

Two and Three Dimensional Partition of Unity Interpolation by Product-Type Functions

Roberto Cavoretto*

Department of Mathematics “G. Peano”, University of Torino, Via Carlo Alberto 10, 10123 Torino, Italy

Received: 3 Mar. 2014, Revised: 3 Jun. 2014, Accepted: 4 Jun. 2014

Published online: 1 Jan. 2015

Abstract: In this paper we analyze the behavior of product-type radial basis functions (RBFs) and splines, which are used in a partition of unity interpolation scheme as local approximants. In particular, we deal with the case of bivariate and trivariate interpolation on a relatively large number of scattered data points. Thus, we propose the local use of compactly supported RBF and spline interpolants, which take advantage of being expressible in the multivariate setting as a product of univariate functions. Numerical experiments show good accuracy and stability of the partition of unity method combined with these product-type interpolants, comparing it with the one obtained by replacing compactly supported RBFs and splines with Gaussians.

Keywords: multivariate approximation, local interpolation schemes, partition of unity methods, radial basis functions, splines, scattered data

1 Introduction

We consider the problem of interpolating a continuous function $g : \Omega \rightarrow \mathbb{R}$ on a compact domain $\Omega \subset \mathbb{R}^m$, $m = 2, 3$, defined on a finite set $\mathcal{X}_N = \{\mathbf{x}_i, i = 1, 2, \dots, N\}$ of *data points* or *nodes*, which are situated in Ω . It consists of finding an interpolant $\mathcal{I} : \Omega \rightarrow \mathbb{R}$ such that, given the \mathbf{x}_i and the corresponding function values g_i , the interpolation conditions $\mathcal{I}(\mathbf{x}_i) = g(\mathbf{x}_i)$, $i = 1, 2, \dots, N$, are satisfied.

In particular, we are interested in considering the interpolation of large scattered data sets, a problem which has gained much interest in several areas of applied sciences and scientific computing, where the need of having accurate, fast and stable algorithms is often essential (see, e.g. [19, 22, 26, 27]). Among the various multivariate approximation techniques, the partition of unity methods such as Shepard's type interpolants turn out to be particularly effective in *meshfree* or *meshless* interpolation (see [2, 8, 12, 13, 14, 16, 23, 24, 28, 30, 31]).

In this paper, we propose the use of product-type compactly supported radial basis functions (RBFs) and splines, which are used in a partition of unity interpolation scheme as local approximants. Specifically, here we consider two families of basis functions, known as *Wendland functions* and *Lobachevsky splines*, which

can be expressed in the multivariate setting as products of univariate functions. The former are well-known in approximation theory and practice, and are usually used as radial functions in the field of multivariate interpolation and approximation (see e.g. [10, 19, 21, 34]). On the other hand, the latter, firstly considered in probability theory [20, 29], have successfully been proposed for multivariate scattered data interpolation and integration in [4, 5, 6] and for landmark-based image registration in [1, 3]. We remark that both of these families of univariate functions (depending on a shape parameter) are compactly supported, strictly positive definite, and enjoy noteworthy theoretical and computational properties, such as the spline convergence to the Gaussian function. Furthermore, we observe that the multivariate spline interpolant is noteworthy because it is neither a mesh-based formula nor a radial one, but it asymptotically behaves like a Gaussian interpolant (see [4, 9]). Numerical experiments point out that, for sufficiently regular basis functions such as the product-type radial basis function and spline of smoothness C^4 , the partition of unity scheme combined with RBF and spline interpolants is comparable in accuracy with that obtained by using Gaussian ones, even if they are usually much better conditioned than Gaussians.

* Corresponding author e-mail: roberto.cavoretto@unito.it

The paper is organized as follows. In Section 2 at first we consider the problem of scattered data interpolation by product-type Wendland functions and Lobachevsky splines, recalling their analytic expressions and some properties; then, we describe the partition of unity method, which makes use of product-type RBF and spline interpolants as local approximants. In Section 3 we refer to the corresponding partition of unity algorithms designed for bivariate and trivariate interpolation. Section 4 summarizes several numerical results in order to analyze accuracy and stability of the local approximation scheme combined with compactly supported radial basis function and spline interpolants, also comparing their errors with those of the Gaussian. Finally, Section 5 deals with conclusions and future work.

2 Partition of unity scheme

Let us consider a continuous function $g : \Omega \rightarrow \mathbb{R}$ on a compact domain $\Omega \subset \mathbb{R}^m$, $m \geq 1$, a set $\mathcal{X}_N = \{\mathbf{x}_i = (x_{1i}, x_{2i}, \dots, x_{mi}), i = 1, 2, \dots, N\} \subset \Omega$ of scattered data points, and the set $\mathcal{G}_N = \{g(\mathbf{x}_i), i = 1, 2, \dots, N\}$ of the corresponding function values.

