Table 11, that the results obtained in the numerical experiments satisfy the theoretical ones. To finish these numerical tests, we perform an interpolation of the function

	$r=3$		$r=4$	
ℓ	E^ℓ	o^ℓ	E^ℓ	o^ℓ
5	1.3938e-06		2.3237e-08	
6	2.9147e-08	5.58	1.8794e-10	6.95
7	5.4251e-10	5.75	1.1793e-12	7.32
8	1.0811e-11	5.65	5.3291e-15	**7.79**
9	1.9518e-13	**5.79**	5.5511e-16	-

Table 11: Non-uniform grid refinement analysis for the **new** WENO-6 algorithm for the function (47).

$$f_4(x_1,x_2)=\begin{cases}\frac{1}{16}(x_1+x_2)\sin(16\pi x_1)\sin(16\pi x_2), & x_1+x_2\leq 0,\\ \frac{1}{16}(x_1+x_2)\sin(16\pi x_1)\sin(16\pi x_2)+0.1, & x_1+x_2>0,\end{cases} \tag{51}$$

at the points:

$$\tilde{B}^7=\{(\tau(k_1)x_1+(1-\tau(k_1))y_1,\tau(k_2)x_2+(1-\tau(k_2))y_2):(x_1,x_2),(y_1,y_2)\in\tilde{X}^7,\ \tau(k)=0.3+0.1k,\ k=0,1,2,3,4\}$$

being $\tilde{X}^7$ a non-uniform grid constructed employing the formula described in Eq. (50) (see Figure 8(a)). We show the result in Figure 8(c) and (d) confirming that Gibbs phenomenon does not appear.

8. Conclusions and future work

In this work, we have presented a new WENO method with a progressive order of accuracy close to the discontinuities. This new algorithm is simpler computationally than the one presented in [3] and more general. In particular, our main contributions can be summarised in the following: we design a new recursive method to compute the interpolation; we construct non-linear explicit formulas to calculate the weights to interpolate any point in the central interval; we propose an algorithm for non-uniform grids, and finally, we extend the method to n dimensions. This strategy presents a drawback, the stencils used to interpolate are square and the number of points is larger than necessary to get the optimal order of accuracy. To improve this fact and to apply this technique in partial differential equations contexts are the principal lines of our future work.

References

[1] S. Amat, J. Ruiz (2017): "New WENO smoothness indicators computationally efficient in presence of corner discontinuities", *J. Sci. Comput.*, 71 1265-1237.

[2] S. Amat, J. Ruiz, C.W. Shu, (2019): "On New Strategies to Control the Accuracy of WENO Algorithms Close to Discontinuities", *SIAM Journal on Numerical Analysis* 57:3, 1205-1237.

[3] S. Amat, J. Ruiz, C.W. Shu, D. F. Yáñez, (2020): "A new WENO-2r Algorithm with progressive order of accuracy close to discontinuities", *SIAM J. Numer. Anal., 58(6)*, 3448-3474.

