Journal of Computational and Applied Mathematics 473 (2026) 116891

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

Non-linear Partition of Unity method☆

José M. Ramón [a], Juan Ruiz-Álvarez [b], Dionisio F. Yáñez [a,*]

[a] Departamento de Matemáticas. Universidad de Valencia, Valencia, Spain
[b] Departamento de Matemática Aplicada y Estadística. Universidad Politécnica de Cartagena, Cartagena, Spain

ARTICLE INFO

Keywords:
WENO
High accuracy approximation
Improved adaption to discontinuities
RBF
41A05
41A10
65D05
65M06
65N06

ABSTRACT

This paper introduces the Non-linear Partition of Unity Method (NL-PUM), a novel technique integrating Radial Basis Function interpolation and Weighted Essentially Non-Oscillatory algorithms. As far as we know, this is the first time that a non-linear PUM method is introduced in the literature. The main advance of this proposal is providing an algorithm that keeps the properties of the PUM at smooth zones, while introducing a non-linear modification close to the discontinuities to avoid oscillations. This is done automatically computing estimations of the smoothness of the data and replacing the PUM by Sheppard method when all the data is affected by a discontinuity. Thus, the computation of smoothness indicators and the use of compactly supported base functions ensure precision in regions affected by the presence of discontinuities. Error bounds are calculated and validate the effectiveness of the new method, showing improved interpolation capabilities at discontinuity regions as well as at smooth zones. A battery of experiments is presented to check the theoretical results provided.

1. Introduction

Kernel-based methods are useful and powerful tools employed in several fields of science, such as computer-aided geometric design, the numerical solution of partial differential equations, image processing, and others. These schemes are satisfactorily used in interpolation, regression, and machine learning due to their easy and quick implementation and their simple generalization for any dimension.

The problem consists in finding a good approximation to an unknown function, $f : \mathbb{R}^n \to \mathbb{R}$, from some data values $F = \{f_i = f(\mathbf{x}_i),\ i = 1, \dots, N\}$ given a set of arbitrarily distributed points on an open and bounded domain $\Omega \subset \mathbb{R}^n$, $X = \{\mathbf{x}_i \in \Omega,\ i = 1, \dots, N\}$. The Radial Basis Function (RBF) method consists of developing the function as a linear combination of a basis of radial functions centered at the data points X. There exists a vast literature about it (see, e.g., [1–3]), however, the efficiency of the method can be improved in different ways. One approximation, called the *partition of unity method* (PUM or PU method) (see e.g., [1–5]), involves partitioning the entire domain into multiple smaller subdomains and applying the RBF method to each of them. Ultimately, a convex combination of these interpolants yields the final approximation. Satisfactory results are obtained when these methods are employed to data originating from continuous functions. For example, see the approximation presented in Fig. 1(a) where we approximate Franke's function:

$$\begin{aligned} f(x,y) = &\frac{3}{4}e^{-1/4((9x-2)^2+(9y-2)^2)} + \frac{3}{4}e^{-1/49(9x+1)^2-1/10(9y+1)} \\ &+ \frac{1}{2}e^{-1/4((9x-7)^2+(9y-3)^2)} - \frac{1}{5}e^{-(9x-4)^2-(9y-7)^2}, \end{aligned} \tag{1}$$

☆ Dionisio F. Yáñez has been supported through project CIAICO/2021/227, Spain (Proyecto financiado por la Conselleria de Innovación, Universidades, Ciencia y Sociedad digital de la Generalitat Valenciana), by grant PID2020-117211GB-I00 and by PID2023-146836NB-I00 funded by MCIN/AEI/10.13039/501100011033, Spain.

* Corresponding author.
E-mail addresses: Jose.Manuel.Ramon@uv.es (J.M. Ramón), juan.ruiz@upct.es (J. Ruiz-Álvarez), Dionisio.Yanez@uv.es (D.F. Yáñez).

https://doi.org/10.1016/j.cam.2025.116891
Received 16 January 2025; Received in revised form 23 April 2025
Available online 4 July 2025
0377-0427/© 2025 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

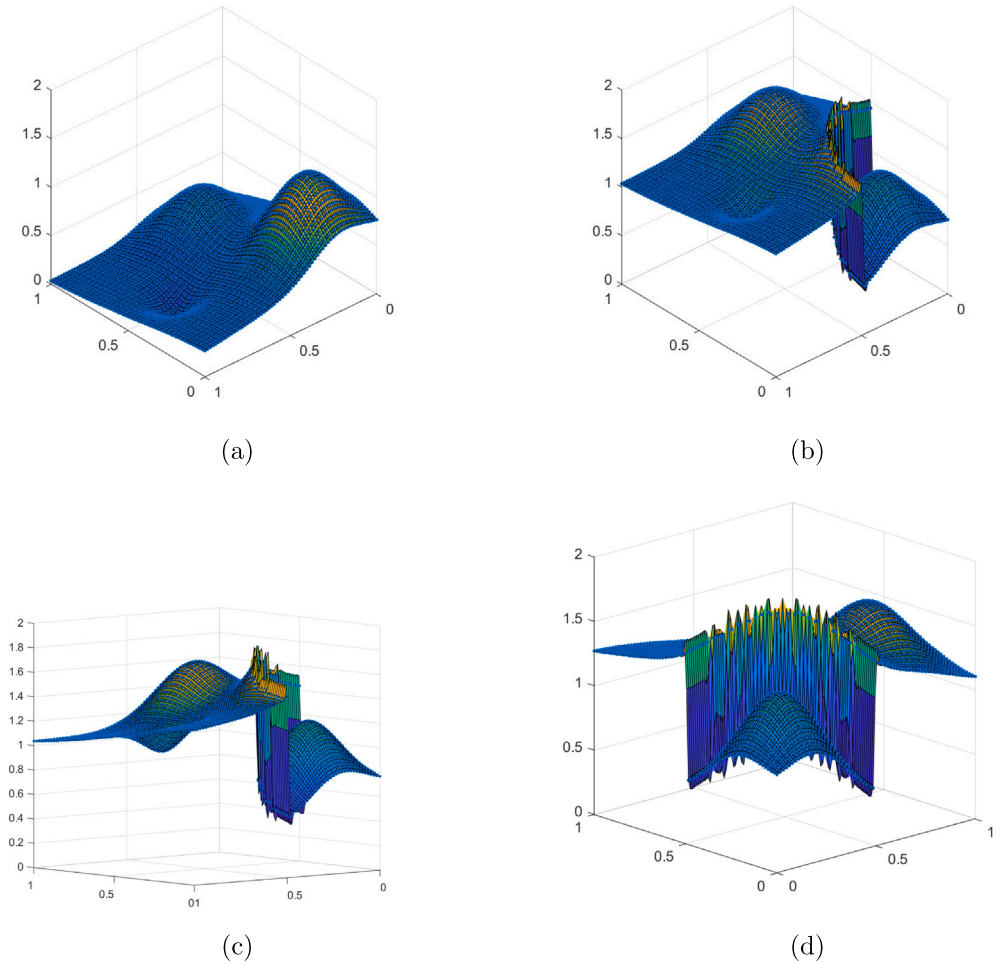

Fig. 1. (a) Approximation to Franke's function using PUM, (b), (c), (d) Approximation to function f_1, Eq. (2) using PUM.

using the PU method with data points located at $X = \{(i/64, j/64) : i, j = 0, \ldots, 64\}$. However, these methods yield inadequate results when the data originates from functions with discontinuities. See for example Fig. 1(b), (c) and (d), where we present the result obtained when applying the PUM to the piecewise smooth function:

$$f_1(x, y) = \begin{cases} 1 + f(x, y), & x^2 + y^2 - 0.5^2 \geq 0, \\ f(x, y), & x^2 + y^2 - 0.5^2 < 0. \end{cases} \tag{2}$$

It is clear that we can observe some oscillations close to the discontinuities. These problems can also be found in more advanced implementations of the PUM method, for example the one presented in [6]. In order to avoid these undesirable effects, constructions inspired by those presented in [7,8] can be applied to approximate this kind of data. Thus, our aim in this work is to reformulate the classical PUM to make the weight functions used depend on the smoothness of the data. The objective is to mitigate oscillations and smearing around discontinuities. To reach this objective, the basic idea is to modify the weights of the partition of unity by replacing them with non-linear ones that depend on the data F. This way, when a discontinuity crosses a subdomain, the approximation is not considered. It is not difficult to prove that the order of accuracy is conserved in the entire domain, except in a band close to the discontinuity. As mentioned in the abstract, as far as we know, this is the first time that such a non-linear PUM method is presented.

The paper is divided in the following sections: In Section 2, we start by reviewing the RBF interpolation and showing some theorems where the error bounds are estimated, Section 3 is devoted to introduce the PUM and to prove the theoretical results related to the order of accuracy. Afterwards, in Section 4, we review WENO method from the PUM perspective, identifying each component of WENO with its corresponding component in PUM. The non-linear method is presented in Section 5, where we explain its theoretical properties and its implementation. Some numerical results are presented in Section 6, and some conclusions and future work are outlined in the last section.

