

**University of West Bohemia
Faculty of Applied Sciences**

Vector field approximation methods

Ing. Michal Šmolík

**Doctoral thesis
submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in Computer Science and Engineering**

**Supervisor: Prof. Ing. Václav Skala, CSc.
Department: Department of Computer Science and Engineering**

Pilsen 2019

Západočeská univerzita v Plzni
Fakulta aplikovaných věd

Metody aproximace vektorových dat

Ing. Michal Šmolík

Disertační práce
k získání akademického titulu doktor
v oboru Informatika a výpočetní technika

Školitel: Prof. Ing. Václav Skala, CSc.
Katedra: Katedra informatiky a výpočetní techniky

Plzeň 2019

Prohlášení

Předkládám tímto k posouzení a obhajobě disertační práci zpracovanou na závěr doktorského studia na Fakultě aplikovaných věd Západočeské univerzity v Plzni. Prohlašuji, že tuto práci jsem zpracoval samostatně s použitím odborné literatury a dostupných pramenů uvedených v seznamu, jenž je součástí této práce.

V Plzni dne 18. října 2019

Ing. Michal Šmolík

Contents

1	Introduction	9
1.1	Objectives of the thesis	10
2	Theory	11
2.1	Vector field	11
2.1.1	Linearization of vector field	12
2.1.2	Critical points	13
2.1.2.1	Classification of critical points	14
2.1.2.2	Location of critical points	15
2.2	Radial Basis Functions	18
2.2.1	Radial Basis Function Approximation	20
3	Related work	22
3.1	Vector field simplification	22
3.2	Vector field approximation & compression	23
4	Overview of contributions	26
4.1	Vector field Radial basis function approximation	26
4.2	3D vector field approximation and critical points reduction using Radial basis functions	40
4.3	Large scattered data interpolation with Radial basis functions and space subdivision	45
4.4	Efficient simple large scattered 3D vector fields Radial basis functions approximation using space subdivision	60
4.5	Vector Field Radial Basis Functions Approximation with Streamlines Curvature	75
4.6	Radial Basis Function and Multi-level 2D Vector Field Approximation	85
4.7	Classification of critical points using a second order derivative	105
4.8	Vector field second order derivative approximation and geometrical characteristics	111
4.9	Vector field interpolation with Radial basis functions	124
4.10	Vector field RBF interpolation on a sphere	132
4.11	Spherical RBF vector field interpolation – experimental study	136
4.12	A comparative study of LOWESS and RBF approximations for visualization	141
5	Conclusion	157
5.1	Future work	157
A	Activities	159
A.1	Publications in Journals	159
A.2	Publications on International Conferences	160
A.3	Participation in Scientific Projects	161

Bibliography

162

Abstract

Vector fields or flow fields are results of numerical simulations or real measurements. The data size is very large especially for 3D vector fields. This thesis focuses on vector fields approximation and compression using meshless techniques. The approximation of vector fields focuses mostly on maintaining the most important characteristic of the vector field, i.e. critical points and global character of the vector field. During the research, several techniques that use Radial basis functions and different approaches for vector field approximation and compression, were developed.

This thesis provides a survey for the collection of 12 most important commented research papers which were written by the author of this thesis with coauthors during the author's doctoral study. The author has published 6 journal papers, has submitted additional 2 journal papers into impacted journals, and has published 22 conference papers.

In the last chapter are summarized the results achieved in this thesis and are discussed the possible directions of the future research in the field of vector field approximation.

This dissertation thesis was supported by the following projects:

- GA17-05534S – Meshless methods for large scattered spatio-temporal vector data visualization
The Czech Science Foundation
- LH12181 – Development of Algorithms for Computer Graphics and CAD/CAM systems
Czech Ministry of Education, Youth and Sports (MEYS)
- LG13047 – EURO: Activities within Eurographics Association
Czech Ministry of Education, Youth and Sports (MEYS)
- SGS-2019-016 – Synthesis and Analysis of Geometric and Computing Models
University of West Bohemia (UWB)
- SGS-2016-013 – Advanced Graphics and Computational Systems
University of West Bohemia (UWB)
- SGS-2013-029 – Advanced Computing and Information Systems
University of West Bohemia (UWB)

Copyright © 2019 University of West Bohemia, Czech Republic

Abstrakt

Vektorové pole nebo pole proudění jsou výsledkem numerických simulací nebo reálných měření. Objem těchto dat je veliký, zvláště pro případ 3D vektorových dat. Tato disertační práce se zaměřuje na aproximaci a kompresi vektorových dat pomocí bez-meshových technik. Aproximace vektorových dat se z největší části zaměřuje na zachování nejvíce důležitých charakteristik vektorového pole, tj. kritické body a globální charakter vektorového pole. Během výzkumu bylo vyvinuto několik technik používajících Radiální bázové funkce a různé přístupy pro aproximaci a kompresi vektorových dat.

Tato práce zprostředkovává přehled o 12 nejvíce důležitých komentovaných publikacích, které byly napsány autorem a jeho spoluautory během doktorských studií autora. Autor publikoval 6 časopiseckých publikací, zaslal k recenzím do impaktovaných časopisů další 2 články a dále publikoval 22 konferenčních příspěvků.

V poslední kapitole jsou shrnuty výsledky dosažené v této práci a jsou diskutovány možné další směry budoucího výzkumu v oblasti aproximace vektorových dat.

Tato disertační práce byla podporována následujícími projekty:

- GA17-05534S – Meshless metody pro vizualizaci velkých časově-prostorových vektorových dat
Grantová agentura České republiky
- LH12181 – NECPA - Vývoj algoritmů počítačové grafiky a pro CAD/CAM systémy
Ministerstvo školství, mládeže a tělovýchovy (MŠMT)
- LG13047 – EURO - Aktivity v rámci Eurographics Association
Ministerstvo školství, mládeže a tělovýchovy (MŠMT)
- SGS-2019-016 – Syntéza a analýza geometrických a výpočetních modelů
Západočeská univerzita v Plzni (ZČU)
- SGS-2016-013 – Pokročilé grafické a výpočetní systémy
Západočeská univerzita v Plzni (ZČU)
- SGS-2013-029 – Pokročilé výpočetní a informační systémy
Západočeská univerzita v Plzni (ZČU)

Copyright © 2019 Západočeská univerzita v Plzni, Česká republika

Acknowledgement

Hereby I would like to thank my thesis advisor Václav Skala for a great support from the beginning to the end. He has introduced me to this interesting topic and supported me with valuable advice and ideas that inspired my research work.

Besides my advisor, I would like to thank my colleagues at the Department of computer science and engineering for their discussions and suggestions which improved the quality of my research. Moreover, I would like to thank my family and close friends for great support during my studies.

1. Introduction

The concept of flow plays an important role in many fields of science. Classical application fields are, for example, the automotive and aerospace industry, where the investigation of the air flow around vehicles is an important task. However, the same concepts are used in the simulation and analysis of water flow in turbines of power plants, of blood flow in vessels, the propagation of smoke in buildings, and weather simulations, to mention just a few. The visualization of data gained from the simulation/measurement of such processes is relevant for the domain users as visualization has the potential to ease the understanding of such complex flow phenomena. In this context, topological flow visualization methods have been developed, with the aim to give insight into the overall behaviour of the flow. A characteristic of this class of methods is the segmentation of the flow domain into regions of substantially different flow behaviour, providing a topology of the flow domain.

The theory of dynamical systems goes back to the 19th century work of Henri Poincaré. An introduction to dynamical systems can be found for example in Katok and Hasselblatt [Katok and Hasselblatt, 1995]. In the context of this thesis, the case of deterministic, continuous and autonomous dynamical systems is most interesting, because such systems can be used to formulate velocity fields of a steady fluid flow. Many patterns in a flow can be described and analysed by concepts from dynamical systems theory, such as critical points, separatrices and periodic orbits. Perry and Chong [Perry and Chong, 1987] give a comprehensive overview of such *2D* and *3D* flow patterns. Helman and Hesselink introduced these methods to the scientific visualization community, and used them under the notion of vector field topology for the visualization of computed and measured velocity fields, first in *2D* [Helman and Hesselink, 1989] and later in *3D* [Helman and Hesselink, 1991]. Vector field topology was further popularized both by Asimov's excellent tutorial [Asimov, 1993] and by Globus et al.'s TOPO module [Globus et al., 1991] for NASA's FAST visualization software.

Mesh methods are standard tools for the simulation of flow problems, they enable efficient and reliable approximations of the differential equations in fluid flow. However, in certain applications, for example in the presence of large geometric deformations of the boundary or rotating and moving obstacles, i.e. situations which may frequently occur in the context of fluid with structure interaction problems, the maintenance of a conforming mesh may be almost impossible. Different techniques have been developed to deal with these problems in the mesh based context; the fictitious domain and fictitious boundary methods, techniques employing overlapping grids, sliding mesh, level set methods, or standard arbitrary Lagrangian Eulerian formulations with frequent re-meshing.

A different way to handle complex flow problems is to employ a comparably new and innovative class of methods, which enables the approximation of partial differential equations based on a set of nodes, without the need for an additional mesh. In recent years meshless/meshfree

methods have gained considerable attention in engineering and applied mathematics. The variety of problems that are now being addressed by these techniques continues to expand and the quality of the obtained results demonstrates the effectiveness of many of the methods currently available. These meshfree methods are generally able to solve problems where meshes bring up difficulties. However, they are comparably time consuming, which limits their usefulness in the simulation of challenging real-life problems.

1.1 Objectives of the thesis

The main focus of this thesis is on meshless approximation of vector fields in $2D$ and $3D$. Vector fields are often very large and very complex data sets. Such data came from numerical simulations or experimental measurements and need to be stored for further use. However, the data size is very large and thus some compression techniques are needed.

The compression can be lossy, while the important features of the vector field remains unchanged and the less important parts can be slightly modified during the approximation of the vector field.

The Radial basis functions (RBF) are used for the vector field approximation, which is a meshless technique used for scattered data approximation. The RBF with different modifications and different approaches are used to approximate and compress vector fields in this thesis. Moreover, the RBF approximation results in an analytical description of the vector field, which can be used for further processing.

This thesis is composed of two main parts. The first one is the introduction part with the basic definitions and the state of the art. The second one is the composition of several individual journal or conference articles with short description of each one.

2. Theory

2.1 Vector field

The term vector field or flow denotes an abstract concept adopted in many application fields. Fluid dynamics, for instance, is concerned with the study of fluid flows, i.e. the motion of fluids: typical examples include the motion of water in a pump or a turbine, the stream of air around a car or an airplane, blood in a vessel, oil or gas in a pipe, and many other. Flow visualization usually deals with data generated via measurements, simulations or modeling, and the results are commonly expressed as vector fields.

A vector field is a function that assigns to each point a vector. Vector fields come mostly from numerical simulations, i.e. Computational Fluid Dynamics (CFD) [Kansa, 1990a], [Kansa, 1990b], [Blazek, 2015] and Finite Element Method (FEM) [Molina-Aiz et al., 2010], [Chung et al., 2015].

Vector fields can be described for general vectors in E^n , i.e. Euclidean space in n dimensions, but for our purposes it is sufficient to consider E^2 or E^3 . A vector field in E^3 is a map that for each point $\mathbf{x} = [x, y, z]^T$ in a domain assigns a vector

$$\mathbf{v}(\mathbf{x}(t), t) = [v_x(\mathbf{x}(t), t), v_y(\mathbf{x}(t), t), v_z(\mathbf{x}(t), t)]^T, \quad (2.1)$$

where each of the function v_x or v_y or v_z is a scalar field and is dependent on the position and time, i.e. \mathbf{v} is time dependent vector field, or does not depend on time, i.e. \mathbf{v} is time independent vector field.

A field line is a line that is everywhere tangent to a given vector field at a particular time, see Figure 2.1. That is, every point on the line has a tangent that coincides with the vector at that location in the vector field. It can be constructed by tracing a path in the direction of the vector field while keeping time fixed. Field lines defines the direction of the vector field.

Let $\mathbf{x}(t)$ be a field line with parameter t representing the time. The field line is then given by a system of ordinary differential equations, which can be written as

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{v}(\mathbf{x}(t)) \quad \mathbf{x}(0) = \mathbf{x}_{start}, \quad (2.2)$$

where \mathbf{x}_{start} is an initial point. When numerical integration is used to solve the system of equations for visualization purposes, the point \mathbf{x}_0 is often referred to as a seed point. In terms of differential equations, the visualization of the field lines of a vector field correspond to phase portraits of solutions of the system (2.2). This will be an important concept in the discussion of classification of critical points later.

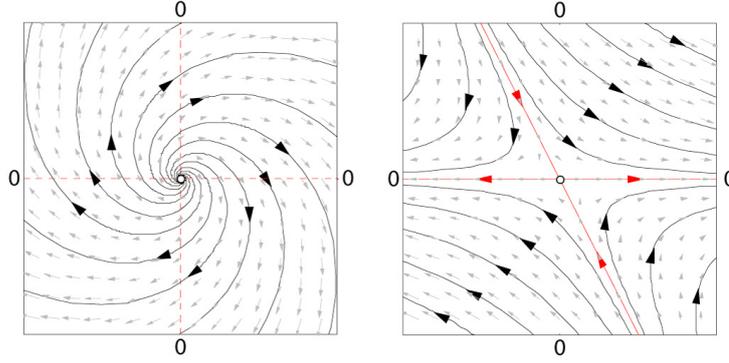


Figure 2.1: Example of linear vector fields with oriented field lines and vector field arrows located in a regular grid.

2.1.1 Linearization of vector field

In this section we discuss how to locally approximate a system by its linearization. These approximations are used at studying the local behavior of a system, where the nonlinear effects are expected to be small. The Taylor series expansion must be utilized locally to find the relation between \mathbf{v} and position \mathbf{x} , supposing the flow \mathbf{v} to be sufficiently smooth and differentiable. In such case, the expansion of \mathbf{v} around some point \mathbf{x}_0 is

$$\mathbf{v}(\mathbf{x}) = \mathbf{v}(\mathbf{x}_0) + \frac{\partial \mathbf{v}}{\partial \mathbf{x}} (\mathbf{x} - \mathbf{x}_0). \quad (2.3)$$

This equation can be rewritten using the matrix notation as

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \begin{bmatrix} \frac{\partial v_x}{\partial x} & \frac{\partial v_x}{\partial y} \\ \frac{\partial v_y}{\partial x} & \frac{\partial v_y}{\partial y} \end{bmatrix} \begin{bmatrix} x - x_0 \\ y - y_0 \end{bmatrix} \quad (2.4)$$

for the case of $2D$ vector field. In the case of $3D$ vector field, the equation (2.3) can be rewritten using the matrix notation as

$$\begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} + \begin{bmatrix} \frac{\partial v_x}{\partial x} & \frac{\partial v_x}{\partial y} & \frac{\partial v_x}{\partial z} \\ \frac{\partial v_y}{\partial x} & \frac{\partial v_y}{\partial y} & \frac{\partial v_y}{\partial z} \\ \frac{\partial v_z}{\partial x} & \frac{\partial v_z}{\partial y} & \frac{\partial v_z}{\partial z} \end{bmatrix} \begin{bmatrix} x - x_0 \\ y - y_0 \\ z - z_0 \end{bmatrix}. \quad (2.5)$$

If we consider an inverted pendulum whose open loop dynamics are given by the following equation

$$\frac{d\mathbf{x}}{dt} = \begin{bmatrix} y \\ \sin x - \gamma y \end{bmatrix}, \quad (2.6)$$

where a coefficient of viscous friction is γ , x is the angle and y is the angular rate. One of the

critical points is the point $\mathbf{x}_0 = [\pi, 0]^T$ (see section 2.1.2 for definition of critical point). The linearization of the dynamical system at this point is according to (2.3) equal to

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & -\gamma \end{bmatrix} \begin{bmatrix} x - \pi \\ y \end{bmatrix}. \quad (2.7)$$

We call the system (2.9) the linear approximation of the original nonlinear system or the linearization at the point \mathbf{x}_0 . The phase portrait of this linear approximation and original nonlinear phase portrait can be seen in Figure 2.2 (value of viscous friction was selected as $\gamma = 1$).

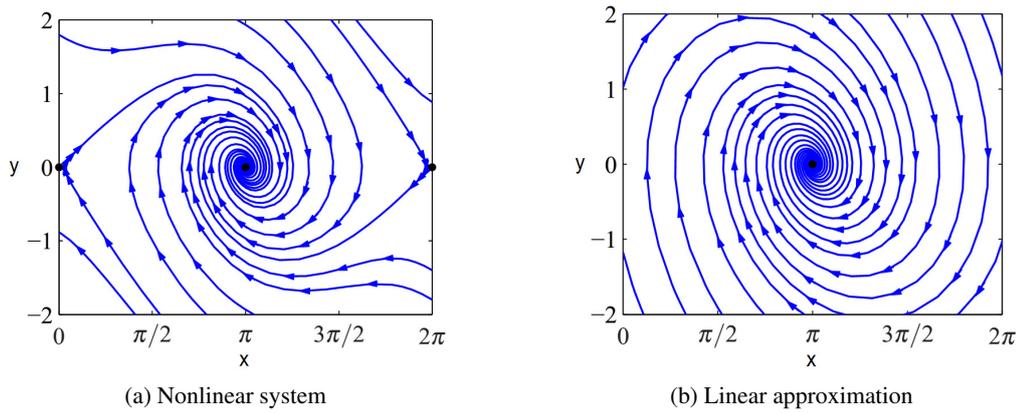


Figure 2.2: Comparison between the phase portraits for the full nonlinear system (a) and its linear approximation around the critical point at $\mathbf{x}_0 = [\pi, 0]^T$ (b). Notice that near the critical point at the center of the plots, the phase portraits are almost identical, i.e. the dynamics in both phase portraits are almost identical.

2.1.2 Critical points

Critical point, singular point, null point, neutral point or equilibrium point (\mathbf{x}_0) of the vector field is a point at which the magnitude of the vector vanishes

$$\frac{d\mathbf{x}}{dt} = \mathbf{v}(\mathbf{x}) = \mathbf{0}, \quad (2.8)$$

i.e. all components are equal to zero

$$\begin{bmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \\ \frac{dz}{dt} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \\ \frac{dz}{dt} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \quad (2.9)$$

A critical point is said to be isolated, or simple, if the vector field is non vanishing in an open neighborhood around the critical point. Thus for all surrounding points \mathbf{x}_ε of the critical point \mathbf{x}_0 the equation (2.8) does not apply, i.e.

$$\frac{d\mathbf{x}_\varepsilon}{dt} \neq \mathbf{0}. \quad (2.10)$$

Critical points can be of different order, where the order of the critical point is identical to the topological degree of the critical point, see Figure 2.3. For a closed curve $\tau(x_0)$ surrounding a single critical point x_0 in a continuous 2D vector field, the topological degree of x_0 is defined as the following curve integral

$$I(x_0) = \frac{1}{2\pi} \int_{\tau(x_0)} d\theta \quad (2.11)$$

and for 3D vector field is the topological degree defined as the following surface integral

$$I(x_0) = \frac{1}{4\pi} \int_{\tau(x_0)} d\theta, \quad (2.12)$$

where θ is the angle of vector field at location x . The value of $I(x_0)$ is always an integer, and the order of the critical point is the absolute value of this integer. A curve, resp. surface, $\tau(x_0)$ enclosing a region which does not contain a critical point has a topological degree of zero.

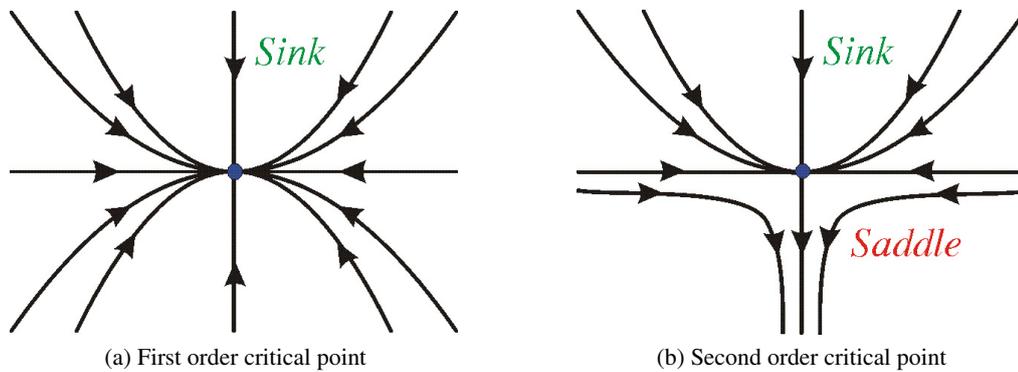


Figure 2.3: Example of critical points with different topological degree.

2.1.2.1 Classification of critical points

In the study of steady flow/autonomous dynamical systems certain features such as critical points, separatrices and closed orbits play an important role. In 1989, Helman and Hesslink introduced these concepts to the visualization community under the name of vector field topology [Helman and Hesslink, 1989].

There exist a finite set of fundamentally different critical points, defined by the number of inflow and outflow directions, spiraling structures etc., and combinations of these. Since the set is finite, each critical point can be classified. Such a classification defines the field completely in a close neighborhood around the critical point. By knowing the location and classification of critical points in a vector field, the topology of the field is known in small areas around these.

The fact that a linear model can be used to study the behavior of a nonlinear system near a critical point is a powerful one. We can perform the Taylor series expansion at the critical point x_0

$$v(x) = v(x_0) + \frac{\partial v}{\partial x} (x - x_0). \quad (2.13)$$

However, the first term of (2.13) is equal to zero vector, as magnitude of the vector field vanishes at the critical point. Thus linear approximation is of the form

$$\mathbf{v}(\mathbf{x}) = \frac{\partial \mathbf{v}}{\partial \mathbf{x}} (\mathbf{x} - \mathbf{x}_0) = \mathbf{J} (\mathbf{x} - \mathbf{x}_0), \quad (2.14)$$

where \mathbf{J} is the Jacobian matrix. The critical points are classified based on the vector field around them. This vector field is described using the Jacobian matrix. The eigenvalues and eigenvectors of Jacobian matrix are very important for vector field classification and description. For the eigenvalues and eigenvectors the following equation applies

$$\mathbf{J}\mathbf{u} = \lambda\mathbf{u}, \quad (2.15)$$

where λ is the eigenvalue and \mathbf{u} is the corresponding eigenvector. To compute the eigenvalues, we have to solve

$$(\mathbf{J} - \lambda\mathbf{I})\mathbf{u} = \mathbf{0}, \quad (2.16)$$

where \mathbf{I} is the identity matrix. Knowing Cramer's rule, a linear system of equations has nontrivial solution if the determinant vanishes, so the solutions of equation (2.16) are given by

$$\det(\mathbf{J} - \lambda\mathbf{I}) = 0. \quad (2.17)$$

A real eigenvector of the Jacobian matrix defines a direction such that if we move slightly from the critical point in that direction, the field is parallel to the direction we moved. Thus, at the critical point, the real eigenvectors are tangent to the trajectories that end on the point. The sign of the corresponding eigenvalue determines whether the trajectory is outgoing (repelling) or incoming (attracting) at the critical point. The imaginary part of an eigenvalue denotes circulation about the point.

