SIGRAD 2016
M. Hayashi (Editors)

A Radial Basis Function Approximation for Large Datasets

Z. Majdisova[1] and V. Skala[1]

[1]Department of Computer Science and Engineering, Faculty of Applied Sciences, University of West Bohemia, Univerzitni 8, CZ 30614 Plzen, Czech Republic

Abstract

Approximation of scattered data is often a task in many engineering problems. The Radial Basis Function (RBF) approximation is appropriate for large scattered datasets in d-dimensional space. It is non-separable approximation, as it is based on a distance between two points. This method leads to a solution of overdetermined linear system of equations.

In this paper a new approach to the RBF approximation of large datasets is introduced and experimental results for different real datasets and different RBFs are presented with respect to the accuracy of computation. The proposed approach uses symmetry of matrix and partitioning matrix into blocks.

Categories and Subject Descriptors (according to ACM CCS): G.1.2 [Numerical Analysis]: Approximation—Approximation of Surfaces and Contours

1. Introduction

Interpolation and approximation are the most frequent operations used in computational techniques. Several techniques have been developed for data interpolation or approximation, but they mostly expect an ordered dataset, e.g. rectangular mesh, structured mesh, unstructured mesh etc. However, in many engineering problems, data are not ordered and they are scattered in $d-$dimensional space, in general. Usually, in technical applications the conversion of a scattered dataset to a semi-regular grid is performed using some tessellation techniques. However, this approach is quite prohibitive for the case of $d-$dimensional data due to the computational cost.

Interesting techniques are based on the Radial Basis Function (RBF) method which was originally introduced by [Har71]. They are widely used across of many fields solving technical and non-technical problems. The RBF applications can be found in neural networks, data visualization [PRF14], surface reconstruction [CBC*01], [TO02], [PS11], [SPN13], [SPN14], solving partial differential equations [LCC13], [HSfY15], etc. The RBF techniques are really meshless and are based on collocation in a set of scattered nodes. These methods are independent with respect to the dimension of the space. The computational cost of this techniques increase nonlinearly with the number of points in the given dataset and linearly with the dimensionality of data.

There are two main groups of basis functions: global RBFs and Compactly Supported RBFs (CS-RBFs) [Wen06]. Fitting scattered data with CS-RBFs leads to a simpler and faster computation, but techniques using CS-RBFs are sensitive to the density of scattered data. Global RBFs lead to a linear system of equations with a dense matrix and their usage is based on sophisticated techniques such as the fast multipole method [Dar00]. Global RBFs are useful in repairing incomplete datasets and they are insensitive to the density of scattered data.

For the processing of scattered data we can use the RBF interpolation or the RBF approximation. The RBF interpolation, e.g. presented by [Ska15], is based on a solution of a linear system of equations:

$$\mathbf{Ac} = \mathbf{h}, \tag{1}$$

where $\mathbf{A}$ is a matrix of this system, $\mathbf{c}$ is a column vector of variables and $\mathbf{h}$ is a column vector containing the right sides of equations. In this case, $\mathbf{A}$ is an $N \times N$ matrix, where N is the number of points in the given scattered dataset, the variables are weights for basis functions and the right sides of equations are values in the given points. The disadvantage of RBF interpolation is the large and usually ill-conditioned matrix of the linear system of equations. Moreover, in the case of an oversampled dataset or intended reduction, we want to reduce the given problem, i.e. reduce the number of weights and used basis functions, and preserve good preci-

sion of the approximated solution. The approach which includes the reduction is called the RBF approximation. In the following section, the method recently introduced in [Ska13] is described in detail. This approach requires less memory and offer higher speed of computation than the method using Lagrange multipliers [Fas07]. Further, a new approach to RBF approximation of large datasets is presented in the Section 3. These approach uses symmetry of matrix and partitioning matrix into blocks.

2. RBF Approximation

For simplicity, we assume that we have an unordered dataset $\{\mathbf{x}_i\}_1^N \in E^2$. However, this approach is generally applicable for d-dimensional space. Further, each point $\mathbf{x}_i$ from the dataset is associated with a vector $\mathbf{h}_i \in E^p$ of the given values, where p is the dimension of the vector, or scalar value, i.e. $h_i \in E^1$. For an explanation of the RBF approximation, let us consider the case when each point $\mathbf{x}_i$ is associated with a scalar value h_i, e.g. a $2^1/_2D$ surface. Let us introduce a set of new reference points $\{\boldsymbol{\xi}_j\}_1^M$, see Figure 1.

