Communications in Applied and Industrial Mathematics,
ISSN 2038-0909, e-431
DOI: 10.1685/journal.caim.431

Partition of unity algorithm for two-dimensional interpolation using compactly supported radial basis functions

Roberto Cavoretto

Dipartimento di Matematica "Giuseppe Peano"
Università degli Studi di Torino, Italy
roberto.cavoretto@unito.it

Communicated by Luca Formaggia

Abstract

In this paper we present a fast algorithm for two-dimensional interpolation of large scattered data sets. It is based on the partition of unity method, which makes use of compactly supported radial basis functions as local approximants. This interpolation technique is characterized by the use of an efficiently implemented nearest neighbour searching procedure, which exploits a suitable and optimal partition of the domain in strips obtained in a completely automatic way. Analysis of computational complexity and numerical results show efficiency and accuracy of the proposed algorithm, also considering an application to Earth's topography.

Keywords: `partition of unity methods, fast algorithms, scattered data, interpolation, compactly supported functions.`

AMS subject classification: `65D05, 65D15, 65D17.`

1. Introduction.

In the last decades, efficient methods and algorithms using radial basis functions (RBFs) have gained popularity in various areas of scientific computing such as multivariate interpolation, approximation theory, meshfree (or meshless) methods, neural networks, computer graphics, computer aided geometric design (CAGD) and machine learning. In particular, the need of having fast algorithms and powerful and flexible software is of great interest mainly in applications, where the amount of data to be interpolated is often very large, say many thousands or millions of data (see, e.g., [1–6] for an overview).

In [7] a modified Shepard's algorithm for bivariate interpolation of large scattered data sets was proposed. The main novelty of the paper consisted in constructing a partition of the domain in a suitable number of parallel strips

Received on November 3rd, 2012. Accepted on February 21st, 2013. Published on February 28th, 2013.

and in the use of local neighbourhoods, in order to obtain an optimized strip searching procedure to be applied in the nearest neighbour searching technique. This process allowed us to achieve good efficiency. Moreover, this algorithm was proposed in [8] for spherical interpolation (see [9] for further details), and then extended in [10] to the more general *Partition of Unity Method* [11].

Now, in this paper we present a partition of unity algorithm for bivariate interpolation, suitably modifying the efficient searching procedure implemented in [10] for interpolation on the sphere to interpolation on a plane domain. In particular, we here combine the partition of unity method with compactly supported radial basis functions (CSRBFs) like Wendland's functions [6]. In fact, the use of a local approach along with Wendland's functions of arbitrary low smoothness, which are less ill-conditioned than higher-order basis functions such as Gaussians and other RBFs of infinite smoothness, give us good accuracy and stability. Furthermore, this approximation technique and the related fast algorithm can successfully be used in applications of two-dimensional interpolation processes.

The paper is organized as follows. In Section 2 we consider CSRBF interpolation. Section 3 presents the partition of unity method using CSRBFs as local approximants. In Section 4 the partition of unity algorithm is described, whereas in Section 5 its computational complexity is analyzed. Section 6 shows some numerical tests concerning effectiveness and accuracy of the considered algorithm for both some test cases and an application to Earth's topography data.

2. CSRBF interpolation.

Let $\mathcal{X}_n = \{x_i\}_{i=1}^n$ be a set of distinct nodes, arbitrarily distributed in a domain $\Omega \subseteq \mathbb{R}^N$, $N \geq 1$, with an associated set $\mathcal{F}_n = \{f_i\}_{i=1}^n$ of data values. Thus, we define the classical interpolation problem.

Problem 2.1. *Let $\mathcal{X}_n = \{x_i\}_{i=1}^n$ denote a set of distinct data points on the domain $\Omega \subseteq \mathbb{R}^N$, and let $\mathcal{F}_n = \{f_i\}_{i=1}^n$ denote the set of the corresponding data values of an unknown function $f : \Omega \to \mathbb{R}$. Find a (continuous) function $s : \Omega \to \mathbb{R}$, which satisfies the interpolation conditions*

$$s(x_i) = f_i, \quad i = 1, 2, \ldots, n. \tag{1}$$

Now, referring to CSRBF interpolation, we can express this problem as follows.

