J Sci Comput
DOI 10.1007/s10915-017-0596-3

On an New Algorithm for Function Approximation with Full Accuracy in the Presence of Discontinuities Based on the Immersed Interface Method

Sergio Amat[1] · **Zhilin Li**[2] · **Juan Ruiz**[3]

Received: 22 December 2016 / Revised: 25 October 2017 / Accepted: 31 October 2017

Abstract This paper is devoted to the construction and analysis of an adapted and nonlinear multiresolution algorithm designed for interpolation or approximation of discontinuous univariate functions. The adaption attained allows to avoid numerical artifacts that appear when using linear algorithms and, at the same time, to obtain a high order of accuracy close to the singularities. It is known that linear algorithms are stable and convergent for smooth functions, but diffusion and Gibbs effect appear if the functions are piecewise continuous. Our aim is to develop an algorithm for function approximation with full accuracy that is capable to adapt to corners (kinks) and jump discontinuities, that uses a centered stencil and that does not use extrapolation. In order to reach this goal, we will need some information about the jumps in the function that we want to approximate and its derivatives. If this information is available, the algorithm is the most compact possible in the sense that the stencil is fixed and we do not need a stencil selection procedure as other algorithms do, such as ENO subcell resolution (ENO-SR). If the information about the jumps is not available, we will show a

Sergio Amat has been supported through 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). Zhilin Li has been partially supported supported by the NSF Grant DMS-1522768. Juan Ruiz has been supported through the *Programa de Apoyo a la investigación de la fundación Séneca-Agencia de Ciencia y Tecnología de la Región de Murcia* 19374/PI714, through the national research Project MTM2015-64382-P (MINECO/FEDER) and by the *Fundación Seneca* through the young researchers program *Jiménez de la Espada*.

✉ Juan Ruiz
juan.ruiza@uah.es

Sergio Amat
sergio.amat@upct.es

Zhilin Li
zhilin@math.ncsu.edu

1 Department of Applied Mathematics and Statistics, Universidad Politécnica de Cartagena (UPCT), Cartagena, Spain

2 Department of Mathematics, North Carolina State University (NCSU) (USA), Raleigh, NC, USA

3 Department of Physics and Mathematics, Universidad de Alcalá (UAH), Alcalá de Henares, Spain

Published online: 08 November 2017

technique to approximate it. The algorithm is based on linear interpolation plus correction terms that provide the desired accuracy close to corners or jump discontinuities.

Keywords IIM · Finite difference methods · Correction terms · Multiresolution schemes · Improved adaption to discontinuities · Signal processing

Mathematics Subject Classification 65D05 · 65D17 · 65M06 · 65N06

1 Introduction

In the literature about interpolation and approximation, the classical approach consists in using linear methods, see for example chapter one of [1]. It is known that these methods are convergent and stable for smooth functions but that they introduce Gibbs effect and diffusion when dealing with piecewise continuous functions. These numerical effects make it difficult to use linear methods for solving many problems where solutions present discontinuities or where we need to deal with discontinuous data. Examples are the solutions of PDEs that arise from conservation laws and that contain shocks in the flux or the approximation of piecewise continuous functions.

In the context of Harten's multiresolution, a set of data usually comes from a point-value or a cell-average discretization. In the point-value setting, the discretized data and the values of the original piecewise continuous function will be the same at a given set of points. In this setting it is only possible to detect corner singularities, i.e. kinks, as the position of jump discontinuities is lost during the process of discretization. In the cell-average setting, the data comes from the integration of a function in certain intervals. In this setting it is possible to detect the position of corner and jump discontinuities. Both settings will be presented in Sect. 2. Starting from the given data, the function is then reconstructed using interpolation. That means that a linear, a quasilinear or a non linear prediction operator will be used in order to try to recover an approximation of the original function. If the prediction operator is linear, then a linear interpolation will be the tool used in order to recover an approximation of the original function. As mentioned before, the problem that arises from using linear prediction operators is related to the accuracy of the approximation near the discontinuities: the order of approximation is lost due to Gibbs effect and diffusion. Any stencil that touches the discontinuity will be affected, as it will be shown in Sect. 3, and the order of accuracy is lost. Because of this fact, the increasing of the length of the stencil will result in larger regions affected around discontinuities. Quasilinear and nonlinear methods appear as alternatives to the disadvantages that linear methods present when dealing with discontinuities. A widely studied quasilinear method is ENO (Essentially Non Oscillatory) reconstruction, which strategy is to choose stencils that do not cross discontinuities. ENO method was first introduced by Harten et al. [2,3] in the context of conservation laws. Using ENO strategy, the regions affected by discontinuities are reduced to an interval that contains the discontinuity. This is possible because ENO strategy selects among several, the stencil that is not affected by the discontinuity. The interested reader can refer to [4–9] and the references therein in order to obtain more information about ENO. ENO-Subcell resolution (ENO-SR) strategy [10,11] appeared as an improvement of ENO. In [12], the authors present a modification of the subcell resolution technique that assures the detection of the discontinuity if the grid-spacing is smaller than a certain threshold and they prove that full accuracy is attained for corner and jump discontinuities. Thus, ENO-SR scheme is a good candidate for

solving problems where an appropriate treatment of discontinuities is a must. Even though, ENO presents the important drawback of being uncentered, due to the stencil selection procedure, and ENO-SR, in addition, uses extrapolation as a final step. The extrapolation and the stencil selection strategy are known to introduce instability [13] in many applications [8].

In recent approaches [7,14–21], the authors introduce a nonlinear processing of data that attains adaption to discontinuities. In [2–5], the authors propose nonlinear shock capturing methods to solve PDEs that arise from conservation laws and that contain shocks in the flux. In [14,15,22,23], just to mention some examples, the authors adapt these methods to function approximation. These methods do not use stencil selection strategies, being the stencil fixed and centered. Some of them are known to be stable [19] and free of Gibbs effect [13,24]. Even though, for non linear algorithms it is also known that diffusion is always present and that full accuracy can not be attained.

In [25], the author introduced the immersed boundary method (IBM). It was originally designed to deal with singular forces placed on an interface in the context of conservation laws. More specifically, the problem was to simulate the numerical flux of blood inside the heart. The main handicap that this method presents is that, in order to deal with the singular forces that arise from this problem, they are smeared out reducing the accuracy of the method to first order. In [26,27] the authors introduce the immersed interface method (IIM) which aim is to rise the global order of accuracy of the solution of PDEs when dealing with discontinuous coefficients, solution, flux or source term across the interface. In fact, the IIM manages to obtain global second order accuracy. To attain this goal, the IIM uses correction terms in order to adapt the classical finite difference stencil to the presence of discontinuities. In order to do this, it is fundamental to know the position of the discontinuity up to certain accuracy. In this article we will try to adapt the method introduced in [26,27] to the approximation of piecewise continuous functions.

Our aim is to design a centered algorithm for univariate functions that only uses interpolation and that attains full accuracy in the presence of corner and jump discontinuities. In order to do this, we will make use of centered Newton's polynomials plus correction terms that will allow to rise the accuracy close to discontinuities. In all the intervals not suspicious of containing a singularity, the algorithm applies Newton's linear interpolation. In [28], the authors propose correction terms for the finite difference discretization of the Navier Stokes equation. The main idea of the mentioned work is to correct the finite difference approximations of the derivatives using Taylor expansions around the interface and known jump conditions of the function and its derivatives at singularities. We will use the same strategy in order to adapt Newton polynomials to the presence of discontinuities. Depending on the problem, the jump conditions at the discontinuities might be available. If this is the case, the algorithm presented is the most compact possible, in the sense that it always uses the same stencil, it is centered and it presents full accuracy, in contrast with ENO-SR that broadens the stencil to adapt to the presence of discontinuities. In the case that the jump conditions are not available, a method to obtain them will be presented and we will discuss how the accuracy obtained for these values affects the accuracy of the method. It will be very important to know the position of discontinuities up to certain accuracy. In order to obtain this information we will use the strategy presented in [12], as the theory for the accuracy attained in the detections is available. Other detection mechanisms are at hand. For example those presented in [29], Chapter 10.1.

This paper is organized as follows: in Sect. 2 we briefly introduce the Harten multiresolution setting for point-values and cell-averages and how the discontinuities can be detected and localized up to the desired accuracy in the mentioned two settings. Section 3 shows

how to find the correction terms for the prediction operators in the point-value and in the cell-average setting. Finally, Sect. 4 presents some numerical experiments where it is shown how the new algorithm performs with corner and jump discontinuities in one dimension.

2 Harten's Multiresolution and Detection of Discontinuities

As some theoretical background about multiresolution will be needed in order to describe the new algorithm proposed, in what follows we will introduce the Harten's framework for multiresolution considering the point-value and the cell-average setting. Both configurations will be used for function approximation and for subdivision experiments in one dimension. The interested reader should refer to [22] for a deeper presentation of Harten's multiresolution framework, as the mentioned article presents an excellent review on the subject.

Let's start by the *point-value setting*. Let us consider the set of continuous functions in $[0, 1]$, $\mathcal{C}[0, 1]$, the space of finite sequences V^k of length $N_k = J_k + 1$ and let X^k be a uniform partition of the interval $[0, 1]$ in J_k subintervals. The partition corresponding to the multiresolution level k will be defined by,

$$X^k = \{x_i^k\}_{i=0}^{J_k}, \quad k = L, \ldots, 0, \quad \text{where,} \quad x_i^{k-1} = x_{2i}^k, \quad i = 0, \ldots, J_{k-1},$$

with $J_{k-1} = \frac{J_k}{2}$ and J_0 the number of subintervals at the coarsest scale $k = 0$. The discretization operator $f^k = \mathcal{D}_k f$, will be defined as,

$$f_i^k = (\mathcal{D}_k f)_i = f(x_i^k), \quad f^k = \left\{ f_i^k \right\}_{i=0}^{J_k}. \tag{1}$$

If we consider a set of nested grids, such that $X^{k-1} \subseteq X^k$, $\forall k$, we can define the decimation operator,

$$f_i^{k-1} = \left(D_k^{k-1} f^k \right)_i = f_{2i}^k, i = 0, \ldots, J_{k-1}.$$

Now we can define the reconstruction operator $\mathcal{R}_k$ that must satisfy the consistency condition $\mathcal{D}_k \mathcal{R}_k f^k = f^k$, what is to say that the reconstruction at the nodes $\{x_i^k\}$ must be equal to $\{f_i^k\}$,

$$\left(\mathcal{R}_k f^k \right)(x_i^k) = f_i^k = f(x_i^k).$$

Thus, $\mathcal{R}_k f^k(x)$ must be a bounded function that interpolates the data $\{f_i^k\}$ at the nodes $\{x_i^k\}$. Any interpolation technique can be chosen as reconstruction, and we use the notation,

$$I(x, f^k) = \left(\mathcal{R}_k f^k \right)(x).$$

From this expression, it is easy to see that the prediction from the scale $k-1$ to the scale k will be,

$$\left(P_{k-1}^k f^{k-1} \right)_i = \left(\mathcal{D}_k I \left(x, f^{k-1} \right) \right)_i = I(x_i^k, f^{k-1}).$$

In our case, $I(x, f^k) = \left(\mathcal{R}_k f^k \right)(x)$ are Newton polynomials particularized at $x_{j+1/2}$ plus correction terms if a discontinuity affects the stencil. The prediction error will have the expression,

$$e_i^k = f_i^k - I(x_i^k, f^{k-1}),$$

that will be equal to zero for all the nodes x_{2i}^k. Thus, we define the *details* $d^k = \{d_i^k\}$ as the interpolation errors at the nodes x_{2i-1}, i.e. $d_i^k = e_{2i-1}^k$.

Let us now introduce the *cell-average setting*. Let's consider a set of nested grids in $[0, 1]$:

$$X^k = \{x_j^k\}_{j\in\mathbb{Z}}, \quad x_j^k = jh_k, \quad h_k = 2^{-k}/N_0, \quad N_k = 2^k N_0, \quad k = 0, \ldots, L,$$

where we use the discretization

$$\bar{f}_j^k = (\mathcal{D}_k f)_j = \frac{1}{h_k}\int_{x_{j-1}^k}^{x_j^k} f(x)dx, \quad j \in \mathbb{Z}. \tag{2}$$

Usually the initial data correspond to the finest resolution level $\bar{f}^L$. Using the additivity of the integral, we obtain the decimation steps between scales:

$$\bar{f}_j^{k-1} = \left(D_k^{k-1}\bar{f}^k\right)_j = \frac{1}{h_{k-1}}\int_{x_{j-1}^{k-1}}^{x_j^{k-1}} f(x)dx = \frac{1}{2h_k}\int_{x_{2j-2}^k}^{x_{2j}^k} f(x)dx = \frac{\bar{f}_{2j-1}^k + \bar{f}_{2j}^k}{2}. \tag{3}$$

We can define the errors of reconstruction as $e^k = P_{k-1}^k\bar{f}^{k-1} - \bar{f}^k$. Now, using the consistency requirement for P_{k-1}^k and the expression of the errors of reconstruction,

$$\begin{aligned}\bar{f}_j^{k-1} &= \left(D_k^{k-1}P_{k-1}^k\bar{f}^{k-1}\right)_j = \frac{\left(P_{k-1}^k\bar{f}^{k-1}\right)_{2j-1} + \left(P_{k-1}^k\bar{f}^{k-1}\right)_{2j}}{2} \\ &= \frac{\bar{f}_{2j-1}^k + e_{2j-1}^k + \bar{f}_{2j}^k + e_{2j}^k}{2}.\end{aligned} \tag{4}$$

Thus, if $\bar{f}^{k-1} = D_k^{k-1}\bar{f}^k$, then the two last Eqs. (3) and (4), imply that the prediction errors at even cells can be obtained directly from odd cells and vice versa,

$$e_{2j-1}^k + e_{2j}^k = 0.$$

The consistency requirement of the reconstruction operator $\mathcal{D}_k\mathcal{R}_k f^k = \bar{f}^k$, can be directly satisfied through the *primitive function*, that allows to fulfil this requirement using interpolation. Let's define the function $\{F_i^k\}$ as,

$$F_i^k = h_k\sum_{j=1}^{i}\bar{f}_j^k = \int_0^{x_i^k} f(y)dy. \tag{5}$$

If we consider F as the primitive of f, i.e. $F(x) = \int_0^x f(y)dy$, that will be a continuous function, then the sequence $\{F_i^k\}$ is a discretization in the point-value framework of F at the resolution level k. Then, if we take $F_0^k = 0$ for all k, the values of the primitive are related with the cell values through the expression,

$$\bar{f}_i^k = \frac{F_i^k - F_{i-1}^k}{h_k}, \tag{6}$$

If $I(x, F^k)$ is a reconstruction operator in the primitive that uses interpolation, it is easy to check that it is consistent, see [22]. We can define the reconstruction operator in the cell-averages using the reconstruction operator in the primitives, that satisfies the consistency

requirement

$$\mathcal{R}(x, \bar{f}^k) = \frac{d}{dx} I(x, F^k) \longrightarrow \left(\mathcal{D}_k \left(\mathcal{R}_k(x, \bar{f}^k)\right)\right)(x_i) = \frac{1}{h_k} \int_{x_{i-1}^k}^{x_i^k} \frac{d}{dx} I(x, F^k)$$
$$= \frac{F_i^k - F_{i-1}^k}{h_k} = \bar{f}_i^k. \tag{7}$$

Now, considering only the prediction errors at (for example) the odd cells of the grid X^k, we immediately obtain a one-to-one correspondence:

$$\bar{f}_j^{k-1} = \frac{\bar{f}_{2j}^k + \bar{f}_{2j-1}^k}{2}, \quad d_j^k = \bar{f}_{2j-1}^k - \left(P_{k-1}^k \bar{f}^{k-1}\right)_{2j-1}, \tag{8}$$

$$\bar{f}_{2j-1}^k = \left(P_{k-1}^k \bar{f}^{k-1}\right)_{2j-1} + d_j^k, \quad \bar{f}_{2j}^k = 2\bar{f}_j^{k-1} - \bar{f}_{2j-1}^k. \tag{9}$$

In our case, the prediction operator will be Newton polynomials in the primitive plus correction terms close to the discontinuities. In [14–21] other prediction operators can be found.