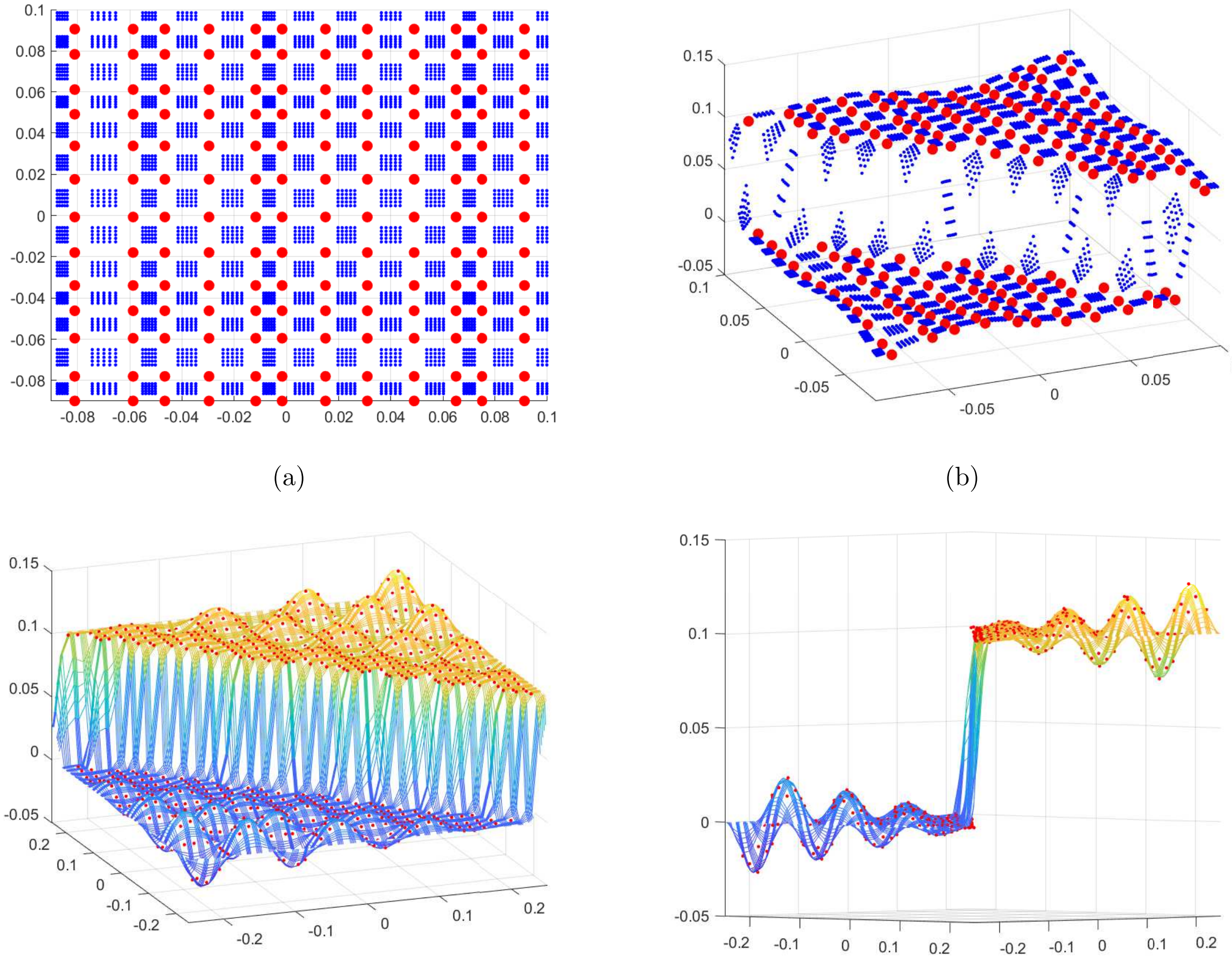
[4] S. Amat, J. Ruiz, C.W. Shu, D. F. Yáñez, (2022): "On the approximation of derivative values using a WENO algorithm with progressive order of accuracy close to discontinuities", *Comp. Appl. Math., 41, 297.*

[5] F. Aràndiga, A. Baeza, A. M. Belda, P. Mulet, (2011): "Analysis of WENO schemes for full and Global accuracy", *SIAM J. Numer. Anal., 49(2)*, 893-915.

[6] F. Aràndiga, A. M. Belda, P. Mulet, (2010): "Point-Value WENO Multiresolution Applications to Stable Image Compression", *J. Sci. Comput.*, 43(2), 158–182.

[7] F. Aràndiga and P. Mulet and V. Renau (2012): "Non-separable two-dimensional weighted ENO interpolation", *App. Num. Math.* 62, 8, 975–987.

[8] A. Baeza, R. Bürger, P. Mulet and D. Zorio (2019): "On the Efficient Computation of Smoothness Indicators for a Class of WENO Reconstructions", *J. Sci. Comput.*, 80, 62, 1240-–1263.



(a)

(b)

(c)

(d)

Figure 8: (a) and (b) Non-uniform grid. Red: nodes, blue: points of interpolation. (c) and (d). Interpolation to the function (51) using the new WENO-6 algorithm

[9] R. Borges, M. Carmona, B. Costa and W. S. Don(2008): "An improved weighted essentially non-oscillatory scheme for hyperbolic conservation laws", *J. Comput. Phys.*, 227 (6), 3191-3211.

[10] H. Feng, C. Huang, R. Wang(2014): "A new mapped weighted essentially non-oscillatory scheme", *Appl. Math. Comput.*, 232, 453-468.

[11] M. Gasca and E. A. Lebrón (1987): "On Aitken-Neville Formulae for Multivariate Interpolation", *North-holland Mathematics Studies*, 133, 133–140.

[12] G. Jiang and C.-W. Shu, (1996): "Efficient Implementation of Weighted ENO Schemes", *J. Comput. Phys.*, 126(1), 202-228.

