Computational Science and Its Application, ICSSA 2019 proceedings, Part I, LNCS 11619, pp.337-350, ISSN 0302-9743
ISBN 978-3-030-24288-6, Springer, 2019 https://doi.org/10.1007/978-3-030-24289-3_25

Efficient Simple Large Scattered 3D Vector Fields Radial Basis Functions Approximation Using Space Subdivision

Michal Smolik* and Vaclav Skala

Faculty of Applied Sciences, University of West Bohemia,
Plzen, Czech Republic
{smolik,skala}@kiv.zcu.cz

Abstract. The Radial basis function (RBF) approximation is an efficient method for scattered scalar and vector data fields. However its application is very difficult in the case of large scattered data. This paper presents RBF approximation together with space subdivision technique for large vector fields.

For large scattered data sets a space subdivision technique with overlapping $3D$ cells is used. Blending of overlapped $3D$ cells is used to obtain continuity and smoothness. The proposed method is applicable for scalar and vector data sets as well. Experiments proved applicability of this approach and results with the tornado large vector field data set are presented.

Keywords: Vector field; Radial basis functions; critical point; tornado; simplification; approximation; space subdivision; data compression; visualization.

1 Introduction

Interpolation or approximation methods of scattered $3D$ vector field data mostly use tessellation of the given domain, i.e. triangulation or tetrahedranization, etc. Space subdivision techniques are often used to increase speed-up and decrease memory requirements in combination of adaptive hierarchical methods, i.e. quadtree, octree etc. However, the Radial Basis Functions (RBF) is not a separable (by dimension) approximation. In general, the meshless methods mostly based on RBF

Data are split into subdomains, processed and blended together with partition of unity in [28]. The contribution [28] is an extension of well-known method [16], which construct surface model from large data sets using multi-level partition of unity. Downsampling [17] leads to a coarse-fine hierarchy, where points in each hierarchy level are used incrementally for better approximation. Parallel

* The research was supported by projects Czech Science Foundation (GACR) No. GA17-05534S and partially by SGS 2019-016.

Computational Science and Its Application, ICSSA 2019 proceedings, Part I, LNCS 11619, pp.337-350, ISSN 0302-9743
ISBN 978-3-030-24288-6, Springer, 2019 https://doi.org/10.1007/978-3-030-24289-3_25

version of this approach [29] claims $O(N)$ computational complexity using generalized minimal residual method (GMRE) with the Schwartz iterative method [3]. Optimization of centers and weights of RBF methods was explored in [25] with combination of hierarchical decomposition. There are many other related modifications of RBF approximation with a specific focus available, e.g. parallelism of [7] for mesh deformation, incremental RBF interpolation [1], computation of RBF with Least square error [12] with preconditioning aspects and domain decomposition.

The method for topological information visualization for vector fields is well known [11]. The vector fields are very complex data sets and the topological skeleton represents a compact visualization. The vector field topology can be simplified using [26]. This approach computes clusters of critical points, where the distance is represented by the weight of merging critical points. The critical points in one cluster are merged together and can create a higher order critical point or cancel each other. The method generates the piece-wise linear representation after building clusters containing singularities. The paper [27] presents an approach for simplified visualization of vector fields. The authors prove that the $3D$ vector field inside some closed region can be represented by the $2D$ vector field on the surface over this region. The vector filed that uses the Delaunay triangulation is described in [4]. It removes vertices from the Delaunay triangulation close to critical points and prevents topological changes using local metric while removing some vertices. Numerical comparison between global and local RBF methods was explored in [2] to find out the advantages and disadvantages of local RBF methods use for $3D$ vector field approximation. The classification of critical points using Hessian matrix is presented in [21]. Vector field approximation for the $2D$ case preserving topology and memory reduction was presented in [10]. It is based on segmentation and flow in a separate region is approximated by a linear function. The paper [24] and [23] proposes an approach for RBF approximation of vector field and selection of important critical points. Robust detection of critical points is described in [20].

We propose a new simple and robust approach for large scattered $3D$ vector fields data approximation using space subdivision. Usually, the whole data set needs to be processed at once [14], [13]. Other relevant methods are not easy to implement. Using the space subdivision methods with respecting the continuity of the resulting approximation, the proposed approach enables to process large data vector fields.

2 Proposed Approach

The $3D$ vector field data sets come usually from numerical simulations and are very large. Such vector fields can be approximated for the visualization purposes or to minimize the data set size. In our proposed approach for approximation of $3D$ vector fields we use modified algorithm described in [22], which computes $2D$ interpolation of height data sets.

Computational Science and Its Application, ICSSA 2019 proceedings, Part I, LNCS 11619, pp.337-350, ISSN 0302-9743
ISBN 978-3-030-24288-6, Springer, 2019 https://doi.org/10.1007/978-3-030-24289-3_25

In the following part we introduce a new approach for large $3D$ vector field data approximation using RBF and space subdivision respecting continuity of the final approximation result. Space subdivision application leads to significant computational speed-up, decrease of memory requirements and better robustness of computation, too.

