



# Fully accurate approximation of piecewise smooth functions using corrected B-spline quasi-interpolants

Zhilin Li<sup>1</sup> · Kejia Pan<sup>2</sup> · Juan Ruiz<sup>3</sup> · Dionisio F. Yáñez<sup>4</sup>

Received: 18 August 2023 / Revised: 21 February 2024 / Accepted: 22 February 2024  
© The Author(s) 2024

## Abstract

This article is devoted to introducing new spline quasi-interpolants for the sharp approximation of data in one and several dimensions. The new construction aims to obtain accurate approximations close to singularities of the function from which the data are obtained. The technique relies in an accurate knowledge of the position of the singularity, which can be known or approximated, and that allows for the obtention of accurate approximations of the jumps of the function and its derivatives at this point. With this information, it is possible to compute correction terms for the B-spline bases of the spline that are affected by the presence of singularities. The result is a piecewise smooth reconstruction with full accuracy (meaning by *full accuracy*, the accuracy of the spline at smooth zones). The numerical experiments presented support the theoretical results obtained.

**Keywords** IIM · B-spline · Correction terms · Improved adaption to discontinuities · Signal processing

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

---

✉ Juan Ruiz  
juan.ruiz@upct.es

Zhilin Li  
zhilin@math.ncsu.edu

Kejia Pan  
kejiapan@csu.edu.cn

Dionisio F. Yáñez  
dionisio.yanez@uv.es

<sup>1</sup> Department of Mathematics, North Carolina State University, Raleigh, USA

<sup>2</sup> School of Mathematics and Statistics, Central South University, Changsha, China

<sup>3</sup> Department of Applied Mathematics and Statistics, Universidad Politécnica de Cartagena, Cartagena, Spain

<sup>4</sup> Department of Mathematics, Universidad de Valencia, Valencia, Spain

## 1 Introduction

Approximation and interpolation methods are powerful tools that are currently used in several real applications. Some examples are the design of curves and surfaces, geolocation, image processing, applications in medicine, and many others. Specifically, interpolants based on splines or approximants constructed using B-splines have been used for image compression (see, e.g. Unser 1999; Forster 2011), computer-aided geometric design (see, e.g. Boehm 1986; Sarfraz 1998), generation of curves (see e.g. Zheng et al. 2005) or real applications such as ship hull design (see e.g. Rogers and Satterfield 1980; Nowacki 2010; Katsoulis et al. 2019) or medical image processing (see, e.g. Lehmann et al. 2001).

The term quasi-interpolation is used to denote the construction of accurate approximations of a particular set of data obtained from a given function. As mentioned in Lyche et al. (2018), a quasi-interpolant to  $f$  provides a “reasonable” approximation to  $f$ . For example, interpolation and least squares are considered quasi-interpolation techniques. It is always desirable that the computational cost to obtain these approximations is low. Usually, quasi-interpolation techniques are obtained through the linear combination of blending functions with compact support. These blending functions are usually selected so that they are a convex partition of unity. It is also considered advantageous that the coefficients of the linear combination only involve a local stencil of the original data, in order to maintain the locality of the approximation obtained. The objective is to attain local control of the approximation and also to assure numerical stability. Quasi-interpolants constructed using polynomial spline spaces are widely known and used and are considered a powerful tool for the approximation of data that can be extended easily to several dimensions using tensor product (de Boor 1990; de Boor and Fix 1973; Speleers 2017; Lyche and Schumaker 1975; Sablonnière 2005). The main characteristic of the methods based on B-splines is the smoothness of the resultant function when the data do not present any strong gradient or discontinuity. However, they produce non-desirable effects around the singularities. Recently, some papers have dealt with this problem (see Amat et al. 2023; Aràndiga et al. 2023).

In particular, we consider a uniform sampling of the function  $f \in C^{p+1}(\mathbb{R})$  and denote the nodes of the mesh by  $x_n = nh$ ,  $n \in \mathbb{Z}$ . The values of the function over the mesh are  $\{f_n = f(nh)\}$ . With this information, we can compute approximations of  $f$  with  $O(h^{p+1})$  order of accuracy using local combinations of B-spline bases  $B_p$  of degree  $p$  (Lyche et al. 2018; Speleers 2017), with equidistant nodes  $S_p = \left\{-\frac{p+1}{2}, \dots, \frac{p+1}{2}\right\}$  and support  $I_p = \left[-\frac{p+1}{2}, \frac{p+1}{2}\right]$ , by means of the operator

$$Q_p(f)(x) = \sum_{n \in \mathbb{Z}} L_p(f_{n,p}) B_p\left(\frac{x}{h} - n\right), \quad (1)$$

where the linear operator  $L_p : \mathbb{R}^{2\lfloor \frac{p}{2} \rfloor + 1} \rightarrow \mathbb{R}$  is defined following Speleers (2017) as

$$L_p(f_{n,p}) = \sum_{j=-\lfloor \frac{p}{2} \rfloor}^{\lfloor \frac{p}{2} \rfloor} c_j^p f_{n+j}, \quad (2)$$

where  $c_j^p, j = -\lfloor \frac{p}{2} \rfloor, \dots, \lfloor \frac{p}{2} \rfloor$  can be written as

$$c_j^p = \sum_{l=0}^{\lfloor \frac{p+1}{2} \rfloor - 1} \frac{t(2l + p + 1, p + 1)}{\binom{2l+p+1}{p+1}} \sum_{i=0}^{2l} \frac{(-1)^i}{i!(2l-i)!} \delta_{l-i+\lfloor \frac{p+1}{2} \rfloor, j+1+\lfloor \frac{p}{2} \rfloor}, \tag{3}$$

with  $\delta_{i,j}$  being the Kronecker delta function:

$$\delta_{i,j} = \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{if } i \neq j. \end{cases}$$

The  $t(i, j)$  are the central factorial numbers of the first kind (Speleers 2017; Butzer et al. 1989). A recursive computation of these numbers can be done using the following expressions:

$$t(i, j) = \begin{cases} 0, & \text{if } j > i, \\ 1, & \text{if } j = i, \\ t(i - 2, j - 2) - \left(\frac{i-2}{2}\right)^2 t(i - 2, j), & \text{if } 2 \leq j < i, \end{cases}$$

with

$$t(i, 0) = 0, \quad t(i, 1) = \prod_{l=1}^{i-1} \left(\frac{i}{2} - l\right), \quad i \geq 2,$$

where  $t(0, 0) = t(1, 1) = 1, t(0, 1) = t(1, 0) = 0$ .

In this work, we will centre our attention in cases  $p = 2, 3$ , although the same technique can be used for greater  $p$ . Thus, we particularise the quasi-interpolation operator  $Q_p$  with the form of (1) for  $p = 2, 3$ :

$$Q_p(f)(x) = \sum_{n \in \mathbb{Z}} L_p(f_{n,3}) B_p\left(\frac{x}{h} - n\right), \quad n \in \mathbb{Z}, \tag{4}$$

where

$$L_p(f_{n-1}, f_n, f_{n+1}) = \sum_{j=-1}^1 c_j^p f_{n+j},$$

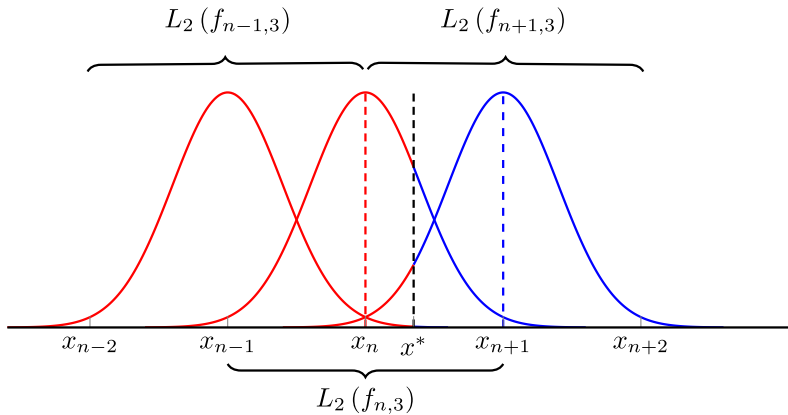
$$\text{with } c_{-1}^2 = c_1^2 = -\frac{1}{8}, \quad c_0^2 = \frac{5}{4}; \quad c_{-1}^3 = c_1^3 = -\frac{1}{6}, \quad c_0^3 = \frac{4}{3}. \tag{5}$$

It is known that this quasi-interpolation operator reproduces the space of polynomials of degree less or equal than  $p$  on finite intervals  $I$ , Lyche et al. (2018), such that

$$\|f - Q_p(f)\|_{\infty, I} = O(h^{p+1}), \tag{6}$$

as  $h \rightarrow 0$ . It is also known that, close to jump discontinuities, the operator losses the accuracy due to the presence of oscillations, and close to kinks (discontinuities in the first derivative), due to the smearing of the singularity.

The aim of this work is to design correction terms for the linear operator  $Q_p$ , in order to reconstruct with full accuracy piecewise smooth functions that present singularities at one or more points. With this objective in mind, we will take inspiration from the immersed interface method (IIM) (see, e.g. Leveque and Li 1994; Li and Ito 2006), which was originally proposed for the solution of elliptic partial differential equations with discontinuities. Correction terms



**Fig. 1** An example of a quadratic B-spline base and the division of the domain in two subdomains, which depend on the location of the singularity  $x^*$

for quadratic and cubic splines will be presented in Sects. 2 and 3, where we also analyse some properties of the construction. Section 5 is dedicated to present some numerical experiments that support the theoretical results obtained.

## 2 Correction of the quadratic B-spline quasi-interpolant

We start by assuming that the position  $x^* \in [x_{n-\frac{1}{2}}, x_{n+\frac{1}{2}}]$  of an isolated singularity is known or can be approximated with enough accuracy. It is known that, for data given in the point values, i.e. samplings of a function  $f$  over an interval, the location of a jump discontinuity in the function is lost during the discretization process, but kinks (discontinuities in the first-order derivative) can still be located. For data given in the cell-average setting, i.e. the data are obtained from the integration of a function in certain intervals, jumps in the function and in the first-order derivative can be located. In this article, we will centre our attention on the point value setting, as the extension to the cell-average setting is straightforward considering the primitive function (Aràndiga and Donat 2000).

The obtention of correction terms can be done expanding in a smart way some of the values of  $f_n$  contained in the operator  $L_p$  using Taylor expansions. Of course, after finishing this process, we need to assure that the B-splines that compose the base, do not cross the singularity at  $x^*$ . This means that we need to divide the domain in subdomains limited by the location of the singularity or singularities. An example of a quadratic B-spline base and the division of the domain in two subdomains can be seen in Fig. 1. We have represented the subdomain to the left of  $x^*$  in red, and the subdomain to the right in blue. In this figure we have also presented the operator  $L_p(f_{n,3})$  for  $p = 2$ . We can see how modifying the values of this operator, which cross the singularity, is not enough to attain full accuracy, as the B-spline (that is multiplied by this operator) might still cross the singularity.

Let us now explain how to proceed to obtain the correction terms that we propose. Let  $l$  be a natural number,  $f^- \in C^l([a, x^*])$ ,  $f^+ \in C^l([x^*, b])$ , where we have used the superindex  $-$  to express the function values at the left of the singularity, and the  $+$  superindex to express the function values to the right of the singularity, with

$$f(x) = \begin{cases} f^-(x), & x \leq x^*, \\ f^+(x), & x > x^*, \end{cases} \tag{7}$$

and writing the jumps in the derivatives of  $f$  as

$$[f^{(l)}] = \underbrace{f_x^+ \dots x(x^*)}_l - \underbrace{f_x^- \dots x(x^*)}_l, \tag{8}$$

where we have denoted the  $l$ th derivative of  $f^\pm$  at  $x^*$  by  $f_x^\pm \dots x(x^*)$ . Let us discuss later about how to approximate these jumps, and which accuracy is needed.

