# ON NEW STRATEGIES TO CONTROL THE ACCURACY OF WENO ALGORITHMS CLOSE TO DISCONTINUITIES*

SERGIO AMAT†, JUAN RUIZ ‡, AND CHI-WANG SHU §

**Abstract.** In this paper we construct and analyze new nonlinear optimal weights for WENO interpolation which are capable of raising the order of accuracy close to jump discontinuities in the function or in the first derivative (kinks). The new nonlinear optimal weights are constructed using a strategy inspired by the original WENO algorithm, and they work very well for kinks or jump discontinuities, leading to optimal theoretical accuracy. This is the first part of a series of two papers. In this first part we analyze the performance of the new algorithms proposed for univariate function approximation in the point values (interpolation problem). In the second part, we will extend the analysis to univariate function approximation in the cell averages (reconstruction problem) and to the solution of problems in the context of hyperbolic conservation laws.

Our aim is twofold: to raise the order of accuracy of the WENO type interpolation schemes both near discontinuities in the function or in the first derivative (kinks) and in the intervals which contain a kink. The first problem can be solved using the new nonlinear optimal weights, but the second one requires a new strategy that locates the position of the singularity inside the cell in order to attain adaption, this new strategy is inspired by the ENO-SR schemes proposed by Harten in A. Harten, ENO schemes with subcell resolution, J. Comput. Phys. 83 (1) (1989) 148 – 184. Thus, we will introduce two different algorithms in the point values. The first one can deal with kinks and jump discontinuities for intervals not containing the singularity. The second algorithm can also deal with intervals containing kinks, as they can be detected from the point values, but jump discontinuities can not, as the information of their position is lost during the discretization process. As mentioned before, the second part of this work will be devoted to the cell averages and, in this context, it will be possible to work with jump discontinuities as well.

**Key words.** WENO schemes, new optimal weights, improved adaption to discontinuities, signal processing.

**AMS subject classifications.** 65D05, 65D17, 65M06, 65N06

## 1. Introduction.

The reconstruction of a piecewise continuous function from some discretized data points is an important problem in the approximation theory. We will consider two possible ways of discretizing the initial set of data: it might come from a sampling of a piecewise continuous function or from the averaging of a function in $L^1$ over certain intervals. In the first case we are talking about a *point value* discretization and in the second case about a *cell average* discretization. This is the first part of a series of two articles where we present a new algorithm for approximation of piecewise smooth functions. In this part we will only consider the point value discretization. The second part [1] will be dedicated to the cell average discretization and its application to the solution of conservation laws.

When approximating a function from discretized data, we can choose to use linear or nonlinear algorithms. Linear algorithms usually present accuracy problems when the stencil crosses a discontinuity: Gibbs oscillations usually appear and the accuracy is lost locally around the discontinuity. The increasing of the length of the stencil

*Submitted to the editors September $19^{th}$, 2018.

**Funding:** This work was funded by the *Programa de Apoyo a la investigación de la fundación Séneca-Agencia de Ciencia y Tecnología de la Región de Murcia* 19374/PI714 and through the national research project MTM2015-64382-P (MINECO/FEDER) and by NSF grant DMS-1719410

†Department of Applied Mathematics and Statistics, Universidad Politécnica de Cartagena (UPCT) (Spain), (sergio.amat@upct.es),

‡**Corresponding author**. Department of Physics and Mathematics, Universidad de Alcalá (UAH) (Spain), (juan.ruiza@uah.es).

§Division of Applied Mathematics, Brown University (USA), (shu@dam.brown.edu).

does not solve the problem and usually results in larger zones affected by oscillations. ENO (essentially non-oscillatory) interpolation solves this problem choosing stencils that do not cross the discontinuity. This algorithm was introduced in [2, 3] for solving conservation laws problems. ENO scheme manages to reduce the zones affected by oscillations to the interval where the discontinuity is placed. This task is done using a stencil selection strategy which allows us to choose the smoothest stencil. The reader interested in obtaining more information about ENO algorithm can refer to the following incomplete list of references [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12].

In [13], Liu, Osher and Chan proposed WENO (weighted ENO) algorithm, which aim was to improve the results obtained by ENO method. This technique was proposed in [13] as a nonlinear convex combination of the approximations obtained by different interpolants constructed over sub-stencils, all of them fragments of a bigger stencil. The weights used were calculated through an estimation of the smoothness of the interpolants used.

The smoothness of the data is estimated using *smoothness indicators* that are functions that take divided differences as arguments. In [14] the authors presented new smoothness indicators which, crucially, were more efficient than those proposed initially in [13]. The computation of these indicators is done through a measurement based on the sum of the $L^2$ norms of the derivatives of the interpolatory polynomials at the interval where we want to obtain the prediction. The computational cost of this measurement is smaller than the one obtained using the total variation and its result is smoother and easier to compute than the total variation. The nonlinear weights are designed in such a way that the stencils that cross a discontinuity pose an insignificant contribution to the resulting interpolation. The purpose of the WENO algorithm proposed in the seminal reference [13] was to optimize the stencil used by the ENO algorithm at smooth zones, in order to attain a higher order of accuracy. The interested reader can refer to [5, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33] and especially to [34, 35] and the references therein in order to get a more complete picture of the state of the art about WENO.

As it was originally conceived, the WENO strategy only imposes restrictions to the weights of the convex combination in smooth zones: the main objective is to reach maximum order of accuracy when the data is smooth in the whole bigger stencil. However, close to discontinuities the value of the weights is mainly taking care of the essentially non-oscillatory property, not the order of accuracy, hence the order of accuracy is not optimized if there is more than one sub-stencil free of discontinuities. Basically, this property of the WENO algorithm is due to the usage of fixed *optimal-weights* when constructing the nonlinear weights of the convex combination of interpolants. This problem can be easily appreciated if we perform a grid refinement analysis around a discontinuity and we obtain the numerical accuracy obtained by the algorithm. The interested reader can have a look to the experiments presented in [36], especially Tables 1 and 3 or Figure 5.

Our aim in this article is to increase the accuracy of the WENO method close to kinks or jump discontinuities when the data is discretized in the point values. We will tackle this task by proposing new nonlinear optimal weights. The main objective is to attain the maximum theoretical accuracy close to discontinuities in the function or in the first derivative, while keeping maximum accuracy in smooth zones. New smoothness indicators were introduced in [36] in order to allow the WENO scheme to simultaneously detect kinks and jump discontinuities in the point values. At the same time, these smoothness indicators show the same properties as the original smoothness indicators proposed in [14]. Using these smoothness indicators we propose two new algorithms. The first one aims to attain optimal control of the accuracy of the interpolation around discontinuities, but not in the interval that contains the discontinuity. The objective of the second one is to raise the accuracy of the WENO algorithm also in the intervals that contain a kink. It is well known that the classical WENO scheme loses its accuracy when a discontinuity is placed in the central interval of the stencil. The state of the art literature includes algorithms that try to solve this issue of the classical WENO scheme. For example, in [37] the authors succeed in increasing the accuracy of the approximation, but they do not obtain the maximum accuracy theoretically possible. In this paper we increase the order of accuracy in the central cell of the stencil and obtain optimal accuracy.

This paper is organized as follows: Section 2 introduces the discretization of data that will be used in the whole article and shows how the WENO algorithm for point values works. Subsection 2.1 presents a very brief description of the nonlinear WENO weights. In Subsection 2.2 we review some smoothness indicators that appear in the literature. Section 3 is devoted to the new WENO algorithm. Subsection 3.1 is dedicated to the introduction of new smoothness indicators more adapted for working with kinks. Subsection 3.2 explains how to redesign the WENO optimal weights in order to control the accuracy close to discontinuities, but not in the interval that contains the discontinuity. Subsection 3.3 presents a strategy through which we are able of raising the accuracy in the interval that contains the kink and, at the same time, controlling the order of accuracy close to it. Subsection 3.4 analyzes the ENO property for the two algorithms presented. Section 4 is dedicated to test the new algorithms through some numerical experiments. In particular, we analyze the performance of the new algorithm using discretized univariate functions that show kinks and jump discontinuities. Finally, Section 5 presents the conclusions.

## 2. Weighted essentially non-oscillatory (WENO) algorithm for point values.

In this section we introduce the classical WENO method. The concepts presented in this section are already classic and can be found in many references, see for example [13, 14, 36, 37], but their presence is strictly necessary to keep the paper self-contained and to introduce the different notations that we will use.

Let us consider the space of finite sequences $V$, a uniform partition $X$ of the interval $[a, b]$ in $J$ subintervals, and the set of piecewise continuous functions in the interval $[a, b]$,

$$X = \{x_i\}_{i=0}^{J}, \quad x_0 = a, \quad h = x_i - x_{i-1}, \quad x_J = b.$$

We will use a point value discretization of the data,

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

We can see that the previous discretization preserves the information locally at the sites $x_i$. Although it is possible to locate the position of kinks, as shown in Figure 1, there is no hope in locating the exact positions of jumps, as they are lost in the discretization process [38], as shown in Figure 2. We will always consider that discontinuities are far enough from each other (for WENO algorithm and stencils of 6 points we will consider that we have at least four discretization grid-points between any adjacent discontinuities).

In this section we introduce the WENO scheme. As mentioned before, this algorithm allows us to obtain a high order of accuracy at smooth zones of $f$ and, at the same time, it manages to avoid Gibbs oscillations close to discontinuities. This technique appeared as an improvement of ENO reconstructions [2, 3]. The ENO scheme

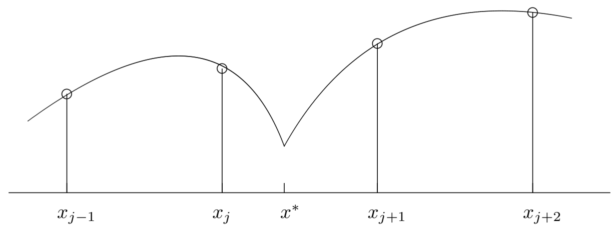


FIG. 1. *This figure represents a kink placed in the interval $(x_j, x_{j+1})$ at a position $x^*$. If we consider that the discretized data is in the point values, we can recover an approximation of the position of the discontinuity crossing an interpolating polynomial built using the data to the right of the discontinuity with another interpolating polynomial built using the data to the left.*

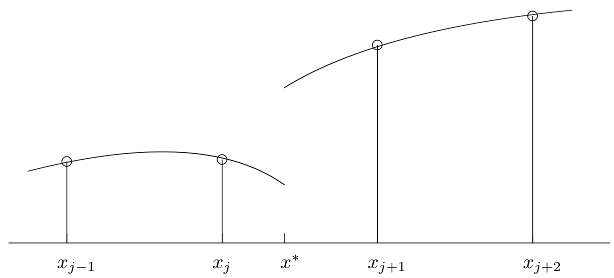


FIG. 2. *This figure represents a jump discontinuity placed in the interval $(x_j, x_{j+1})$ at a position $x^*$. In this case it is not possible to recover the position of the discontinuity from the discretized data in the point values [38].*

uses a stencil selection procedure and manages to obtain an order of accuracy $r+1$. In order to do this, this scheme deals with several stencils of length $r+1$. The ENO scheme uses divided differences in order to determine which stencil is the smoothest. The WENO scheme uses smoothness indicators based on (divided) differences to determine the smoothness of the stencil.

We will denote the different stencils by $S_i^m(j) = \{x_{j-m+i}, \cdots, x_{j+i-1}\}$. The WENO scheme uses the same stencil of $2r$ nodes $S_r^{2r}(j) = \{x_{j-r}, \cdots, x_{j+r-1}\}$ as the ENO method when trying to interpolate in the interval $(x_{j-1}, x_j)$. Using this stencil, WENO manages to reach order of accuracy $2r$ [13] at smooth regions of $f$. In our notation, $S_k^r(j),\ \ k = 0, \cdots, r-1$ will represent the $r$ sub-stencil of length $r+1$ that contains the interval $(x_{j-1}, x_j)$, where we want to interpolate:

$$S_k^r(j) = \{x_{j-r+k}, \cdots, x_{j+k}\}, \quad k = 0, \cdots, r-1. \tag{2.1}$$

Figure 3 presents a diagram where we show the big stencil $S_r^{2r}(j)$ and the sub-stencils $S_k^r(j), k = 0, \cdots, r-1$ considered for the particular case $r = 3$.

Let's consider the following convex combination,

$$q_{j-r}(x) = \sum_{k=0}^{r-1} \omega_k^r(j) p_{j-r+k}^r(x), \tag{2.2}$$

where $\omega_k^r(j) \geq 0,\ k = 0, \cdots, r-1$ and $\sum_{k=0}^{r-1} \omega_k^r(j) = 1$. In (2.2), $p_{j-r+k}^r(x)$ represents the interpolatory polynomial of degree $r$ defined on the stencil $S_k^r(j)$. The prediction

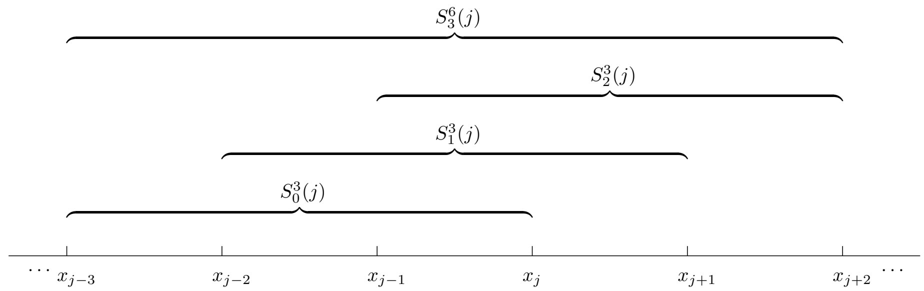


FIG. 3. *In this diagram we represent for* $r=3$ *the big stencil* $S_r^{2r}(j)=\{x_{j-r},\cdots,x_{j+r-1}\}$ *and the substencils* $S_k^r(j)=\{x_{j-r+k},\cdots,x_{j+k}\}$ *for* $k=0,\cdots,r-1$.

operator for the mid point of the target interval $(x_{j-1},x_j)$ is given by

$$I\left(x_{j-\frac{1}{2}};f\right)=\sum_{k=0}^{r-1}\omega_k^r(j)p_{j-r+k}^r\left(x_{j-\frac{1}{2}}\right). \tag{2.3}$$

The value of the weights is chosen in order to obtain order of accuracy $2r$ at $x_{j-\frac{1}{2}}$ at smooth regions of the function $f$. In [13], the authors use an interpolant satisfying,

$$p_{j-r}^{2r-1}\left(x_{j-\frac{1}{2}}\right)=f\left(x_{j-\frac{1}{2}}\right)+O\left(h^{2r}\right), \tag{2.4}$$

on the big stencil $\{x_{j-r},\cdots,x_{j+r-1}\}$, if we suppose that the function is smooth there. We can also build $r$ approximations

$$p_{j-r+k}^r\left(x_{j-\frac{1}{2}}\right)=f\left(x_{j-\frac{1}{2}}\right)+O\left(h^{r+1}\right), \tag{2.5}$$

using the small stencils $S_k^r(j)$. The optimal linear weights must satisfy that $C_k^r(j)\geq 0,\forall k$ and also that $\sum_{k=0}^{r-1}C_k^r(j)=1$, such that

$$p_{j-r}^{2r-1}\left(x_{j-\frac{1}{2}}\right)=\sum_{k=0}^{r-1}C_k^r(j)p_{j-r+k}^r\left(x_{j-\frac{1}{2}}\right). \tag{2.6}$$

The formulae for the optimal weights are easy to obtain if we use Newton interpolating polynomials. In [16], the authors give a proof for the following expression,

$$C_k^r(j)=\frac{1}{2^{2r-1}}\binom{2r}{2k+1},\quad k=0,\cdots,r-1. \tag{2.7}$$

For $r=3$ the optimal weights are $C_0^3(j)=\frac{3}{16},C_1^3(j)=\frac{10}{16},C_2^3(j)=\frac{3}{16}$. In fact, in [39] the authors prove that the weights $C_k^r(j)$ can be written as polynomials. However, we are usually interested in computing the reconstruction in specific points of the considered interval. In this case the polynomials $C_k^r(j)$ take some specific positive values. We will consider this case.