There are three main steps of the algorithm: space subdivision, data approximation of each cell and blending, i.e. joining approximations over overlapping cells. The Algorithm 1 and 2 present relevant pseudocodes.

Algorithm 1 Pseudocode of the proposed approach for RBF approximation.

```
procedure RBF(Points P)                                  ▷ $P_i = \{\boldsymbol{x}_i, \boldsymbol{v}_i\}$
    for all cells in grid do
        Enlarge cell for approximation by $\Psi$
        $p \leftarrow$ Points in enlarged cell
        $\xi \leftarrow$ RBF centers in enlarged cell
        Compute RBF approximation of $p$
```

Algorithm 2 Pseudocode of approximated value calculation using the proposed RBF approximation method.

```
procedure RBF(Point p)                                   ▷ $p = \{x, y, z\}$
    Find neighboring cells
    Determine distances to cells
    Compute approximated RBF values for all cells
    Blend RBF approximated values together               ▷ using distances to cells
```

2.1 Space Subdivision

The divide and conquer (D&C) strategy is used in the proposed algorithm. The input data set is divided into several domains. In this paper for simplicity of explanation, we use a rectangular grid for divide and conquer strategy, where the grid size for $3D$ data set is $n \times m \times l$. We can use any kind of space division, however the proposed approach is easy to explain sung the regular orthogonal grid and thus it was used in the presented experiments for its simplicity.

The given data need to be splitted into overlapping cells respecting the created grid for application of the space subdivision. Each domain of the grid is enlarged to a cell which includes some neighboring points from the neighborhood domains (it will be explained latter on), see Fig. 1.

Computational Science and Its Application, ICSSA 2019 proceedings, Part I, LNCS 11619, pp.337-350, ISSN 0302-9743
ISBN 978-3-030-24288-6, Springer, 2019 https://doi.org/10.1007/978-3-030-24289-3_25

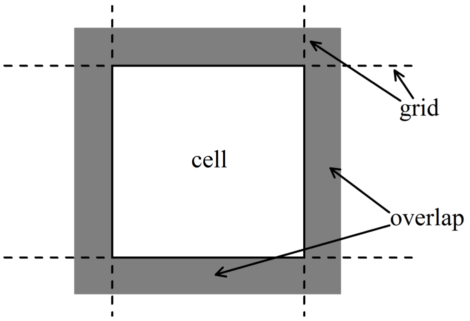


Fig. 1. $3D$ regular orthogonal grid ($2D$ analogy) of one cell. Each cell has points that are inside the domain plus pints from the overlapping parts (grey color).

2.2 Cells RBF Approximation

In the proposed approach, we use the "global" Thin Plate Spline (TPS) radial basis function, which is shape parameter free and minimizes the tension of the final approximation [5]. The TPS has the following formula

$$\varphi(r) = r^2 \log r = \frac{1}{2} r^2 \log r^2 \tag{1}$$

Now, the given points are splited into overlapping $3D$ cells. The RBF approximation needs the centers of radial basis functions. The RBF centers have the Halton distribution [8] and are placed inside the enlarged cell. The number of centers for RBF approximation of each cell can be selected according to the required quality of approximation.

Points inside of a cell are approximated using the RBF approximation with the TPS function. This approximation uses the standard solution of the linear system of equations (2). Each cell is approximated independently and therefore the computation can be done totally in parallel, which increases the performance and speed-up, too. However, the memory requirements would be higher as multiple RBF matrices need to be stored simultaneously. This should be considered when determining the size of a grid for space subdivision.

$$\boldsymbol{v}_i = \boldsymbol{v}(\boldsymbol{x}_i) = \sum_{j=1}^{M} \boldsymbol{\lambda}_j \varphi(\|\boldsymbol{x}_i - \boldsymbol{\xi}_j\|), \qquad \text{for } \forall i \in \{1, \dots, N\} \tag{2}$$

where $\boldsymbol{v}_i = [v_i^{(x)}, v_i^{(y)}, v_i^{(z)}]$, M is the number of the RBF centers. Solution of the linear system of equations is a vector $\boldsymbol{\lambda} = [\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \dots, \boldsymbol{\lambda}_M]^T$, where $\boldsymbol{\lambda}_i = [\lambda_i^x, \lambda_i^y, \lambda_i^z]^T$. These values will be used later. However, the matrix for the RBF computation can be discarded as it will not be needed any more.

Computational Science and Its Application, ICSSA 2019 proceedings, Part I, LNCS 11619, pp.337-350, ISSN 0302-9743
ISBN 978-3-030-24288-6, Springer, 2019 https://doi.org/10.1007/978-3-030-24289-3_25

2.3 Reconstruction Function and Cells Blending

The already computed approximated cells overlap. To get the final continuous representation of the $3D$ vector field, we need to join the RBF approximations of cells.