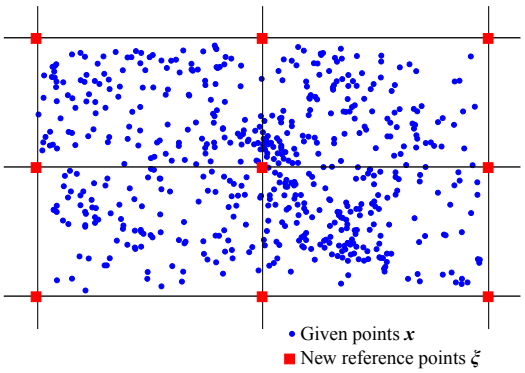


Figure 1: *The RBF approximation and reduction of points.*

These reference points may not necessarily be in a uniform grid. It is appropriate that their placement reflects the given surface (e.g. the terrain profile, etc.) as well as possible. The number of reference points $\boldsymbol{\xi}_j$ is M, where $M \ll N$. Now, the RBF approximation is based on the distance computation of the given point $\mathbf{x}_i$ and the reference point $\boldsymbol{\xi}_j$.

The approximated value is determined similarly as for interpolation (see [Ska15]):

$$f(\mathbf{x}) = \sum_{j=1}^{M} c_j \phi(r_j) = \sum_{j=1}^{M} c_j \phi(\|\mathbf{x} - \boldsymbol{\xi}_j\|), \tag{2}$$

where $\phi(r_j)$ is a used RBF centered at point $\boldsymbol{\xi}_j$ and the approximating function $f(\mathbf{x})$ is represented as a sum of these RBFs, each associated with a different reference point $\boldsymbol{\xi}_j$, and weighted by a coefficient c_j which has to be determined.

It can be seen that we get an overdetermined linear system of equations for the given dataset:

$$\begin{aligned} h_i = f(\mathbf{x}_i) &= \sum_{j=1}^{M} c_j \phi(\|\mathbf{x}_i - \boldsymbol{\xi}_j\|) \\ &= \sum_{j=1}^{M} c_j \phi_{i,j} \qquad i = 1, \ldots, N. \end{aligned} \tag{3}$$

The linear system of equations (3) can be represented in a matrix form as:

$$\mathbf{A}\mathbf{c} = \mathbf{h}, \tag{4}$$

where the number of rows is $N \gg M$ and M is the number of unknown weights $[c_1, \ldots, c_M]^T$, i.e. the number of reference points. Equation (4) represents system of linear equations:

$$\begin{pmatrix} \phi_{1,1} & \cdots & \phi_{1,M} \\ \vdots & \ddots & \vdots \\ \phi_{i,1} & \cdots & \phi_{i,M} \\ \vdots & \ddots & \vdots \\ \phi_{N,1} & \cdots & \phi_{N,M} \end{pmatrix} \begin{pmatrix} c_1 \\ \vdots \\ c_M \end{pmatrix} = \begin{pmatrix} h_1 \\ \vdots \\ h_i \\ \vdots \\ h_N \end{pmatrix}. \tag{5}$$

The presented system is overdetermined, i.e. the number of equations N is higher than the number of variables M. This linear system of equations can be solved by the least squares method as $\mathbf{A}^T\mathbf{A}\mathbf{c} = \mathbf{A}^T\mathbf{h}$ or singular value decomposition, etc.

3. RBF Approximation for Large Data

In practice, the real datasets contain a large number of points which results into high memory requirements for storing the matrix $\mathbf{A}$ of the overdetermined linear system of equations (5). For example when we have dataset contains $3,000,000$ points, number of reference points is $10,000$ and double precision floating point is used then we need 223.5 GB memory for storing the matrix $\mathbf{A}$ of the overdetermined linear system of equations (5). Unfortunately, we do not have an unlimited capacity of RAM memory and therefore calculation of unknown weights c_j for RBF approximation would be prohibitively computationally expensive due to memory swapping, etc. In this section, a proposed solution to this problem is described.