Looking at Fig. 2 and at the expression of the operator  $L_p$  in (5), it should be clear that the approximation error of the spline in the interval  $[x_{n-\frac{1}{2}}, x_{n+\frac{1}{2}}]$  that contains the singularity, is due to the fact that the operator  $L_2(f_{n,3})$  and  $L_2(f_{n+1,3})$  are crossing the discontinuity. Moreover, it is also due to the fact that the B-spline bases are also crossing the discontinuity. Thus, now it should be clear that, not only it is necessary to correct the operator, but also to divide the domain so that information from only one side of the discontinuity is used in the computation. In order to compute the amount of error that each B-spline function contributes to the total error of the approximation in the interval  $[x_{n-\frac{1}{2}}, x_{n+\frac{1}{2}}]$ , we can just use Taylor expansions on the expression of the  $L_p$  operator. For example, we can see in Fig. 2 that  $L_2(f_{n,3})$  crosses the singularity, so we can use the Taylor expansions of the value  $f_j$  around  $x^*$ , and then use the jump relations  $[f], [f'], [f'']$  in (8) to express the values to the left of the singularity in terms of the values to its right (or viceversa).

If we assume that we know the jumps in (8), or accurate enough approximations, using Taylor expansions, we can write

$$\begin{aligned} f^-(x_n) &= f_n^- = f^-(x^*) - f_x^-(x^*)\alpha + \frac{1}{2}f_{xx}^-(x^*)\alpha^2 - \frac{1}{3!}f_{xxx}^-(x^*)\alpha^3 + O(h^4), \\ f^+(x_n) &= f_n^+ = f^+(x^*) - f_x^+(x^*)\alpha + \frac{1}{2}f_{xx}^+(x^*)\alpha^2 - \frac{1}{3!}f_{xxx}^+(x^*)\alpha^3 + O(h^4), \\ f^-(x_{n+1}) &= f_{n+1}^- = f^-(x^*) + f_x^-(x^*)\beta + \frac{1}{2}f_{xx}^-(x^*)\beta^2 + \frac{1}{3!}f_{xxx}^-(x^*)\beta^3 + O(h^4), \\ f^+(x_{n+1}) &= f_{n+1}^+ = f^+(x^*) + f_x^+(x^*)\beta + \frac{1}{2}f_{xx}^+(x^*)\beta^2 + \frac{1}{3!}f_{xxx}^+(x^*)\beta^3 + O(h^4), \end{aligned} \tag{9}$$

where  $\alpha = x^* - x_n$  and  $\beta = x_{n+1} - x^*$ . Subtracting, we obtain

$$\begin{aligned} f_n^+ &= f_n^- + [f] - [f']\alpha + \frac{1}{2}[f'']\alpha^2 - \frac{1}{3!}[f''']\alpha^3 + O(h^4), \\ f_{n+1}^+ &= f_{n+1}^- + [f] + [f']\beta + \frac{1}{2}[f'']\beta^2 + \frac{1}{3!}[f''']\beta^3 + O(h^4). \end{aligned} \tag{10}$$

Using the same process, similar expressions can be obtained for the values  $f_{n-2}, f_{n-1}$ , and  $f_{n+2}$ , which are involved in the computation of the approximation in the interval  $[x_{n-\frac{1}{2}}, x_{n+\frac{1}{2}}]$ , as presented in Fig. 1.

For the quadratic spline, in order to analyse the local truncation error introduced by each B-spline, it is much simpler to consider one of the elements  $L_p(f_{n,3}) \cdot B_p(\frac{\cdot}{h} - n)$  of the quasi-interpolant in (4) and to determine in which part of the support of the B-spline the discontinuity falls. Considering the B-spline  $B_2(\frac{\cdot}{h} - n)$  represented in Fig. 1 (the central one), we are ready to state the following Lemma, which will provide the accuracy attained by the quadratic quasi-interpolant when the data are affected by a discontinuity.

**Lemma 1** *Let consider the following partition:*

$$\begin{aligned}
 [x_{n-\frac{3}{2}}, x_{n+\frac{3}{2}}] &= [x_{n-\frac{3}{2}}, x_{n-1}) \cup [x_{n-1}, x_n) \cup [x_n, x_{n+1}) \cup [x_{n+1}, x_{n+\frac{3}{2}}] \\
 &=: I_n^{-1} \cup I_n^0 \cup I_n^1 \cup I_n^2,
 \end{aligned}$$

and a singularity  $x^* \in I_n^j$ ,  $j = -1, 0, 1, 2$  then

$$L_2(f_{n,3}) - L_2(f_{n,3}^\pm) = C_j^\pm(f_{n,3}) + O(h^3). \tag{11}$$

being for  $j = -1$

$$\begin{cases} C_{-1}^-(f_{n,3}) = [f] + [f'](\beta + h) + [f'']\left(\frac{1}{2}\beta^2 + \beta h + \frac{3}{8}h^2\right), & \text{if } x < x^*, \\ C_{-1}^+(f_{n,3}) = 0, & \text{if } x > x^*, \end{cases} \tag{12}$$

for  $j = 0$

$$\begin{cases} C_0^-(f_{n,3}) = \frac{9}{8}[f] + \frac{1}{8}[f'](9\beta - h) + \frac{1}{8}[f''](5\beta^2 - \frac{1}{2}(\beta + h)^2), & \text{if } x < x^*, \\ C_0^+(f_{n,3}) = \frac{1}{8}[f] - \frac{1}{8}[f']\alpha + \frac{1}{16}[f'']\alpha^2, & \text{if } x > x^*, \end{cases} \tag{13}$$

for  $j = 1$

$$\begin{cases} C_1^-(f_{n,3}) = -\frac{1}{8}[f] - \frac{1}{8}[f']\beta - \frac{1}{16}[f'']\beta^2, & \text{if } x < x^*, \\ C_1^+(f_{n,3}) = -\frac{9}{8}[f] + \frac{1}{8}[f'](9\alpha - h) - \frac{1}{8}[f''](5\alpha^2 - \frac{1}{2}(\alpha + h)^2), & \text{if } x > x^*, \end{cases} \tag{14}$$

and for  $j = 2$

$$\begin{cases} C_2^-(f_{n,3}) = 0, & \text{if } x < x^*, \\ C_2^+(f_{n,3}) = -[f] + [f'](\alpha + h) - [f'']\left(\frac{1}{2}\alpha^2 + \alpha h + \frac{3}{8}h^2\right), & \text{if } x > x^*, \end{cases} \tag{15}$$

with  $\beta = x_{n+1} - x^*$  and  $\alpha = x^* - x_{n+1}$ .

**Proof** We have to consider the five cases presented in the Lemma. Let us start by the first one:

- If  $x^* \in I_n^{-1}$ , then the approximation is  $O(h^3)$  for  $x > x^*$ , as all the values in the stencil of the operator  $L_2(f_{n,3}) = -\frac{1}{8}f_{n-1}^+ + \frac{5}{4}f_n^+ - \frac{1}{8}f_{n+1}^+$  belong to the + side of the domain, so  $C_{-1}^+(f_{n,3}) = 0$ . For  $x < x^*$ , we have that

$$\begin{aligned}
 f_{n-1}^+ &= f_{n-1}^- + [f] + [f']\beta + \frac{1}{2}[f'']\beta^2 + O(h^3), \\
 f_n^+ &= f_n^- + [f] + [f'](\beta + h) + \frac{1}{2}[f''](\beta + h)^2 + O(h^3), \\
 f_{n+1}^+ &= f_{n+1}^- + [f] + [f'](\beta + 2h) + \frac{1}{2}[f''](\beta + 2h)^2 + O(h^3),
 \end{aligned} \tag{16}$$

thus

$$\begin{aligned}
 L_2(f_{n,3}) &= -\frac{1}{8}f_{n-1}^+ + \frac{5}{4}f_n^+ - \frac{1}{8}f_{n+1}^+ \\
 &= -\frac{1}{8}f_{n-1}^- + \frac{5}{4}f_n^- - \frac{1}{8}f_{n+1}^- + \left([f] + [f'](\beta + h) \right. \\
 &\quad \left. + [f'']\left(\frac{1}{2}\beta^2 + \beta h + \frac{3}{8}h^2\right)\right) + O(h^3) \\
 &= -\frac{1}{8}f_{n-1}^- + \frac{5}{4}f_n^- - \frac{1}{8}f_{n+1}^- + C_{-1}^-(f_{n,3}) + O(h^3).
 \end{aligned} \tag{17}$$

- If  $x^* \in I_n^0$ , then

$$L_2(f_{n,3}) = -\frac{1}{8}f_{n-1}^- + \frac{5}{4}f_n^+ - \frac{1}{8}f_{n+1}^+.$$

Therefore, we have two cases. If  $x < x^*$ , proceeding as before and replacing the + values in terms of the - ones, we obtain

$$\begin{aligned} L_2(f_{n,3}) &= -\frac{1}{8}f_{n-1}^- + \frac{5}{4}f_n^+ - \frac{1}{8}f_{n+1}^+ \\ &= -\frac{1}{8}f_{n-1}^- + \frac{5}{4}f_n^- - \frac{1}{8}f_{n+1}^- \\ &\quad + \left(\frac{9}{8}[f] + \frac{1}{8}[f'](9\beta - h) + \frac{1}{8}[f'']\left(5\beta^2 - \frac{1}{2}(\beta + h)^2\right) + O(h^3)\right) \tag{18} \\ &= -\frac{1}{8}f_{n-1}^- + \frac{5}{4}f_n^- - \frac{1}{8}f_{n+1}^- + C_0^-(f_{n,3}) + O(h^3), \end{aligned}$$

with  $\beta = x_n - x^*$ .

If  $x > x^*$ , then we have to replace the - values in terms of the + ones. We obtain

$$\begin{aligned} L_2(f_{n,3}) &= -\frac{1}{8}f_{n-1}^- + \frac{5}{4}f_n^+ - \frac{1}{8}f_{n+1}^+ \\ &= -\frac{1}{8}f_{n-1}^+ + \frac{5}{4}f_n^+ - \frac{1}{8}f_{n+1}^+ + \left(\frac{1}{8}[f] - \frac{1}{8}[f']\alpha + \frac{1}{16}[f'']\alpha^2\right) + O(h^3) \\ &= -\frac{1}{8}f_{n-1}^+ + \frac{5}{4}f_n^+ - \frac{1}{8}f_{n+1}^+ + C_0^+(f_{n,3}) + O(h^3), \end{aligned} \tag{19}$$

with  $\alpha = x^* - x_{n-1}$ .

- This case is symmetric to the previous one. If  $x^* \in I_n^1$ , then

$$L_2(f_{n,3}) = -\frac{1}{8}f_{n-1}^- + \frac{5}{4}f_n^- - \frac{1}{8}f_{n+1}^+.$$

Therefore, we have two cases again. If  $x < x^*$ ,

$$\begin{aligned} L_2(f_{n,3}) &= -\frac{1}{8}f_{n-1}^- + \frac{5}{4}f_n^- - \frac{1}{8}f_{n+1}^+ \\ &= -\frac{1}{8}f_{n-1}^- + \frac{5}{4}f_n^- - \frac{1}{8}f_{n+1}^- - \left(\frac{1}{8}[f] + \frac{1}{8}[f']\beta + \frac{1}{16}[f'']\beta^2\right) + O(h^3) \\ &= -\frac{1}{8}f_{n-1}^- + \frac{5}{4}f_n^- - \frac{1}{8}f_{n+1}^- + C_1^-(f_{n,3}) + O(h^3), \end{aligned} \tag{20}$$

with  $\beta = x_{n+1} - x^*$ .