The RBF approximation usually has problems with a precision on a border [15], [19] and thus we cannot use the whole enlarged cell for blending. The overlapping part of each border is Ψ. For the blending phase we will use only half of the overlapping part, see the blue part in Fig. 2, Therefore the size of this overlapping part is ψ, i.e. $(2\psi = \Psi)$.

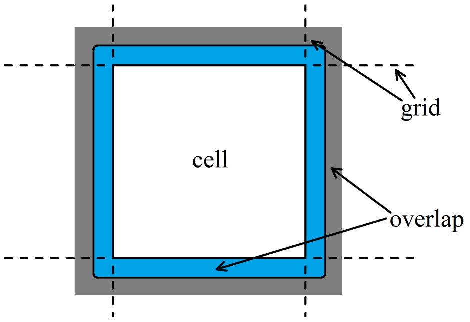


Fig. 2. Visualization of the overlap part used for blending (blue color).

To blend all the neighborhood cell together, we use modified trilinear interpolation ("blending") of those neighborhood cells. The computed value obtained for each cell is to be weighted by a coefficient α. The coefficients α are determined as

$$\alpha' = \left[1 - min\left(1, \frac{distance\ from\ the\ border}{\psi}\right)\right]^2, \tag{3}$$

where *distance from the border* is the shortest distance from the location to the border using the Euclidean metric. The final blending coefficients α_i are computed using Eq. (3) as

$$\alpha_i = \frac{\alpha_i{}'}{\sum_{j=1}^{2^k} \alpha_j{}'}, \tag{4}$$

where $i = \{1, \ldots, 2^k\}$ and k is the dimension, i.e. $k = 3$ for $3D$ vector field data set. The visualization of blending functions for blending of two approximations can be seen in Fig. 3. The initial and the final phase of blending function is more attracted to value 0, resp. 1, thus the final approximation is more smooth.

After computing the proposed RBF approximation with space subdivision and blending, we end up with an analytical form of the approximated vector field.

Computational Science and Its Application, ICSSA 2019 proceedings, Part I, LNCS 11619, pp.337-350, ISSN 0302-9743
ISBN 978-3-030-24288-6, Springer, 2019   https://doi.org/10.1007/978-3-030-24289-3_25

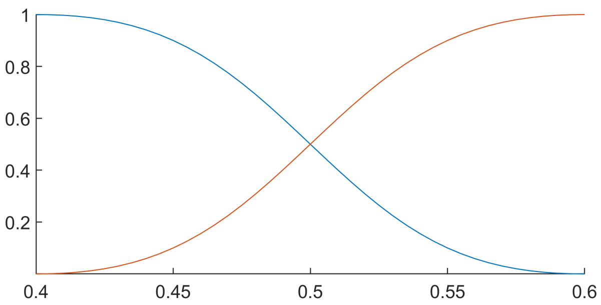


Fig. 3. Blending functions for blending of two approximations.

This vector field is the simplified representation of the original data set. Moreover the analytical formula of the vector field can be used for further processing and visualization.

2.4 Speed-up of the Proposed Approach (Approximation)

The RBF approximation has actually two parts. First, the RBF coefficients computation. And second, computation of the function value for the given position $\boldsymbol{x}$.

The space subdivision is used to speed-up the computation of vector field radial basis function approximation, i.e. computation of $\boldsymbol{\lambda}$ values, and reduces memory requirements, too.

The asymptotic time complexity of solving overdetermined system of linear equations with QR decomposition [6] and Householder matrix transformation [9] is

$$O\left(2NM^2 - \frac{2}{3}M^3\right), \tag{5}$$

where N is the total number of input points, M is the number of centers for RBF and $N > M$.

Let us assume that the input vector field data set has an uniform distribution of points and the input vector field is divided into G cells. The best size of overlapping part was experimentally selected as $\Psi = 30\%$, i.e. $\psi = 15\%$. The smaller overlapping part can result in non-smooth blending and larger overlapping part will result in higher computation costs while the approximation quality will not increase much more.

The number of points inside the enlarged cell is different depending on the location of the cell. In Fig. 4 are visualized 3 different type of cells, when the cells with the same color have the same number of points inside the enlarged cell. There is one more group of cells, that has the same number of points inside the enlarged cell. This group of cells is inside the cube visualized in Fig. 4. In

Computational Science and Its Application, ICSSA 2019 proceedings, Part I, LNCS 11619, pp.337-350, ISSN 0302-9743
ISBN 978-3-030-24288-6, Springer, 2019 https://doi.org/10.1007/978-3-030-24289-3_25

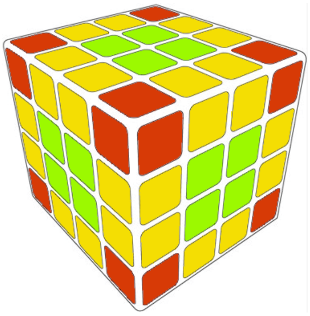


Fig. 4. Visualization of different type of cells according to the number of points inside the enlarged cell.

our computations of time complexity, we will assume, that the number of points inside each enlarged cell is the same and is equal to

$$n = (1 + 2\Psi)^3 \frac{N}{G}, \tag{6}$$

where G is the total number of cells and n is the number of points inside the enlarged cell. The constant Ψ is the size of overlapping parts.