Now suppose that we have a point \mathbf{x} on a line determined by an eigenvector \mathbf{u} (for an eigenvalue λ). That is, $\mathbf{x} = t\mathbf{u}$ for some scalar t . Then

$$\mathbf{v}(\mathbf{x}) = \mathbf{J}\mathbf{x} = \mathbf{J}(t\mathbf{u}) = t\mathbf{J}\mathbf{u} = t\lambda\mathbf{u}. \quad (2.18)$$

This direction vector $\mathbf{v}(\mathbf{x})$ is a multiple of \mathbf{u} and thus the eigenvectors determine the main tangent trajectories of the vector field around the critical point \mathbf{x}_0 , see Figure 2.4.

Classification of first order critical points can be done using eigenvalues of Jacobian matrix. We have 6 types of first order critical points in $2D$ (see Figure 2.5) and 8 types of first order critical points in $3D$ (see Figure 2.6).

2.1.2.2 Location of critical points

Location of critical points is important step in flow field visualization and feature extraction. There exist several approaches for extracting isolated zeros of scalar, vector and tensor fields [Ben-Israel, 1966], [Gjøystdal, 2004], [Greene, 1992], [Mann and Rockwood, 2002], [Press et al., 1989], [Wang et al.,], [Garth et al., 2004]. Combinatorial methods become particularly attractive, as they are not sensitive to numerical instabilities or the details of a particular implementation. The paper [Bhatia et al., 2014] introduced a robust method for detecting singularities in vector

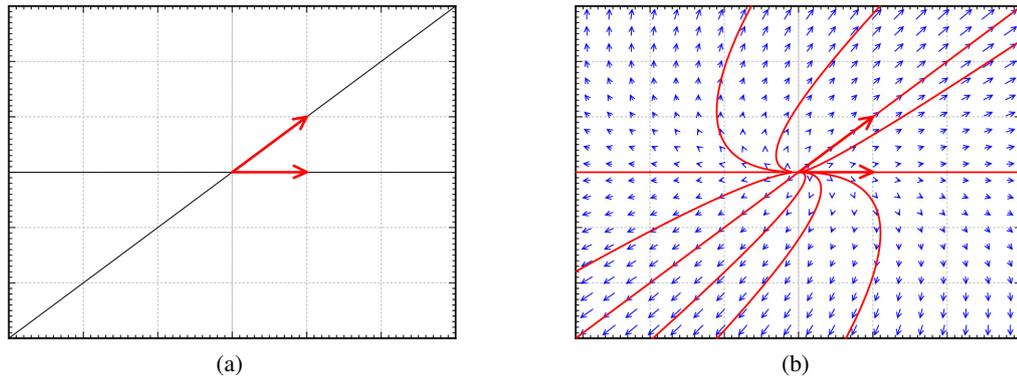


Figure 2.4: Example of a vector field. Eigenvectors of J (a). Example source vector field with eigenvectors and solutions (b).

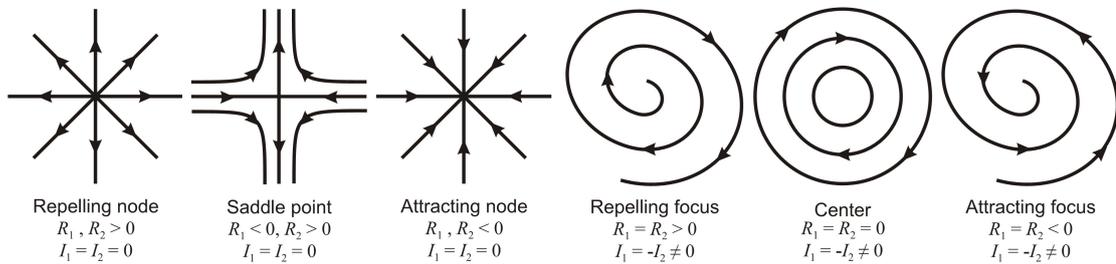


Figure 2.5: Classification of 2D first order critical points. R_1, R_2 denote the real parts of the eigenvalues of the Jacobian matrix while I_1, I_2 denote their imaginary parts.

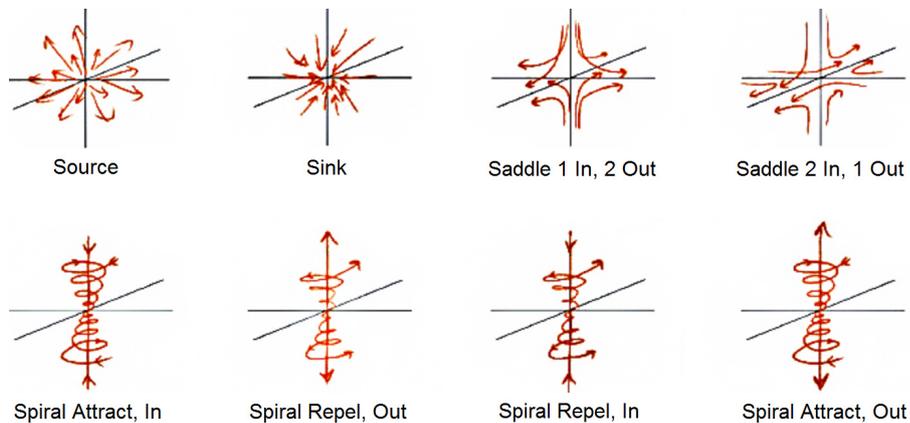


Figure 2.6: Classification of 3D first order critical points.

fields. They establish, in combinatorial terms, necessary and sufficient conditions for the existence of a critical point in a cell of a simplified mesh for a large class of interpolation functions. These conditions are entirely local and lead to a provably consistent and practical algorithm to identify cells containing singularities. Other selected algorithms for critical points location are presented in the following paragraphs.

Newton-Raphson method

We will discuss the simplest multidimensional root finding method, Newton-Raphson [Ben-Israel, 1966], [Press et al., 1989]. This method gives a very efficient means of converging to a root, if we have a sufficiently good initial guess. It can also spectacularly fail to converge, indicating (though not proving) that our putative root does not exist nearby.

Let \mathbf{p}_0 be a good estimate of critical point \mathbf{x}_0 and let $\mathbf{x}_0 = \mathbf{p}_0 + \mathbf{h}$. Since the true critical point is \mathbf{x}_0 , and $\mathbf{h} = \mathbf{x}_0 - \mathbf{p}_0$, the vector \mathbf{h} measures how far the estimate \mathbf{p}_0 is from the truth critical point.

In the neighborhood of \mathbf{p}_0 each of the functions $\mathbf{v} = [v_x, v_y]^T$ can be expanded in a Taylor series

$$\mathbf{v}(\mathbf{p}) = \mathbf{v}(\mathbf{p}_0) + \frac{\partial \mathbf{v}}{\partial \mathbf{x}}(\mathbf{p} - \mathbf{p}_0), \quad (2.19)$$

where

$$\mathbf{J} = \frac{\partial \mathbf{v}}{\partial \mathbf{x}}, \quad (2.20)$$

and \mathbf{J} is the Jacobian matrix. By setting $\mathbf{v}(\mathbf{p}) = \mathbf{0}$, we obtain a set of linear equations for the corrections $\mathbf{h} = (\mathbf{p} - \mathbf{p}_0)$ that move each function closer to zero simultaneously, namely

$$\mathbf{J}\mathbf{h} = -\mathbf{v}(\mathbf{p}_0). \quad (2.21)$$

This equation can be solved by the well known *LU* decomposition [Van Loan and Golub, 1983]. The corrections are then added to the solution vector

$$\mathbf{p}_{new} = \mathbf{p}_{old} + \mathbf{h} \quad (2.22)$$

and the process is iterated to convergence. In general it is a good to check the degree to which both functions and variables have converged.

Analytic method

The paper [Gjøystdal, 2004] describes an analytical method for location of critical points. The 3D vector field can be divided into small cells and interpolated using trilinear interpolation with the following definition

$$\begin{aligned} v_x(x, y, z) &= a_1 + b_1x + c_1y + d_1z + e_1xy + f_1xz + g_1yz + h_1xyz \\ v_y(x, y, z) &= a_2 + b_2x + c_2y + d_2z + e_2xy + f_2xz + g_2yz + h_2xyz \\ v_z(x, y, z) &= a_3 + b_3x + c_3y + d_3z + e_3xy + f_3xz + g_3yz + h_3xyz. \end{aligned} \quad (2.23)$$

When finding a critical point, the three equations must fulfill the following equations

$$\begin{aligned} v_x(x, y, z) &= 0 \\ v_y(x, y, z) &= 0 \\ v_z(x, y, z) &= 0. \end{aligned} \quad (2.24)$$

This system of equations can be solved using an analytic method. The strategy is as follows: First the interpolation coefficients $(a_1, a_2, a_3, b_1, \dots, h_3)$ must be computed. Variable z is then eliminated from the first equation in (2.23) and is inserted into the two other equations in (2.23) giving two new functions $\xi(x, y)$ and $\Phi(x, y)$. For a given x there must be a common value y that solves both equations

$$\begin{aligned}\xi(x, y) &= 0 \\ \Phi(x, y) &= 0.\end{aligned}\tag{2.25}$$

To determine this common factor, if any, one possibility is to use the determinant of the Sylvester matrix to find a polynomial which can be solved for x , see [Gardiner et al., 1992].

When the roots of the polynomial are determined using the numerical method, the next step is to determine which roots are the candidates for the x component of the critical points. The candidates are the values that are real and inside the closed unit interval. The candidate values of x are inserted into $\xi(x, y)$ and $\Phi(x, y)$ to see if there is at least one common root for y . For each pair (x, y) that solves both equations, we need to find the corresponding z value.

Not all values (x, y, z) that are computed correspond to true critical points in the cell. Every candidate point is therefore forwarded to a validation step to test if it is a true critical point or not.

Octree and topological degree method

According to [Greene, 1992], [Mann and Rockwood, 2002], [Max and Weinkauff, 2009] the space is subdivided into several cubes in a regular grid. If the index of a cube is nonzero, then a critical point was found, or perhaps a collection of critical points inside the cube. Then the cube has to be subdivided and thus create an octree structure. The subdivision step takes place until the cube has nonzero index and the size of cube is smaller than some defined smallest resolution. At this point a center of the cube is said to be the critical point.

Some of the critical points can be miss as the topological degree of critical points is additive. When two critical points with topological degree $+1$ and -1 are in the same cube, then the resulting topological degree of the cube is 0.

2.2 Radial Basis Functions

Radial basis function (RBF) is a technique for scattered data interpolation [Pan and Skala, 2011] and approximation [Fasshauer, 2007], [Skala, 2015]. The RBF interpolation and approximation is computationally more expensive compared to interpolation and approximation methods that use an information about mesh connectivity, because input data are not ordered and there is no known relation between them, i.e. tessellation is not made. Although RBF has a higher computational cost, it can be used for d -dimensional problem solution in many applications, e.g. solution of partial differential equations [Larsson and Fornberg, 2003], [Zhang et al., 2000], image reconstruction [Uhlir and Skala, 2005], neural networks [Karim and Adeli, 2003], [Ghosh-Dastidar et al., 2008], [Yingwei et al., 1998], GIS systems [Majdisova and Skala, 2017a], [Pan and Skala, 2012], optics [Prakash et al., 2012] etc. It should be noted that it does not require any triangulation or tessellation mesh in general. There is no need to know any connectivity

of interpolation points, all points are tied up only with distances of each other. Using all these distances we can form the interpolation or approximation matrix, which will be shown later.

The RBF is a function whose value depends only on the distance from its center point. Due to the use of distance functions, the RBFs can be easily implemented to reconstruct the surface using scattered data in 2D, 3D or higher dimensional spaces. It should be noted that the RBF interpolation and approximation is not separable by dimension.

Radial function interpolants have a helpful property of being invariant under all Euclidean transformations, i.e. translations, rotations and reflections. It does not matter whether we first compute the RBF interpolation function and then apply a Euclidean transformation, or if we first transform all the data and then compute the radial function interpolants. This is a result of the fact that Euclidean transformations are characterized by orthonormal transformation matrices and are therefore two-norm invariant.

Radial basis functions can be divided into two groups according to their influence. The first group are "global" RBFs [Schagen, 1979]. Application of global RBFs usually leads to ill-conditioned system, especially in the case of large data sets with a large span [Majdisova and Skala, 2017b], [Skala, 2017]. Typical examples of global RBFs are

$$\begin{array}{ll}
 \text{Thin Plate Spline (TPS)} & \varphi(r) = r^2 \log r \\
 \text{Gauss function} & \varphi(r) = e^{-(\varepsilon r)^2} \\
 \text{Inverse Quadric (IQ)} & \varphi(r) = \frac{1}{1 + (\varepsilon r)^2} \\
 \text{Inverse Multiquadric (IMQ)} & \varphi(r) = \frac{1}{\sqrt{1 + (\varepsilon r)^2}} \\
 \text{Multiquadric (MQ)} & \varphi(r) = \sqrt{1 + (\varepsilon r)^2}
 \end{array} \tag{2.26}$$

The "local" RBFs were introduced in [Wendland, 1995], [Wendland, 2006] as compactly supported RBF (CSRBF) and satisfy the following condition:

$$\begin{aligned}
 \varphi(r) &= (1 - r)_+^q P(r) \\
 &= \begin{cases} (1 - r)^q P(r) & 0 \leq r \leq 1 \\ 0 & r > 1 \end{cases}
 \end{aligned} \tag{2.27}$$

where $P(r)$ is a polynomial function, r is the distance of two points and q is a parameter. The subscript in $(1 - r)_+^q$ means:

$$(1 - r)_+ = \begin{cases} (1 - r) & (1 - r) \geq 0 \\ 0 & (1 - r) < 0 \end{cases} \tag{2.28}$$

Typical examples of local RBFs are

$$\begin{aligned}
 \varphi(r) &= (1-r)_+ \\
 \varphi(r) &= (1-r)_+^3(3r+1) \\
 \varphi(r) &= (1-r)_+^5(8r^2+5r+1) \\
 \varphi(r) &= (1-r)_+^2 \\
 \varphi(r) &= (1-r)_+^4(4r+1) \\
 \varphi(r) &= (1-r)_+^6(35r^2+18r+3) \\
 \varphi(r) &= (1-r)_+^8(32r^3+25r^2+8r+1)
 \end{aligned} \tag{2.29}$$

2.2.1 Radial Basis Function Approximation

RBF interpolation was originally introduced by [Hardy, 1971] and is based on computing the distance of two points in any k -dimensional space. The interpolated value, and approximated value as well, is determined as (see [Skala, 2013]):

$$h(\mathbf{x}) = \sum_{j=1}^M \lambda_j \varphi(\|\mathbf{x} - \boldsymbol{\xi}_j\|) \tag{2.30}$$

where λ_j are weights of the RBFs, M is the number of the radial basis functions, φ is the radial basis function and $\boldsymbol{\xi}_j$ are centers of radial basis functions. For a given dataset of points with associated values, i.e. in the case of scalar values $\{\mathbf{x}_i, h_i\}_1^N$, where $N \gg M$, the following over-determined linear system of equations is obtained:

$$h_i = h(\mathbf{x}_i) = \sum_{j=1}^M \lambda_j \varphi(\|\mathbf{x}_i - \boldsymbol{\xi}_j\|) \quad \text{for } \forall i \in \{1, \dots, N\} \tag{2.31}$$

where λ_j are weights to be computed; see Figure 2.7 for a visual interpretation of (2.30) or (2.31) for a $2\frac{1}{2}D$ function. Point in $2\frac{1}{2}D$ is a $2D$ point associated with a scalar value. The same also applies to $3D$ point associated with a scalar value, thus $3\frac{1}{2}D$ point.

Equation (2.31) can be rewritten in a matrix form as

$$\mathbf{A}\boldsymbol{\lambda} = \mathbf{h}, \tag{2.32}$$

where $A_{ij} = \varphi(\|\mathbf{x}_i - \boldsymbol{\xi}_j\|)$ is the entry of the matrix in the i -th row and j -th column, the number of rows $N \gg M$, M is the number of unknown weights $\boldsymbol{\lambda} = [\lambda_1, \dots, \lambda_M]^T$, i.e. a number of reference points, and $\mathbf{h} = [h_1, \dots, h_N]^T$ is a vector of values in the given points. The presented system is over-determined, 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 (LSE) as

$$\mathbf{A}^T \mathbf{A} \boldsymbol{\lambda} = \mathbf{A}^T \mathbf{h}, \tag{2.33}$$

where the matrix $\mathbf{A}^T \mathbf{A}$ is symmetrical. Another way to solve the over-determined system of linear equations is to use a QR decomposition algorithm. The idea of the QR algorithm is to

transform matrix A into product of two matrices Q and R [Lawson and Hanson, 1995], [Heath, 2018]. Then the equation (2.32) can be expressed as

$$QR\lambda = h, \quad (2.34)$$

where Q is an orthonormal $N \times N$ matrix and R is upper triangular $N \times M$ matrix. Since for orthonormal matrices is $Q^{-1} = Q^T$, the equation can be modified as

$$R\lambda = Q^T h. \quad (2.35)$$

After this it is easy to inverse the upper triangular matrix R and get the final solution

$$\lambda = R^{-1} (Q^T h). \quad (2.36)$$

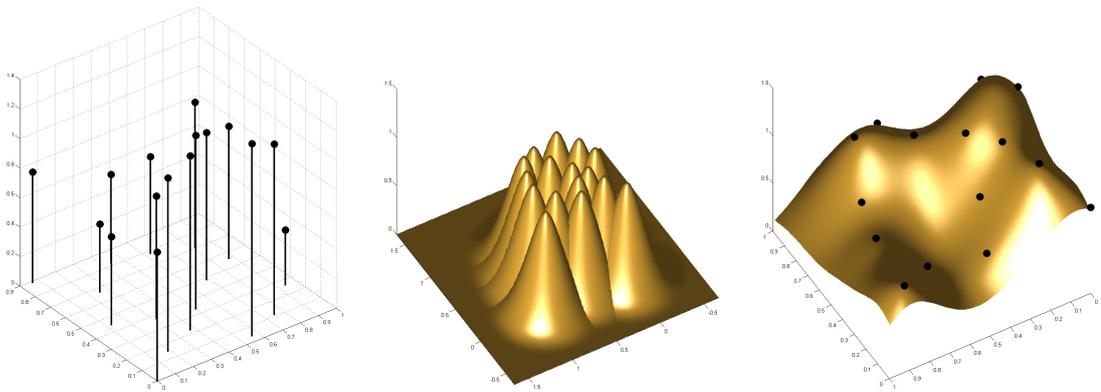


Figure 2.7: Data values, the RBF collocation functions, the resulting interpolant.

The RBF approximation can be done using "global" or "local" functions. When using "global" radial basis functions, the matrix A will be full, but when using "local" radial basis functions, the matrix A might be sparse, which can be beneficial when solving the over-determined system of linear equations $A\lambda = h$.

In the case of the vector data, i.e. $\{x_i, h_i\}_1^N$ values h_i are actually vectors, the RBF is to be performed for each coordinate of the vector h_i .

3. Related work

There exist many related publications in the area of vector fields approximation, simplification and compression. The selected most relevant papers are shortly described in the following chapters.

3.1 Vector field simplification

The paper [Tricoche et al., 2000] presents a new approach for turbulent vector fields simplification. Critical points are clustered based on the distance measurements and each cluster is replaced with only one higher order critical point. The vector field in each cluster is then replaced with a piece-wise linear vector field. When using different maximal distance between critical points in a cluster, one can obtain hierarchy of simplified vector fields. An extension of this work is presented in [Tricoche et al., 2002]. This approach enables to perform the simplification on any irregular grid, instead of previous work dealing only with structured grids.

The paper [Tricoche et al., 2001] removes pairs of critical points from the vector field. The removed pair of critical must not change the topological degree of the vector field, thus they remove critical points of opposite index (a saddle point with either a source or sink). Moreover, the removed critical points must be linked by a separatrix. The simplification does not change the grid of vector field data in any way, only small local changes of the vector field are performed.

An other approach for vector field simplification is presented in [Skraba et al., 2015], which is the extension of the approach described in [Skraba et al., 2014]. The simplification scheme is derived from the topological notion of robustness, i.e. a notion related to persistence [Edelsbrunner et al., 2000], [Reininghaus et al., 2011a] is used to represent the stability of critical points and evaluate their significance with respect to perturbations of the vector field. The proposed approach creates a hierarchical simplification scheme and then computes a piece-wise linear approximation of the vector field. This approach is also usable for unsteady, i.e. time-varying, vector fields. An extension for 3D vector field is presented in [Skraba et al., 2016]. It does not need to compute the entire 3D vector field topology, which can be hard to compute in some cases. Furthermore, the algorithm can remove critical points in any sub-region of the domain whose degree is zero and handle complex boundary configurations.

The construction and topological simplification of combinatorial vector fields on 2D manifolds is presented in [Reininghaus and Hotz, 2011]. Using Forman's combinatorial Morse theory for vector fields [Forman, 1998] they extract the topological skeleton including all periodic orbits and use it for simplification of vector field. The algorithm is able to process noisy data and creates multi-scale simplification of the vector field. The extension of this work is presented in [Reininghaus et al., 2011b]. This paper presents an approximation algorithm for combinatorial

vector field topology with a significantly lower complexity than [Reininghaus and Hotz, 2011]. Due to its simplicity it can be easily parallelized to improve the runtime even more.

The paper [Weinkauff et al., 2005] extracts and classifies higher order critical points of $3D$ vector fields. The paper presents the equivalence between classification of $3D$ critical points and the $2D$ vector field on the surface around that critical point. First order critical points can be grouped and then according the $2D$ vector field on the surface around them replaced with only one higher order critical point [Weinkauff et al., 2004]. This approach was developed especially for the visualization purpose as it leads to expressive visualizations of topologically complex $3D$ vector fields.