Definition 2.1. Given a set $\mathcal{X}_n = \{x_i\}_{i=1}^n$ of distinct data points arbitrarily distributed on $\Omega \subseteq \mathbb{R}^N$, and the associated set $\mathcal{F}_n = \{f_i\}_{i=1}^n$ of data values

DOI: 10.1685/journal.caim.431

of a function $f : \Omega \to \mathbb{R}$, a compactly supported radial basis function interpolant $s : \Omega \to \mathbb{R}$ has the form

$$s(x) = \sum_{j=1}^{n} a_j \phi(d(x, x_j)), \qquad x \in \Omega, \tag{2}$$

where $d(x, x_j) = ||x - x_j||_2$ is the Euclidean distance, $\phi : [0, \infty) \to \mathbb{R}$ is called *compactly supported radial basis function*, and s satisfies the interpolation conditions (1).

With regard to existence and uniqueness of the interpolation problem, we recall that the solution exists and is unique in the interpolation space

$$T_\phi = \text{span}\{\phi(d(\cdot, x_1)), \ldots, \phi(d(\cdot, x_n))\} \tag{3}$$

if and only if the corresponding interpolation matrix $A \in \mathbb{R}^{n \times n}$, that is

$$A_{i,j} = \phi(d(x_i, x_j)), \qquad 1 \le i, j \le n, \tag{4}$$

is nonsingular. In fact, it is known that a sufficient condition to have nonsingularity is that the corresponding matrix is positive definite. Thus, if A is a positive definite matrix, then all its eigenvalues are positive and A is nonsingular (see, e.g., [3]). Note that here we do not add any polynomial term to the radial basis function as in general one can do in RBF interpolation.

Definition 2.2. Let $\mathcal{X}_n = \{x_i\}_{i=1}^n$ be a set of n distinct data points on $\Omega \subseteq \mathbb{R}^N$. A continuous function $\phi : [0, \infty) \to \mathbb{R}$ is called *positive definite* of order n on Ω, if

$$\sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j \phi(d(x_i, x_j)) \ge 0, \tag{5}$$

for any $a = [a_1, a_2, \ldots, a_n]^T \in \mathbb{R}^n$. The function ϕ is called *strictly positive definite* of order n if the quadratic form (5) is zero only for $a \equiv 0$. If ϕ is strictly positive definite for any n, then it is called strictly positive definite.

Then, if ϕ is strictly positive definite, the interpolant (2) is unique, since the corresponding interpolation matrix (4) is positive definite and hence nonsingular.

An example of strictly positive definite CSRBFs is given by Wendland's functions (see [6]). Some of the most commonly used functions for $N = 2$ (that are also strictly positive definite and radial on $\mathbb{R}^N$, $N \le 3$) are here listed along with their degree of smoothness

$$\begin{array}{ll} \phi_1(r) = (1 - cr)_+^4 \, (4cr + 1), & \text{(Wendland's } C^2 \text{ function)} \\ \phi_2(r) = (1 - cr)_+^6 \, \left(35c^2r^2 + 18cr + 3\right), & \text{(Wendland's } C^4 \text{ function)} \end{array} \tag{6}$$

where $(\cdot)_+$ denotes the truncated power function, $c \in \mathbb{R}^+$ is a shape parameter, and $r = ||x - x_i||_2$. These functions are nonnegative for $r \in [0, 1/c]$, and are zero for $r > 1/c$.

3. Partition of unity method.

The partition of unity method was suggested in [12,13] in the mid 1990s in the context of meshfree Galerkin methods for the solution of partial differential equations (PDEs), even if now it is also an effective method for fast computation in the field of approximation theory (see, e.g, [1,3,6,11]).