2.1 Detecting the Position of Discontinuities

When working with data which origin is the discretization of piecewise continuous functions, that present certain regularity to both sides of discontinuities, it is possible to detect the presence of such discontinuities. The most usual are kinks (or corner singularities), that are jumps in the first derivative of the function, and jump discontinuities, that are jumps in the function. Both kinds of discontinuities can be detected depending on the discretization used. Kinks can be detected using the point-value discretization (1), i.e. a sampling of the function. There is no hope of detecting the position of jump discontinuities using this kind of discretization: the exact position is lost during the discretization process. Although, jump discontinuities can be detected using a cell-average discretization (2). What is more, jump discontinuities transform into kinks if we make use of the primitive function (5).

We will entrust the detection part of our algorithm to the method proposed in [12], as the authors also present the theoretical analysis that will be very useful for our purposes.

Let's describe how the strategy proposed in [12] works. Being h the grid spacing, the algorithm makes use of second order differences, $\Delta_h^2 f(x) = f(x) - 2f(x+h) + f(x+2h)$, as indicators of the presence of cells suspicious of containing a discontinuity. The detection mechanism defines a set of intervals, $I_k := [kh, (k+1)h]$, with $k \in \mathbb{Z}$, that are potential candidates of containing singularities. In [12] the authors propose the two following rules for a given approximation of order m that label the cells as B if they are suspicious of containing a discontinuity or G if they are not.

1. If

$$|\Delta_h^2 f((k-1)h)| > |\Delta_h^2 f((k-1 \pm n)h)|, \quad n = 1, \ldots, m,$$

   then the intervals I_{k-1} and I_k are labeled as B. A priory it is not possible to know which among the intervals I_{k-1} and I_k contains the singularity, so both are labeled as suspicious.
2. If

$$|\Delta_h^2 f((k-1)h)| > |\Delta_h^2 f((k+n)h)|, \quad n = 1, \ldots, m-1,$$

and

$$|\Delta_h^2 f((k-1)h)| > |\Delta_h^2 f((k-1-n)h)|, \quad n = 1, \ldots, m-1,$$

then the interval I_k is labeled as B. In this case, the two largest divided differences include the interval I_k, which is a candidate to contain the discontinuity.

All other intervals are labeled as G and it is assumed that they do not contain a singularity.

The algorithm is designed in such a way that for a sufficiently small h, the interval that contains the singularity is always labeled as B, while all the intervals labeled as G are in smooth regions of f. In fact, the authors present a lemma where they prove that the singularity is always detected under some critical scale h_c, that has the following expression,

$$h_c := \begin{cases} \frac{|[f']|}{4 \sup_{x \in \mathbb{R} \setminus \{x^*\}} |f''(x)|}, & \text{for the point values,} \\ \frac{|[f]|}{4 \sup_{x \in \mathbb{R} \setminus \{x^*\}} |f'(x)|}, & \text{for the cell averages.} \end{cases} \tag{10}$$

where $[f']$ is the jump of the first derivative f' and $[f]$ is the jump of the function at the point x^*. Then for $h < h_c$, the interval I_k which contains x^* is labeled as B. Moreover, if x^* is close to one endpoint of the interval I_k by at most a quarter of its size, then the adjacent interval to this endpoint is also labeled as B. Even though, it is possible that false detections occur and an interval I_k is labeled as B in a smooth region. The proof for the expressions in (10) can be found respectivelly in Lemma 2 and Section 7 of [12].

It is clear that the described algorithm is capable of detecting jumps in the function and the first derivative. Although, we can only hope to find the position of jump discontinuities in the cell-averages, as it is impossible in the point-values. Kinks are susceptible to be found in the point-values and in the cell averages. Thus, in order to obtain the position of the kinks, we choose a simple approximation: we build Lagrange polynomials to the left and to the right of the suspicious interval, labeled as B or BB. Then, we define a function H as the difference between the two polynomials, and we suppose that there is a unique root of this function inside the suspicious interval. Note that for a sufficiently small grid-spacing, only one root of H can fall inside the suspicious interval. In Lemma 3, statement 2 of [12] the authors proof this fact. Thus, the position of the discontinuity can be easily obtained finding the roots of H. The interested reader can refer to [11] for a nice discussion about the process. The accuracy of the results depends on the order of the polynomials, as the authors prove in Lemma 3, statement 3 of [12]. In the cell-average setting, jumps in the function transform into kinks in the primitive and their position can be easily obtained following the process described.

3 Newton Polynomials and Associated Correction Terms for Point-Values and Cell-Averages

In this section we will analyze how to correct the finite difference approximations of derivatives that appear in the Newton form of Lagrange polynomials in the point-values or the cell-averages. These polynomials are intrinsically designed to work with continuous smooth data. When the stencil touches a discontinuity, the approximation obtained through linear interpolation is lost as the function does not present the needed regularity. The Newton form of the interpolating polynomial for the point-values can be written as,

$$p_n(x) = f[x_0] + f[x_0, x_1](x - x_0) + \cdots + f[x_j, \ldots, x_{j+n}](x - x_0)(x - x_1)\ldots(x - x_{n-1}), \tag{11}$$

where $f[x_j, \ldots, x_{j+n}]$ are divided differences. Then at any point x, the error formula for the polynomial approximation is

$$f(x) - p_n(x) = \frac{1}{n!} f^{(n)}(\xi)(x - x_0)(x - x_1) \ldots (x - x_n), \tag{12}$$

where ξ is some point lying in the interval $(x_0, x_1, \ldots, x_n)$. How large the error is, depends on how small the derivative $f^{(n)}(\xi)$ is over this interval, what is to say, how smooth the function f is. If the stencil touches a jump discontinuity, the linear interpolation is only capable of reaching $O(1)$ approximations, as the jump is contained in the function. If the stencil touches a corner singularity, the linear interpolation is only capable of reaching $O(h)$ accuracy, as the jump is contained in the first derivative of the function, etc... Adding more points to the stencil will not increase this accuracy. As we will later see, adding corrections terms to the finite difference approximations of the derivatives will allow to increase the accuracy of the approximation at the interpolation point if a discontinuity affects the stencil. It is intuitive that these correction terms will directly depend on the jumps that the function or its derivatives might present.

3.1 Newton Polynomials in the Point-Values

Without loss of generality, let's limit the analysis to stencils of four points. Let's suppose that we are working with the stencil $\{f_{j-1}, f_j, f_{j+1}, f_{j+2}\}$ placed at the points $\{x_{j-1}, x_j, x_{j+1}, x_{j+2}\}$. In this case, we have that the discontinuity can be contained in the intervals $\{x_{j-1}, x_j\}$, $\{x_j, x_{j+1}\}$ or $\{x_{j+1}, x_{j+2}\}$. Depending on the position of the discontinuity, we can choose among different bases in order to express the Newton form of the interpolating polynomial at the point x. For example, using the expression in (11) for $n = 3$ and the previous stencil, we can choose the following four different expressions of the interpolating polynomial,

$$\begin{aligned} p_1(x) &= f_{j-1} + \frac{f_j - f_{j-1}}{h}(x - x_{j-1}) + \frac{f_{j+1} - 2f_j + f_{j-1}}{2h^2}(x - x_{j-1})(x - x_j) \\ &\quad + \frac{f_{j+2} - 3f_{j+1} + 3f_j - f_{j-1}}{6h^3}(x - x_{j-1})(x - x_j)(x - x_{j+1}), \end{aligned} \tag{13}$$

$$\begin{aligned} p_2(x) &= f_j + \frac{f_{j+1} - f_j}{h}(x - x_j) + \frac{f_{j+2} - 2f_{j+1} + f_j}{2h^2}(x - x_j)(x - x_{j+1}) \\ &\quad + \frac{f_{j+2} - 3f_{j+1} + 3f_j - f_{j-1}}{6h^3}(x - x_j)(x - x_{j+1})(x - x_{j+2}), \end{aligned} \tag{14}$$

$$\begin{aligned} p_3(x) &= f_{j+2} + \frac{f_{j+2} - f_{j+1}}{h}(x - x_{j+2}) + \frac{f_{j+2} - 2f_{j+1} + f_j}{2h^2}(x - x_{j+2})(x - x_{j+1}) \\ &\quad + \frac{f_{j+2} - 3f_{j+1} + 3f_j - f_{j-1}}{6h^3}(x - x_{j+2})(x - x_{j+1})(x - x_j), \end{aligned} \tag{15}$$

$$\begin{aligned} p_4(x) &= f_{j+1} + \frac{f_{j+1} - f_j}{h}(x - x_{j+1}) + \frac{f_{j+1} - 2f_j + f_{j-1}}{2h^2}(x - x_{j+1})(x - x_j) \\ &\quad + \frac{f_{j+2} - 3f_{j+1} + 3f_j - f_{j-1}}{6h^3}(x - x_{j+1})(x - x_j)(x - x_{j-1}). \end{aligned} \tag{16}$$

In next section we will use these expressions in order to give priority to one side of the stencil, depending on the position of the discontinuity. Note that the resulting polynomial is always the same and we only change the basis used to express it.

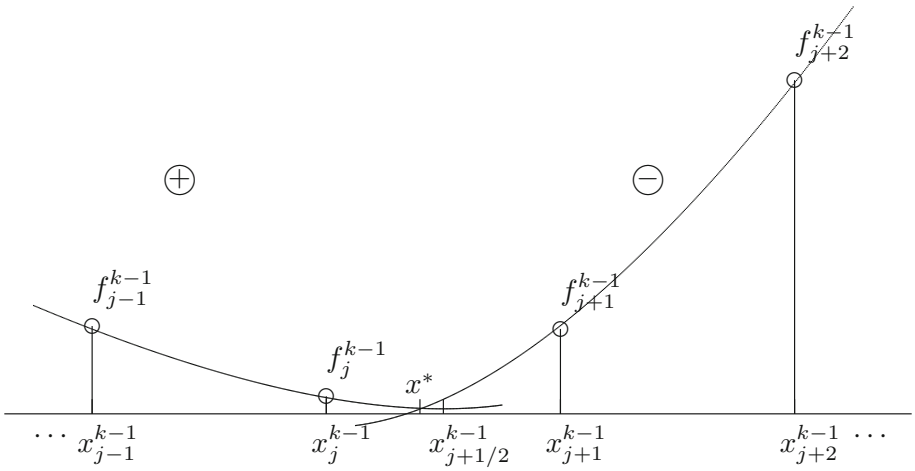

Fig. 1 In this figure we can see an example of a corner singularity placed in the interval $\left(x_j^{k-1}, x_{j+1/2}^{k-1}\right)$. The left side of the discontinuity x^* is labeled as the $+$ side and the right side of x^* as the $-$ side

3.2 Correction Terms for Newton Polynomials in the Point-Values

Inspired by the correction terms introduced by the authors in [26], we can try to obtain some correction terms that improve the accuracy of the finite differences that appear in the expression of the polynomials in (13), (14), (15), (16) when the interpolation point is placed at $x_{j+1/2} = x_j + \frac{h}{2}$. We will need the jump relations,

$$\begin{aligned} [f] &= f^+(x^*) - f^-(x^*), \\ \left[f'\right] &= f_x^+(x^*) - f_x^-(x^*), \\ \left[f''\right] &= f_{xx}^+(x^*) - f_{xx}^-(x^*), \\ \left[f'''\right] &= f_{xxx}^+(x^*) - f_{xxx}^-(x^*). \end{aligned} \tag{17}$$

We will consider the setting shown in Figs. 1, 2, 3 and 4. We can see how the domain to the left of the discontinuity has been labeled as $+$ and the domain to the right has been labeled as $-$. The position of the discontinuity has been marked with x^*. Let's consider the four different cases that follow and let's suppose that we know the jump relations in (17):

- If the discontinuity is placed in the interval $\{x_j, x_{j+1/2}\}$ at the point x^*, then $x_{j+1/2}$ belongs to the $-$ side of the discontinuity. In this case, we can use the expression of $p_3(x)$ in order to avoid information from the $+$ side of the discontinuity in the independent term and the first order difference. The second and third order divided differences must be expressed in terms of the $-$ side using the jump relations. An example of this case is presented in Fig. 1. Analyzing this figure we can see that, for the situation depicted, it is convenient to use the expression of $p_3(x)$ in (15), as it is the polynomial that has least differences affected by the discontinuity.
- If the discontinuity is placed in the interval $\{x_{j+1/2}, x_{j+1}\}$ at the point x^*, then $x_{j+1/2}$ belongs to the $+$ side of the discontinuity. Then, we can use the expression of $p_1(x)$ in order to avoid information from the $-$ side of the discontinuity in the independent term and the first order differences. In this case, the second and third order divided differences

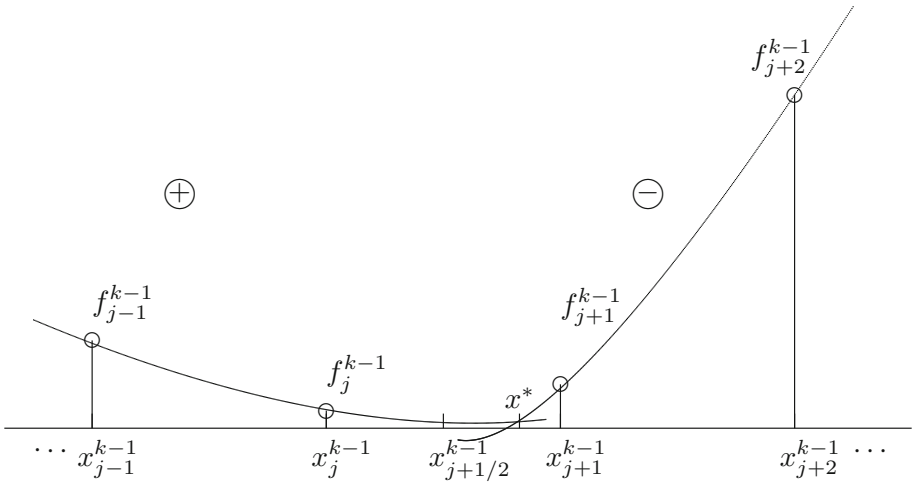

Fig. 2 In this figure we can see an example of a corner singularity placed in the interval $\left(x_{j+1/2}^{k-1}, x_{j+1}^{k-1}\right)$. The left side of the discontinuity x^* is labeled as the $+$ side and the right side of x^* as the $-$ side

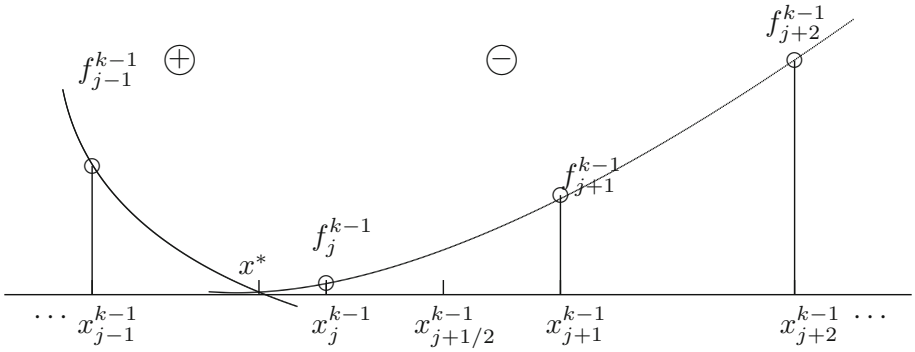

Fig. 3 In this figure we can see an example of a corner singularity placed in the interval $\left(x_{j-1}^{k-1}, x_j^{k-1}\right)$. The left side of the discontinuity x^* is labeled as the $+$ side and the right side of x^* as the $-$ side

must be expressed in terms of the $+$ side using the jump relations. An example of this case is presented in Fig. 2. In this figure we can see that it is convenient to use the expression of $p_1(x)$ shown in (13).