The Morse decomposition technique [Chen et al., 2008], [Szymczak and Zhang, 2012], [Szymczak, 2013] is used to simplify the vector field in [Sipeki and Szymczak, 2014]. This paper describes a method to simplify the Morse decomposition by iteratively merging pairs of Morse sets that are adjacent in the Morse connection graph. The result of this approach is a hierarchy of Morse decompositions based on flexible, user-specified criteria. Another approach that uses the Morse decomposition technique is [Gyulassy and Natarajan, 2005] and [Gyulassy et al., 2006]. Critical points paired by the Morse-Smale complex [Gyulassy et al., 2007] identify topological features and their importance. The simplification procedure leaves important critical points untouched, and is therefore useful for extracting desirable features.

Delaunay triangulation is used to simplify the vector field dataset in [Dey et al., 2007]. The authors reduce the mesh, where the vector field is defined, to simplify the vector field. Vertices are iteratively removed from the mesh while areas around critical points are untouched and for each removed vertex is computed local error metric of the vector field. The simplification algorithm can create a simplified vector field with some maximal error bound. Very similar approach for simplification of vector fields over tetrahedral meshes is presented in [Platis and Theoharis, 2004]. This approach uses half-edge colapses to reduce the tetrahedral mesh. Another approach using the Delaunay triangulation is [Cao et al., 2015]. The vertices are removed and also the positions of vertices are updated based on the Hessian matrix and thus create optimal triangulation for specific vector field.

3.2 Vector field approximation & compression

Nowadays the computational power of supercomputers increases and researchers are able to produce from numerical simulations more and more detailed vector fields. This vector fields are so large, that it is needed to approximate and compress the vector field for backup storage.

The paper [Koch et al., 2016] presents a vector field approximation algorithm. The vector field is divided into several segments, where the vector field inside each segment is approximated using an affine linear function [Koch et al., 2013]. The number of segments is as small as possible and the approximation must preserves the original vector field topology. The vector field approximation error is below some threshold value selected by user and each point has the minimal approximation error. To find the optimal solution of this problem is computationally demanding, thus the authors proposed a greedy solution of this problem.

Only some samples of the vector field are selected in [Agranovsky et al., 2015] for vector field compression. The selection of subsamples is done in three different ways. The first one is the random selection of subsamples, next the selection at regular locations corresponding to multi-resolution reduction and finally the last one is a greedy selection technique. This subsamples

are selected according to the properties of the flow, the algorithm targets areas with turbulence and rotation, where the subsampling is more detailed.

The Radial basis function approximation of vector field is used in [Cabrera et al., 2013]. This paper does research in the performance of RBF approximation of vector fields. There are compared the local and global RBFs, where the local RBFs are computationally less demanding than the global RBFs, which produce more smooth results. More precisely, they compare the local Hermite interpolation technique [Jumarhon et al., 2000] using inverse multiquadrics against the non-symmetric collocation method of Kansa [Kansa, 1990a], [Kansa, 1990b]. The results indicate that unlike global method, local technique and in particular the local Hermite RBF may solve problems of great scale since the system of equations has a sparse matrix with corresponding low condition number. On the other hand, they observed that for Dirichlet–Neumann boundary conditions the Hermite method produces results which are less accurate than those from global collocation schemes.

Topological symplification and topology preserving compression of vector fields are combined in [Theisel et al., 2003a]. The important features of the vector field should be preserved, i.e. critical points and separatrices, while unimportant features can be changed. In the first step of the algorithm is computed the complete topological skeleton of the vector field. Next are computed and assigned weights to critical points and separatrix. This weights are used to distinguish between important and unimportant features. In the final step, i.e. the approximation, are preserved only the selected important topological features.

The $2D$ vector field defined on a triangular mesh can be approximated using the algorithm in [Theisel et al., 2003b]. This approach preserves the complete topology, i.e. the critical points and the connectivity of the separatrices. The approximation is done iteratively through collapsing edges, while authors present the approach how to detect if this local change will change the global topology or not. This decision is made only by a local analysis whether or not the global topology is preserved. This approach focuses only on preserving the topology, however, it does not minimize the the vector field approximation error.

The compression of time varying vector field is presented in [Yuan et al., 2019]. This approach compress difference between two frames in time. This difference vector field is simplified using octree subdivision technique, where important features are preserved, i.e. including small turbulence areas.

The constrained clustering is used in [Lodha et al., 2000] in a vector field approximation algorithm. This algorithm is similar to [Telea and Van Wijk, 1999]. The vector field is sampled on a $2D$ grid with associated vector at each node. The cluster is a set of nodes, where each node has assigned one vector and thus also all the sub-nodes have assigned this vector. The decision if merge two nodes in the cluster is made based on a cost function. This cost function computes the error criteria based on global error metrics (to preserve the topology), local magnitude error, angular error and node importance error. The extension of this work is presented in [Lodha et al., 2003]. This work uses two different refinement techniques. The first technique uses binary tree subdivision and linear interpolation. The second algorithm is driven by triangular quadtree subdivision with Coons patch quadratic interpolation.

The multi-level approximation of vector field is presented in [de Leeuw and van Liere, 2000]. Only the important critical points are presented at each level of detail. For the reduction of critical points is used an algorithm for critical points cancellation [Čomić and De Floriani, 2008], [Skraba et al., 2016], [Wang et al., 2013]. After building a hierarchy of vector field with its critical points, the authors end-up with a compressed representation of a vector field. Another similar approach

for generating a multi-resolution representation of the vector field is presented in [Heckel et al., 1999].

4. Overview of contributions

I published many papers during the doctoral studies. Only some of them are directly related to the topic of this dissertation thesis. The selected important publications that are directly related to the topic of this dissertation thesis are summarized in the following chapters and at the same time the full text of the published research papers is attached.

4.1 Vector field Radial basis function approximation

The vector fields are often very complex and large as well. The paper [Smolik et al., 2018] presents an approach for vector field approximation and simplification. The important properties of the vector field are preserved in the final approximation. This paper propose an algorithm to determine if a critical point is important, i.e. the critical point has a global character and large influence. Critical points with only local influence can be omitted in the final approximation and thus removed from the vector field.

For the vector field approximation are used the Radial basis functions. To preserve the important critical points in the final approximation, the Lagrange multipliers are used for this purpose. The radial basis functions are placed at the important locations in the vector field, i.e. position of critical points and extremes of either v_x or v_y component of the vector field.

The final RBF approximation preserves all the important details and preserves the global character of the vector field, while keeping low vector field approximation error and high compression ratio.

Citation:

- Michal Smolik, Vaclav Skala, and Zuzana Majdisova. Vector field radial basis function approximation. *Advances in Engineering Software*, 123(1):117–129, 2018. (IF = 4.194)



Research paper

Vector field radial basis function approximation[☆]Michal Smolik^{*}, Vaclav Skala, Zuzana Majdisova

Faculty of Applied Sciences, University of West Bohemia, Plzen, Czech Republic



ARTICLE INFO

Keywords:

Vector field
Radial basis functions
Critical point
Lagrange multipliers
Simplification
Fourier transform
Cosine transform

MSC:

00-01
99-00

ABSTRACT

Vector field simplification aims to reduce the complexity of the flow by removing features according to their relevance and importance. Our goal is to preserve only the important critical points in the vector field and thus simplify the vector field for the visualization purposes. We use Radial Basis Functions (RBF) approximation with Lagrange multipliers for vector field approximation. The proposed method was experimentally verified on synthetic and real weather forecast data sets. The results proved the quality of the proposed approximation method compared to other existing approaches. A significant contribution of the proposed method is an analytical form of the vector field which can be used in further processing.

1. Introduction

Interpolation and approximation are probably the most frequent operations used in computational techniques [1]. Several techniques have been developed for data interpolation and approximation, but they require some kind of data “ordering”, e.g. structured mesh, rectangular mesh, unstructured mesh etc. A typical example is a solution of partial differential equations (PDE), where derivatives are replaced by differences and rectangular or hexagonal meshes are used in the vast majority of cases. However, in many engineering problems, data are not ordered and they are scattered in k -dimensional space, in general. The k -dimensional space is sometimes not only spatial but also contains a time dimension or a dimension relating to age or temperature or other environmental conditions. Usually, in technical applications the scattered data are tessellated using triangulation, but this approach is quite prohibitive for the case of k -dimensional data interpolation or approximation because of the computational cost [2].

The technique for visualizing topological information in fluid flows is well known [3]. However, when the technique is used in complex and information-rich data sets, the result will be a cluttered image which is difficult to interpret. The paper [4] presents a simplification approach that removes pairs of critical points from the dataset, based on relevance measures. The approach does no grid changes since the whole method uses small local changes of the vector values defining the vector field. A simplification of vector field can be achieved by merging

critical points within a prescribed radius into higher order critical points [5]. After building clusters containing the singularities to merge, the method generates a piecewise linear representation of the vector field in each cluster containing only one higher order singularity. Paper [6] presents a method to segment regions around a higher order critical point into areas of different 3D flow behavior. This method can be applied to any area of interest, e.g. around clusters of critical points. This can be used for a topological simplification tool by replacing the topological skeleton inside the area of interest. Combination of topological simplification technique and topology preserving compression for 2D vector fields is presented in [7]. A vector field is compressed in such way that its important topological features are preserved while its unimportant features are allowed to collapse and disappear. Dey et al. [8] present a Delaunay based algorithm for simplifying vector fields. The algorithm controls a local metric during removing vertices from Delaunay triangulation and maintains regions near critical points to prevent topological changes. The paper [9] uses a filtering technique based on the vorticity of the vector field to eliminate the less interesting critical points. The magnitude of the curl of the scalar field provides a basis to control the boundary thresholds as well as the number of critical points to include in the vector field. The paper [10] presents a technique for the visualization of multi-level topology in flow data sets. It provides the user with a mechanism to visualize the topology without excessive cluttering while maintaining the global structure of the flow. Skraba et al. [11,12] enable the pruning of sets of critical points

[☆] The research was supported by projects Czech Science Foundation (GACR) No. 17-05534S and partially by SGS 2016-013.

^{*} Corresponding author.

E-mail address: smolik@kiv.zcu.cz (M. Smolik).

<https://doi.org/10.1016/j.advengsoft.2018.06.013>

Received 25 April 2018; Received in revised form 13 June 2018; Accepted 24 June 2018

Available online 21 July 2018

0965-9978/ © 2018 Elsevier Ltd. All rights reserved.

according to a quantitative measure of their stability, that is, the minimum amount of vector field perturbation required to remove them. This leads to a hierarchical simplification scheme that encodes flow magnitude in its perturbation metric. A topological denoising technique based on a global energy optimization is proposed in [13], which allows the topology-controlled denoising of scalar fields. It allows processing small patches of the domain independently while still avoiding the introduction of new critical points. In the paper [14], they performed a numerical investigation of the differences between RBF global and local methods, in order to investigate the possible advantage of using local methods for the approximation of vector fields. The paper [15] presents a vector field approximation for two-dimensional vector fields that preserves their topology and significantly reduces the memory footprint. This approximation is based on a segmentation. The flow within each segmentation region is approximated by an affine linear function.

2. Vector field

Vector fields on surfaces are important objects, which appear frequently in scientific simulation in CFD (Computational Fluid Dynamics) [16,17] or modeling by FEM (Finite Element Method) [18,19]. To be visualized, such vector fields are usually linearly approximated for the sake of simplicity and performance considerations.

The vector field can be easily analyzed when having an approximation of the vector field near some location point. The important places to be analyzed are so called critical points [20]. Analyzing the vector field behavior near these points gives us the information about the characteristic of the vector field.

Critical points \mathbf{x}_0 of the vector field are points at which the magnitude of the vector vanishes

$$\frac{d\mathbf{x}}{dt} = \mathbf{v}(\mathbf{x}) = \mathbf{0}, \tag{1}$$

i.e. all components are equal to zero

$$\begin{bmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{2}$$

A critical point is said to be isolated, or simple, if the vector field is non vanishing in an open neighborhood around the critical point. Thus for all surrounding points \mathbf{x}_e of the critical point \mathbf{x}_0 the equation (1) does not apply, i.e.

$$\frac{d\mathbf{x}_e}{dt} \neq \mathbf{0}, \tag{3}$$

At critical points, the direction of the field line is indeterminate, and they are the only points in the vector field where field lines can intersect (asymptotically). The terms singular point, null point, neutral point or equilibrium point are also frequently used to describe critical points.

These points are important because together with the nearby surrounding vectors, they have more information encoded in them than any such group in the vector field, regarding the total behavior of the field. The critical points are classified based on the vector field around these points, see Fig. 1.

3. Radial basis functions

Radial basis function (RBF) is a technique for scattered data interpolation [21] and approximation [22,23]. The RBF interpolation and approximation is computationally more expensive compared to interpolation and approximation methods that use an information about mesh connectivity, because input data are not ordered and there is no known relation between them, i.e. tessellation is not made. Although RBF has a higher computational cost, it can be used for d -dimensional problem solution in many applications, e.g. solution of partial differential equations [24,25], image reconstruction [26], neural networks

[27–29], GIS systems [30,31], optics [32] etc. It should be noted that it does not require any triangulation or tessellation mesh in general. There is no need to know any connectivity of interpolation points, all points are tied up only with distances of each other. Using all these distances we can form the interpolation or approximation matrix, which will be shown later.

The RBF is a function whose value depends only on the distance from its center point. Due to the use of distance functions, the RBFs can be easily implemented to reconstruct the surface using scattered data in 2D, 3D or higher dimensional spaces. It should be noted that the RBF interpolation and approximation is not separable by dimension.

Radial function interpolants have a helpful property of being invariant under all Euclidean transformations, i.e. translations, rotations and reflections. It does not matter whether we first compute the RBF interpolation function and then apply a Euclidean transformation, or if we first transform all the data and then compute the radial function interpolants. This is a result of the fact that Euclidean transformations are characterized by orthonormal transformation matrices and are therefore two-norm invariant. Radial basis functions can be divided into two groups according to their influence. The first group are “global” RBFs [33]. Application of global RBFs usually leads to ill-conditioned system, especially in the case of large data sets with a large span [34,35].

The “local” RBFs were introduced in [36] as compactly supported RBF (CSRBF) and satisfy the following condition:

$$\begin{aligned} \varphi(r) &= (1-r)_+^q P(r) \\ &= \begin{cases} (1-r)^q P(r) & 0 \leq r \leq 1 \\ 0 & r > 1 \end{cases} \end{aligned} \tag{4}$$

where $P(r)$ is a polynomial function, r is the distance of two points and q is a parameter. The subscript in $(1-r)_+^q$ means:

$$(1-r)_+ = \begin{cases} (1-r) & (1-r) \geq 0 \\ 0 & (1-r) < 0 \end{cases} \tag{5}$$

3.1. Radial basis function approximation

RBF interpolation was originally introduced by Hardy [37] and is based on computing the distance of two points in any k -dimensional space. The interpolated value, and approximated value as well, is determined as (see [38]):

$$h(\mathbf{x}) = \sum_{j=1}^M \lambda_j \varphi(\|\mathbf{x} - \xi_j\|) \tag{6}$$

where λ_j are weights of the RBFs, M is the number of the radial basis functions, φ is the radial basis function and ξ_j are centers of radial basis functions. For a given dataset of points with associated values, i.e. in the case of scalar values $\{\mathbf{x}_i, h_i\}_1^N$, where $N \gg M$, the following over-determined linear system of equations is obtained:

$$\begin{aligned} h_i &= h(\mathbf{x}_i) = \sum_{j=1}^M \lambda_j \varphi(\|\mathbf{x}_i - \xi_j\|) \\ \text{for } \forall i \in \{1, \dots, N\} \end{aligned} \tag{7}$$

where λ_j are weights to be computed; see Fig. 2 for a visual interpretation of (6) or (7) for a $2\frac{1}{2}D$ function. Point in $2\frac{1}{2}D$ is a 2D point associated with a scalar value. The same also applies to 3D point associated with a scalar value, thus $3\frac{1}{2}D$ point.

Eq. (7) can be rewritten in a matrix form as

$$\mathbf{A}\boldsymbol{\lambda} = \mathbf{h}, \tag{8}$$

where $A_{ij} = \varphi(\|\mathbf{x}_i - \xi_j\|)$ is the entry of the matrix in the i th row and j th column, the number of rows $N \gg M$, M is the number of unknown weights $\boldsymbol{\lambda} = [\lambda_1, \dots, \lambda_M]^T$, i.e. a number of reference points, and $\mathbf{h} = [h_1, \dots, h_N]^T$ is a vector of values in the given points. The presented

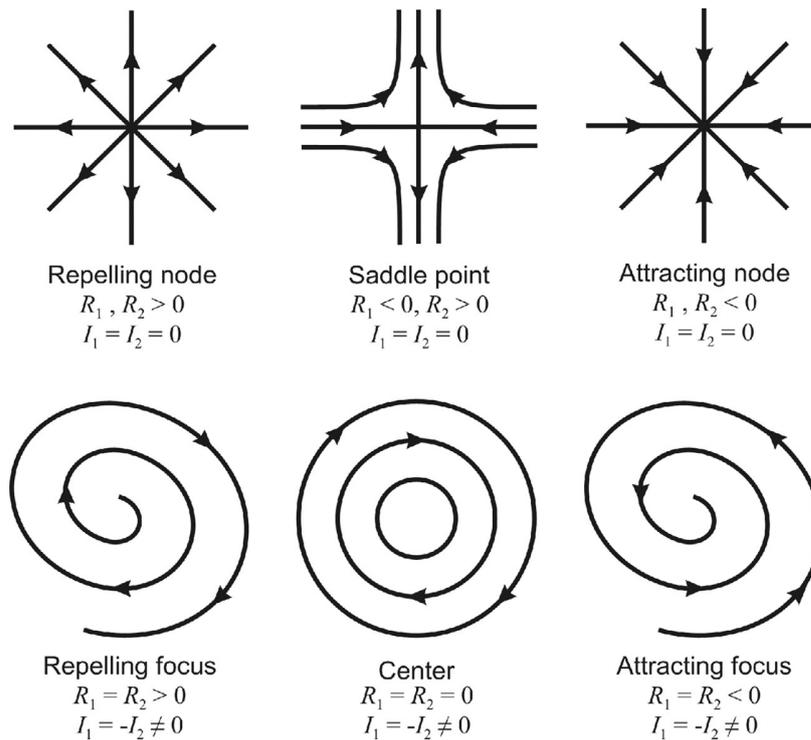


Fig. 1. Classification of 2D first order critical points. R_1, R_2 denote the real parts of the eigenvalues of the Jacobian matrix while I_1, I_2 denote their imaginary parts [20].

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 (LSE) as

$$A^T A \lambda = A^T h, \tag{9}$$

where the matrix $A^T A$ is symmetrical.

The RBF approximation can be done using “global” or “local” functions. When using “global” radial basis functions, the matrix A will be full, but when using “local” radial basis functions, the matrix A might be sparse, which can be beneficial when solving the overdetermined system of linear equations $A \lambda = h$.

In the case of the vector data, i.e. $\{x_i, h_i\}_1^N$ values h_i are actually vectors, the RBF is to be performed for each coordinate of the vector h_i .

3.2. RBF approximation with Lagrange multiplier

We want to minimize a multivariate function $f(x, y)$ subject to a constraint $g(x, y) = 0$. The method of Lagrange multipliers relies on the intuition that at a minimum, $f(x, y)$ cannot be decreasing in the direction of any neighboring point where $g(x, y) = 0$. Thus, the gradient of $f(x, y)$ is perpendicular to the constraint $g(x, y) = 0$ and thus the gradients of f and g are parallel, see Fig. 3, i.e.

$$\nabla f(x, y) = \eta \nabla g(x, y), \tag{10}$$

where η represents some constant.

The method of Lagrange multipliers is a powerful tool for solving this class of problems without the need to explicitly solve the conditions and use them to eliminate extra variables. Lagrange presented a special

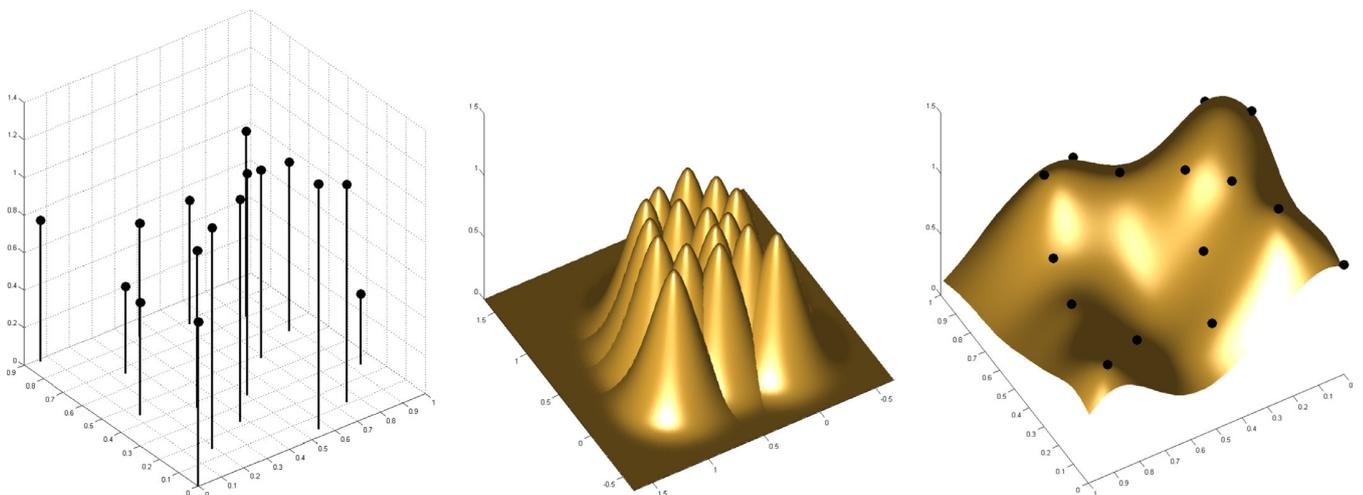


Fig. 2. Data values, the RBF collocation functions, the resulting interpolant.