In Section 2, it was introduced that overdetermined system of equations can be solved by the least squares method. For this method the $M \times M$ square matrix:

$$\mathbf{B} = \mathbf{A}^T\mathbf{A} \tag{6}$$

is to be determined. Advantages of matrix $\mathbf{B}$ are that it is a symmetric matrix and moreover only two vectors of length N are needed te determine of one entry, i.e.:

$$b_{ij} = \sum_{k=1}^{N} \phi_{ki} \cdot \phi_{kj}, \tag{7}$$

where b_{ij} is the entry of the matrix $\mathbf{B}$ in the $i-$th row and $j-$th column.

To save memory requirements and data bus (PCI) load block operations with matrices are used. Based on the above properties of the matrix $\mathbf{B}$, only the upper triangle of this matrix is computed. Moreover the matrix is partitioned into $M_B \times M_B$ blocks, see Figure 2, and the calculation is performed sequentially for each block:

$$\mathbf{B}_{kl} = (\mathbf{A}_{*,k})^T (\mathbf{A}_{*,l})$$
$$k = 1, \ldots, \frac{M}{M_B}, \qquad l = k, \ldots, \frac{M}{M_B}, \tag{8}$$

where $\mathbf{B}_{kl}$ is sub-matrix in the $k-$th row and $l-$th column and $\mathbf{A}_{*,k}$ is defined as:

$$\mathbf{A}_{*,k} = \begin{pmatrix} \phi_{1,(k-1)\cdot M_B+1} & \cdots & \phi_{1,k\cdot M_B} \\ \vdots & \ddots & \vdots \\ \phi_{i,(k-1)\cdot M_B+1} & \cdots & \phi_{i,k\cdot M_B} \\ \vdots & \ddots & \vdots \\ \phi_{N,(k-1)\cdot M_B+1} & \cdots & \phi_{N,k\cdot M_B} \end{pmatrix}. \tag{9}$$

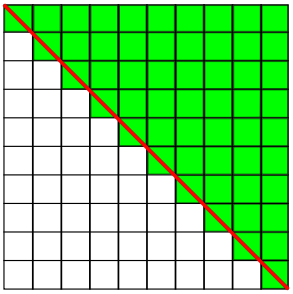


Figure 2: *$M \times M$ square matrix which is partitioned into $M_B \times M_B$ blocks. Main diagonal of matrix is represented by red color and illustrates the symmetry of matrix. Blocks, which must be computed, are represented by green color.*

The size of block M_B is chosen so that M_B is multiple of M and there is no swapping, i.e.:

$$M_B \cdot (M_B + 2 \cdot N) \cdot prec < \text{size of RAM [B]}, \tag{10}$$

where *prec* is size of data type in bytes.

4. Experimental results

The presented modification of the RBF approximation method has been tested on synthetic and real data. Let us introduce results for two real datasets.

The first dataset was obtained from LiDAR data of the Serpent Mound in Adams Country, Ohio[†]. The second dataset is LiDAR data of the Mount Saint Helens in Skamania Country, Washington[†]. Each point of these datasets is associated with its elevation. Summary of the dimensions of terrain for the given datasets is in Table 1.

Table 1: *Summary of the dimensions of terrain for tested datasets. Note that one feet [ft] corresponds to 0.3048 meter [m].*

Dimensions	Serpent Mound	St. Helens
number of points	3,265,110	6,743,176
lowest point [ft]	166.7800	3,191.5269
highest point [ft]	215.4800	8,330.2219
width [ft]	1,085.1199	26,232.3696
length [ft]	2,698.9601	35,992.6861

For experiments, two different radial basis functions have been used, see Table 2. Shape parameters α for used RBFs were determined experimentally with regard to the quality of approximation and they are presented in Table 3. Note that value of shape parameter α is inversely proportional to range of datasets.