If  $x > x^*$ , then we have to replace the - values in terms of the + ones as before. We

obtain

$$\begin{aligned}
 L_2(f_{n,3}) &= -\frac{1}{8}f_{n-1}^- + \frac{5}{4}f_n^- - \frac{1}{8}f_{n+1}^+ \\
 &= -\frac{1}{8}f_{n-1}^+ + \frac{5}{4}f_n^+ - \frac{1}{8}f_{n+1}^+ \\
 &\quad + \left( -\frac{9}{8}[f] + \frac{1}{8}[f'](9\alpha - h) - \frac{1}{8}[f''](5\alpha^2 - \frac{1}{2}(\alpha + h)^2) \right) + O(h^3) \tag{21} \\
 &= -\frac{1}{8}f_{n-1}^+ + \frac{5}{4}f_n^+ - \frac{1}{8}f_{n+1}^+ + C_1^-(f_{n,3}) + O(h^3),
 \end{aligned}$$

with  $\alpha = x^* - x_{n-1}$ .

- The next case is symmetric to the first one. If  $x^* \in I_n^2$ , then the approximation is  $O(h^3)$  for  $x < x^*$ . Remind that, in this case, all the values in the stencil of the operator  $L_2(f_{n,3}) = -\frac{1}{8}f_{n-1}^- + \frac{5}{4}f_n^- - \frac{1}{8}f_{n+1}^-$  belong to the  $-$  side of the domain, so  $C^-(f_{n,3}) = 0$ . For  $x > x^*$ , we have that

$$\begin{aligned}
 L_2(f_{n,3}) &= -\frac{1}{8}f_{n-1}^- + \frac{5}{4}f_n^- - \frac{1}{8}f_{n+1}^- \\
 &= -\frac{1}{8}f_{n-1}^+ + \frac{5}{4}f_n^+ - \frac{1}{8}f_{n+1}^+ \\
 &\quad + \left( -[f] + [f'](\alpha + h) - [f'']\left(\frac{1}{2}\alpha^2 + \alpha h + \frac{3}{8}h^2\right) \right) + O(h^3) \tag{22} \\
 &= -\frac{1}{8}f_{n-1}^+ + \frac{5}{4}f_n^+ - \frac{1}{8}f_{n+1}^+ + C_2^+(f_{n,3}) + O(h^3),
 \end{aligned}$$

with  $\alpha = x^* - x_{n+1}$ .

□

After that, we define the values

$$C^\pm(f_{n+j,3}) = C_j^\pm(f_{n,3}), \quad j = -1, 0, 1, 2.$$

With this lemma, and these definitions, we design the following non-linear operator:

$$\tilde{L}_2(f_{n+j,3})(x) = L_2(f_{n+j,3}) + C_{n+j}^2(x) \tag{23}$$

being

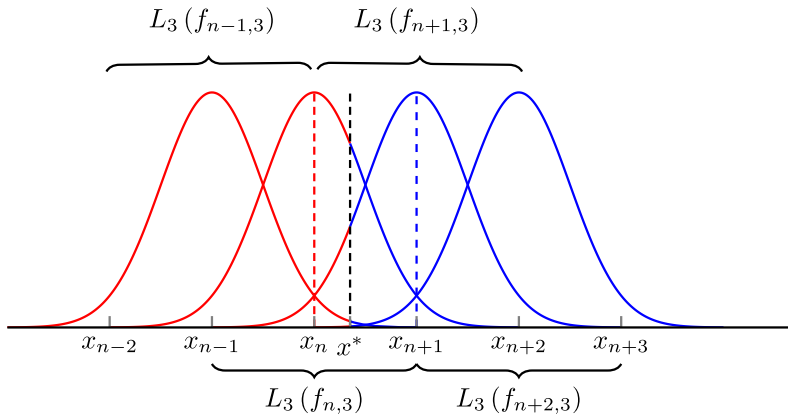
$$C_{n+j}^2(x) = \begin{cases} -C^-(f_{n+j,3}), & x^* \in [x_n, x_{n+1}], \quad x < x^*, \\ -C^+(f_{n+j,3}), & x^* \in [x_n, x_{n+1}], \quad x > x^*, \\ 0, & \text{in other case,} \end{cases} \tag{24}$$

with  $j = -1, 0, 1, 2$ , and its associate operator:

$$Q_2(f)(x) = \sum_{n \in \mathbb{Z}} \tilde{L}_2(f_{n,3})(x) B_2\left(\frac{x}{h} - n\right). \tag{25}$$

The approximation obtained using this new operator is of order of accuracy  $O(h^3)$ . We give a proof in the following theorem.





**Fig. 2** An example of a cubic B-spline base and the division of the domain in two subdomains. In this figure,  $x^*$  represents the location of the singularity

**Theorem 1** *Let us consider  $j_0 \in \mathbb{Z}$  and a finite interval  $I$  with  $[x_{j_0-\frac{3}{2}}, x_{j_0+\frac{3}{2}}] \subset I$ . If there exists a singularity placed at  $x^*$  in the interval  $[x_{j_0-\frac{3}{2}}, x_{j_0+\frac{3}{2}}]$ , then*

$$\|Q_2(f) - f\|_{\infty, I} = O(h^3).$$

**Proof** The proof is straightforward if we have a look to the expressions of the local truncation error in (17), (18), (19), (20), (21), and (22). □

**Corollary 1** *If the jump relations in (8) are to be approximated, the accuracy needed is  $O(h^3)$  for  $[f]$ ,  $O(h^2)$  for  $[f']$ , and so on.*

**Proof** Having a look to the expressions of the error in (12), (13), (14) and (15), the proof is straightforward taking into account the contribution to the error of the approximation of  $[f]$ ,  $[f']$ , and so on. □

**Corollary 2** *If the location of the discontinuity is to be approximated, the accuracy needed is  $O(h^3)$ .*

**Proof** Having a look to the expressions of the error in (12), (13), (14) and (15), the proof is straightforward taking into account the contribution to the error of the approximation of  $\beta$ . □

**Corollary 3** *The error of the corrected spline is smooth and retains the smoothness of the B-splines bases.*

**Proof** From the expressions of the error in (12), (13), (14) and (15), it is clear that the error of the spline before or after the correction retains the smoothness of the B-splines bases. □

### 3 Correction of the cubic B-spline quasi-interpolant

The local truncation error for cubic splines can be obtained in a way similar to the one followed for quadratic splines. In Fig. 2, we have represented a base of cubic B-splines and the division of the domain in two subdomains as we did in Fig. 1.

As in the previous section, we can state the following Lemma, which provides the accuracy attained by the cubic quasi-interpolant. For the proof, we can proceed in the same way as before (moving the discontinuity to different positions and considering only one B-spline), or consider the four B-splines of the base and a discontinuity placed at a fix position. In this case, we have chosen the second option, as we found it clearer.

**Lemma 2** *Let us consider a singularity placed at  $x^*$  in the interval  $[x_n, x_{n+1}]$ , then*

$$L_3(f_{n+j,3}) - L_3^\pm(f_{n+j,3}) = D^\pm(f_{n+j,3}) + O(h^4), \quad j = -1, 0, 1, 2$$

being for  $j = -1$

$$\begin{aligned} D^+(f_{n-1,3}) &= -[f] + [f'](\alpha + h) - \frac{1}{6}[f''] (3\alpha^2 + 6\alpha h + 2h^2) \\ &\quad + \frac{1}{6}[f''']\alpha(\alpha^2 + 3\alpha h + 2h^2), \quad \text{if } x < x^*, \\ D^-(f_{n-1,3}) &= 0, \quad \text{if } x > x^*, \end{aligned} \tag{26}$$

for  $j = 0$

$$\begin{aligned} D^-(f_{n,3}) &= -\frac{1}{6} \left( [f] + [f']\beta + \frac{1}{2}[f'']\beta^2 + \frac{1}{3!}[f''']\beta^3 \right), \quad \text{if } x < x^* \\ D^+(f_{n,3}) &= -\frac{7}{6}[f] + \frac{1}{6}[f'](7\alpha - h) - \frac{1}{6}[f''] \left( 4\alpha^2 - \frac{1}{2}(\alpha + h)^2 \right) \\ &\quad + \frac{1}{6}[f'''] \left( \frac{4}{3}\alpha^3 - \frac{1}{6}(\alpha + h)^3 \right), \quad \text{if } x > x^*, \end{aligned} \tag{27}$$

for  $j = 1$

$$\begin{aligned} D^-(f_{n+1,3}) &= \frac{7}{6}[f] + \frac{1}{6}[f'](7\beta - h) + \frac{1}{6}[f''] (4\beta^2 - \frac{1}{2}(\beta + h)^2) \\ &\quad + \frac{1}{6}[f'''] \left( \frac{4}{3}\beta^3 - \frac{1}{6}(\beta + h)^3 \right), \quad \text{if } x < x^* \\ D^+(f_{n+1,3}) &= \frac{4}{3} \left( -[f] + [f']\alpha - \frac{1}{2}[f'']\alpha^2 + \frac{1}{3!}[f''']\alpha^3 \right), \quad \text{if } x > x^*, \end{aligned} \tag{28}$$

and for  $j = 2$

$$\begin{aligned} D^+(f_{n+2,3}) &= 0, \quad \text{if } x > x^* \\ D^-(f_{n+2,3}) &= [f] + [f'](\beta + h) + \frac{1}{6}[f''] (3\beta^2 + 6\beta h + 2h^2) \\ &\quad + \frac{1}{6}[f''']\beta(\beta^2 + 3\beta h + 2h^2), \quad \text{if } x < x^*, \end{aligned} \tag{29}$$

with  $\alpha = x^* - x_n$  and  $\beta = x_{n+1} - x^*$ .

**Proof** Let us consider the four B-spline bases that appear in Fig. 2 and that contribute to the approximation in the interval  $[x_n, x_{n+1}]$ .

- Let us consider  $j = -1$ . From Fig. 2, we can clearly observe that  $L_3(f_{n-1,3})$  is correctly approximating in the interval  $[x_n, x^*]$ . Thus,  $C^-(f_{n-1,3}) = 0$ . On the other hand, in the interval  $(x^*, x_{n+1}]$ , the approximation provided is not correct, as the operator  $L_3(f_{n-1,3})$  uses information from the left of the discontinuity. Thus, the local truncation error can

be calculated using Taylor expansions around  $x^*$ . Proceeding in the same way as we did to obtain (10), we can write

$$\begin{aligned}
 f_{n-2}^- &= f_{n-2}^+ - [f] + [f'](2h + \alpha) - \frac{1}{2}[f''](2h + \alpha)^2 + \frac{1}{3!}[f'''](2h + \alpha)^3 + O(h^4), \\
 f_{n-1}^- &= f_{n-1}^+ - [f] + [f'](h + \alpha) - \frac{1}{2}[f''](h + \alpha)^2 + \frac{1}{3!}[f'''](h + \alpha)^3 + O(h^4), \quad (30) \\
 f_n^- &= f_n^+ - [f] + [f']\alpha - \frac{1}{2}[f'']\alpha^2 + \frac{1}{3!}[f''']\alpha^3 + O(h^4),
 \end{aligned}$$

so the local truncation error introduced using  $L_3(f_{n-1,3})$  in the interval  $(x^*, x_{n+1})$  can be obtained just replacing the expressions in (30):

$$\begin{aligned}
 L_3(f_{n-1,3}) &= \frac{4}{3}f_{n-1}^- - \frac{1}{6}(f_{n-2}^- + f_n^-) \\
 &= \frac{4}{3}\left(f_{n-1}^+ - [f] + [f'](h + \alpha) - \frac{1}{2}[f''](h + \alpha)^2 + \frac{1}{3!}[f'''](h + \alpha)^3 + O(h^4)\right) \\
 &\quad - \frac{1}{6}(f_{n-2}^+ - [f] + [f'](2h + \alpha) - \frac{1}{2}[f''](2h + \alpha)^2 \\
 &\quad + \frac{1}{3!}[f'''](2h + \alpha)^3 + O(h^4)) \\
 &\quad - \frac{1}{6}(f_n^+ - [f] + [f']\alpha - \frac{1}{2}[f'']\alpha^2 + \frac{1}{3!}[f''']\alpha^3 + O(h^4)) \quad (31) \\
 &= \frac{4}{3}f_{n-1}^+ - \frac{1}{6}(f_{n-2}^+ + f_n^+) \\
 &\quad + \left(-[f] + [f'](\alpha + h) - \frac{1}{6}[f''](3\alpha^2 + 6\alpha h + 2h^2) \right. \\
 &\quad \left. + \frac{1}{6}[f''']\alpha(\alpha^2 + 3\alpha h + 2h^2)\right) + O(h^4) \\
 &= \frac{4}{3}f_{n-1}^+ - \frac{1}{6}(f_{n-2}^+ + f_n^+) + D^+(f_{n-1,3}) + O(h^4).
 \end{aligned}$$