2. Radial basis function interpolation

There is extensive literature on the RBF interpolation method (see e.g. [1–3]), we will principally employ the notation presented in [2,9]. We consider a set of scattered data $X = \{\mathbf{x}_i\}_{i=1}^N$ in a bounded domain $\Omega \subset \mathbb{R}^n$, with an associated set of data values $F = \{f_i := f(\mathbf{x}_i)\}_{i=1}^N$ which are the evaluations of certain unknown function $f : \Omega \to \mathbb{R}$ at the nodes $\mathbf{x}_i$. The scattered data interpolation problem is the following: to find an operator, $\mathcal{I}(f) : \Omega \to \mathbb{R}$, such that:

$$\mathcal{I}(f)(\mathbf{x}_i) = f(\mathbf{x}_i), \quad i = 1, \dots, N.$$

Ideally, we aim to minimize the distance between $\mathcal{I}(f)$ and f in a specified norm. First, we define a radial function $\Phi : \Omega \to \mathbb{R}$ when there exists a univariate function $\phi : [0, \infty[\to \mathbb{R}$ such that

$$\Phi(\mathbf{x}) = \phi(\|\mathbf{x}\|_2), \quad \forall\, \mathbf{x} \in \Omega,$$

with $\|\cdot\|_2$ the Euclidian norm in $\mathbb{R}^n$. Now, we denote as $\Phi_i(\mathbf{x}) = \phi(\|\mathbf{x} - \mathbf{x}_i\|_2)$, $i = 1, \dots, N$ and use a radial basis function expansion to formulate the problem:

$$\mathcal{I}(f)(\mathbf{x}) = \sum_{i=1}^N c_i \Phi_i(\mathbf{x}) = \sum_{i=1}^N c_i \phi(\|\mathbf{x} - \mathbf{x}_i\|_2), \quad \mathbf{x} \in \Omega, \tag{3}$$

being c_i the coefficients that solve the interpolation system $A\mathbf{c} = \mathbf{f}$ with

$$A = \begin{bmatrix} \phi(\|\mathbf{x}_1 - \mathbf{x}_1\|_2) & \phi(\|\mathbf{x}_1 - \mathbf{x}_2\|_2) & \cdots & \phi(\|\mathbf{x}_1 - \mathbf{x}_N\|_2) \\ \phi(\|\mathbf{x}_2 - \mathbf{x}_1\|_2) & \phi(\|\mathbf{x}_2 - \mathbf{x}_2\|_2) & \cdots & \phi(\|\mathbf{x}_2 - \mathbf{x}_N\|_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi(\|\mathbf{x}_N - \mathbf{x}_1\|_2) & \phi(\|\mathbf{x}_N - \mathbf{x}_2\|_2) & \cdots & \phi(\|\mathbf{x}_N - \mathbf{x}_N\|_2) \end{bmatrix}, \ \mathbf{c} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{bmatrix}, \ \mathbf{f} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{bmatrix}.$$

It is clear that A is symmetric, and certain conditions on ϕ must be imposed to ensure that the matrix A is positive definite, thereby guaranteeing a unique solution to the system. The following definition is useful for this purpose (see [2]).

Definition 1. [2] A real continuous function $\Phi : \mathbb{R}^n \to \mathbb{R}$ is called positive definite on $\mathbb{R}^n$ if

$$\sum_{j=1}^N \sum_{i=1}^N c_j c_i \Phi_i(\mathbf{x}_j) \geq 0 \tag{4}$$

for any N pairwise different points $\{x_i\}_{i=1}^N \subset \mathbb{R}^n$ and $\mathbf{c} = [c_1, \dots, c_N] \in \mathbb{R}^N$. It is strictly positive definite on $\mathbb{R}^n$ if the form (4) is zero only for $\mathbf{c} = \mathbf{0}$.

In the literature we can find many examples of functions ϕ which produce strictly positive definite functions, such as the Gaussian function $\phi_G(\|\mathbf{x}\|_2) = e^{-\|\mathbf{x}\|_2^2}$, that is also C^∞. Another example are Wendland functions, defined as

$$\begin{aligned} \phi_{W_0}(\|\mathbf{x}\|_2) &= (1 - \|\mathbf{x}\|_2)_+^2, \\ \phi_{W_2}(\|\mathbf{x}\|_2) &= (1 - \|\mathbf{x}\|_2)_+^4 (4\|\mathbf{x}\|_2 + 1), \\ \phi_{W_4}(\|\mathbf{x}\|_2) &= (1 - \|\mathbf{x}\|_2)_+^6 (35\|\mathbf{x}\|_2^2 + 18\|\mathbf{x}\|_2 + 3), \end{aligned} \tag{5}$$

based on the cutoff function

$$(x)_+ = \begin{cases} x & \text{for } x \geq 0, \\ 0 & \text{for } x < 0, \end{cases}$$

which are C^0, C^2 and C^4 respectively. Matérn functions are another example, that are defined as

$$\begin{aligned} \phi_{M_0}(\|\mathbf{x}\|_2) &= e^{-\|\mathbf{x}\|_2}, \\ \phi_{M_2}(\|\mathbf{x}\|_2) &= (1 + \|\mathbf{x}\|_2) e^{-\|\mathbf{x}\|_2}, \\ \phi_{M_4}(\|\mathbf{x}\|_2) &= \left(3 + 3\|\mathbf{x}\|_2 + \|\mathbf{x}\|_2^2\right) e^{-\|\mathbf{x}\|_2}, \end{aligned} \tag{6}$$

satisfying that $\phi_{M_k} \in C^k$, $k = 0, 2, 4$. Other classes of functions, such as generalized inverse multiquadrics or Poisson radial functions, also produce strictly positive definite functions, see [2] for more details.

The first characteristic of this interpolant, Eq. (3), is the smoothness, as it is a linear combination of smooth functions, Φ_i. Consequently, the continuity of the operator only depends on the continuity of the function $\phi(\|\cdot\|_2)$. Secondly, the error estimates and the order of accuracy for the interpolation process have been studied extensively (see [1–3] and the references cited therein for further details). To introduce the results, we need to present three relevant concepts: First of all, we define the fill distance, which measures the extent to which the data in the set X adequately cover the domain Ω, as

$$h = h_{X,\Omega} = \sup_{\mathbf{x} \in \Omega} \min_{\mathbf{x}_j \in X} \|\mathbf{x} - \mathbf{x}_j\|_2.$$

Following this, we refer to the space (see [2,5])

$$C_\nu^{2k}(\mathbb{R}^n) = \{f \in C^{2k} : D^\beta f(\mathbf{x}) = O(\|\mathbf{x}\|_2^\nu), \text{ as } \|\mathbf{x}\|_2 \to 0 \text{ if } |\beta| = 2k\},$$

and finally, we state that a region $\Omega \subseteq \mathbb{R}^n$ satisfies **an interior cone condition**, [2,3], if there exists an angle $\theta \in (0, \pi/2)$ and a radius $r > 0$ such that for every $\mathbf{x} \in \Omega$ there exists a unit vector $\boldsymbol{\xi}(\mathbf{x})$ such that the cone

$$C = \{\mathbf{x} + \lambda \mathbf{y} : \mathbf{y} \in \mathbb{R}^n, \ \|\mathbf{y}\|_2 = 1, \ \mathbf{y}^T \boldsymbol{\xi}(\mathbf{x}) \geq \cos\theta, \ \lambda \in [0, r]\}$$

is contained in Ω. The following lemma presented by Wendland in [3] indicates that a ball always satisfies an interior cone condition. This result will be relevant in our numerical experiments when we construct a particular partition of unity method.

Lemma 2.1. *Every ball with radius $\delta > 0$ satisfies an interior cone condition with radius $\delta > 0$ and angle $\theta = \pi/3$.*

With these three elements: the fill distance, the space $C_v^{2k}(\mathbb{R}^n)$ and the interior cone condition, we review the following result.

Theorem 2.2. *[2] Suppose $\Omega \subset \mathbb{R}^n$ is bounded and satisfies an interior cone condition. Suppose $\Phi \in C_v^{2k}(\mathbb{R}^n)$ strictly positive definite. Then, the error between $f \in \mathcal{N}_\Phi(\Omega) := span\{\Phi(\cdot - \mathbf{x}), \mathbf{x} \in \Omega\}$ and its interpolant $\mathcal{I}(f)$ defined in Eq. (3) can be bounded by*

$$|f(\mathbf{x}) - \mathcal{I}(f)(\mathbf{x})| \leq C h^{k+\frac{v}{2}} \|f\|_{\mathcal{N}_\Phi(\Omega)}$$

for all $\mathbf{x} \in \Omega$, h is sufficiently small and where C is a constant independent of $\mathbf{x}$, f and Φ.