Table 2: *Used RBFs*

RBF	type	$\phi(\boldsymbol{r})$
Gaussian RBF	global	$e^{-(\alpha r)^2}$
Wendland's $\phi_{3,1}$	local	$(1-\alpha r)_+^4(4\alpha r+1)$

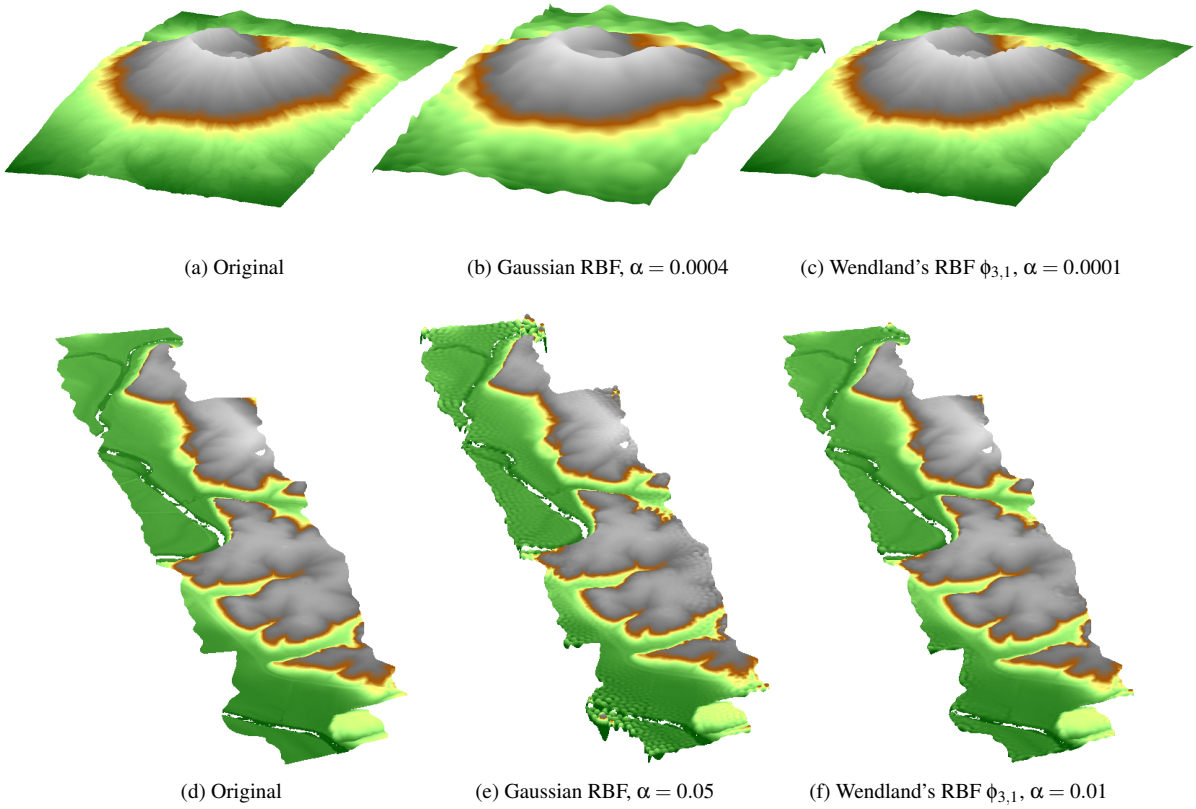
Table 3: *Experimentally determined shape parameters α for used RBFs*

RBF	shape parameter	
	Serpent Mound	**St. Helens**
Gaussian RBF	$\alpha = 0.05$	$\alpha = 0.0004$
Wendland's $\phi_{3,1}$	$\alpha = 0.01$	$\alpha = 0.0001$

The set of reference points equals the subset of the given dataset for which we determine the RBF approximation. Moreover, the distribution of reference points is uniform and the set of reference points has a cardinality 10,000 in both experiments.

Approximation of Mount Saint Helens for both BRFs and its original are shown in Figure 3a-3c. In Figure 3b can be seen that the RBF approximation with the global Gaussian RBFs cannot preserve the sharp rim of a crater. Further, visualization of magnitude of error at each point of the original points cloud is presented in Figure 4 and Figure 5. It can be seen that the RBF approximation with the global Gaussian

[†] http://www.liblas.org/samples/



(a) Original (b) Gaussian RBF, $\alpha = 0.0004$ (c) Wendland's RBF $\phi_{3,1}$, $\alpha = 0.0001$

(d) Original (e) Gaussian RBF, $\alpha = 0.05$ (f) Wendland's RBF $\phi_{3,1}$, $\alpha = 0.01$

Figure 3: *Serpent Mound in Adams Country, Ohio (top) and Mount Saint Helens is Skamania Country, Washington (bottom)*

RBFs returns worse result than RBF approximation with local Wendland's $\phi_{3,1}$ basis functions in terms of the error. In Table 4 can be seen the value of mean absolute error, its deviation and mean relative error for both approximations.