- When  $j = 2$ , the local truncation error can be obtained in a symmetric way. In particular, in the interval  $(x^*, x_{n+1}]$ ,  $L_3(f_{n+2,3})$  would be approximating correctly, while in the interval  $[x_n, x^*)$ , the local truncation error is

$$\begin{aligned}
 L_3(f_{n+2,3}) &= \frac{4}{3}f_{n+2}^+ - \frac{1}{6}(f_{n+1}^+ + f_{n+3}^+) \\
 &= \frac{4}{3}f_{n+2}^- - \frac{1}{6}(f_{n+1}^- + f_{n+3}^-) \\
 &\quad + \left([f] + [f'](\beta + h) + \frac{1}{6}[f''](3\beta^2 + 6\beta h + 2h^2) \right. \quad (32) \\
 &\quad \left. + \frac{1}{6}[f''']\beta(\beta^2 + 3\beta h + 2h^2)\right) + O(h^4) \\
 &= \frac{4}{3}f_{n+2}^- - \frac{1}{6}(f_{n+1}^- + f_{n+3}^-) + D^-(f_{n+2,3}) + O(h^4).
 \end{aligned}$$

- When  $j = 0$ , the local truncation error has to be considered in both intervals  $[x_n, x^*)$  and  $(x^*, x_{n+1}]$ . Proceeding as before, for the approximation in the interval  $[x_n, x^*)$ , we can write that

$$f_{n+1}^+ = f_{n+1}^- + [f] + [f']\beta + \frac{1}{2}[f'']\beta^2 + \frac{1}{3!}[f''']\beta^3 + O(h^4), \quad (33)$$

Then,  $L_3(f_{n,3})$  is

$$\begin{aligned}
 L_3(f_{n,3}) &= \frac{4}{3}f_n^- - \frac{1}{6}(f_{n-1}^- + f_{n+1}^+) \\
 &= \frac{4}{3}f_n^- - \frac{1}{6}f_{n-1}^- \\
 &\quad - \frac{1}{6}\left(f_{n+1}^- + [f] + [f']\beta + \frac{1}{2}[f'']\beta^2 + \frac{1}{3!}[f''']\beta^3 + O(h^4)\right) \\
 &= \frac{4}{3}f_n^- - \frac{1}{6}(f_{n-1}^- + f_{n+1}^-) \\
 &\quad + \left(-\frac{1}{6}\left([f] + [f']\beta + \frac{1}{2}[f'']\beta^2 + \frac{1}{3!}[f''']\beta^3 + O(h^4)\right)\right) \\
 &= \frac{4}{3}f_n^- - \frac{1}{6}(f_{n-1}^- + f_{n+1}^-) + D^-(f_{n,3}) + O(h^4).
 \end{aligned} \tag{34}$$

In the interval  $(x^*, x_{n+1}]$ , we have that

$$\begin{aligned}
 f_{n-1}^- &= f_{n-1}^+ - [f] + [f'](h + \alpha) - \frac{1}{2}[f''](h + \alpha)^2 + \frac{1}{3!}[f'''](h + \alpha)^3 + O(h^4), \\
 f_n^- &= f_n^+ - [f] + [f']\alpha - \frac{1}{2}[f'']\alpha^2 + \frac{1}{3!}[f''']\alpha^3 + O(h^4),
 \end{aligned} \tag{35}$$

so  $L_3(f_{n,3})$  is

$$\begin{aligned}
 L_3(f_{n,3}) &= \frac{4}{3}f_n^- - \frac{1}{6}(f_{n-1}^- + f_{n+1}^+) \\
 &= \frac{4}{3}\left(f_n^+ - [f] + [f']\alpha - \frac{1}{2}[f'']\alpha^2 + \frac{1}{3!}[f''']\alpha^3\right) \\
 &\quad - \frac{1}{6}\left(f_{n-1}^+ - [f] + [f'](h + \alpha) - \frac{1}{2}[f''](h + \alpha)^2 + \frac{1}{3!}[f'''](h + \alpha)^3\right) \\
 &\quad - \frac{1}{6}f_{n+1}^+ + O(h^4) \\
 &= \frac{4}{3}f_n^+ - \frac{1}{6}(f_{n-1}^+ + f_{n+1}^+) \\
 &\quad + \left(-\frac{7}{6}[f] + \frac{1}{6}[f'](7\alpha - h) - \frac{1}{6}[f''](4\alpha^2 - \frac{1}{2}(\alpha + h)^2)\right. \\
 &\quad \left.+ \frac{1}{6}[f''']\left(\frac{4}{3}\alpha^3 - \frac{1}{6}(\alpha + h)^3\right)\right) + O(h^4) \\
 &= \frac{4}{3}f_n^+ - \frac{1}{6}(f_{n-1}^+ + f_{n+1}^+) + D^+(f_{n,3}) + O(h^4).
 \end{aligned} \tag{36}$$

- The last case is symmetric to the first one. Again, the local truncation error has to be considered in both intervals  $[x_n, x^*)$  and  $(x^*, x_{n+1}]$ . Thus, in the interval  $[x_n, x^*)$ , we can write that

$$\begin{aligned}
 L_3(f_{n+1,3}) &= \frac{4}{3}f_n^- - \frac{1}{6}(f_{n+1}^+ + f_{n+2}^+) \\
 &= \frac{4}{3}f_n^- - \frac{1}{6}\left(\left(f_{n+1}^- + [f] + [f']\beta + \frac{1}{2}[f'']\beta^2 + \frac{1}{3!}[f''']\beta^3 + O(h^4)\right)\right. \\
 &\quad \left.+ \left(f_{n+1}^- + [f] + [f'](\beta + h) + \frac{1}{2}[f''](\beta + h)^2 + \frac{1}{3!}[f'''](\beta + h)^3 + O(h^4)\right)\right) \\
 &= \frac{4}{3}f_n^- - \frac{1}{6}(f_{n-1}^- + f_{n+1}^-) \\
 &\quad + \left(\frac{7}{6}[f] + \frac{1}{6}[f'](7\beta - h) + \frac{1}{6}[f''](4\beta^2 - \frac{1}{2}(\beta + h)^2)\right. \\
 &\quad \left.+ \frac{1}{6}[f''']\left(\frac{4}{3}\beta^3 - \frac{1}{6}(\beta + h)^3\right)\right) + O(h^4) \\
 &= \frac{4}{3}f_n^- - \frac{1}{6}(f_{n-1}^- + f_{n+1}^-) + D^-(f_{n+1,3}) + O(h^4).
 \end{aligned}
 \tag{37}$$

And in the interval  $(x^*, x_{n+1}]$ , we have that

$$\begin{aligned}
 L_3(f_{n+1,3}) &= \frac{4}{3}f_n^- - \frac{1}{6}(f_{n+1}^+ + f_{n+2}^+) \\
 &= \frac{4}{3}\left(-[f] + [f']\alpha - \frac{1}{2}[f'']\alpha^2 + \frac{1}{3!}[f''']\alpha^3 + O(h^4)\right) - \frac{1}{6}(f_{n+1}^+ + f_{n+2}^+) \\
 &= \frac{4}{3}f_n^+ - \frac{1}{6}(f_{n-1}^+ + f_{n+1}^+) \\
 &\quad + \frac{4}{3}\left(-[f] + [f']\alpha - \frac{1}{2}[f'']\alpha^2 + \frac{1}{3!}[f''']\alpha^3 + O(h^4)\right) \\
 &= \frac{4}{3}f_n^+ - \frac{1}{6}(f_{n-1}^+ + f_{n+1}^+) + D^+(f_{n+1,3}) + O(h^4).
 \end{aligned}
 \tag{38}$$

□

After this lemma, we define the following non-linear operator:

$$\tilde{L}_3(f_{n+j,3})(x) = L_3(f_{n+j,3}) + C_{n+j}^3(x)
 \tag{39}$$

being

$$C_{n+j}^3(x) = \begin{cases} -D^-(f_{n+j,3}), & x^* \in [x_n, x_{n+1}], \quad x < x^*, \\ -D^+(f_{n+j,3}), & x^* \in [x_n, x_{n+1}], \quad x > x^*, \\ 0, & \text{in other case,} \end{cases}
 \tag{40}$$

with  $j = -1, 0, 1, 2$ , and its associate operator:

$$Q_3(f)(x) = \sum_{n \in \mathbb{Z}} \tilde{L}_3(f_{n,3})(x) B_3\left(\frac{x}{h} - n\right).
 \tag{41}$$

Now, we can easily give a proof for the following theorem:

**Theorem 2** *Let us consider  $n_0 \in \mathbb{Z}$  and a finite interval  $I$  with  $[x_{n_0}, x_{n_0+1}] \subset I$ . If there exists a singularity placed at  $x^*$  in the interval  $[x_{n_0}, x_{n_0+1}]$ , then*

$$\|Q_3(f) - f\|_{\infty, I} = O(h^4).$$

**Proof** As in the previous section, the proof is straightforward if we have a look to the expressions of the local truncation error in (31), (32), (34), (36), (37), and (38). □

**Remark 1** Similar considerations can be taken here regarding the accuracy needed for the jump relations in (8) (as in Corollary 1). But the conclusions are similar: the accuracy needed is  $O(h^4)$  for  $[f]$ ,  $O(h^3)$  for  $[f']$ , and so on.

**Remark 2** Using the same reasoning as in Corollary 2, the accuracy needed for the approximation of the location of the discontinuity (if we do not know it) must be  $O(h^4)$  for cubic splines.

**Remark 3** As considered in Corollary 3, the error of the corrected cubic spline is smooth and retains the smoothness of the B-splines bases. Just having a look to the expressions of the error in (26), (27), (28) and (29), it is clear that the error of the spline before or after the correction retains the smoothness of the B-splines bases.

**Remark 4** As mentioned at the beginning of Sect. 2, the extension of the techniques presented in this article to data given in the cell-averages setting is straightforward considering the primitive function (Aràndiga and Donat 2000), which is a function given in the point values.

## 4 The new method in higher dimensions

In this section, we want to discuss how to extend the results presented in previous sections to higher dimensions using tensor product. We have selected the tensor product approach because it intuitively extends the results of B-splines from one dimension and it is widely used. This method takes advantage of the separable way in which we can write spline bases in multiple dimensions. However, this approach may pose challenges if discontinuities are not handled with care. There are non-separable methods in the literature for approximating multivariate functions with discontinuities in different contexts, see for example, the discussion about this topic introduced in Mateř and Meignen (2015), Arandiga et al. (2008), and Arandiga et al. (2010). In what follows, the reader can find more details about the mentioned extension using tensor product.