- If the discontinuity is placed in the interval $\{x_{j-1}, x_j\}$ at the point x^*, then $x_{j+1/2}$ belongs to the $-$ side of the discontinuity. In this case, we can use the expression of $p_2(x)$ in order to avoid information from the $+$ side of the discontinuity in the independent term, the first and the second order differences. The third order differences must be expressed in terms of the $-$ side using the jump relations. An example of this case is presented in Fig. 3. In this figure we can see that it is convenient to use the expression of $p_2(x)$ shown in (14). Other options are available, like using $p_3(x)$, but the result will be the same.
- If the discontinuity is placed in the interval $\{x_{j+1}, x_{j+2}\}$ at the point x^*, then $x_{j+1/2}$ belongs to the $+$ side of the discontinuity. In this case, we can use the expression of $p_4(x)$ in order to avoid information from the $-$ side of the discontinuity in the independent

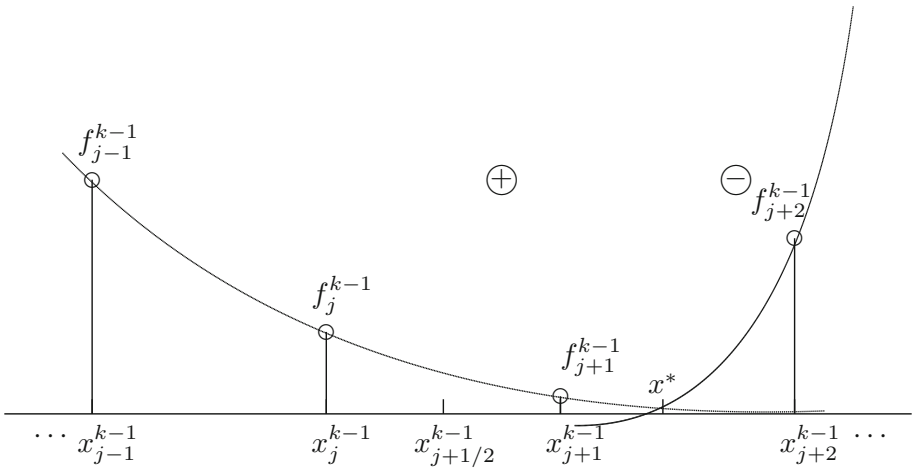

Fig. 4 In this figure we can see an example of a corner singularity placed in the interval $\left(x_{j+1}^{k-1}, x_{j+2}^{k-1}\right)$. The left side of the discontinuity x^* is labeled as the $+$ side and the right side of x^* as the $-$ side

term, the first and the second order divided differences. Again, the third order divided differences must be expressed in terms of the $-$ side using the jump relations. An example of this case is presented in Fig. 4. In this figure we can see that it is convenient to use the expression of $p_4(x)$ shown in (16). Other options are available, like using $p_1(x)$, but, as before, the result will be the same.

3.3 Obtainment of the Jump Conditions

As we will later see, in order to obtain the correction terms for the finite differences that appear in (13), (14), (15), (16) we will need the jump relations in (17). For some particular problems these relations are known. One example is the Stokes or the Navier-Stokes equations, where expressions for the jump conditions for the pressure and the velocity are available, see for example [27], chapter 10. Another example is the Hele-Shaw flow, where the Laplace-Young condition is available to obtain the jump relations for the pressure, see for example [27], chapter 11. If the jump relations (17) are known, the reader can jump directly to Sect. 3.4 where the correction terms are obtained using the jump relations. If they are unknown, we can obtain a good approximation following the process described in this subsection.

The correction terms are based on the local truncation error that results from expanding the values f_j that appear in the finite differences around the interpolation point. The jump relations will allow to express a value that belongs to the $-$ side in terms of the $+$ side and vice versa. In order to obtain the jump relations (17) we need to know the position of the discontinuity x^* up to certain accuracy. In Sect. 2.1 we explained how to obtain an approximation of the position of the discontinuity x^* with the desired accuracy. If we work with stencils of four points, we will need $O(h^4)$ accuracy in the detection. Let's suppose that we know that the discontinuity is placed at a distance α from x_j in the interval $\{x_j, x_{j+1}\}$, just as depicted in Fig. 5. If we want $O(h^4)$ accuracy for the approximation of the jump relations (17), we will need to use four points to each side of the discontinuity. Let's suppose that we use the stencil $\{f_{j-3}^+, f_{j-2}^+, f_{j-1}^+, f_j^+, f_{j+1}^-, f_{j+2}^-, f_{j+3}^-, f_{j+4}^-\}$ placed at the positions $\{x_{j-3}, x_{j-2}, x_{j-1}, x_j, x_{j+1}, x_{j+2}, x_{j+3}, x_{j+4}\}$. Then, we can obtain the values of the

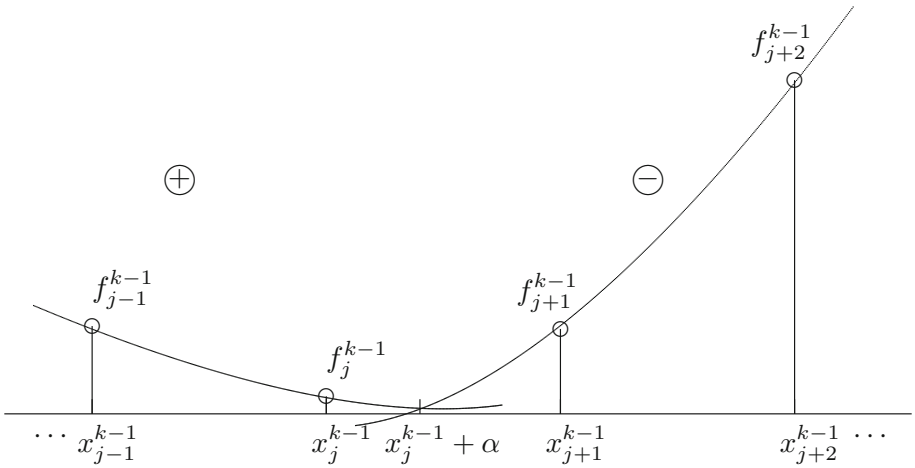

Fig. 5 In this figure we can see an example of a corner singularity placed in the interval $\left(x_j^{k-1}, x_{j+1}^{k-1}\right)$ at a position $x^* = x_j^{k-1} + \alpha$

derivatives of f from both sides of the discontinuity just using Taylor expansion around x^* and setting the following system of equations for the $+$ side,

$$\begin{aligned}
f_j^+ &= f^+(x^*) - f_x^+(x^*)\alpha + \frac{1}{2}f_{xx}^+(x^*)\alpha^2 - \frac{1}{3!}f_{xxx}^+(x^*)\alpha^3,\\
f_{j-1}^+ &= f^+(x^*) - f_x^+(x^*)(h+\alpha) + \frac{1}{2}f_{xx}^+(x^*)(h+\alpha)^2 - \frac{1}{3!}f_{xxx}^+(x^*)(h+\alpha)^3,\\
f_{j-2}^+ &= f^+(x^*) - f_x^+(x^*)(2h+\alpha) + \frac{1}{2}f_{xx}^+(x^*)(2h+\alpha)^2 - \frac{1}{3!}f_{xxx}^+(x^*)(2h+\alpha)^3,\\
f_{j-3}^+ &= f^+(x^*) - f_x^+(x^*)(3h+\alpha) + \frac{1}{2}f_{xx}^+(x^*)(3h+\alpha)^2 - \frac{1}{3!}f_{xxx}^+(x^*)(3h+\alpha)^3,
\end{aligned} \tag{18}$$

and for the $-$ side,

$$\begin{aligned}
f_{j+1}^- &= f^-(x^*) + f_x^-(x^*)(h-\alpha) + \frac{1}{2}f_{xx}^-(x^*)(h-\alpha)^2 + \frac{1}{3!}f_{xxx}^-(x^*)(h-\alpha)^3,\\
f_{j+2}^- &= f^-(x^*) + f_x^-(x^*)(2h-\alpha) + \frac{1}{2}f_{xx}^-(x^*)(2h-\alpha)^2 + \frac{1}{3!}f_{xxx}^-(x^*)(2h-\alpha)^3,\\
f_{j+3}^- &= f^-(x^*) + f_x^-(x^*)(3h-\alpha) + \frac{1}{2}f_{xx}^-(x^*)(3h-\alpha)^2 + \frac{1}{3!}f_{xxx}^-(x^*)(3h-\alpha)^3,\\
f_{j+4}^- &= f^-(x^*) + f_x^-(x^*)(4h-\alpha) + \frac{1}{2}f_{xx}^-(x^*)(4h-\alpha)^2 + \frac{1}{3!}f_{xxx}^-(x^*)(4h-\alpha)^3,
\end{aligned} \tag{19}$$

being $\alpha = x^* - x_j$. Solving the two systems (18) and (19), where $f^+(x^*)$, $f_x^+(x^*)$, $f_{xx}^+(x^*)$, $f_{xxx}^+(x^*)$ and $f^-(x^*)$, $f_x^-(x^*)$, $f_{xx}^-(x^*)$, $f_{xxx}^-(x^*)$ are the unknowns, the jump relations (17) can be easily obtained. Once we have the jump relations (17), we can express $f^+(x^*)$, $f_x^+(x^*)$, $f_{xx}^+(x^*)$, $f_{xxx}^+(x^*)$ in terms of $f^-(x^*)$, $f_x^-(x^*)$, $f_{xx}^-(x^*)$, $f_{xxx}^-(x^*)$ and vice versa.

It is interesting to stop here and think about the accuracy obtained when approximating the jump conditions in the way explained. Let's express the systems (18) and (19) in matrix

form with the corresponding local truncation error,

$$\begin{pmatrix} f_j^+ \\ f_{j-1}^+ \\ f_{j-2}^+ \\ f_{j-3}^+ \end{pmatrix} = \begin{pmatrix} 1 & \alpha & \frac{\alpha^2}{2} & -\frac{\alpha^3}{3!} \\ 1 & (h+\alpha) & \frac{(h+\alpha)^2}{2} & -\frac{(h+\alpha)^3}{3!} \\ 1 & (2h+\alpha) & \frac{(2h+\alpha)^2}{2} & -\frac{(2h+\alpha)^3}{3!} \\ 1 & (3h+\alpha) & \frac{(3h+\alpha)^2}{2} & -\frac{(3h+\alpha)^3}{3!} \end{pmatrix} \begin{pmatrix} f^+(x^*) \\ f_x^+(x^*) \\ f_{xx}^+(x^*) \\ f_{xxx}^+(x^*) \end{pmatrix} + \begin{pmatrix} O(h^4) \\ O(h^4) \\ O(h^4) \\ O(h^4) \end{pmatrix}. \quad (20)$$

Obtaining now the inverse of the system matrix,

$$A = \begin{pmatrix} \frac{6h^3+11h^2\alpha+6h\alpha^2+\alpha^3}{6h^3} & -\frac{\alpha(6h^2+5h\alpha+\alpha^2)}{2h^3} & \frac{\alpha(3h^2+4h\alpha+\alpha^2)}{2h^3} & -\frac{\alpha(2h^2+3h\alpha+\alpha^2)}{6h^3} \\ \frac{11h^2+12h\alpha+3\alpha^2}{6h^3} & -\frac{6h^2+10h\alpha+3\alpha^2}{2h^3} & \frac{3h^2+8h\alpha+3\alpha^2}{2h^3} & -\frac{2h^2+6h\alpha+3\alpha^2}{6h^3} \\ \frac{2h+\alpha}{h^3} & -\frac{5h+3\alpha}{h^3} & \frac{4h+3\alpha}{h^3} & -\frac{h+\alpha}{h^3} \\ h^{-3} & -3h^{-3} & 3h^{-3} & -h^{-3} \end{pmatrix}. \quad (21)$$

For the system in (19), the same procedure can be followed,

$$\begin{pmatrix} f_{j+1}^- \\ f_{j+2}^- \\ f_{j+3}^- \\ f_{j+4}^- \end{pmatrix} = \begin{pmatrix} 1 & h-\alpha & \frac{(h-\alpha)^2}{2} & \frac{(h-\alpha)^3}{6} \\ 1 & 2h-\alpha & \frac{(2h-\alpha)^2}{2} & \frac{(2h-\alpha)^3}{6} \\ 1 & 3h-\alpha & \frac{(3h-\alpha)^2}{2} & \frac{(3h-\alpha)^3}{6} \\ 1 & 4h-\alpha & \frac{(4h-\alpha)^2}{2} & \frac{(4h-\alpha)^3}{6} \end{pmatrix} \begin{pmatrix} f^-(x^*) \\ f_x^-(x^*) \\ f_{xx}^-(x^*) \\ f_{xxx}^-(x^*) \end{pmatrix} + \begin{pmatrix} O(h^4) \\ O(h^4) \\ O(h^4) \\ O(h^4) \end{pmatrix}. \quad (22)$$

The inverse of the matrix of the system in (22) also depends only on α and h.