2.1 Product-type functions

2.1.1 Radial basis functions

In order to construct an interpolation formula generated by compactly supported RBFs, for even $p \geq 0$ we consider the product-type interpolant of the form

$$F_p(\mathbf{x}) = \sum_{j=1}^N c_j \phi_{pj}(\mathbf{x}), \quad \mathbf{x} \in \Omega,$$

requiring $F_p(\mathbf{x}_i) = g(\mathbf{x}_i)$, $i = 1, 2, \dots, N$. The interpolant F_p is a linear combination of products of univariate shifted and rescaled functions ζ_p , called *Wendland functions* [32], i.e.

$$\phi_{pj}(\mathbf{x}) \equiv \phi_{pj}(\mathbf{x}; \delta) = \prod_{h=1}^m \zeta_p(\delta(x_h - x_{hj})), \quad (1)$$

where, setting $r = (x_h - x_{hj})$, for $j = 1, 2, \dots, N$

$$\begin{aligned} \zeta_0(\delta r) &\doteq (1 - \delta r)_+^3, \\ \zeta_2(\delta r) &\doteq (1 - \delta r)_+^3 (3\delta r + 1), \\ \zeta_4(\delta r) &\doteq (1 - \delta r)_+^5 (8(\delta r)^2 + 5\delta r + 1), \\ \zeta_6(\delta r) &\doteq (1 - \delta r)_+^7 (21(\delta r)^3 + 19(\delta r)^2 + 7\delta r + 1), \end{aligned}$$

and $\delta \in \mathbb{R}^+$ is a shape parameter. Note that the support of ζ_p is $[-1/\delta, 1/\delta]$. The coefficients $\mathbf{c} = \{c_j\}$ are computed by solving the linear system

$$\mathbf{A}\mathbf{c} = \mathbf{g},$$

where the interpolation matrix

$$\mathbf{A} = \{a_{ij}\} = \{\phi_{pj}(\mathbf{x}_i)\}, \quad i, j = 1, 2, \dots, N, \quad (2)$$

is symmetric and depends on the values of p and δ in (1). Since these compactly supported RBFs are strictly positive definite, the interpolation matrix \mathbf{A} in (2) is positive definite for any set of distinct nodes (see [19]).

Furthermore, since Wendland functions are (univariate) strictly positive definite functions, we can construct multivariate strictly positive definite functions from univariate ones (see, e.g., [34]).

Theorem 1. Suppose that $\lambda_1, \lambda_2, \dots, \lambda_m$ are strictly positive definite and integrable functions on \mathbb{R} , then

$$\Lambda(\mathbf{x}) = \lambda_1(x_1)\lambda_2(x_2)\cdots\lambda_m(x_m), \quad \mathbf{x} = (x_1, x_2, \dots, x_m) \in \mathbb{R}^m,$$

is a strictly positive definite function on \mathbb{R}^m .

2.1.2 Spline functions

For even $n \geq 2$, we construct the product-type spline interpolant of g at the nodes \mathbf{x}_i in the form

$$\tilde{F}_n(\mathbf{x}) = \sum_{j=1}^N \tilde{c}_j \tilde{\phi}_{nj}(\mathbf{x}), \quad \mathbf{x} \in \Omega,$$

requiring $\tilde{F}_n(\mathbf{x}_i) = g(\mathbf{x}_i)$, $i = 1, 2, \dots, N$. The interpolant \tilde{F}_n is a linear combination of products of univariate shifted and rescaled functions f_n^* , known as *Lobachevsky splines* [4], i.e.

$$\tilde{\phi}_{nj}(\mathbf{x}) \equiv \tilde{\phi}_{nj}(\mathbf{x}; \alpha) = \prod_{h=1}^m f_n^*(\alpha(x_h - x_{hj})), \quad (3)$$

where for $j = 1, 2, \dots, N$

$$\begin{aligned} f_n^*(\alpha(x_h - x_{hj})) &= \sqrt{\frac{n}{3}} \frac{1}{2^n(n-1)!} \sum_{k=0}^n (-1)^k \binom{n}{k} \\ &\quad \times \left[\sqrt{\frac{n}{3}} \alpha(x_h - x_{hj}) + (n-2k) \right]_+^{n-1}, \end{aligned}$$

or, equivalently,

$$\begin{aligned} f_n^*(\alpha(x_h - x_{hj})) &= \sqrt{\frac{n}{3}} \frac{1}{2^n(n-1)!} \sum_{k=0}^{\lfloor \frac{n}{2} + \frac{1}{2} \sqrt{\frac{n}{3}} \alpha x \rfloor} (-1)^k \binom{n}{k} \\ &\quad \times \left[\sqrt{\frac{n}{3}} \alpha(x_h - x_{hj}) + (n-2k) \right]_+^{n-1}, \end{aligned}$$

and $\alpha \in \mathbb{R}^+$ is a shape parameter. Here the support of f_n^* is given by $[-\sqrt{3n}/\alpha, \sqrt{3n}/\alpha]$. The coefficients $\tilde{\mathbf{c}} = \{\tilde{c}_j\}$ are obtained by solving the system of linear equations

$$\tilde{\mathbf{A}}\tilde{\mathbf{c}} = \mathbf{g},$$

whose interpolation matrix

$$\tilde{\mathbf{A}} = \{\tilde{a}_{ij}\} = \{\tilde{\phi}_{nj}(\mathbf{x}_i)\}, \quad i, j = 1, 2, \dots, N, \quad (4)$$

is symmetric and depends on the choice of n and α in (3). As also these splines turn out to be strictly positive definite for any even $n \geq 2$, the interpolation matrix \tilde{A} in (4) is positive definite for any distinct node set.

From the central limit theorem (see [20,29]), we remark that the m -variate spline converges for $n \rightarrow \infty$ to the m -variate Gaussian, i.e.

$$\lim_{n \rightarrow \infty} \prod_{i=1}^m f_n^*(\alpha x_i) = \frac{1}{(2\pi)^{m/2}} \exp\left(-\frac{\alpha^2(\sum_{i=1}^m x_i^2)}{2}\right).$$

Hence, these product-type functions asymptotically behave like radial functions, though they are not radial in themselves.