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

$$\sum_{j=1}^{d} W_j(x) = 1, \qquad x \in \Omega. \tag{7}$$

Then, for each cell Ω_j we consider a compactly supported radial basis function R_j as local approximant and form the global approximant given by

$$\mathcal{I}(x) = \sum_{j=1}^{d} R_j(x) W_j(x), \qquad x \in \Omega. \tag{8}$$

Note that if the local approximants satisfy the interpolation conditions at data point x_i, i.e.

$$R_j(x_i) = f(x_i), \tag{9}$$

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

$$\mathcal{I}(x_i) = f(x_i), \qquad \text{for } i = 1, 2, \ldots, n. \tag{10}$$

In order to be able to formulate error bounds we need some technical conditions. Then, we require the partition of unity functions W_j to be *k-stable*, i.e. each $W_j \in C^k(\mathbb{R}^N)$, $j = 1, 2, \ldots, d$, and for every multi-index $\alpha \in \mathbb{N}_0^m$ with $|\alpha| \leq k$ there exists a constant $C_\alpha > 0$ such that

$$||D^\alpha W_j||_{L_\infty(\Omega_j)} \leq C_\alpha / \delta_j^{|\alpha|}, \tag{11}$$

where $\delta_j = \mathrm{diam}(\Omega_j)$.

DOI: 10.1685/journal.caim.431

In accordance with the statements in [11] we require some additional regularity assumptions on the *covering* $\{\Omega_j\}_{j=1}^d$.

Definition 3.1. Suppose that $\Omega \subseteq \mathbb{R}^N$ is bounded and $\mathcal{X}_n = \{x_i\}_{i=1}^n \subseteq \Omega$ are given. An open and bounded covering $\{\Omega_j\}_{j=1}^d$ is called *regular* for $(\Omega, \mathcal{X}_n)$ if the following properties are satisfied:

(a) for each $x \in \Omega$, the number of cells Ω_j with $x \in \Omega_j$ is bounded by a global constant K;
(b) each cell Ω_j satisfies an interior cone condition (see, e.g., [6]);
(c) 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 $C_\nu^k(\mathbb{R}^N)$ of all functions $f \in C^k$ whose derivatives of order $|\alpha| = k$ satisfy $D^\alpha f(x) = \mathcal{O}(||x||_2^\nu)$ for $||x||_2 \to 0$, we consider the following convergence result (see, e.g., [3,6] and references therein).

Theorem 3.1. *Let $\Omega \subseteq \mathbb{R}^N$ be open and bounded and suppose that $\mathcal{X}_n = \{x_i\}_{i=1}^n \subseteq \Omega$. Let $\phi \in C_\nu^k(\mathbb{R}^N)$ be strictly conditionally positive definite function of order m. Let $\{\Omega_j\}_{j=1}^d$ be a regular covering for $(\Omega, \mathcal{X}_n)$ and let $\{W_j\}_{j=1}^d$ be k-stable for $\{\Omega_j\}_{j=1}^d$. Then the error between $f \in \mathcal{N}_\phi(\Omega)$, where $\mathcal{N}_\phi$ is the native space of ϕ, and its partition of unity interpolant (8) can be bounded by*

$$|D^\alpha f(x) - D^\alpha \mathcal{I}(x)| \leq C h_{\mathcal{X}_n,\Omega}^{(k+\nu)/2-|\alpha|} |f|_{\mathcal{N}_\phi(\Omega)}, \tag{12}$$

for all $x \in \Omega$ and all $|\alpha| \leq k/2$.

Note that the partition of unity preserves the local approximation order for the global fit. Hence, we can efficiently compute large CSRBF interpolants by solving small CSRBF interpolation problems and then combine them together with the global partition of unity $\{W_j\}_{j=1}^d$. This approach enables us to decompose a large problem into many small problems, ensuring that the accuracy obtained for the local fits is carried over to the global fit.