[13] Y. Ha, C.H. Kim, Y.J. Lee., J. Yoon, (2013): "An improved weighted essentially non-oscillatory scheme with a new smoothness indicator", *J. Comput. Phys.*, 232, 68-86.

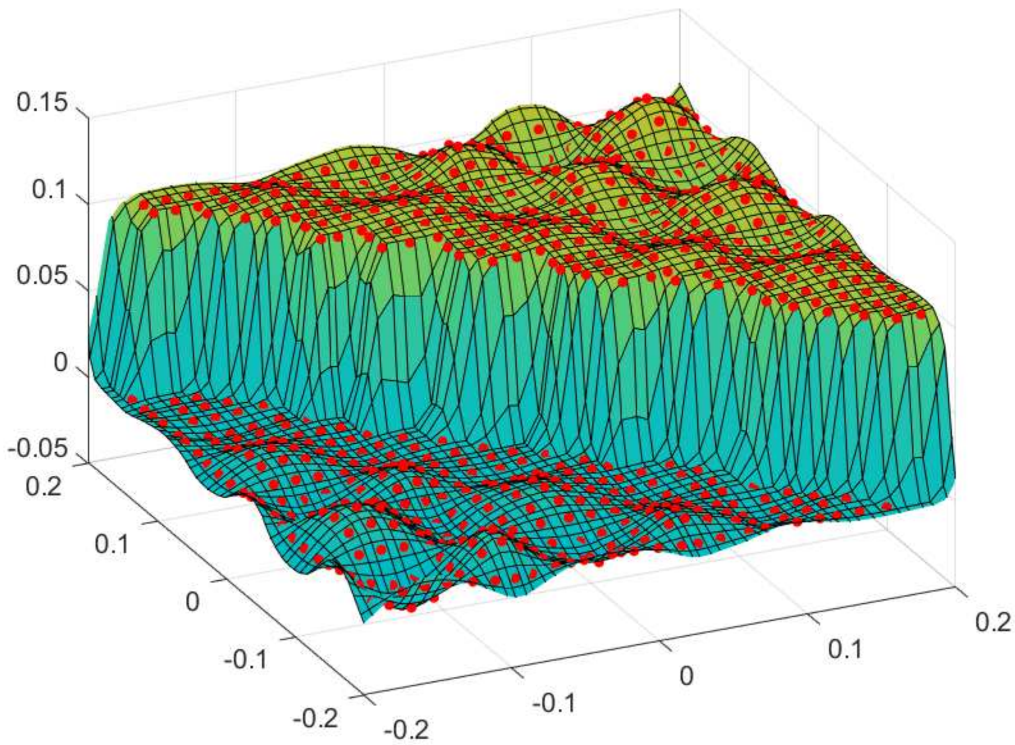
[14] D. Levy, G. Puppo and G. Russo, (1999): "Central WENO schemes for hyperbolic systems of conservation laws," *Esaim Math. Model. Numer. Anal.*, 33(3), 547–571.

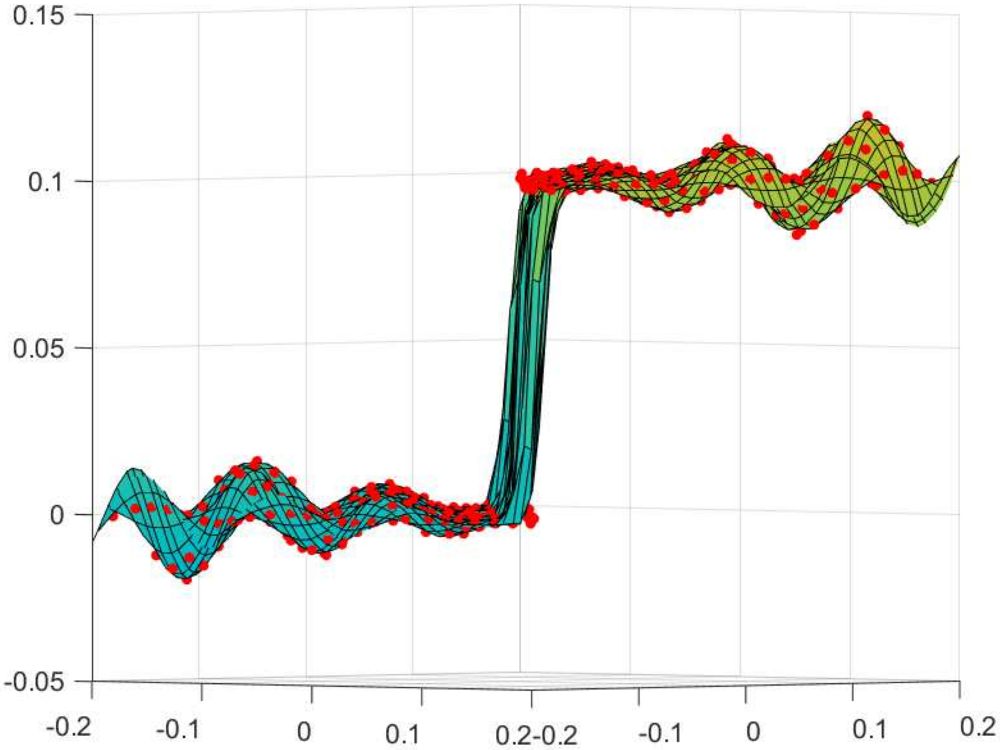
[15] X.-D. Liu and S. Osher and T. Chan, (1994): "Weighted Essentially Non-oscillatory Schemes", *J. Comput. Phys.*, 115, 200-212.