Finally, since also these splines are (univariate) strictly positive definite functions for even $n \geq 2$, from Theorem 1 it follows that we can construct multivariate strictly positive definite functions from univariate ones.

2.2 Interpolation method

In this subsection we present the partition of unity method. It was firstly suggested in [7,25] in the context of meshfree Galerkin methods for the solution of partial differential equations, but then it became a common tool also used in the field of approximation theory (see [34]).

The basic idea of the partition of unity method is to start with a partition of the open and bounded domain $\Omega \subseteq \mathbb{R}^m$ into d subdomains Ω_j such that $\Omega \subseteq \bigcup_{j=1}^d \Omega_j$ with some mild overlap among the subdomains. Firstly, we take a partition of unity, which consists of a family of compactly supported, non-negative, continuous functions W_j with $\text{supp}(W_j) \subseteq \Omega_j$ such that

$$\sum_{j=1}^d W_j(\mathbf{x}) = 1.$$

Then, we consider the global approximant of the form

$$\mathcal{I}(\mathbf{x}) = \sum_{j=1}^d H_j(\mathbf{x})W_j(\mathbf{x}), \quad \mathbf{x} \in \Omega, \tag{5}$$

where

$$H_j(\mathbf{x}) = \sum_{k=1}^{m_j} c_k \varphi_k(\mathbf{x}),$$

is a local interpolant, which is constructed using the m_j nodes belonging to the subdomain Ω_j and solving a local interpolation problem. In particular, H_j denotes either a RBF interpolant F_{p_j} or a spline interpolant \tilde{F}_{n_j} , whereas φ_k represents the generic k -th basis, ϕ_{pk} or $\tilde{\phi}_{nk}$. In fact, we note that if the local approximants satisfy the interpolation conditions at data point \mathbf{x}_i , i.e.

$$H_j(\mathbf{x}_i) = g(\mathbf{x}_i),$$

the global approximant also interpolates at this node, i.e.

$$\mathcal{I}(\mathbf{x}_i) = g(\mathbf{x}_i), \quad i = 1, 2, \dots, N.$$

Then, we give the following definition (see [33]).

Definition 1. Let $\Omega \subseteq \mathbb{R}^m$ be a bounded set. Let $\{\Omega_j\}_{j=1}^d$ be an open and bounded covering of Ω . This means that all Ω_j are open and bounded and that $\Omega \subseteq \bigcup_{j=1}^d \Omega_j$. Set $\delta_j = \text{diam}(\Omega_j) = \sup_{\mathbf{x}, \mathbf{y} \in \Omega_j} \|\mathbf{x} - \mathbf{y}\|_2$. We call a family of nonnegative functions $\{W_j\}_{j=1}^d$ with $W_j \in C^k(\mathbb{R}^m)$ a k -stable partition of unity with respect to the covering $\{\Omega_j\}_{j=1}^d$ if

- 1) $\text{supp}(W_j) \subseteq \Omega_j$;
- 2) $\sum_{j=1}^d W_j(\mathbf{x}) \equiv 1$ on Ω ;
- 3) for every $\beta \in \mathbb{N}_0^m$ with $|\beta| \leq k$ there exists a constant $C_\beta > 0$ such that

$$\|D^\beta W_j\|_{L^\infty(\Omega_j)} \leq C_\beta / \delta_j^{|\beta|},$$

for all $1 \leq j \leq d$.

In accordance with the statements in [33] we require additional regularity assumptions on the covering $\{\Omega_j\}_{j=1}^d$.

Definition 2. Suppose that $\Omega \subseteq \mathbb{R}^m$ is bounded and $\mathcal{X}_N = \{\mathbf{x}_i, i = 1, 2, \dots, N\} \subseteq \Omega$ are given. An open and bounded covering $\{\Omega_j\}_{j=1}^d$ is called regular for (Ω, \mathcal{X}_N) if the following properties are satisfied:

- 1) for each $\mathbf{x} \in \Omega$, the number of subdomains Ω_j with $\mathbf{x} \in \Omega_j$ is bounded by a global constant K ;
- 2) each subdomain Ω_j satisfies an interior cone condition;
- 3) the local fill distances $h_{\mathcal{X}_j, \Omega_j}$ are uniformly bounded by the global fill distance $h_{\mathcal{X}_N, \Omega}$, where $\mathcal{X}_j = \mathcal{X}_N \cap \Omega_j$.

Therefore, after defining the space $C_v^k(\mathbb{R}^m)$ of all functions $g \in C^k$ whose derivatives of order $|\beta| = k$ satisfy $D^\beta g(\mathbf{x}) = \mathcal{O}(\|\mathbf{x}\|_2^v)$ for $\|\mathbf{x}\|_2 \rightarrow 0$, we consider the following convergence result (see, e.g., [19,34]).