Let be  $1 \leq k \in \mathbb{N}$ , an open set  $\Omega \subseteq \mathbb{R}^k$  and a function  $f : \mathbb{R}^k \rightarrow \mathbb{R}$  with  $f \in C^l(\mathbb{R}^k)$ . We suppose  $h > 0$ ,  $\mathbf{n} = (n_1, \dots, n_k) \in \mathbb{Z}^k$  and consider

$$\begin{aligned} f_{\mathbf{n}} &= f(n_1h, \dots, n_kh), \\ f_{\mathbf{n},3} &= \{f_{(n_1+j_1, \dots, n_k+j_k)} = f(n_1h + j_1h, \dots, n_kh + j_kh) : |j_s| \leq 1, s = 1, \dots, k\}. \end{aligned} \quad (42)$$

With this notation, we define the tensor product of the operator  $L_p$  with  $p = 2, 3$ , Eq. (5), as

$$L_{\mathbf{p}}(f_{\mathbf{n},3}) = L_{(p_1, \dots, p_k)}(f_{\mathbf{n},3}) = \sum_{j_1=-1}^1 \dots \sum_{j_k=-1}^1 c_{j_1}^{p_1} \dots c_{j_k}^{p_k} f_{(n_1+j_1, \dots, n_k+j_k)},$$

with  $\mathbf{p} = (p_1, \dots, p_k) \in \{2, 3\}^k$  since  $L_p$  is linear. For example if  $k = 2$  and  $\mathbf{p} = (p, p)$ , then

$$\begin{aligned} L_{(p,p)}(f_{\mathbf{n},3}) &= L_{(p,p)}(f_{(n_1,n_2),3}) = \sum_{j_1=-1}^1 \sum_{j_2=-1}^1 c_{j_1}^p c_{j_2}^p f_{(n_1+j_1,n_2+j_2)} \\ &= \sum_{j_1=-1}^1 c_{j_1}^p (c_{-1}^p f_{(n_1-1,n_2+j_2)} + c_0^p f_{(n_1,n_2+j_2)} + c_1^p f_{(n_1+1,n_2+j_2)}) \\ &= c_{-1}^p (c_{-1}^p f_{(n_1-1,n_2-1)} + c_0^p f_{(n_1,n_2-1)} + c_1^p f_{(n_1+1,n_2-1)}) + \\ &\quad + c_0^p (c_{-1}^p f_{(n_1-1,n_2)} + c_0^p f_{(n_1,n_2)} + c_1^p f_{(n_1+1,n_2)}) + \\ &\quad + c_1^p (c_{-1}^p f_{(n_1-1,n_2+1)} + c_0^p f_{(n_1,n_2+1)} + c_1^p f_{(n_1+1,n_2+1)}). \end{aligned}$$

Thus, if  $p = 2$ , we get

$$\begin{aligned} L_{(2,2)}(f_{\mathbf{n},3}) &= -\frac{1}{8} \left( -\frac{1}{8} f_{(n_1-1,n_2-1)} + \frac{5}{4} f_{(n_1,n_2-1)} - \frac{1}{8} f_{(n_1+1,n_2-1)} \right) + \\ &\quad + \frac{5}{4} \left( -\frac{1}{8} f_{(n_1-1,n_2)} + \frac{5}{4} f_{(n_1,n_2)} - \frac{1}{8} f_{(n_1+1,n_2)} \right) + \\ &\quad - \frac{1}{8} \left( -\frac{1}{8} f_{(n_1-1,n_2+1)} + \frac{5}{4} f_{(n_1,n_2+1)} - \frac{1}{8} f_{(n_1+1,n_2+1)} \right) \tag{43} \\ &= \frac{1}{64} f_{(n_1-1,n_2-1)} - \frac{5}{32} f_{(n_1,n_2-1)} + \frac{1}{64} f_{(n_1+1,n_2-1)} - \frac{5}{32} f_{(n_1-1,n_2)} \\ &\quad + \frac{25}{16} f_{(n_1,n_2)} \\ &\quad - \frac{5}{32} f_{(n_1+1,n_2)} + \frac{1}{64} f_{(n_1-1,n_2+1)} - \frac{5}{32} f_{(n_1,n_2+1)} + \frac{1}{64} f_{(n_1+1,n_2+1)}. \end{aligned}$$

and if  $p = 3$ , we obtain

$$\begin{aligned} L_{(3,3)}(f_{\mathbf{n},3}) &= \frac{1}{36} f_{(n_1-1,n_2-1)} - \frac{2}{9} f_{(n_1,n_2-1)} + \frac{1}{36} f_{(n_1+1,n_2-1)} + \frac{16}{9} f_{(n_1-1,n_2)} + \frac{16}{9} f_{(n_1,n_2)} \tag{44} \\ &\quad - \frac{2}{9} f_{(n_1+1,n_2)} + \frac{1}{36} f_{(n_1-1,n_2+1)} - \frac{2}{9} f_{(n_1,n_2+1)} + \frac{1}{36} f_{(n_1+1,n_2+1)}. \end{aligned}$$

Let us start by expressing (4) in several dimensions:

$$Q_{\mathbf{p}}(f)(\mathbf{x}) = \sum_{\mathbf{n} \in \mathbb{Z}^k} L_{\mathbf{p}}(f_{\mathbf{n},3}) B_{\mathbf{p}}\left(\frac{\mathbf{x}}{h} - \mathbf{n}\right), \tag{45}$$

with

$$B_{\mathbf{p}}\left(\frac{\mathbf{x}}{h} - \mathbf{n}\right) = \prod_{j=1}^k B_{p_j}\left(\frac{x_j}{h} - n_j\right).$$

In particular, if  $k = 2$  and  $\mathbf{p} = (p, p)$ , we get

$$Q_{(p,p)}(f)(x_1, x_2) = \sum_{n_2, n_1 \in \mathbb{Z}} L_{(p,p)}(f_{(n_1,n_2),3}) B_p\left(\frac{x_1}{h} - n_1\right) B_p\left(\frac{x_2}{h} - n_2\right). \tag{46}$$

It is clear that this construction allows to use the correction terms presented in Sects. 2 and 3 by rows and by columns. In this case, as the correction is applied dimension by dimension,

the unidimensional algorithm presented in Aràndiga et al. (2005) can still be used. We define the following operation between two non-linear  $\tilde{L}_p$  operators:

**Definition 1** Let  $\tilde{L}_{p_1}, \tilde{L}_{p_2}$  be two operators defined in Eqs. (23) and (39), with  $p_1, p_2 = 2, 3$ , let  $\mathbf{n} \in \mathbb{Z}^2$  and  $f_{\mathbf{n},3} = (f_{(n_1+j_1, n_2+j_2)})_{j_1, j_2=-1}^1$ , then we define the product:

$$\tilde{L}_{p_2} \otimes \tilde{L}_{p_1}(f_{\mathbf{n},3})(x_1, x_2) = \sum_{j_2=-1}^1 \left( c_{j_2}^{p_2} \tilde{L}_{p_1}(f_{n_1,3,n_2+j_2})(x_1) + C_{(n_1, n_2+j_2)}^{p_2}(x_2) \right) \quad (47)$$

where  $f_{n_1,3,n_2+j_2} = (f_{(n_1-1, n_2+j_2)}, f_{(n_1, n_2+j_2)}, f_{(n_1+1, n_2+j_2)})$  and  $C_{(k_1, k_2)}(x_i)$  are the corrections terms obtained in Lemmas 1 and 2, applied over the variable  $x_i, i = 1, 2$ .

For example, if  $p_1 = p_2 = 2$ , then

$$\begin{aligned} \tilde{L}_2 \otimes \tilde{L}_2(f_{\mathbf{n},3})(x_1, x_2) &= \sum_{j_2=-1}^1 \left( c_{j_2}^2 \tilde{L}_{p_1}(f_{n_1,3,n_2+j_2})(x_1) + C_{(n_1, n_2+j_2)}^2(x_2) \right) \\ &= \frac{5}{4} \tilde{L}_{p_1}(f_{n_1,3,n_2})(x_1) - \frac{1}{8} (\tilde{L}_{p_1}(f_{n_1,3,n_2-1})(x_1) + \tilde{L}_{p_1}(f_{n_1,3,n_2+1})(x_1)) \\ &\quad + C_{(n_1, n_2)}^2(x_2) + C_{(n_1, n_2-1)}^2(x_2) + C_{(n_1, n_2+1)}^2(x_2) \\ &= \frac{5}{4} \left( \frac{5}{4} f_{(n_1, n_2)} - \frac{1}{8} (f_{(n_1-1, n_2)} + f_{(n_1+1, n_2)}) + C_{(n_1, n_2)}^2(x_1) \right) \\ &\quad - \frac{1}{8} \left( \frac{5}{4} f_{(n_1, n_2-1)} - \frac{1}{8} (f_{(n_1-1, n_2-1)} + f_{(n_1+1, n_2-1)}) + C_{(n_1, n_2-1)}^2(x_1) \right) \\ &\quad - \frac{1}{8} \left( \frac{5}{4} f_{(n_1, n_2+1)} - \frac{1}{8} (f_{(n_1-1, n_2+1)} + f_{(n_1+1, n_2+1)}) + C_{(n_1, n_2+1)}^2(x_1) \right) \\ &\quad + C_{(n_1, n_2)}^2(x_2) + C_{(n_1, n_2-1)}^2(x_2) + C_{(n_1, n_2+1)}^2(x_2). \end{aligned} \quad (48)$$

Definition 1 can be extended using recursively the next relation:

$$\tilde{L}_{p_3} \otimes \tilde{L}_{p_2} \otimes \tilde{L}_{p_1}(f_{\mathbf{n},3}) := \tilde{L}_{p_3} \otimes (\tilde{L}_{p_2} \otimes \tilde{L}_{p_1})(f_{\mathbf{n},3}).$$

At this point, we can design a version of  $\tilde{L}_p$  in several dimensions. We take into account that the operator is non-linear, it depends on the points where the discontinuity is. Let  $\mathbf{x} = (x_1, \dots, x_k) \in \mathbb{R}^k$  be a point, then

$$\tilde{L}_{\mathbf{p}}(f_{\mathbf{n},3})(\mathbf{x}) = \tilde{L}_{(p_1, \dots, p_k)}(f_{\mathbf{n},3})(\mathbf{x}) = \tilde{L}_{p_k} \otimes \dots \otimes \tilde{L}_{p_1}(f_{\mathbf{n},3})(\mathbf{x}). \quad (49)$$

Finally, the new non-linear spline in several dimensions based on B-splines is the following:

$$\mathcal{Q}_{\mathbf{p}}(f)(\mathbf{x}) = \sum_{\mathbf{n} \in \mathbb{Z}^k} \tilde{L}_{\mathbf{p}}(f_{\mathbf{n},3})(\mathbf{x}) B_{\mathbf{p}}\left(\frac{\mathbf{x}}{h} - \mathbf{n}\right). \quad (50)$$

In particular, if  $k = 2$  and  $\mathbf{p} = (p, p)$ , we get

$$\begin{aligned} \mathcal{Q}_{(p,p)}(f)(x_1, x_2) &= \sum_{n_2, n_1 \in \mathbb{Z}} \tilde{L}_{(p,p)}(x_1, x_2)(f_{(n_1, n_2),3}) B_p\left(\frac{x_1}{h} - n_1\right) B_p\left(\frac{x_2}{h} - n_2\right) \\ &= \sum_{n_2, n_1 \in \mathbb{Z}} \tilde{L}_p \otimes \tilde{L}_p(f_{\mathbf{n},3})(x_1, x_2) B_p\left(\frac{x_1}{h} - n_1\right) B_p\left(\frac{x_2}{h} - n_2\right). \end{aligned} \quad (51)$$



The processing can be done either by rows and then by columns or the other way around with similar results. From the expression in (48), we can see that we have chosen the first option. We can easily extend this result to any number of dimensions.

Now, we can discuss about the location of the singularity curve in several dimensions. For bivariate piecewise smooth functions, the interface where the singularity occurs is a curve in the plane. For trivariate functions, it is a surface in the space, etc. An accurate representation of the interface is still needed to obtain a reconstruction of the data with high order of accuracy. This representation can be done using a level set function. In this article we suppose that we have a level set function through which we can locate the position of the singularity curve.