4. Partition of unity algorithm.

In this section we present the partition of unity algorithm for bivariate interpolation of scattered data lying on the domain $\Omega = [0,1] \times [0,1] \subset \mathbb{R}^2$. This technique is based on the partition of Ω in a suitable number of strips, the construction of a number of d cells Ω_j such that $\Omega \subseteq \bigcup_{j=1}^d \Omega_j$ with some mild overlap among the cells, and then the employment of an

optimized strip searching procedure. Finally, the partition of unity method is combined with CSRBFs.

For simplicity, we subdivide the description of the algorithm in three parts, namely distribution, localization and evaluation phases. We remark that only one strip structure is used for both localization and evaluation phases.

4.1. *Distribution phase.*

Let us now consider a set $\mathcal{X}_n = \{(x_i, y_i)\}_{i=1}^n$ of data points, a set $\mathcal{F}_n = \{f_i\}_{i=1}^n$ of data values, a set $\mathcal{C}_d = \{(\bar{x}_i, \bar{y}_i)\}_{i=1}^d$ cell points, and a set $\mathcal{E}_s = \{(\tilde{x}_i, \tilde{y}_i)\}_{i=1}^s$ of evaluation points.

Initially, the set $\mathcal{X}_n$ of nodes is ordered with respect to a common direction (e.g. the y-axis), by applying a *quicksort$_y$ procedure*. Then, for each cell point $(\bar{x}_i, \bar{y}_i)$, $i = 1, 2, \ldots, d$, a local circular cell is constructed, whose half-size (the radius) depends on the cell number d, that is

$$\delta_{cell} = \sqrt{\frac{2}{d}}. \tag{13}$$

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

After the number of strips to be considered is found taking

$$q = \left\lceil \frac{1}{\delta_{cell}} \right\rceil, \tag{14}$$

the strips are numbered from 1 to q. Such a choice follows directly from the side length of the domain Ω, that here is 1, and the cell radius δ_{cell}.

Now, the following three structures are considered:

- a suitable family of q strips of equal width $\delta_{strip} \equiv \delta_{cell}$ (with possible exception of one of them) on the unit square and parallel to the x-axis is constructed;
- the set $\mathcal{X}_n$ of nodes is partitioned by the strip structure into q subsets $\mathcal{X}_{n_k}$, whose elements are (x_{k1}, y_{k1}), (x_{k2}, y_{k2}), ..., (x_{kn_k}, y_{kn_k}), $k = 1, 2, \ldots, q$.
- the set $\mathcal{C}_d$ of cell points is partitioned by the strip structure into q subsets $\mathcal{C}_{d_k}$, whose elements are $(\bar{x}_{k1}, \bar{y}_{k1})$, $(\bar{x}_{k2}, \bar{y}_{k2})$, ..., $(\bar{x}_{kd_k}, \bar{y}_{kd_k})$, $k = 1, 2, \ldots, q$.

4.2. *Localization phase.*

In order to identify the strips to be examined in the searching procedure, we consider two steps as follows:

DOI: 10.1685/journal.caim.431

(i) Since $\delta_{strip} \equiv \delta_{cell}$, the ratio between these quantities is denoted by $i^* = \delta_{cell}/\delta_{strip} = 1$. So the number $j^* = 2i^* + 1$ of strips to be examined for each node is 3.

(ii) For each strip k, $k = 1, 2, \ldots, q$, a strip searching procedure is considered, examining the nodes from the strip $k - i^*$ to the strip $k + i^*$. Note that if $k - i^* < 1$ or $k + i^* > q$ it will be assign $k - i^* = 1$ and $k + i^* = q$.