In this work, we focus on studying the differences between the function and the interpolator. Error estimates for the derivatives are proved in [2,3]. To conclude this review section, we introduce the following results for Matérn function, Eq. (6), as they will be used in the numerical experiments. For this purpose, we need to define the Sobolev spaces $W_2^m(\mathbb{R}^n)$ introduced in [2] as

$$W_2^m(\Omega) = \{f \in L_2(\Omega) \cap C(\Omega) : D^\beta f \in L_2(\Omega) \ \text{ for all } \ |\beta| \leq m, \ \beta \in \mathbb{N}^n\}.$$

Proposition 2.3. *[2] For the strictly positive Matérn functions ϕ_{M_k}, $k = 0, 2, 4$, Eq. (6), and $\Omega \subseteq \mathbb{R}^2$ is satisfied that*

$$|f(\mathbf{x}) - \mathcal{I}_{M_k}(f)(\mathbf{x})| \leq C h^{\frac{k+1}{2}} \|f\|_{W_2^{(k+3)/2}(\Omega)}$$

for all $\mathbf{x} \in \Omega$, h is sufficiently small, and $f \in W_2^{(k+3)/2}(\Omega)$.

Next subsection is devoted to introduce the partition of unity method for radial basis functions, which is a well-known approximation method that is satisfactorily used when the number of data points is very large.

3. Partition of unity approximation

The partition of unity method (PUM) is proposed to facilitate efficient computation using meshfree approximation methods. The approach is straightforward: it involves decomposing the large problem into smaller subproblems while preserving the order of accuracy. Accordingly, using the same notation as above, let $\Omega \subset \mathbb{R}^n$ represent our open and bounded domain, which we partition into M subdomains Ω_j such that [5,10]:

- The domain is contained in the union of the subdomains, i.e.
$$\Omega \subseteq \bigcup_{j=1}^{M} \Omega_j.$$
- For every $\mathbf{x} \in \Omega$ the number of subdomains Ω_j with $\mathbf{x} \in \Omega_j$ is bounded by a global constant K.
- There exists a constant $C_r > 0$ and an angle $\theta \in (0, \pi/2)$ such that every patch $\Omega_j \cap \Omega$ satisfies an interior cone condition with angle θ and radius $r = C_r h_{X,\Omega}$.
- The local fill distance h_{X_j,Ω_j} with $X_j = \Omega_j \cup X$ is uniformly bounded by the global fill distance $h_{X,\Omega}$.

This is called a regular covering for (Ω, X), [3]. For this covering, we choose a partition of unity, i.e. a family of compactly supported, non-negative, continuous functions, $w_j : \Omega_j \to \mathbb{R}$ with

$$\sum_{j=1}^{M} w_j(\mathbf{x}) = 1 \ \forall \ \mathbf{x} \in \Omega, \quad \text{supp}(w_j) \subseteq \Omega_j,$$

and define the global interpolator as:

$$\mathcal{I}_{\text{PUM}}(f)(\mathbf{x}) = \sum_{j=1}^{M} w_j(\mathbf{x}) \mathcal{I}_j(f)(\mathbf{x}), \tag{7}$$

where $\mathcal{I}_j(f) : \Omega_j \to \mathbb{R}$ is a local RBF interpolant on each subdomain Ω_j.

In [2] some technical conditions over the partition of unity functions are imposed to establish error bounds. We assume that the partition of unity functions is k-stable. This is, each $w_j \in C^k(\mathbb{R}^n)$ and for every multi-index $\boldsymbol{\alpha}$ with $|\boldsymbol{\alpha}| \leq k$ the following inequality is satisfied:

$$\|D^{\boldsymbol{\alpha}} w_j\|_{\infty,\Omega_j} \leq \frac{C_{\boldsymbol{\alpha}}}{\delta_j^{|\boldsymbol{\alpha}|}},$$

where $C_{\boldsymbol{\alpha}}$ is some positive constant and δ_j is the diameter of Ω_j, i.e. $\delta_j = \sup_{\mathbf{x},\mathbf{y}\in\Omega_j} \|\mathbf{x}-\mathbf{y}\|_2$. With these restrictions over the covering and the partition of unity function, we show the following result, [2].

Theorem 3.1. *Suppose $\Omega \subseteq \mathbb{R}^n$ is open and bounded, and let $X = \{\mathbf{x}_i\}_{i=1}^N \subseteq \Omega$. Let $\Phi \in C_\nu^{2k}(\mathbb{R}^n)$ be strictly positive definite. Let $\{\Omega_j\}_{j=1}^M$ be a regular covering for (Ω, X) and let $\{w_j\}_{j=1}^M$ be k-stable for $\{\Omega_j\}_{j=1}^M$. Then the error between $f \in \mathcal{N}_\Phi(\Omega)$ and its partition of unity interpolant, Eq. (7), can be bounded by*

$$|f(\mathbf{x}) - \mathcal{I}_{PUM}(f)(\mathbf{x})| \leq C h^{k+\frac{\nu}{2}} \|f\|_{\mathcal{N}_\Phi(\Omega)},$$

for all $\mathbf{x} \in \Omega$.

It is clear that the order of accuracy does not vary using PUM or RBF method, as we can see in Theorems 2.2 and 3.1. A simple way, suggested in [2,5], to construct PUM is to consider Shepard's approximation, i.e.

$$w_j(\mathbf{x}) = \frac{\varphi_j(\mathbf{x})}{\sum_{k=1}^M \varphi_k(\mathbf{x})}, \quad j = 1, \dots, M, \tag{8}$$

where φ_j are compactly supported, with their support contained in Ω_j, and can be chosen, for example, as Wendland functions, Eq. (5).

4. Reviewing weighted essentially non-oscillatory algorithm as a PUM

The WENO scheme has gained widespread recognition as an effective method for approximating data values, especially in contexts where discontinuities are prevalent (see e.g. [8]). Initially introduced for approximating the solution of hyperbolic partial differential equations, the WENO approach is structured to achieve high-order accuracy while mitigating the unwanted oscillations that typically occur near discontinuities [11]. This is accomplished through the adaptive assignment of weights that prioritize smoother portions of the data and diminish the contribution of stencils that intersect discontinuities.

In what follows, we review the WENO-2r method in the point-values context (see [12,13]) relating each of its components to those corresponding to the PUM. The analogy is done for the algorithms in one variable, but it can be generalized to several variables [14]. In this case, let $\{x_i\}_{i=0}^J$ be a uniform partition of the interval $[a,b]$ in J subintervals, $x_i = a + i \cdot h$, $\quad h = \frac{b-a}{J}$, and let $\{f_i\}_{i=0}^J$ be the point-values discretization of an unknown function f at the nodes x_i, to interpolate at $x \in (x_{i-1}, x_i)$. WENO-2r algorithm uses the stencil $X = \{x_{i-r}, \dots, x_{i+r-1}\}$, that is composed of $N = 2r$ nodes, we denote it by $\tilde{I}(f)$. We can divide the stencil X in r substencils of $r+1$ nodes, $S_j = \{x_{i-r+j}, \dots, x_{i+j}\}$, $j = 0, \dots, r-1$ and prove that there exist $\tilde{w}_j : [x_{i-1}, x_i] \to \mathbb{R}$, $j = 0, \dots, r-1$, [14], called optimal weights, such that:

$$\tilde{I}(f)(x) = \sum_{j=0}^{r-1} \tilde{w}_j(x) \tilde{I}_j(f)(x),$$

with $\tilde{w}_j(x) > 0$ and $\sum_{j=0}^{r-1} \tilde{w}_j(x) = 1$. This is a PU-like method, with $\Omega_j = [x_{i-r+j}, x_{i+j}]$ and $\tilde{I}_j(f)$ the Lagrange interpolatory polynomial using the nodes S_j. Now, the WENO method consists in replacing $\tilde{w}_j(x)$ by non-linear weights $\tilde{\mathcal{W}}_j(x)$, such that

$$\mathcal{I}_{\text{WENO}}(f)(x) = \sum_{j=0}^{r-1} \tilde{\mathcal{W}}_j(x) \tilde{I}_j(f)(x), \tag{9}$$

where the non-linear weights are defined as:

$$\tilde{\mathcal{W}}_j(x) = \frac{\tilde{\alpha}_j(x)}{\sum_{k=0}^{r-1} \tilde{\alpha}_k(x)}, \text{ with } \tilde{\alpha}_j(x) = \frac{\tilde{w}_j(x)}{(\epsilon + \tilde{I}_j)^t}, \tag{10}$$

being $\tilde{I}_j$ smoothness indicators which indicate if a discontinuity crosses Ω_j, ϵ is a small positive parameter to avoid divisions by zero, and t is typically a constant to enhance accuracy in smooth regions. The idea is to make a non-linear version of the interpolator $\mathcal{I}_{\text{PUM}}$, Eq. (7), (NL-PUM) using the ideas of WENO method. The key is to define an adequate smoothness indicator for our framework.