The proposed RBF approximation method time complexity can be estimated as:

$$O\left(G\left(2nm^2 - \frac{2}{3}m^3\right)\right), \tag{7}$$

where m is the number of centers for RBF approximation. The value of m is calculated as

$$m = n\frac{M}{N}. \tag{8}$$

The speed-up of the proposed algorithm for vector field RBF approximation compared to the standard RBF approximation is

$$\nu = \frac{O\left(2NM^2 - \frac{2}{3}M^3\right)}{O\left(G\left(2nm^2 - \frac{2}{3}m^3\right)\right)} = \frac{G^3(1-3N)}{(1+2\Psi)^9\left(G - 3N\left(1+2\Psi\right)^3\right)}, \tag{9}$$

where Ψ is the size of overlapping parts. For large values of N, i.e. $N > 10^6$, the expected speed-up is given as Eq. (10) and the visualization of speed-up is in Fig. 5.

$$\nu \approx \frac{G^3}{(1+2\Psi)^{12}}. \tag{10}$$

An example of the speed-up for the size of overlapping 30% is as the following

$$\nu \approx \frac{G^3}{(1+2\cdot 0.3)^{12}} = \frac{G^3}{1.6^{12}} \approx \frac{G^3}{281}. \tag{11}$$

Computational Science and Its Application, ICSSA 2019 proceedings, Part I, LNCS 11619, pp.337-350, ISSN 0302-9743
ISBN 978-3-030-24288-6, Springer, 2019 https://doi.org/10.1007/978-3-030-24289-3_25

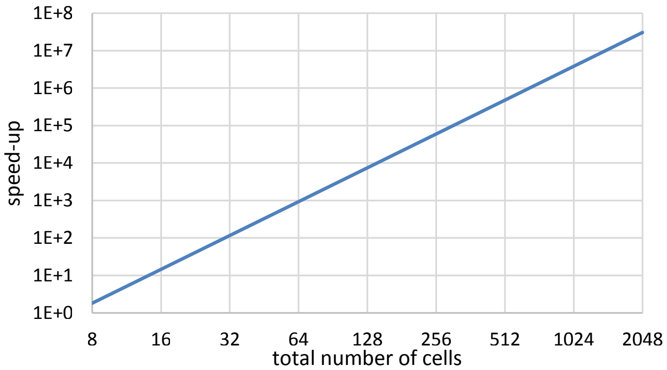


Fig. 5. Expected speed-up of the proposed approach of vector field RBF approximation compared to the standard one (note that the axes are logarithmic).

2.5 Speed-up of the Function Evaluation

In this part we present how the function evaluation speed-up the vector field RBF approximation computation. Moreover, it also speed-up the evaluation of the approximation function as well. For the standard RBF function evaluation, the time complexity can be estimated as:

$$O\left(M\right). \tag{12}$$

In the case of the proposed algorithm, the time of RBF evaluation can be estimated as:

$$O\left(2^3 m\right), \tag{13}$$

where the maximum number of blended approximations is 2^3, i.e. 8. Using Eqs. (12) and (13), we can determine the theoretical speed-up of the proposed method for evaluation of one function value of the vector field RBF approximation:

$$\eta = \frac{O\left(M\right)}{O\left(2^3 m\right)} = O\left(\frac{G}{2^3\left(1+2\Psi\right)^3}\right), \tag{14}$$

where Ψ is the size of overlapping parts. For most grid resolutions, i.e. number of cells, the speed-up $\eta \gg 1$, is shown in Fig. 6. Note that the η axis, i.e. speed-up, is in logarithmic scaling.

3 Experimental Results

In this part we present experimental results. The proposed $3D$ vector field RBF approximation is especially convenient for large vector field data set approximation. Firstly we test the algorithm using small synthetic data sets to present and prove properties of the proposed approximation method.

Computational Science and Its Application, ICSSA 2019 proceedings, Part I, LNCS 11619, pp.337-350, ISSN 0302-9743
ISBN 978-3-030-24288-6, Springer, 2019 https://doi.org/10.1007/978-3-030-24289-3_25

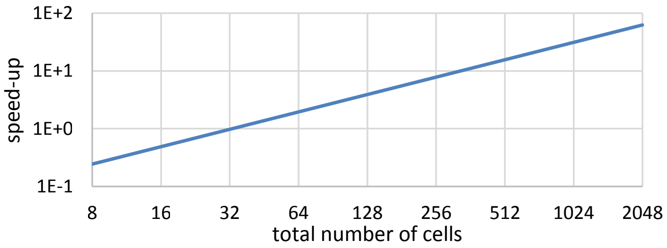


Fig. 6. Expected speed-up of function evaluation of the proposed approach for vector field RBF approximation compared to the standard one (note that the axes are logarithmic).