After defining the strips to be examined, a strip searching procedure is applied for each cell point of $\mathcal{C}_{d_k}$, $k = 1, 2, \ldots, q$, to determine all nodes belonging to a cell. The number of nodes of the cell centered at $(\bar{x}_i, \bar{y}_i)$ is counted and stored in m_i, $i = 1, 2, \ldots, d$.

Thus, a local interpolant R_j, $j = 1, 2, \ldots, d$, is constructed for each cell point.

4.3. *Evaluation phase.*

The set $\mathcal{E}_s$ of evaluation points is ordered with respect to a common direction (e.g. the y-axis), by applying a *quicksort$_y$ procedure*. Then, the set $\mathcal{E}_s$ is partitioned into q subsets $\mathcal{E}_{p_k}$, $k = 1, 2, \ldots, q$, so that the evaluation points of $\mathcal{E}_{p_k}$ belong to the k-th strip.

A strip searching procedure is applied for each evaluation point of $\mathcal{E}_{p_k}$, $k = 1, 2, \ldots, q$, in order to find all those belonging to a cell of center $(\bar{x}_i, \bar{y}_i)$ and radius δ_{cell}. The number of cells containing the i-th evaluation point is counted and stored in r_i, $i = 1, 2, \ldots, s$.

Thus, a local approximant $R_j(x, y)$ and a weight function $W_j(x, y)$, $j = 1, 2, \ldots, d$, is found for each evaluation point. Finally, applying the global fit (8), the surface can be approximated at any evaluation point $(x, y) \in \mathcal{E}_s$.

5. Complexity analysis.

The partition of unity algorithm involves the use of the standard sorting routine quicksort, which requires on average a complexity $\mathcal{O}(m \log m)$, where m is the number of nodes to be sorted. Specifically, we have a distribution phase consisting of building the data structure, in which the computational cost has 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. Moreover, in order to compute the local CSRBF interpolants, we have to solve d linear systems of small dimensions, thus requiring a computational cost of order $\mathcal{O}(m_i^3)$, $i = 1, 2, \ldots, d$, for each cell, where m_i is the number of nodes in the i-th cell. Finally, for the k-th evaluation point of $\mathcal{E}_s$ we also need a cost of $r_k \cdot \mathcal{O}(m_i)$, $i = 1, 2, \ldots, d$, $k = 1, 2, \ldots, s$, where r_k is the number of cells containing the k-th evaluation point.

6. Numerical experiments.

6.1. *Results on test cases.*

In this subsection we report some tests to verify performance and effectiveness of the partition of unity algorithm on scattered data sets. The code is implemented in C language, while numerical results are carried out on a Intel Core 2 Duo Computer (2.1 GHz). In the experiments we consider three sets of scattered data points consisting of $n = (2^k + 1)^2$, $k = 6, 7, 8$, Halton nodes [14]. The partition of unity algorithm is run considering $d = 4^j$, $j = 5, 6, 7$, cell points and $s = 33 \times 33$ evaluation (grid) points, which are contained in the unit square $\Omega = [0, 1] \times [0, 1]$. Here, for the global interpolant (8) we use Shepard's weight.

The performance of the algorithm is verified taking the data values by the following two test functions (see, e.g., [7])

$$f_1(x, y) = \frac{1}{2} y \cos^4 \left[4 \left(x^2 + y - 1 \right) \right],$$
$$f_2(x, y) = 2 \cos(10x) \sin(10y) + \sin(10xy).$$

Since we are interested in analyzing effectiveness of the proposed algorithm, in Table 1 we report CPU times (in seconds) obtained by running the partition of unity algorithm.