**2.1. Nonlinear weights.** In [13], the non linear weights $\omega_k^r(j)$ are designed to satisfy the following relation at smooth zones,

$$\omega_k^r(j) = C_k^r(j) + O(h^m), \quad k = 0, \cdots, r-1 \tag{2.8}$$

with $m \leq r-1$. Then, at these zones the interpolation error satisfies,

$$f(x_{j-\frac{1}{2}}) - q_{j-r}(x_{j-\frac{1}{2}}) = O(h^{r+m+1}). \tag{2.9}$$

When $m = r-1$ in (2.8), (2.9) assures that the accuracy attained is $2r$. That said accuracy is the same as the one obtained using the interpolant $p_{j-r}^{2r-1}(x)$ that uses all the nodes in the big stencil. The weights must also be designed in such a way that they satisfy the ENO property. This means that the contribution to the convex combination (2.3) of polynomials built from stencils crossing discontinuities should be insignificant. As mentioned in [13], the weights should also be easy to compute. The expression for the weights is,

$$\omega_k^r(j) = \frac{\alpha_k^r(j)}{\sum_{i=0}^{r-1} \alpha_i^r(j)}, \quad = 0, \cdots, r-1 \text{ where } \alpha_k^r(j) = \frac{C_k^r(j)}{(\epsilon + I_k^r(j))^t}. \tag{2.10}$$

This expression for the weights satisfies that $\sum_k \omega_k^r(j) = 1$. In (2.10) $I_k^r(j)$ represents a smoothness indicator for $f(x)$ on the stencil $S_k^r(j)$. $t$ is an integer that has the purpose of assuring the maximum order of accuracy close to the discontinuities. The value of this parameter varies in the literature. For example, in [14] the authors choose $t = 2$ and in [13], it is set to $t = r$. In the theoretical proofs about the accuracy, we will determine which value of $t$ we should take in our algorithm. The positive parameter $\epsilon$ that appears in the denominator of (2.10) is included to avoid divisions by zero. Some references can be found in the literature [15, 16], where the authors prove that $\epsilon$ plays a role when we are interpolating close to critical points at smooth zones. In this article we will show that the smoothness indicators used satisfy the requirements exposed in [13, 16] and necessary to attain the desired accuracy. We will also analyze the role played by the parameter $\epsilon$ and explicitly set the value it must take in order to obtain optimal results with the new algorithms presented.

As we will refer all the time to the big stencil $S_r^{2r}(j)$ and in order to ease the notation, we will drop $(j)$ in $S_k^r(j), \omega_k^r(j), C_k^r(j), \alpha_k^r(j)$ and use simply $S_k^r, \omega_k^r, C_k^r, \alpha_k^r$.

**2.2. Classical smoothness indicators.** As mentioned before, the computation of the smoothness indicators is done through a measurement based on the sum of the $L^2$ norms of the derivatives of the interpolatory polynomials at the interval where we want to obtain the prediction [14],

$$I_k^r(j) = \sum_{l=1}^{r-1} h^{2l-1} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \left( \frac{d^l}{dx^l} p_{j-r+k}^k(x) \right)^2 dx. \tag{2.11}$$

In [16] another expression for the smoothness indicators is introduced, this time for data discretized in the point values,

$$I_k^r(j) = \sum_{l=1}^{r} h^{2l-1} \int_{x_{j-1}}^{x_j} \left( \frac{d^l}{dx^l} p_{j-r+k}^r(x) \right)^2 dx. \tag{2.12}$$

The smoothness indicators presented before are suitable for jump discontinuities, but they do not work well for kinks.

In [36] we propose a new expression for the smoothness indicators that works well for kinks and for data discretized in the point values,

$$I_k^r(j) = \sum_{l=2}^{r} h^{2l-1} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \left( \frac{d^l}{dx^l} p^r_{j-r+k}(x) \right)^2 dx. \tag{2.13}$$

This is the smoothness indicator used for computation of Hamilton-Jacobi equations [40], for which kinks are the generic singularities in the viscosity solutions.

## 3. The new WENO algorithm for point values.

In this section we introduce the new WENO algorithm. The difference with the classical WENO algorithm introduced in previous section is mainly located in the design of the WENO weights. We also make use of new smoothness indicators more suitable for working in the point values.

### 3.1. New smoothness indicators.

The smoothness indicator in (2.13) integrates in the interval $(x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}})$, but it seems more logical to integrate in the interval $(x_{j-1}, x_j)$, where the point where we will interpolate is in the middle. Thus, in this article we propose to use the smoothness indicators in the point values given by the expression

$$I_k^n(j) = \sum_{l=2}^{\min(r,n)} h^{2l-1} \int_{x_{j-1}}^{x_j} \left( \frac{d^l}{dx^l} p^n_{j-r+k}(x) \right)^2 dx, \tag{3.1}$$

where $n$ is the degree of the polynomial and goes from $n = r-1, \cdots, 2r-1$. As we will see in Subsection 3.2, for $r = 3$ our algorithms make use of smoothness indicators of 3, 4, 5 and 6 points in order to optimize the accuracy of the new nonlinear interpolation proposed. Thus, we will need to build polynomials from degree two to five with the aim of replacing in (3.1) and obtaining such smoothness indicators. We work with the stencil of six points $S_3^6 = \{x_{j-3}, x_{j-2}, x_{j-1}, x_j, x_{j+1}, x_{j+2}\}$ and we will obtain the smoothness indicators integrating in the interval $(x_{j-1}, x_j)$. The point values used will be $\{f_{j-3}, f_{j-2}, f_{j-1}, f_j, f_{j+1}, f_{j+2}\}$.

In order to obtain compact expressions for the smoothness indicators in terms of finite differences, the polynomials can be expressed in the Newton form. The polynomials of degree $n$ starting at the node $j$ have the form,

(3.2)

$$\begin{aligned} p_j^n(x) = f_j &+ \frac{f_{j+1}-f_j}{h}(x-x_j) + \frac{f_j - 2f_{j+1} + f_{j+2}}{2h^2}(x-x_j)(x-x_{j+1}) \\ &+ \frac{-f_j + 3f_{j+1} - 3f_{j+2} + f_{j+3}}{6h^3}(x-x_j)(x-x_{j+1})(x-x_{j+2}) + \cdots + \frac{\delta_j^n}{n!h^n} \prod_{k=j}^{j+n-1} (x-x_k), \end{aligned}$$

where $\delta_j^n = f[x_j, \cdots, x_{j+n}]h^n$ are finite differences of order $n$. Using a stencil of six points, we can build four different polynomials of degree two $\{p^2_{j-3}(x), p^2_{j-2}(x), p^2_{j-1}(x), p^2_j(x)\}$, three of degree three $\{p^3_{j-3}(x), p^3_{j-2}(x), p^3_{j-1}(x)\}$, two of degree four $\{p^4_{j-3}(x), p^4_{j-2}(x)\}$ and one of degree five $\{p^5_{j-1}(x)\}$. We have used the notation $p^n_{j-r+k}$, for $r = 3$. We will use all of these polynomials to obtain smoothness indicators. As before, we will drop $(j)$ in the notation of the smoothness indicators $I_k^r(j)$ and simply use $I_k^r$.

The smoothness indicators of three points obtained through (3.1) for $n = r-1 = 2$ and the polynomials $\{p^2_{j-3}(x), p^2_{j-2}(x), p^2_{j-1}(x), p^2_j(x)\}$ can be expressed in terms of finite differences as,

$$I^2_{-1} = (\delta^2_{j-3})^2, \quad I^2_0 = (\delta^2_{j-2})^2, \quad I^2_1 = (\delta^2_{j-1})^2, \quad I^2_2 = (\delta^2_{j})^2, \tag{3.3}$$

with $\delta^2_i = f_i - 2\,f_{i+1} + f_{i+2}$. The smoothness indicators of four points obtained through (3.1) for $n = r = 3$ and the polynomials in $\{p^3_{j-3}(x), p^3_{j-2}(x), p^3_{j-1}(x),\}$ can be expressed in terms of finite differences as,

$$\begin{aligned} I^3_0 &= \frac{10}{3}(\delta^3_{j-3})^2 + 3\delta^3_{j-3}\delta^2_{j-3} + (\delta^2_{j-3})^2, \\ I^3_1 &= \frac{4}{3}(\delta^3_{j-2})^2 + \delta^3_{j-2}\delta^2_{j-2} + (\delta^2_{j-2})^2, \\ I^3_2 &= \frac{4}{3}(\delta^3_{j-1})^2 - \delta^3_{j-1}\delta^2_{j-1} + (\delta^2_{j-1})^2, \end{aligned} \tag{3.4}$$

with $\delta^3_i = -f_i + 3\,f_{i+1} - 3\,f_{i+2} + f_{i+3}$. The smoothness indicators of five points obtained using the same process and the polynomials $\{p^4_{j-3}(x), p^4_{j-2}(x)\}$ are,

$$\begin{aligned} I^4_0 &= \frac{19}{6}\delta^4_{j-3}\delta^3_{j-3} + \frac{2}{3}\delta^4_{j-3}\delta^2_{j-3} + \frac{547}{240}(\delta^4_{j-3})^2 + \frac{10}{3}(\delta^3_{j-3})^2 + 3\delta^3_{j-3}\delta^2_{j-3} + (\delta^2_{j-3})^2 \\ I^4_1 &= \frac{89}{80}(\delta^4_{j-2})^2 - \frac{1}{6}\delta^4_{j-2}\delta^3_{j-2} - \frac{1}{3}\delta^4_{j-2}\delta^2_{j-2} + \frac{4}{3}(\delta^3_{j-2})^2 + \delta^3_{j-2}\delta^2_{j-2} + (\delta^2_{j-2})^2, \end{aligned} \tag{3.5}$$

with $\delta^4_i = f_i - 4\,f_{i+1} + 6\,f_{i+2} - 4f_{i+3} + f_{i+4}$. The smoothness indicator of six points obtained using the same process and the polynomial $p^5_{j-3}(x)$ is,

$$\begin{aligned} I^5_0 = {} & \frac{1727}{1260}(\delta^5_{j-3})^2 + \frac{203}{240}\delta^5_{j-3}\delta^4_{j-3} - \frac{13}{30}\delta^5_{j-3}\delta^3_{j-3} - \frac{1}{6}\delta^5_{j-3}\delta^2_{j-3} \\ & + \frac{19}{6}\delta^3_{j-3}\delta^4_{j-3} + \frac{2}{3}\delta^2_{j-3}\delta^4_{j-3} + \frac{547}{240}(\delta^4_{j-3})^2 + \frac{10}{3}(\delta^3_{j-3})^2 + 3\delta^2_{j-3}\delta^3_{j-3} + (\delta^2_{j-3})^2, \end{aligned} \tag{3.6}$$

with $\delta^5_i = -f_i + 5\,f_{i+1} - 10\,f_{i+2} + 10f_{i+3} - 5f_{i+4} + f_{i+5}$. To obtain these expressions we have applied the formula in (3.1) integrating always in the interval $(x_{j-1}, x_j)$.

THEOREM 3.1. *At smooth zones, the smoothness indicators (3.4), (3.5) and (3.6) calculated using the expression in (3.1) can be simplified to*

$$I^n_k = \left(h^2 f''_{j-1/2}\right)^2 \cdot (1 + O(h^2)), \quad n = 3, 4, 5.$$

*Proof.* At smooth zones, obtaining the Taylor expansion of the values of the stencil $\{f_{j-3}, f_{j-2}, f_{j-1}, f_j, f_{j+1}, f_{j+2}\}$ around $x_{j-1/2}$ and replacing them in the expressions of the smoothness indicators in (3.4), (3.5) and (3.6), we obtain that $I^4_0, I^4_1$ and $I^5_0$ are equal to $D_1$, $I_0^3$ and $I_2^3$ are equal to $D_2$ and $I_1^3$ is equal to $D_3$, with

$$
\begin{aligned}
D_1 &= h^4\left(\frac{d^2f}{dx^2}\left(x_{j-1/2}\right)\right)^2+\frac{13}{12}h^6\left(\frac{d^3f}{dx^3}\left(x_{j-1/2}\right)\right)^2\\
&\quad+\frac{1}{12}h^6\left(\frac{d^2f}{dx^2}\left(x_{j-1/2}\right)\right)\frac{d^4f}{dx^4}\left(x_{j-1/2}\right)+O(h^7).\\
D_2 &= h^4\left(\frac{d^2f}{dx^2}\left(x_{j-1/2}\right)\right)^2+\frac{13}{12}h^6\left(\frac{d^3f}{dx^3}\left(x_{j-1/2}\right)\right)^2\\
&\quad-\frac{7}{12}h^6\left(\frac{d^2f}{dx^2}\left(x_{j-1/2}\right)\right)\frac{d^4f}{dx^4}\left(x_{j-1/2}\right)+O(h^7).\\
D_3 &= h^4\left(\frac{d^2f}{dx^2}\left(x_{j-1/2}\right)\right)^2+\frac{13}{12}h^6\left(\frac{d^3f}{dx^3}\left(x_{j-1/2}\right)\right)^2\\
&\quad+\frac{5}{12}h^6\left(\frac{d^2f}{dx^2}\left(x_{j-1/2}\right)\right)\frac{d^4f}{dx^4}\left(x_{j-1/2}\right)+O(h^7).
\end{aligned} \tag{3.7}
$$

Collecting $h^4\left(\frac{d^2f}{dx^2}\left(x_{j-1/2}\right)\right)^2$, we get the result. ∎

**3.2. Obtaining optimal weights close to discontinuities in the point values.** If the optimal weights are obtained close to a discontinuity in the way specified in (2.6) without any other consideration, the accuracy can be lost when there is more than one smooth substencil. A representation of a typical example of this situation is shown in Figures 4 and 5. The idea is that when a stencil is affected by a discontinuity, WENO is not designed to use all the available smooth information. In fact, the only conditions imposed to obtain the weights of the convex combination of polynomials of WENO interpolation in (2.2) is that they must depend on the smoothness of the function (they are large if the corresponding sub-stencil is smooth and small otherwise), and that at smooth zones the convex combination must provide optimal accuracy. For example, if we are working with stencils of 6 points and a convex combination of three polynomials of degree 3, then, in a situation like the ones depicted in Figures 4 and 5, WENO interpolation will typically provide $O(h)$ accuracy at the interval that contains the discontinuity and $O(h^4)$ accuracy at the other intervals of the stencils that are affected by the discontinuity, even though there is available information to obtain $O(h^5)$ accuracy at the point $x_{j-1/2}$ shown in Figures 4 and 5. If we obtain $O(h^5)$ accuracy in the mentioned interval, it is just by coincidence, as the weights as originally proposed in [14] are not designed to optimize the use of the stencil. It is possible to optimize the weights of the convex combination making the optimal weights also depend on the smoothness of the function, such that the optimal order is attained in all the stencils affected by the discontinuity.