Secondly, the experimental results with real data sets containing $5.5 \cdot 10^8$ points are presented. Experiments proved that the proposed method is capable to process significantly larger data on a desktop computer.

3.1 Synthetic Data Set

Firstly, we tested the blending of two $1\frac{1}{2}D$ simple functions together to verify expected properties of the proposed approach. We used two blending functions from Fig. 3 and performed the blending on two functions that are visualized in Fig. 7. The two functions are blended in interval [0.4; 0.6] and the result is visualized in Fig. 7.

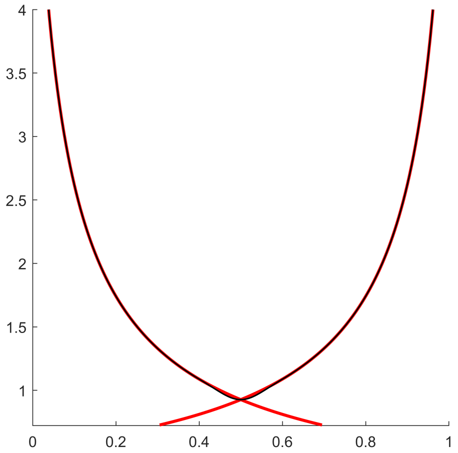
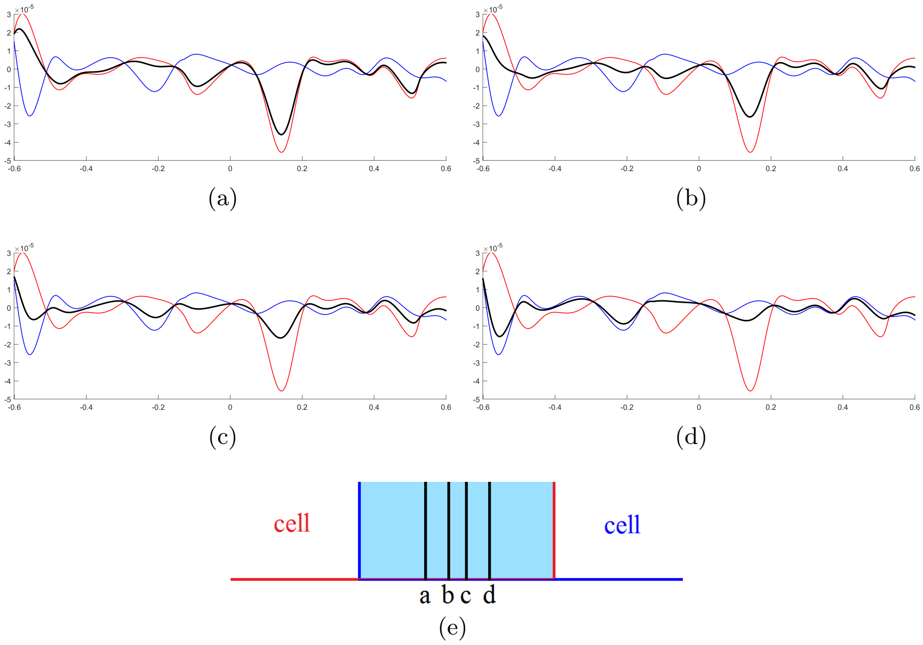


Fig. 7. Blending of two functions (red) and the result after blending (black).

Computational Science and Its Application, ICSSA 2019 proceedings, Part I, LNCS 11619, pp.337-350, ISSN 0302-9743
ISBN 978-3-030-24288-6, Springer, 2019 https://doi.org/10.1007/978-3-030-24289-3_25

Secondly, we tested the blending of two $2\frac{1}{2}D$ functions together. The result of blending two $2\frac{1}{2}D$ functions together at different locations is visualized at Fig. 8. It can be seen that the blending result is continuous and smooth, as expected.



(a)

(b)

(c)

(d)

(e)

Fig. 8. Blending of two $2\frac{1}{2}D$ functions together. Visualization of blending for different "cut" of $2\frac{1}{2}D$ function (a-d), see (e) for "cut" location.