$$B = \begin{pmatrix} -\frac{-24h^3+26h^2\alpha-9h\alpha^2+\alpha^3}{6h^3} & \frac{(-h+\alpha)(12h^2-7h\alpha+\alpha^2)}{2h^3} & -\frac{(-h+\alpha)(8h^2-6h\alpha+\alpha^2)}{2h^3} & \frac{(-h+\alpha)(6h^2-5h\alpha+\alpha^2)}{6h^3} \\ -\frac{26h^2-18h\alpha+3\alpha^2}{6h^3} & \frac{19h^2-16h\alpha+3\alpha^2}{2h^3} & -\frac{14h^2-14h\alpha+3\alpha^2}{2h^3} & \frac{11h^2-12h\alpha+3\alpha^2}{6h^3} \\ -\frac{-3h+\alpha}{h^3} & \frac{-8h+3\alpha}{h^3} & -\frac{-7h+3\alpha}{h^3} & \frac{-2h+\alpha}{h^3} \\ -h^{-3} & 3h^{-3} & -3h^{-3} & h^{-3} \end{pmatrix}. \quad (23)$$

As $\alpha = O(h)$, looking at the expressions of A and B in (21) and (23), it is easy to give an estimate of the precision of the calculated values from both sides of the discontinuity,

$$\begin{pmatrix} f^+(x^*) \\ f_x^+(x^*) \\ f_{xx}^+(x^*) \\ f_{xxx}^+(x^*) \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{pmatrix} \begin{pmatrix} f_j^+ \\ f_{j-1}^+ \\ f_{j-2}^+ \\ f_{j-3}^+ \end{pmatrix} + \begin{pmatrix} O(h^4) \\ O(h^3) \\ O(h^2) \\ O(h) \end{pmatrix}, \quad (24)$$

and,

$$\begin{pmatrix} f^-(x^*) \\ f_x^-(x^*) \\ f_{xx}^-(x^*) \\ f_{xxx}^-(x^*) \end{pmatrix} = \begin{pmatrix} B_{11} & B_{12} & B_{13} & B_{14} \\ B_{21} & B_{22} & B_{23} & B_{24} \\ B_{31} & B_{32} & B_{33} & B_{34} \\ B_{41} & B_{42} & B_{43} & B_{44} \end{pmatrix} \begin{pmatrix} f_j^- \\ f_{j-1}^- \\ f_{j-2}^- \\ f_{j-3}^- \end{pmatrix} + \begin{pmatrix} O(h^4) \\ O(h^3) \\ O(h^2) \\ O(h) \end{pmatrix}. \quad (25)$$

Now we just have to obtain the difference between (24) and (25) in order to arrive to (17) and the corresponding precision,

$$\begin{pmatrix} [f] \\ [f'] \\ [f''] \\ [f'''] \end{pmatrix} = \begin{pmatrix} f^+(x^*) - f^-(x^*) \\ f_x^+(x^*) - f_x^-(x^*) \\ f_{xx}^+(x^*) - f_{xx}^-(x^*) \\ f_{xxx}^+(x^*) - f_{xxx}^-(x^*) \end{pmatrix} + \begin{pmatrix} O(h^4) \\ O(h^3) \\ O(h^2) \\ O(h) \end{pmatrix}. \tag{26}$$

The same considerations lead us to the size of the jump conditions at smooth zones in the case that a bad detection occurs,

$$\begin{pmatrix} [f] \\ [f'] \\ [f''] \\ [f'''] \end{pmatrix} = \begin{pmatrix} f^+(x^*) - f^-(x^*) \\ f_x^+(x^*) - f_x^-(x^*) \\ f_{xx}^+(x^*) - f_{xx}^-(x^*) \\ f_{xxx}^+(x^*) - f_{xxx}^-(x^*) \end{pmatrix} = \begin{pmatrix} O(h^4) \\ O(h^3) \\ O(h^2) \\ O(h) \end{pmatrix}. \tag{27}$$

3.4 Correction Terms

In this section we will obtain the correction terms for the finite differences that appear in (13), (14), (15) and (16). Let's start by the correction terms for the finite differences that appear in $p_3(x)$, that corresponds to the case depicted in Fig. 1. Looking at that figure, we can see that the point where we want to interpolate $x_{j+1/2}$ belongs to the $-$ side of the interface as the discontinuity is in the interval $(x_j, x_{j+1/2})$. Thus, we can obtain the Taylor series of f_{j+1} and f_{j+2}, that also belong to the $-$ side, around $x_{j+1/2}$. f_{j-1} and f_j belong to the $+$ side of the discontinuity, so we need to express them in terms of the $-$ side using the information we have about the discontinuity. In order to do this, we can use the Taylor expansions of these values around x^*, use the jump relations $[f]$, $[f']$, $[f'']$, $[f''']$ in (17) to express the $+$ values in terms of the $-$ values, and then expand again around $x_{j+1/2}$ assuming that $f^-(x)$ and $f^+(x)$ are three times differentiable functions. Supposing that we know the jump conditions or that we have a good approximation, we are ready to obtain the Taylor expansions of f_{j-1}^+ and f_j^+ around the interpolation point $x_{j+1/2}$. The expressions that we will use to obtain the correction terms are the following,

$$\begin{aligned} f_{j+2}^- &= f^-(x_{j+1/2}) + f_x^-(x_{j+1/2})\left(\frac{3h}{2}\right) + \frac{1}{2} f_{xx}^-(x_{j+1/2})\left(\frac{3h}{2}\right)^2 \\ &\quad + \frac{1}{3!} f_{xxx}^-(x_{j+1/2})\left(\frac{3h}{2}\right)^3 + O(h^4), \end{aligned} \tag{28}$$

$$\begin{aligned} f_{j+1}^- &= f^-(x_{j+1/2}) + f_x^-(x_{j+1/2})\left(\frac{h}{2}\right) + \frac{1}{2} f_{xx}^-(x_{j+1/2})\left(\frac{h}{2}\right)^2 \\ &\quad + \frac{1}{3!} f_{xxx}^-(x_{j+1/2})\left(\frac{h}{2}\right)^3 + O(h^4), \end{aligned} \tag{29}$$

$$\begin{aligned} f_j^+ &= f^+(x^*) - f_x^+(x^*)\alpha + \frac{1}{2} f_{xx}^+(x^*)\alpha^2 - \frac{1}{3!} f_{xxx}^+(x^*)\alpha^3 + O(h^4) = [f] \\ &\quad + f^-(x^*) - \left([f'] + f_x^-(x^*)\right)\alpha + \frac{1}{2}\left([f''] + f_{xx}^-(x^*)\right)\alpha^2 \\ &\quad - \frac{1}{3!}\left([f'''] + f_{xxx}^-(x^*)\right)\alpha^3 + O(h^4) = [f] \end{aligned}$$

$$
\begin{aligned}
&+\left(f^-(x_{j+1/2})-f_x^-(x_{j+1/2})\right.\\
&\cdot\left(\frac{h}{2}-\alpha\right)+\frac{1}{2}f_{xx}^-(x_{j+1/2})\left(\frac{h}{2}-\alpha\right)^2\\
&\left.-\frac{1}{3!}f_{xxx}^-(x_{j+1/2})\left(\frac{h}{2}-\alpha\right)^3+O(h^4)\right)-\left(\left[f'\right]\right.\\
&+f_x^-(x_{j+1/2})-f_{xx}^-(x_{j+1/2})\left(\frac{h}{2}-\alpha\right)\\
&\left.+\frac{1}{2}f_{xxx}^-(x_{j+1/2})\left(\frac{h}{2}-\alpha\right)^2+O(h^3)\right)\alpha+\frac{1}{2}\left(\left[f''\right]\right.\\
&\left.+f_{xx}^-(x_{j+1/2})-f_{xxx}^-(x_{j+1/2})\left(\frac{h}{2}-\alpha\right)+O(h^2)\right)\alpha^2\\
&-\frac{1}{3!}\left(\left[f'''\right]+f_{xxx}^-(x_{j+1/2})+O(h)\right)\alpha^3, \qquad (30)
\end{aligned}
$$

$$
\begin{aligned}
f_{j-1}^+ &= f^+(x^*)-f_x^+(x^*)(h+\alpha)+\frac{1}{2}f_{xx}^+(x^*)(h+\alpha)^2\\
&\quad-\frac{1}{3!}f_{xxx}^+(x^*)(h+\alpha)^3+O(h^4)=[f]+f^-(x^*)\\
&\quad-\left(\left[f'\right]+f_x^-(x^*)\right)(h+\alpha)+\frac{1}{2}\left(\left[f''\right]+f_{xx}^-(x^*)\right)(h+\alpha)^2\\
&\quad-\frac{1}{3!}\left(\left[f'''\right]+f_{xxx}^-(x^*)\right)(h+\alpha)^3+O(h^4)\\
&=[f]+\left(f^-(x_{j+1/2})-f_x^-(x_{j+1/2})\left(\frac{h}{2}-\alpha\right)\right.\\
&\quad+\frac{1}{2}f_{xx}^-(x_{j+1/2})\left(\frac{h}{2}-\alpha\right)^2-\frac{1}{3!}f_{xxx}^-(x_{j+1/2})\\
&\quad\left.\cdot\left(\frac{h}{2}-\alpha\right)^3+O(h^4)\right)-\left(\left[f'\right]+f_x^-(x_{j+1/2})-f_{xx}^-(x_{j+1/2})\left(\frac{h}{2}-\alpha\right)\right.\\
&\quad\left.+\frac{1}{2}f_{xxx}^-(x_{j+1/2})\cdot\left(\frac{h}{2}-\alpha\right)^2+O(h^3)\right)(h+\alpha)\\
&\quad+\frac{1}{2}\left(\left[f''\right]+f_{xx}^-(x_{j+1/2})-f_{xxx}^-(x_{j+1/2})\left(\frac{h}{2}-\alpha\right)\right.\\
&\quad\left.+O(h^2)\right)\cdot(h+\alpha)^2-\frac{1}{3!}\left(\left[f'''\right]+f_{xxx}^-(x_{j+1/2})+O(h)\right)(h+\alpha)^3. \qquad (31)
\end{aligned}
$$

Now we just have to substitute these expressions in the divided differences that appear in $p_3(x)$, shown in (15), in order to obtain the correction terms.

Replacing (28), (29), (30) and (31) in the expression of $p_3(x)$ in (15) and having into account (26), we obtain the local truncation error for the approximation,

$$p_3(x_{j+1/2}) = f(x_{j+1/2}) + \frac{1}{2}[f] + \frac{1}{16}h[f'] - \frac{1}{2}\alpha[f'] - \frac{1}{32}h^2[f''] + \frac{1}{4}\alpha^2[f''] + \frac{1}{96}h^3[f'''] - \frac{1}{12}\alpha^3[f'''] + \frac{1}{32}\alpha h^2[f'''] + \frac{1}{32}\alpha^2 h[f'''] - \frac{1}{16}\alpha h[f''] + O(h^4), \tag{32}$$

and from this expression we know that we can express the approximation of $f(x_{j+1/2})$ as,

$$\begin{aligned} f(x_{j+1/2}) &= p_3(x_{j+1/2}) - \frac{1}{2}[f] - \frac{1}{16}h[f'] + \frac{1}{2}\alpha[f'] + \frac{1}{32}h^2[f''] \\ &\quad - \frac{1}{4}\alpha^2[f''] - \frac{1}{96}h^3[f'''] + \frac{1}{12}\alpha^3[f'''] - \frac{1}{32}\alpha h^2[f'''] \\ &\quad - \frac{1}{32}\alpha^2 h[f'''] + \frac{1}{16}\alpha h[f''] + O(h^4) \\ &= p_3(x_{j+1/2}) + C_3 + O(h^4), \end{aligned} \tag{33}$$

where C_3 is the correction term for the case depicted in Fig. 1. Thus, the approximation proposed gives a local truncation error that is $O(h^4)$ at well detected discontinuities in the interval $(x_j, x_{j+1/2})$ (see Fig. 1) if the correction terms are included. At smooth zones, if the algorithm has encountered a false detection, we also obtain $O(h^4)$ local truncation error if we have in mind (27) and (32) and that $\alpha \leq h$.

Let's now analyze the case depicted in Fig. 2, where the discontinuity is placed at the right half of the central interval of the stencil. As mentioned before, in this example it is convenient to use $p_1(x)$ in (13). In this case, $x_{j+1/2}$ belongs to the $+$ side and we can obtain the Taylor series of f_j and f_{j-1}, that also belong to the $+$ side, around $x_{j+1/2}$. f_{j+1} and f_{j+2} belong to the $-$ side of the discontinuity, so we need to express them in terms of the $+$ side using the jump conditions (17). In order to do so, we can proceed as before and use the Taylor expansions of these values around x^*, use the jump relations $[f], [f'], [f''], [f''']$ and then expand again around $x_{j+1/2}$. For brevity we will only show the final expressions, as the process to obtain them has been explained before for the first case analyzed. The local truncation error of the approximation at $f(x_{j+1/2})$ is in this case,

$$\begin{aligned} f(x_{j+1/2}) &= p_1(x_{j+1/2}) + 1/2[f] + \frac{7}{16}h[f'] - \frac{1}{2}\alpha[f'] - \frac{5}{32}\alpha h^2[f'''] \\ &\quad - \frac{7}{16}\alpha h[f''] + \frac{7}{32}\alpha^2 h[f'''] + \frac{5}{32}h^2[f''] + \frac{1}{4}\alpha^2[f''] \\ &\quad + \frac{1}{96}h^3[f'''] - \frac{1}{12}\alpha^3[f'''] + O(h^4) \\ &= p_1(x_{j+1/2}) + C_1 + O(h^4), \end{aligned} \tag{34}$$

where C_1 is the correction term for the case depicted in Fig. 2. Thus, the approximation proposed results in a local truncation error that is $O(h^4)$ at well detected discontinuities in the interval $(x_{j+1/2}^{k-1}, x_{j+1}^{k-1})$. At smooth zones, if the algorithm has encountered a false detection, we also attain $O(h^4)$ accuracy if we have in mind (27).

Let's now analyze the case depicted in Fig. 3, where the discontinuity is placed at the interval (x_{j-1}, x_j), that is to the left of the central interval of the stencil. In this example, it is convenient to use $p_2(x)$ in (14) in order to minimize the effect of the discontinuity. In this case, $x_{j+1/2}$ belongs to the $-$ side of the interface. Thus, we need to obtain the Taylor series of f_j, f_{j+1} and f_{j+2}, that also belong to the $-$ side, around $x_{j+1/2}$. f_{j-1} belongs to the $+$ side of the discontinuity, so we need to express it in terms of the $-$ side using the jump

conditions (17). In order to do so, we can proceed as before and use the Taylor expansions of this value around x^*, use the jump relations $[f]$, $[f']$, $[f'']$, $[f''']$, in order to express it in terms of the $-$ side, and then expand again around $x_{j+1/2}$. In this case the local truncation for the approximation at $f(x_{j+1/2})$ is,

$$\begin{aligned} f(x_{j+1/2}) &= p_2(x_{j+1/2}) + \frac{1}{16}[f] - \frac{1}{16}\alpha[f'] + \frac{1}{32}\alpha^2[f''] - \frac{1}{96}\alpha^3[f'''] + O(h^4) \\ &= p_2(x_{j+1/2}) + C_2 + O(h^4), \end{aligned} \tag{35}$$

where C_2 is the correction term for the case depicted in Fig. 3. Thus, the approximation proposed attains again a local truncation error that is $O(h^4)$ at well detected discontinuities in the interval (x_{j-1}, x_j). The same considerations as before can be taken for accuracy and false detections in this case.