In this case, we will analyze how to attain optimal order with exactly the same stencil and sub-stencils that WENO method uses. Thus, we will use the formula for the interpolant in (2.3). In order to ease the presentation of the new optimal weights, we analyze the case $r = 3$ that corresponds to $n = 2r = 6$ points. Let's start with the three stencils of four points $S_0^3 = \{x_{j-3}, x_{j-2}, x_{j-1}, x_j\}$, $S_1^3 = \{x_{j-2}, x_{j-1}, x_j, x_{j+1}\}$ and $S_2^3 = \{x_{j-1}, x_j, x_{j+1}, x_{j+2}\}$. The point values used will be $\{f_{j-3}, f_{j-2}, f_{j-1}, f_j, f_{j+1}, f_{j+2}\}$. With these conditions, it is straightforward to build polynomials in the Newton form shown in (3.2). We can denote them by $p^3_{j-3+k}(x)$, such that $r = 3$ denotes the degree of the polynomial and $j-3+k$ the node where the substencil starts. Nevertheless, it is more convenient to ease again the notation dropping the dependence with $j$ and simply write $p_k^r(x)$, as we will be referring all the time to the stencil $S_3^6 = \{x_{j-3}, x_{j-2}, x_{j-1}, x_j, x_{j+1}, x_{j+2}\}$. All the polynomials are evaluated at the point of interpolation $x_{j-1/2}$, as shown in Figures 4 and 5. Then for $r = 3$ we will be dealing with $p_0^3(x), p_1^3(x)$ and $p_2^3(x)$ for the convex combination in (2.6) and $p_0^4(x), p_1^4(x), p_2^5(x)$ for calculating the nonlinear optimal weights. It is straightforward to prove that taking the weights shown in (2.7) for $r = 3$,

$$\begin{aligned} p_0^4(x) &= 2C_0^3 p_0^3(x) + C_1^3 p_1^3(x) \\ p_1^4(x) &= C_1^3 p_1^3(x) + 2C_2^3 p_2^3(x), \end{aligned} \tag{3.8}$$

with $C_0^3, C_1^3$, and $C_2^3$ the optimal weights for $r = 3$ presented in (2.7). It is clear that in the case represented in Figure 5 it would be convenient to use $p_1^4(x)$ in order to interpolate at $x_{j-1/2}$ (and for the case presented in Figure 4, $p_0^4(x)$ is the best option). However, the WENO scheme does not assure that the convex combination of $p_0^3(x_{j-1/2})$ and $p_1^3(x_{j-1/2})$ will be equal to $p_0^4(x_{j-1/2})$. If the discontinuity is located in the intervals $(x_{j-2}, x_{j-1})$ or $(x_j, x_{j+1})$, WENO should obtain $O(h^4)$ accuracy as there is always a smooth stencil of four points. In order to assure maximum accuracy, we can design three vectors of optimal weights $\mathbf{C_0^4}, \mathbf{C_0^5}, \mathbf{C_1^4}$, each of which is suitable for a particular position of the discontinuity. The vectors will have the following expression,

$$\mathbf{C_0^4} = (2C_0^3, C_1^3, 0), \tag{3.9}$$

$$\mathbf{C_0^5} = (C_0^3, C_1^3, C_2^3), \tag{3.10}$$

$$\mathbf{C_1^4} = (0, C_1^3, 2C_2^3). \tag{3.11}$$

$\mathbf{C_0^4}$ is appropriate in the case presented in Figure 4. $\mathbf{C_1^4}$ is adequate for the case in Figure 5. Finally, $\mathbf{C_0^5}$ works well when there is no discontinuity. A weighted average of these vectors will result in non-linear optimal weights that would replace the optimal weights of WENO algorithm. The weights of the mentioned average will be computed using the same technique introduced in [14] for averaging WENO interpolatory polynomials: smoothness indicators. Thus, in order to assure optimal accuracy, we will use smoothness indicators for the polynomials of $4, 5$ and $6$ points that arise from the selected 6 points stencil. Let's now denote by $\tilde{\omega}_k^n$ the quotients,

$$\tilde{\omega}_0^4 = \frac{\tilde{\alpha}_0^4}{\tilde{\alpha}_0^4 + \tilde{\alpha}_0^5 + \tilde{\alpha}_1^4}, \quad \tilde{\omega}_0^5 = \frac{\tilde{\alpha}_0^5}{\tilde{\alpha}_0^4 + \tilde{\alpha}_0^5 + \tilde{\alpha}_1^4}, \quad \tilde{\omega}_1^4 = \frac{\tilde{\alpha}_1^4}{\tilde{\alpha}_0^4 + \tilde{\alpha}_0^5 + \tilde{\alpha}_1^4}, \tag{3.12}$$

with,

$$\tilde{\alpha}_0^4 = \frac{1}{(\epsilon + I_0^4)^t}, \quad \tilde{\alpha}_0^5 = \frac{1}{(\epsilon + I_0^5)^t}, \quad \tilde{\alpha}_1^4 = \frac{1}{(\epsilon + I_1^4)^t}. \tag{3.13}$$

Now we can just define the adapted optimal weights as,

$$(\tilde{C}_0^3, \tilde{C}_1^3, \tilde{C}_2^3) = \tilde{\omega}_0^4 \mathbf{C_0^4} + \tilde{\omega}_0^5 \mathbf{C_0^5} + \tilde{\omega}_1^4 \mathbf{C_1^4}. \tag{3.14}$$

These nonlinear optimal weights $\tilde{C}_k^r$ are used in place of the optimal weights $C_k^r$ in the expression (2.10). The smoothness indicators that appear in (2.10) are obtained using four points, and have the expression shown in (3.4). We keep this part of the algorithm untouched and we only modify the optimal weights, that now are nonlinear.

A first explanation of why this technique works is the following:

- If all the sub-stencils $S_k^n$, $n=3,4,5$, (three of four points, two of five points and one of six points) are smooth and $f''_{j-1/2}\neq 0$, all of them are $I_k^n = (h^2 f''_{j-1/2})^2\cdot(1+O(h^2))$, $n=3,4,5$ (as shown in Theorem 3.1). Then, at sufficiently smooth zones, the nonlinear weights in (3.12) satisfy the expression

$$\tilde{\omega}_k^n = \frac{\tilde{\alpha}_k^n}{\sum_{j=2}^{r}\sum_{i=0}^{r-j}\tilde{\alpha}_i^{j+r-1}} = \frac{1}{(\epsilon+I_k^n)^t}\frac{1}{\sum_{j=2}^{r}\sum_{i=0}^{r-j}\frac{1}{(\epsilon+I_i^{j+r-1})^t}}, \quad n=4,5, \tag{3.15}$$

where we are taking into account that the stencil has $2r$ points and $r=3$. Replacing now $I_k^n = \left(h^2 f''_{j-1/2}\right)^2\cdot(1+O(h^2))$, $n=4,5$, as shown in Theorem 3.1, and taking $\epsilon$ small enough, we obtain

$$\tilde{\omega}_k^r = \frac{(1+O(h^2))^t}{3(1+O(h^2))^t},$$

but $(1+O(h^2))^t = 1+O(h^2)$ and $\frac{1}{(1+O(h^2))^t} = 1+O(h^2)$, so,

$$\tilde{\omega}_k^r = \frac{(1+O(h^2))^t}{3(1+O(h^2)^t} = \frac{1}{3}(1+O(h^2)).$$

For the particular case $r=3$, (3.14) transforms into,

$$\begin{aligned}(\tilde{C}_0^3,\tilde{C}_1^3,\tilde{C}_2^3) &= \frac{1}{3}\left(1+O(h^2)\right)(\mathbf{C_0^4}+\mathbf{C_0^5}+\mathbf{C_1^4}) = \mathbf{C_0^5}+O(h^2)\\ &= (C_0^3,C_1^3,C_2^3)+O(h^2),\end{aligned} \tag{3.16}$$

that are the original optimal weights $C_k^r(j)$ in (2.6) and proposed in [14] plus a small perturbation that, as we will see in Theorem 3.2, does not affect the order of accuracy. Applying exactly the same process but to the WENO weights in (2.10), using as optimal weights those in (3.16) we obtain,

$$\omega_k^r = \frac{\tilde{\alpha}_k^r}{\sum_{i=0}^{r-1}\tilde{\alpha}_i^r} = \frac{C_k^r+O(h^2)}{(\epsilon+I_k^r)^t}\frac{1}{\sum_{i=0}^{r-1}\frac{C_i^r+O(h^2)}{(\epsilon+I_i^r)^t}},$$

Replacing again the expression $I_k^r = I_i^r = \left(h^2 f''_{j-1/2}\right)^2\cdot(1+O(h^2))$, for $r=3$, and taking $\epsilon$ small enough, we obtain,

$$\omega_k^r = \frac{C_k^r+O(h^2)}{(1+O(h^2))^t}\frac{(1+O(h^2))^t}{1+O(h^2)} = C_k^r+O(h^2), \tag{3.17}$$

and the result is that the WENO weights are

$$\omega_k^r = C_k^r+O(h^2) \tag{3.18}$$

- The cases shown in Figure 4 and 5 are symmetric, so we can just analyze the case presented in Figure 4. It is clear that a kink in the interval $(x_{j+1},x_{j+2})$ will produce that the smoothness indicators $I_1^4$ and $I_0^5$ shown in (3.5) and (3.6) respectively will take a value $O(h^2)$ due to the presence of the discontinuity

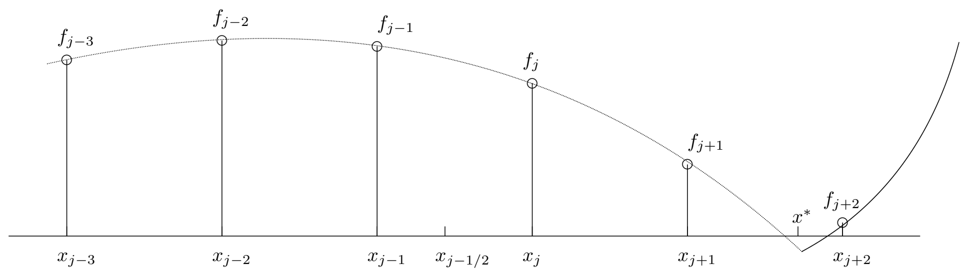


FIG. 4. *Representation of a kink placed at $x^*$ in the interval $(x_{j+1}, x_{j+2})$.*

in the first derivative, while $I_0^4 = \left(h^2 f''_{j-1/2}\right)^2 \cdot (1 + O(h^2)) = O(h^4)$ as that part of the stencil is smooth. If that is the case, then

$$\begin{aligned}
\tilde{\alpha}_0^4 &= \frac{1}{(\epsilon + O(h^4))^t},\\
\tilde{\alpha}_0^5 &= \frac{1}{(\epsilon + O(h^2))^t},\\
\tilde{\alpha}_1^4 &= \frac{1}{(\epsilon + O(h^2))^t}.
\end{aligned}$$

Then we have that,

$$\tilde{w}_k^n = \frac{\tilde{\alpha}_k^n}{\sum_{j=2}^{r}\sum_{i=0}^{r-j} \tilde{\alpha}_i^{j+r-1}} = \frac{1}{(\epsilon + I_k^n)^t} \frac{1}{\sum_{j=2}^{r}\sum_{i=0}^{r-j} \frac{1}{(\epsilon + I_i^{j+r-1})^t}}, \quad n = 4, 5,$$

and $r = 3$. Assuming that $\epsilon$ is small enough, we obtain that the weights are,

$$\begin{aligned}
\tilde{w}_0^4 &= \frac{1}{(\epsilon + I_0^4)^t} \frac{1}{\sum_{j=2}^{r}\sum_{i=0}^{r-j} \frac{1}{(\epsilon + I_i^{j+r-1})^t}} = \frac{1}{(\epsilon + I_0^4)^t} \frac{1}{\frac{1}{(\epsilon + I_0^4)^t}(1 + O(h^{2t}))} = \frac{1}{1 + O(h^{2t})}\\
&= 1 + O(h^{2t}),\\
\tilde{w}_1^4 &= \frac{1}{(\epsilon + I_1^4)^t} \frac{1}{\sum_{j=2}^{r}\sum_{i=0}^{r-j} \frac{1}{(\epsilon + I_i^{j+r-1})^t}} = \frac{1}{(\epsilon + I_1^4)^t} \frac{1}{\frac{1}{(\epsilon + I_0^4)^t}(1 + O(h^{2t}))} = O(h^{2t}),\\
\tilde{w}_0^5 &= \frac{1}{(\epsilon + I_0^5)^t} \frac{1}{\sum_{j=2}^{r}\sum_{i=0}^{r-j} \frac{1}{(\epsilon + I_i^{j+r-1})^t}} = \frac{1}{(\epsilon + I_0^5)^t} \frac{1}{\frac{1}{(\epsilon + I_0^4)^t}(1 + O(h^{2t}))} = O(h^{2t}).
\end{aligned}$$

Then, the adapted optimal weights have the expression,

$$(\tilde{C}_0^3, \tilde{C}_1^3, \tilde{C}_2^3) = \mathbf{C_0^4} + O(h^{2t}) = (2C_0^3, C_1^3, 0) + O(h^{2t}), \tag{3.19}$$

Exactly the same conclusions can be reached if a jump discontinuity in the function is found in the interval $(x_{j+1}, x_{j+2})$. The only difference is that in this case $I_1^4$ and $I_0^5$ are both $O(1)$ and,

$$\tilde{w}_0^4 = \frac{1}{(\epsilon+I_0^4)^t}\frac{1}{\displaystyle\sum_{j=2}^{r}\sum_{i=0}^{r-j}\frac{1}{(\epsilon+I_i^{j+r-1})^t}} = \frac{1}{(\epsilon+I_0^4)^t}\frac{1}{\frac{1}{(\epsilon+I_0^4)^t}(1+O(h^{4t}))} = \frac{1}{1+O(h^{4t})}$$
$$= 1+O(h^{4t}),$$
$$\tilde{w}_1^4 = \frac{1}{(\epsilon+I_1^5)^t}\frac{1}{\displaystyle\sum_{j=2}^{r}\sum_{i=0}^{r-j}\frac{1}{(\epsilon+I_i^{j+r-1})^t}} = \frac{1}{(\epsilon+I_1^4)^t}\frac{1}{\frac{1}{(\epsilon+I_0^4)^t}(1+O(h^{4t}))} = O(h^{4t}),$$
$$\tilde{w}_0^5 = \frac{1}{(\epsilon+I_0^5)^t}\frac{1}{\displaystyle\sum_{j=2}^{r}\sum_{i=0}^{r-j}\frac{1}{(\epsilon+I_i^{j+r-1})^t}} = \frac{1}{(\epsilon+I_0^5)^t}\frac{1}{\frac{1}{(\epsilon+I_0^4)^t}(1+O(h^{4t}))} = O(h^{4t}).$$

Then, the adapted optimal weights have the expression,

$$(\tilde{C}_0^3,\tilde{C}_1^3,\tilde{C}_2^3) = \mathbf{C_0^4} + O(h^{4t}) = (2C_0^3, C_1^3, 0) + O(h^{4t}). \tag{3.20}$$