Table 1. CPU times (in seconds) obtained by running the partition of unity algorithm.

n	d	CPU times
4225	1024	0.359
16641	4096	1.250
66049	16384	6.453

Then, to test accuracy of the local algorithm, in Tables 2 and 3 we report the root mean square errors (RMSEs) computed on the two considered test functions. Error computation is achieved by considering CSRBFs ϕ_1 and ϕ_2, taking a good value of the shape parameter, i.e., $c = 1$. In fact, in Figures 1 and 2 we analyze the behaviour of the RMSEs by varying the shape parameters c of ϕ_1 and ϕ_2 in the interval $[0.1, 2]$ for the test function f_1; similar results were also obtained for f_2. From this analysis we observe that Wendland's functions C^2 and C^4 are stable for any value of c, enabling us a sort of "free choice" of the related shape parameter. This property makes Wendland's functions suited for application, where it is usually easier to have instability problems.

DOI: 10.1685/journal.caim.431

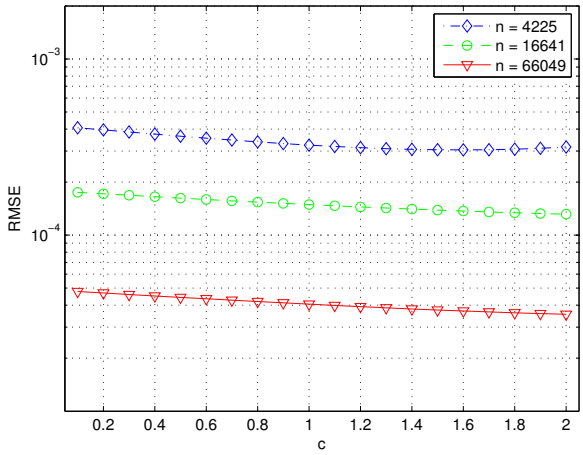

Figure 1. RMSEs obtained by varying the shape parameter of ϕ_1 for f_1.

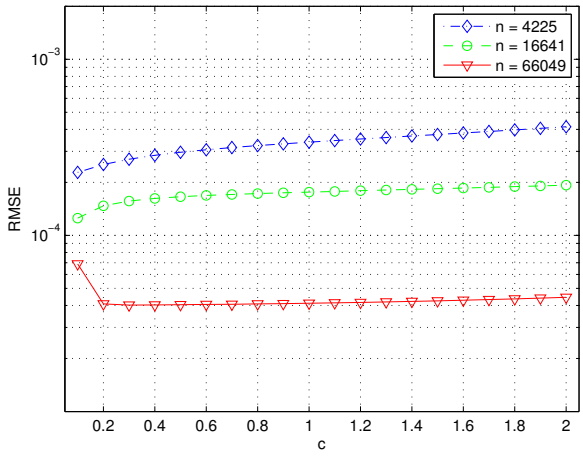

Figure 2. RMSEs obtained by varying the shape parameter of ϕ_2 for f_1.

Table 2. RMSEs obtained by CSRBFs with $c = 1$ for f_1.

n	4225	16641	66049
ϕ_1	3.2439E − 4	1.4893E − 4	4.0539E − 5
ϕ_2	3.3841E − 4	1.7614E − 4	4.1192E − 5

Finally, we compare accuracy and efficiency of the partition of unity algorithm with the modified Shepard's algorithm presented in [7]. Both

Table 3. RMSEs obtained by CSRBFs with $c = 1$ for f_2.

n	4225	16641	66049
ϕ_1	1.7701E − 3	4.4138E − 4	1.1060E − 4
ϕ_2	1.0402E − 3	2.8310E − 4	4.4449E − 5

algorithms are based on the partition of the domain in strips and the use of a strip-based searching procedure in the process of node localization. Numerical tests reported in Table 4 show comparable results in accuracy and higher efficiency of the partition of unity algorithm then the modified Shepard's algorithm, the latter being run taking $n_L = 16$ and $n_W = 13$ as localization parameters (see [7]). See e.g. also [2] for a comparison with the proposed algorithm.