Finally, let's analyze the case depicted in Fig. 4, where the discontinuity is placed at the interval that is to the right of the central interval of the stencil. In this example, it is convenient to use $p_4(x)$ in (16). Following the same process we obtain the local truncation error for $f(x_{j+1/2})$,

$$\begin{aligned} f(x_{j+1/2}) &= p_4(x_{j+1/2}) + 1/16h\alpha[f''] - 1/32h\alpha^2[f'''] \\ &\quad + 1/32h^2\alpha[f'''] - 1/16[f] - 1/16h[f'] \\ &\quad + 1/16\alpha[f'] - 1/32h^2[f''] - 1/32\alpha^2[f''] - \frac{1}{96}h^3[f'''] \\ &\quad + \frac{1}{96}\alpha^3[f'''] + O(h^4) \\ &= p_4(x_{j+1/2}) + C_4 + O(h^4), \end{aligned} \tag{36}$$

where C_4 is the correction term for the case depicted in Fig. 4. Thus, the approximation proposed attains again a local truncation error that is $O(h^4)$ at well detected discontinuities in the interval (x_{j+1}, x_{j+2}). Again, the same considerations as before can be taken for accuracy and false detections here.

Remark 1 At this point it is convenient to think about the process of detection. If the algorithm detects a discontinuity at the central interval of the stencil, as depicted in Figs. 1 and 2, the immediately preceding stencil (situation presented in Fig. 4) and the immediately succeeding (situation presented in Fig. 3) will be affected by the discontinuity. Thus, whenever we detect a discontinuity at the central interval of the stencil we will use the expressions in (15) or (13) amended adequately with the correction terms presented before. Then, it will be necessary to reprocess the previous and the following stencil using expressions in (16) and (14) plus correction terms respectively in order to treat adequately the discontinuity when interpolating at $x_{j-1/2}$ and $x_{j+3/2}$.

Remark 2 Of course, the expression of the polynomials $p_1(x)$, $p_2(x)$, $p_3(x)$ and $p_4(x)$ is the same, taking the form at $x_{j+1/2}$,

$$p_i(x_{j+1/2}) = -\frac{1}{16}f_{j-1} + \frac{9}{16}f_j + \frac{9}{16}f_{j+1} - \frac{1}{16}f_{j+2}, \quad i = 1, 2, 3, 4. \tag{37}$$

This means that the process described before is simplified, as we can always substitute (37) in the expressions (33), (34), (35) and (36) in order to interpolate at $x_{j+1/2}$, and just add the corresponding correction terms in the cases that they are needed. What is to say, we will use,

$$f(x_{j+1/2}) = -\frac{1}{16}f_{j-1} + \frac{9}{16}f_j + \frac{9}{16}f_{j+1} - \frac{1}{16}f_{j+2} + C_i + O\left(h^4\right), \quad i = \{1,2,3,4\}, \tag{38}$$

where the correction terms in (33), (34), (35) and (36) will take into account the jumps in the function and its derivatives in order to provide $O(h^4)$ accuracy.

From all the previous considerations we can state the following theorem.

Theorem 1 (Point-values and corner singularities) *Let f be a continuous function with an isolated singularity of f' at x^* and denote by $\mathcal{S}_h f$ our adapted interpolating polynomial of fourth order. For all continuous f with derivatives up to degree four uniformly bounded on $\mathbb{R}\backslash\{x^*\}$, where x^* is the position of the corner singularity, the adapted interpolating polynomial of fourth order $\mathcal{S}_h f$ satisfies,*

$$||f - \mathcal{S}_h f||_{L^\infty} \leq Ch^2 \sup_{\mathbb{R}\backslash\{x^*\}} |f''|, \tag{39}$$

for all $h > 0$, with $C > 0$ independent of f. Moreover there exists $0 < K < 1$ independent of f such that for $h < Kh_c$ with h_c defined in (10), we have,

$$||f - \mathcal{S}_h f||_{L^\infty} \leq Ch^4 \max\{sup_{\mathbb{R}\backslash\{x^*\}}|f^{(iv)}|, |[f^{(iv)}]|\}. \tag{40}$$

Proof If $h > Kh_c$ the discontinuity is susceptible of not being detected. If this is the case, the ENO-SR method and our algorithm reduce to the classical centered Lagrange interpolation and both have the same error. In Theorem 1 of [12], the authors provide a proof for the bound specified in the first part of the theorem.

For the second part of the theorem, lets suppose that y is the position of the discontinuity. x^* is the approximation of y obtained finding the roots of $H(x) = p^+(x) - p^-(x)$ at the suspicious cell, and $p^+(x)$, $p^-(x)$ are mth order polynomials to the right and to the left of the discontinuity. Following the third statement of Lemma 3 in [12], the error obtained in the approximation y of the position of the discontinuity will be,

$$|x^* - y| \leq C\frac{h^m \sup_{\mathbb{R}\backslash\{x^*\}} |f^{(m)}|}{|[f']|}. \tag{41}$$

As introduced in Sect. 3.3, we need to obtain the distance $\alpha = x^* - x_j^{k-1}$ in order to compute the correction terms. We can see that the error obtained when approximating the distance α (see for example Fig. 5) is given precisely by (41). We need $|x^* - y| = O(h^4)$ in order to assure global $O(h^4)$ accuracy. Looking at the expressions of the correction terms in (33), (34), (35) and (36), we need $m = 4$ in (41). This means that $p^+(x)$ and $p^-(x)$ must be fourth order polynomials. Satisfying this condition and using Taylor expansion to develop expressions (33), (34), (35) and (36) to $O(h^5)$ we finish the proof. We only present the expansion of (33), as the rest can be obtained in a similar way,

$$\begin{aligned} f(x_{j+1/2}) = {} & p_3(x_{j+1/2}) - \frac{1}{2}[f] - \frac{1}{16}h[f'] + \frac{1}{2}\alpha[f'] + \frac{1}{32}h^2[f''] \\ & - \frac{1}{4}\alpha^2[f''] - \frac{1}{96}h^3[f'''] + \frac{1}{12}\alpha^3[f'''] \\ & - \frac{1}{32}\alpha h^2[f'''] - \frac{1}{32}\alpha^2 h[f'''] + \frac{1}{16}\alpha h[f''] - \frac{3}{128}(f^-)^{(iv)}h^4 \\ & - \left(\frac{1}{96}h\alpha^3 + \frac{1}{64}h^2\alpha^2 + \frac{1}{96}h^3\alpha + \frac{1}{384}h^4 - \frac{1}{48}\alpha^4\right)[f^{iv}] + O(h^5) \end{aligned}$$

$$= p_3(x_{j+1/2}) + C_3 - \left(\frac{1}{96}h\alpha^3 + \frac{1}{64}h^2\alpha^2 + \frac{1}{96}h^3\alpha + \frac{1}{384}h^4 - \frac{1}{48}\alpha^4\right)[f^{iv}]$$
$$- \frac{3}{128}(f^-)^{(iv)}h^4 + O(h^5), \tag{42}$$

thus,

$$f(x_{j+1/2}) - p_3(x_{j+1/2}) - C_3 = -\left(\frac{1}{96}h\alpha^3 + \frac{1}{64}h^2\alpha^2 + \frac{1}{96}h^3\alpha + \frac{1}{384}h^4 - \frac{1}{48}\alpha^4\right)[f^{iv}]$$
$$- \frac{3}{128}(f^-)^{(iv)}h^4 + O(h^5),$$

and then,

$$||f - \mathcal{S}_h f||_{L^\infty} \leq Ch^4, \max\{sup_{\mathbb{R}\setminus\{x^*\}}|f^{(iv)}|, |[f^{(iv)}]|\}. \tag{43}$$

□

Remark 3 It is important to realize that the nonlinearity of the algorithm resides only in the obtention of the jump conditions, that is data dependent, i.e. in multiresolution applications the jump conditions can be obtained at each scale. If the jump conditions are known or they are obtained at the coarsest scale and then used at the rest of the scales, then the algorithm is linear. Moreover, obtaining the jump conditions at the coarsest scale and then using them at the finer scales would be equivalent to an error control strategy. It is known that error control is useful for stabilizing non stable algorithms and that error control applied to stable algorithms lead to better compression properties compared to unstable algorithms plus error control [6,30].

In order to present the new method in a nutshell, we introduce Algorithm 1.

Remark 4 Algorithm 1 can be directly adapted for multiresolution just calculating the errors of approximation at each scale and storing them in the encoding step. In the decoding step, the algorithm is exactly the same, but we would add the stored errors to the predictions obtained by the new algorithm.

3.5 Correction Terms for Newton Polynomials in the Cell-Averages

Working with the primitive function (5) and taking into account the expression in (6), it is easy to arrive to the expressions of the polynomials for the cell-averages from (37) at smooth zones. When the stencil touches the discontinuity, it is also possible to obtain the expressions of the polynomials going to the primitive function and using (38). Once we have the primitive F^k, we can interpolate using the point-values of the primitive function (just as described in previous subsection) and then go back to the cell-averages using (6). It is clear that during this process the correction terms for the primitive are only divided by $h/2$ in the cell averages. A second option is to work directly in the cell-averages taking the following steps:

1. Find the position of the discontinuity and decide if it is to the right or to the left of the centre of the central cell of the stencil.
2. Use the polynomial in (37) translated to the cell-averages to interpolate at $x_{j+1/2}$. In (50) we show the expression for (37) in the cell-averages.
3. Once we have taken previous steps, we just have to add the corresponding correction term (33), (34), (35) or (36) presented in previous subsection divided by $h/2$. These correction terms can be expressed directly in the cell-averages, as we will show in what follows.

Algorithm 1 The new algorithm in the point-values

for $j = L, \ldots, 1$ (Loop for the L scales of subdivision) **do**
  for $i = 2, \ldots, N_{j-2}$ (Loop for the physical domain at the scale j) **do**
    Use the detection mechanism presented in [12] centered at the point x_i^j.
    Build equation $H(x) = g_1(x) - g_2(x) = 0$ and obtain its roots in order to find the position α of the discontinuity.
    if a discontinuity is placed inside the central interval (x_j, x_{j+1}) of the stencil **then**
      Obtain *jump conditions* if they are not known.
      if the discontinuity is placed at the left half of the interval (x_j, x_{j+1}) **then**
        Use (33) to obtain $f(x_{j+1/2}) = -\frac{1}{16}f_{j-1} + \frac{9}{16}f_j + \frac{9}{16}f_{j+1} - \frac{1}{16}f_{j+2} + C_3 + O(h^4)$
        Use (35) to obtain $f(x_{j-1/2}) = -\frac{1}{16}f_{j-1} + \frac{9}{16}f_j + \frac{9}{16}f_{j+1} - \frac{1}{16}f_{j+2} + C_2 + O(h^4)$
        Use (36) to obtain $f(x_{j+3/2}) = -\frac{1}{16}f_{j-1} + \frac{9}{16}f_j + \frac{9}{16}f_{j+1} - \frac{1}{16}f_{j+2} + C_4 + O(h^4)$
        Set $i = i + 1$ in order not to process $f(x_{j+3/2})$ in the next iteration.
      else if the discontinuity is placed at the right half of the interval (x_j, x_{j+1}) **then**
        Use (34) to obtain $f(x_{j+1/2}) = -\frac{1}{16}f_{j-1} + \frac{9}{16}f_j + \frac{9}{16}f_{j+1} - \frac{1}{16}f_{j+2} + C_1 + O(h^4)$
        Use (35) to obtain $f(x_{j-1/2}) = -\frac{1}{16}f_{j-1} + \frac{9}{16}f_j + \frac{9}{16}f_{j+1} - \frac{1}{16}f_{j+2} + C_2 + O(h^4)$
        Use (36) to obtain $f(x_{j+3/2}) = -\frac{1}{16}f_{j-1} + \frac{9}{16}f_j + \frac{9}{16}f_{j+1} - \frac{1}{16}f_{j+2} + C_4 + O(h^4)$
        Set $i = i + 1$ in order not to process $f(x_{j+3/2})$ in the next iteration.
      else
        False detection, so we apply fourth order central Newton interpolation without correction terms.
      end if
    else
      No detection of discontinuity, so we apply fourth order central Newton interpolation without correction terms.
    end if
  end for
end for

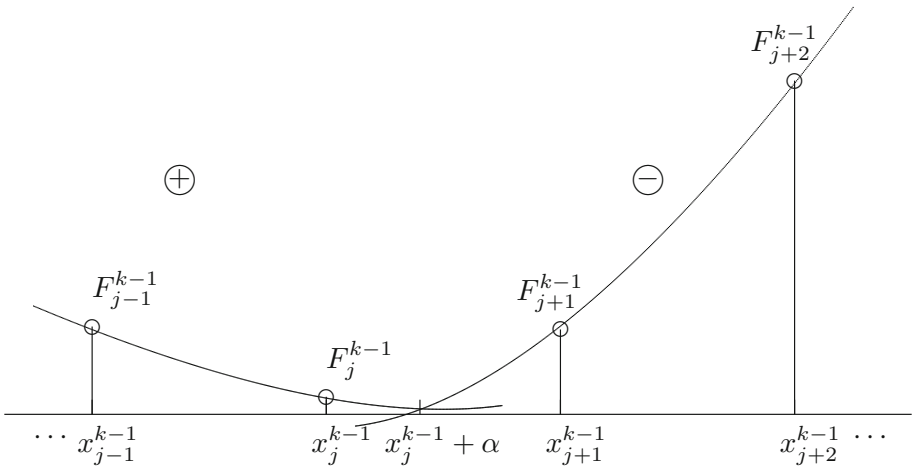

Fig. 6 In this figure we can see an example of a corner singularity in the primitive placed in the interval $(x_j^{k-1}, x_{j+1}^{k-1})$ at a position $x^* = x_j^{k-1} + \alpha$

Let's suppose that we are working in the cell-averages and that the discontinuity has been found in the interval (x_j, x_{j+1}) at a distance α from x_j, as shown in Fig. 6. In this case, working with the point-values of the primitive function (5), we have that the system in (18) can be written in matrix form as,

$$\begin{pmatrix} F_j^+ \\ F_{j-1}^+ \\ F_{j-2}^+ \\ F_{j-3}^+ \end{pmatrix} = \begin{pmatrix} 1 & -\alpha & \frac{1}{2}\alpha^2 & -\frac{1}{3!}\alpha^3 \\ 1 & -(h+\alpha) & \frac{1}{2}(h+\alpha)^2 & -\frac{1}{3!}(h+\alpha)^3 \\ 1 & -(2h+\alpha) & \frac{1}{2}(2h+\alpha)^2 & -\frac{1}{3!}(2h+\alpha)^3 \\ 1 & -(3h+\alpha) & \frac{1}{2}(3h+\alpha)^2 & -\frac{1}{3!}(3h+\alpha)^3 \end{pmatrix} \begin{pmatrix} F^+(x^*) \\ F_x^+(x^*) \\ F_{xx}^+(x^*) \\ F_{xxx}^+(x^*) \end{pmatrix}. \tag{44}$$