If the discontinuity is placed in the interval $(x_{j-3}, x_{j-2})$, the conclusions would be exactly the same but

$$(\tilde{C}_0^3,\tilde{C}_1^3,\tilde{C}_2^3) = \mathbf{C_1^4} + O(h^{2t}) = (0, C_1^3, 2C_2^3) + O(h^{2t}), \tag{3.21}$$

for a kink, or

$$(\tilde{C}_0^3,\tilde{C}_1^3,\tilde{C}_2^3) = \mathbf{C_1^4} + O(h^{4t}) = (0, C_1^3, 2C_2^3) + O(h^{4t}), \tag{3.22}$$

for a jump discontinuity.
Now, let's see what we obtain using WENO algorithm with these adapted optimal weights instead of the original optimal weights (2.10). If there is a kink in the interval $(x_{j+1}, x_{j+2})$ we know that $I_0^3 = O(h^4)$, $I_1^3 = O(h^4)$ and $I_2^3 = O(h^2)$. If we assume that $\epsilon$ is small enough, we suppose that we have obtained as nonlinear optimal weights $(\tilde{C}_0^3,\tilde{C}_1^3,\tilde{C}_2^3) = (2C_0^3, C_1^3, 0) + O(h^{2t})$ as shown in (3.19) and we take into account that, in this case, $\tilde{C}_0^3+\tilde{C}_1^3+\tilde{C}_2^3 = 2C_0^3 + C_1^3 + O(h^{2t}) = 1 + O(h^{2t})$, then,

(3.23)
$$\omega_0^3 = \frac{\tilde{C}_0^3}{(\epsilon+I_0^3)^t}\frac{1}{\displaystyle\sum_{i=0}^{r-1}\frac{\tilde{C}_i^3}{(\epsilon+I_i^3)^t}} = \frac{\tilde{C}_0^3}{(\epsilon+I_0^3)^t}\frac{1}{\frac{1}{(\epsilon+I_0^3)^t}(\tilde{C}_0^3+\tilde{C}_1^3(1+O(h^2))+O(h^{2t}))}$$
$$= \frac{\tilde{C}_0^3}{1+O(h^2)} = \tilde{C}_0^3 + O(h^2),$$

$$\omega_1^3 = \frac{\tilde{C}_1^3}{(\epsilon+I_1^3)^t}\frac{1}{\sum_{i=0}^{r-1}\frac{\tilde{C}_i^3}{(\epsilon+I_i^3)^t}} = \frac{\tilde{C}_1^3}{(\epsilon+I_1^3)^t}\frac{1}{\frac{1}{(\epsilon+I_1^3)^t}(\tilde{C}_0^3(1+O(h^2))+\tilde{C}_1^3+O(h^{2t}))}$$
$$= \frac{\tilde{C}_1^3}{1+O(h^2)} = \tilde{C}_1^3 + O(h^2),$$
$$\omega_2^3 = \frac{\tilde{C}_2^3}{(\epsilon+I_2^3)^t}\frac{1}{\sum_{i=0}^{r-1}\frac{\tilde{C}_i^3}{(\epsilon+I_i^3)^t}} = \frac{\tilde{C}_2^3}{(\epsilon+I_2^3)^t}\frac{1}{\frac{1}{(\epsilon+I_0^3)^t}(\tilde{C}_0^3+\tilde{C}_1^3(1+O(h^2))+O(h^{2t}))}$$
$$= \frac{\tilde{C}_2^3}{O(h^{2t})}\frac{1}{\frac{1}{O(h^{4t})}(1+O(h^2))} = O(h^{2t}).$$

If there is a jump, the analysis is analogous and

$$\omega_0^3 = \tilde{C}_0^3 + O(h^4), \quad \omega_1^3 = \tilde{C}_1^3 + O(h^4), \quad \omega_2^3 = O(h^{4t}). \tag{3.24}$$

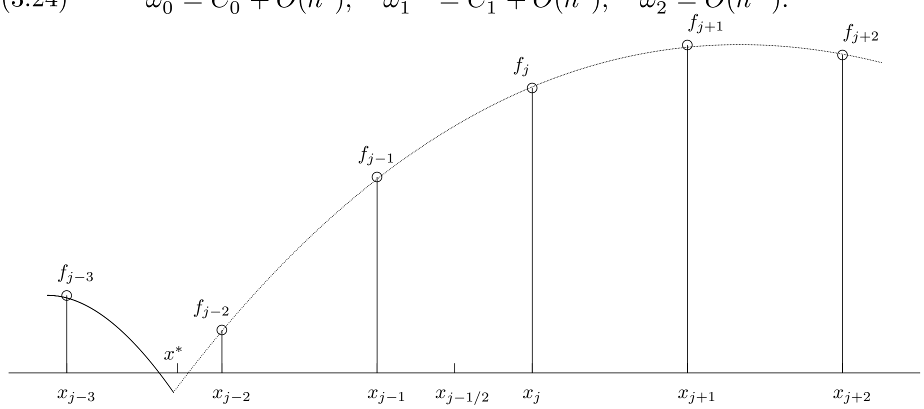


FIG. 5. *Representation of a kink placed at $x^*$ in the interval $(x_{j-3}, x_{j-2})$.*

- If there is a kink or a jump in the function affecting the stencil in the intervals $(x_j, x_{j+1})$ or $(x_{j-2}, x_{j-1})$, there is no hope of attaining adaption through the modification of the optimal weights. The best order of accuracy that can be obtained is $O(h^4)$, the same as the classical WENO algorithm attains, as there is only one smooth stencil. In this case the adapted optimal weights (3.12) would be,

$$\tilde{\omega}_0^4 = O(1), \quad \tilde{\omega}_0^5 = O(1), \quad \tilde{\omega}_1^4 = O(1). \tag{3.25}$$

In this situation, basically it is WENO strategy who decides the weights for each polynomial in (2.2). Let's see how WENO algorithm will behave. Let's analyze the case when the discontinuity is placed in the interval $(x_j, x_{j+1})$ as the case when the discontinuity is in the interval $(x_{j-2}, x_{j-1})$ is symmetric. As we did before, we can apply the same process to the WENO weights in (2.10), using as optimal weights those in (3.25). We know that we obtain,

$$\omega_j^3 = \frac{\tilde{\alpha}_j^3}{\sum_{i=0}^{r-1}\tilde{\alpha}_i^3} = \frac{O(1)}{(\epsilon+I_j^3)^t}\frac{1}{\sum_{i=0}^{r-1}\frac{O(1)}{(\epsilon+I_i^3)^t}}, \quad j=0,1,2,$$

and that $I_0^3 = O(h^4)$, $I_1^3 = O(h^2)$ and $I_2^3 = O(h^2)$. If we assume that $\epsilon$ is small enough, and we suppose that we have obtained as nonlinear optimal weights $(\tilde{C}_0^3, \tilde{C}_1^3, \tilde{C}_2^3)$ then,

(3.26)

$$\omega_0^3 = \frac{\tilde{C}_0^3}{(\epsilon + I_0^3)^t} \frac{1}{\displaystyle\sum_{i=0}^{r-1} \frac{\tilde{C}_i^3}{(\epsilon + I_i^3)^t}} = \frac{\tilde{C}_0^3}{(\epsilon + I_0^3)^t} \frac{1}{\frac{\tilde{C}_0^3}{(\epsilon+I_0^3)^t}(1 + O(h^{2t}))} = 1 + O(h^{2t}),$$

$$\omega_1^3 = \frac{\tilde{C}_1^3}{(\epsilon + I_1^3)^t} \frac{1}{\displaystyle\sum_{i=0}^{r-1} \frac{\tilde{C}_i^3}{(\epsilon + I_i^3)^t}} = \frac{\tilde{C}_1^3}{(\epsilon + I_1^3)^t} \frac{1}{\frac{\tilde{C}_0^3}{(\epsilon+I_0^3)^t}(1 + O(h^{2t}))}$$

$$= \frac{\tilde{C}_1^3}{O(h^{2t})} \frac{1}{\frac{\tilde{C}_0^3}{O(h^{4t})}(1 + O(h^{2t}))} = O(h^{2t}),$$

$$\omega_2^3 = \frac{\tilde{C}_2^3}{(\epsilon + I_2^3)^t} \frac{1}{\displaystyle\sum_{i=0}^{r-1} \frac{\tilde{C}_i^3}{(\epsilon + I_i^3)^t}} = \frac{\tilde{C}_2^3}{(\epsilon + I_2^3)^t} \frac{1}{\frac{\tilde{C}_0^3}{(\epsilon+I_0^3)^t}(1 + O(h^{2t}))}$$

$$= \frac{\tilde{C}_2^3}{O(h^{2t})} \frac{1}{\frac{\tilde{C}_0^3}{O(h^{4t})}(1 + O(h^{2t}))} = O(h^{2t}),$$

and the result is that the first stencil of WENO algorithm receives a weight that is very close to 1 while the others are close to 0. If there is a jump discontinuity in the interval $(x_j, x_{j+1})$, the analysis is analogous and

$$\omega_0^3 = 1 + O(h^{4t}), \quad \omega_1^3 = O(h^{4t}), \quad \omega_2^3 = O(h^{4t}). \tag{3.27}$$

- Using the the new algorithm, it is clear that the hypothetical situation presented in Figure 6 will result in a loss of accuracy when the discontinuity is placed at the central interval of the stencil.

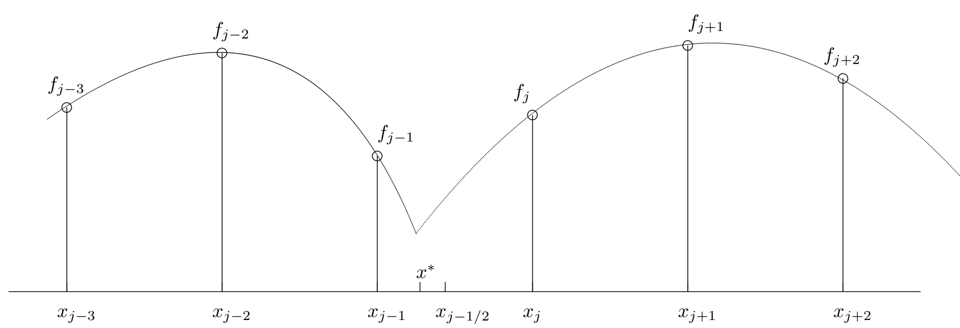


FIG. 6. *Representation of a kink placed placed at $x^*$ in the interval $(x_{j-1}, x_j)$.*

Let's consider the stencil $S_r^{2r} = \{x_{j-3}, x_{j-2}, x_{j-1}, x_j, x_{j+1}, x_{j+2}\}$ and the point values $\{f_{j-3}, f_{j-2},$ $f_{j-1}, f_j, f_{j+1}, f_{j+2}\}$. Now, we can prove the following theorem about the weights, that will also provide us information about the value of $t$ and how small $\epsilon$ must be.

THEOREM 3.2. *Let's assume that $r = 3$, $t \geq 1$, $\epsilon \leq h^4$ and that the grid spacing $h$ is small enough such that there is only one discontinuity in the considered stencil. In this situation we can take into account the four following situations:*

- *If the nonlinear optimal weights satisfy the following relation at smooth zones where,*

$$(\tilde{C}_0^3, \tilde{C}_1^3, \tilde{C}_2^3) = \tilde{\omega}_0^4 \mathbf{C_0^4} + \tilde{\omega}_0^5 \mathbf{C_0^5} + \tilde{\omega}_1^4 \mathbf{C_1^4} = \mathbf{C_0^5} + O(h^2),$$

  *with* $\mathbf{C_0^5} = (C_0^3, C_1^3, C_2^3)$, *then* $\sum_{k=0}^{r-1} \omega_k^r p_k^r(x_{j-1/2}) = f(x_{j-1/2}) + O(h^6)$.
- *If there is a discontinuity in the interval* $(x_{j-3}, x_{j-2})$ *and the nonlinear optimal weights satisfy the following relation,*

$$(\tilde{C}_0^3, \tilde{C}_1^3, \tilde{C}_2^3) = \tilde{\omega}_0^4 \mathbf{C_0^4} + \tilde{\omega}_0^5 \mathbf{C_0^5} + \tilde{\omega}_1^4 \mathbf{C_1^4} = \mathbf{C_1^4} + O(h^2),$$

  *with* $\mathbf{C_1^4} = (0, C_1^3, 2C_2^3)$, *then* $\sum_{k=0}^{r-1} \omega_k^r p_k^r(x_{j-1/2}) = f(x_{j-1/2}) + O(h^5)$.
- *If there is a discontinuity in the interval* $(x_{j+1}, x_{j+2})$ *and the nonlinear optimal weights satisfy the following relation,*

$$(\tilde{C}_0^3, \tilde{C}_1^3, \tilde{C}_2^3) = \tilde{\omega}_0^4 \mathbf{C_0^4} + \tilde{\omega}_0^5 \mathbf{C_0^5} + \tilde{\omega}_1^4 \mathbf{C_1^4} = \mathbf{C_0^4} + O(h^2),$$

  *with* $\mathbf{C_0^4} = (2C_0^3, C_1^3, 0)$, *then* $\sum_{k=0}^{r-1} \omega_k^r p_k^r(x_{j-1/2}) = f(x_{j-1/2}) + O(h^5)$.
- *If there is a discontinuity in the intervals* $(x_j, x_{j+1})$ *or* $(x_{j-2}, x_{j-1})$, *then the nonlinear optimal weights satisfy the following relation,*

$$(\tilde{C}_0^3, \tilde{C}_1^3, \tilde{C}_2^3) = \tilde{\omega}_0^4 \mathbf{C_0^4} + \tilde{\omega}_0^5 \mathbf{C_0^5} + \tilde{\omega}_1^4 \mathbf{C_1^4} = (O(1), O(1), O(1)),$$

  *and* $\sum_{k=0}^{r-1} \omega_k^r p_k^r(x_{j-1/2}) = f(x_{j-1/2}) + O(h^4)$.