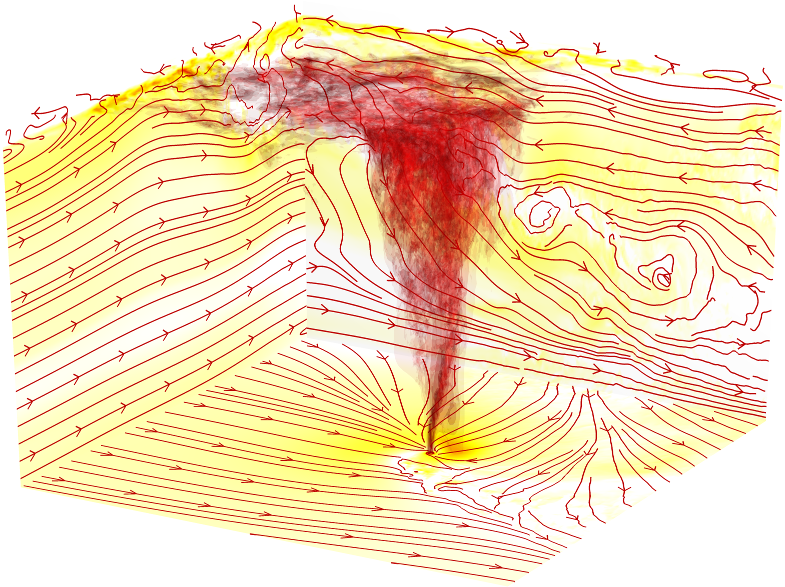
3.2 Real Data Set

In these experiments, we used the EF5 tornado data set (from [18])[1], see Fig. 9a. The data set contains $5.5 \cdot 10^8$ $3D$ points with associated $3D$ vector.

We computed the vector field approximation using the proposed approach with different number of centers for radial basis functions. The vector field RBF approximation when using only 0.1% of the number of input points as the number of RBF centers is visualized in Fig. 9b. It means, that the vector field approximation is visually almost identical with the original vector field data set even thought a high compression ratio $(1 : 10^3)$ is achieved. Visualization of $2D$ slices is visualized in Fig. 10. Again, the approximated vector field is almost identical with the original vector field.

[1] Data set of EF5 tornado courtesy of Leigh Orf from Cooperative Institute for Meteorological Satellite Studies, University of Wisconsin, Madison, WI, USA.

Computational Science and Its Application, ICSSA 2019 proceedings, Part I, LNCS 11619, pp.337-350, ISSN 0302-9743
ISBN 978-3-030-24288-6, Springer, 2019 https://doi.org/10.1007/978-3-030-24289-3_25



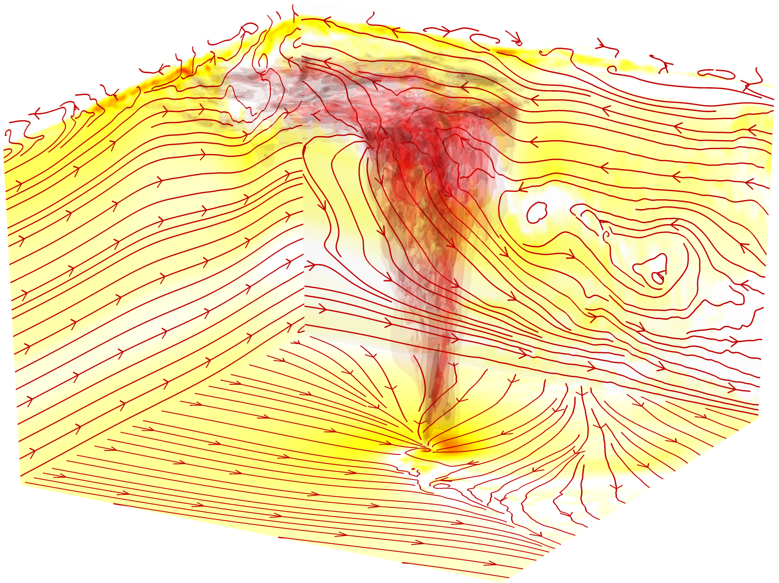


Fig. 9. Visualization of the $3D$ tornado vector field data set. Red central part represents the shape of tornado vortex and the yellow color on faces represents the speed of vector field. The original vector field (top) and the RBF approximated vector field (bottom).

Computational Science and Its Application, ICSSA 2019 proceedings, Part I, LNCS 11619, pp.337-350, ISSN 0302-9743
ISBN 978-3-030-24288-6, Springer, 2019 https://doi.org/10.1007/978-3-030-24289-3_25