5. Non-linear partition of unity method

As we mentioned in the previous section, Section 4, we construct a non-linear PUM starting with the interpolator $\mathcal{I}_{\text{PUM}}(f)$, Eq. (7):

$$\mathcal{I}_{\text{PUM}}(f)(\mathbf{x}) = \sum_{j=1}^{M} w_j(\mathbf{x}) \mathcal{I}_j(f)(\mathbf{x}),$$

replacing the weight functions w_j described in Eq. (8) by non-linear ones in the same way as in Eq. (10), i.e.

$$\mathcal{W}_j(\mathbf{x}) = \frac{\alpha_j(\mathbf{x})}{\sum_{k=1}^{M} \alpha_k(\mathbf{x})}, \text{ with } \alpha_j(\mathbf{x}) = \gamma_j \varphi_j(\mathbf{x}) = \frac{\varphi_j(\mathbf{x})}{(\epsilon + I_j)^t},$$

where $0 < \gamma_j = (\epsilon + I_j)^{-t}$ is a positive constant, and φ_j are compactly supported functions with support on Ω_j. Note that the new functions α_j are compactly supported and, if we choose φ_j as Wendland functions, then α_j are Wendland functions too. The constant $\epsilon = 10^{-14}$ is set to avoid non-zero denominators. The parameter t is relevant to correctly detect the discontinuities and we choose $t = 6$. Finally, I_j, $j = 1, \dots, M$ are the smoothness indicators. To construct them, we consider

$$S_j = \Omega_j \cap X = \{\mathbf{x}_{j_i} \in \Omega_j \cap X : i = 1, \dots, N_j\},$$

with $N_j > 3$, and the linear least square problem

$$p_j = \underset{p \in \Pi_1(\mathbb{R}^2)}{\arg\min} \sum_{i=1}^{N_j} (f(\mathbf{x}_{j_i}) - p(\mathbf{x}_{j_i}))^2,$$

where $\Pi_1(\mathbb{R}^2)$ is the set of polynomials of degree less than or equal to one, and define

$$I_j = \frac{1}{N_j} \sum_{i=1}^{N_j} |f(\mathbf{x}_{j_i}) - p_j(\mathbf{x}_{j_i})|. \tag{11}$$

Then I_j satisfies the conditions:

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

$$I_j = O(h^2) \text{ if } f \text{ is smooth in } \Omega_j. \tag{12}$$

- When a discontinuity crosses Ω_j then:

$$I_j \nrightarrow 0 \text{ as } h \to 0.$$

Therefore, the new non-linear operator will be:

$$\mathcal{I}_{\text{NL-PUM}}(f)(\mathbf{x}) = \sum_{j=1}^{M} \mathcal{W}_j(\mathbf{x}) \mathcal{I}_j(f)(\mathbf{x}). \tag{13}$$

The key to this new method is that it gives a greater weight to the data placed in smooth regions, while neglecting data located near the discontinuity, as this data is multiplied by a weight of order $O(h^{2t})$. In the next subsection we will analyze the properties of the interpolator.

5.1. Properties of the new NL-PUM

As the new interpolator, Eq. (13), is defined in the same way as PUM, multiplying the compactly supported functions by constants (that are dependent on the data $F = \{f_i\}_{i=1}^N$) then the smoothness of the new operator is similar to PUM, in the sense that its smoothness is solely determined by the smoothness of the functions $\varphi(\|\cdot\|_2)$ and $\phi(\|\cdot\|_2)$.

Secondly, we determine the order of accuracy when we apply $\mathcal{I}_{\text{NL-PUM}}(f)$ to data that comes from a function which is piecewise continuous, i.e. we suppose an open and bounded domain $\Omega \subset \mathbb{R}^n$ and consider a curve $\zeta : \mathbb{R}^n \to \mathbb{R}$, $\zeta \in C^1$ and the sets

$$\Gamma = \{\mathbf{x} \in \Omega : \zeta(\mathbf{x}) = 0\}, \quad \Omega^+ = \{\mathbf{x} \in \Omega : \zeta(\mathbf{x}) \geq 0\}, \quad \Omega^- = \Omega \setminus \Omega^+$$

that is to say that Ω is divided into two subsets, $\Omega = \Omega^+ \cup \Omega^-$. Let us assume that our data originates from a function of the type:

$$f(\mathbf{x}) = \begin{cases} f_+(\mathbf{x}), & \mathbf{x} \in \Omega_+, \\ f_-(\mathbf{x}), & \mathbf{x} \in \Omega_-, \end{cases}$$

with $f_\pm \in C^k(\overline{\Omega_\pm})$, $k \in \mathbb{N}$, but

$$\lim_{\mathbf{x} \to \mathbf{x}_0 \in \Gamma} f_-(\mathbf{x}) \neq f_+(\mathbf{x}_0),$$

then our scattered data $X = \{\mathbf{x}_i\}_{i=1}^N$ are divided in two $X_\pm = X \cap \Omega_\pm$. We select our partition $\{\Omega_j\}_{j=1}^M$ as in Section 3 and define patches that are contaminated by a discontinuity as follows.

Definition 2 (*Contaminated Patches*)**.** Using the notation presented above, $\Omega_k \in \{\Omega_j\}_{j=1}^M$ is a contaminated patch if there exist $\Omega_k \cap X_+ \neq \emptyset$ and $\Omega_k \cap X_- \neq \emptyset$.

We determine if a subset $\Omega_k \in \{\Omega_j\}_{j=1}^M$ is a contaminated patch if $I_k > h = h_{X,\Omega}$. Also, the points close to the discontinuity will be marked using the following definition.

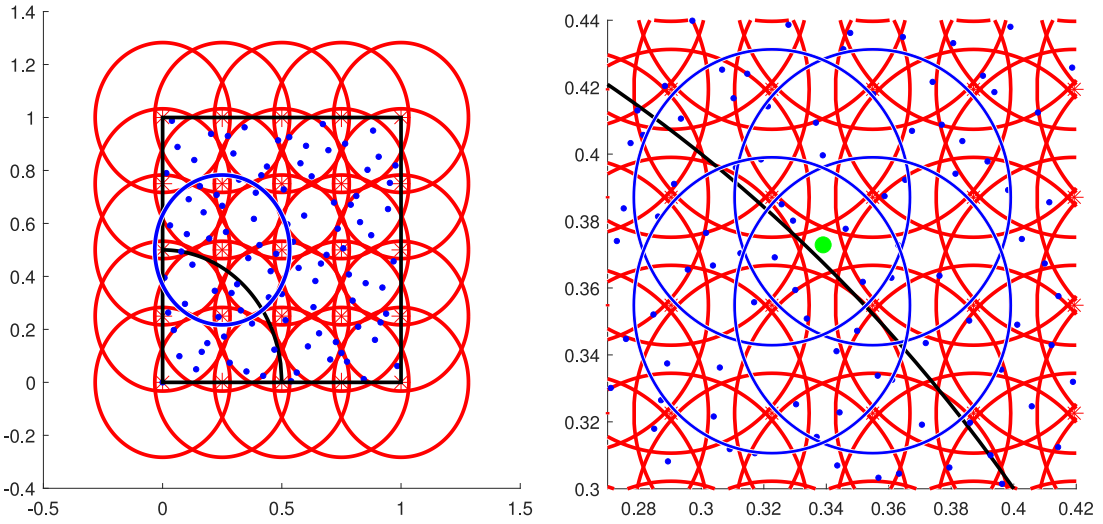

Fig. 2. Black square: frontier of Ω, red circumferences: frontiers of the patches, black curve: discontinuity, blue circumferences: contaminated patches, blue points: data, red stars: centers of the patches, green point (right): discontinuity point.

Definition 3 (*Discontinuity Points*)**.** Using the notation above presented, let $\mathbf{x} \in \Omega$ be a point of the domain, $\varepsilon > 0$ be a fixed threshold and

$$S(\mathbf{x}) = \{j \in \{1, \dots, M\} : \mathbf{x} \in \text{supp}(\Omega_j),\ \varphi_j(\mathbf{x}) > \varepsilon\},$$
$$\tilde{S}(\mathbf{x}) = \{j \in \{1, \dots, M\} : \mathbf{x} \in \text{supp}(\Omega_j),\ \Omega_k \cap X_+ \neq \emptyset,\ \Omega_k \cap X_- \neq \emptyset,\ \varphi_j(\mathbf{x}) > \varepsilon\},$$

then $\mathbf{x}$ is a discontinuity point if

$$S(\mathbf{x}) = \tilde{S}(\mathbf{x}).$$

This definition means that a point is a discontinuity point only if all the patches used to interpolate at that point are contaminated, Definition 2. Last definition, Definition 3, is quite relevant to design the new algorithm. This is due to the fact that, at the discontinuity points, the strategy to interpolate has to be changed due to the oscillations produced by the RBF method close to the discontinuity curve. Hence, the most problematic case is the one in which all the patches containing the point where the interpolation is desired, $\mathbf{x}$, are contaminated, except for one whose weight, $\varphi_j(\mathbf{x})$, is close to zero. For this reason, we impose the threshold condition in the previous definition, Definition 3. The value ε is also relevant for determining the parameter t, since if, for some subindex j_0, a discontinuity intersects Ω_{j_0} and $\varphi_{j_0}(\mathbf{x}) > \varepsilon$, albeit by a very small margin, it is necessary to ensure that the weight assigned to this patch is sufficiently large. Consequently, the parameter t must also be chosen to be high enough depending of h, Eq. (12). In Fig. 2 we can see a picture with the representation of the two previous concepts, Definitions 2 and 3. We represent the data points using blue dots, the patch boundaries with red circles, the discontinuity curve and the domain boundary with black lines, and the patch centers with red stars. In the figure on the left, the contaminated patch is indicated with a blue circle. In the figure on the right, the point at which the function is to be approximated is marked in green, while the patches used for interpolation are highlighted in blue. It can be observed that all the patches are contaminated, indicating that the green point is a discontinuity point.