*Proof.*

- Let's prove the first statement of the theorem. As shown in (3.9), the components of the vector $\mathbf{C_0^5}$ are the $C_k^r(j)$ in (2.7). We can start by writing the error of interpolation obtained by the expression in (2.2) at $x_{j-1/2}$,

$$\begin{aligned}\sum_{k=0}^{r-1} \omega_k^r p_k^r(x_{j-1/2}) - f_{j-1/2} = &\sum_{k=0}^{r-1} \omega_k^r p_k^r(x_{j-1/2}) - f_{j-1/2} + \sum_{k=0}^{r-1} C_k^r p_k^r(x_{j-1/2}) \\ &- \sum_{k=0}^{r-1} C_k^r p_k^r(x_{j-1/2}),\end{aligned}$$

  where the $C_k^r$ are the WENO optimal weights in (2.7). Grouping terms we obtain,

$$\begin{aligned}\sum_{k=0}^{r-1} \omega_k^r p_k^r(x_{j-1/2}) - f_{j-1/2} &= \sum_{k=0}^{r-1} \omega_k^r p_k^r(x_{j-1/2}) - \sum_{k=0}^{r-1} C_k^r p_k^r(x_{j-1/2}) \\ &+ \sum_{k=0}^{r-1} C_k^r p_k^r(x_{j-1/2}) - f_{j-1/2} \\ &= \sum_{k=0}^{r-1} (\omega_k^r - C_k^r) p_k^r(x_{j-1/2}) + O(h^{2r}).\end{aligned}$$

And due to the fact that $\sum_{k=0}^{r-1}\omega_k^r = \sum_{k=0}^{r-1} C_k^r = 1$,

$$\begin{aligned}\sum_{k=0}^{r-1}\omega_k^r p_k^r(x_{j-1/2}) - f_{j-1/2} &= \sum_{k=0}^{r-1}(\omega_k^r - C_k^r)p_k^r(x_{j-1/2}) + O(h^{2r})\\ &+ \sum_{k=0}^{r-1}(\omega_k^r - C_k^r)f_{j-1/2}\\ &= \sum_{k=0}^{r-1}(\omega_k^r - C_k^r)(p_k^r(x_{j-1/2}) - f_{j-1/2}) + O(h^{2r})\end{aligned}$$

From (3.17) it turns out that $(\omega_k^r - C_k^r) = O(h^m)$ with $m = 2$,

$$\sum_{k=0}^{r-1}\omega_k^r p_k^r(x_{j-1/2}) - f_{j-1/2} = O(h^m)O(h^{r+1}) + O(h^{2r}) = O(h^{\min(m+r+1,2r)}). \tag{3.28}$$

For the particular case $r = 3$, we obtain optimal accuracy $O(h^6)$ at smooth zones.

- The second and third statements of the theorem are symmetric so let's prove only the second statement. As shown in (3.8), $\mathbf{C_1^4} = (0, C_1^3, 2C_2^3)$. We can reproduce the proof in the previous point and write the error of interpolation obtained in this case by the expression in (2.2) at $x_{j-1/2}$,

$$\begin{aligned}\sum_{k=0}^{r-1}\omega_k^r p_k^r(x_{j-1/2}) - f_{j-1/2} &= \sum_{k=0}^{r-1}\omega_k^r p_k^r(x_{j-1/2}) - f_{j-1/2} + \sum_{k=0}^{r-1}\mathbf{C_1^4}(k)p_k^r(x_{j-1/2})\\ &- \sum_{k=0}^{r-1}\mathbf{C_1^4}(k)p_k^r(x_{j-1/2}),\end{aligned}$$

where the $\mathbf{C_1^4}$ are the WENO optimal weights that would provide $O(h^5)$ accuracy in this case, as shown in (3.8). Grouping terms we obtain,

$$\begin{aligned}\sum_{k=0}^{r-1}\omega_k^r p_k^r(x_{j-1/2}) - f_{j-1/2} &= \sum_{k=0}^{r-1}\omega_k^r p_k^r(x_{j-1/2}) - \sum_{k=0}^{r-1}\mathbf{C_1^4}(k)p_k^r(x_{j-1/2})\\ &+ \sum_{k=0}^{r-1}\mathbf{C_1^4}(k)p_k^r(x_{j-1/2}) - f_{j-1/2}\\ &= \sum_{k=0}^{r-1}(\omega_k^r - \mathbf{C_1^4}(k))p_k^r(x_{j-1/2}) + O(h^{2r-1}).\end{aligned}$$

And due to the fact that $\sum_{k=0}^{r-1}\omega_k^r = \sum_{k=0}^{r-1}\mathbf{C_1^4}(k) = 1$,

$$\begin{aligned}\sum_{k=0}^{r-1}\omega_k^r p_k^r(x_{j-1/2}) - f_{j-1/2} &= \sum_{k=0}^{r-1}(\omega_k^r - \mathbf{C_1^4}(k))(p_k^r(x_{j-1/2}) - f_{j-1/2}) + O(h^{2r-1})\\ &= O(h^m)O(h^{r+1}) + O(h^{2r-1}) = O(h^{\min(m+r+1,2r-1)}),\end{aligned} \tag{3.29}$$

From (3.21) and (3.23), if $t \geq 1$, it turns out that if $(\omega_k^r - \mathbf{C_1^4}(k)) = O(h^m)$ with $m = 2$ for kinks and $m = 4$ for jumps. Thus, for the particular case

$r=3$, we obtain optimal accuracy $O(h^5)$ for the situation presented in Figure 5.

- The proof of the fourth statement of the theorem corresponds to the case when there is a discontinuity in the interval $(x_j, x_{j+1})$ or $(x_{j-2}, x_{j-1})$. In this case there is only one smooth substencil of four points and WENO algorithm reaches the maximum accuracy without any modification. Lets consider the case analyzed in (3.26) when the discontinuity is in the interval $(x_j, x_{j+1})$ as the other case is symmetric. From (3.26) we can consider the vector $\mathbf{C}=(1,0,0)$. Following the same process as before, we have that,

$$\sum_{k=0}^{r-1}\omega_k^r p_k^r(x_{j-1/2}) - f_{j-1/2} = \sum_{k=0}^{r-1}\omega_k^r p_k^r(x_{j-1/2}) - f_{j-1/2} + \sum_{k=0}^{r-1}\mathbf{C}(k)p_k^r(x_{j-1/2}) - \sum_{k=0}^{r-1}\mathbf{C}(k)p_k^r(x_{j-1/2}),$$

Grouping terms we obtain,

$$\begin{aligned}\sum_{k=0}^{r-1}\omega_k^r p_k^r(x_{j-1/2}) - f_{j-1/2} &= \sum_{k=0}^{r-1}\omega_k^r p_k^r(x_{j-1/2}) - \sum_{k=0}^{r-1}\mathbf{C}(k)p_k^r(x_{j-1/2}) \\ &+ \sum_{k=0}^{r-1}\mathbf{C}(k)p_k^r(x_{j-1/2}) - f_{j-1/2} \\ &= \sum_{k=0}^{r-1}(\omega_k^r - \mathbf{C}(k))p_k^r(x_{j-1/2}) + O(h^{r+1}).\end{aligned}$$

And due to the fact that $\sum_{k=0}^{r-1}\omega_k^r = \sum_{k=0}^{r-1}\mathbf{C}(k) = 1$,

$$\begin{aligned}\sum_{k=0}^{r-1}\omega_k^r p_k^r(x_{j-1/2}) - f_{j-1/2} &= \sum_{k=0}^{r-1}(\omega_k^r - \mathbf{C}(k))(p_k^r(x_{j-1/2}) - f_{j-1/2}) + O(h^{r+1}) \\ &= O(h^m)O(h^{r+1}) + O(h^{r+1}) = O(h^{\min(m+r+1,r+1)}).\end{aligned} \tag{3.30}$$

From (3.26) and (3.27) it is clear that $m=2t$ for kinks and $m=4t$ for jumps. Thus, for the particular case $r=3$ with $t\geq 1$, we obtain optimal accuracy $O(h^4)$ if we find a discontinuity in the intervals $(x_j, x_{j+1})$ or $(x_{j-2}, x_{j-1})$. □

The previous theorem leads to the following corollary

COROLLARY 3.3. *Considering the initial hypothesis $r=3$, $t\geq 1$, and $\epsilon \leq h^4$ the new WENO interpolant is at least as good as WENO interpolant close to discontinuities.*

*Proof.* The proof is straightforward from the hypothesis and the proof of previous theorem. It basically consists in comparing the order of accuracy that WENO would obtain with the accuracy that the new WENO method obtains. In order to do this, we can just follow the proof of the previous theorem:

- If there is no discontinuity affecting the stencil, for $r=3$ WENO obtains $O(h^6)$ accuracy and from (3.28) the new WENO algorithm also obtains $O(h^6)$ accuracy.

- If there is a discontinuity in the interval $(x_{j-3}, x_{j-2})$ or $(x_{j+1}, x_{j+2})$, WENO algorithm typically obtains $O(h^4)$ accuracy. From (3.29) the new WENO algorithm obtains $O(h^5)$ accuracy with $t \geq 1$.
- If there is a discontinuity in the intervals $(x_j, x_{j+1})$ or $(x_{j-2}, x_{j-1})$, WENO algorithm obtains $O(h^4)$ accuracy. From (3.30) the new WENO algorithm obtains $O(h^4)$ accuracy with $t \geq 1$.

A *small enough* value of $\epsilon$ in (3.13) and in (2.10) is a value of order $O(h^4)$, as this is the minimum value of the new smoothness indicators (3.4), (3.5) and (3.6), that is reached at smooth zones, as can be seen from Theorem 3.1. □

**3.3. Increasing the accuracy at the central interval of the stencil in the point values.** In this subsection we will analyze how to increase the accuracy attained by WENO method when a kink is placed at the central interval of the stencil, as shown in Figure 6. It is important to remember that in the point value setting, the position of jump discontinuities is lost during the discretization process and we can not hope to localize their exact position [38]. In [1] we extend the algorithm presented in this article for working in the cell averages and to the solution of conservation laws. We can use the smoothness indicators of three points shown in (3.3) in order to detect the presence of a kink in the central interval of the big stencil. If we use these smoothness indicators and a kink is placed in the interval $(x_{j-1}, x_j)$, then the first substencil $S^2_{-1} = \{x_{j-3}, x_{j-2}, x_{j-1}\}$ and the fourth substencil $S^2_2 = \{x_j, x_{j+1}, x_{j+2}\}$ are smooth and $I^2_{-1}$ and $I^2_2$ take a value that is $O(h^4)$, while the second and third stencils $S^2_0 = \{x_{j-2,} x_{j-1}, x_j\}$, $S^2_1 = \{x_{j-1}, x_j, x_{j+1}\}$ are not smooth so $I^2_0$ and $I^2_1$ take value that is $O(h^2)$. This is a hint that should lead us to think that there is a kink at the interval $(x_{j-1}, x_j)$. For isolated discontinuities, we will have the following cases:

- If there is not a discontinuity in the interval $(x_{j-1}, x_j)$, then $I^2_{-1}, I^2_0, I^2_1$ and $I^2_2$ are $O(h^4)$.
- If there is a discontinuity in the interval $(x_{j-1}, x_j)$, then $I^2_{-1}$ and $I^2_2$ are $O(h^4)$ and $I^2_0$ and $I^2_1$ are $O(h^2)$.
- If there is a discontinuity at $x_{j-1}$, then $I^2_{-1}, I^2_1$ and $I^2_2$ are $O(h^4)$ and $I^2_0$ is $O(h^2)$.
- If there is a discontinuity at $x_j$, then $I^2_{-1}, I^2_0$ and $I^2_2$ are $O(h^4)$ and $I^2_1$ is $O(h^2)$.

Our objective is to localize the position of the discontinuity and, depending on its position with respect to $x_{j-1/2}$, then extrapolate at $x_{j-1/2}$ using $S^2_{-1}$ or $S^2_2$. This process is inspired by Harten's ENO subcell resolution algorithm. Due to the bigger errors associated to the extrapolation process we would like to use it only when it is strictly necessary. Moreover, the extrapolation process that we propose reduces the stencil in order to avoid the discontinuity and, hence, implies order reduction if the location process fails and detects a discontinuity at a smooth zone. Thus, we only want to apply it at real singularities.

Being $h$ the grid spacing, when $h(I^2_0 + I^2_1) > I^2_{-1} + I^2_2$ the interval $(x_{j-1}, x_j)$ is considered suspicious of containing a discontinuity. In this case, we build the second order interpolating polynomial $p^2_0(x)$, using the data $\{f_{j-3}, f_{j-2}, f_{j-1}\}$, that corresponds to the stencil $S^2_0$, and the second order interpolating polynomial $p^2_3(x)$, using the data $\{f_j, f_{j+1}, f_{j+2}\}$, that correspond to $S^2_3$. Then we build the function $g(x) = p^2_0(x) - p^2_3(x)$. Supposing that there is only one zero of $g(x)$ inside the interval $(x_{j-1}, x_j)$, that zero corresponds to the position of the discontinuity with $O(h^3)$ accuracy. Even though, we do not need to find the zeros of $g(x)$ but only to know if one of them is placed in the interval $(x_{j-1}, x_{j-1/2})$ or in the interval $(x_{j-1/2}, x_j)$. Evaluating $g(x)$ at $x_{j-1}, x_{j-1/2}$ and $x_j$ and using Bolzano's theorem we can know in which of the previous subintervals we can find the discontinuity, if there is one. If the discontinuity is placed in the interval $(x_{j-1}, x_{j-1/2})$, we will use $p_3^2(x)$ to extrapolate at $x_{j-1/2}$ in order to obtain an $O(h^3)$ approximation. On the other hand, if the discontinuity is placed in the interval $(x_{j-1/2}, x_j)$, we will use $p_0^2(x)$ to extrapolate at $x_{j-1/2}$. This technique assures $O(h^3)$ accuracy when a kink is placed in the central interval of the stencil. This process is somewhat similar to the one described by Harten in [41] to construct ENO *subcell-resolution* algorithm for conservation laws. Of course, the grid must be fine enough so that the discontinuity can be detected. Thus, it must be assured that the grid spacing is below a critical value $h_c$ that guarantees the detection of the singularity. The smoothness indicators used in this work are based on second order differences, which are the base of the detection algorithm in [38]. As a consequence, the value of the critical grid spacing $h_c$ can be directly taken from Section 4, Lemma 2 of [38]. The interested reader can refer to [38] for a deeper explanation of this point.

**3.4. ENO property.** It is important to remember that the technique presented in Section 3.2 or 3.3 is basically a WENO algorithm where we modify the original optimal weights in order to assure the maximum possible order close to discontinuities.

The new WENO technique assures that the resulting polynomial satisfies the following properties:

- It is a piecewise polynomial interpolation composed of polynomials of even degree $r$.
- Every polynomial must satisfy the so-called *essentially non-oscillatory property*, through the emulation of the ENO algorithm [14]:
  - If the function $f$ is smooth at the stencil $S_k^r$, then the weight related to this stencil will verify $w_k^r = O(1)$.
  - If the function $f$ has a singularity at the stencil $S_k^r$, then the corresponding $w_k^r$ will verify $w_k^r \leq O\left(h^r\right)$.

If the weights $w_k^r$ that appear in (2.3) are designed to satisfy the ENO property, then $q_{j-r}(x)$ in (2.2) is a nonlinear convex combination of polynomials built using smooth stencils and where the contribution of stencils crossing discontinuities is negligible.

THEOREM 3.4. *The new algorithm satisfies the ENO property for $t \geq 2$, satisfying at the same time Theorem 3.2.*

*Proof.* For $t \geq 1$ Theorem 3.2 is satisfied, so for $t \geq 2$ it is also satisfied. From (3.18), (3.23) and (3.26) we can see that for $t \geq 2$:

- If the function $f$ is smooth at the stencil $S_k^r$, then the weight related to this stencil will verify $w_k^r = O(1)$.
- If the function $f$ has a singularity at the stencil $S_k^r$, then the corresponding $w_k^r$ will verify $w_k^r \leq O\left(h^r\right)$.

This is precisely the ENO property. ∎

**4. Numerical experiments.** In this section we have used the functions plotted in Figure 7 and presented in (4.1), (4.2) and (4.3). The function in (4.1) is a piecewise polynomial of degree eight. The function in (4.2) is the product of two sinusoidal functions plus a polynomial. The function in (4.3) presents a jump discontinuity. We have used a stencil of six points, i.e. $r = 3$ in (2.1), so no one of the functions proposed can be interpolated exactly. Following Corollary 3.3 and Subsection 3.4, we have chosen the parameters $t = 2$ and $\epsilon \leq h^4$ for the new algorithm shown in Subsection 3.2. For WENO it is enough to choose $t = 2$ and $\epsilon \leq h^2$, as shown in [15]. We have used in all the experiments the smoothness indicators proposed in (3.1).