#### 4.1 Discussion about how to locate the singularity using a level set surface in 2D

The location of the singularity in one-dimensional problems is a subject that has been already discussed in the literature, see for example (Aràndiga et al. 2005) and the references therein. About how to keep track of the location of the discontinuity in several dimensions, several approaches can also be found in the literature in other contexts, see for example (Matei and Meignen 2015; Arandiga et al. 2008, 2010; Floater et al. 2014; Romani et al. 2019; Allasia et al. 2009; Bozzini and Rossini 2000, 2013; Gout et al. 2008; Bracco et al. 2019) and the references therein. In the context of our multivariate data approximation, we have followed two approaches. If we have a look to the local truncation errors in Lemma 1 for quadratic splines or for those in Lemma 2 for cubic splines, we can see that the distance  $\beta$ , i.e. the location of the discontinuity, needs to be approximated (in the case it is unknown) with  $O(h^3)$  accuracy for quadratic splines or  $O(h^4)$  for cubic splines, if we do not want to affect the accuracy of the spline approximation close to the discontinuity. If a level set function  $\varphi(\mathbf{x})$  is used to locate the singularity, a simple one-dimensional approach can be followed to obtain the distance to the discontinuity from singular points (we consider singular points, those central points of the operator  $L_p$  in (4) for which the support of  $L_p$  crosses the discontinuity). This approach has been described in the context of the numerical approximation of the solution of PDEs with interfaces (see, e.g. Leveque and Li 1994; Li and Ito 2006). It is based on Taylor expansions and it is enough to obtain approximations of  $\alpha$  with  $O(h^3)$  accuracy in Lemma (1). Particularising the explanation in Leveque and Li (1994) and Li and Ito (2006) to one dimension, we can suppose that  $x$  is a grid point near the interface, then the position of the interface is  $x^* = x + \beta$ . Since  $x$  is close to the interface,  $\beta$  is small and we can use Taylor's expansion on the level set function  $\varphi(x)$  that is regular. Thus, writing  $\varphi(x^*) = \varphi(x + \beta) = 0$ , we get a second-order equation for the distance  $\beta$ ,

$$\varphi(x) + \varphi_x(x)\beta + \frac{1}{2}\varphi_{xx}(x)\beta^2 = 0. \quad (52)$$

As the level set function is smooth, the order of accuracy for  $\beta$  is  $O(h^3)$  if the approximation of the derivatives is obtained using the standard five points stencil. This approach is valid for quadratic splines (as an approach of  $O(h^3)$  accuracy for  $\beta$  is enough). For cubic splines, other approach is needed: either to add another term in the expansion to (52) and use some Newton's iterations to solve the equation, or try other possibilities. For example, assuming that we are still using a level set function for locating the position of the discontinuity curve, we can always obtain the absolute value of the level set to obtain a kink placed over the discontinuity curve. Then, we can use the algorithm in Aràndiga et al. (2005) to obtain accurate approximations of the location of the discontinuity in the  $x$  direction or the  $y$  direction. Both approaches present problems close to the boundary of the domain, so we

need to assume that there is a belt of some points all along the boundary of the domain where we cannot obtain an accurate approximation.

## 4.2 Process to approximate data in several dimensions

In this subsection, we explain the steps that we need to follow in order to approximate data in several dimensions:

1. We start from multivariate data at a low resolution, and a level set function which zero level set locates the position of the discontinuity.
2. We select one dimension of the data.
3. We process the data in the dimension selected using the unidimensional quadratic or cubic spline plus correction terms described in the previous sections.
4. We interpolate the level set function in the dimension selected (remind that it is a smooth function), for example, using the same quadratic or cubic spline interpolation (without correction terms), respectively. In order to improve computational cost, we can store and interpolate the level set within a wide enough tube around the discontinuity curve.
5. If there are no more dimensions to compute, we finish the process. Otherwise, we select another dimension of the data and go to step 3.

## 5 Numerical experiments

In this section, we present plots of the reconstructions obtained through classical and corrected quadratic and cubic splines. We will also present grid refinement experiments that aim to support the theoretical results obtained about the accuracy of the new algorithm. To do so, we start from data obtained from the sampling of a piecewise smooth function. The location of the discontinuity is considered known in the case of jump discontinuities in the function, but we will be able to approximate its location when the jumps are in the first-order derivative. In this case, we trust the approximation of the location of the singularity to the algorithm presented in Aràndiga et al. (2005).

For the grid refinement analysis, we obtain the infinity norm  $E^l = \|f^l - \tilde{f}^l\|_{L^\infty}$  over the domain, where  $l$  represents the step of the refinement process, which consists in reducing the grid size  $h$  to  $h/2$  when going from  $l$  to  $l + 1$ . Once the error has been computed, the numerical accuracy can be obtained through the classical formula:

$$O^l = \log_2 \left( \frac{E^l}{E^{l+1}} \right).$$

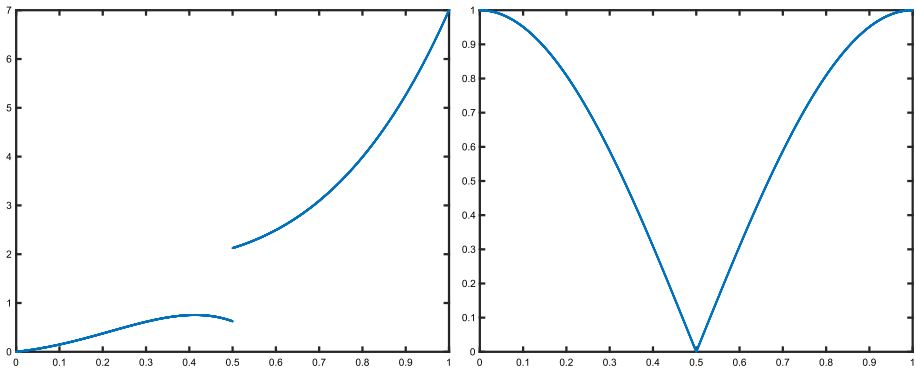
For the univariate numerical experiments, we consider the piecewise continuous function presented in (53):

$$f(x) = \begin{cases} -20x^4 + x^3 + 5x^2 + x, & \text{if } 0 \leq x < 0.5, \\ 4x^4 + x^3 + x^2 - x + 2, & \text{if } 0.5 \leq x \leq 1, \end{cases} \quad (53)$$

and the one in (54):

$$f(x) = |\cos(\pi x)|. \quad (54)$$

Both cases are presented in Fig. 3. In the first case, the function presents a jump discontinuity at  $x = 0.5$ , so we assume that the location of the discontinuity is known, but



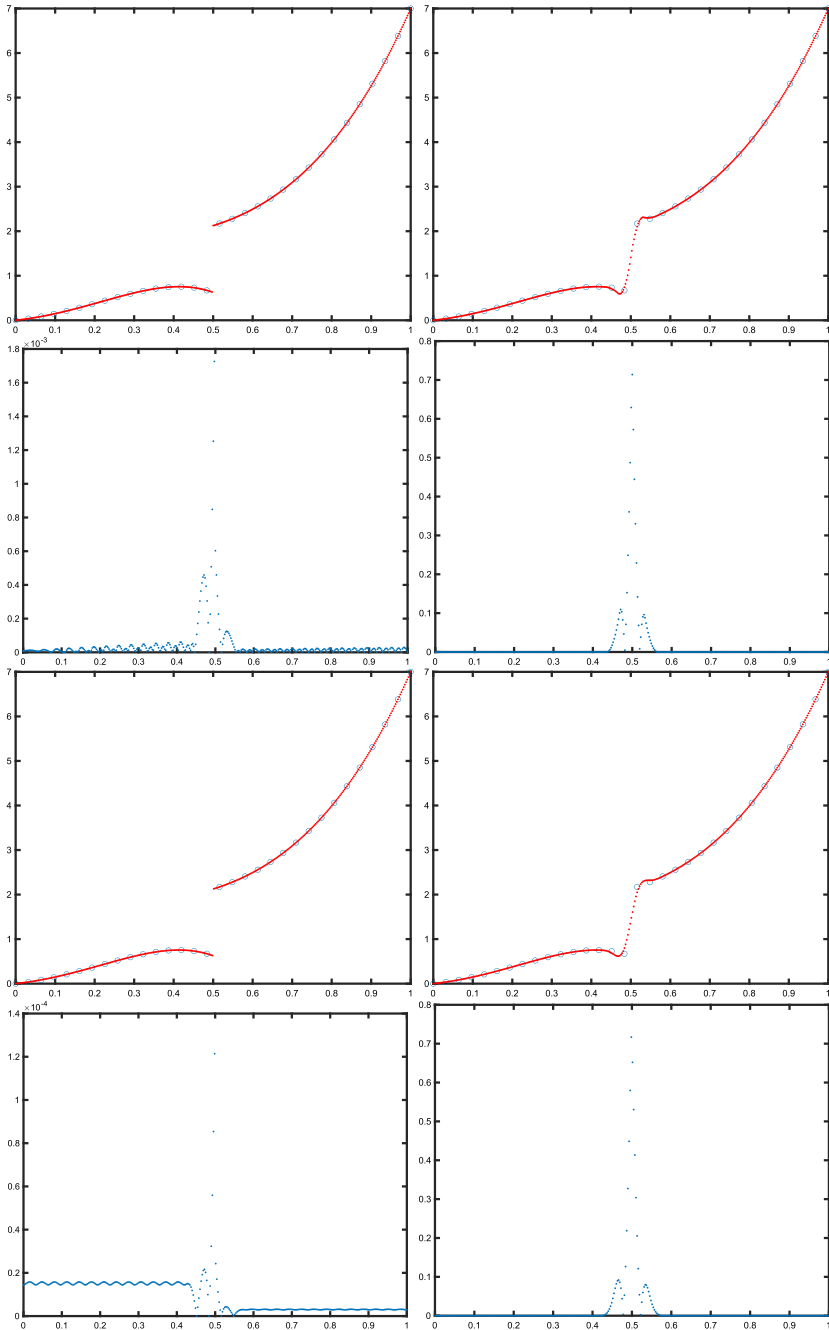
**Fig. 3** Function in (53) and (54) used in the numerical experiments

the jump in the function and the derivatives are obtained with enough accuracy using one-sided interpolation (Amat et al. 2018). For the second case, the function presents jumps in the derivatives at  $x = 0.5$ , but it is continuous, so the location of the singularity can be approximated using, for example, the algorithm in Aràndiga et al. (2005). As in the previous case, the jumps in the function and the derivatives can be also approximated using one-sided interpolation.

### 5.1 A first experiment with a function that presents a jump discontinuity

Let us start by the function presented in Fig. 3 to the left. It corresponds to the function in (53). In Fig. 4, we present the result obtained when approximating a sampling of 32 initial points of this function, using the classical quadratic (first and second rows) and cubic (third and fourth rows) splines with (left) and without corrections (right). Rows one and three show the approximation and rows two and four show the absolute value of the error. For these experiments, we have approximated 10 uniform points between the initial data nodes for the cubic spline and 11 uniform points for the quadratic.

We can see that the corrections allow to improve the accuracy of the reconstruction close to the singularity. Let us now analyse if the full order of accuracy (meaning  $O(h^3)$  accuracy for the quadratic spline and  $O(h^4)$  for the cubic one) is recovered. To do so, we present the results of a grid refinement experiment in Table 1. We can see that the numerical accuracy supports the theoretical results obtained in Theorems 1 and 2. We have also represented these results in a semilogarithmic scale in Fig. 5 to show the decreasing of the error, so that we can compare it with the theoretical results. To the left of this figure, we can observe the result of the quadratic spline, and to the right, the result of the cubic spline. In both plots, we present with red stars the errors in the infinity norm obtained by the classical spline, and with blue stars, the error of the corrected one. We have represented with dashed lines the theoretical decreasing of the error, which corresponds to  $O(1)$  accuracy for classical splines,  $O(h^3)$  for corrected quadratic splines and  $O(h^4)$  for corrected cubic splines.