Theorem 2. Let $\Omega \subseteq \mathbb{R}^m$ be open and bounded and suppose that $\mathcal{X}_N = \{\mathbf{x}_i, i = 1, 2, \dots, N\} \subseteq \Omega$. Let $\varphi \in C_v^k(\mathbb{R}^m)$ be a strictly positive definite function. Let $\{\Omega_j\}_{j=1}^d$ be a regular covering for (Ω, \mathcal{X}_N) and let $\{W_j\}_{j=1}^d$ be k -stable for $\{\Omega_j\}_{j=1}^d$. Then the error between $g \in \mathcal{N}_\varphi(\Omega)$, where \mathcal{N}_φ is the native space of φ , and its partition of unity interpolant (5) can be bounded by

$$\|D^\beta g(\mathbf{x}) - D^\beta \mathcal{I}(\mathbf{x})\| \leq Ch_{\mathcal{X}_N, \Omega}^{(k+v)/2-|\beta|} |g|_{\mathcal{N}_\varphi(\Omega)},$$

for all $\mathbf{x} \in \Omega$ and all $|\beta| \leq k/2$.

Remark. We observe that the partition of unity method preserves the local approximation order for the global fit. In fact, we can efficiently compute large product-type interpolants by solving small interpolation problems and then combine them along with the global partition of unity $\{W_j\}_{j=1}^d$. This approach allows us to decompose a

large problem into many small problems, ensuring at the same time that the accuracy obtained for the local fits is carried over to the global one. In particular, the partition of unity method can be thought as a Shepard's method with higher-order data, since local approximations H_j instead of data values g_j are used.

3 Algorithms

In this section, we describe in detail the partition of unity algorithms used in the interpolation processes, first referring to the bivariate algorithm and then to the trivariate one. They are characterized by the partition of the domain Ω in square or cube cells, enabling us to use efficient searching procedures. The basic versions of these interpolation algorithms have been proposed and widely tested in [17,18]. Here, such algorithms have been modified and efficiently updated to locally apply the product-type interpolants.

3.1 Bivariate interpolation

INPUT:

- N , number of data;
- $\mathcal{X}_N = \{(x_{1i}, x_{2i}), i = 1, 2, \dots, N\}$, set of data points;
- $\mathcal{G}_N = \{g_i, i = 1, 2, \dots, N\}$, set of data values;
- d , number of subdomains;
- $\mathcal{C}_d = \{(\bar{x}_{1i}, \bar{x}_{2i}), i = 1, 2, \dots, d\}$, set of subdomain points;
- s , number of evaluation points;
- $\mathcal{E}_s = \{(\tilde{x}_{1i}, \tilde{x}_{2i}), i = 1, 2, \dots, s\}$, set of evaluation points.

OUTPUT:

- $\mathcal{A}_s = \{\mathcal{I}(\tilde{x}_{1i}, \tilde{x}_{2i}), i = 1, 2, \dots, s\}$, set of approximated values.

Stage 1. The set \mathcal{X}_N of nodes and the set \mathcal{E}_s of evaluation points are ordered with respect to the x_2 -axis direction by applying a *quicksort*_{v₂} procedure.

Stage 2. For each subdomain point $(\bar{x}_{1i}, \bar{x}_{2i}), i = 1, 2, \dots, d$, a local circular subdomain is constructed, whose radius depends on the subdomain number d , that is

$$\delta_{subdom} = \sqrt{\frac{2}{d}}. \quad (6)$$

This value is suitably chosen, supposing to have a nearly uniform node distribution and assuming that the ratio $n/d \approx 4$.

Stage 3. A double structure of crossed strips is constructed as follows:

- i) a first family of q strips, parallel to the x_1 -axis direction, is considered taking

$$q = \left\lceil \frac{1}{\delta_{subdom}} \right\rceil, \quad (7)$$

and a *quicksort*_{x₁} procedure is applied to order the nodes belonging to each strip;

- ii) a second family of q strips, parallel to the x_2 -axis direction, is considered.

Note that each of the two strip structures are ordered and numbered from 1 to q ; moreover, the choice in (7) follows directly from the side length of the domain Ω (unit square), that here is 1, and the subdomain radius δ_{subdom} in (6).

Stage 4. The unit square is partitioned by a square-based structure consisted of q^2 squares, whose length of the sides is given by $\delta_{square} \equiv \delta_{subdom}$. Then, the following structure is considered:

- the sets \mathcal{X}_N , \mathcal{C}_d and \mathcal{E}_s are partitioned by the square structure into q^2 subsets \mathcal{X}_{N_k} , \mathcal{C}_{d_k} and \mathcal{E}_{s_k} , $k = 1, 2, \dots, q^2$,

where N_k , d_k and s_k are the number of points in the k -th square.

Stage 5. In order to identify the squares to be examined in the searching procedure, we consider the following two steps:

- (I) since $\delta_{square} \equiv \delta_{subdom}$, the ratio between these quantities is denoted by $i^* = \delta_{subdom}/\delta_{square} = 1$. Thus, the number $j^* = (2i^* + 1)^2$ of squares to be examined for each node is 9.
- (II) for each square $k = [v, w]$, $v, w = 1, 2, \dots, q$, a square-based searching procedure is considered, examining the points from the square $[v - i^*, w - i^*]$ to the square $[v + i^*, w + i^*]$, with the exception of those points close to the boundary of Ω , where we reduce the total number of squares to be examined.