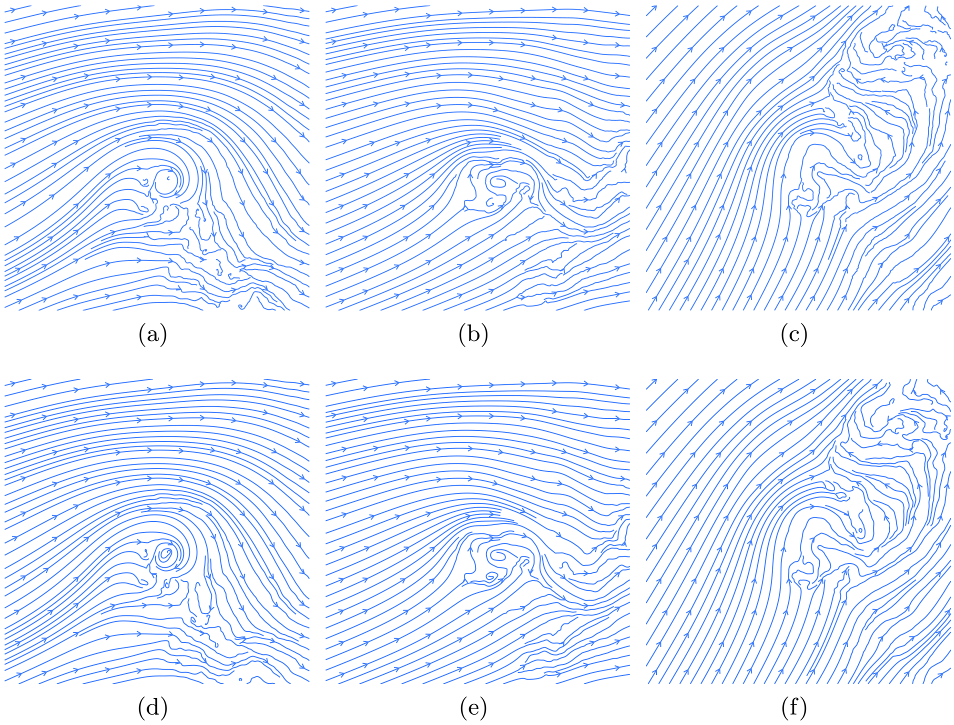


Fig. 10. Visualization of three $2D$ vector field slices. The top row represents the original vector field (a-c) and the bottom row represents the approximated vector field (d-f).

The approximation error for different number of centers for radial basis functions is visualized in Fig. 11. The approximation error is computed using the formula

$$Err = \frac{\sum_{i=1}^{N} \|\boldsymbol{v}_i - \bar{\boldsymbol{v}}_i\|}{\sum_{i=1}^{N} \|\bar{\boldsymbol{v}}_i\|}, \tag{15}$$

where $\bar{\boldsymbol{v}}_i$ is the original vector, $\boldsymbol{v}_i$ is the approximated vector and N is the number of vectors.

This experiments also proved expected precision depending on the number of centers of the RBF approximation.

4 Conclusion

We presented a new approach for large scale $3D$ vector field meshless approximation using RBF. The method significantly speeds-up the RBF parameters calculation, i.e. $\boldsymbol{\lambda}$ values, and the final RBF evaluation as well.

Computational Science and Its Application, ICSSA 2019 proceedings, Part I, LNCS 11619, pp.337-350, ISSN 0302-9743
ISBN 978-3-030-24288-6, Springer, 2019 https://doi.org/10.1007/978-3-030-24289-3_25

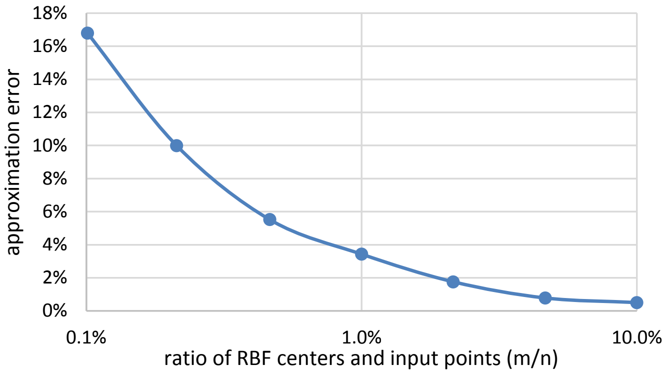


Fig. 11. The approximation error for different number of RBF centers.

The proposed approximation method is based on partially overlapping cells. These overlapping cells are continuously blended together in order to obtain approximation of the whole large data set. Due to the space subdivision, the approach decreases memory and computational requirements. The proposed algorithm can be parallelized easily as well.

Experiments made on synthetic and real data proved high performance and computational robustness. The result of the proposed is an analytical description of simplified $3D$ vector field. This is very useful in further processing of the vector field and visualization as well.

Acknowledgments

The authors would like to thank their colleagues at the University of West Bohemia, Plzen, for their discussions and suggestions. The research was supported by projects Czech Science Foundation (GACR) No. GA17-05534S and partially by SGS 2019-016.

References

1. R. K. Beatson, W. A. Light, and S. D. Billings. Fast solution of the radial basis function interpolation equations: Domain decomposition methods. *SIAM J. Scientific Computing*, 22(5):1717–1740, 2001.
2. D. A. C. Cabrera, P. Gonzalez-Casanova, C. Gout, L. H. Juárez, and L. R. Reséndiz. Vector field approximation using radial basis functions. *Journal of Computational and Applied Mathematics*, 240:163–173, 2013.
3. X.-C. Cai and M. Sarkis. A restricted additive Schwarz preconditioner for general sparse linear systems. *SIAM Journal on scientific computing*, 21(2):792–797, 1999.
4. T. K. Dey, J. A. Levine, and R. Wenger. A delaunay simplification algorithm for vector fields. In *Computer Graphics and Applications, 2007. PG'07. 15th Pacific Conference on*, pages 281–290. IEEE, 2007.