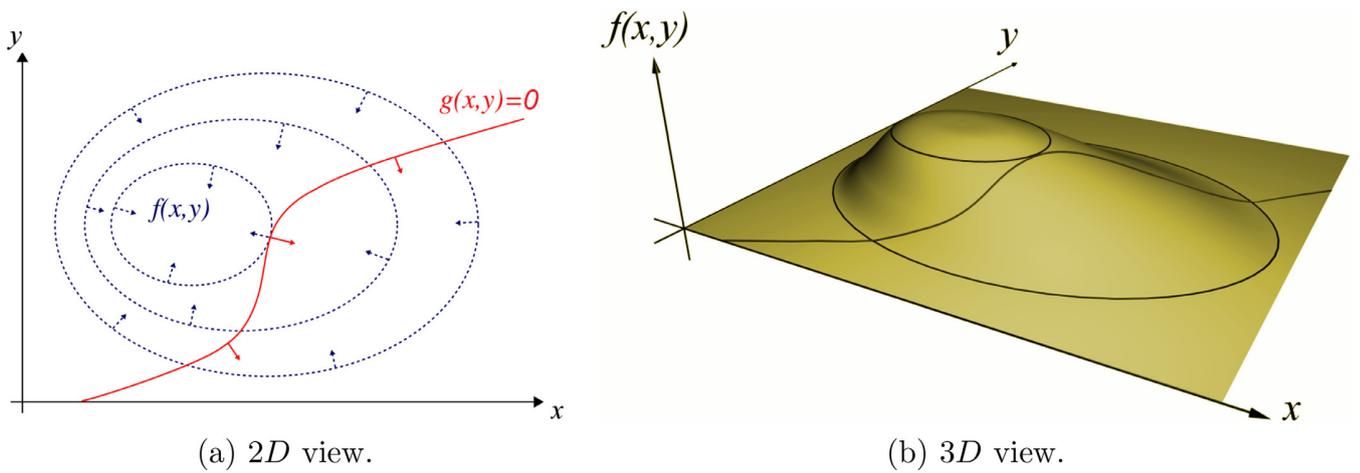


Fig. 3. The red line shows the constraint $g(x, y) = 0$. The blue dotted lines are contours of $f(x, y)$. The point where the red line tangentially touches a blue contour is the maximum of $f(x, y)$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

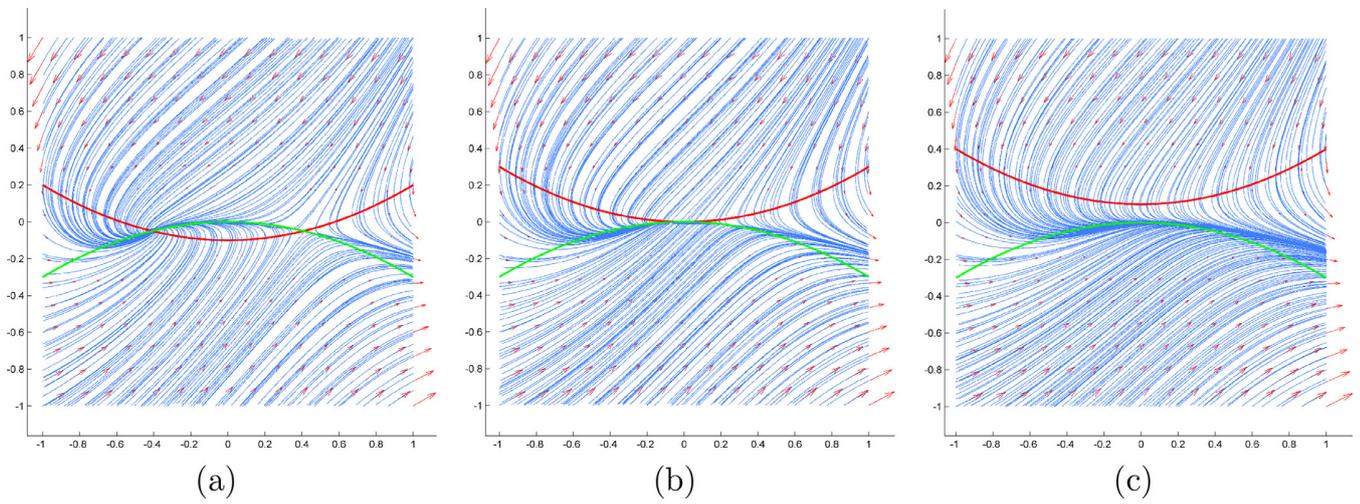


Fig. 4. Vector fields with almost parallel zero iso-lines. All the vector fields have the same global character.

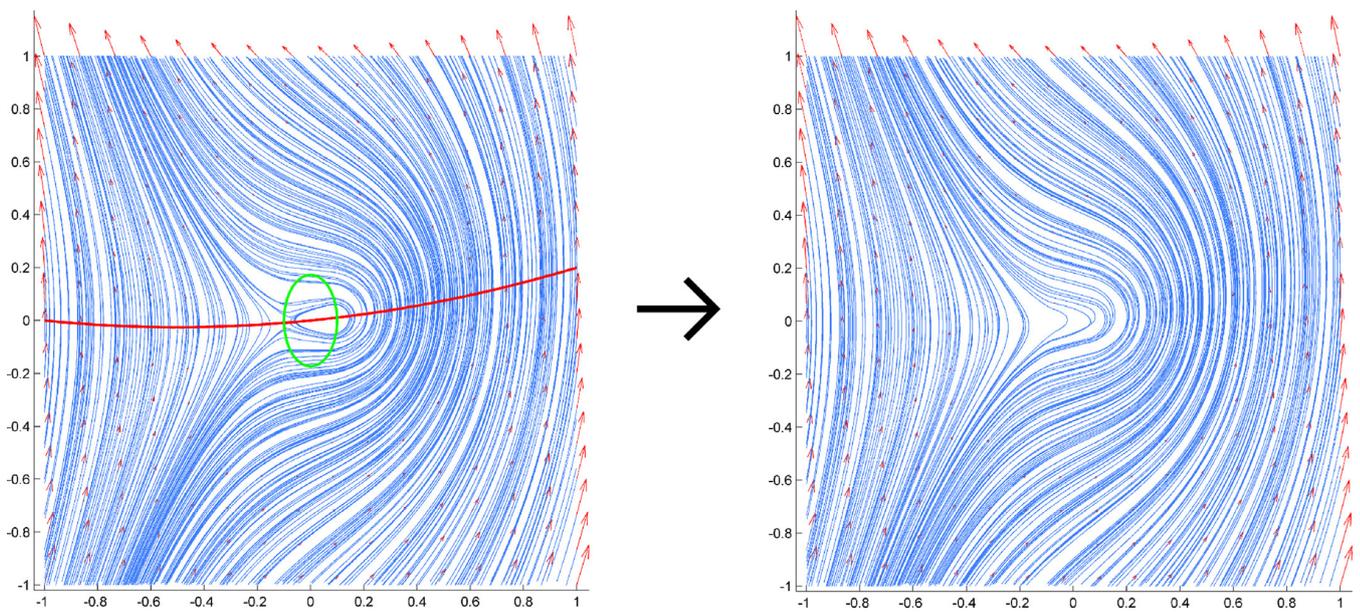


Fig. 5. Vector fields with one short zero iso-lines. Both the vector fields have the same global character.

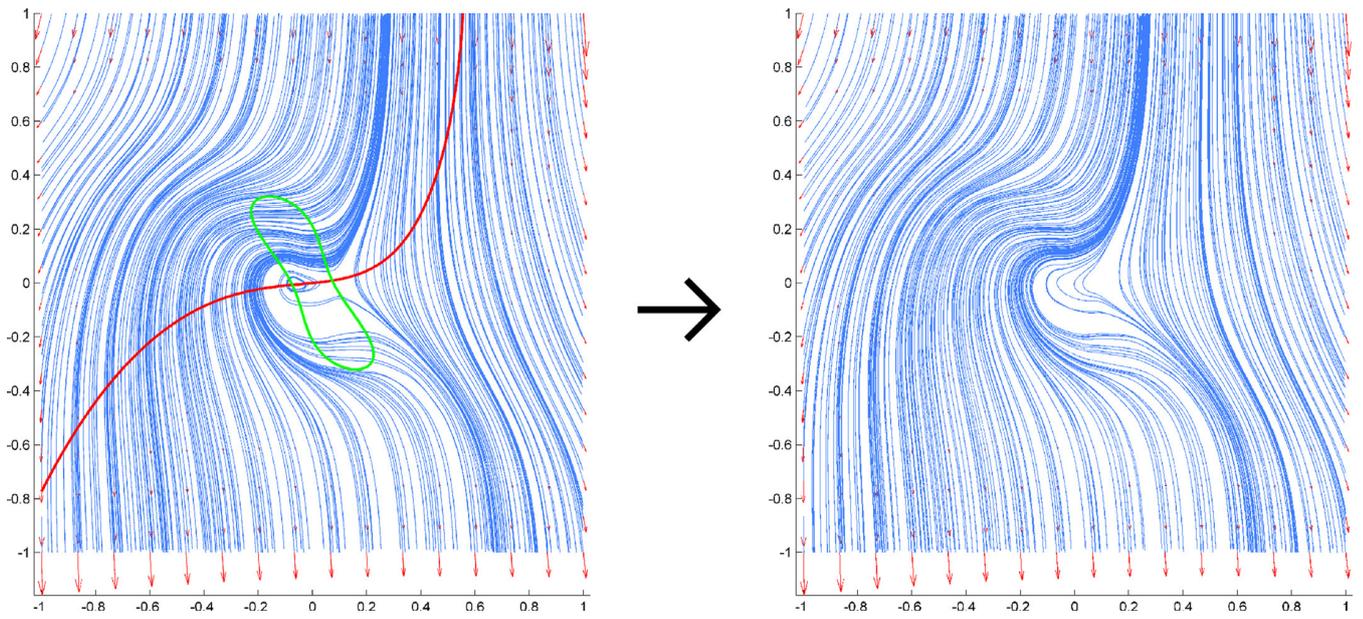


Fig. 6. Vector fields with one short zero iso-lines. Both the vector fields have the same global character.

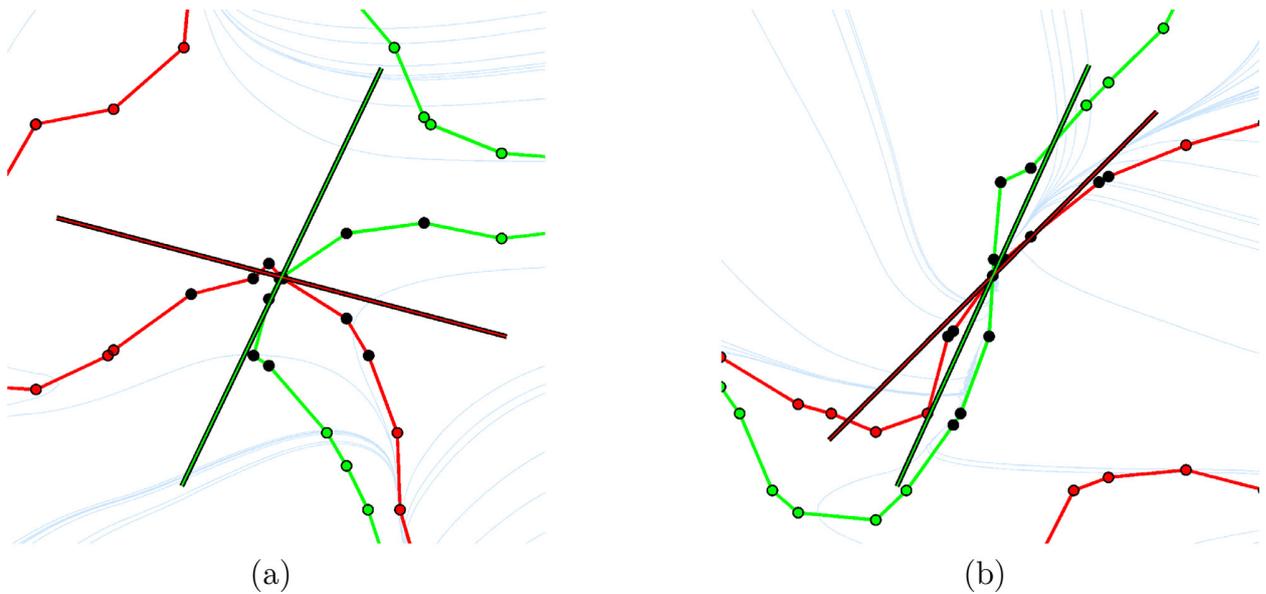


Fig. 7. Examples of zero iso-line approximation. The black line is the zero iso-line approximation at the critical point. The red and green curves are zero iso-lines. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

new function which takes in all the same input variables as f and g , along with η , thought of now as a variable called Lagrange multiplier. The Lagrange function is [39–41]

$$F(x, y, \eta) = f(x, y) + \eta g(x, y). \tag{11}$$

To find a minimum of (11) we need to set the gradient of (11) equal to zero

$$\begin{aligned} \frac{\partial F}{\partial \mathbf{x}} &= \mathbf{0} \\ \frac{\partial F}{\partial \eta} &= 0, \end{aligned} \tag{12}$$

where $\mathbf{x} = [x, y]^T$.

We will use Lagrange multipliers together with RBF approximation. The RBF approximation function is

$$h(\mathbf{x}) = \sum_{j=1}^M \lambda_j \varphi(\|\mathbf{x} - \xi_j\|), \tag{13}$$

where ξ_j are centers of radial basis functions, λ_j are weights of radial basis functions and φ is a radial basis function. We want to minimize

$$f(\lambda) = \sum_{i=1}^N \left(\sum_{j=1}^M \lambda_j \varphi(\|\mathbf{x}_i - \xi_j\|) - h_i \right)^2, \tag{14}$$

where N is the number of input points for approximation and h_i are associated function values at \mathbf{x}_i . The constrain for some points $\mathbf{x}^{(0)} = [\mathbf{x}^{(0)}_1, \dots, \mathbf{x}^{(0)}_{N_c}]$ that the RBF function is equal to zero is

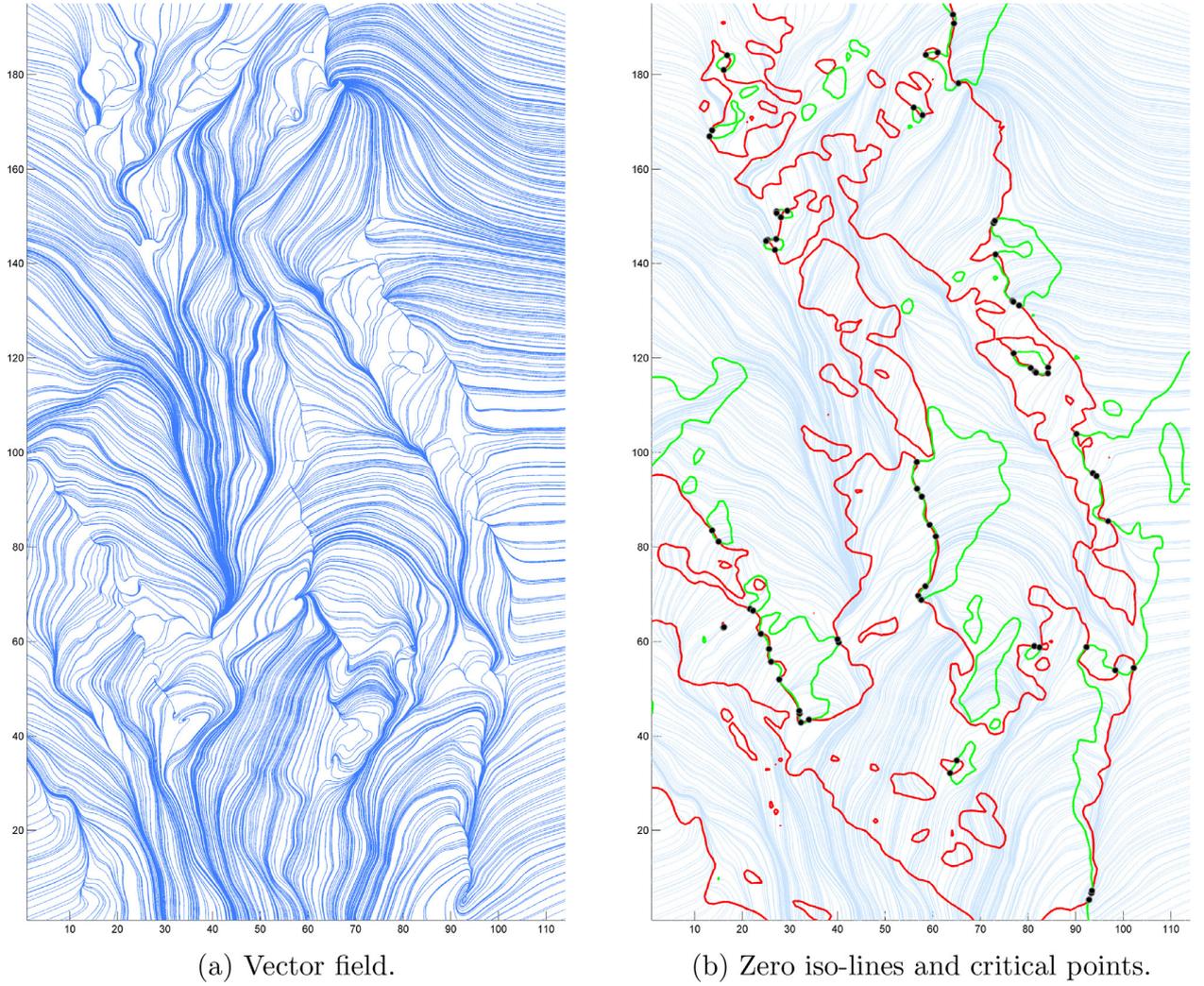


Fig. 8. Visualization of the input data set. The vector field (a) and the zero iso-lines with all critical points (b).

$$g(\mathbf{x}^{(0)_c}) = \sum_{j=1}^M \lambda_j \varphi(\|\mathbf{x}^{(0)_c} - \xi_j\|) = 0$$

for $\forall c \in \{1, \dots, N_c\}$,

(15)

where N_c is the number of constrains.

Using (14) as $f(x, y)$ in (11) and (15) as $g(x, y)$ in (11) we form the Lagrange function

$$F(\lambda, \eta) = \sum_{i=1}^N \left(\sum_{j=1}^M \lambda_j \varphi(\|\mathbf{x}_i - \xi_j\|) - h_i \right)^2 + \sum_{c=1}^{N_c} \eta_c \left(\sum_{j=1}^M \lambda_j \varphi(\|\mathbf{x}^{(0)_c} - \xi_j\|) \right).$$
(16)

This formula can be rewritten in a matrix form as

$$F(\lambda, \eta) = (\mathbf{A}\lambda - \mathbf{h})^T (\mathbf{A}\lambda - \mathbf{h}) + \lambda^T \mathbf{R}^T \eta$$

$$= \lambda^T \mathbf{A}^T \mathbf{A} \lambda - 2\mathbf{h}^T \mathbf{A} \lambda + \mathbf{h}^T \mathbf{h} + \lambda^T \mathbf{R}^T \eta,$$
(17)

where $A_{ij} = \varphi(\|\mathbf{x}_i - \xi_j\|)$ is the entry of the matrix \mathbf{A} in the i -th row and the j -th column, $R_{cj} = \varphi(\|\mathbf{x}^{(0)_c} - \xi_j\|)$ is the entry of the matrix \mathbf{R} , $\lambda = [\lambda_1, \dots, \lambda_M]^T$, $\mathbf{h} = [h_1, \dots, h_N]^T$ and $\eta = [\eta_1, \dots, \eta_{N_c}]^T$. To find the minimum of (17) we need to compute the partial derivatives of (17)

$$\frac{\partial F}{\partial \lambda} = 2\mathbf{A}^T \mathbf{A} \lambda + \mathbf{R}^T \eta - 2\mathbf{A}^T \mathbf{h}$$
(18)

and

$$\frac{\partial F}{\partial \eta} = \mathbf{R} \lambda.$$
(19)

To find the minimum, both partial derivatives (18), (19) must be equal zero

$$\frac{\partial F}{\partial \lambda} = \mathbf{0} \Rightarrow 2\mathbf{A}^T \mathbf{A} \lambda + \mathbf{R}^T \eta - 2\mathbf{A}^T \mathbf{h} = \mathbf{0},$$

$$\frac{\partial F}{\partial \eta} = \mathbf{0} \Rightarrow \mathbf{R} \lambda = \mathbf{0}.$$
(20)

Using (20), we can form the system of linear equations for RBF approximation with Lagrange multipliers

$$\begin{bmatrix} 2\mathbf{A}^T \mathbf{A} & \mathbf{R}^T \\ \mathbf{R} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \lambda \\ \eta \end{bmatrix} = \begin{bmatrix} 2\mathbf{A}^T \mathbf{h} \\ \mathbf{0} \end{bmatrix}$$
(21)

Solving those equations and finding λ , we made the RBF approximation and can compute the function values using (13).