Then, after defining which and how many squares are to be examined, the square-based searching procedure is applied:

- for each subdomain point of \mathcal{C}_{d_k} , $k = 1, 2, \dots, q^2$, to determine all nodes belonging to a subdomain. The number of nodes of the subdomain centred at $(\bar{x}_{1i}, \bar{x}_{2i})$ is counted and stored in m_i , $i = 1, 2, \dots, d$;
- for each evaluation point of \mathcal{E}_{s_k} , $k = 1, 2, \dots, q^2$, to find all those belonging to a subdomain of centre $(\bar{x}_{1i}, \bar{x}_{2i})$ and radius δ_{subdom} . The number of subdomains containing the i -th evaluation point is counted and stored in r_i , $i = 1, 2, \dots, s$.

Stage 6. A local interpolant H_j , $j = 1, 2, \dots, d$, is found for each subdomain point.

Stage 7. A local approximant H_j and a weight function W_j , $j = 1, 2, \dots, d$, is found for each evaluation point.

Stage 8. Applying the global fit (5), one can find approximated values computed at any evaluation point $(\tilde{x}_1, \tilde{x}_2) \in \mathcal{E}_s$.

3.2 Trivariate interpolation

INPUT:

- N , number of data;
- $\mathcal{X}_N = \{(x_{1i}, x_{2i}, x_{3i}), i = 1, 2, \dots, N\}$, set of data points;
- $\mathcal{G}_N = \{g_i, i = 1, 2, \dots, N\}$, set of data values;
- d , number of subdomains;
- $\mathcal{C}_d = \{(\bar{x}_{1i}, \bar{x}_{2i}, \bar{x}_{3i}), i = 1, 2, \dots, d\}$, set of subdomain points;
- s , number of evaluation points;
- $\mathcal{E}_s = \{(\tilde{x}_{1i}, \tilde{x}_{2i}, \tilde{x}_{3i}), i = 1, 2, \dots, s\}$, set of evaluation points.

OUTPUT:

- $\mathcal{A}_s = \{\mathcal{I}(\tilde{x}_{1i}, \tilde{x}_{2i}, \tilde{x}_{3i}), i = 1, 2, \dots, s\}$, set of approximated values.

Stage 1. The set \mathcal{X}_N of nodes and the set \mathcal{E}_s of evaluation points are ordered with respect to the x_3 -axis direction by applying a *quicksort_{x₃} procedure*.

Stage 2. For each subdomain point $(\bar{x}_{1i}, \bar{x}_{2i}, \bar{x}_{3i}), i = 1, 2, \dots, d$, a local spherical subdomain is constructed, whose spherical radius depends on the subdomain number d , that is

$$\delta_{subdom} = \frac{\sqrt{2}}{\sqrt[3]{d}}. \tag{8}$$

Although other choices of δ_{subdom} are possible, this value is suitably chosen, supposing to have a nearly uniform node distribution and assuming that the ratio $n/d \approx 8$.

Stage 3. A triple structure of intersecting parallelepipeds is constructed as follows:

- i) a first family of q parallelepipeds, parallel to the x_1 -axis direction, is considered taking

$$q = \left\lceil \frac{1}{\delta_{subdom}} \right\rceil, \tag{9}$$

and a *quicksort_{x₁} procedure* is applied to order the nodes belonging to each parallelepiped;

- ii) a second family of q parallelepipeds, parallel to the x_2 -axis direction, is constructed and a *quicksort_{x₂} procedure* is used to order the nodes belonging to each of the resulting parallelepipeds;
- iii) a third family of q parallelepipeds, parallel to the x_3 -axis, is considered.

Note that each of the three families of parallelepipeds are ordered and numbered from 1 to q ; the choice in (9) follows directly from the side length of the domain, i.e. the unit cube, and the subdomain radius δ_{subdom} in (8).

Stage 4. The unit cube is partitioned by a cube-based structure consisted of q^3 cubes, whose side length is $\delta_{cube} \equiv \delta_{subdom}$. Then, the following structure is considered:

- the sets $\mathcal{X}_N, \mathcal{C}_d$ and \mathcal{E}_s are partitioned by the cube structure into q^3 subsets $\mathcal{X}_{N_k}, \mathcal{C}_{d_k}$ and $\mathcal{E}_{s_k}, k = 1, 2, \dots, q^3$, where N_k, d_k and s_k are the number of points in the k -th cube.

Stage 5. In order to identify the cubes to be examined in the searching procedure, we consider the following two steps:

- (I) since $\delta_{cube} \equiv \delta_{subdom}$, the ratio between these quantities is denoted by $i^* = \delta_{subdom} / \delta_{cube} = 1$. Thus, the number $j^* = (2i^* + 1)^3$ of cubes to be examined for each node is 27.
- (II) for each cube $k = [u, v, w], u, v, w = 1, 2, \dots, q$, a cube-partition searching procedure is considered, examining the points from the cube $[u - i^*, v - i^*, w - i^*]$ to the cube $[u + i^*, v + i^*, w + i^*]$, with the exception of those points close to the boundary of Ω , where we reduce the total number of cubes to be examined.

Then, after defining which and how many cubes are to be examined, the cube-partition searching procedure is applied:

- for each subdomain point of $\mathcal{C}_{d_k}, k = 1, 2, \dots, q^3$, to determine all nodes belonging to a subdomain. The number of nodes of the subdomain centred at $(\bar{x}_{1i}, \bar{x}_{2i}, \bar{x}_{3i})$ is counted and stored in $m_i, i = 1, 2, \dots, d$;
- for each evaluation point of $\mathcal{E}_{s_k}, k = 1, 2, \dots, q^3$, in order to find all those belonging to a subdomain of centre $(\bar{x}_{1i}, \bar{x}_{2i}, \bar{x}_{3i})$ and radius δ_{subdom} . The number of subdomains containing the i -th evaluation point is counted and stored in $r_i, i = 1, 2, \dots, s$.