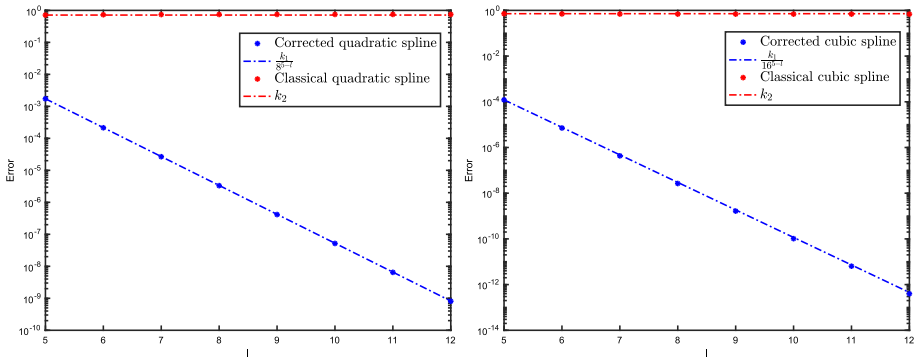


**Fig. 4** In the first row of the figure, we can observe the approximation of the function in (53) using 32 initial points through the corrected quadratic spline (left) and the classical quadratic spline (right). The second row shows the absolute error for each one of the splines. The third and fourth rows show the same results obtained by the corrected cubic spline (left) and the classical one (right)

**Table 1** Grid refinement analysis for the accuracy of the corrected and classical splines using the infinity norm at the high-resolution nodes

$m = 2^l$	$2^5$	$2^6$	$2^7$	$2^8$	$2^9$	$2^{10}$	$2^{11}$	$2^{12}$
Error corr. quadratic spline ( $E^l$ )	1.72587e-03	2.14192e-04	2.66571e-05	3.32427e-06	4.15024e-07	5.18456e-08	6.47866e-09	8.09705e-10
Order ( $\log_2(E^l/E^{l+1})$ )	-	3.01035	3.00631	3.00342	3.00177	3.00090	3.00045	3.00023
Error class. quadratic spline ( $E^l$ )	7.13931e-01	7.32170e-01	7.41145e-01	7.45589e-01	7.47798e-01	7.48900e-01	7.49450e-01	7.49725e-01
Order ( $\log_2(E^l/E^{l+1})$ )	-	-3.63924e-02	-1.75780e-02	-8.62404e-03	-4.26967e-03	-2.12412e-03	-1.05937e-03	-5.29007e-04
Error corr. cubic spline ( $E^l$ )	1.21409e-04	7.11761e-06	4.31004e-07	2.65177e-08	1.64442e-09	1.02376e-10	6.38589e-12	3.98570e-13
Order ( $\log_2(E^l/E^{l+1})$ )	-	4.09233	4.04562	4.02267	4.01130	4.00563	4.00284	4.00198
Error class. cubic spline ( $E^l$ )	7.16830e-01	7.00700e-01	6.92683e-01	6.88697e-01	6.86711e-01	6.85720e-01	6.85224e-01	6.84977e-01
Order ( $\log_2(E^l/E^{l+1})$ )	-	3.28342e-02	1.66018e-02	8.32622e-03	4.16689e-03	2.08408e-03	1.04216e-03	5.21103e-04

The original data have been sampled from the function in (53). The low-resolution nodes have been sampled with  $m = 2^l$  nodes and then we have approximated 10 uniform points between the initial data nodes for the cubic spline and 11 uniform points for the quadratic



**Fig. 5** Representation of the results of the grid refinement analysis for the numerical accuracy obtained for quadratic (left) and cubic (right) splines and presented in Table 1. The original data come from the function in (53). The error has been represented in a semilogarithmic scale so that we can appreciate the decreasing of the error. In both plots, we can observe, represented in red stars, the errors in the infinity norm obtained by the classical spline, and with blue stars, the error of the corrected one. We have represented with dashed lines the theoretical decreasing of the error

### 5.2 A second experiment with a piecewise smooth function that presents a jump at least in the first derivative

In this second experiment, we will use data that comes from the sampling of the function in (54). This function presents a jump in the first (and higher) order derivatives. In this case, we trust in the algorithm proposed in Aràndiga et al. (2005) for the location of the singularity. As in the previous experiment, the jumps in the function and the derivatives are approximated using one-sided interpolation (Amat et al. 2018). Figure 6 shows the results obtained when approximating a sampling of 32 initial points of the function in (54), using the classical quadratic (first and second rows) and cubic (third and fourth rows) splines with (left) and without corrections (right). Rows one and three show the approximation and rows 2 and 4 show the absolute value of the error. As we did in the previous experiment, we have approximated 10 uniform points between the initial data nodes for the cubic spline and 11 uniform points for the quadratic.

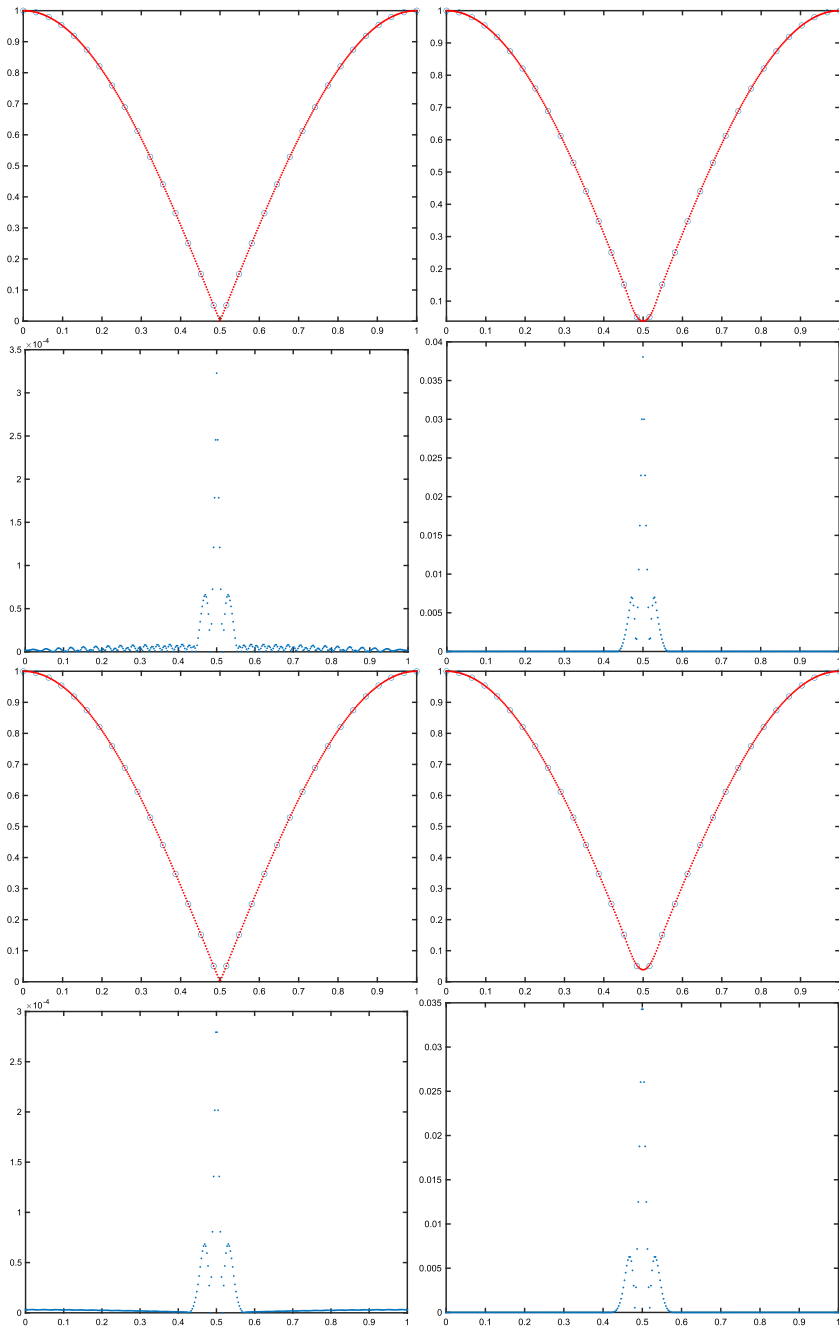
Table 2 and Fig. 7 present the results of a grid refinement analysis in the infinity norm. The conclusions that can be reached are similar to the ones obtained in previous experiment: Even in the presence of singularities, the corrected splines recover the accuracy of the spline at smooth zones in the infinity norm.

### 5.3 A third experiment for the approximation of a bivariate function using a cubic spline

We dedicate this subsection to present an experiment for the approximation of a bivariate function using the corrected cubic spline (results for quadratic splines allow to obtain similar conclusions). We set the level set function

$$\varphi(x, y) = x^2 + y^2 - r^2, \tag{55}$$

with  $r = 0.5$  and  $0 \leq x \leq 1, 0 \leq y \leq 1$ . The function presents a jump, so we use the level set function to track the position of the singularity using the first algorithm explained in



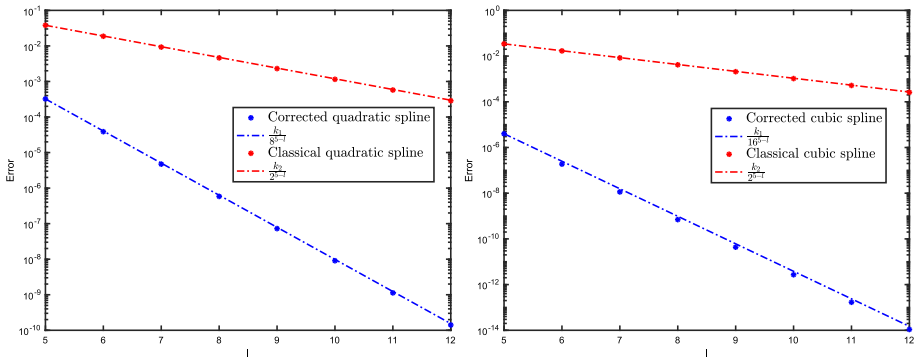
**Fig. 6** In the first row of the figure, we can observe the approximation of the function in (53) using 32 initial points through the corrected quadratic spline (left) and the classical quadratic spline (right). The second row shows the absolute error for each one of the splines. The third and fourth rows show the same results obtained by the corrected cubic spline (left) and the classical one (right)

**Table 2** Grid refinement analysis for the accuracy of the corrected and classical splines using the infinity norm at the high-resolution nodes

$m = 2^l$	$2^5$	$2^6$	$2^7$	$2^8$	$2^9$	$2^{10}$	$2^{11}$	$2^{12}$
Error corr. quadratic spline ( $E^l$ )	3.22831e-04	3.86807e-05	4.72820e-05	5.84294e-06	7.26148e-08	9.05044e-09	1.12965e-09	1.41105e-10
Order ( $\log_2(E^l/E^{l+1})$ )	-	3.06109	3.03225	3.01652	3.00836	3.00420	3.00211	3.00104
Error class. quadratic spline ( $E^l$ )	3.80518e-02	1.87058e-02	9.27707e-03	4.62008e-03	2.30548e-03	1.15161e-03	5.75524e-04	2.87692e-04
Order ( $\log_2(E^l/E^{l+1})$ )	-	1.02448	1.01174	1.00575	1.00285	1.00142	1.00071	1.00035
Error corr. cubic spline ( $E^l$ )	3.99431e-06	1.87492e-07	1.13586e-08	6.98917e-10	4.33424e-11	2.69840e-12	1.68421e-13	1.08802e-14
Order ( $\log_2(E^l/E^{l+1})$ )	-	4.41305	4.04497	4.02252	4.01127	4.00561	4.00196	3.95230
Error class. cubic spline ( $E^l$ )	3.42778e-02	1.68463e-02	8.35435e-03	4.16049e-03	2.07614e-03	1.03705e-03	5.18271e-04	2.59072e-04
Order ( $\log_2(E^l/E^{l+1})$ )	-	1.02484	1.01183	1.00577	1.00285	1.00142	1.00071	1.00035

The original data have been sampled from the function in (53). The low-resolution nodes have been sampled with  $m = 2^l$  nodes and then we have approximated 10 uniform points between the initial data nodes for the cubic spline and 11 uniform points for the quadratic





**Fig. 7** Representation of the results of the grid refinement analysis for the numerical accuracy obtained for quadratic (left) and cubic (right) splines and presented in Table 2. The original data comes from the function in (54). The error has been represented in a semilogarithmic scale so that we can appreciate the decreasing of the error. In both plots we can observe, represented in red stars, the errors in the infinity norm obtained by the classical spline, and with blue stars, the error of the corrected one. We have represented with dashed lines the theoretical decreasing of the error

Sect. 4.1. To obtain the results in several dimensions, we apply the extension of the algorithm introduced in Sect. 4. To check the numerical accuracy, we perform a grid refinement analysis in the infinity norm for the next bivariate function,

$$f(x, y) = \begin{cases} x^4 + y^4, & \text{if } \varphi(x, y) \leq 0, \\ x^3 + y^3 + 5, & \text{if } \varphi(x, y) > 0. \end{cases} \tag{56}$$

The results are presented in Table 3. We can see how the new technique keeps the numerical order of accuracy of the cubic spline at smooth zones in the infinity norm. The classical spline introduces oscillations and smearing close to the discontinuity curve, so it can not conserve the accuracy. In Fig. 8, we can see the reconstruction (in the first row of the figure) and the error (in the second row) obtained by the cubic spline in 2D plus correction terms (first column) and without correction terms (second column). The colour bars in the two figures of the error allow to appreciate the size of the error, which is mainly placed around the discontinuity curve.