In all the experiments presented, in order to obtain lower resolution versions of the initial data, we start from a discretized version at a higher resolution and then we take one of every $2^n$ samples. Using interpolation we recover a high resolution approximation of the original data. We have chosen to interpolate at the odd knots.

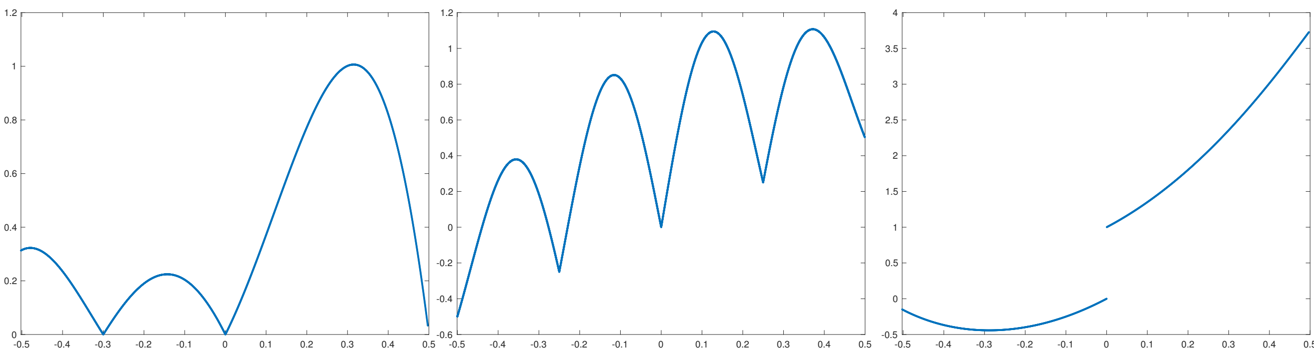

FIG. 7. *In this figure we represent the functions (4.1), (4.2) and (4.3) that will be used for the numerical experiments of this section.*

**Example 1** Let's start with the function plotted in Figure (7) to the left,

$$f(x) = |(x-3)(x-1.5)(x-0.5)(x+\eta)(x+0.3)(x+0.6)(x+5)(x+1.5)|, \tag{4.1}$$

for $-0.5-\xi \leq x < 0.5-\xi$. Setting grid spacing to $h_i = \frac{1}{2^i}, i = 6, 7, \cdots, 12$, we check the accuracy of the interpolation through a grid refinement analysis close to the discontinuity at $x = 0$. In order to obtain the error, we compare with the function discretized with $h_{i+1} = \frac{1}{2^{i+1}}$. The worst case is when the discontinuity does not fall in a grid point (otherwise, there is always a smooth stencil). In order to assure the worse case for all the discretizations used, we place the discontinuity around which we will do the grid refinement analysis at $x = -\eta$, with $\eta = (2/3)h_{13}$ and we place the left side of the interval at $x = -0.5-\xi$ with $\xi = 10h_{13}$. These considerations are only taken for doing the grid refinement analysis and would not be necessary in a real application of the algorithm.

We consider the errors at the nodes $\{x_{2j-6}, x_{2j-4}, x_{2j-2}, x_{2j}, x_{2j+2}, x_{2j+4}, x_{2j+6}\}$, being $x_{2j}$ the prediction at the interval that contains the discontinuity (in this first experiment close to $x = 0$). Table 1 shows the results obtained by WENO algorithm using the smoothness indicator proposed in (3.1) and whose expressions are shown in (3.4). Table 2 shows the results obtained at the same points for the new optimal weights, described in Subsection 3.2, and WENO. The two tables show the order of accuracy between the successive errors when refining the grid. We can see how the accuracy is lost by both algorithms at the interval that contains the discontinuity. Also, as explained in Subsection 3.2, WENO is designed to obtain optimal order at smooth zones and to eliminate spurious oscillations close to discontinuities, but not optimizing the order in this last case. This fact can be seen in Table 1 at $x_{2j-4}$ and $x_{2j+4}$. In both cases there are two smooth stencils, containing in total 5 points belonging to the same side of the singularity. This means that the maximum theoretical accuracy that can be obtained is $O(h^5)$ and WENO algorithm obtains $O(h^4)$. As it can be analyzed in Table 2, using the new optimal weights presented in Subsection 3.2, we attain the maximum theoretical accuracy in the whole interval except at the interval that contains the singularity. Table 3 shows the results obtained by the new algorithm proposed in Subsection 3.3. We can see how the algorithm reproduces the behavior of the algorithm presented in Subsection 3.2 in terms of accuracy, at the intervals that are close to the discontinuity but do not contain it. At the interval that contains the discontinuity we have managed to raise the accuracy using the strategy inspired by ENO-Subcell resolution algorithm that was presented in Subsection 3.3. Figure 8 shows the absolute error distribution for the three algorithms when interpolating the function in (4.1) using $2^{12} = 4096$ samples. To the left we can see the error obtained by WENO algorithm, at the center the error obtained using the new weights presented in Subsection 3.2 and to the right the error obtained by the new algorithm presented in Subsection 3.3. We can see how the error presented in this last plot is six orders of magnitude smaller than in the other two plots.

| | $x_{2j-6}$ | | $x_{2j-4}$ | | $x_{2j-2}$ | | $x_{2j}$ | | $x_{2j+2}$ | | $x_{2j+4}$ | | $x_{2j+6}$ | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $i$ | $e_i$ | $\log_2\left(\frac{e_i}{e_{i+1}}\right)$ | $e_i$ | $\log_2\left(\frac{e_i}{e_{i+1}}\right)$ | $e_i$ | $\log_2\left(\frac{e_i}{e_{i+1}}\right)$ | $e_i$ | $\log_2\left(\frac{e_i}{e_{i+1}}\right)$ | $e_i$ | $\log_2\left(\frac{e_i}{e_{i+1}}\right)$ | $e_i$ | $\log_2\left(\frac{e_i}{e_{i+1}}\right)$ | $e_i$ | $\log_2\left(\frac{e_i}{e_{i+1}}\right)$ |
| 6 | 1.848e-08 | - | 4.867e-07 | - | 2.031e-06 | - | 1.076e-03 | - | 5.188e-05 | - | 1.475e-05 | - | 2.299e-08 | - |
| 7 | 2.943e-10 | 5.973 | 2.972e-08 | 4.034 | 1.271e-07 | 3.998 | 1.075e-03 | 0.001 | 1.771e-06 | 4.873 | 2.325e-07 | 5.988 | 3.274e-10 | 6.133 |
| 8 | 4.668e-12 | 5.978 | 1.843e-09 | 4.011 | 7.934e-09 | 4.002 | 1.052e-03 | 0.032 | 1.901e-08 | 6.542 | 2.696e-09 | 6.430 | 4.914e-12 | 6.058 |
| 9 | 7.359e-14 | 5.987 | 1.148e-10 | 4.005 | 4.941e-10 | 4.005 | 1.017e-03 | 0.048 | 5.484e-10 | 5.115 | 1.178e-10 | 4.516 | 7.546e-14 | 6.025 |
| 10 | 1.154e-15 | 5.995 | 7.141e-12 | 4.007 | 2.974e-11 | 4.054 | 8.539e-04 | 0.253 | 3.132e-11 | 4.130 | 7.171e-12 | 4.038 | 1.171e-15 | 6.010 |
| 11 | 1.908e-17 | **5.918** | 4.482e-13 | 3.994 | 1.941e-12 | 3.938 | 1.491e-04 | 2.518 | 2.220e-12 | 3.818 | 4.673e-13 | 3.940 | 1.735e-17 | **6.077** |
| 12 | 0 | - | 2.800e-14 | **4.000** | 1.212e-13 | **4.001** | 1.532e-04 | **-0.039** | 1.227e-13 | **4.177** | 2.806e-14 | **4.058** | 1.301e-18 | 3.737 |

TABLE 1

*Grid refinement analysis for the smoothness indicators presented in (3.1) and WENO algorithm for the function in (4.1). We can see how the accuracy is reduced at the interval that contains the singularity and around it. At $x_{2j-4}$ and $x_{2j+4}$ there is enough information to obtain $O(h^5)$ accuracy, but WENO is not designed to optimize the accuracy close to the discontinuities.*

| | $x_{2j-6}$ | | $x_{2j-4}$ | | $x_{2j-2}$ | | $x_{2j}$ | | $x_{2j+2}$ | | $x_{2j+4}$ | | $x_{2j+6}$ | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $i$ | $e_i$ | $\log_2\left(\frac{e_i}{e_{i+1}}\right)$ | $e_i$ | $\log_2\left(\frac{e_i}{e_{i+1}}\right)$ | $e_i$ | $\log_2\left(\frac{e_i}{e_{i+1}}\right)$ | $e_i$ | $\log_2\left(\frac{e_i}{e_{i+1}}\right)$ | $e_i$ | $\log_2\left(\frac{e_i}{e_{i+1}}\right)$ | $e_i$ | $\log_2\left(\frac{e_i}{e_{i+1}}\right)$ | $e_i$ | $\log_2\left(\frac{e_i}{e_{i+1}}\right)$ |
| 6 | 1.848e-08 | - | 6.681e-09 | - | 2.052e-06 | - | 1.076e-03 | - | 3.048e-05 | - | 1.084e-05 | - | 2.299e-08 | - |
| 7 | 2.943e-10 | 5.973 | 1.240e-10 | 5.751 | 1.278e-07 | 4.005 | 1.077e-03 | -0.000 | 9.762e-07 | 4.964 | 3.377e-08 | 8.326 | 3.275e-10 | 6.133 |
| 8 | 4.668e-12 | 5.978 | 9.334e-12 | 3.732 | 7.961e-09 | 4.005 | 1.062e-03 | 0.020 | 1.457e-08 | 6.066 | 5.100e-11 | 9.371 | 4.914e-12 | 6.058 |
| 9 | 7.359e-14 | 5.987 | 3.789e-13 | 4.623 | 4.956e-10 | 4.006 | 1.034e-03 | 0.039 | 5.593e-10 | 4.703 | 6.597e-13 | 6.272 | 7.546e-14 | 6.025 |
| 10 | 1.154e-15 | 5.995 | 1.294e-14 | 4.872 | 2.952e-11 | 4.069 | 8.537e-04 | 0.276 | 3.125e-11 | 4.162 | 1.677e-14 | 5.298 | 1.171e-15 | 6.010 |
| 11 | 1.908e-17 | **5.918** | 4.454e-16 | 4.861 | 1.942e-12 | 3.926 | 1.511e-04 | 2.499 | 2.121e-12 | 3.881 | 5.594e-16 | 4.906 | 1.735e-17 | **6.077** |
| 12 | 0 | - | 1.409e-17 | **4.982** | 1.213e-13 | **4.001** | 1.540e-04 | **-0.028** | 1.232e-13 | **4.106** | 1.518e-17 | **5.204** | 8.674e-19 | 4.322 |

TABLE 2

*Grid refinement analysis for the new optimal weights presented in Subsection 3.2 and WENO algorithm for the function in (4.1). We can see how the accuracy is lost at the interval that contains the singularity, but it is controlled close to it, decreasing step by step as we move towards the singularity.*

| | $x_{2j-6}$ | | $x_{2j-4}$ | | $x_{2j-2}$ | | $x_{2j}$ | | $x_{2j+2}$ | | $x_{2j+4}$ | | $x_{2j+6}$ | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $i$ | $e_i$ | $\log_2\left(\frac{e_i}{e_{i+1}}\right)$ | $e_i$ | $\log_2\left(\frac{e_i}{e_{i+1}}\right)$ | $e_i$ | $\log_2\left(\frac{e_i}{e_{i+1}}\right)$ | $e_i$ | $\log_2\left(\frac{e_i}{e_{i+1}}\right)$ | $e_i$ | $\log_2\left(\frac{e_i}{e_{i+1}}\right)$ | $e_i$ | $\log_2\left(\frac{e_i}{e_{i+1}}\right)$ | $e_i$ | $\log_2\left(\frac{e_i}{e_{i+1}}\right)$ |
| 6 | 1.848e-08 | - | 6.681e-09 | - | 8.870e-05 | - | 1.416e-04 | - | 3.048e-05 | - | 1.084e-05 | - | 2.299e-08 | - |
| 7 | 2.943e-10 | 5.973 | 1.240e-10 | 5.751 | 1.278e-07 | 9.439 | 1.610e-05 | 3.136 | 9.762e-07 | 4.964 | 3.377e-08 | 8.326 | 3.275e-10 | 6.133 |
| 8 | 4.668e-12 | 5.978 | 9.334e-12 | 3.732 | 7.961e-09 | 4.005 | 1.911e-06 | 3.075 | 1.457e-08 | 6.066 | 5.100e-11 | 9.371 | 4.914e-12 | 6.058 |
| 9 | 7.359e-14 | 5.987 | 3.789e-13 | 4.623 | 4.956e-10 | 4.006 | 2.325e-07 | 3.039 | 5.593e-10 | 4.703 | 6.597e-13 | 6.272 | 7.546e-14 | 6.025 |
| 10 | 1.154e-15 | 5.995 | 1.294e-14 | 4.872 | 2.952e-11 | 4.069 | 2.810e-08 | 3.049 | 3.125e-11 | 4.162 | 1.677e-14 | 5.298 | 1.171e-15 | 6.010 |
| 11 | 1.908e-17 | **5.918** | 4.454e-16 | 4.861 | 1.942e-12 | 3.926 | 3.573e-09 | 2.975 | 2.121e-12 | 3.881 | 5.594e-16 | 4.906 | 1.735e-17 | **6.077** |
| 12 | 0 | - | 1.409e-17 | **4.982** | 1.213e-13 | **4.001** | 4.450e-10 | **3.005** | 1.232e-13 | **4.106** | 1.518e-17 | **5.204** | 8.674e-19 | 4.322 |

TABLE 3

*Grid refinement analysis for the algorithm presented in Subsection 3.3 for the function in (4.1). We can see how the accuracy is raised at the interval that contains the singularity and how the order decreases in a controlled way, step by step as we move towards the singularity.*

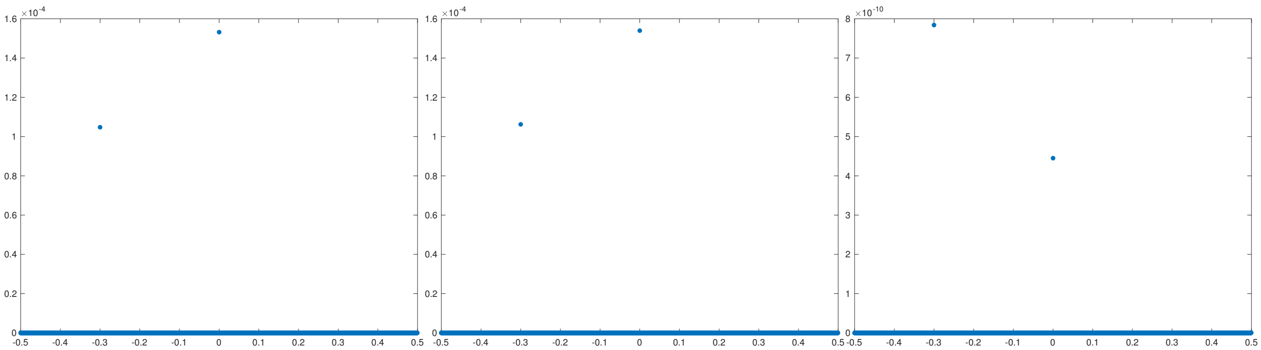

FIG. 8. *Absolute error obtained when reconstructing the function in (4.1) through WENO (left), through the algorithm presented in Subsection 3.2 (center) and through the algorithm presented in Subsection 3.3 (right). The original data was* 8192 *points and the subsampled data was* 4096 *points.*