With these definitions, the following result is straightforward based on Theorem 3.1, where an error bound is deduced dependent on the smoothness of the function Φ and its native space $\mathcal{N}_{\Phi}$.

Theorem 5.1. *Suppose $\Omega \subseteq \mathbb{R}^n$ is open and bounded, and let $X = \{\mathbf{x}_i\}_{i=1}^N \subseteq \Omega$. Let $\Phi \in C_v^{2k}(\mathbb{R}^n)$ be strictly positive definite. Let $\{\Omega_j\}_{j=1}^M$ be a regular covering for (Ω, X) and let $\{w_j\}_{j=1}^M$ be k-stable for $\{\Omega_j\}_{j=1}^M$. Then, if $\mathbf{x} \in \Omega_+$, it is not a discontinuity point, Def. 3, and $f_\pm \in \mathcal{N}_{\Phi}(\Omega_\pm)$, the error between f and its non-linear partition of unity interpolant, Eq. (13), with $t \geq k/2 + \nu/4$ can be bounded by*

$$|f(\mathbf{x}) - \mathcal{I}_{NL\text{-}PUM}(f)(\mathbf{x})| \leq C h^{k+\frac{\nu}{2}} \|f\|_{\mathcal{N}_{\Phi}(\Omega_+)}.$$

Analogously, if $\mathbf{x} \in \Omega_-$ and it is not a discontinuity point.

In other words, if $\mathbf{x} \in \Omega$ and there exists, at least, a patch Ω_k free of discontinuities, then the order of accuracy is conserved. Consequently, the discontinuity points have to be treated in a special manner. Also, if we suppose (without loss of generality) that

$$\lim_{\mathbf{x} \to \mathbf{x}_0 \in \Gamma} f_+(\mathbf{x}) < \lim_{\mathbf{x} \to \mathbf{x}_0 \in \Gamma} f_-(\mathbf{x}),$$

we cannot ensure that if $\tilde{\mathbf{x}} \in \Omega$ is a discontinuity point then

$$\lim_{\mathbf{x}\to\mathbf{x}_0\in\Gamma} f_+(\mathbf{x}) \leq \mathcal{I}_{\text{NL-PUM}}(f)(\tilde{\mathbf{x}}) \leq \lim_{\mathbf{x}\to\mathbf{x}_0\in\Gamma} f_-(\mathbf{x}).$$

For this reason, in order to avoid some oscillations of the interpolator $\mathcal{I}_{\text{NL-PUM}}(f)$ close to the discontinuities, we modify our algorithm and explain it in the next subsection.

5.2. On the implementation of the method. Avoiding oscillations close to the discontinuities

We compute our method using the implementation explained by Cavoretto et al. in [9], based on the partitioning data procedures proposed in [4]. Let $\{\hat{\mathbf{x}}_k\}_{k=1}^{L} \subset \Omega$ be a sample where we want to approximate the function f, we start with the covering of the domain Ω, consider some centers $P = \{\bar{\mathbf{x}}_j\}_{j=1}^{M} \subset \Omega$, fix a radius δ and construct the subdomains $\Omega_j = B(\bar{\mathbf{x}}_j, \delta)$ where $B(\bar{\mathbf{x}}_j, \delta)$ is a ball of center $\bar{\mathbf{x}}_j$ and radius δ. It is important to remark, as noted in [2,9], that if a nearby node distribution is chosen, M is an adequate number of PU subdomains if

$$\frac{N}{M} \approx 2^n,$$

then, the radius has to satisfy the condition, [9]:

$$\delta \geq \frac{1}{M^{\frac{1}{n}}}. \tag{14}$$

At this point, we estimate the smoothness indicators I_j, Eq. (11) and mark the contaminated patches, Definition 2. The interpolation data located in each subdomain j are found using the partitioning data structures explained in [9]. After this, we estimate the non-linear weights at each point $\hat{\mathbf{x}}_k$ using the smoothness indicators calculated, and label the discontinuity points, we called them $\tilde{\mathbf{x}}_j$. We apply the classical Shepard's method only for these points, with the same compactly supported function used in PUM. In summary, we describe the algorithm in two passes:

1. We use the software designed in [9] and mark the discontinuity points, $\tilde{\mathbf{x}}$.
2. We apply the Shepard's method to these discontinuity points.

Due to the partitioning data structures introduced in [9], which allow us to organize data very efficiently across different subdomains, the modifications inserted to make the algorithm non-linear are not too costly in terms of computational efficiency.

6. Numerical experiments

In this section, we perform some tests in order to check two principal properties of the new non-linear algorithm, NL-PUM, in comparison with the classical one, PUM. We will conduct the experiments using a uniform grid and Halton's points as data. Firstly, in Section 6.1, we employ our method with a continuous function, in particular with Franke's function, Eq. (1), to verify the order of accuracy. Next subsection is devoted to analyze the elimination of the non-desired oscillations close to the discontinuities, we come back to the example presented in the introduction, Section 1, and show the powerful capabilities of the new algorithm for this type of functions. We take the same parameters as [9] in all the experiments, i.e., the number of patches in one direction is $\lfloor\sqrt{N}/2\rfloor$ and the radius $\delta = \sqrt{2/M}$.

6.1. Order of accuracy

We begin by analyzing the order of accuracy using the widely recognized Franke's function, Eq. (1), using as data points a regular grid defined by $X_{N=2^l+1} = \{(i/2^l, j/2^l) : i, j = 0, \ldots, 2^l\}$, and a set of $N = (2^l+1)^2$ Halton's scattered data points, [15]. We denote the fill distance by h_l, and the errors by $e_k^l = |f(\hat{\mathbf{x}}_k) - \mathcal{I}^l(\hat{\mathbf{x}}_k)|$, where $\{\hat{\mathbf{x}}_k\}_{k=1}^{60}$ is a regular grid in $[0,1]^2$. We define the maximum discrete ℓ^2 norms and their respective convergence rates as follows:

$$\text{MAE}_l = \max_{i=1,\ldots,60} e_i^l, \quad \text{RMSE}_l = \left(\frac{1}{60}\sum_{i=1}^{60}(e_i^l)^2\right)^{\frac{1}{2}},$$
$$r_l^\infty = \frac{\log(\text{MAE}_{l-1}/\text{MAE}_l)}{\log(h_{l-1}/h_l)}, \quad r_l^2 = \frac{\log(\text{RMSE}_{l-1}/\text{RMSE}_l)}{\log(h_{l-1}/h_l)}.$$

In the first example, we choose Matèrn function, ϕ_{M_2} for the RBF problems, and the Wendland C^2 function ϕ_{W_2}, for the partition of unity. We present the results in Table 1, and it is clear that they are very similar for both the linear and non-linear methods. The expected theoretical order of accuracy, Prop. 2.3, is improved for the two algorithms in this example. With respect to the errors, the NLPUM yields slightly better results when the data is placed on an uniform grid.

To finish this subsection, we repeat the experiment using the functions ϕ_{M_4} for the RBF method and ϕ_{W_4} for the partition of unity. The results are shown in Table 2. Again, we can see that the errors and orders of accuracy are very similar.

Table 1
Errors and rates using PUM and NLPUM for Franke's test function: two first columns evaluated at grid points and two final columns evaluated at Halton's points. For RBF problems the C^2 Matérn function ϕ_{M_2} has been used, and for partition of unity the Wendland function ϕ_{W_2} is used, i.e. $\varphi_j = \phi_{W_2}$, for all j.