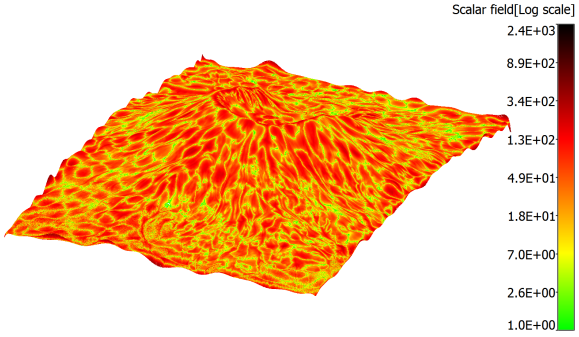


Figure 4: *Approximation of Mount Saint Helens with* $10,000$ *global Gaussian basis functions with shape parameter* $\alpha = 0.0004$ *false-colored by magnitude of error.*

Results of the RBF approximation for Serpent Mound and its original are shown in Figure 3d-3f. It can be seen that

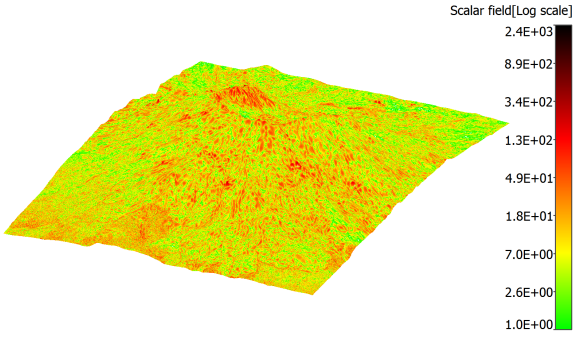


Figure 5: *Approximation of Mount Saint Helens with* $10,000$ *local Wendland's* $\phi_{3,1}$ *basis functions with shape parameter* $\alpha = 0.0001$ *false-colored by magnitude of error.*

the approximation using local Wendland's $\phi_{3,1}$ basis function (Figure 3f) returns again better result than approximation using the global Gaussian RBF (Figure 3e) in terms of the error. It is also seen in Figure 6 and Figure 7 where magnitude of error at each point of original points cloud is visualized. Moreover, we can see that the highest errors occur

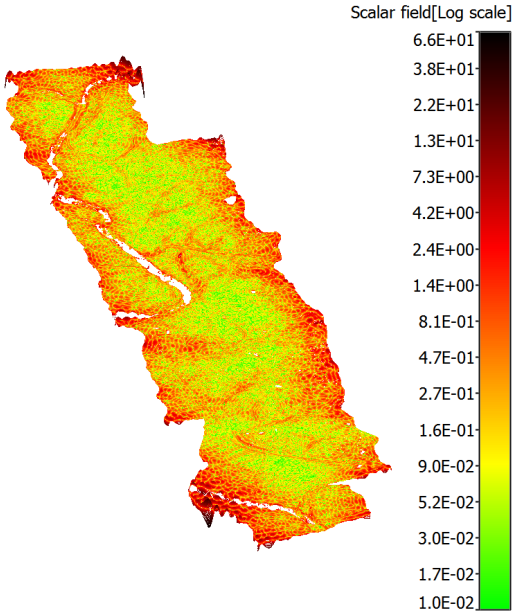


Figure 6: *Approximation of the Serpent Mound with* $10,000$ *global Gaussian basis functions with shape parameter* $\alpha = 0.05$ *false-colored by magnitude of error.*