4. Proposed approach

Importance measure for minima and maxima of 2D scalar fields called scale space persistence is introduced in [42]. This method is based on the mathematical concepts scale space, homological

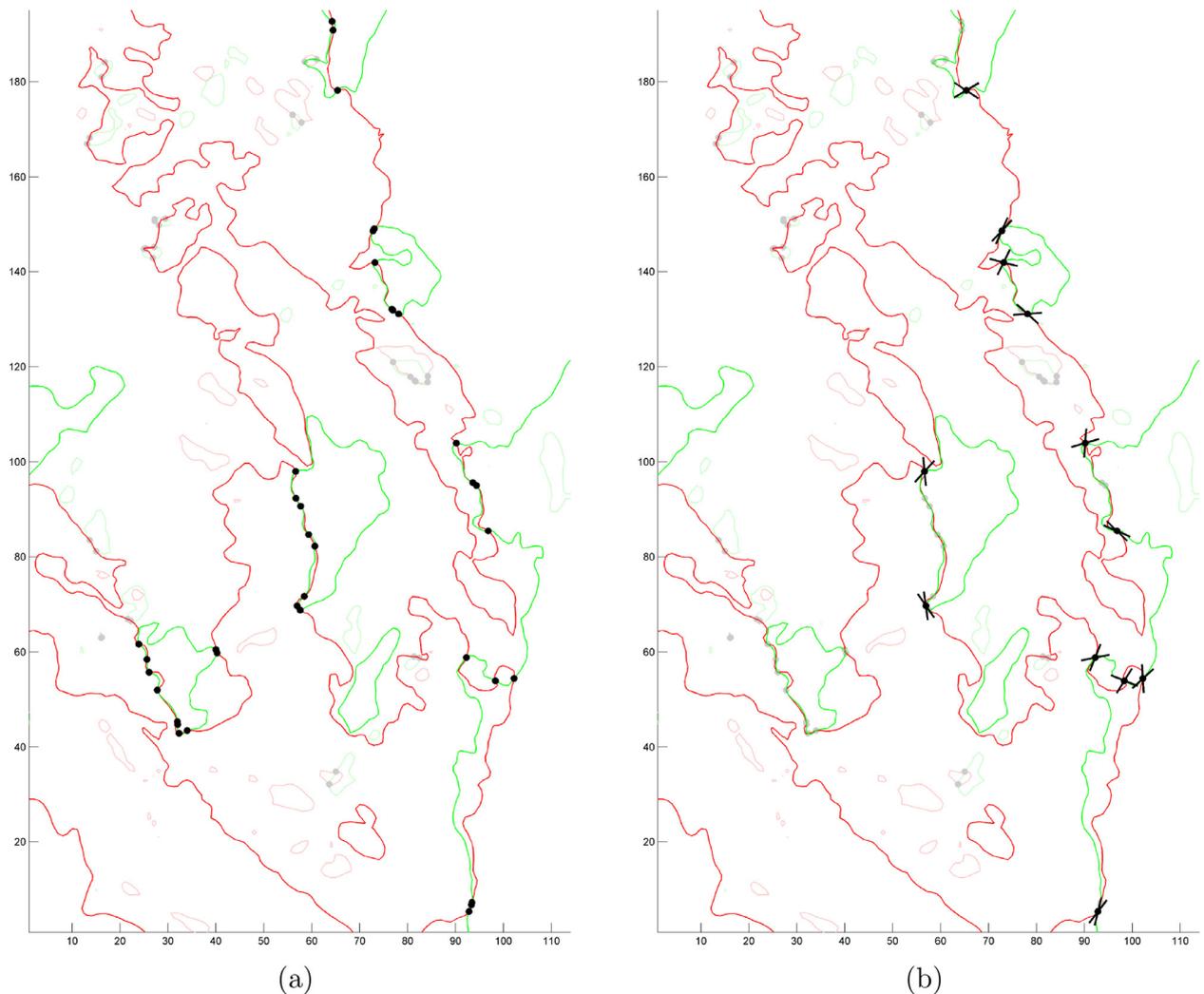


Fig. 9. First (a) and second (b) step of critical points reduction algorithm.

persistence, discrete Morse theory, and combinatorial feature flow fields. The combination of these powerful approaches to data analysis results in an importance measure that can deal with noisy data containing outliers. Robustness, a notion related to persistence, is used to represent the stability of critical points and evaluate their significance with respect to perturbations of the vector field. Intuitively, the robustness of a critical point is the minimum amount of perturbation that is required to cancel this critical point within a local neighborhood.

The proposed algorithm aims to reduce critical points in the vector field and thus simplify it. Critical points are located on places where the zero iso-lines of v_x and v_y intersect. The importance of the critical point is based on how the two zero iso-lines of v_x and v_y intersect and how long are these iso-lines.

If two iso-lines are almost parallel at a critical point, then this critical point does not have high importance as can be seen in the synthetic example in Fig. 4. The two zero iso-lines have two intersections (Fig. 4a), thus the vector field has two critical points at that locations. If we change the vector field in Fig. 4a so that the vector field has only one critical point and both the v_x and v_y zero iso-lines touches at one point, i.e. at critical point, see Fig. 4b. If we change the vector field Fig. 4a even more, we can end up with the vector field Fig. 4c, which has no critical point. All the vector fields Fig. 4a–c have the same global character and differ only in the number of critical points. Due to inaccuracy in measurement or numerical simulation or vector field data

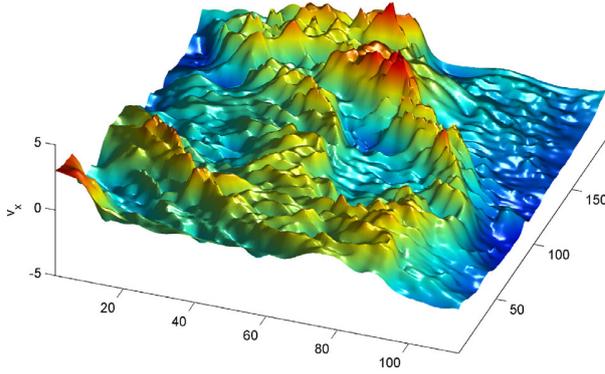
noise, there may be confusion between the individual cases in Fig. 4a–c. Some non existing critical points can occur in the data set and are disturbing for visualization purposes.

If two iso-lines have an intersection, where one iso-line is a relatively short closed curve, the influence of the critical point is only local and has almost no global influence for the vector field. Examples are visualized in Figs. 5 and 6. We can change the vector field Fig. 5 (left) that the v_y component of the vector field is always greater than zero, i.e. there is no zero iso-line, see Fig. 5 (right). However this modified vector field has no critical point, the global character of the vector field remains the same as in Fig. 5 (left). The same implies for vector field in Fig. 6 as well. Critical points created with a relatively short closed curve have the influence proportional to the size of the zero iso-line.

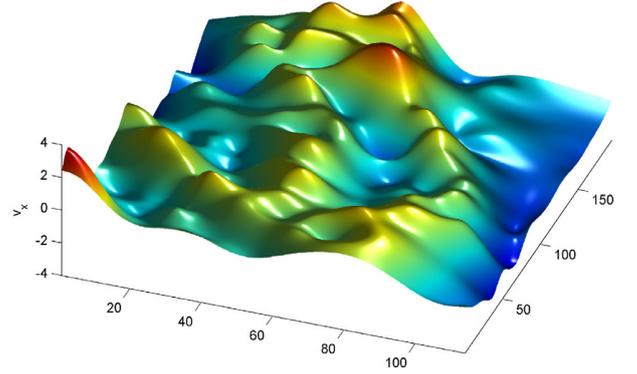
It can be seen that for the both cases mentioned above, i.e. almost parallel iso-lines in Fig. 4 and relatively short closed curve in Figs. 5 and 6, the vector fields have the same global character with the critical points as without them. In the following text we will show how to detect critical points affecting only local behavior of the vector field.

4.1. Critical points reduction

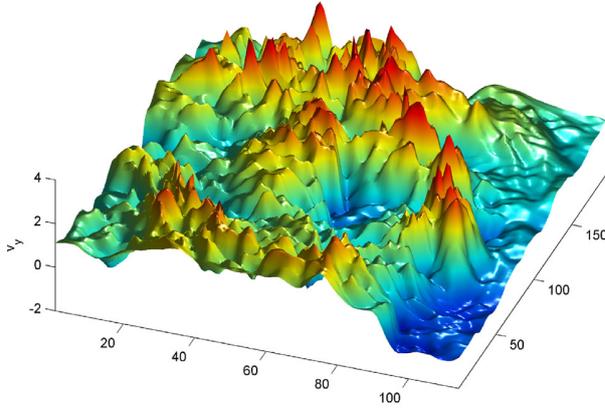
The first step of the proposed method is the critical points detection. Critical points in the vector field can be located using several methods [43–45]. Critical points of the 2D vector field are places, where the two



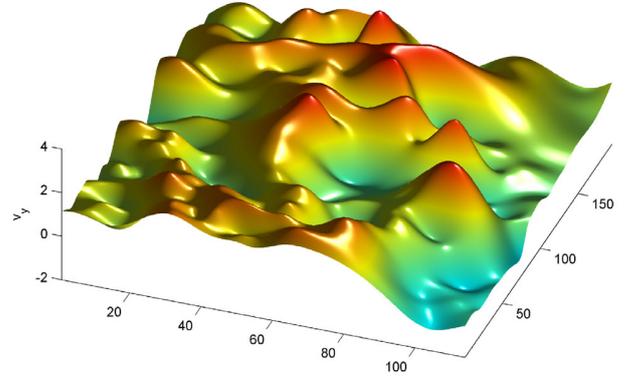
(a) Original v_x component of the vector field.



(b) Approximated v_x component of the vector field.



(c) Original v_y component of the vector field.



(d) Approximated v_y component of the vector field.

Fig. 10. Original and RBF approximated components of vector field.

zero iso-lines of v_x and v_y component intersect. Our goal is to trace both zero iso-lines in both directions from the critical point and compute the approximated zero iso-lines.

For the v_x zero iso-line approximation we need to compute intersections with the input data grid. From these intersections, we find some number of the closest ones to the critical point \mathbf{x}_0 . In the testing we used 6 closest intersection points $\{\mathbf{x}_c^{(1)}, \dots, \mathbf{x}_c^{(6)}\}$. This points need to be interlaced with an implicit line. For this purpose we use the method of total least square error computation in E^2 taken from [46]. This method computes implicit line

$$a^{(x)}x + b^{(x)}y + d^{(x)} = 0, \quad (22)$$

where $a^{(x)}$, $b^{(x)}$ and $d^{(x)}$ are parameters of implicit line for v_x , similarly $a^{(y)}$, $b^{(y)}$ and $d^{(y)}$ for v_y . This line has a normal vector

$$\mathbf{n}^{(x)} = [a^{(x)}, b^{(x)}]^T, \quad (23)$$

and thus the direction vector of the implicit line is

$$\mathbf{u}^{(x)} = [b^{(x)}, -a^{(x)}]^T. \quad (24)$$

The direction vector is a smooth approximation of the zero iso-line, see Fig. 7. For the v_y zero iso-line smooth approximation, we compute $\mathbf{u}^{(y)}$ using the same approach like for $\mathbf{u}^{(x)}$.

If two zero iso-lines are almost parallel (as in Fig. 4), the two corresponding direction vectors $\mathbf{u}^{(x)}$ and $\mathbf{u}^{(y)}$ computed using (24) are almost parallel as well. The angle θ between this two vectors is computed using

$$\theta = \text{acos} \left(\frac{\mathbf{u}^{(x)} \cdot \mathbf{u}^{(y)}}{\|\mathbf{u}^{(x)}\| \|\mathbf{u}^{(y)}\|} \right). \quad (25)$$

To distinguish cases, when vectors are almost parallel and when not, we can compute only the argument of function acos in (25)

$$v = \frac{\mathbf{u}^{(x)} \cdot \mathbf{u}^{(y)}}{\|\mathbf{u}^{(x)}\| \|\mathbf{u}^{(y)}\|}. \quad (26)$$

The two vectors $\mathbf{u}^{(x)}$ and $\mathbf{u}^{(y)}$ are almost parallel, when

$$|v| \rightarrow 1. \quad (27)$$

In our approach we are going to remove all critical points for which the condition (27) is true for some small ϵ in

$$|v| \geq 1 - \epsilon. \quad (28)$$

This critical points have only local character and are not important for the global character of the vector field.

The next type of critical points that have only local character is described in Figs. 5 and 6. To detect this type of critical points, we need to trace the zero iso-line of v_x and v_y and compute how long this zero iso-lines are. If one zero iso-line is shorter than some threshold, then the critical point have only local character and can be removed. The threshold length needs to be determined experimentally relatively to the size of input data set.

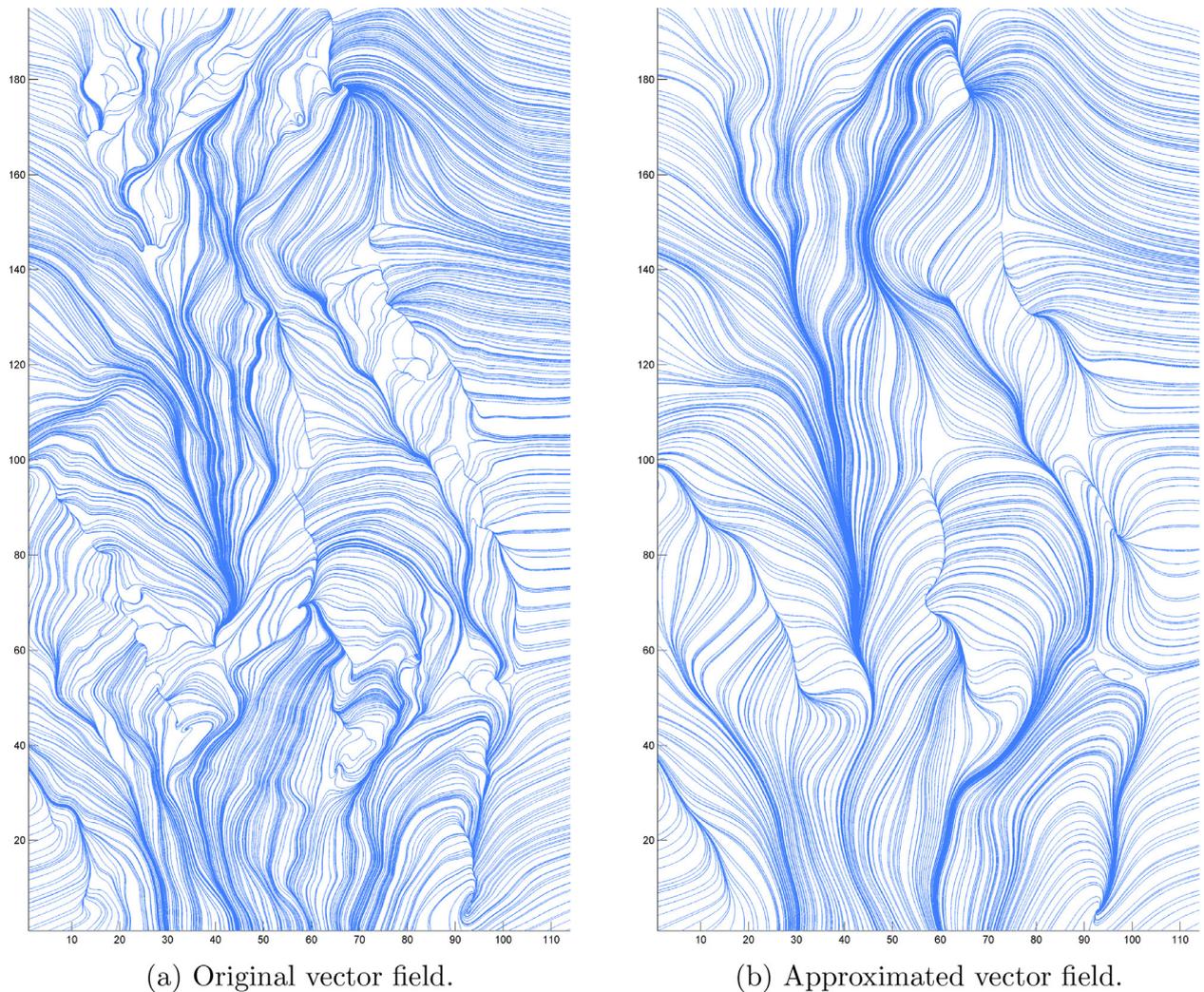


Fig. 11. Original vector field (a) and approximated vector field using the proposed approach (b).

4.2. Vector field RBF approximation

Now, we know the critical points that can be removed from the vector field as they change the vector field only locally and thus have no importance for the global character of the vector field. The remaining critical points need to be preserved in the approximated vector field. The vector field will be approximated using the RBF with Lagrange multipliers as described in Section 3.2, so that the important critical points will be preserved in the approximated vector field. For the RBF approximation, we need to determine the centers of radial basis functions.

Centers of the RBF approximation need to be located in the positions of the important critical points. The additional centers of RBF need to be located at the extremes of v_x , resp. v_y . The number of extremes will be too high for real data due to noise, measurement inaccuracies and calculation. Therefore, the v_x , resp. v_y , component of the vector field is smoothed using Gaussian low-pass filter before computing extremes. Now, we can compute the RBF approximation with Lagrange multiplier as described in Section 3.2. The conditions for Lagrange multiplier are the zero values at critical points locations.

After computing RBF approximation, we have an analytical description of the vector field. This approximated vector field can be used for visualization purposes as the global character of the vector field remains the same and only small local changes are neglected.

5. Results

In this section, we show the results of our proposed approach. This approach for vector field approximation using radial basis functions is especially convenient for visualization purposes, data understanding and data compression. For testing purposes, we used a numerical forecast wind data set taken from¹ [47]. This data set consists of $2.2 \cdot 10^4$ sampled points with associated vectors of wind flow, see Fig. 8a. The vector field contains 69 critical points as can be seen in Fig. 8b, but only few of them are important and have the global character.

We need to select only the important critical points and eliminate the others. First set of critical points, that can be eliminated, are critical points laying on a relatively short iso-lines. For processing only critical points with longer iso-lines length are considered. The threshold of zero iso-line length was experimentally selected as half of the minimum of x and y axis range, i.e.

$$\frac{\min\{range_x, range_y\}}{2} \quad (29)$$

¹ Data set of wind flow at a height of 10m over the surface of the Czech Republic courtesy of the Institute of Computer Science of the Czech Academy of Sciences.

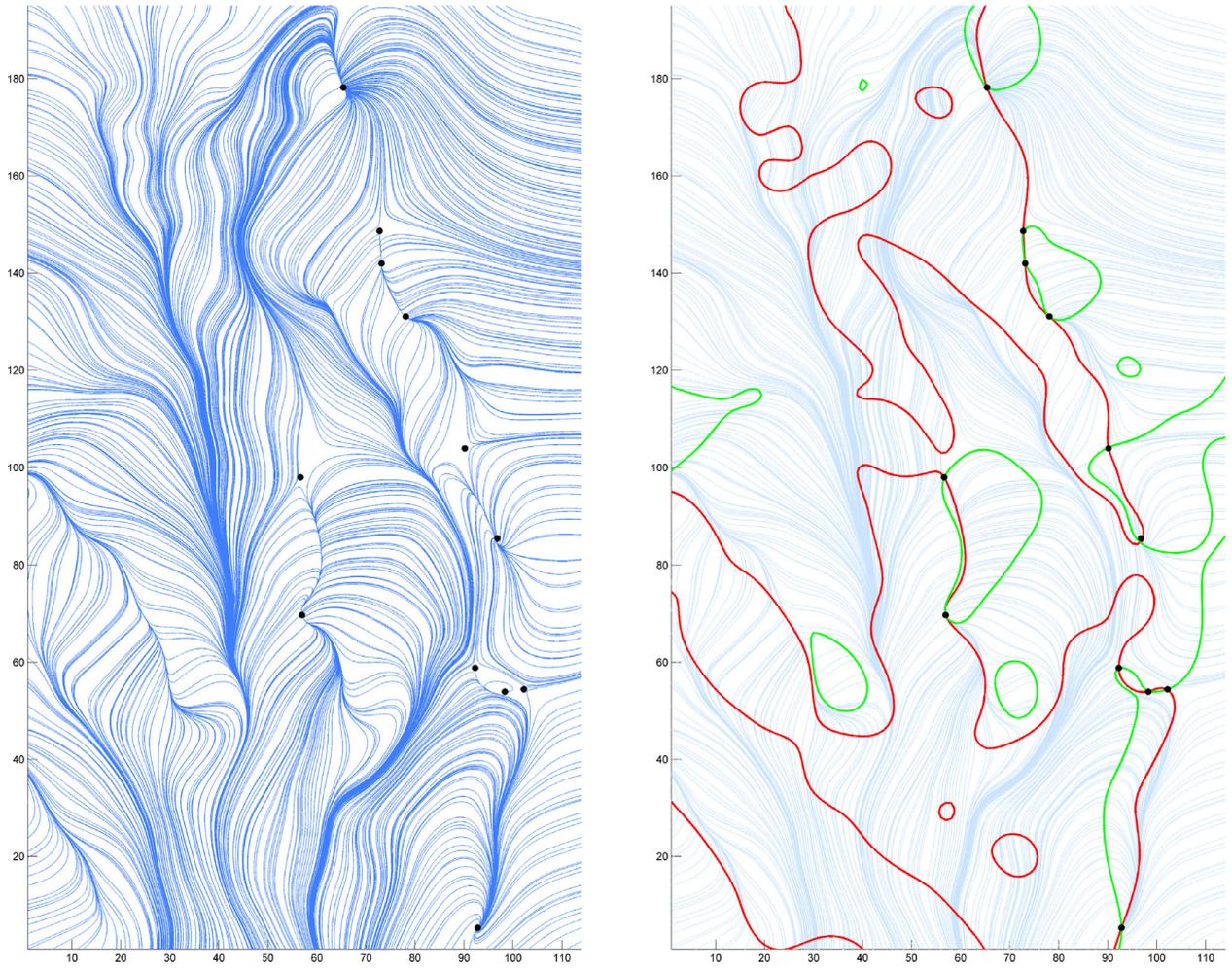
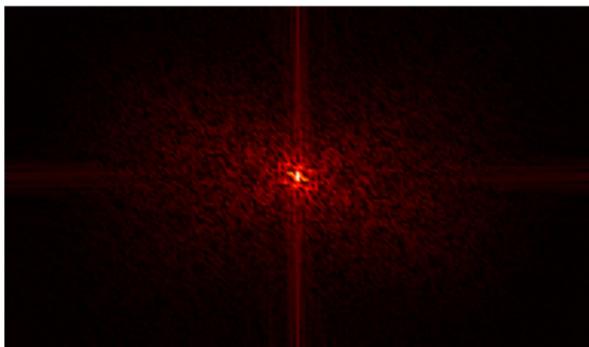
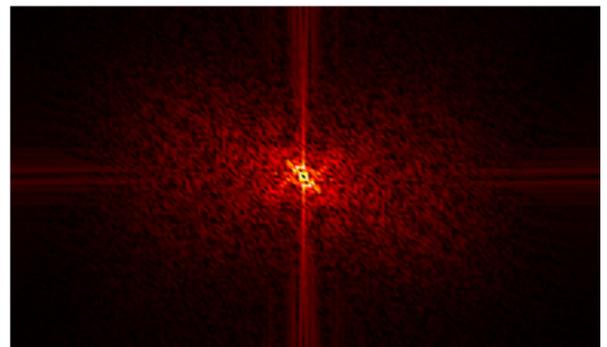


Fig. 12. Approximated vector field with its all critical points (a) and zero iso-lines of the approximated vector field (b).



(a) Frequency portrait.



(b) Phase offset.

Fig. 13. Fourier transform of the original data set. Figure (a) represents the frequency portrait and (b) represents the phase offset (both images are rotated 90 counterclockwise).