Stage 6. A local interpolant $H_j, j = 1, 2, \dots, d$, is found for each subdomain point.

Stage 7. A local approximant H_j and a weight function $W_j, j = 1, 2, \dots, d$, is found for each evaluation point.

Stage 8. Applying the global interpolant (5), one can find approximated values computed at any evaluation point $(\tilde{x}_1, \tilde{x}_2, \tilde{x}_3) \in \mathcal{E}_s$.

3.3 Complexity

The algorithms make a repeated use of the *quicksort* routine along different directions. They require on average a time complexity $\mathcal{O}(M \log M)$, M being the number of nodes to be sorted. In particular, we need a preprocessing phase to build the data structure, whose computational cost is of the order $\mathcal{O}(N \log N)$ for the sorting of all N nodes and $\mathcal{O}(s \log s)$ for the sorting of all s evaluation points in Stage 1. Then, to compute the local product-type interpolants, we solve d linear systems of (relatively) small sizes and the computational cost is of order $\mathcal{O}(m_i^3), i = 1, 2, \dots, d$, for each subdomain, where m_i is the number of nodes in the i -th subdomain (see Stage 6). Moreover, in Stage 5, 7 and 8 we also have a cost of $r_k \cdot \mathcal{O}(m_i), i = 1, 2, \dots, d, k = 1, 2, \dots, s$, for the k -th evaluation point of \mathcal{E}_s .

4 Numerical results

In this section we investigate the performances of the partition of unity interpolants obtained by using product-type interpolants as local approximants. In doing so, we focus on accuracy and stability of such interpolation formulas, considering some sets \mathcal{X}_N of Halton scattered data points contained in $\Omega = [0, 1]^m \subset \mathbb{R}^m$, for $m = 2, 3$. They are uniformly distributed random points and generated by using the MATLAB program `haltonseq.m` [19]. In particular, here we consider some sets of Halton points of size $N = 16641, 66049$, and $N = 35937, 274625$, for bivariate and trivariate interpolation, respectively. Note that in this local interpolation scheme we need to solve d linear systems, one for each subdomain Ω_j , $j = 1, 2, \dots, d$, of relatively small size $m_j \times m_j$.

Thus, we analyze the behavior of the partition of unity scheme combined with radial basis function (R) and spline (S) interpolants of smoothness C^2 ($p = 2$ and $n = 4$) and C^4 ($p = 4$ and $n = 6$), which are denoted by R2, R4 and S2, S4, respectively. Moreover, for a comparison we also report the results obtained by using the Gaussian (G) as local approximant.

All the results reported below are obtained computing on the following bivariate and trivariate Franke's test functions (see, e.g., [2, 11]), i.e.,

$$g_1(x_1, x_2) = \frac{3}{4}e^{-\frac{(9x_1-2)^2+(9x_2-2)^2}{4}} + \frac{3}{4}e^{-\frac{(9x_1+1)^2}{49} - \frac{9x_2+1}{10}} \\ + \frac{1}{2}e^{-\frac{(9x_1-7)^2+(9x_2-3)^2}{4}} - \frac{1}{5}e^{-(9x_1-4)^2 - (9x_2-7)^2},$$

and

$$g_2(x_1, x_2, x_3) = \frac{3}{4}e^{-\frac{(9x_1-2)^2+(9x_2-2)^2+(9x_3-2)^2}{4}} \\ + \frac{3}{4}e^{-\frac{(9x_1+1)^2}{49} - \frac{9x_2+1}{10} - \frac{9x_3+1}{10}} \\ + \frac{1}{2}e^{-\frac{(9x_1-7)^2+(9x_2-3)^2+(9x_3-5)^2}{4}} \\ - \frac{1}{5}e^{-(9x_1-4)^2 - (9x_2-7)^2 - (9x_3-5)^2},$$

the root mean square error (RMSE), whose formula is given by

$$RMSE = \sqrt{\frac{1}{s} \sum_{i=1}^s |g(\mathbf{x}_i) - \mathcal{I}(\mathbf{x}_i)|^2}.$$

However, in order to have a more complete picture on the local interpolation scheme, we have tested other functions which we here omit for shortness, since accuracy and stability have shown a uniform behavior of the errors.

In Figures 1–2 we show the behavior of compactly supported radial basis function and spline interpolation errors both for 2D and 3D cases by varying the value of the shape parameters $\alpha \in [1, 10]$, $\delta \in [0.1, 1.9]$. Note that there is an exact correspondence between the spline and

Gaussian shape parameter α , whereas it does not hold by varying δ ; hence, though this study turns out to be significant from a numerical standpoint, a direct comparison of δ and α should be viewed as purely indicative.