Where we have dropped the $O(h^4)$ term for simplicity, as we are not interested now in analyzing the accuracy. The inverse of the matrix of the system in (44) depends only on α and h, as shown in (21). Let's express now the inverse of the system in (44) in terms of the cell-averages using the expression of the primitive values (5),

$$\begin{pmatrix} F^+(x^*) \\ F_x^+(x^*) \\ F_{xx}^+(x^*) \\ F_{xxx}^+(x^*) \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{pmatrix} \begin{pmatrix} h\sum_{i=1}^{j} \bar{f}_i^+ \\ h\sum_{i=1}^{j-1} \bar{f}_i^+ \\ h\sum_{i=1}^{j-2} \bar{f}_i^+ \\ h\sum_{i=1}^{j-3} \bar{f}_i^+ \end{pmatrix}. \tag{45}$$

Working in the point-values of the primitive, the system in (19) can be written in matrix form as,

$$\begin{pmatrix} F_{j+1}^- \\ F_{j+2}^- \\ F_{j+3}^- \\ F_{j+4}^- \end{pmatrix} = \begin{pmatrix} 1 & h-\alpha & 1/2\ (h-\alpha)^2 & 1/6\ (h-\alpha)^3 \\ 1 & 2\,h-\alpha & 1/2\ (2\,h-\alpha)^2 & 1/6\ (2\,h-\alpha)^3 \\ 1 & 3\,h-\alpha & 1/2\ (3\,h-\alpha)^2 & 1/6\ (3\,h-\alpha)^3 \\ 1 & 4\,h-\alpha & 1/2\ (4\,h-\alpha)^2 & 1/6\ (4\,h-\alpha)^3 \end{pmatrix} \begin{pmatrix} F^-(x^*) \\ F_x^-(x^*) \\ F_{xx}^-(x^*) \\ F_{xxx}^-(x^*) \end{pmatrix}. \tag{46}$$

The inverse of the matrix of the system in (46) also depends only on α and h, as shown in (23). Let's express now the inverse of the system in (46) in terms of the cell-averages using the expression of the primitive values in (5),

$$\begin{pmatrix} F^-(x^*) \\ F_x^-(x^*) \\ F_{xx}^-(x^*) \\ F_{xxx}^-(x^*) \end{pmatrix} = \begin{pmatrix} B_{11} & B_{12} & B_{13} & B_{14} \\ B_{21} & B_{22} & B_{23} & B_{24} \\ B_{31} & B_{32} & B_{33} & B_{34} \\ B_{41} & B_{42} & B_{43} & B_{44} \end{pmatrix} \begin{pmatrix} h\sum_{i=1}^{j+1} \bar{f}_i^- \\ h\sum_{i=1}^{j+2} \bar{f}_i^- \\ h\sum_{i=1}^{j+3} \bar{f}_i^- \\ h\sum_{i=1}^{j+4} \bar{f}_i^- \end{pmatrix}. \tag{47}$$

From these expressions and after some algebraic manipulations, the jump relations can be expressed in terms of the cell-averages as,

$$\begin{aligned} [F] = {} & \frac{\left(3\,h\alpha^2+\alpha^3+2\,h^2\alpha\right)}{6h^2}\bar{f}_{j-2} + \frac{\left(-2\,\alpha^3-7\,h^2\alpha-9\,h\alpha^2\right)}{6h^2}\bar{f}_{j-1} \\ & + \frac{\left(\alpha^3+11\,h^2\alpha+6\,h\alpha^2\right)}{6h^2}\bar{f}_j - \bar{f}_{j+1} + \frac{\left(-26\,h^2\alpha+9\,h\alpha^2-\alpha^3+18\,h^3\right)}{6h^2}\bar{f}_{j+2} \\ & + \frac{\left(-18\,h^3+31\,h^2\alpha-15\,h\alpha^2+2\,\alpha^3\right)}{6h^2}\bar{f}_{j+3} + \frac{\left(6\,h^3-11\,h^2\alpha-\alpha^3+6\,h\alpha^2\right)}{6h^2}\bar{f}_{j+4}, \end{aligned}$$

$$
\begin{aligned}
[F'] &= \frac{\left(2\,h^2+6\,h\alpha+3\,\alpha^2\right)}{6h^2}\bar{f}_{j-2}+\frac{\left(-18\,h\alpha-6\,\alpha^2-7\,h^2\right)}{6h^2}\bar{f}_{j-1}\\
&+\frac{\left(11\,h^2+12\,h\alpha+3\,\alpha^2\right)}{6h^2}\bar{f}_{j}+\frac{\left(18\,h\alpha-26\,h^2-3\,\alpha^2\right)}{6h^2}\bar{f}_{j+2}\\
&+\frac{\left(-30\,h\alpha+31\,h^2+6\,\alpha^2\right)}{6h^2}\bar{f}_{j+3}+\frac{\left(-11\,h^2+12\,h\alpha-3\,\alpha^2\right)}{6h^2}\bar{f}_{j+4},\\
[F''] &= \frac{(h+\alpha)}{h^2}\bar{f}_{j-2}-\frac{(2\,\alpha+3\,h)}{h^2}\bar{f}_{j-1}+\frac{(2\,h+\alpha)}{h^2}\bar{f}_{j}+\frac{(3\,h-\alpha)}{h^2}\bar{f}_{j+2}\\
&+\frac{(-5\,h+2\,\alpha)}{h^2}\bar{f}_{j+3}+\frac{(2\,h-\alpha)}{h^2}\bar{f}_{j+4},\\
[F'''] &= \frac{1}{h^2}\bar{f}_{j-2}-\frac{2}{h^2}\bar{f}_{j-1}+\frac{1}{h^2}\bar{f}_{j}-\frac{1}{h^2}\bar{f}_{j+2}+\frac{2}{h^2}\bar{f}_{j+3}-\frac{1}{h^2}\bar{f}_{j+4},
\end{aligned}
\tag{48}
$$

where we have used the expressions for A in (21) and for B in (23).

As conclusion, we can say that it is possible to work directly in the cell-averages. This is clear if we have into account that at soft zones we will only need the expression in (37) for the primitive values,

$$
p_i(x_{j+1/2}) = p(x_{j+1/2}) = -\frac{1}{16}F_{j-1}+\frac{9}{16}F_j+\frac{9}{16}F_{j+1}-\frac{1}{16}F_{j+2},\quad i=1,2,3,4, \tag{49}
$$

where, in this case, $p(x)$ is the interpolant for the point-values of the primitive. Using (6) it can be transformed to the cell-averages,

$$
\bar{p}(x_{j+\frac{1}{2}}) = \frac{p(x_{j+\frac{1}{2}})-F_j}{h/2} = \frac{1}{8}\left(\bar{f}_j+8\bar{f}_{j+1}-\bar{f}_{j+2}\right), \tag{50}
$$

At the discontinuities we have the four possible interpolants for the primitive that can be obtained from the expressions for the point-values (33), (34), (35) and (36),

$$
\begin{aligned}
p_3(x_{j+1/2}) &= p(x_{j+1/2})+\frac{1}{2}[F]+\frac{1}{16}h[F']-\frac{1}{2}\alpha[F']-\frac{1}{32}h^2[F'']\\
&+\frac{1}{4}\alpha^2[F'']+\frac{1}{96}h^3[F''']-\frac{1}{12}\alpha^3[F''']+\frac{1}{32}\alpha h^2[F''']\\
&+\frac{1}{32}\alpha^2 h[F''']-\frac{1}{16}\alpha h[F''],\\
p_1(x_{j+1/2}) &= p(x_{j+1/2})-1/2[F]-\frac{7}{16}h[F']+\frac{1}{2}\alpha[F']+\frac{5}{32}\alpha h^2[F''']\\
&+\frac{7}{16}\alpha h[F'']-\frac{7}{32}\alpha^2 h[F''']-\frac{5}{32}h^2[F'']-\frac{1}{4}\alpha^2[F'']\\
&-\frac{1}{96}h^3[F''']+\frac{1}{12}\alpha^3[F'''],\\
p_2(x_{j+1/2}) &= p(x_{j+1/2})-\frac{1}{16}[F]+\frac{1}{16}\alpha[F']-\frac{1}{32}\alpha^2[F'']+\frac{1}{96}\alpha^3[F'''],\\
p_4(x_{j+1/2}) &= p(x_{j+1/2})-1/16h\alpha[F'']+1/32h\alpha^2[F''']-1/32h^2\alpha[F''']\\
&+1/16[F]+1/16h[F']-1/16\alpha[F']+1/32h^2[F'']+1/32\alpha^2[F'']\\
&+\frac{1}{96}h^3[F''']-\frac{1}{96}\alpha^3[F'''],
\end{aligned}
$$

where the expressions for $[F]$, $\left[F'\right]$, $\left[F''\right]$ and $\left[F'''\right]$ can be found in (48) and the expression for $p(x_{j+1/2})$ can be found in (49). In order to obtain the interpolants for the cell-averages we only need to use (6) using $h/2$ as grid-spacing,

$$\begin{aligned}
\bar{p}_3(x_{j+1/2}) &= \frac{p_3(x_{j+1/2}) - F_j}{\frac{h}{2}} = \bar{p}(x_{j+1/2}) + \frac{2}{h}\Big(\frac{1}{2}[F] + \frac{1}{16}h[F'] \\
&\quad - \frac{1}{2}\alpha[F'] - \frac{1}{32}h^2[F''] + \frac{1}{4}\alpha^2[F''] + \frac{1}{96}h^3[F'''] \\
&\quad - \frac{1}{12}\alpha^3[F'''] + \frac{1}{32}\alpha h^2[F'''] + \frac{1}{32}\alpha^2 h[F'''] - \frac{1}{16}\alpha h[F'']\Big), \\
\bar{p}_1(x_{j+1/2}) &= \frac{p_1(x_{j+1/2}) - F_j}{\frac{h}{2}} = \bar{p}(x_{j+1/2}) + \frac{2}{h}\Big(-\frac{1}{2}[F] - \frac{7}{16}h[F'] + \frac{1}{2}\alpha[F'] \\
&\quad + \frac{5}{32}\alpha h^2[F'''] + \frac{7}{16}\alpha h[F''] - \frac{7}{32}\alpha^2 h[F'''] - \frac{5}{32}h^2[F''] \\
&\quad - \frac{1}{4}\alpha^2[F''] - \frac{1}{96}h^3[F'''] + \frac{1}{12}\alpha^3[F'''] \Big), \\
\bar{p}_4(x_{j+1/2}) &= \frac{p_4(x_{j+1/2}) - F_j}{\frac{h}{2}} = \bar{p}(x_{j+1/2}) \\
&\quad + \frac{2}{h}\Big(-1/16h\alpha[F''] + 1/32h\alpha^2[F'''] - 1/32h^2\alpha[F'''] \\
&\quad + 1/16[F] + 1/16h[F'] - 1/16\alpha[F'] + 1/32h^2[F''] + 1/32\alpha^2[F''] \\
&\quad + \frac{1}{96}h^3[F'''] - \frac{1}{96}\alpha^3[F'''] \Big), \\
\bar{p}_2(x_{j+1/2}) &= \frac{p_2(x_{j+1/2}) - F_j}{\frac{h}{2}} = \bar{p}(x_{j+1/2}) \\
&\quad + \frac{2}{h}\Big(-\frac{1}{16}[F] + \frac{1}{16}\alpha[F'] - \frac{1}{32}\alpha^2[F''] + \frac{1}{96}\alpha^3[F'''] \Big).
\end{aligned} \tag{51}$$

Mind that the correction terms only depend on the jump conditions and that the jump conditions can be expressed in terms of h, α and the cell-average values (48).

Once we know for sure that it is possible to work directly in the cell-averages, we can state a theorem about the accuracy of the adapted polynomial in the cell-averages, just as we did in Theorem 1 of previous subsection for the point-values.

Theorem 2 (Cell-averages and jump discontinuities) *Let's define* $\mathcal{A}_h\bar{f} := (\mathcal{S}_h F)'$, *where the* $\mathcal{S}_h F$ *is the adapted polynomial of fourth order working with the primitive values of the cell-averages* $\bar{f}$, *then, for all continuous* f *with derivatives up to degree three uniformly bounded on* $\mathbb{R}\backslash\{x^*\}$, *where* x^* *is the position of a jump discontinuity, the adapted polynomial of third order in the cell-averages* $\mathcal{A}_h\bar{f}$ *satisfies,*

$$||f - \mathcal{A}_h\bar{f}||_{L^1} \leq Ch^2, \tag{52}$$

for all $h > 0$. *Moreover there exists* $0 < K < 1$ *independent of* f *such that for* $h < Kh_c$ *with* h_c *defined in* (10), *we have that,*

$$||f - \mathcal{A}_h\bar{f}||_{L^1} \leq Ch^3. \tag{53}$$

The proof is straightforward taking into account the primitive function and the proof in Theorem 1.

Remark 5 Algorithm 1 can be used directly for the cell averages changing the predictions in the point values for those presented in (51) with $\bar{p}(x_{j+\frac{1}{2}})$ given in (50). The detection of the discontinuity presented in [12] can be done directly in the cell averages using first order differences instead of second order differences. The expressions of the prediction, the jump relations, if they are unknown, and the correction terms have been given in the cell-averages previously in this subsection.

4 Numerical Experiments: Accuracy, Stability and Comparison Among Different Algorithms

In this section we present some experiments oriented to compare the results of the new algorithm proposed with ENO-SR in the point-value and the cell-average setting. It is known that ENO-SR is capable of obtaining full accuracy close to discontinuities [10]. We will check the error of interpolation obtained at every grid point or every cell inside an interval around the discontinuity and then perform a grid refinement analysis in order to check if the new algorithm also attains full accuracy close to discontinuities. We will also try to determine the numerical stability of the algorithms considered.

Along this section we will use the discrete norms,

$$||f^k||_1 = \frac{1}{N_k}\sum_j |f_j^k|$$

$$||f^k||_2^2 = \frac{1}{N_k}\sum_j |f_j^k|^2$$

$$||f^k||_\infty = \sup_j |f_j^k|,$$

where k is the scale of multiresolution and N_k is the size of the discretized data at each scale.

4.1 Experiments in the Point-Values

In this subsection we perform some experiments in order to show the order of convergence of the new algorithm proposed in the point-values. Thus, we will obtain the discretized data through the sampling described in (1). We will also present some experiments for subdivision. We will use the function in (55) presented in Fig. 7. Similar results have been obtained for a large battery of piecewise continuous functions.

For the grid refinement analysis, we define the order of accuracy of the reconstruction as,

$$order = \log_2\left(\frac{e_i}{e_{i+1}}\right), \tag{54}$$

being e_i the absolute error obtained with a grid spacing h and e_{i+1} the absolute error obtained with a grid spacing $h/2$.