After ignoring this short zero iso-lines, we obtain reduced set of critical points, see Fig. 9a. To eliminate the second group of critical points, we need to compute (26) and compare it to some threshold value ϵ in (28). Experimentally, we selected the value of $\epsilon = 0.06$, which is relevant to an angle equal 20° , i.e. $\cos^{-1}(1 - \epsilon) = 20^\circ$. As the result we end up with only 12 critical points, this means that we eliminated 57 critical points, see Fig. 9b.

Both the vector field components v_x and v_y were approximated by

RBF using the Lagrange multiplier. The result of approximation is visualized at Fig. 10. It can be seen that both approximations have more smoothed course of the function compared to the original functions. This property is beneficial for the visualization purposes as the final vector field does not contain small fluctuations.

Having approximation of both vector field components v_x and v_y , we can visualize the approximated vector field, see Figs. 11b and 12. It can be seen that the approximated vector field has visually the same global

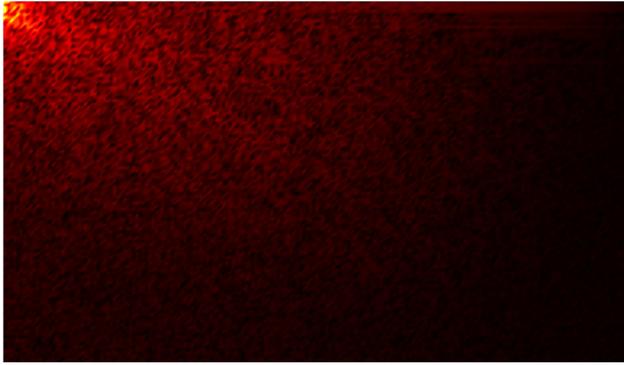


Fig. 14. Frequency portrait of the Cosine transform of the original data set.

character as the original vector field.

5.1. Comparison with existing approach

The proposed approach for vector field approximation using RBF needs to be compared to a different competing approach. We selected an approach which uses the discrete Fourier transform (DFT) to approximate a vector field and a second approach which uses the discrete Cosine transform (DCT) to approximate a vector field.

The discrete Fourier transform computes for both components v_x and v_y of the vector field the Fourier transform using the formula

$$F(\alpha, \beta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} v(x, y) e^{-2\pi i(x\alpha + y\beta)} dx dy, \tag{30}$$

where α and β are frequencies that represent the original data set after the Fourier transform, see Fig. 13. The inverse Fourier transform is computed using the formula

$$v(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(\alpha, \beta) e^{2\pi i(x\alpha + y\beta)} d\alpha d\beta. \tag{31}$$

The vector field can be approximated using the Fourier transform. We used the discrete Fourier transform in our experiments. Then we

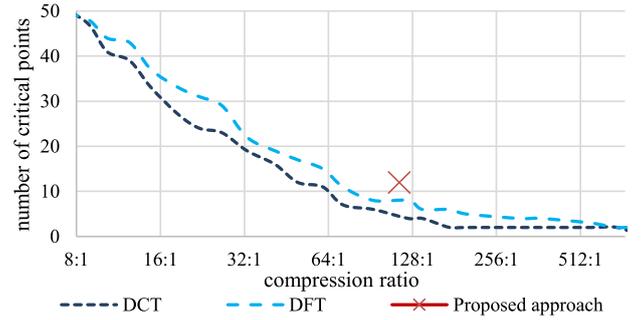


Fig. 16. Comparison of the number of critical points for the proposed RBF vector field approximation and the DFT and the DCT vector field approximation for different compression ratios. The total number of critical points in the input data set is 69.

used only some number of low frequency coefficients, thereby removing the high frequencies from the input data set and simplifying the vector field. This simplified vector field can be compared with our proposed approach for vector field RBF approximation.

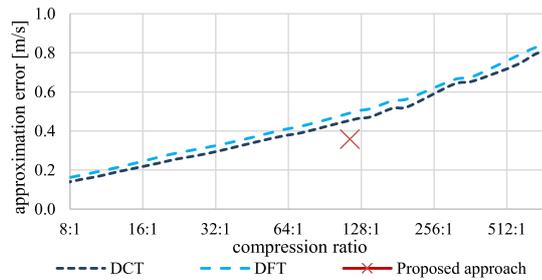
The discrete Cosine transform computes for both components v_x and v_y of the vector field the Cosine transform using the formula

$$F(\alpha, \beta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} v(x, y) \cos\left[\pi\left(x + \frac{1}{2}\right)\alpha\right] \cos\left[\pi\left(y + \frac{1}{2}\right)\beta\right] dx dy, \tag{32}$$

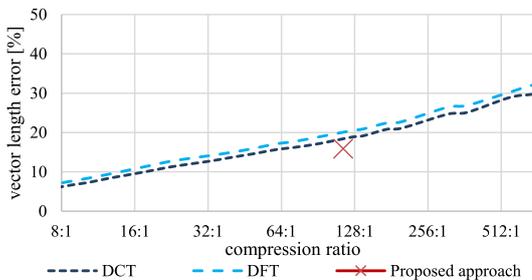
where α and β are frequencies that represent the original data set after the Cosine transform, see Fig. 14. The inverse Cosine transform is computed using the formula

$$v(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(\alpha, \beta) \cos\left[\pi\left(x + \frac{1}{2}\right)\alpha\right] \cos\left[\pi\left(y + \frac{1}{2}\right)\beta\right] d\alpha d\beta. \tag{33}$$

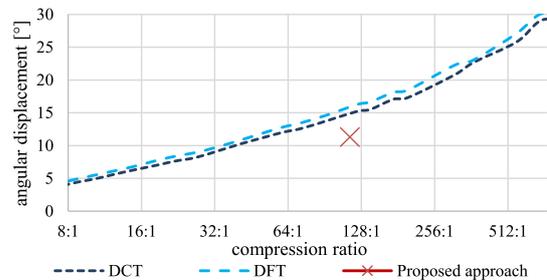
The vector field can be approximated using the Cosine transform. We used the discrete Cosine transform in our experiments. Then we used only some number of low frequency coefficients, thereby



(a) The average difference approximation error.

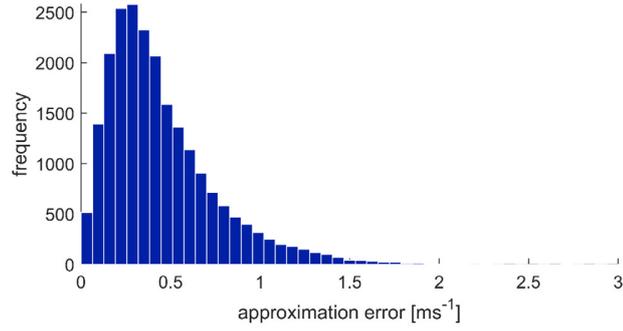


(b) The average vector length error.

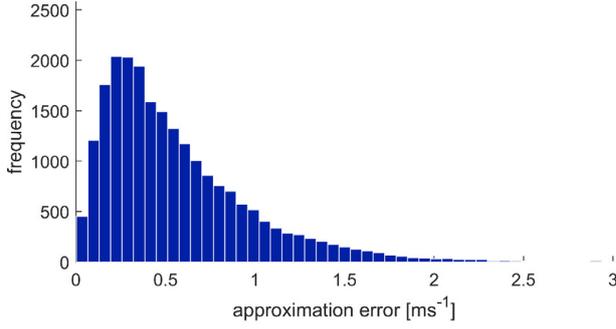


(c) The average angular displacement error.

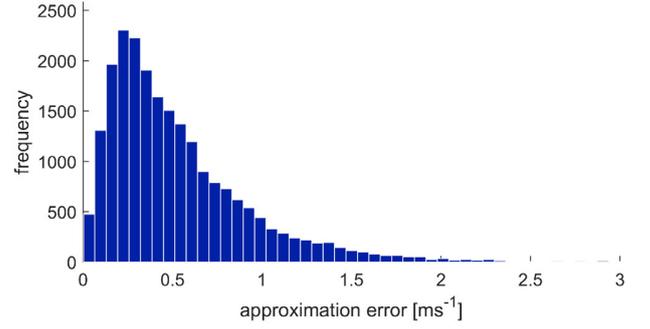
Fig. 15. Comparison of approximation error for the proposed RBF vector field approximation with the DFT vector field approximation and the DCT vector field approximation for different compression ratios. The total number of input points for vector field approximation is around $2.2 \cdot 10^4$; for approximation with compression ratio 8: 1 are used $2.8 \cdot 10^3$ reference points; and for compression ratio 512: 1 only 43 reference points are used.



(a) Proposed RBF approximation.

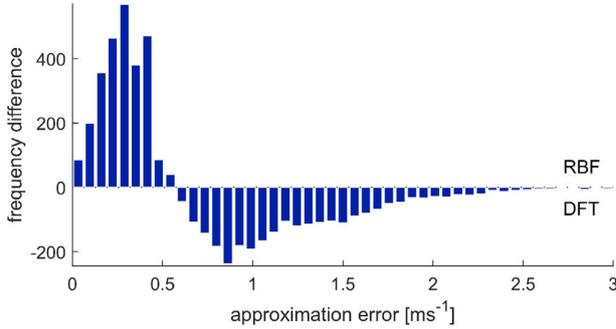


(b) DFT approximation.

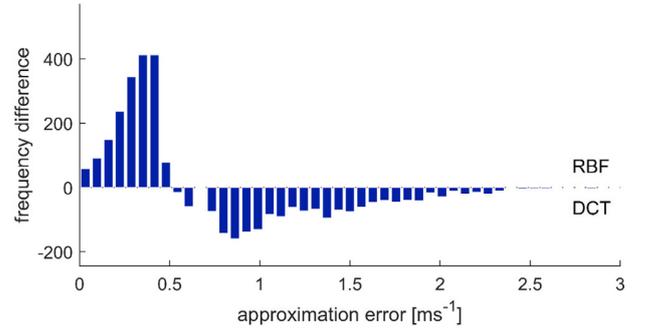


(c) DCT approximation.

Fig. 17. The histograms of vector field average difference approximation error for compression ratio 115: 1.



(a)



(b)

Fig. 18. The difference histogram of vector field average difference approximation error (“RBF” - “DFT”) (a) and (“RBF” - “DCT”) (b), see Fig. 17, for compression ratio 115: 1.

removing the high frequencies from the input data set and simplifying the vector field. This simplified vector field can be compared with our proposed approach for vector field RBF approximation.

The approximation error can be measured using different formulas. The first way is to compute the average difference of the approximated vector field and the original vector field. The average difference is computed using

$$Err = \frac{\sum_{i=1}^N \|v_i - \bar{v}_i\|}{N}, \quad (34)$$

where v_i is the approximated vector, \bar{v}_i is the original vector and N is the number of the original samples. The approximation error for different compression ratios is visualized in Fig. 15a. The total number of input points for vector field approximation is around $2.2 \cdot 10^4$; the average vector length, i.e. average speed, is 2.21 m/s.

Next, we can measure the average vector length error, i.e. the average speed error. This error is computed using

$$Err = \frac{\sum_{i=1}^N \|v_i\| - \|\bar{v}_i\|}{\sum_{i=1}^N \|\bar{v}_i\|}. \quad (35)$$

The average vector length error is divided by the average vector length of the original data set, i.e. this error is relative and the result for different compression ratios is visualized in Fig. 15b. As the data set contains critical points (zero points), standard relative error computation using the following formula cannot be applied, as there will be a division by zero and thus infinitely large relative error.

$$Err = \frac{1}{N} \sum_{i=1}^N \frac{\|v_i\| - \|\bar{v}_i\|}{\|\bar{v}_i\|}. \quad (36)$$

Last, we can measure the average angular displacement error using

$$Err = \frac{1}{N} \sum_{i=1}^N \text{acos} \left(\frac{v_i \cdot \bar{v}_i}{\|v_i\| \|\bar{v}_i\|} \right). \quad (37)$$

It can be seen that our proposed method have lower approximation

error in all the three cases mentioned before.

As the next step we tested the number of critical points in the approximated vector fields. Our aim is to reduce the number of critical points and preserve only the important critical points. However, if we are reducing the critical points even more we are still able to preserve more critical points in the approximated vector field than the DFT and the DCT approximation, see Fig. 16.

In spite of the fact that we know the distribution of approximation errors, it is hard to compare the two histograms of approximation errors in Fig. 17. Therefore we computed the difference histogram, see Fig. 18.

It can be seen that the proposed RBF approximation method has more smaller approximation errors and fewer higher approximation errors than the discrete Fourier approximation as well as the discrete cosine transform approximation, which means that the proposed method is significantly more accurate while preserving the same compression ratio and important features of the vector field.

6. Conclusion

We presented a new approach for simplification and approximation of a vector field using radial basis functions. Important critical points are preserved in the approximated vector field. The algorithm proved its simplicity and ability to approximate a complex vector field. The proposed algorithm was compared with the standard Fourier approximation algorithm and the Cosine approximation algorithm. It proved its capability of high compression while maintaining a low approximation error. The proposed method also leads to an analytical RBF form of the vector field which can be used for further processing. This is a significant advantage over other methods.

In the future the proposed approach will be extended to approximate 3D vector fields, as the visualization of complex and noisy vector fields can be confusing.

Acknowledgments

The authors would like to thank their colleagues at the University of West Bohemia, Plzen and colleagues at the Institute of computer science, the Czech academy of sciences, for their discussions and suggestions, and anonymous reviewers for their valuable comments and hints provided. The research was supported by project Czech Science Foundation (GACR) No. 17-05534S and partially by SGS 2016-013.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.advengsoft.2018.06.013.

References

- [1] Davis PJ. Interpolation and approximation. Courier Corporation 978-0-486-62495-2; 1975.
- [2] O'Rourke J. Computational geometry in C. Cambridge university press 0-521-64010-5; 1998.
- [3] Laramee RS, Hauser H, Zhao L, Post FH. Topology-based flow visualization, the state of the art. Topology-based methods in visualization. Springer; 2007. p. 1–19.
- [4] Tricoche X, Scheuermann G, Hagen H. Continuous topology simplification of planar vector fields. Proceedings of the conference on Visualization'01. IEEE Computer Society; 2001. p. 159–66.
- [5] Tricoche X, Scheuermann G, Hagen H. A topology simplification method for 2D vector fields. Visualization 2000. Proceedings. IEEE; 2000. p. 359–66.
- [6] Weinkauff T, Theisel H, Shi K, Hege H-C, Seidel H-P. Extracting higher order critical points and topological simplification of 3D vector fields. Visualization, 2005. VIS 05. IEEE. IEEE; 2005. p. 559–66.
- [7] Theisel H, Rossel C, Seidel H-P. Combining topological simplification and topology preserving compression for 2D vector fields. Computer graphics and applications, 2003. Proceedings. 11th Pacific conference on. IEEE; 2003. p. 419–23.
- [8] Dey TK, Levine JA, Wenger R. A delaunay simplification algorithm for vector fields. Computer graphics and applications, 2007. PG'07. 15th Pacific conference on. IEEE; 2007. p. 281–90.
- [9] Wong PC, Foote H, Leung R, Jurrus E, Adams D, Thomas J. Vector fields simplification—a case study of visualizing climate modeling and simulation data sets. Visualization 2000. Proceedings. IEEE; 2000. p. 485–8.
- [10] de Leeuw W, van Liere R. Multi-level topology for flow visualization. Computers & Graphics 2000;24(3):325–31.
- [11] Skrabpa P, Wang B, Chen G, Rosen P. 2D vector field simplification based on robustness. Visualization symposium (PacificVis), 2014 IEEE Pacific. IEEE; 2014. p. 49–56.
- [12] Skrabpa P, Wang B, Chen G, Rosen P. Robustness-based simplification of 2D steady and unsteady vector fields. IEEE Trans Vis Comput Graph 2013;21(8):930–44.
- [13] Günther D, Jacobson A, Reininghaus J, Seidel H-P, Sorkine-Hornung O, Weinkauff T. Fast and memory-efficiently topological denoising of 2D and 3D scalar fields. IEEE Trans Vis Comput Graph 2014;20(12):2585–94.
- [14] Cabrera DAC, Gonzalez-Casanova P, Gout C, Juárez LH, Reséndiz LR. Vector field approximation using radial basis functions. J Comput Appl Math 2013;240:163–73.
- [15] Koch S, Kasten J, Wiebel A, Scheuermann G, Hlawitschka M. 2D Vector field approximation using linear neighborhoods. Vis Comput 2016;32(12):1563–78.
- [16] Blazek J. Computational fluid dynamics: principles and applications. Butterworth-Heinemann 978-0-08-044506-9; 2015.
- [17] Sanderse B, Pijl S, Koren B. Review of computational fluid dynamics for wind turbine wake aerodynamics. Wind Energy 2011;14(7):799–819.
- [18] Molina-Aiz F, Fatnassi H, Boulard T, Roy J, Valera D. Comparison of finite element and finite volume methods for simulation of natural ventilation in greenhouses. Comput Electron Agric 2010;72(2):69–86.
- [19] Chung ET, Efendiev Y, Lee CS. Mixed generalized multiscale finite element methods and applications. Multiscale Model Simul 2015;13(1):338–66.
- [20] Helman J, Hesselink L. Representation and display of vector field topology in fluid flow data sets. IEEE Comput 1989;22(8):27–36.
- [21] Pan R, Skala V. A two-level approach to implicit surface modeling with compactly supported radial basis functions. Eng Comput 2011;27(3):299–307.
- [22] Fasshauer GE. Meshfree approximation methods with MATLAB. 6. World Scientific; 2007.
- [23] Skala V. Meshless interpolations for computer graphics, visualization and games. In: Eurographics 2015 - Tutorials. 2015.
- [24] Larsson E, Fornberg B. A numerical study of some radial basis function based solution methods for elliptic PDEs. Comput Math Appl 2003;46(5):891–902.
- [25] Zhang X, Song KZ, Lu MW, Liu X. Meshless methods based on collocation with radial basis functions. Comput Mech 2000;26(4):333–43.
- [26] Uhlir K, Skala V. Reconstruction of damaged images using radial basis functions. Signal processing conference, 2005 13th European. IEEE; 2005. p. 1–4.
- [27] Karim A, Adeli H. Radial basis function neural network for work zone capacity and queue estimation. J Transp Eng 2003;129(5):494–503.
- [28] Ghosh-Dastidar S, Adeli H, Dadmehr N. Principal component analysis-enhanced cosine radial basis function neural network for robust epilepsy and seizure detection. IEEE Trans Biomed Eng 2008;55(2):512–8.
- [29] Yingwei L, Sundararajan N, Saratchandran P. Performance evaluation of a sequential minimal radial basis function (RBF) neural network learning algorithm. IEEE Trans Neural Networks 1998;9(2):308–18.
- [30] Majdisova Z, Skala V. Big geo data surface approximation using radial basis functions: a comparative study. Comput Geosci 2017;109:51–8.
- [31] Pan R, Skala V. Surface reconstruction with higher-order smoothness. Vis Comput 2012;28(2):155–62.
- [32] Prakash G, Kulkarni M, Sripathi U. Using RBF neural networks and Kullback-Leibler distance to classify channel models in free space optics. Optical engineering (ICOE), 2012 international conference on. IEEE; 2012. p. 1–6.
- [33] Schagen IP. Interpolation in two dimensions - a new technique. IMA J Appl Math 1979;23(1):53–9.
- [34] Majdisova Z, Skala V. Radial basis function approximations: comparison and applications. Appl Math Model 2017;51:728–43.
- [35] Skala V. RBF Interpolation with CSRBF of large data sets. Procedia Comput Sci 2017;108:2433–7.
- [36] Wendland H. Computational aspects of radial basis function approximation. Stud Comput Math 2006;12:231–56.
- [37] Hardy RL. Multiquadric equations of topography and other irregular surfaces. J Geophys Res 1971;76(8):1905–15.
- [38] Skala V. Fast interpolation and approximation of scattered multidimensional and dynamic data using radial basis functions. WSEAS Trans Math 2013;12(5):501–11.
- [39] Lee J, Strazicich MC. Minimum lagrange multiplier unit root test with two structural breaks. Rev Econ Stat 2003;85(4):1082–9.
- [40] Glowinski R, Pan T-W, Hesla TI, Joseph DD. A distributed lagrange multiplier/fictitious domain method for particulate flows. Int J Multiphase Flow 1999;25(5):755–94.
- [41] Liu I-S. Method of lagrange multipliers for exploitation of the entropy principle. Arch Ration Mech Anal 1972;46(2):131–48.
- [42] Reininghaus J, Kotava N, Guenther D, Kasten J, Hagen H, Hotz I. A scale space based persistence measure for critical points in 2D scalar fields. IEEE Trans Vis Comput Graph 2011;17(12):2045–52.
- [43] Wang W, Wang W, Li S. Detection and classification of critical points in piecewise linear vector fields. J Visualization; 1–15.
- [44] Mann S, Rockwood A. Computing singularities of 3D vector fields with geometric algebra. Proceedings of the conference on Visualization'02. IEEE Computer Society; 2002. p. 283–90.
- [45] Bhatia H, Gyulassy A, Wang H, Bremer P-T, Pascucci V. Robust detection of singularities in vector fields. Topological methods in data analysis and Visualization III. Springer; 2014. p. 3–18.
- [46] Skala V. Total least square error computation in E2: a new simple, fast and robust algorithm. Proceedings of the 33rd computer graphics international. ACM; 2016. p. 1–4.
- [47] Corbet L, Vlček O, Eben K, Liczki J, Benešová N, Modlák M. Regional air quality forecasting for the Czech Republic. 15th International conference on harmonisation within atmospheric dispersion modelling for regulatory purposes, HARMO. 2013. p. 150–4.

4.2 3D vector field approximation and critical points reduction using Radial basis functions

The paper [Smolik et al., 2019] is an extension of the journal paper [Smolik et al., 2018] and proposes an approach for 3D vector field approximation. The proposed approach was especially designed to simplify and compress tornado data sets. This kind of vector fields have one main feature, which is the rotation around the vertical axis. Such vector field can be cut in 2D horizontal slices. In each cut are lots of vortices, some of them are important and some of them can be discarded as they have only small local influence in the vector field.

We used the algorithm [Smolik et al., 2018] to select the important critical points that should be preserved in the final approximation of vector field. Due to this approach it is possible to reduce the number of critical points in the whole 3D vector field. After this selection is performed the RBF approximation with specific location of radial basis functions is computed.