Numerical experiments highlight that the variation of α and δ may greatly influence the quality of approximation results. Moreover, the behavior of such errors turns out to be more regular and uniform for compactly supported radial basis functions and splines of product-type, whereas it is quite unstable for the Gaussian above all when we take small values of α . More precisely, comparing for example spline and Gaussian results, this study points out that the interpolation scheme with product-type interpolants is more stable than the one with the Gaussian. In fact, in the latter case we have interpolation matrices that are very ill-conditioned, thus producing the fast and sudden error variation shown in Figure 1 for (i) $\alpha \in [1, 3.9]$, (ii) $\alpha \in [1, 6.3]$, (iii) $\alpha \in [1, 2.5]$ and (iv) $\alpha \in [1, 3.9]$. Thus, in practice, this error analysis also provides useful information on the conditioning of the different local approximants. Furthermore, analyzing all the tests done, we observe that the product-type functions such as the R4 and S4 (of smoothness C^4) are comparable in accuracy with the Gaussians.

Specifically, we remark that the best level of Gaussian accuracy is reached when $\alpha \in [1, 6]$, which is exactly the interval where the Gaussian is unstable. Conversely, though the optimality interval remains the same also for the spline, in that region we have a complete stability; similar considerations, even if depending on the value δ , can also be done for the radial basis function case. Hence, it follows that these tests point out reliability of product-type interpolants also when they are used as local approximants in partition of unity interpolation schemes.

Finally, in Tables 1–2 we report the errors obtained considering optimal values of δ and α , i.e. taking values for which we obtain the smallest RMSEs.

Table 1 RMSEs obtained by using optimal values of δ and α for g_1 .

N	16641		66049	
	RMSE	$\delta_{opt}, \alpha_{opt}$	RMSE	$\delta_{opt}, \alpha_{opt}$
R2	2.4893E−5	0.88	6.9639E−6	0.84
R4	1.8259E−6	0.46	2.9154E−7	0.66
S2	2.3715E−5	2.9	6.5901E−6	2.7
S4	1.5553E−6	2.5	2.5635E−7	2.6
G	7.9092E−7	2.9	3.1658E−7	2.6

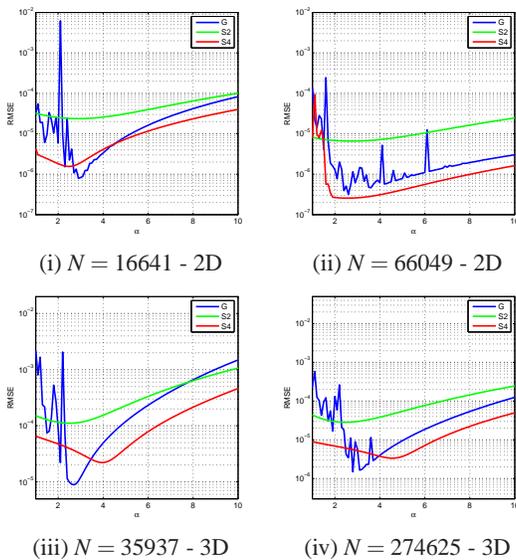


Fig. 1 RMSEs obtained by varying the shape parameters α for bivariate and trivariate Franke’s functions.

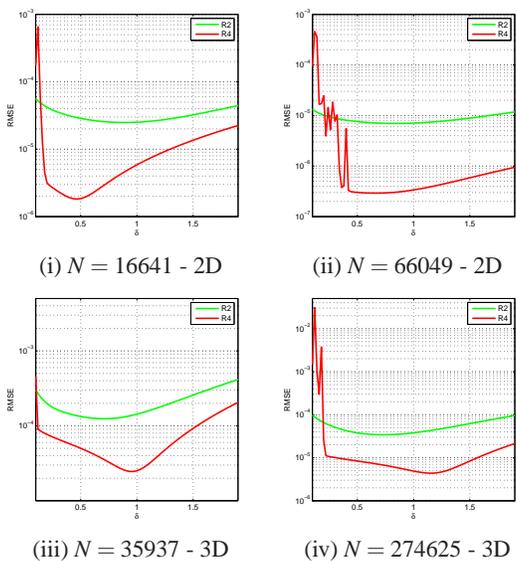


Fig. 2 RMSEs obtained by varying the shape parameters δ for bivariate and trivariate Franke’s functions.

5 Conclusions and future work

In this paper we studied the performance of product-type interpolants when they are applied as local approximants in a local interpolation scheme such as the partition of unity method. More precisely, we analyzed the behavior of such interpolants in the bivariate and trivariate setting,

Table 2 RMSEs obtained by using optimal values of δ and α for g_2 .

N	35937		274625	
	RMSE	$\delta_{opt}, \alpha_{opt}$	RMSE	$\delta_{opt}, \alpha_{opt}$
R2	1.2433E - 4	0.70	3.4168E - 5	0.74
R4	2.4558E - 5	0.96	4.3705E - 6	1.16
S2	1.1088E - 4	2.5	2.8425E - 5	2.5
S4	2.2148E - 5	4.0	3.3542E - 6	4.6
G	8.8797E - 6	2.7	1.4928E - 6	2.8

considering some sets of scattered data points. This study highlighted that these product-type functions work well also in local approaches. Finally, as confirmed by several numerical experiments, they turn out to be more stable than Gaussians and in general, for sufficiently regular basis functions, comparable in accuracy.

As future work, starting from [15] we are going to implement adaptive partition of unity algorithms for scattered data interpolation in high-dimensions, adopting suitable data structures like *kd-trees* and *range trees*.

Acknowledgement

The author thanks the financial support of the Department of Mathematics “G. Peano”, University of Torino, project “Numerical analysis for life sciences” (2012).