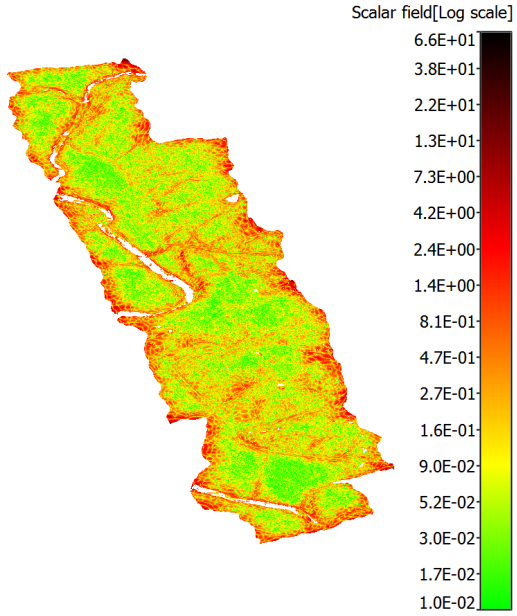


Figure 7: *Approximation of the Serpent Mound with* $10,000$ *local Wendland's* $\phi_{3,1}$ *basis functions with shape parameter* $\alpha = 0.01$ *false-colored by magnitude of error.*

on the boundary of terrain, which is a general problem of RBF methods. Value of mean absolute error, its deviation and mean relative error due to elevation for both used RBFs are again mentioned in Table 4.

Mutual comparison both datasets in terms of the mean relative error (Table 4) indicates that mean relative error for Serpent Mount is smaller than for Mount Saint Helens. It is caused by the presence of vegetation, namely forest, in LiDAR data of the Mount Saint Helens. This vegetation operates in our RBF approximation as noise and therefore the resulting mean relative error is higher.

The implementation of the RBF approximation has been performed in Matlab and tested on PC with the following configuration:

- CPU: Intel® Core™ i7-4770 (4× 3.40GHz + hyper-threading),
- memory: 32 GB RAM,
- operating system Microsoft Windows 7 64bits.

For the approximation of the Serpent Mound with $10,000$ local Wendland's $\phi_{3,1}$ basis function with shape parameter $\alpha = 0.01$ the running times for different sizes of blocks were measured. These times were converted relative to the time for 100×100 blocks and are presented in Figure 8. We can see that for the approximation matrix which is partitioned into small blocks (i.e. smaller than 25×25 blocks) the time performance is large. This is caused by overhead costs. On the other hand, for the approximation matrix which is partitioned into large blocks (i.e. larger than 125×125 blocks) the running time begins to grow above the permissible limit due to memory swapping.

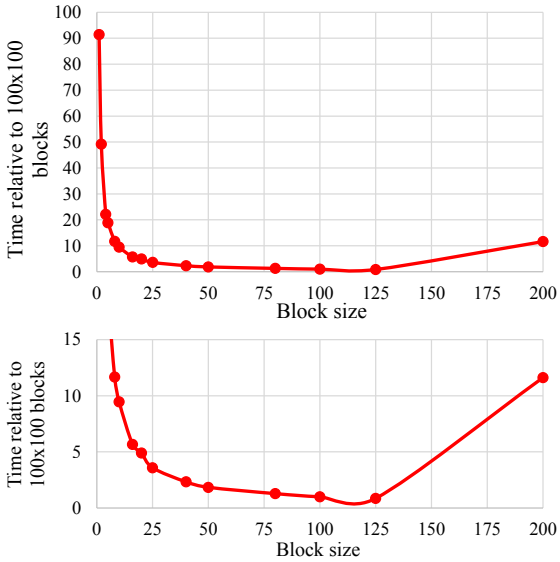


Figure 8: *Time performance for approximation of the Serpent Mound depending on the block size. The times are presented relative to the time for* 100×100 *blocks.*

Table 4: *The RBF approximation error for testing datasets and different radial basis functions. Note that one feet [ft] corresponds to 0.3048 meter [m].*

Error	Serpent Mound		St. Helens	
	Gaussian RBF	**Wendland's $\phi_{3,1}$**	**Gaussian RBF**	**Wendland's $\phi_{3,1}$**
mean absolute error [ft]	0.4477	0.2289	44.4956	12.1834
deviation of error [ft]	1.4670	0.1943	680.3659	169.2800
mean relative error [%]	0.0024	0.0012	0.0087	0.0023

5. Conclusions

This paper presents a new approach to the RBF approximation of large datasets. The proposed approach uses symmetry of matrix and partitioning matrix into blocks, thus preventing memory swapping. The experiments made proved that the proposed approach is able to determine the RBF approximation for large dataset. Moreover, from the experimental results we can see that use of a local RBFs is better than global RBFs, if data are sufficiently sampled. Futher, it is obvious that approximation using the global Gaussian RBFs has problems with the preservation of sharp edges. The experiments made also proved that RBF methods have problems with the accuracy of calculation on the boundary of an object, which is a well known property, and the magnitude of the RBF approximation error is influenced by the presence of a noise.