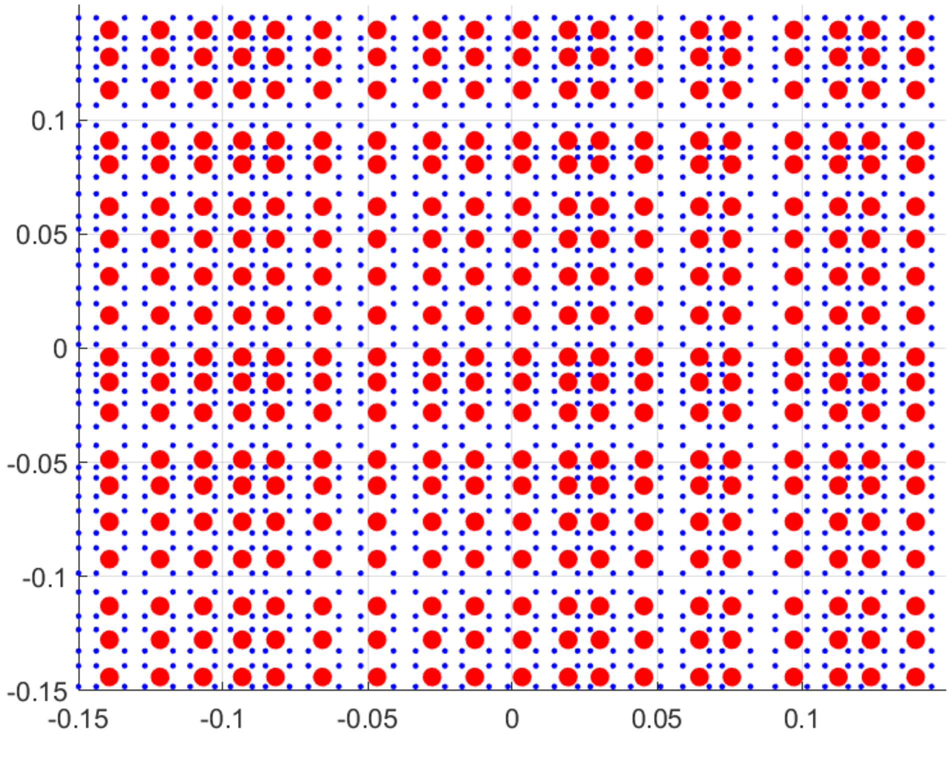
[16] A. K. Henrick, T. D. Aslam and J. M. Powers,(2005): “Mapped weighted essentially non-oscillatory schemes: achieving optimal order near critical points,” *J. Comput. Phys.*, 207 (2), 542-567.