Example 1 We consider the function in (55), plotted in Fig. 7,

$$f(x) = \begin{cases} -\sin(5(x-0.5-\alpha)), & 0 \le x < \alpha, \\ 5(x-0.5-\alpha)^2, & \alpha \le x < 1, \end{cases} \tag{55}$$

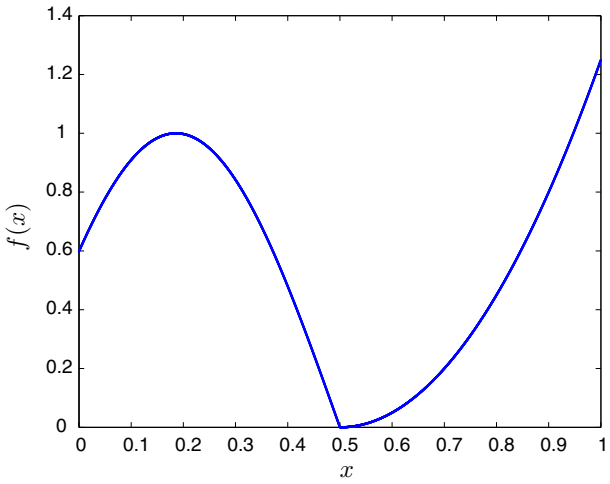

Fig. 7 Representation of the function in (55)

with $\alpha = \frac{h_0}{7}$, being $h_0 = \frac{1}{2^{12}}$. In this experiment we set $h = \frac{1}{2^i}$, $i = 5, 6, 7, 8, \ldots, 11$, i.e. $n = 32, 64, 128, \ldots, 2048$ grid points, in order to check the accuracy of the interpolation through a grid refinement analysis. In order to obtain the results, we perform one scale of subdivision, i.e. interpolation, with the number of points n shown in the tables. Table 1 shows the error distribution obtained for $x \in \{\frac{1}{2} - \frac{3h}{2}, \frac{1}{2} - h, \frac{1}{2} - \frac{h}{2}, \frac{1}{2}, \frac{1}{2} + \frac{h}{2}, \frac{1}{2} + h, \frac{1}{2} + \frac{3h}{2}\}$ for ENO-SR in the point-value setting. Table 2 shows the error distribution obtained at the same points for the new algorithm. The two tables show the order of accuracy defined in (54) when refining the precision of the grid. We can see how both algorithms present full accuracy. Even though, the errors obtained by the new algorithm are smaller or equal than those attained by ENO-SR in most of the cases presented. Mind that the function to the right of the discontinuity is a parabola, so the machine precision is reached from the beginning of the experiment. Logically, this fact affects the numerical order of accuracy of both algorithms. For both algorithms the error is directly zero in some cases, so we can not obtain the order of accuracy.

In Fig. 8 we can see an experiment for subdivision. We have set four scales of multiresolution and we have truncated all the details for the function in (55). The finest grid is 4096 points. This means that the experiment is equivalent to doing four scales of subdivision starting from 256 points. Figure 8 to the left shows the absolute error of the ENO-SR reconstruction around the discontinuity. Figure 8 to the right shows the absolute error obtained using the new algorithm. It is interesting to see that the greatest error for both algorithms is due to a false detection at the maximum of the function close to $x = 0.2$. The error distribution for the new algorithm around the false detection is better that the one obtained by ENO-SR. What is more, the error obtained by the new algorithm is smaller in all the norms, as it can be observed in Table 3.

As a conclusion for this section, we can say that the numerical results confirm what it could be expected from the theoretical analysis of the algorithm presented in Sect. 3. The correction terms introduced allow to reduce the local truncation error and to obtain full accuracy in the presence of corner singularities in the point values. The errors presented in Tables 1, 2 and

Table 1 Grid refinement analysis for ENO-SR applied to the function in (55) in the point-value setting

n	$x = 1/2 - 3h/2$		$x = 1/2 - h$		$x = 1/2 - h/2$		$x = 1/2$		$x = 1/2 + h/2$		$x = 1/2 + h$		$x = 1/2 + 3h/2$	
	e_i	$Order$	e_i	$Order$	e_i	$Order$	e_i	$Order$	e_i	$Order$	e_i	$Order$	e_i	$Order$
32	6.102e−06	–	8.173e−06	–	4.689e−06	–	3.253e−18	–	0	–	6.939e−18	–	6.939e−18	–
64	1.928e−07	4.984	2.602e−07	4.973	1.477e−07	4.989	1.735e−18	0.907	1.301e−18	–	8.674e−19	3.000	0	–
128	6.050e−09	4.994	8.175e−09	4.992	4.632e−09	4.995	3.795e−19	2.193	2.168e−19	2.585	2.168e−19	2.000	0	–
256	1.897e−10	4.995	2.562e−10	4.996	1.453e−10	4.994	1.559e−19	1.284	2.711e−20	3.000	5.421e−20	2.000	1.084e−19	–
512	5.960e−12	4.992	8.041e−12	4.994	4.572e−12	4.990	7.623e−21	4.354	0	–	2.711e−20	1.000	2.711e−20	2.000
1024	1.882e−13	4.985	2.532e−13	4.989	1.448e−13	4.981	6.776e−21	0.170	0	–	3.388e−21	3.000	6.776e−21	2.000
2048	6.002e−15	4.971	8.035e−15	4.978	4.646e−15	4.962	3.229e−21	1.069	1.271e−21	–	8.470e−22	2.000	1.694e−21	2.000

The algorithm presents full accuracy for corner singularities

Table 2 Grid refinement analysis for the new algorithm applied to the function in (55) in the point-value setting

i	$x = 1/2 - 3h/2$		$x = 1/2 - h$		$x = 1/2 - h/2$		$x = 1/2$		$x = 1/2 + h/2$		$x = 1/2 + h$		$x = 1/2 + 3h/2$	
	e_i	$Order$	e_i	$Order$	e_i	$Order$	e_i	$Order$	e_i	$Order$	e_i	$Order$	e_i	$Order$
32	5.311e−06	–	3.240e−06	–	4.689e−06	–	6.722e−18	–	3.469e−18	–	0	–	1.388e−17	–
64	1.695e−07	4.969	1.022e−07	4.987	1.477e−07	4.989	1.735e−18	1.954	0	–	8.674e−19	–	0	–
128	5.329e−09	4.991	3.205e−09	4.995	4.632e−09	4.995	5.963e−19	1.541	1.084e−19	–	2.168e−19	2.000	0	–
256	1.671e−10	4.996	1.005e−10	4.995	1.453e−10	4.994	2.236e−19	1.415	2.711e−20	2.000	0	–	0	–
512	5.241e−12	4.994	3.160e−12	4.991	4.572e−12	4.990	3.473e−20	2.687	0	–	0	–	0	–
1024	1.649e−13	4.990	9.990e−14	4.983	1.448e−13	4.981	5.082e−20	−0.549	3.388e−21	–	3.388e−21	–	6.776e−21	–
2048	5.227e−15	4.980	3.194e−15	4.967	4.646e−15	4.962	4.394e−21	3.532	8.470e−22	2.000	0	–	0	–

The algorithm presents full accuracy for corner singularities

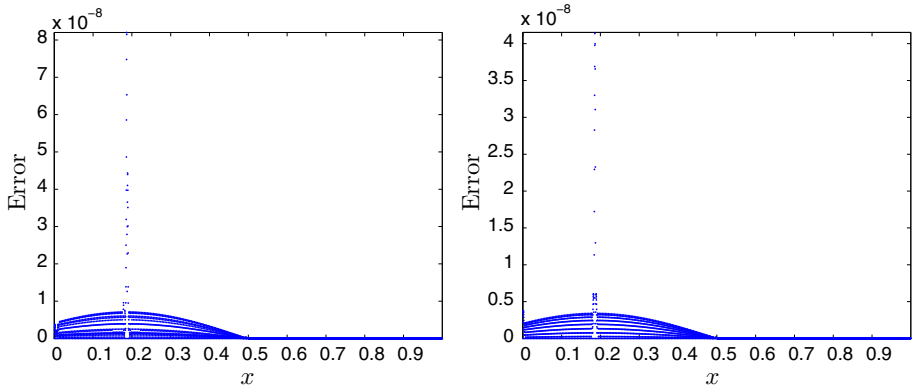

Fig. 8 Representation of the absolute error obtained for the function in (55) using ENO-SR (left) and the new algorithm proposed (right) with 4 scales of multiresolution and truncating all the details. The finest grid is $n = 4096$ points and the coarsest grid is 256 points

Table 3 Norms of the error obtained for the function in (55) using ENO-SR and the new algorithm proposed with 4 scales of multiresolution and truncating all the details

ξ	l_1	l_2	l_∞
ENO-SR	1.4485e−09	1.6544e−17	8.2021e−08
New algorithm	7.7406e−10	4.8693e−18	4.152e−08

The finest grid is $n = 4096$ points and the coarsest grid is 256 points

3 show that the error obtained by the new algorithm is smaller than the one obtained by ENO-SR in all the norms studied.

4.2 Numerical Estimation of the Stability Constant in the Point-Values

The main problem that ENO scheme presents is that, due to the stencil selection procedure and the extrapolation, it is known to introduce numerical instabilities in certain applications [30]. The same problems affect ENO-SR scheme [8]. In [30] the authors use an error-control strategy in order to stabilize ENO algorithm. It is also known that WENO algorithm [31] is stable [8].

In this section we obtain and compare estimations of the stability constant for ENO-SR algorithm, for WENO algorithm and for the new algorithm proposed. Our objective is to show that the new algorithm obtains stability constants that are similar to those obtained by WENO algorithm. In order to do so, we will compare the results obtained by ENO-SR and the new algorithm with those obtained by WENO.

In order to estimate numerically the stability constant for the l_1 norm, we consider the following setup: given a discrete sequence $f^L = (f_j^L)$, we descend in the multiresolution pyramid obtaining its multiresolution representation $Mf^L = \{f^0, d^1, \ldots, d^L\}$ ($L = 4$ in our numerical test). We truncate the detail coefficients of this representation which are larger than a certain tolerance parameter ξ obtaining the perturbed representation $\{\hat{f}^0, \hat{d}^1, \ldots, \hat{d}^L\}$. We then measure the size of the perturbation by

Table 4 Estimations of the stability constant for ENO-SR, the new algorithm and WENO for different values of the truncation parameter ξ, with $L = 4$ multiresolution levels and for the function presented in (55)

ξ	1e−16	1e−15	1e−14	1e−13	1e−12	1e−11	1e−10	1e−9	1e−8	1e−7
ENO-SR										
Stability C.	0.4732	0.2937	0.3220	0.4778	0.5008	0.5065	0.5190	0.5625	0.5733	0.5796
No. details	2324	1920	1916	1890	1512	795	356	120	2	0
l_1	6.6722e−18	2.5266e−17	3.2120e−17	3.8191e−16	6.0167e−14	5.2760e−13	5.8786e−12	8.6631e−11	1.2147e−09	1.4082e−09
l_∞	3.8858e−16	1.3323e−15	9.3901e−15	9.7394e−14	1.0036e−12	1.1459e−11	1.2295e−10	1.1950e−09	1.1212e−08	8.2728e−08
New alg.										
Stability C.	0.4340	0.2801	0.3259	0.4916	0.5000	0.5010	0.5007	0.5030	0.5023	0.5224
No. details	2174	1920	1915	1870	884	675	335	114	1	0
l_1	4.6058e−18	1.5394e−17	2.2625e−17	6.6260e−16	1.5313e−13	8.2165e−13	5.0602e−12	5.6285e−11	6.6631e−10	7.7406e−10
l_∞	3.3307e−16	8.8818e−16	9.2929e−15	9.9901e−14	9.4479e−13	9.9996e−12	9.9108e−11	9.5357e−10	5.6809e−09	4.1520e−08
WENO										
Stability C.	0.4127	0.3729	0.4779	0.4983	0.5065	0.5065	0.5065	0.5065	0.5065	0.5065
No. details	1549	377	147	129	20	20	20	20	20	20
l_1	9.1546e−18	7.0630e−17	8.0888e−16	2.7015e−15	5.2617e−14	5.2617e−14	5.2617e−14	5.2617e−14	5.2617e−14	5.2617e−14
l_∞	2.2204e−16	4.8850e−15	4.8850e−15	9.7977e−14	8.2068e−13	8.2068e−13	8.2068e−13	8.2068e−13	8.2068e−13	8.2068e−13

The results obtained by WENO algorithm are given only as a reference. The number of details kept, the l_1 and the l_∞ norms of the error have also been included in the table

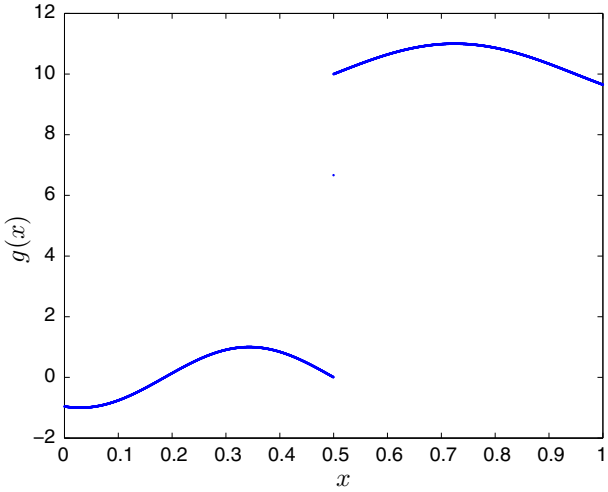

Fig. 9 Representation of the function in (59)

$$\mu = ||f^0 - \hat{f}^0||_1 + \sum_{k=1}^{L} ||d^k - \hat{d}^k||_1. \tag{56}$$

Next, we use the decoding algorithm to obtain an approximation to the original discrete sequence given by $\hat{f}^L = M^{-1}\{\hat{f}^0, \hat{d}^1, \ldots, \hat{d}^L\}$. We measure the error committed as

$$E = ||f^L - \hat{f}^L||_1. \tag{57}$$

A numerical estimation of the stability constant is provided by the ratio

$$C_s = \frac{E}{\mu}. \tag{58}$$

In order to obtain the results displayed in Table 4, we have considered the function presented in Fig. 7 and introduced in (55). We have presented the numerical stability constant, the number of non-zero details, the l_1 norm of the error and the l_∞ norm of the error for different values of the truncation parameter ξ. The size of the sampled function is 4096 points and we use $L = 4$ levels of multiresolution. The range of the truncation parameter ξ has been selected such that the number of coefficients is very high for the first column of the table and zero for the last one. This means that if we increase ξ once we have arrived to the last element shown in the table, the stability constant will always take the value of this last element. We can see that the three algorithms presented in the table obtain similar stability constants for the function in (55). The results can not be very different for ENO-SR and the new algorithm, as the detection part is the same for both schemes and the order of accuracy is the same in both cases. Even though, we can see that for a smaller number of details, the new algorithm attains smaller norms of the error. We have included the results obtained by WENO algorithm as a reference of the size of the stability constants obtained by a stable algorithm for the function presented in (55). It is important to take into account that we have presented the results for WENO with a stencil of 6 points, so the maximum accuracy obtained is $O(h^6)$ while the maximum accuracy obtained by ENO-SR or the new algorithm with a stencil of 4 points is $O(h^4)$. Also, WENO is not capable of detecting the discontinuity and approximating with