Grid points					Halton's points			
	PUM		NLPUM		PUM		NLPUM	
l	MAE_l	r_l^∞	MAE_l	r_l^∞	MAE_l	r_l^∞	MAE_l	r_l^∞
4	7.5944e−03	–	2.8436e−01	–	1.4157e−02	–	2.6937e−02	–
5	1.6541e−03	2.1989	8.6890e−02	1.7105	2.6921e−03	2.1831	2.9567e−03	2.9058
6	3.7494e−04	2.1413	8.8016e−05	9.9472	9.1058e−04	2.3481	6.6469e−04	3.2331
7	1.1278e−04	1.7332	2.6376e−05	1.7385	2.0617e−04	1.9743	3.0466e−04	1.0369
8	2.8906e−05	1.9640	7.0429e−06	1.9050	6.0296e−05	1.4750	7.9830e−05	1.6068
	PUM		NLPUM		PUM		NLPUM	
l	RMSE_l	r_l^2	RMSE_l	r_l^2	RMSE_l	r_l^2	RMSE_l	r_l^2
4	1.3800e−03	–	9.6360e−02	–	6.8732e−03	–	9.6360e−02	–
5	2.2904e−04	2.5910	4.7798e−03	4.3334	3.3489e−04	2.2581	4.7798e−03	3.9740
6	5.8939e−05	1.9583	1.5566e−05	8.2624	3.8497e−05	2.7589	1.5566e−05	4.6860
7	1.8977e−05	1.6350	1.9141e−06	3.0236	5.9659e−06	1.8766	1.9141e−06	2.4782
8	4.0380e−06	2.2325	2.5281e−07	2.9206	7.4670e−07	1.7219	2.5281e−07	2.4933

Table 2
Errors and rates using PUM and NLPUM for Franke's test function: two first columns evaluated at grid points and two final columns evaluated at Halton's points. For RBF problems the C^4 Matérn function ϕ_{M_4} has been used, and for partition of unity the Wendland function ϕ_{W_4} is used, i.e. $\varphi_j = \phi_{W_4}$, for all j.

Grid points					Halton's points			
	PUM		NLPUM		PUM		NLPUM	
l	MAE_l	r_l^∞	MAE_l	r_l^∞	MAE_l	r_l^∞	MAE_l	r_l^∞
4	6.4397e−03	–	2.9103e−01	–	8.1704e−03	–	8.7701e−03	–
5	7.6400e−04	3.0754	9.0838e−02	1.6798	8.9459e−04	2.9091	1.1375e−03	2.6863
6	7.3358e−05	3.3805	2.0297e−05	12.1278	3.0378e−04	2.3397	1.6615e−04	4.1672
7	9.0336e−06	3.0216	1.6754e−06	3.5986	2.2676e−05	3.4490	4.5792e−05	1.7129
8	1.5305e−06	2.5613	2.5780e−07	2.7002	2.3915e−06	2.6987	1.0517e−05	1.7650
	PUM		NLPUM		PUM		NLPUM	
l	RMSE_l	r_l^2	RMSE_l	r_l^2	RMSE_l	r_l^2	RMSE_l	r_l^2
4	1.0412e−03	–	1.0168e−01	–	9.6168e−04	–	2.6447e−03	–
5	1.0970e−04	3.2465	4.7209e−03	4.4289	1.0395e−04	2.9260	1.2230e−04	4.0427
6	1.2027e−05	3.1893	2.7164e−06	10.7631	1.3923e−05	4.3550	6.5149e−06	6.3522
7	1.8291e−06	2.7170	1.5998e−07	4.0858	1.7160e−06	2.7826	6.6212e−07	3.0389
8	1.9361e−07	3.2400	1.0651e−08	3.9088	2.0360e−07	2.5573	5.6579e−08	2.9512

6.2. Avoiding discontinuities

In this subsection, we conduct a series of experiments to analyze the behavior of the new interpolator near discontinuities. In all the experiments the initial number of data points is $N = 65^2$.

We begin with a set of experiments in which the original domain $[0,1]^2$ is divided into two subdomains. We assume that the data originates from Franke's function f, as defined in Eq. (1), in one subdomain, and from $1 + f$ in the other, i.e., we define the functions:

$$f_q(x,y) = \begin{cases} 1 + f(x,y), & (x,y) \in \Omega^+, \\ f(x,y), & (x,y) \in \Omega^-, \end{cases}$$

where $\Gamma_q = \{(x,y) \in [0,1]^2 : \zeta_q(x,y) = 0\}$, $\Omega^+ = \{\mathbf{x} \in [0,1]^2 : \zeta_q(\mathbf{x}) \geq 0\}$, $\Omega^- = [0,1]^2 \setminus \Omega^+$. In this series of experiments we employ ϕ_{M_2} as function for RBF problems and $\varphi = \phi_{W_2}$ for the partition of unity.

Let us start with the example presented in the introduction, Section 1, where the discontinuity curve is the following

$$\zeta_1(x,y) = x^2 + y^2 - 0.5^2.$$

In this case, we plot in Fig. 3 the result of the experiment with $L = 120^2$ equidistant points in the square $[0,1]^2$. It is clear, Fig. 3 right column, that the NL-PUM presents an adequate behavior close to the discontinuities whereas the PUM produces some oscillations. When the figure is rotated (Fig. 3 rows 2 and 3), the differences become even more evident.

We also discuss the results when the discontinuities are horizontal, vertical and diagonal lines. For this, we computed a first example with:

$$\Gamma_2 = (\{0.5\} \times [0.5, 1]) \cup ([0, 0.5] \times \{0.5\}),$$

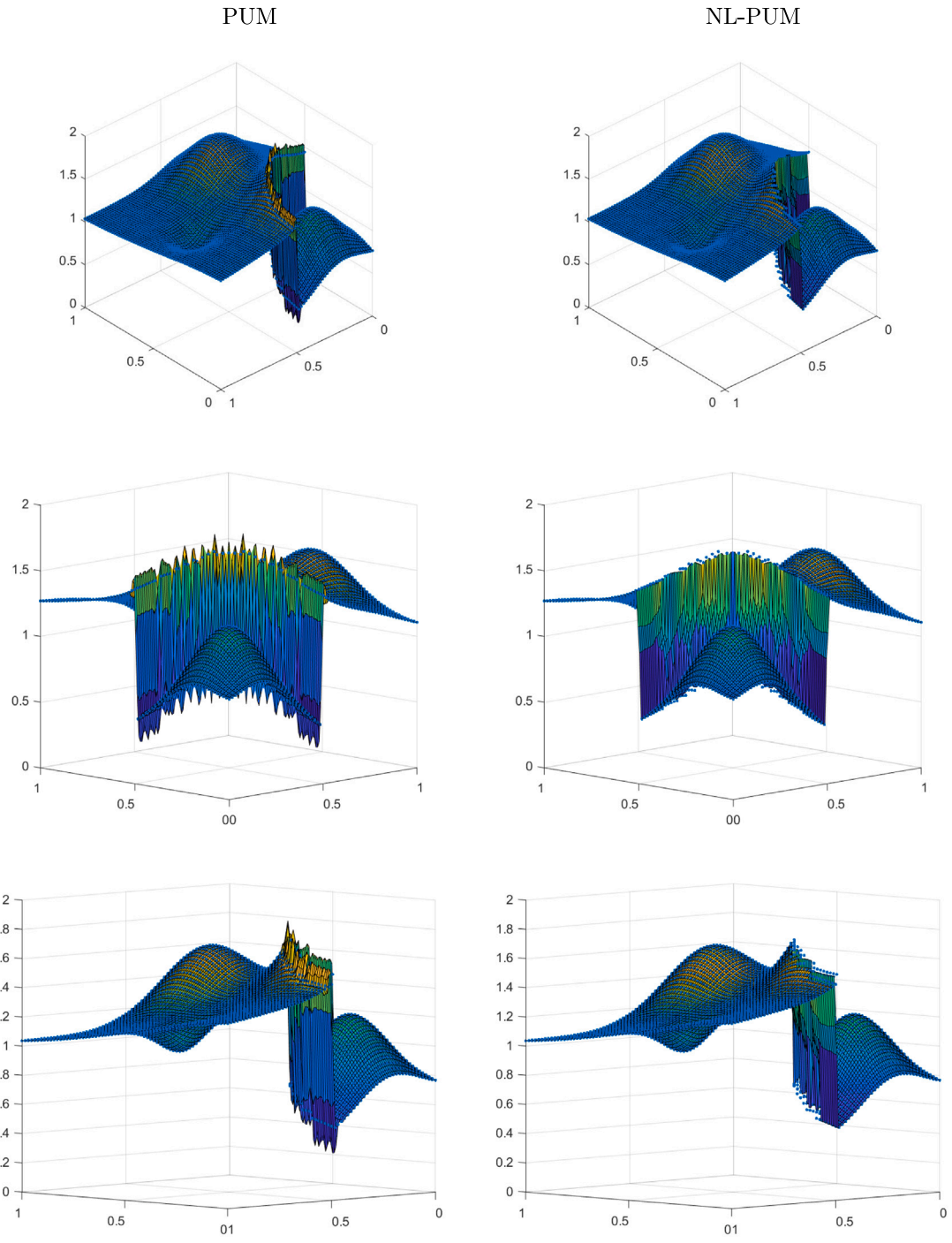

Fig. 3. Approximation of the function f_1 using PUM, and NL-PUM at grid points where we use ϕ_{M_2} for the RBF problems and ϕ_{W_2} for the partition of unity. Second and third rows are rotations of the plots presented in the first row.

and a second with

$$\zeta_3(x, y) = x + y - 1.$$

We illustrate the interpolators for the function f_2 in Fig. 4 and for f_3 in Fig. 5. In these cases we take $N = 65^2$ initial Halton's points and we approximate at $L = 120^2$ final points. Again, we can see that the new method avoids the oscillations.