Computational Science and Its Application, ICSSA 2019 proceedings, Part I, LNCS 11619, pp.337-350, ISSN 0302-9743
ISBN 978-3-030-24288-6, Springer, 2019 https://doi.org/10.1007/978-3-030-24289-3_25

5. J. Duchon. Splines minimizing rotation-invariant semi-norms in sobolev spaces. In *Constructive theory of functions of several variables*, pages 85–100. Springer, 1977.
6. G. H. Golub and C. F. Van Loan. *Matrix computations*, volume 3. JHU Press, 2012.
7. G. Haase, D. Martin, and G. Offner. Towards RBF interpolation on heterogeneous HPC systems. In *Large-Scale Scientific Computing - 10th International Conference, LSSC 2015*, pages 182–190, 2015.
8. J. H. Halton. Algorithm 247: Radical-inverse quasi-random point sequence. *Communications of the ACM*, 7(12):701–702, 1964.
9. A. S. Householder. Unitary triangularization of a nonsymmetric matrix. *Journal of the ACM (JACM)*, 5(4):339–342, 1958.
10. S. Koch, J. Kasten, A. Wiebel, G. Scheuermann, and M. Hlawitschka. 2D vector field approximation using linear neighborhoods. *The Visual Computer*, 32(12):1563–1578, 2016.
11. R. S. Laramee, H. Hauser, L. Zhao, and F. H. Post. Topology-based flow visualization, the state of the art. In *Topology-based methods in visualization*, pages 1–19. Springer, 2007.
12. L. Ling and E. J. Kansa. Preconditioning for radial basis functions with domain decomposition methods. *Mathematical and Computer Modelling*, 40(13):1413–1427, 2004.
13. Z. Majdisova and V. Skala. A radial basis function approximation for large datasets. In *Proceedings of SIGRAD 2016*, number 127, pages 9–14. 2016.
14. Z. Majdisova and V. Skala. Big geo data surface approximation using radial basis functions: A comparative study. *Computers & Geosciences*, 109:51–58, 2017.
15. Z. Majdisova and V. Skala. Radial basis function approximations: Comparison and applications. *Applied Mathematical Modelling*, 51:728–743, 2017.
16. Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, and H.-P. Seidel. Multi-level partition of unity implicits. In *ACM Siggraph 2005 Courses*, pages 463–470. ACM, 2005.
17. Y. Ohtake, A. G. Belyaev, and H. Seidel. A multi-scale approach to 3D scattered data interpolation with compactly supported basis function. In *2003 International Conference on Shape Modeling and Applications (SMI 2003)*, pages 153–164, 292, 2003.
18. L. Orf, R. Wilhelmson, and L. Wicker. Visualization of a simulated long-track ef5 tornado embedded within a supercell thunderstorm. *Parallel Computing*, 55:28–34, 2016.
19. V. Skala. Fast interpolation and approximation of scattered multidimensional and dynamic data using radial basis functions. *WSEAS Transaction on Mathematics*, 12(5):501–511, 2013.
20. V. Skala and M. Smolik. A new approach to vector field interpolation, classification and robust critical points detection using radial basis functions. In *Computer Science On-line Conference*, pages 109–115. Springer, 2018.
21. M. Smolik and V. Skala. Classification of critical points using a second order derivative. *Procedia Computer Science*, 108:2373–2377, 2017.
22. M. Smolik and V. Skala. Large scattered data interpolation with radial basis functions and space subdivision. *Integrated Computer-Aided Engineering*, 25(1):49–26, 2018.
23. M. Smolik, V. Skala, and Z. Majdisova. 3D vector field approximation and critical points reduction using radial basis functions. In *International Conference on Applied Physics, System Science and Computers*. Springer, 2018.
24. M. Smolik, V. Skala, and Z. Majdisova. Vector field radial basis function approximation. *Advances in Engineering Software*, 123(1):117–129, 2018.

Computational Science and Its Application, ICSSA 2019 proceedings, Part I, LNCS 11619, pp.337-350, ISSN 0302-9743
ISBN 978-3-030-24288-6, Springer, 2019 https://doi.org/10.1007/978-3-030-24289-3_25

25. J. Süßmuth, Q. Meyer, and G. Greiner. Surface reconstruction based on hierarchical floating radial basis functions. *Computer Graphics Forum*, 29(6):1854–1864, 2010.
26. X. Tricoche, G. Scheuermann, and H. Hagen. A topology simplification method for 2D vector fields. In *Visualization 2000. Proceedings*, pages 359–366. IEEE, 2000.
27. T. Weinkauf, H. Theisel, K. Shi, H.-C. Hege, and H.-P. Seidel. Extracting higher order critical points and topological simplification of 3D vector fields. In *Visualization, 2005. VIS 05. IEEE*, pages 559–566. IEEE, 2005.
28. J. Yang, Z. Wang, C. Zhu, and Q. Peng. Implicit surface reconstruction with radial basis functions. In *International Conference on Computer Vision and Computer Graphics*, pages 5–12. Springer, 2007.
29. R. Yokota, L. A. Barba, and M. G. Knepley. PetRBF-a parallel O(N) algorithm for radial basis function interpolation with gaussians. *Computer Methods in Applied Mechanics and Engineering*, 199(25):1793–1804, 2010.