Table 5 Grid refinement analysis for ENO-SR for the function in (59) in the cell-average setting

n	$x = 1/2 - 3h/2$		$x = 1/2 - h$		$x = 1/2 - h/2$		$x = 1/2$		$x = 1/2 + h/2$		$x = 1/2 + h$		$x = 1/2 + 3h/2$	
	e_i	$Order$	e_i	$Order$	e_i	$Order$	e_i	$Order$	e_i	$Order$	e_i	$Order$	e_i	$Order$
32	5.300e−04	–	1.212e−03	–	1.776e−03	–	5.524e−03	–	7.814e−04	–	7.134e−04	–	6.113e−04	–
64	2.326e−04	1.189	2.585e−04	2.229	2.782e−04	2.675	7.093e−04	2.961	1.011e−04	2.951	9.890e−05	2.851	9.555e−05	2.678
128	3.514e−05	2.726	3.599e−05	2.845	3.663e−05	2.925	8.925e−05	2.990	1.274e−05	2.988	1.267e−05	2.964	1.257e−05	2.926
256	4.590e−06	2.937	4.617e−06	2.963	4.637e−06	2.982	1.117e−05	2.998	1.596e−06	2.997	1.594e−06	2.991	1.591e−06	2.982
512	5.800e−07	2.984	5.808e−07	2.991	5.814e−07	2.995	1.397e−06	2.999	1.996e−07	2.999	1.996e−07	2.998	1.995e−07	2.996
1024	7.269e−08	2.996	7.272e−08	2.998	7.274e−08	2.999	1.747e−07	3.000	2.496e−08	3.000	2.495e−08	2.999	2.495e−08	2.999
2048	9.093e−09	2.999	9.094e−09	2.999	9.094e−09	3.000	2.184e−08	3.000	3.120e−09	3.000	3.120e−09	3.000	3.119e−09	3.000

The algorithm presents full accuracy for jump discontinuities

Table 6 Grid refinement analysis for the new algorithm for the function in (59) in the cell-average setting

n	$x = 1/2 - 3h/2$		$x = 1/2 - h$		$x = 1/2 - h/2$		$x = 1/2$		$x = 1/2 + h/2$		$x = 1/2 + h$		$x = 1/2 + 3h/2$	
	e_i	$Order$	e_i	$Order$	e_i	$Order$	e_i	$Order$	e_i	$Order$	e_i	$Order$	e_i	$Order$
32	6.505e−04	–	1.007e−03	–	1.776e−03	–	5.524e−03	–	7.814e−04	–	4.627e−04	–	4.176e−04	–
64	1.523e−04	2.094	1.650e−04	2.609	2.782e−04	2.675	7.093e−04	2.961	1.011e−04	2.951	6.045e−05	2.936	5.901e−05	2.823
128	2.151e−05	2.825	2.191e−05	2.912	3.663e−05	2.925	8.925e−05	2.990	1.274e−05	2.988	7.639e−06	2.984	7.594e−06	2.958
256	2.767e−06	2.958	2.780e−06	2.979	4.637e−06	2.982	1.117e−05	2.998	1.596e−06	2.997	9.575e−07	2.996	9.561e−07	2.990
512	3.484e−07	2.990	3.488e−07	2.995	5.814e−07	2.995	1.397e−06	2.999	1.996e−07	2.999	1.198e−07	2.999	1.197e−07	2.997
1024	4.363e−08	2.997	4.364e−08	2.999	7.274e−08	2.999	1.747e−07	3.000	2.496e−08	3.000	1.497e−08	3.000	1.497e−08	2.999
2048	5.456e−09	2.999	5.456e−09	3.000	9.094e−09	3.000	2.184e−08	3.000	3.120e−09	3.000	1.872e−09	3.000	1.872e−09	3.000

The algorithm presents full accuracy for jump discontinuities

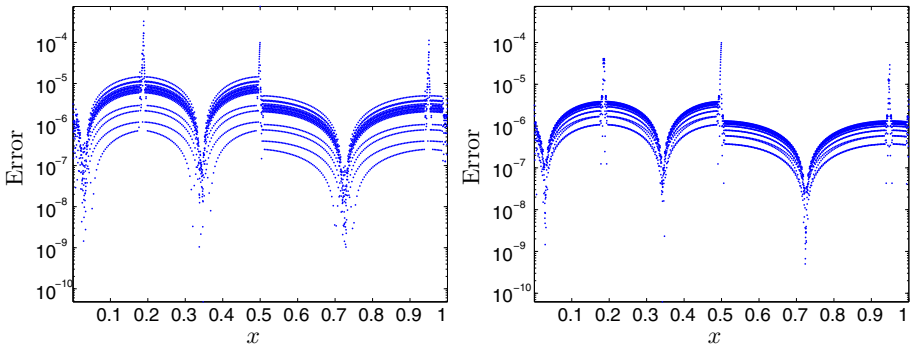

Fig. 10 Semilogarithmic representation of the absolute error obtained for the function in (59) using ENO-SR (left) and the new algorithm proposed (right) with 4 scales of multiresolution and truncating all the details. The finest grid is $n = 4096$ cells and the coarsest grid is 256 cells. We can see how the new method attains a smaller error in the whole domain

high accuracy close to discontinuities. These two reasons explain the difference in the number of coefficients at the beginning and the end of the table. Among the ENO-SR and the new algorithm, the new algorithm attains a higher compression rate (less number o details for a determined truncation parameter ξ) while reducing the norms of the errors.

4.3 Experiments in the Cell-Averages

In this section we have discretized the function in (59) using the expression in (2). Then we decimate the discretized function using (3) and we reconstruct it from the lower resolution to the highest resolution. In order to obtain the reconstruction, we can work using the primitive function (5) or directly in the cell-averages. The function has been plotted in Fig. 9. The domain is the same as in previous section, $x \in [0, 1]$.

Example 2 Let's consider the function in (59)

$$g(x) = \begin{cases} -\sin(10(x - 0.5 - \alpha)) & \text{if } 0 < x \leq \alpha, \\ \sin(7(x - 0.5 - \alpha)) + 10 & \text{if } \alpha < x \leq 1, \end{cases} \tag{59}$$

with $\alpha = \frac{h_0}{3}$, being $h_0 = \frac{1}{2^{12}}$. Setting $h = \frac{1}{2^i}, i = 5, 6, 7, 8, \ldots, 11$, and using the discretization operator in (2) we check the accuracy of the interpolation through a grid refinement analysis. Tables 5 and 6 show the results obtained for the order of accuracy at $x \in \{\frac{1}{2} - \frac{3h}{2}, \frac{1}{2} - h, \frac{1}{2} - \frac{h}{2}, \frac{1}{2}, \frac{1}{2} + \frac{h}{2}, \frac{1}{2} + h, \frac{1}{2} + \frac{3h}{2}\}$, for ENO-SR and the new algorithm respectively. We can see how both algorithms attain full accuracy even in the presence of a jump discontinuity. At the discontinuity, the error obtained by both algorithms is similar. Even though, away form the discontinuity, the error obtained by the new algorithm is smaller. This piece of evidence will be confirmed in the next experiment.

In Fig. 10 we show the results of a subdivision experiment similar to the one we performed for the point-values. We have set four scales of multiresolution and we have truncated all the details for the function in (59). The original grid has 4096 cells. Figure 10 to the left shows the absolute error of the ENO-SR reconstruction around the discontinuity using a semilogarithmic scale in order to better appreciate the error. Figure 10 to the right shows the result obtained for the same experiment but using the new algorithm proposed. We can see

Table 7 Norms of the error obtained for the function in (59) using ENO-SR and the new algorithm proposed in the cell-averages with 4 scales of multiresolution and truncating all the details

ξ	l_1	l_2	l_∞
ENO-SR	3.7883e−06	2.5169e−10	7.5165e−4
New algorithm	1.713e−06	1.4913e−10	7.2951e−4

The finest grid is $n = 4096$ cells and the coarsest grid is 256 cells

that the results obtained by both algorithms close to the discontinuity are similar. Moreover, the error obtained by the new algorithm away from the discontinuity is better than the one obtained by ENO-SR. Table 7 shows the norm of the errors obtained for this experiment. The result obtained by the new algorithm is better in the l_1 and l_2 norms. In the l_∞ norm the results are similar, as can be observed in Fig. 10.

5 Conclusions

In this article we have presented a new centered algorithm for function approximation and multiresolution in the point-value and the cell-average setting. The new algorithm attains full accuracy and does not use a stencil selection procedure nor extrapolation, techniques that are known to introduce instability in certain applications. What is more, the stability constants obtained for the new algorithm are very similar to those obtained by WENO algorithm, that is known to be stable. The new algorithm is based on the correction of the derivatives of Newton polynomials at stencils that cross a discontinuity. The correction terms are quite simple and have closed expressions, so they can be directly added to the prediction at the central point of the stencil whenever a discontinuity is detected. The numerical experiments have shown that the new algorithm is capable of improving the accuracy results attained by ENO-SR at smooth zones, that is already a remarkable aspect. A proof for the full accuracy of the algorithm has been given. The numerical experiments show that the numerical accuracy matches the theoretical accuracy obtained for the point-values and the cell-averages.

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

References

1. de Boor, C.: A Practical Guide to Splines, vol. 27. Springer, New York (1980)
2. Harten, A., Osher, S.: Uniformly high-order accurate nonoscillatory schemes. I. SIAM J. Numer. Anal. **24**(2), 279–309 (1987)
3. Harten, A., Engquist, B., Osher, S., Chakravarthy, S.R.: Uniformly high order accurate essentially non-oscillatory schemes. III. J. Comput. Phys. **71**(2), 231–303 (1987)
4. Harten, A.: Multiresolution representation of data II. SIAM J. Numer. Anal. **33**(3), 1205–1256 (1996)
5. Harten, A.: Multiresolution representation of data: a general framework. SIAM J. Numer. Anal. **33**(3), 1205–1256 (1996)
6. Amat, S., Aràndiga, F., Cohen, A., Donat, R., Garcia, G., von Oehsen, M.: Data compression with ENO schemes: a case study. Appl. Comput. Harmon. Anal. **11**(2), 273–288 (2001)
7. Serna, S., Marquina, A.: Power ENO methods: a fifth-order accurate weighted power ENO method. J. Comput. Phys. **194**(2), 632–658 (2004)

8. Cohen, A., Dyn, N., Matei, B.: Quasi linear subdivision schemes with applications to ENO interpolation. Appl. Comput. Harmon. Anal. **15**, 89–116 (2003)
9. Amat, S., Busquier, S., Trillo, J.C.: On multiresolution schemes using a stencil selection procedure: applications to ENO schemes. Numer. Algorithms **44**(1), 45–68 (2007)
10. Harten, A.: ENO schemes with subcell resolution. J. Comput. Phys. **83**(1), 148–184 (1989)
11. Aràndiga, F., Donat, R., Mulet, P.: Adaptive interpolation of images. Signal Process. **83**(2), 459–464 (2003)
12. Aràndiga, F., Cohen, A., Donat, R., Dyn, N.: Interpolation and approximation of piecewise smooth functions. SIAM J. Numer. Anal. **43**(1), 41–57 (2005)
13. Amat, S., Dadourian, K., Liandrat, J.: On a nonlinear subdivision scheme avoiding Gibbs oscillations and converging towards c^s functions with $s > 1$. Math. Comput. **80**(80), 959–971 (2011)
14. Amat, S., Liandrat, J.: On the stability of the PPH nonlinear multiresolution. Appl. Comput. Harm. Anal. **18**(2), 198–206 (2005)
15. Amat, S., Liandrat, J., Ruiz, J., Trillo, J.: On a compact non-extrapolating scheme for adaptive image interpolation. J. Frankl. Inst. **349**(5), 1637–1647 (2012)
16. Amat, S., Dadourian, K., Liandrat, J., Ruiz, J., Trillo, J.C.: On a class of L^1-stable nonlinear cell-average multiresolution schemes. J. Comput. Appl. Math. **234**(4), 1129–1139 (2010)
17. Amat, S., Liandrat, J., Ruiz, J., Trillo, J.: On a nonlinear cell-average multiresolution scheme for image compression. SeMA J. **1**(60), 75–92 (2012)
18. Amat, S., Ruiz, J., Trillo, J.C.: Adaptive interpolation of images using a new nonlinear cell-average scheme. Math. Comput. Simul. **82**(9), 1586–1596 (2012)
19. Amat, S., Dadourian, K., Liandrat, J.: Analysis of a class of nonlinear subdivision schemes and associated multiresolution transforms. Adv. Comput. Math. **34**(3), 253–277 (2011)
20. Amat, S., Donat, R., Liandrat, J., Trillo, J.: Analysis of a new nonlinear subdivision scheme. Applications in image processing. Found. Comput. Math. **6**(2), 193–225 (2006)
21. Amat, S., Ruiz, J., Trillo, J.C.: Improving the compression rate versus L^1 error ratio in cell-average error control algorithms. Numer. Algorithms **67**(1), 145–162 (2014)
22. Aràndiga, F., Donat, R.: Nonlinear multiscale decompositions: the approach of A. Harten. Numer. Algorithms **23**(2–3), 175–216 (2000)
23. Aràndiga, F., Belda, A., Mulet, P.: Point-value WENO multiresolution applications to stable image compression. J. Sci. Comput. **43**(2), 158–182 (2010)
24. Amat, S., Dadourian, K., Liandrat, J.: On a nonlinear 4-point ternary and interpolatory multiresolution scheme eliminating the Gibbs phenomenom. Int. J. Numer. Anal. Model. **2**(7), 261–280 (2010)
25. Peskin, C.S.: The immersed boundary method. Acta Numer. **11**, 479–517 (2002)
26. Leveque, R.J., Li, Z.: The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. SIAM J. Numer. Anal. **31**(4), 1019–1044 (1994)
27. Li, Z., Ito, K.: The Immersed Interface Method: Numerical Solutions of PDEs Involving Interfaces and Irregular Domains (Frontiers in Applied Mathematics). SIAM, Philadelphia (2006)
28. Li, Z., Lai, M.-C.: The immersed interface method for the Navier–Stokes equations with singular forces. J. Comput. Phys. **1**(171), 822–842 (2001)
29. Gonzalez, R., Woods, R.: Digital Image Processing. Prentice-Hall, Inc., Upper Saddle River (2002)
30. Amat, S., Aràndiga, F., Cohen, A., Donat, R.: Tensor product multiresolution analysis with error control for compact image representation. Signal Process. **82**(4), 587–608 (2002)
31. Liu, X.-D., Osher, S., Chan, T.: Weighted essentially non-oscillatory schemes. J. Comput. Phys. **115**(1), 200–212 (1994)