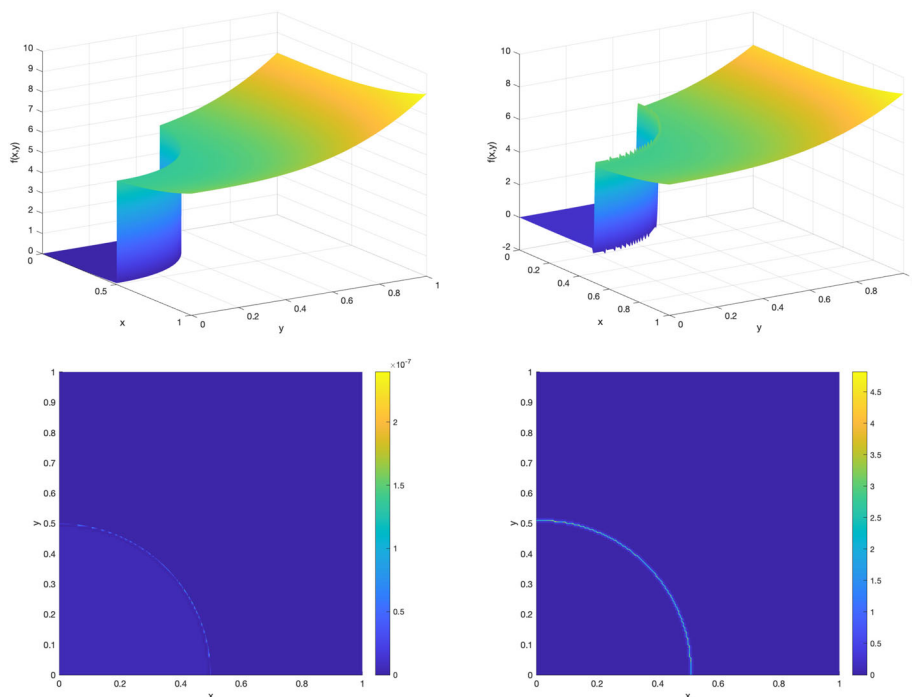
### 6 Conclusions

In this article, we have presented a new algorithm that allows for the approximation of piecewise smooth functions with full accuracy using cubic and quadratic splines. By full accuracy, we mean the accuracy of the spline at smooth zones. The algorithm is based on the computation of simple correction terms that can be added to the approximation obtained by the classical cubic or quadratic splines. We have given proofs for the accuracy of the approximation obtained through the new algorithm. Through a tensor product strategy, we have extended the one-dimensional results to any number of dimensions. The numerical experiments presented show that the reconstructions are piecewise smooth and that they conserve the accuracy of the splines at smooth zones even close to the singularities. The

**Table 3** Grid refinement analysis for the accuracy of the corrected and classical cubic splines using the infinity norm at the high resolution nodes

$m = 2^l$	$2^4$	$2^5$	$2^6$	$2^7$	$2^8$	$2^9$	$2^{10}$
Error corr. cubic spline ( $E^l$ )	7.1681e-04	6.4706e-05	4.3413e-06	2.3958e-07	1.2550e-08	1.1682e-09	6.3841e-11
Order ( $\log_2(E^l/E^{l+1})$ )	-	3.4696	3.8977	4.1795	4.2547	3.4254	4.1936
Error class. cubic spline ( $E^l$ )	3.9525	4.5343	4.4017	4.7992	4.8018	4.8035	4.8042
Order ( $\log_2(E^l/E^{l+1})$ )	-	-1.9810e-01	4.2821e-02	-1.2476e-01	-7.8009e-04	-5.0274e-04	-2.1877e-04

The original data have been sampled from the function in (56). The low-resolution nodes have been sampled with  $m = 2^l \times 2^l$  nodes and the high-resolution data with  $10(m - 1) + m \times 10(m - 1) + m$  points



**Fig. 8** In the first row of the figure, we can observe the approximation of the function in (56) using  $128 \times 128$  initial nodes through the corrected cubic spline (left) and the classical cubic spline (right). We approximate at 10 equidistant positions between every two nodes in each dimension, which results in a final resolution of  $1398 \times 1398$ . The second row shows the absolute error for each one of the splines. The colour bar in this last two figures allows to appreciate the size of the error, mainly placed over the discontinuity curve

analysis of the numerical accuracy in the infinity norm supports the theoretical results obtained in one and two dimensions.

**Funding** Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature. Dr. Z. Li has been partially supported by a Simon's grant 633724 and by Fundación Séneca grant 21760/IV/22. Dr. Juan Ruiz has been supported by the Spanish national research project PID2019-108336GB-I00 and by the Fundación Séneca grant 21728/EE/22 (Este trabajo es resultado de las estancias (21760/IV/22) y (21728/EE/22) financiadas por la Fundación Séneca-Agencia de Ciencia y Tecnología de la Región de Murcia con cargo al Programa Regional de Movilidad, Colaboración Internacional e Intercambio de Conocimiento "Jiménez de la Espada" (Plan de Actuación 2022)). Dr. Dionisio F. Yáñez has been supported through project CIAICO/2021/227 (Proyecto financiado por la Conselleria de Innovación, Universidades, Ciencia y Sociedad digital de la Generalitat Valenciana) and by grant PID2020-117211GB-I00 funded by MCIN/AEI/10.13039/501100011033.

**Data availability** All data generated or analysed during this study are included in this article.

## Declarations

**Conflict of interest** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give

appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Allasia G, Besenghi R, Cavoretto R (2009) Adaptive detection and approximation of unknown surface discontinuities from scattered data. *Simul Model Pract Theory* 17(6):1059–1070
- Amat S, Li Z, Ruiz J (2018) On a new algorithm for function approximation with full accuracy in the presence of discontinuities based on the immersed interface method. *J Sci Comput* 75(3):1500–1534
- Amat S, Levin D, Ruiz-Álvarez J, Yáñez DF (2023) A new b-spline type approximation method for non-smooth functions. *Appl Math Lett* 141:108628
- Aràndiga F, Donat R (2000) Nonlinear multiscale decompositions: the approach of A. Harten. *Numer Algorithms* 23(2–3):175–216
- Aràndiga F, Cohen A, Donat R, Dyn N (2005) Interpolation and approximation of piecewise smooth functions. *SIAM J Numer Anal* 43(1):41–57
- Arandiga F, Cohen A, Donat R, Dyn N, Matei B (2008) Approximation of piecewise smooth functions and images by edge-adapted (ENO-EA) nonlinear multiresolution techniques. *Appl Comput Harmon Anal* 24(2):225–250 (**Special Issue on Mathematical Imaging—Part II**)
- Arandiga F, Cohen A, Donat R, Matei B (2010) Edge detection insensitive to changes of illumination in the image. *Image Vis Comput* 28(4):553–562
- Aràndiga F, Donat R, López-Ureña S (2023) Nonlinear improvements of quasi-interpolating splines to approximate piecewise smooth functions. *Appl Math Comput* 448:127946
- Boehm W (1986) Multivariate spline methods in CAGD. *Comput Aided Des* 18(2):102–104
- Bozzini M, Rossini M (2000) Approximating surfaces with discontinuities. *Math Comput Modell* 31(6):193–213
- Bozzini M, Rossini M (2013) The detection and recovery of discontinuity curves from scattered data. *J Comput Appl Math* 240:148–162
- Bracco C, Davydov O, Giannelli C, Sestini A (2019) Fault and gradient fault detection and reconstruction from scattered data. *Comput Appl Geom Des* 75:101786
- Butzer PL, Schmidt K, Stark EL, Vogt L (1989) Central factorial numbers; their main properties and some applications. *Numer Funct Anal Optim* 10:419–488
- de Boor C (1990) Quasiinterpolants and approximation power of multivariate splines. Springer Netherlands, Dordrecht, pp 313–345
- de Boor C, Fix G (1973) Spline approximation by quasi-interpolants. *J Approx Theory* 8(1):19–45
- Floater M, Lyche T, Mazure M-L, Mørken K, Schumaker LL (2014) Mathematical methods for curves and surfaces, vol 8177 of Lecture notes in computer science. Springer, Berlin
- Forster B (2011) Splines and multiresolution analysis. *Handbook of mathematical methods in imaging*. Springer, Berlin
- Gout C, Guyader CL, Romani L, Saint-Guirons A (2008) Approximation of surfaces with fault(s) and/or rapidly varying data, using a segmentation process, DM-splines and the finite element method. *Numer Algorithms* 48:67–92
- Katsoulis T, Wang X, Kaklis P (2019) A t-splines-based parametric modeller for computer-aided ship design. *Ocean Eng* 191:106433
- Lehmann T, Gonner C, Spitzer K (2001) Addendum: B-spline interpolation in medical image processing. *IEEE Trans Med Imaging* 20(7):660–665
- Leveque RJ, Li Z (1994) The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *SIAM J Numer Anal* 31(4):1019–1044
- Li Z, Ito K (2006) The immersed interface method: numerical solutions of PDEs involving interfaces and irregular domains (*Frontiers in applied mathematics*). Society for Industrial and Applied Mathematics (SIAM), Philadelphia
- Lyche T, Schumaker LL (1975) Local spline approximation methods. *J Approx Theory* 15(4):294–325
- Lyche T, Manni C, Speleers H (2018) Foundations of spline theory: B-splines, spline approximation, and hierarchical refinement. In: Lyche T, Manni C, Speleers H (eds) *Splines and PDEs: from approximation theory to numerical linear algebra*, vol 2219 of Lecture notes in mathematics. Springer, Berlin

- Matef B, Meignen S (2015) Nonlinear and nonseparable bidimensional multiscale representation based on cell-average representation. *IEEE Trans Image Process* 24(11):4570–4580
- Nowacki H (2010) Five decades of computer-aided ship design. *Comput-Aid Des* 42(11):956–969 (**Computer aided ship design: some recent results and steps ahead in theory, methodology and practice**)
- Rogers DF, Satterfield SG (1980) B-spline surfaces for ship hull design. *ACM SIGGRAPH Comput Graph* 14(3):211–217. <https://doi.org/10.1145/965105.807494>
- Romani L, Rossini M, Schenone D (2019) Edge detection methods based on RBF interpolation. *J Comput Appl Math* 349:532–547
- Sablonnière P (2005) Recent progress on univariate and multivariate polynomial and spline quasi-interpolants. In: Mache DH, Szabados J, de Bruin MG (eds) *Trends and applications in constructive approximation*. Birkhäuser Basel, Basel, pp 229–245
- Sarfraz M (1998) A rational spline with tension: some cagd perspectives. In: *Proceedings. 1998 IEEE conference on information visualization. An international conference on computer visualization and graphics (Cat. No.98TB100246)*, pp 178–183
- Speleers H (2017) Hierarchical spline spaces: quasi-interpolants and local approximation estimates. *Adv Comput Math* 43:235–255
- Unser M (1999) Splines: a perfect fit for signal and image processing. *IEEE Signal Process Mag* 16(6):22–38
- Zheng J, Zhang J, Zhou H, Shen LG (2005) Smooth spline surface generation over meshes of irregular topology. *Vis Comput* 21:858–864

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.