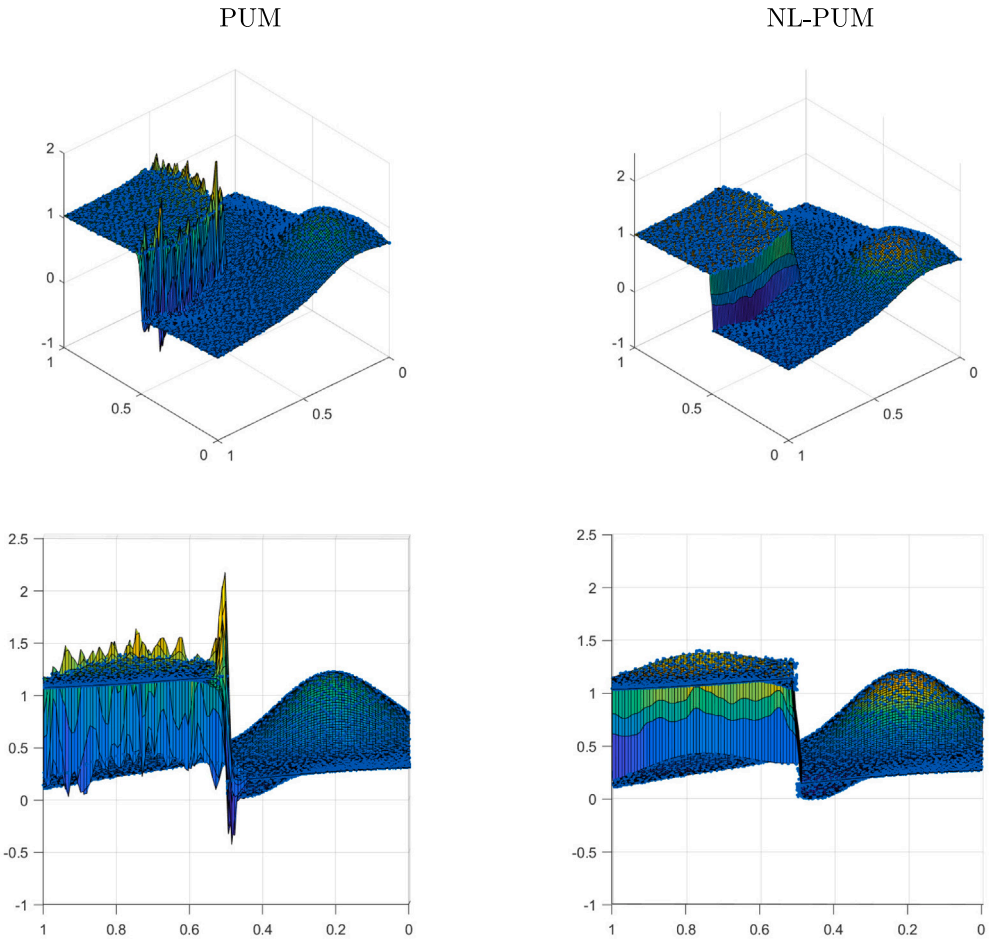

Fig. 4. Approximation to functions f_2 using PUM, and NL-PUM at Halton's points using for RBF problems ϕ_{M_2} and for partition of unity ϕ_{W_2}. The second row is a rotation of the plots presented in the first row.

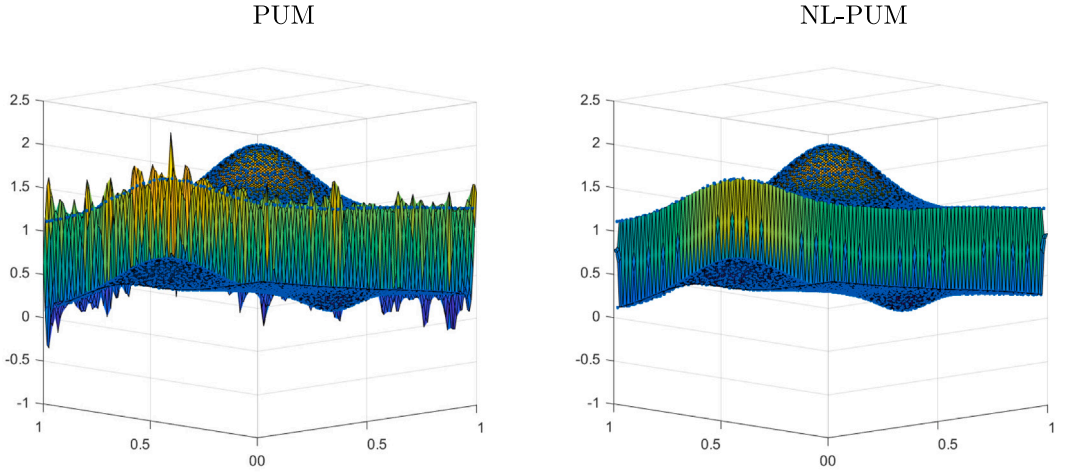

Fig. 5. Approximation to function f_3 using PUM, and NL-PUM at Halton's points using for RBF problems ϕ_{M_2} and for partition of unity ϕ_{W_2}.

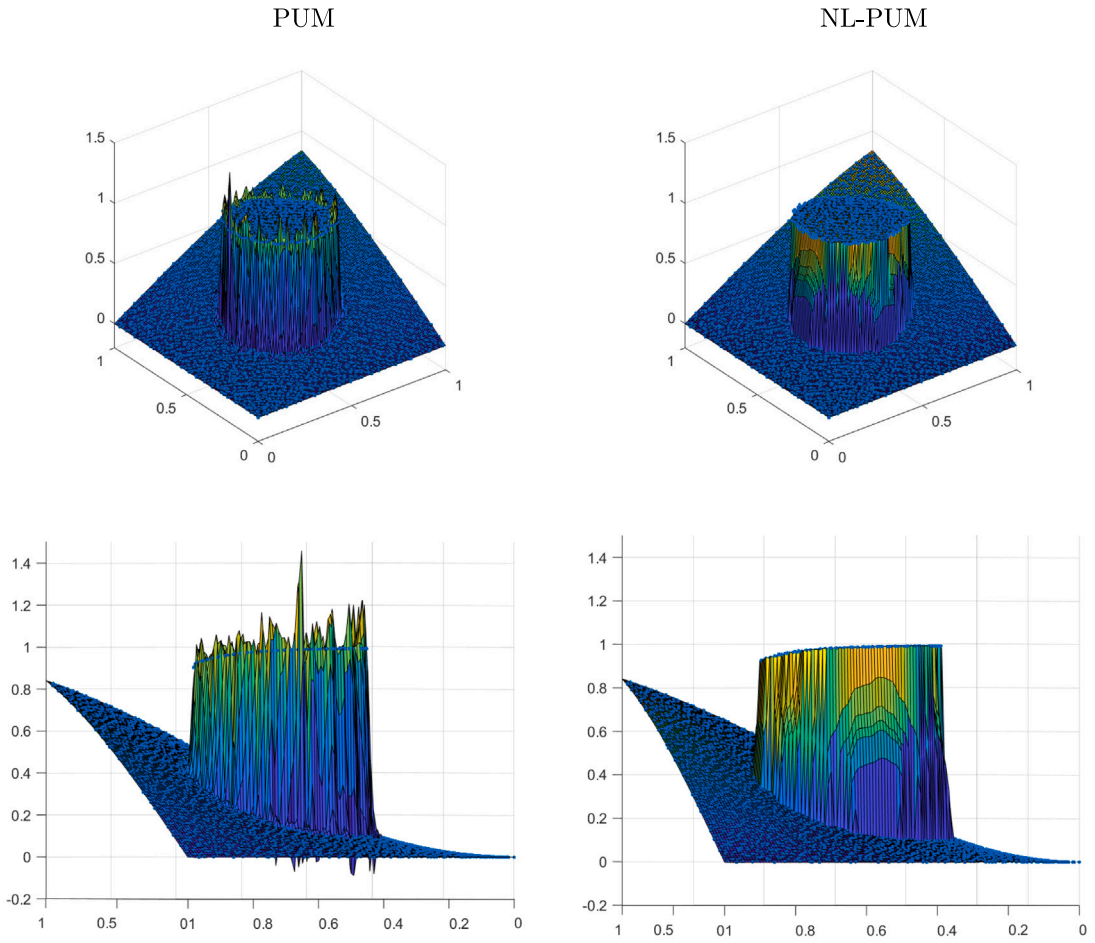

Fig. 6. Approximation to functions z using PUM, and NL-PUM at Halton's points using for RBF problems ϕ_{M_2} and for partition of unity ϕ_{W_2}. The second row is a rotation of the plots presented in the first row.