Table 4. Comparison of RMSEs and CPU times (in seconds) obtained by applying the partition of unity algorithm and the modified Shepard's algorithm in [7] with $c = 0.7$ for f_1.

n		Partition of unity algorithm		Modified Shepard's algorithm	
		RMSE	t_{sec}	RMSE	t_{sec}
4225	ϕ_1	3.4642E − 4	0.36	3.3540E − 4	1.31
	ϕ_2	3.1528E − 4	0.36	5.8690E − 4	1.31
16641	ϕ_1	1.5651E − 4	1.25	1.3801E − 4	3.02
	ϕ_2	1.7078E − 4	1.25	2.7634E − 4	3.02
66049	ϕ_1	4.2711E − 5	6.45	3.0298E − 5	10.31
	ϕ_2	4.0663E − 5	6.45	6.3643E − 5	10.52

6.2. *Application to Earth's topography.*

In this subsection we consider an application to Earth's topography [15]. The case concerns the use of topography data by MATLAB, which is available from the National Geophysical Data Center, NOAA US Department of Commerce under data announcement 88-MGG-02.

The data set is stored in a MAT file, called `topo.mat`, while the variable `topo` contains the altitude data for the Earth. A representation on the plane of this data is given in Figure 3. Specifically, in this case we have 64800 topography data, and among them we randomly select 64700 nodes for the interpolation process, only reserving the remaining 100 points for the cross-validation. It is used to evaluate the approximation results, and the performance of partition of unity algorithm, comparing the predicted

DOI: 10.1685/journal.caim.431

values with the original ones. In order to have a reliable result on the error, it is more appropriate to use relative (or normalized) errors, such as the Relative Root Mean Square Errors (RRMSEs).

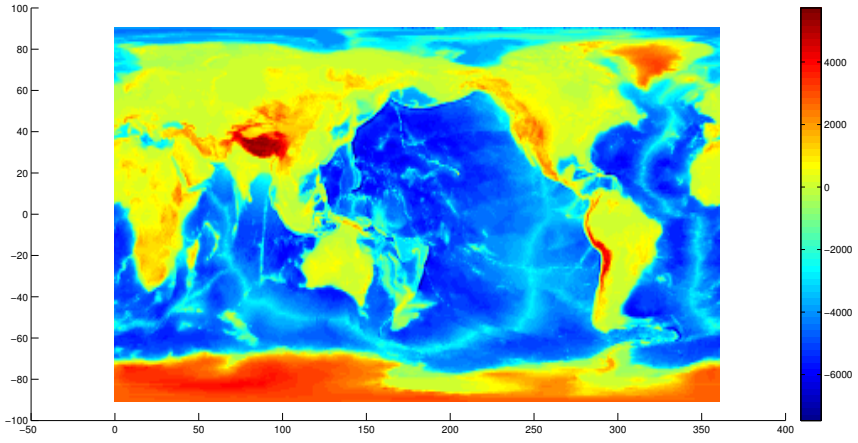

Figure 3. Earth's topography.

Thus, in Table 5 we report the RRMSEs obtained by varying the shape parameter, taking $c = 0.5, 1.0, 1.5, 2.0$. These results point out the goodness of our approach, also in applications where one has to deal with real data.

Table 5. RRMSEs obtained by using the partition of unity algorithm on Earth's topography data.

c	0.5	1.0	1.5	2.0
ϕ_1	$4.3685\text{E}-3$	$4.3672\text{E}-3$	$4.3658\text{E}-3$	$4.3643\text{E}-3$
ϕ_2	$6.0764\text{E}-3$	$6.0468\text{E}-3$	$6.0502\text{E}-3$	$6.0543\text{E}-3$

7. Conclusions and future work.

In this paper we present a local algorithm for scattered data interpolation in a two-dimensional domain. It is based on the partition of unity method and can efficiently be used also when the amount of data to be interpolated is quite large. In particular, this efficient implementation of the partition of unity method is carried out by applying an optimized nearest neighbour searching procedure. Then, an application to Earth's topography shows as our approach might also be employed successfully in various situations of real life.