References

- [1] G. Allasia, R. Cavoretto, A. De Rossi, B. Quatember, W. Recheis, M. Mayr, and S. Demertzis, In: T.E. Simos et al. (Eds.), Proceedings of the ICNAAM 2010, AIP Conf. Proc. **1281**, 716–719 (2010).
- [2] G. Allasia, R. Besenghi, R. Cavoretto and A. De Rossi, Appl. Math. Comput. **217**, 5949–5966 (2011).
- [3] G. Allasia, R. Cavoretto and A. De Rossi, Math. Methods Appl. Sci. **35**, 923–934 (2012).
- [4] G. Allasia, R. Cavoretto and A. De Rossi, Comput. Appl. Math. **32**, 71–87 (2013).
- [5] G. Allasia, R. Cavoretto and A. De Rossi, Int. J. Comput. Math. **90**, 2003–2018 (2013).
- [6] AA Borisenko, Cylindrical multidimensional surfaces in Lobachevsky space, Journal of Soviet Mathematics, Springer (1991).
- [7] I. Babuška and J.M. Melenk, Internat. J. Numer. Methods. Engrg. **40**, 727–758 (1997).
- [8] M.W. Berry and K.S. Minser, ACM Trans. Math. Software **25**, 353–366 (1999).
- [9] R. Brinks, Comput. Appl. Math. **27**, 79–92 (2008).

- [10] M.D. Buhmann, *Radial Basis Functions: Theory and Implementation*, Cambridge Univ. Press, Cambridge, (2003).
- [11] R. Cavoretto and A. De Rossi, *J. Comput. Appl. Math.* **234**, 1505–1521 (2010).
- [12] R. Cavoretto and A. De Rossi, *Appl. Math. Sci.* **4**, 3425–3435 (2010).
- [13] R. Cavoretto and A. De Rossi, *Appl. Math. Lett.* **25**, 1251–1256 (2012).
- [14] R. Cavoretto, In: T.E. Simos et al. (Eds.), *Proceedings of the ICNAAM 2012*, AIP Conf. Proc. **1479**, 1054–1057 (2012).
- [15] R. Cavoretto, *Comput. Appl. Math.*, (2014). DOI: 10.1007/s40314-013-0104-9
- [16] R. Cavoretto and A. De Rossi, *Math. Methods Appl. Sci.*, (2014). DOI: 10.1002/mma.2906
- [17] R. Cavoretto and A. De Rossi, *Comput. Math. Appl.* **67**, (2014) 1024–1038.
- [18] R. Cavoretto and A. De Rossi, submitted for publication (2014).
- [19] G.E. Fasshauer, *Meshfree Approximation Methods with MATLAB*, World Scientific Publishing, Singapore, (2007).
- [20] B.V. Gnedenko, *The Theory of Probability*, MIR, Moscow, (1976).
- [21] A. Iske, *Rend. Sem. Mat. Univ. Pol. Torino* **61**, 247–285 (2003).
- [22] A. Iske, *Rend. Sem. Mat. Univ. Pol. Torino* **69**, 217–246 (2011).
- [23] M.A. Iyer, L.T. Watson and M.W. Berry, In: R. Menezes et al. (Eds.), *Proceedings of the Annual Southeast Conference*, 476–481 (2006).
- [24] D. Lazzaro and L.B. Montefusco, *J. Comput. Appl. Math.* **140**, 521–536 (2002).
- [25] J.M. Melenk and I. Babuška, *Comput. Methods. Appl. Mech. Engrg.* **139**, 289–314 (1996).
- [26] V.P. Nguyen, T. Rabczuk, S. Bordas and M. Duflot, *Math. Comput. Simulation* **79**, 763–813 (2008).
- [27] J. Pouderoux, I. Tobor, J.-C. Gonzato and P. Guitton, In: D. Pfoser et al. (Eds.), *GIS 2004: Proceedings of the Twelfth ACM International Symposium on Advances in Geographic Information Systems*, 232–240 (2004).
- [28] R.J. Renka, *ACM Trans. Math. Software* **14**, 139–148 (1988).
- [29] A. Rényi, *Calcul des Probabilités*, Dunod, Paris, (1966).
- [30] E. Sáenz-de-Cabezón, L.J. Hernández, M.T. Rivas, E. García-Ruiz, V. Marco, I. Pérez-Moreno and F. Javier Sáenz-de-Cabezón, *Math. Comput. Simulation* **82**, 2–14 (2011).
- [31] W.I. Thacker, J. Zhang, L.T. Watson, J.B. Birch, M.A. Iyer and M.W. Berry, *ACM Trans. Math. Software* **37**, Art. 34, 1–20 (2010).
- [32] H. Wendland, *Adv. Comput. Math.* **4**, 389–396 (1995).
- [33] H. Wendland, In: C.K. Chui et al. (Eds.), *Approximation Theory X: Wavelets, Splines, and Applications*, Vanderbilt Univ. Press, Nashville, TN pp. 473–483.(2002)
- [34] H. Wendland, *Scattered Data Approximation*, Cambridge University Press, Cambridge, (2005).



Roberto Cavoretto is a Fixed-Term Research Fellow of Numerical Analysis at the University of Torino, Italy. He got his Ph.D. degree in Science and High Technology (Mathematics) at the University of Torino in 2010. In the same University he held a post-doc position until 2011. His research activity is mainly focused on topics of numerical analysis and applied mathematics such as: scattered data approximation, spherical interpolation, and applications to image registration. He has published research articles in international refereed journals of applied mathematics.