To end this subsection, we apply the algorithms to the same function employed in [16],

$$z(x,y) = \begin{cases} \sin(xy), & (x-0.5)^2 + (y-0.5)^2 \geq 0.25^2, \\ \cos(xy), & (x-0.5)^2 + (y-0.5)^2 < 0.25^2. \end{cases} \tag{15}$$

We plot the result using ϕ_{M_2} for RBF problems and ϕ_{W_2} for partition of unity in Fig. 6 using Halton's points. Finally, we illustrate the interpolants employing ϕ_{M_4} and ϕ_{W_4} in Fig. 7 using gridded data points in the two first rows, and Halton's points in the two last rows. The interpolators in both cases are very similar. Again, we can see that the new method improves the behavior close to the discontinuities. Furthermore, in Fig. 7 we perform the experiment for Halton and grided data points. The oscillations obtained when the classical PUM is applied are highly significant in contrast to the result obtained when the NL-PUM is employed.

7. Conclusions and future work

In this article, we have introduced a non-linear modification of the Partition of Unity Method (NL-PUM). We believe this is the first time such an approach has been presented in the literature. The design of the new method aims to mitigate the oscillations and smearing around discontinuity zones that the classical PUM method produces. Thus, the NL-PUM presented in this study marks a substantial improvement in PUM techniques, offering a robust solution to the numerical artifacts that appear when discontinuities are introduced in the datasets on which the classical PUM has been tested. The new method is founded on integrating Radial Basis Function (RBF) interpolation with Weighted Essentially Non-Oscillatory (WENO) techniques, improving the results obtained by traditional PUM algorithms close to the discontinuities. By dynamically adjusting weights through smoothness indicators, the NL-PUM selectively conserves accuracy in smooth regions while minimizing oscillations and smearing near discontinuities.

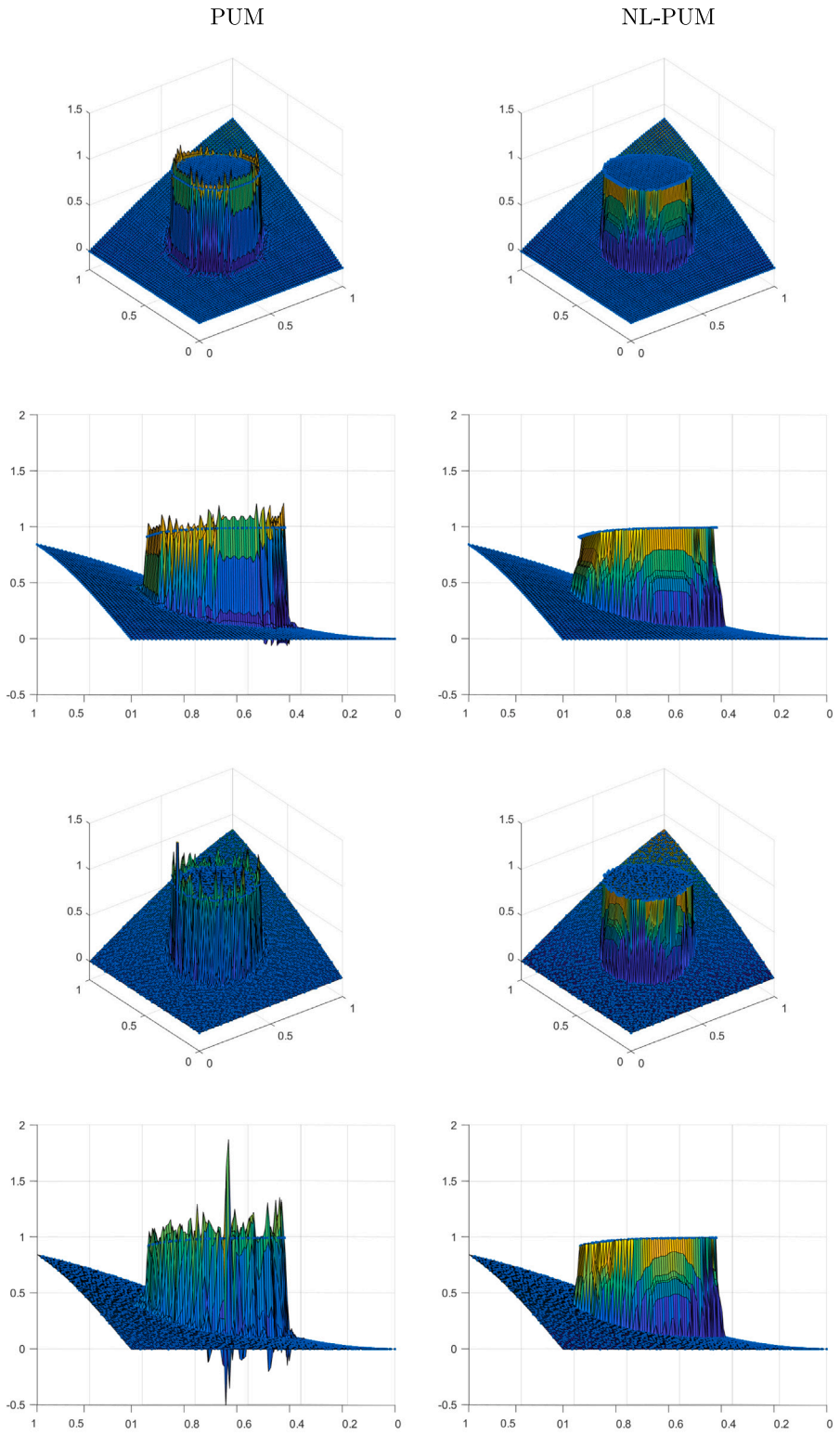

Fig. 7. Approximation to function z using PUM, and NL-PUM where we use ϕ_{M_3} for the RBF problems and ϕ_{W_4} for the partition of unity employing gridded data points (two first rows) and Halton's points (two last rows). The second and fourth rows are a rotation of the plots presented in the first row and third row respectively.

Numerical experiments confirmed that the NL-PUM maintains the order of accuracy expected from the traditional PUM method in continuous domains while exhibiting superior performance in discontinuous settings. This is achieved by identifying and treating *contaminated patches* near discontinuities, effectively eliminating oscillations that commonly arise in such scenarios. The method's ability to consistently produce reliable results, regardless of the orientation or complexity of discontinuities, emphasizes its robustness.

While the NL-PUM demonstrates promising capabilities, it also presents certain limitations, such as its reliance on well-tuned parameters. These factors highlight the need for further research to optimize and simplify the method. Future work could focus on automating parameter selection, extending the framework to higher-dimensional problems, and exploring its application in practical contexts such as image processing and fluid dynamics.

Data availability

No data was used for the research described in the article.

References

[1] M.D. Buhmann, Radial Basis Functions: Theory and Implementations, Cambridge University Press, 2003.
[2] G.E. Fasshauer, Meshfree Approximation Methods with Matlab, World Scientific Publishing, Singapore, 2007.
[3] H. Wendland, Scattered Data Approximation, Cambridge University Press, 2004.
[4] R. Cavoretto, A. De Rossi, A trivariate interpolation algorithm using a cube-partition searching procedure, SIAM J. Sci. Comput. 37 (2015) A1891–A1908.
[5] R. Cavoretto, A. De Rossi, E. Perracchione, Efficient computation of partition of unity interpolants through a block-based searching technique, Comput. Math. Appl. 71 (2016) 2568–2584.
[6] F. Dell'Accio, F. Di Tommaso, F. Larosa, The multinode shepard method: MATLAB implementation, J. Approx. Softw. 1 (2024).
[7] X.-D. Liu, S. Osher, T. Chan, Weighted essentially non-oscillatory schemes, J. Comput. Phys. 115 (1994) 200–212.
[8] C.-W. Shu, High order weighted essentially nonoscillatory schemes for convection dominated problems, SIAM Rev. 51 (1) (1999) 82–126.
[9] R. Cavoretto, A. De Rossi, E. Perracchione, S. Lancellotti, Software implementation of the partition of unity method, Dolomites Res. Notes Approx. 15 (2) (2022) 35–46.
[10] H. Wendland, Fast evaluation of radial basis functions: methods based on partition of unity, in: J. Stöckler C.K. Chui (Ed.), Approximation Theory X: Wavelets, Splines, and Applications, Vanderbilt Univ. Press, Nashville, TN, 2002, pp. 473–483.
[11] G. Jiang, C.-W. Shu, Efficient implementation of weighted ENO schemes, J. Comput. Phys. 126 (1) (1996) 202–228.
[12] S. Amat, J. Ruiz, C.W. Shu, D.F. Yáñez, A new WENO-2*r* algorithm with progressive order of accuracy close to discontinuities, SIAM J. Numer. Anal. 58 (6) (2020) 3448–3474.
[13] F. Aràndiga, A.M. Belda, P. Mulet, Point-value WENO multiresolution applications to stable image compression, J. Sci. Comput. 43 (2) (2010) 158–182.
[14] P. Mulet, J. Ruiz, C.W. Shu, D.F. Yáñez, A non-separable progressive multivariate WENO-2r point value, App. Numer. Math. 204 (2024) 26–47.
[15] J.H. Halton, On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals, Numer. Math. 2 (1960) 84–90.
[16] S. Amat, D. Levin, J. Ruiz-Álvarez, D.F. Yáñez, Non-linear WENO B-spline based approximation method, Numer. Algor. 100 (10) (2024).