**Example 2** Let's continue with the function plotted in Figure (7) at the center,

$$l(x) = |\sin(4\pi(x+\eta))| \cos((2(x+\eta)) + x, \quad -0.5-\xi \leq x < 0.5-\xi. \tag{4.2}$$

As before, we set $h_i = \frac{1}{2^i}, i = 6, 7, \cdots, 12$, in order to check the accuracy of the interpolation through a grid refinement analysis close to the singularity that is placed in the interval $(-0.3, -0.2)$. As before, in order to obtain the error we compare with the function discretized with $h_{i+1} = \frac{1}{2^{i+1}}$. As mentioned in the previous experiment, in order to assure that the singularities do not fall at a grid point, we shift the function by $\eta = (2/3)h_{13}$ and we place the left side of the interval at $x = -0.5 - \xi$ with $\xi = 3h_{13}$. Table 4 shows a grid refinement analysis for the results of WENO algorithm at the singularity placed in the interval $(-0.2, -0.3)$ of function in (4.2). The conclusions that we can reach for this experiment are the same as those we obtained for the previous experiment. We can clearly observe how the accuracy is reduced around the interval that contains the singularity, but not in an optimal way. Table 5 shows the results obtained for the same function but using WENO with the new weights introduced in Subsection 3.2. We can see that the accuracy also decreases around the central interval but, in this case, reducing the order one step at a time as we proceed towards the singularity. Table 6 shows the results obtained using the new algorithm introduced in Subsection 3.3. We can see that the order of accuracy is optimal, including the interval that contains the singularity. Figure 9 shows the absolute error distribution for the three algorithms when interpolating the function in (4.2) using $2^{12}$ samples. As before, we can see how the error presented in the plot to the right is several orders of magnitude smaller than the ones to the left and at the center.

**Example 3** Let's finish with the function plotted in Figure (7) to the right,

$$f(x) = \begin{cases} -4x^7 + x^4 + 5x^2 + 3x, & 0.5 \leq x < 0, \\ -8x^7 + x^4 + 5x^2 + 3x + 1, & 0 \leq x < 0.5, \end{cases} \tag{4.3}$$

In this case we have set again $h_i = \frac{1}{2^i}, i = 7, 8, 9 \cdots 11$, for the grid refinement analysis. The function in (4.3) only presents a jump discontinuity that is placed at $x = 0$. Table 7 shows a grid refinement analysis for the results obtained using the WENO algorithm. Table 8 shows the result obtained using the optimal weights presented in Subsection 3.2. In this case, the algorithm presented in Subsection 3.3 can not be applied, as the

| | $x_{2j-6}$ | | $x_{2j-4}$ | | $x_{2j-2}$ | | $x_{2j}$ | | $x_{2j+2}$ | | $x_{2j+4}$ | | $x_{2j+6}$ | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $i$ | $e_i$ | $\log_2\left(\frac{e_i}{e_{i+1}}\right)$ | $e_i$ | $\log_2\left(\frac{e_i}{e_{i+1}}\right)$ | $e_i$ | $\log_2\left(\frac{e_i}{e_{i+1}}\right)$ | $e_i$ | $\log_2\left(\frac{e_i}{e_{i+1}}\right)$ | $e_i$ | $\log_2\left(\frac{e_i}{e_{i+1}}\right)$ | $e_i$ | $\log_2\left(\frac{e_i}{e_{i+1}}\right)$ | $e_i$ | $\log_2\left(\frac{e_i}{e_{i+1}}\right)$ |
| 6 | 1.812e-06 | - | 6.743e-06 | - | 3.322e-05 | - | 9.801e-04 | - | 1.064e-04 | - | 1.735e-05 | - | 6.014e-07 | - |
| 7 | 2.312e-08 | 6.292 | 3.354e-07 | 4.329 | 1.634e-06 | 4.346 | 9.807e-04 | -0.001 | 1.591e-06 | 6.063 | 3.142e-07 | 5.787 | 2.348e-10 | 11.322 |
| 8 | 3.274e-10 | 6.142 | 1.862e-08 | 4.171 | 8.709e-08 | 4.230 | 9.810e-04 | -0.000 | 5.206e-08 | 4.934 | 1.198e-08 | 4.713 | 1.553e-10 | 0.596 |
| 9 | 4.889e-12 | 6.065 | 1.095e-09 | 4.088 | 4.958e-09 | 4.135 | 9.790e-04 | 0.003 | 3.677e-09 | 3.824 | 8.862e-10 | 3.756 | 3.753e-12 | 5.371 |
| 10 | 7.441e-14 | 6.038 | 6.637e-11 | 4.044 | 2.945e-10 | 4.073 | 9.591e-04 | 0.030 | 2.520e-10 | 3.867 | 5.976e-11 | 3.890 | 6.881e-14 | 5.770 |
| 11 | 1.027e-15 | **6.179** | 4.086e-12 | 4.022 | 1.792e-11 | 4.038 | 9.232e-04 | 0.055 | 1.658e-11 | 3.926 | 3.878e-12 | 3.946 | 1.443e-15 | **5.575** |
| 12 | 1.388e-16 | 2.888 | 2.537e-13 | **4.010** | 1.105e-12 | **4.019** | 7.755e-04 | **0.251** | 1.063e-12 | **3.963** | 2.467e-13 | **3.974** | 1.388e-16 | 3.379 |

TABLE 4

*Grid refinement analysis for the smoothness indicators presented in (3.1) and WENO algorithm for the function in (4.2). We can see how the accuracy is reduced at the interval that contains the singularity and around it. At $x_{2j-4}$ and $x_{2j+4}$ there is enough information to obtain $O(h^5)$ accuracy, but WENO is not designed to optimize the accuracy close to the singularities.*

| | $x_{2j-6}$ | | $x_{2j-4}$ | | $x_{2j-2}$ | | $x_{2j}$ | | $x_{2j+2}$ | | $x_{2j+4}$ | | $x_{2j+6}$ | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $i$ | $e_i$ | $\log_2\left(\frac{e_i}{e_{i+1}}\right)$ | $e_i$ | $\log_2\left(\frac{e_i}{e_{i+1}}\right)$ | $e_i$ | $\log_2\left(\frac{e_i}{e_{i+1}}\right)$ | $e_i$ | $\log_2\left(\frac{e_i}{e_{i+1}}\right)$ | $e_i$ | $\log_2\left(\frac{e_i}{e_{i+1}}\right)$ | $e_i$ | $\log_2\left(\frac{e_i}{e_{i+1}}\right)$ | $e_i$ | $\log_2\left(\frac{e_i}{e_{i+1}}\right)$ |
| 6 | 1.815e-06 | - | 2.339e-06 | - | 3.324e-05 | - | 9.801e-04 | - | 7.058e-05 | - | 1.682e-05 | - | 6.092e-07 | - |
| 7 | 2.315e-08 | 6.293 | 9.660e-08 | 4.598 | 1.634e-06 | 4.346 | 9.808e-04 | -0.001 | 9.005e-07 | 6.292 | 1.722e-07 | 6.610 | 1.660e-10 | 11.842 |
| 8 | 3.275e-10 | 6.143 | 3.324e-09 | 4.861 | 8.709e-08 | 4.230 | 9.812e-04 | -0.001 | 4.733e-08 | 4.250 | 4.230e-09 | 5.347 | 1.556e-10 | 0.093 |
| 9 | 4.890e-12 | 6.066 | 1.090e-10 | 4.930 | 4.958e-09 | 4.135 | 9.798e-04 | 0.002 | 3.626e-09 | 3.706 | 1.241e-10 | 5.091 | 3.756e-12 | 5.372 |
| 10 | 7.438e-14 | 6.039 | 3.496e-12 | 4.963 | 2.945e-10 | 4.073 | 9.672e-04 | 0.019 | 2.517e-10 | 3.848 | 3.725e-12 | 5.058 | 6.881e-14 | 5.771 |
| 11 | 1.027e-15 | **6.179** | 1.107e-13 | 4.981 | 1.792e-11 | 4.038 | 9.386e-04 | 0.043 | 1.658e-11 | 3.924 | 1.143e-13 | 5.027 | 1.443e-15 | **5.575** |
| 12 | 1.943e-16 | 2.402 | 3.192e-15 | **5.116** | 1.105e-12 | **4.020** | 7.753e-04 | **0.276** | 1.063e-12 | **3.963** | 3.358e-15 | **5.089** | 1.388e-16 | 3.379 |

TABLE 5

*Grid refinement analysis for the new optimal weights presented in Subsection 3.2 and WENO algorithm for the function in (4.2). We can see how the accuracy is lost at the interval that contains the singularity, but it is controlled close to it, decreasing step by step as we move towards the singularity.*

| | $x_{2j-6}$ | | $x_{2j-4}$ | | $x_{2j-2}$ | | $x_{2j}$ | | $x_{2j+2}$ | | $x_{2j+4}$ | | $x_{2j+6}$ | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $i$ | $e_i$ | $\log_2\left(\frac{e_i}{e_{i+1}}\right)$ | $e_i$ | $\log_2\left(\frac{e_i}{e_{i+1}}\right)$ | $e_i$ | $\log_2\left(\frac{e_i}{e_{i+1}}\right)$ | $e_i$ | $\log_2\left(\frac{e_i}{e_{i+1}}\right)$ | $e_i$ | $\log_2\left(\frac{e_i}{e_{i+1}}\right)$ | $e_i$ | $\log_2\left(\frac{e_i}{e_{i+1}}\right)$ | $e_i$ | $\log_2\left(\frac{e_i}{e_{i+1}}\right)$ |
| 6 | 1.815e-06 | - | 2.339e-06 | - | 1.862e-03 | - | 2.322e-03 | - | 7.058e-05 | - | 1.682e-05 | - | 6.092e-07 | - |
| 7 | 2.315e-08 | 6.293 | 9.660e-08 | 4.598 | 2.599e-04 | 2.841 | 2.894e-04 | 3.005 | 9.005e-07 | 6.292 | 1.722e-07 | 6.610 | 1.660e-10 | 11.842 |
| 8 | 3.275e-10 | 6.143 | 3.324e-09 | 4.861 | 3.382e-05 | 2.942 | 3.567e-05 | 3.020 | 4.733e-08 | 4.250 | 4.230e-09 | 5.347 | 1.556e-10 | 0.093 |
| 9 | 4.890e-12 | 6.066 | 1.090e-10 | 4.930 | 4.958e-09 | 12.736 | 4.414e-06 | 3.015 | 3.626e-09 | 3.706 | 1.241e-10 | 5.091 | 3.756e-12 | 5.372 |
| 10 | 7.438e-14 | 6.039 | 3.496e-12 | 4.963 | 2.945e-10 | 4.073 | 5.485e-07 | 3.008 | 2.517e-10 | 3.848 | 3.725e-12 | 5.058 | 6.881e-14 | 5.771 |
| 11 | 1.027e-15 | **6.179** | 1.107e-13 | 4.981 | 1.792e-11 | 4.038 | 6.835e-08 | 3.005 | 1.658e-11 | 3.924 | 1.143e-13 | 5.027 | 1.443e-15 | **5.575** |
| 12 | 1.943e-16 | 2.402 | 3.192e-15 | **5.116** | 1.105e-12 | **4.020** | 8.510e-09 | **3.006** | 1.063e-12 | **3.963** | 3.358e-15 | **5.089** | 1.388e-16 | 3.379 |

TABLE 6

*Grid refinement analysis for the algorithm presented in Subsection 3.3 for the function in (4.2). We can see how the accuracy is raised at the interval that contains the singularity and how the order decreases in a controlled way, step by step as we move towards the singularity.*

position of the jump discontinuity has been lost in the discretization process [38]. We can see how the new optimal weights allow to control the reduction of accuracy close to the discontinuity.

| | $x_{2j-6}$ | | $x_{2j-4}$ | | $x_{2j-2}$ | | $x_{2j}$ | | $x_{2j+2}$ | | $x_{2j+4}$ | | $x_{2j+6}$ | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $i$ | $e_i$ | $\log_2\left(\frac{e_i}{e_{i+1}}\right)$ | $e_i$ | $\log_2\left(\frac{e_i}{e_{i+1}}\right)$ | $e_i$ | $\log_2\left(\frac{e_i}{e_{i+1}}\right)$ | $e_i$ | $\log_2\left(\frac{e_i}{e_{i+1}}\right)$ | $e_i$ | $\log_2\left(\frac{e_i}{e_{i+1}}\right)$ | $e_i$ | $\log_2\left(\frac{e_i}{e_{i+1}}\right)$ | $e_i$ | $\log_2\left(\frac{e_i}{e_{i+1}}\right)$ |
| 6 | 8.073e-11 | - | 1.284e-08 | - | 5.596e-08 | - | 4.996e-01 | - | 5.508e-08 | - | 1.294e-08 | - | 1.349e-10 | - |
| 7 | 8.178e-13 | 6.625 | 8.054e-10 | 3.995 | 3.493e-09 | 4.002 | 4.999e-01 | -0.001 | 3.486e-09 | 3.982 | 8.061e-10 | 4.004 | 8.493e-13 | 7.311 |
| 8 | 9.354e-15 | 6.450 | 5.037e-11 | 3.999 | 2.183e-10 | 4.000 | 5.000e-01 | -0.000 | 2.182e-10 | 3.998 | 5.037e-11 | 4.000 | 3.775e-15 | **7.814** |
| 9 | 1.197e-16 | **6.288** | 3.148e-12 | 4.000 | 1.364e-11 | 4.000 | 5.000e-01 | -0.000 | 1.364e-11 | 4.000 | 3.148e-12 | 4.000 | 2.220e-16 | 4.087 |
| 10 | 0 | - | 1.968e-13 | 4.000 | 8.527e-13 | 4.000 | 5.000e-01 | -0.000 | 8.527e-13 | 4.000 | 1.970e-13 | 3.999 | 0 | - |
| 11 | 0 | - | 1.230e-14 | 4.000 | 5.329e-14 | 4.000 | 5.000e-01 | -0.000 | 5.329e-14 | 4.000 | 1.243e-14 | 3.985 | 0 | - |
| 12 | 0 | - | 7.685e-16 | **4.000** | 3.331e-15 | **4.000** | 5.000e-01 | **-0.000** | 3.331e-15 | **4.000** | 6.661e-16 | **4.222** | 2.220e-16 | - |

TABLE 7

*Grid refinement analysis for the smoothness indicator proposed in (3.1) and WENO algorithm for the function in (4.3). We can see how the accuracy is reduced at the central interval of the stencil and around it.*