The result of this approach is a simplified 3D vector field, which has similar behavior as the original one. Moreover, we achieved very high compression ratio ($7 \cdot 10^3 : 1$), while maintaining low approximation error.

Citation:

- Michal Smolik, Vaclav Skala, and Zuzana Majdisova. 3D vector field approximation and critical points reduction using radial basis functions. In *International journal of mechanics*, 13(1):100–103, 2019. (CiteScore = 1.00)

3D Vector Field Approximation and Critical Points Reduction Using Radial Basis Functions

Michal Smolik, Vaclav Skala and Zuzana Majdisova

Abstract—Vector field simplification aims to reduce the complexity of the flow by removing features according to their relevance and importance. However, the important features as critical points with a large range of influence should be preserved. We present a new approach for vector field simplification and approximation using Radial basis functions. The experiments proved the ability to approximate complex 3D tornado data set. In addition, a significant contribution of the proposed method is also an analytical form of the vector field which can be used in further processing. The abstract goes here.

Keywords—Vector field, Radial basis functions, critical point, tornado, simplification, approximation, data compression, visualization.

I. INTRODUCTION

THE vector fields are often very large and complex data set. To process and visualize such kind of data, the approximation and simplification techniques are used. The summary of topology based flow visualization techniques is presented in [1]. Most of this techniques tries to simplify the vector topology, either by removing or collapsing critical points, or by smoothing the vector field to remove small unimportant changes in the flow [2], [3], or by simplifying the topological skeleton of the vector field [4], [5].

The paper [6] simplifies the topology of vector field by collapsing critical points. The algorithm processes vector fields defined on a triangulation of the flow domain, i.e. planar vector fields. During the simplification, there are no topological changes in the triangulation. During the simplification are iteratively selected pairs of critical points that can be collapsed. The critical points can be collapsed into one higher order critical point or they can reset each other, i.e. both critical points have opposite Poincare index. Another approach is presented in [7]. There is defined some maximal distance of two critical points to collapse them. All critical points inside the selected radius are collapsed into one higher order critical point. The paper [8] focuses on visualization of complex 3D vector fields. The authors prove that vector field inside some area can be described with the 2D vector field on the surface around this area. Using this knowledge, they create symbols and visualization based on different behavior of the

vector field inside some area with many critical points. The paper [9] presents an approach for simplification of vector field topology, while preserving important features of the vector field, i.e. critical points and separatrices. Each topology feature gets computed a weight that means the importance. According this weights is then simplified the vector field topology. The vertex deletion from the Delaunay triangulation is used in [10] to simplify the vector field. Based on some metrics is determined if a vertex can be removed from the Delaunay triangulation, so that the vector field will not change significantly. The authors prove that this can be determined using only local neighbors and there is no need to compute this change for the whole vector field. The paper [11] filters out the less important and sometimes even sporadic critical points. The filtering is based on the vector field vorticity and is best suited for regional climate modeling and simulation.

Many approaches for vector field approximation use the Radial basis function (RBF) method [12], [13], [14]. The paper [15] presents an approach for large scattered 3D vector field approximation. It uses the space subdivision to process and speed-up the approximation. The comparison of vector field approximation with local radial basis functions and global radial basis functions is presented in [16]. The paper [17] uses the RBF interpolation in numerical simulation of divergence-free vector fields. Another approaches use the second order derive to describe features of the vector field [18], [19].

II. 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, we use modified algorithm described in [20] to approximate the 3D vector fields. In this paper, we will especially focus on approximation of the EF5 tornado data set (from [21])¹.

The 3D data set is divided into 2D horizontal slices, as the main swirl plane is horizontal. Each 2D vector field slice contains high number of critical points that we want to reduce. We will use the algorithm described in [20] to determine the important critical points. The important critical points should be presented in the final approximated vector field, while the unimportant critical points should be removed from the vector field. The important critical points will be the centers for radial basis functions (RBF) [22]. The radial basis functions should

The research was supported by projects Czech Science Foundation (GACR) No. 17-05534S and partially by SGS 2019-016.

M. Smolik is with the Faculty of Applied Sciences, University of West Bohemia, Plzen, Czech Republic, e-mail: smolik@kiv.zcu.cz.

V. Skala is with the Faculty of Applied Sciences, University of West Bohemia, Plzen, Czech Republic, e-mail: skala@kiv.zcu.cz, web: www.VaclavSkala.eu.

Z. Majdisova is with the Faculty of Applied Sciences, University of West Bohemia, Plzen, Czech Republic, e-mail: majdisz@kiv.zcu.cz.

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

be also placed at the extremes of v_x , v_y , resp. v_z . However the number of extremes is too high and thus this extremes need to be reduced. For this purpose, we use the Gaussian low-pass filter to determine only the important extremes and to discard the small local extremes. The last additional centers of RBF are located at the data set bounding box vertices. The vector field is then approximated using RBF as

$$\mathbf{v}(\mathbf{x}) = \sum_{j=1}^M \lambda_j \varphi(\|\mathbf{x} - \boldsymbol{\xi}_j\|) \quad (1)$$

where λ_j are weights of the RBFs, M is the number of the radial basis functions, φ is the radial basis function and $\boldsymbol{\xi}_j$ are centers of radial basis functions. It is similar as in the potential field case [23]. For a given vector field data set $\{\mathbf{x}_i, \mathbf{v}_i\}_{i=1}^N$, where $N \gg M$, the following overdetermined linear system of equations is obtained:

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

The result of the RBF approximation is an analytical description of the 3D vector field. This is the advantage over other existing methods that use the triangulation, resp. tetrahedrization. The analytical description can be used for further processing of the vector field.

III. EXPERIMENTAL RESULTS

We tested the proposed approach using the EF5 tornado data set from [21]. For the testing purposes, we selected the central part of the data set, where the tornado is located. The size of the vector field data set for approximation is $8 \cdot 10^6$ points (see Fig. 4a).

First step of the proposed approach is the reduction of critical points in horizontal slices of the vector field. The input data set contains 28 902 critical points (see Fig. 1a) and after reduction using algorithm [20], we end up with only 490 critical points (see Fig. 1b).

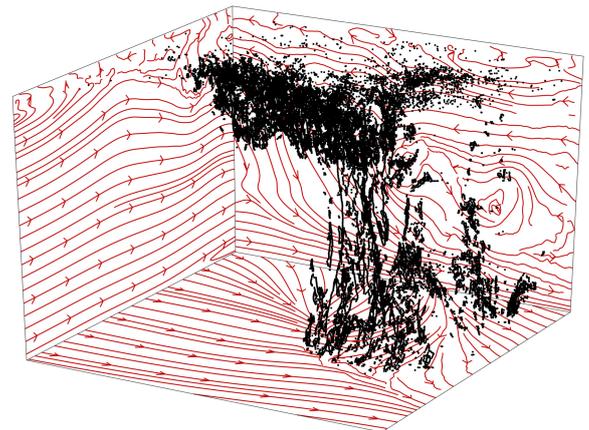
The centers of the RBFs are located at critical points and at the extremes of v_x , v_y , resp. v_z . After smoothing using Gaussian low-pass filter, we located 666 extremes. The total number of centers for radial basis functions is thus 1164 points (see Fig. 2).

The data set is very large for RBF approximation, thus we need to use the local RBF [24], [25], [26] to reduce needed memory (see [20] for selection of RBF). The RBF used for vector field approximation is

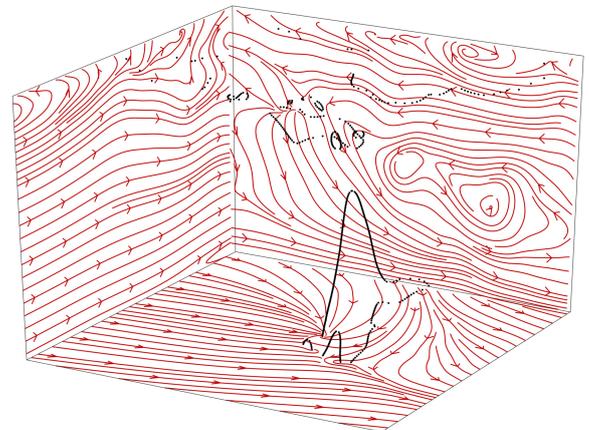
$$\varphi(r) = (1 - r)_+^4 (4r + 1) \quad (3)$$

as it gives the best approximation results based on experimental results and is C^2 continuous, which is appropriate when computing derivatives of the vector field.

After the RBF approximation of the input vector field data set, we visualized the results. The 2D vector field horizontal slices are visualized in Fig. 3. It can be seen, that the global character of the vector field remains the same. There are only local differences as the approximated vector field is



(a) All critical points.



(b) Reduced critical points.

Fig. 1. Visualization of 2D critical points located at horizontal slices of vector field data set (28 902 points (a) and 490 points (b)).

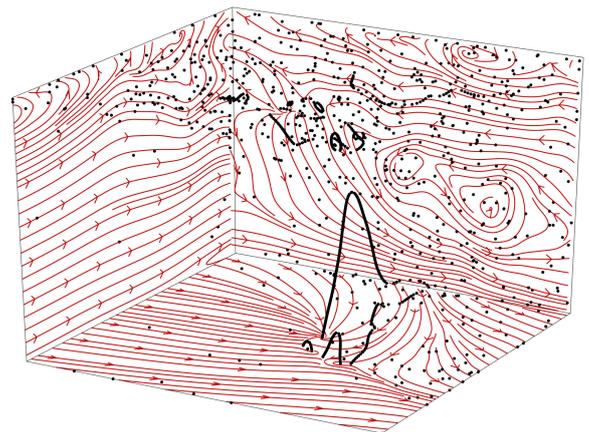


Fig. 2. Visualization of centers of radial basis functions (1164 points).

smoother. There can be seen some small differences in the global direction of the flow, however the compression ratio is about $7 \cdot 10^3 : 1$ which is very high.

Fig. 4 presents the visualization of 3D approximated vector

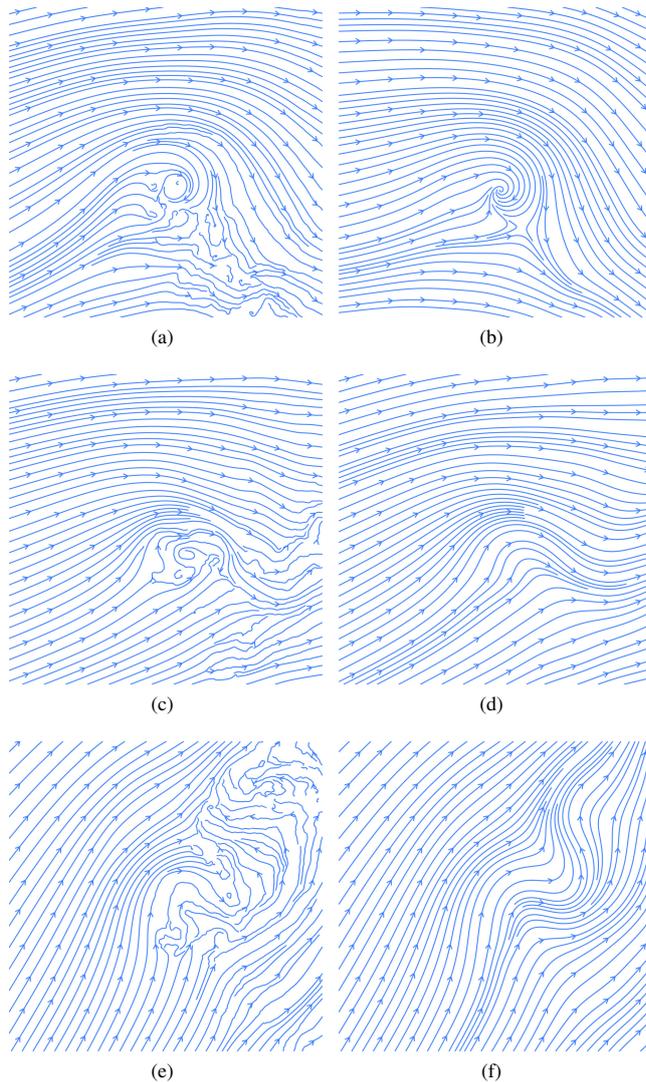


Fig. 3. Visualization of 2D vector field horizontal slices. Left column represents the original vector field and the right column represents the RBF approximated vector field.

field. It can be seen that the main vortex of the tornado has similar shape to the original one. The original one contains just many tiny details. However this is again due to very high compression ratio ($7 \cdot 10^3 : 1$).

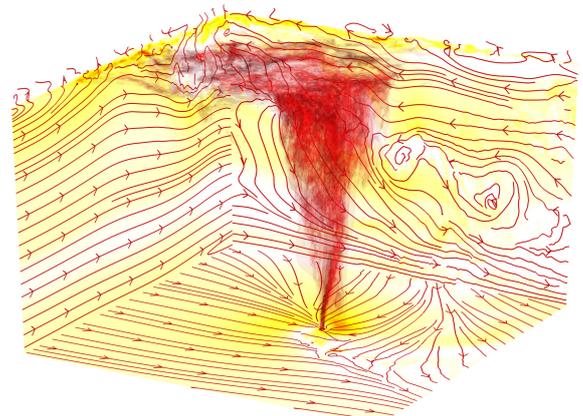
The approximation error can be measured using different formulas. The first way is to compute the average difference of the approximated vector field and the original vector field. The average difference is computed using

$$Err = \frac{\sum_{i=1}^N \|\mathbf{v}_i - \bar{\mathbf{v}}_i\|}{N}, \quad (4)$$

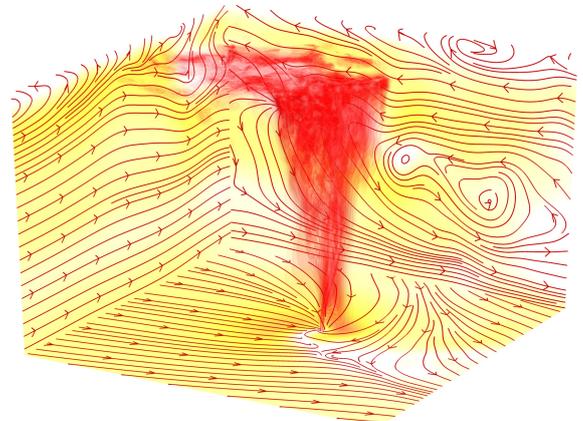
where \mathbf{v}_i is the approximated vector, $\bar{\mathbf{v}}_i$ is the original vector and N is the number of the original samples. The approximation error is visualized in Fig. 5a.

Next, we can measure the average vector length error, i.e. the average speed error. This error is computed using

$$Err = \left| \|\mathbf{v}_i\| - \|\bar{\mathbf{v}}_i\| \right|. \quad (5)$$



(a) Original vector field.



(b) Approximated vector field.

Fig. 4. 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 computed speed error is visualized in Fig. 5b. The speed error of vector field approximation is 3.8 ms^{-1} and the average speed of the vector field is 18.7 ms^{-1} .

IV. CONCLUSION

We presented a new approach for simplification and approximation of complex and large 3D vector fields using RBF. The proposed approach preserves during the simplification and approximation the important critical points and thus it preserves the global character of the 3D vector field as well. As the result, we end up with an analytical description of the approximated vector field, which can be used for further processing of the simplified vector field.

In the future, the proposed approach will be modified to reduce the critical points already in 3D instead of in 2D vector field slices.

ACKNOWLEDGMENTS

The authors would like to thank their colleagues at the University of West Bohemia, Plzen, for their discussions and suggestions, and anonymous reviewers for their valuable

comments and hints provided. The research was supported by projects Czech Science Foundation (GACR) No. 17-05534S and partially by SGS 2019-016.

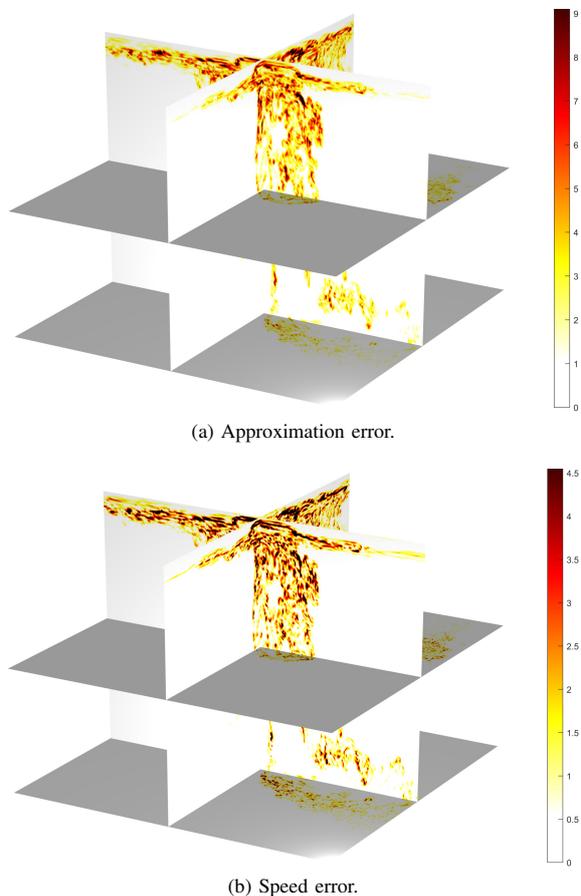


Fig. 5. Visualization of vector field approximation error (a) and visualization of speed error of approximated vector field (b).

REFERENCES

- [1] R. S. Laramée, H. Hauser, L. Zhao, and F. H. Post, "Topology-based flow visualization, the state of the art," in *Topology-based methods in visualization*. Springer, 2007, pp. 1–19.
- [2] D. Günther, A. Jacobson, J. Reininghaus, H.-P. Seidel, O. Sorkine-Hornung, and T. Weinkauff, "Fast and memory-efficiently topological denoising of 2D and 3D scalar fields," *IEEE transactions on visualization and computer graphics*, vol. 20, no. 12, pp. 2585–2594, 2014.
- [3] S. Koch, J. Kasten, A. Wiebel, G. Scheuermann, and M. Hlawitschka, "2D vector field approximation using linear neighborhoods," *The Visual Computer*, vol. 32, no. 12, pp. 1563–1578, 2016.
- [4] P. Skraba, B. Wang, G. Chen, and P. Rosen, "2D vector field simplification based on robustness," in *Visualization Symposium (PacificVis), 2014 IEEE Pacific*. IEEE, 2014, pp. 49–56.
- [5] P. Skraba, B. Wang, and G. Chen, "Robustness-based simplification of 2D steady and unsteady vector fields," *IEEE transactions on visualization and computer graphics*, vol. 21, no. 8, pp. 930–944, 2015.
- [6] X. Tricoche, G. Scheuermann, and H. Hagen, "Continuous topology simplification of planar vector fields," in *Proceedings of the conference on Visualization'01*. IEEE Computer Society, 2001, pp. 159–166.
- [7] X. Tricoche and G. Scheuermann, "A topology simplification method for 2D vector fields," in *Visualization 2000. Proceedings*. IEEE, 2000, pp. 359–366.
- [8] T. Weinkauff, 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*. IEEE, 2005, pp. 559–566.
- [9] H. Theisel, C. Rossli, and H.-P. Seidel, "Combining topological simplification and topology preserving compression for 2D vector fields," in *11th Pacific Conference on Computer Graphics and Applications*. IEEE, 2003, pp. 419–423.
- [10] 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*. IEEE, 2007, pp. 281–290.
- [11] P. C. Wong, H. Foote, R. Leung, E. Jurrus, D. Adams, and J. Thomas, "Vector fields simplification—a case study of visualizing climate modeling and simulation data sets," in *Visualization 2000. Proceedings*. IEEE, 2000, pp. 485–488.
- [12] G. E. Fasshauer, *Meshfree approximation methods with MATLAB*. World Scientific, 2007, vol. 6.
- [13] M. E. Biancolini, *Fast Radial Basis Functions for Engineering Applications*, 1st ed. Springer International Publishing, 2017.
- [14] Z. Majdisova and V. Skala, "Radial basis function approximations: Comparison and applications," *Applied Mathematical Modelling*, vol. 51, pp. 728–743, 2017.
- [15] M. Smolik and V. Skala, "Efficient simple large scattered 3D vector fields radial basis function approximation using space subdivision," in *Computational Science and Its Application, ICSSA 2019 proceedings*, 2019, pp. 337–350.
- [16] 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*, vol. 240, pp. 163–173, 2013.
- [17] C. P. McNally, "Divergence-free interpolation of vector fields from point values—exact $\nabla \cdot B = 0$ in numerical simulations," *Monthly Notices of the Royal Astronomical Society: Letters*, vol. 413, no. 1, pp. 76–80, 2011.
- [18] M. Smolik and V. Skala, "Classification of critical points using a second order derivative," in *ICCS 2017, Procedia Computer Science*, vol. 108. Elsevier, 2017, pp. 2373–2377.
- [19] —, "Vector field second order derivative approximation and geometrical characteristics," in *ICCSA 2017 Conf.*, vol. 10404. Springer, 2017, pp. 148–158.
- [20] M. Smolik, V. Skala, and Z. Majdisova, "Vector field radial basis function approximation," *Advances in Engineering Software*, vol. 123, no. 1, pp. 117–129, 2018.
- [21] L. Orf, R. Wilhelmson, and L. Wicker, "Visualization of a simulated long-track ef5 tornado embedded within a supercell thunderstorm," *Parallel Computing*, vol. 55, pp. 28–34, 2016.
- [22] M. Smolik and V. Skala, "Large scattered data interpolation with radial basis functions and space subdivision," *Integrated Computer-Aided Engineering*, vol. 25, no. 1, pp. 49–26, 2018.
- [23] V. Skala, "Fast interpolation and approximation of scattered multidimensional and dynamic data using radial basis functions," *WSEAS Transaction on Mathematics*, vol. 12, no. 5, pp. 501–511, 2013.
- [24] R. L. Hardy, "Theory and applications of the multiquadric-biharmonic method 20 years of discovery 1968-1988," *Computers & Mathematics with Applications*, vol. 19, pp. 163–208, 1990.
- [25] F. Menandro, "Two new classes of compactly supported radial basis functions for approximation of discrete and continuous data," *Engineering Reports*, vol. 2019;1:e12028, pp. 1–30, 2019.
- [26] F. C. M. Menandro, "Two new classes of compactly supported radial basis functions for approximation of discrete and continuous data," *Engineering Reports*, vol. 1, no. 1, pp. 1–30, 2019.