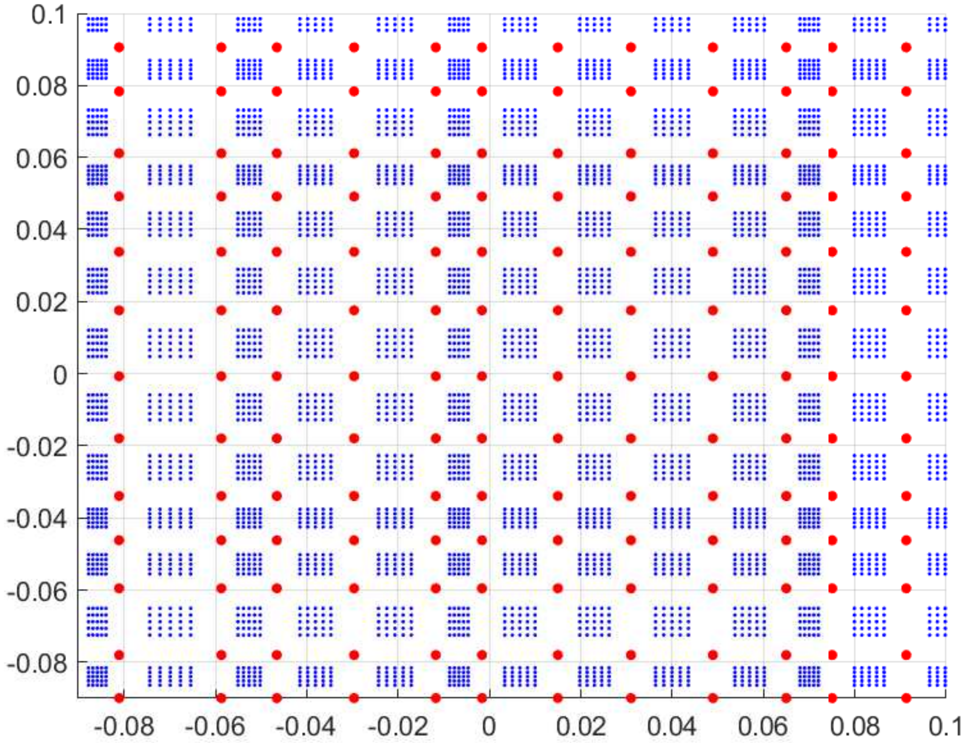
[17] Z. Hong, Z.Y. Ye, K. Ye(2020): “An improved WENO-Z scheme with symmetry-preserving mapping,” *Adv. Aerodyn*, 2, 18.

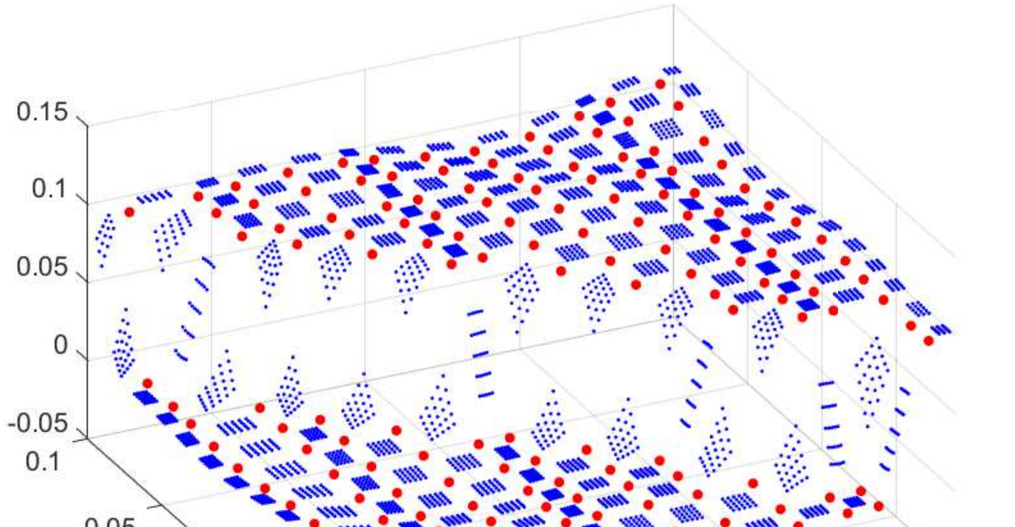
[18] C.-W. Shu, (1999): “High Order Weighted Essentially Nonoscillatory Schemes for Convection Dominated Problems”, *SIAM Review*, 51(1), 82–126.


0.15
0.1
0.05
0
-0.05
0.2
0.1
0
-0.1
-0.2
-0.2
-0.1
0
0.1
0.2


0.15
0.1
0.05
0
-0.05
-0.2
-0.1
0
0.1
0.2-0.2
-0.1
0
0.1
0.2


0.1
0.05
0
-0.05
-0.1
-0.15
-0.15
-0.1
-0.05
0
0.05
0.1


0.1
0.08
0.06
0.04
0.02
0
-0.02
-0.04
-0.06
-0.08
-0.08
-0.06
-0.04
-0.02
0
0.02
0.04
0.06
0.08
0.1


0.15
0.1
0.05
0
-0.05
0.1
0.05
0
-0.05
-0.05
0
0.05
0.1