For the future work, the RBF approximation method can be explored in terms of lower sensitivity to noise, more accurate calculation on the boundary or better approximation of sharp edges and improvements of the computational cost without loss of approximation accuracy.

Acknowledgments

The authors would like to thank their colleagues at the University of West Bohemia, Plzen, for their discussions and suggestions, and also anonymous reviewers for the valuable comments and suggestions they provided. The research was supported by MSMT CR projects LH12181 and SGS 2016-013.

References

[CBC*01] CARR J. C., BEATSON R. K., CHERRIE J. B., MITCHELL T. J., FRIGHT W. R., MCCALLUM B. C., EVANS T. R.: Reconstruction and representation of 3d objects with radial basis functions. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 2001, Los Angeles, California, USA, August 12-17, 2001* (2001), pp. 67–76. 1

[Dar00] DARVE E.: The fast multipole method: Numerical implementation. *Journal of Computational Physics 160*, 1 (2000), 195–240. 1

[Fas07] FASSHAUER G. E.: *Meshfree Approximation Methods with MATLAB*, vol. 6. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 2007. 2

[Har71] HARDY R. L.: Multiquadratic Equations of Topography and Other Irregular Surfaces. *Journal of Geophysical Research 76* (1971), 1905–1915. 1

[HSfY15] HON Y.-C., SARLER B., FANG YUN D.: Local radial basis function collocation method for solving thermo-driven fluid-flow problems with free surface. *Engineering Analysis with Boundary Elements 57* (2015), 2 – 8. {RBF} Collocation Methods. 1

[LCC13] LI M., CHEN W., CHEN C.: The localized {RBFs} collocation methods for solving high dimensional {PDEs}. *Engineering Analysis with Boundary Elements 37*, 10 (2013), 1300 – 1304. 1

[PRF14] PEPPER D. W., RASMUSSEN C., FYDA D.: A meshless method using global radial basis functions for creating 3-d wind fields from sparse meteorological data. *Computer Assisted Methods in Engineering and Science 21*, 3-4 (2014), 233–243. 1

[PS11] PAN R., SKALA V.: A two-level approach to implicit surface modeling with compactly supported radial basis functions. *Eng. Comput. (Lond.) 27*, 3 (2011), 299–307. 1

[Ska13] SKALA V.: Fast Interpolation and Approximation of Scattered Multidimensional and Dynamic Data Using Radial Basis Functions. *WSEAS Transactions on Mathematics 12*, 5 (2013), 501–511. 2

[Ska15] SKALA V.: Meshless interpolations for computer graphics, visualization and games. In *Eurographics 2015 - Tutorials, Zurich, Switzerland, May 4-8, 2015* (2015), Zwicker M., Soler C., (Eds.), Eurographics Association. 1, 2

[SPN13] SKALA V., PAN R., NEDVED O.: Simple 3d surface reconstruction using flatbed scanner and 3d print. In *SIGGRAPH Asia 2013, Hong Kong, China, November 19-22, 2013, Poster Proceedings* (2013), ACM, p. 7. 1

[SPN14] SKALA V., PAN R., NEDVED O.: Making 3d replicas using a flatbed scanner and a 3d printer. In *Computational Science and Its Applications - ICCSA 2014 - 14th International Conference, Guimarães, Portugal, June 30 - July 3, 2014, Proceedings, Part VI* (2014), vol. 8584 of *Lecture Notes in Computer Science*, Springer, pp. 76–86. 1

[TO02] TURK G., O'BRIEN J. F.: Modelling with implicit surfaces that interpolate. *ACM Trans. Graph. 21*, 4 (2002), 855–873. 1

[Wen06] WENDLAND H.: Computational aspects of radial basis function approximation. *Studies in Computational Mathematics 12* (2006), 231–256. 1