As research and future work we expect to refine the proposed algorithm considering a double structure of crossed strips, which should allow us to produce a remarkable reduction of the computational cost thanks to the use of an effective cell-based searching procedure. Moreover, an interesting approach could be to develop an adaptive algorithm which allows us to suitably increase the sizes of the cells when it turns out to be necessary: for example, if the node distribution is not uniform or when the number of data points falling in any cell is not sufficient to obtain a good approximation result. Finally, if one wanted to use radial basis functions such as Gaussians, multiquadrics, inverse multiquadrics etc., which are basically more accurate than Wendland's functions but usually very ill-conditioned, application of preconditioning techniques (see, e.g., [16]) might be of sure interest.

Acknowledgements.

The author thanks the anonymous referee for the detailed and valuable comments.

REFERENCES

1. M. D. Buhmann, *Radial basis functions: theory and implementation.* Cambridge Univ. Press, 2003.
2. J. B. Cherrie, R. K. Beatson, and G. N. Newsam, Fast evaluation of radial basis functions: methods for generalized multiquadrics in $\mathbb{R}^n$, *SIAM J. Sci. Comput.*, vol. 23, pp. 1549–1571, 2002.
3. G. E. Fasshauer, *Meshfree approximation methods with MATLAB.* World Scientific Publishers, 2007.
4. A. Iske, Radial basis functions: basics, advanced topics and meshfree methods for transport problems, *Rend. Sem. Mat. Univ. Pol. Torino*, vol. 61, pp. 247–285, 2003.
5. A. Iske, Scattered data approximation by positive definite kernel functions, *Rend. Sem. Mat. Univ. Pol. Torino*, vol. 69, pp. 217–246, 2011.
6. H. Wendland, *Scattered data approximation.* Cambridge Univ. Press, 2005.
7. G. Allasia, R. Besenghi, R. Cavoretto, and A. De Rossi, Scattered and track data interpolation using an efficient strip searching procedure, *Appl. Math. Comput.*, vol. 217, pp. 5949–5966, 2011.
8. R. Cavoretto and A. De Rossi, Fast and accurate interpolation of large scattered data sets on the sphere, *J. Comput. Appl. Math.*, vol. 234, pp. 1505–1521, 2010.
9. R. Cavoretto, *Meshfree approximation methods, algorithms and applications.* PhD thesis, Dept. Mathematics, University of Turin, 2010.

DOI: 10.1685/journal.caim.431

10. R. Cavoretto and A. De Rossi, Spherical interpolation using the partition of unity method: an efficient and flexible algorithm, *Appl. Math. Lett.*, vol. 25, pp. 1251–1256, 2012.

11. H. Wendland, Fast evaluation of radial basis functions: methods based on partition of unity, in *Approximation theory X: wavelets, splines, and applications* (C. K. Chui, L. L. Schumaker, and J. Stöckler, eds.), pp. 473–483, Vanderbilt Univ. Press, 2002.

12. I. Babuška and J. M. Melenk, The partition of unity method, *Internat. J. Numer. Methods. Engrg.*, vol. 40, pp. 727–758, 1997.

13. J. M. Melenk and I. Babuška, The partition of unity finite element method: basic theory and applications, *Comput. Methods. Appl. Mech. Engrg.*, vol. 139, pp. 289–314, 1996.

14. T.-T. Wong, W.-S. Luk, and P.-A. Heng, Sampling with Hammersley and Halton points, *J. Graphics Tools*, vol. 2, pp. 9–24, 1997.

15. Earth Topography Data, The MathWorks, *http://www.mathworks.com/products/matlab/demos.html?file=/products/demos/shipping/matlab/earthmap.html*.

16. R. Cavoretto, A. De Rossi, M. Donatelli, and S. Serra-Capizzano, Spectral analysis and preconditioning techniques for radial basis function collocation matrices, *Numer. Linear Algebra Appl.*, vol. 19, pp. 31–52, 2012.