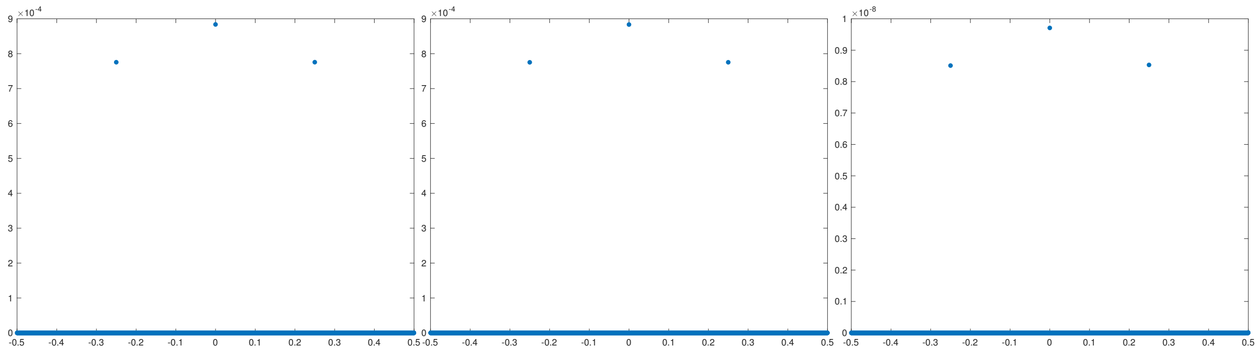

FIG. 9. *Absolute error obtained when reconstructing the function in (4.2) through WENO (left), through the algorithm presented in Subsection 3.2 (center) and through the algorithm presented in Subsection 3.3 (right). The original data was* $2^{13}$ *points and the subsampled data was* $2^{12}$ *points.*

| | $x_{2j-6}$ | | $x_{2j-4}$ | | $x_{2j-2}$ | | $x_{2j}$ | | $x_{2j+2}$ | | $x_{2j+4}$ | | $x_{2j+6}$ | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $i$ | $e_i$ | $\log_2\left(\frac{e_i}{e_{i+1}}\right)$ | $e_i$ | $\log_2\left(\frac{e_i}{e_{i+1}}\right)$ | $e_i$ | $\log_2\left(\frac{e_i}{e_{i+1}}\right)$ | $e_i$ | $\log_2\left(\frac{e_i}{e_{i+1}}\right)$ | $e_i$ | $\log_2\left(\frac{e_i}{e_{i+1}}\right)$ | $e_i$ | $\log_2\left(\frac{e_i}{e_{i+1}}\right)$ | $e_i$ | $\log_2\left(\frac{e_i}{e_{i+1}}\right)$ |
| 6 | 8.073e-11 | - | 1.295e-10 | - | 5.597e-08 | - | 4.996e-01 | - | 5.507e-08 | - | 4.475e-10 | - | 1.349e-10 | - |
| 7 | 8.178e-13 | 6.625 | 1.201e-12 | 6.753 | 3.493e-09 | 4.002 | 4.999e-01 | -0.001 | 3.486e-09 | 3.982 | 3.296e-12 | 7.085 | 8.493e-13 | 7.311 |
| 8 | 9.354e-15 | 6.450 | 1.235e-14 | 6.603 | 2.183e-10 | 4.000 | 5.000e-01 | -0.000 | 2.182e-10 | 3.998 | 2.287e-14 | 7.171 | 3.775e-15 | **7.814** |
| 9 | 1.162e-16 | 6.331 | 1.440e-16 | 6.423 | 1.364e-11 | 4.000 | 5.000e-01 | -0.000 | 1.364e-11 | 4.000 | 2.220e-16 | **6.687** | 2.220e-16 | 4.087 |
| 10 | 1.735e-18 | **6.066** | 1.735e-18 | **6.375** | 8.527e-13 | 4.000 | 5.000e-01 | -0.000 | 8.527e-13 | 4.000 | 0 | - | 0 | - |
| 11 | 0 | - | 0 | - | 5.329e-14 | 4.000 | 5.000e-01 | -0.000 | 5.329e-14 | 4.000 | 0 | - | 0 | - |
| 12 | 0 | - | 0 | - | 3.331e-15 | **4.000** | 5.000e-01 | **-0.000** | 3.331e-15 | **4.000** | 2.220e-16 | - | 0 | - |

TABLE 8

*Grid refinement analysis for the new optimal weights and WENO algorithm for the function in (4.3). We can see how the accuracy is increased around the discontinuity.*

**Example 4** In this experiment we would like to check the computational performance of the new algorithms compared to the classical WENO algorithm. The code has been written in Matlab R2017b and executed in a laptop running OSX version 10.9.5 with a microprocessor Intel Core i5, 1.4GHz and 8 GB of RAM memory. In Table 9 we present the results. In order to obtain each result presented in the table, we have executed 50 times each algorithm with the same initial data, we have obtained the computational time using the tic-toc buil-tin routines of Matlab and then we have obtained the mean of the 50 results. The initial data have been the same as the one used in the Examples 1, 2 and 3 at the same resolution used in the grid refinement analysis presented. The conclusions that we can reach from these experiments is that the new algorithms proposed are more expensive than the classical WENO, but not so much. Comparing the two new algorithms presented in this paper, both behave approximately the same in terms of computational time.

## 5. Conclusions.

In this article we have presented and analyzed two strategies that allow to improve the results obtained by WENO algorithm. The first one consists in a new nonlinear design of the WENO optimal weights. This new strategy allows to control the order of accuracy of the interpolation close to the discontinuity but not in the interval that contains it. The second strategy is inspired by the ENO-SR algorithm [41]. This second algorithm manages to raise the order of accuracy even at the interval that contains the discontinuity. Both strategies make use of new smoothness indicators that are inspired by those presented in [36]. The new algorithms have been theoretically analyzed to determine the value of the parameters $t$ and $\epsilon$ that appear in (2.10) and (3.13). It turns out that the value of these parameters is

| | Example 1 | | | Example 2 | | | Example 3 | | |
|---|---|---|---|---|---|---|---|---|---|
| $i$ | WENO | New WENO | WENO-SR | WENO | New WENO | WENO-SR | WENO | New WENO | WENO-SR |
| 6 | 0.0039189 | 0.0050155 | 0.0067716 | 0.0037825 | 0.0063749 | 0.0055041 | 0.0040808 | 0.0060425 | 0.0060831 |
| 7 | 0.0041907 | 0.0064866 | 0.0072892 | 0.0032467 | 0.0066217 | 0.0074879 | 0.0028794 | 0.0070028 | 0.0056376 |
| 8 | 0.0052122 | 0.01333 | 0.0094176 | 0.0049805 | 0.0093341 | 0.010404 | 0.0046728 | 0.0094143 | 0.011844 |
| 9 | 0.0092705 | 0.021184 | 0.023047 | 0.0092013 | 0.019338 | 0.028683 | 0.0094946 | 0.018851 | 0.018876 |
| 10 | 0.01928 | 0.038014 | 0.045976 | 0.018954 | 0.034816 | 0.038187 | 0.018083 | 0.035791 | 0.036507 |
| 11 | 0.035951 | 0.061106 | 0.062691 | 0.034358 | 0.061173 | 0.060977 | 0.036714 | 0.061167 | 0.061914 |
| 12 | 0.059382 | 0.12116 | 0.12277 | 0.059937 | 0.12127 | 0.12286 | 0.059257 | 0.12214 | 0.12334 |

TABLE 9

*In this table we present the computational time consumed by WENO, the algorithm presented in Subsection 3.2 (labeled as New WENO) and the algorithm presented in Subsection 3.3 (labeled as WENO-SR).*

important in order to assure that the algorithms satisfy the ENO property, presented in Subsection 3.4, and the accuracy requirement for which they have been designed: attaining optimal accuracy control even close to kinks and jump discontinuities. The numerical experiments presented confirm all the theoretical conclusions reached. This work is the first one of a series of two, and is devoted to the point values version of the algorithms presented. The second article [1] will be devoted to the cell averages and how to implement a shock capturing scheme for the accurate solution of hyperbolic conservation laws.

**Acknowledgments.** We would like to thank the referees and the editor for their useful suggestions and comments that, with no doubt, have helped to improve the quality of this paper.

## REFERENCES

[1] Sergio Amat, Juan Ruiz, and Chi-Wang Shu. On new strategies to control the accuracy of WENO algorithm close to discontinuities II: cell averages and conservation laws. *In preparation.*, 2018.

[2] Ami Harten and Stanley Osher. Uniformly high-order accurate nonoscillatory schemes. I. *SIAM J. Numer. Anal.*, 24(2):279 – 309, 1987.

[3] Ami Harten, Bjorn Engquist, Stanley Osher, and Sukumar R Chakravarthy. Uniformly high order accurate essentially non-oscillatory schemes, III. *J. Comput. Phys.*, 71(2):231 – 303, 1987.

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

[5] Chi-Wang Shu. *High Order ENO and WENO Schemes for Computational Fluid Dynamics*, pages 439 – 582. Springer, Berlin, Heidelberg, 1999.

[6] Chi-Wang Shu and Stanley Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes II. *J. Comput. Phys.*, 83(1):32 – 78, 1989.

[7] Francesc Arandiga, Albert Cohen, Rosa Donat, Nira Dyn, and Basarab Matei. Approximation of piecewise smooth functions and images by edge-adapted (ENO-EA) nonlinear multiresolution techniques. *Appl. Comput. Harmon. Anal.*, 24(2):225 – 250, 2008. Special Issue on Mathematical Imaging – Part II.

[8] Sergio Amat, Francesc Aràndiga, Albert Cohen, Rosa Donat, Gregori Garcia, and Markus von Oehsen. Data compression with ENO schemes: A case study. *Appl. Comput. Harmon. Anal.*, 11(2):273 – 288, 2001.

[9] S. Serna and A. Marquina. Power ENO methods: a fifth-order accurate weighted power ENO method. *J. Comput. Phys.*, 194(2):632 – 658, 2004.

[10] A. Cohen, N. Dyn, and B. Matei. Quasi linear subdivision schemes with applications to ENO interpolation. *Appl. Comput. Harmon. Anal.*, 15:89 – 116, 2003.

[11] Sergio Amat, Sonia Busquier, and J. Carlos Trillo. On multiresolution schemes using a stencil selection procedure: applications to ENO schemes. *Numer. Algorithms*, 44(1):45 – 68, 2007.

[12] Sergio Amat, Francesc Aràndiga, Albert Cohen, Rosa Donat, Gregori Garcia, and Markus von

Oehsen. Data compression with ENO schemes: A case study. *Appl. Comput. Harmon. Anal.*, 11(2):273 – 288, 2001.

[13] Xu-Dong Liu, Stanley Osher, and Tony Chan. Weighted essentially non-oscillatory schemes. *J. Comput. Phys.*, 115(1):200 – 212, 1994.

[14] G. Jiang and C.W. Shu. Efficient implementation of weighted ENO schemes. *J. Comput. Phys.*, 126(1):202 – 228, 1996.

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

[16] F. Aràndiga, A.M. Belda, and P. Mulet. Point-value WENO multiresolution applications to stable image compression. *J. Sci. Comput.*, 43(2):158–182, 2010.

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

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

[19] Samala Rathan and G. Naga Raju. A modified fifth-order WENO scheme for hyperbolic conservation laws. *Comput. Math. Appl.*, 75(5):1531 – 1549, 2018.

[20] Cong Huang and Li Li Chen. A simple smoothness indicator for the WENO scheme with adaptive order. *J. Comput. Phys.*, 352:498 – 515, 2018.

[21] Jun Zhu and Jianxian Qiu. A new type of modified WENO schemes for solving hyperbolic conservation laws. *SIAM J. Sci. Comput.*, 39(3):A1089 – A1113, 2017.

[22] Bart S. van Lith, Jan H.M. ten Thije Boonkkamp, and Wilbert L. IJzerman. Embedded WENO: A design strategy to improve existing WENO schemes. *J. Comput. Phys.*, 330:529 – 549, 2017.

[23] Y.-T. Zhang and C.-W. Shu. Chapter 5 - ENO and WENO schemes. In Remi Abgrall and Chi-Wang Shu, editors, *Handbook of Numerical Methods for Hyperbolic Problems. Basic and Fundamental Issues*, volume 17 of *Handbook of Numerical Analysis*, pages xxi – xxiii. Elsevier, 2016.

[24] Di Sun, Feng Qu, and Chao Yan. An efficient adaptive high-order scheme based on the WENO process. *Comput. Fluids*, 140:81 – 96, 2016.

[25] Jun Zhu and Jianxian Qiu. A new fifth order finite difference WENO scheme for solving hyperbolic conservation laws. *J. Comput. Phys.*, 318:110 – 121, 2016.

[26] Xiangxiong Zhang and Chi-Wang Shu. Positivity-preserving high order finite difference WENO schemes for compressible euler equations. *J. Comput. Phys.*, 231(5):2245 – 2258, 2012.

[27] G. A. Gerolymos, D. Sénéchal, and I. Vallet. Very-high-order WENO schemes. *J. Comput. Phys.*, 228(23):8481 – 8524, December 2009.

[28] Yiqing Shen and Gecheng Zha. Improvement of the WENO scheme smoothness estimator. *Int. J. Numer. Methods Fluids*, 64(6):653–675, 2010.

[29] Yuanyuan Liu, Chi-Wang Shu, and Mengping Zhang. High order finite difference WENO schemes for nonlinear degenerate parabolic equations. *SIAM J. Sci. Comput.*, 33(2):939–965, 2011.

[30] Tong Sun. Numerical smoothness and error analysis on WENO for the nonlinear conservation laws. *Numer. Methods. Partial Differ. Equ.*, 29(6):1881–1911, 2013.

[31] F. Aràndiga, M. C. Martí, and P. Mulet. Weights design for maximal order WENO schemes. *J. Sci. Comput.*, 60(3):641 – 659, September 2014.

[32] Qin Li, Pengxin Liu, and Hanxin Zhang. Piecewise polynomial mapping method and corresponding WENO scheme with improved resolution. *Commun. Comput. Phys.*, 18(5):1417 – 1444, 2015.

[33] Dinshaw S. Balsara, Sudip Garain, and Chi-Wang Shu. An efficient class of WENO schemes with adaptive order. *J. Comput. Phys.*, 326(C):780 – 804, December 2016.

[34] Chi-Wang Shu. *Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws*, pages 325 – 432. Springer, Berlin, Heidelberg, 1998.

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

[36] Sergio Amat and Juan Ruiz. New WENO smoothness indicators computationally efficient in the presence of corner discontinuities. *J. Sci. Comput.*, 71(3):1265 – 1302, Jun 2017.

[37] Sergio Amat, Jacques Liandrat, Juan Ruiz, and J. Carlos Trillo. On a power WENO scheme with improved accuracy near discontinuities. *SIAM J. Sci. Comput.*, 39(6):A2472 – A2507, 2017.

[38] Francesc Arandiga, Albert Cohen, Rosa Donat, and Nira Dyn. Interpolation and approximation of piecewise smooth functions. *SIAM J. Numer. Anal.*, 43(1):41–57, 2005.

[39] E. Carlini, R. Ferretti, and G. Russo. A weighted essentially nonoscillatory, large time-step scheme for hamilton–jacobi equations. *SIAM J. Sci. Comput.*, 27(3):1071 – 1091, 2005.
[40] Guangshan Jiang and Danping Peng. Weighted ENO schemes for hamilton-jacobi equations. *SIAM J. Sci. Comput.*, 21:2126–2143, 2000.
[41] A. Harten. ENO schemes with subcell resolution. *J. Comput. Phys.*, 83(1):148 – 184, 1989.