4.3 Large scattered data interpolation with Radial basis functions and space subdivision

Interpolation of scattered data using Radial basis functions leads to a solution of linear systems. With increasing the size of the input data, the memory requirements are increasing even more, as mainly the memory requirements for RBF interpolation are $O(N^2)$. Because of that, there is no possibility to perform the basic RBF interpolation on data sets containing millions of points, or even more. The paper [Smolik and Skala, 2018] proposes an approach for interpolation of such large data sets using RBF.

To speed-up the RBF interpolation and mainly to make it even possible to calculate the RBF interpolation of a large scattered data set on a standard computer, the proposed approach uses the space subdivision. The data set is divided into several overlapping cells. Each cell is interpolated using the RBF with a local radial basis function. Then those interpolations are blended together with the proposed blending schema. After blending, there are no patterns formed by blending and the final result is the correct interpolation of the whole input data set, i.e. the interpolation function goes through all the input points.

The proposed approach offers very significant speed-up of the RBF interpolation compared to the standard RBF interpolation, i.e. for subdivision into 256 overlapping $2D$ cells the expected speed-up is around $(10^4 : 1)$. Moreover, the memory requirements are much lower as well.

Citation:

- Michal Smolik and Vaclav Skala. Large scattered data interpolation with radial basis functions and space subdivision. *Integrated Computer-Aided Engineering*, 25(1):49–62, 2018. (IF = 4.904)

Large scattered data interpolation with radial basis functions and space subdivision

Michal Smolik* and Vaclav Skala

Department of Computer Science and Engineering, Faculty of Applied Sciences, University of West Bohemia, Plzen, Czech Republic

Abstract. We propose a new approach for the radial basis function (RBF) interpolation of large scattered data sets. It uses the space subdivision technique into independent cells allowing processing of large data sets with low memory requirements and offering high computation speed, together with the possibility of parallel processing as each cell can be processed independently. The proposed RBF interpolation was tested on both synthetic and real data sets. It proved its simplicity, robustness and the ability to handle large data sets together with significant speed-up. In the case of parallel processing, speed-up was experimentally proved when 2 and 4 threads were used.

Keywords: Radial basis functions, interpolation, large data, space subdivision, scattered data

1. Introduction

Interpolation and approximation are probably the most frequent operations used in computational techniques [1]. Several techniques have been developed for data interpolation, but they require some kind of data “ordering”, e.g. structured mesh, rectangular mesh, unstructured mesh etc. A typical example is a solution of partial differential equations (PDE), where derivatives are replaced by differences and rectangular or hexagonal meshes are used in the vast majority of cases. However, in many engineering problems, data are not ordered and they are scattered in k -dimensional space, in general. The k -dimensional space is sometimes not only spatial but also contains a time dimension or a dimension relating to age or temperature or other environmental conditions. Usually, in technical applications the scattered data are tessellated using triangulation, but this approach is quite prohibitive for the case of k -dimensional data interpolation because of the computational cost [2].

There exist some techniques using space subdivision to compute a radial basis function (RBF) interpolation. Data point division into sub-domains using an adaptive octree subdivision method and then blending these local functions together with partition of unity is used in [3]. This work is an extension of well-known [4], which uses the multi-level partition of unity to construct surface models from very large sets of points. Spatial down sampling to construct a coarse-to-fine hierarchy of point sets is used in [5]. They interpolate the sets starting from the coarsest level and then they interpolate a point set of the hierarchy, as an offsetting of the interpolating function computed at the previous level. [6] proposed a highly parallel algorithm for RBF interpolation with the time complexity of $O(N)$. The algorithm uses a generalized minimal residual method (GMRES) iterative solver [7] with a restricted additive Schwarz method [8]. The algorithm [9] relies on PetRBF [6]. It improves PetRBF for surface reconstruction and graphics processing unit (GPU) acceleration. It shows how to make a suitable choice of the algorithm parameters for accurate reconstruction from synthetic, real or incomplete datasets. The algorithm uses domain decomposition to acquire high parallelization. The solution of the original system is built up by solving set of smaller subproblems that interact

*Corresponding author: Michal Smolik, Department of Computer Science and Engineering, Faculty of Applied Sciences, University of West Bohemia, Plzen, Czech Republic. E-mail: smolik@kiv.zcu.cz. URL: <http://www.VaclavSkala.eu>.

through their interfaces. [10] optimizes the positions and the weights of the RBF centers and then combines them with a hierarchical domain decomposition technique for the RBF approximation. Other approaches using domain decomposition for the RBF interpolation are [11] which focuses on the parallelization of RBF interpolation with its application for mesh deformation, [12] performs RBF interpolation on divided input points and then iteratively updates all RBF coefficients to create final interpolation, [13] uses multiscale collocation and preconditioners to decrease the condition number of the interpolation matrix, [14] combines the RBF method and the least squares approximation cardinal basis functions (ACBF) preconditioning technique with the domain decomposition method.

All these approaches use space subdivision to compute the RBF interpolation or approximation, but their joining phase is usually not easy to implement. One has many independent interpolations and needs to join them together. These interpolations usually have some overlapping parts and to join them together we need to solve additional systems of equations or iteratively update the resulting interpolation. Our aim is to improve this joining phase and speed-up the calculation of the RBF interpolation as well.

Another approach using virtual points for approximation is used in [15,16]. Of course, there are other meshless techniques than RBF, such as discrete smooth interpolation (DSI) [17], which avoids explicitly computing a function defined everywhere and produces values only at the grid points instead. [18,19] is based on statistical models that include autocorrelation. The scattered data interpolation method described in [20] exploits the topological structure and unsupervised learning algorithm of a 2D self-organizing feature map (SOFM) to iteratively create a polygonal surface mesh that takes a general shape of the underlying object. [21] describes a subdivision surface fitting method based on parameter correction to achieve better error measurement. For each given data point, the closest point on the surface is found. This point is expressed as a linear function of the control mesh vertices via basis functions. This function is then defined in a least squares sense as the summation of the squared distances between the data points and the surface points. Another technique, that can be used for meshless interpolation is function-point clustering method (FPCM) [22] which defines a function having a property of being greater in regions where the density of points is higher and being minimal where the density of data points is lower. Multiresolution analysis and wavelets pro-

vide useful and efficient tools for representing functions at multiple levels of detail [23]. Multiresolution analysis [24] offers a simple, unified, and theoretically sound approach to deal with the problem of extreme complexity of meshes. The method is based on the approximation of an arbitrary initial mesh by a mesh that has subdivision connectivity and is guaranteed to be within a specified tolerance.

Our goal is to propose a new simple method for interpolation of scattered data points. In many applications, it is necessary to process and interpolate a large amount of data, thus our method has to be able to process such large datasets. There are other interpolation methods, but they are usually quite hard to implement.

Our method is to be easy to implement and it must achieve the same quality of interpolation like other methods. Furthermore, the condition of small memory requirements and low time requirements must be met as well.

2. Radial basis functions

Radial basis function (RBF) is a technique for scattered data interpolation [25] and approximation [26, 27]. The RBF interpolation and approximation is computationally more expensive compared to interpolation and approximation methods that use an information about mesh connectivity, because input data are not ordered and there is no known relation between them, i.e. tessellation is not made. Although RBF has a higher computational cost, it can be used for k -dimensional problem solution in many applications, e.g. solution of partial differential equations [28,29], image reconstruction [30], neural networks [31–33], fuzzy systems [34–36], GIS systems [37], optics [38] etc. It should be noted that it does not require any triangulation or tessellation meshing in general. There is no need to know any connectivity of interpolated points, all points are tied up only with distances of each other. Using all these distances we can form the interpolation matrix, which will be shown later.

The RBF is a function whose value depends only on the distance from its center point. Due to the use of distance functions, the RBFs can be easily implemented to reconstruct the surface using scattered data in 2D, 3D or higher dimensional spaces. It should be noted that the RBF interpolation is not separable by a dimension.

Radial function interpolants have a helpful property of being invariant under all Euclidean transformations,

i.e. translations, rotations and reflections. It does not matter whether we first compute the RBF interpolation function and then apply a Euclidean transformation, or if we first transform all the data and then compute the radial function interpolants. This is a result of the fact that Euclidean transformations are characterized by orthonormal transformation matrices and are therefore two-norm invariant. Radial basis functions can be divided into two groups according to their influence. The first group are “global” RBFs [39], for example:

$$\begin{aligned}
\text{Thin Plate Spline} \quad \varphi(r) &= r^2 \log r \\
\text{Gauss function} \quad \varphi(r) &= e^{-(\epsilon r)^2} \\
\text{Inverse Quadric} \quad \varphi(r) &= \frac{1}{1 + (\epsilon r)^2} \quad (1) \\
\text{Inverse Multiquadric} \quad \varphi(r) &= \frac{1}{\sqrt{1 + (\epsilon r)^2}} \\
\text{Multiquadric} \quad \varphi(r) &= \sqrt{1 + (\epsilon r)^2}
\end{aligned}$$

where ϵ is the shape parameter of the radial basis function [40]. Application of global RBFs usually leads to ill-conditioned system, especially in the case of large data sets with a large span [41,42].

The “local” RBFs were introduced in [43] as compactly supported RBF (CSRBF) and satisfy the following condition:

$$\begin{aligned}
\varphi(r) &= (1 - r)_+^q P(r) \\
&= \begin{cases} (1 - r)^q P(r) & 0 \leq r \leq 1 \\ 0 & r > 1 \end{cases} \quad (2)
\end{aligned}$$

where $P(r)$ is a polynomial function and q is a parameter. The subscript in $(1 - r)_+^q$ means:

$$(1 - r)_+ = \begin{cases} (1 - r) & (1 - r) \geq 0 \\ 0 & (1 - r) < 0 \end{cases} \quad (3)$$

Typical examples of CSRBF are

$$\begin{aligned}
\varphi_1(r) &= (1 - \hat{r})_+ \\
\varphi_2(r) &= (1 - \hat{r})_+^3 (3\hat{r} + 1) \\
\varphi_3(r) &= (1 - \hat{r})_+^5 (8\hat{r}^2 + 5\hat{r} + 1) \\
\varphi_4(r) &= (1 - \hat{r})_+^2 \\
\varphi_5(r) &= (1 - \hat{r})_+^4 (4\hat{r} + 1) \\
\varphi_6(r) &= (1 - \hat{r})_+^6 (35\hat{r}^2 + 18\hat{r} + 3) \quad (4)
\end{aligned}$$

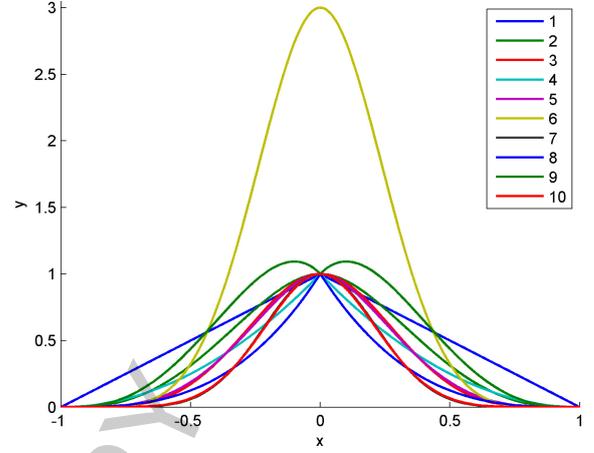


Fig. 1. Examples of CSRBF from Eq. (4).

$$\begin{aligned}
\varphi_7(r) &= (1 - \hat{r})_+^8 (32\hat{r}^3 + 25\hat{r}^2 + 8\hat{r} + 1) \\
\varphi_8(r) &= (1 - \hat{r})_+^3 \\
\varphi_9(r) &= (1 - \hat{r})_+^3 (5\hat{r} + 1) \\
\varphi_{10}(r) &= (1 - \hat{r})_+^7 (16\hat{r}^2 + 7\hat{r} + 1)
\end{aligned}$$

where $\hat{r} = \epsilon r$ and ϵ is the shape parameter of the radial basis function, see Fig. 1 for a visualization of Eq. (4).

2.1. Radial basis function interpolation

RBF interpolation was originally introduced by [44] and is based on computing of the distance of two points in any k -dimensional space. It is defined by the function

$$f(\mathbf{x}) = \sum_{j=1}^M \lambda_j \varphi(\|\mathbf{x} - \mathbf{x}_j\|) \quad (5)$$

where λ_j are weights of the RBFs, M is the number of the radial basis functions, i.e. the number of interpolation points, and φ is the radial basis function. For a given dataset of points with associated values, i.e. in the case of scalar values $\{\mathbf{x}_i, h_i\}_1^M$, the following linear system of equations is obtained:

$$\begin{aligned}
h_i = f(\mathbf{x}_i) &= \sum_{j=1}^M \lambda_j \varphi(\|\mathbf{x}_i - \mathbf{x}_j\|) \\
&\text{for } \forall i \in \{1, \dots, M\} \quad (6)
\end{aligned}$$

where λ_j are weights to be computed; see Fig. 2 for a visual interpretation of Eqs (5) or (6) for a $2\frac{1}{2}D$ function. Point in $2\frac{1}{2}D$ is a $2D$ point associated with a

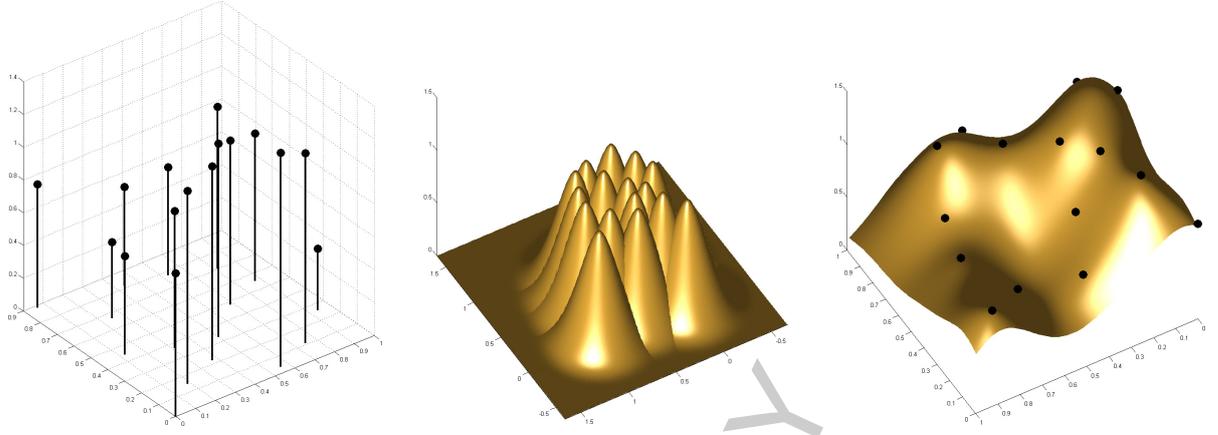


Fig. 2. Data values, the RBF collocation functions, the resulting interpolant.

scalar value. The same also applies to $3D$ point associated with a scalar value, thus $3\frac{1}{2}D$ point.

Equation (6) can be rewritten in a matrix form as

$$\mathbf{A}\boldsymbol{\lambda} = \mathbf{h}. \quad (7)$$

As $\varphi(\|\mathbf{x}_i - \mathbf{x}_j\|) = \varphi(\|\mathbf{x}_j - \mathbf{x}_i\|)$ the matrix \mathbf{A} is symmetrical.

The RBF interpolation can use “global” or “local” functions. When using “global” radial basis functions, the matrix \mathbf{A} will be full, but when using “local” radial basis functions, the matrix \mathbf{A} might be sparse, which can be beneficial when solving the system of linear equations $\mathbf{A}\boldsymbol{\lambda} = \mathbf{h}$.

In the case of the vector data, i.e. $\{\mathbf{x}_i, \mathbf{h}_i\}_1^M$ values \mathbf{h}_i are actually vectors, the RBF is to be performed for each coordinate of the vector \mathbf{h}_i .

3. Proposed approach

In this section we describe our new proposed approach for large data sets RBF interpolation. The proposed interpolation uses space subdivision to speed-up the computation and to significantly reduce high memory requirements [15,37]. The algorithm consists of three main steps. The first one is the space subdivision, the second one is the RBF interpolation and the last one is the joining procedure of interpolated cells (“blending”) to create the final interpolation. The pseudo-code of the proposed approach is in Algorithms 1 and 2. We show the speed-up of the proposed algorithm compared to the standard one for RBF interpolation as well.

3.1. Space subdivision

The approach proposed is based on a divide and co-

Algorithm 1 Pseudocode of the proposed RBF interpolation method.

-
- 1: **procedure** RBF(*Points* P) $\triangleright P_i = \{\mathbf{x}_i, h_i\}$
 - 2: **for all** cells in grid **do**
 - 3: Enlarge cell by $1/\epsilon$ \triangleright where ϵ is the shape parameter
 - 4:
 - 5: $p \leftarrow$ Points in enlarged cell
 - 6: Compute RBF interpolation of p
-

Algorithm 2 Pseudocode of interpolated value calculation using the proposed RBF interpolation method.

-
- 1: **procedure** RBF(*Point* p) $\triangleright p = \{x, y\}$
 - 2: Find neighboring cells
 - 3: Compute distances to cells
 - 4: Compute interpolated RBF values for all cells
 - 5: Blend RBF values together \triangleright using distances to cells
 - 6:
-

nquer (D&C) strategy, and therefore input data set is split into several subsets. In our case, we will use a rectangular grid of the size $n \times m$ domains for $2\frac{1}{2}D$ input data, resp. $n \times m \times l$ domains for $3\frac{1}{2}D$ input data. The grid does not have to be necessarily regular and we can adjust it according to the properties of the input data set. We will use an orthogonal regular grid of domains for simplicity of explanation of the proposed approach.

The input points need to be divided into some cells according to the created grid for the space subdivision. Every domain of the grid needs to be enlarged to a cell and contains a few more points from the neighborhood, see Fig. 3. We will present the reason for this later in this paper.

The input points need to be divided into some num-

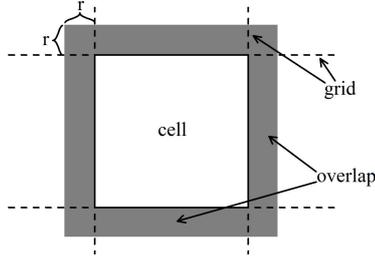


Fig. 3. 2D regular orthogonal grid with one cell visualized. Each cell contains points from the grid domain plus points from the overlapping parts with neighborhood domains.

ber of cells. This number can be estimated according to the memory available. The RBF interpolation matrix for n points in the cell has the size $n \times n$ elements, which are usually stored as double precision numbers. The size of the matrix in bytes is given as

$$\text{size} = \text{sizeof}(\text{double}) \cdot n^2 = 8n^2 \text{ (Byte)}. \quad (8)$$

Using this formula we can easily find out the maximal average number of points in cells, see Fig. 4, and set up easily the size of the grid needed for the subdivision.

Data points are generally scattered, so it might be further possible in the extreme case that nearly all points lie within one cell. In this case it would be necessary to split this cell again. Another possible case is when no point lie inside a cell. In this case, the shape parameter and grid size for RBF interpolation is inappropriately selected and must be changed in the sense that the influencing of the basis function is greater and sufficient for the data interpolation [41].

3.2. Cells RBF interpolation

Now, we have all input points divided into overlapping cells and thus can do the RBF interpolation. Radial basis functions have one parameter, which is the shape parameter ϵ . In the proposed approach, we use the “local” radial basis functions (CSRBFs), as they have the restricted maximal distance for the influence of the RBF interpolation. The shape parameter should be chosen so, that $\frac{1}{\epsilon}$ is equal to the size r of the overlapping of each domain (Fig. 3), resp. vice versa. Points on the border of a cell are exactly r away from the grid domain and RBF center points with a larger distance than r will not have any influence on the interpolated value inside a domain of the grid.

Points inside a cell need to be interpolated using the RBF interpolation with CSRBF. This interpolation is

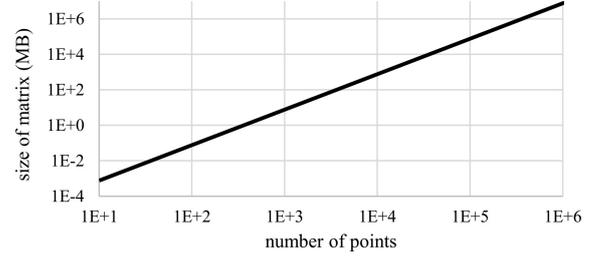


Fig. 4. The size of the RBF interpolation matrix for different number of interpolated points. The matrix is stored in double precision and its size is in MB if full matrix structure is used. The memory requirements are $O(N^2)$, where N is the number of points.

done using the standard calculation of the linear system of Eq. (6). Each cell is interpolated as an independent cell and thus the calculation can be done totally in parallel. This parallel calculation will increase the performance and speed-up the RBF computation for each cell. The only problem that can arise is the memory consumption, as we need to store multiple interpolation matrices at once, so this should be kept in mind when computing the size of a grid for space subdivision.

For each cell we get one set of weighting values of the RBF interpolation $\lambda = [\lambda_1, \lambda_2, \dots, \lambda_n]^T$. These values have to be stored for later use. The matrix used for their calculation, i.e. the RBF interpolation matrix, can be discarded.

3.3. Blending of cells and reconstruction function

The interpolated cells computed in the previous step are overlapping each other. In this section, we show how to join, i.e. blend, them together to create a final continuous interpolation function that covers all the cells and thus all the input points for the interpolation as well.

The total width of overlapping parts is $2r$. To blend all the neighborhood cells together, we will do some kind of bilinear interpolation (“blending”) between them. The computed value from each cell needs to be multiplied with a coefficient α . The coefficients α_i are computed as

$$\alpha' = \min \left(1, \frac{\text{distance from the border}}{2r} \right), \quad (9)$$

where *distance from the border* is the shortest distance from the location to the border and it is calculated using the Euclidean metric. However, for the axes-aligned grid, the distance can be calculated using

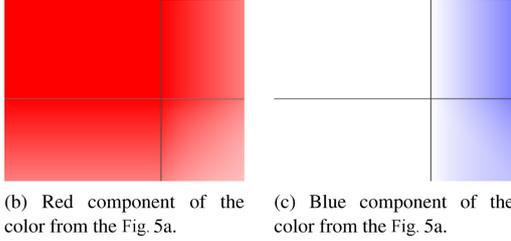
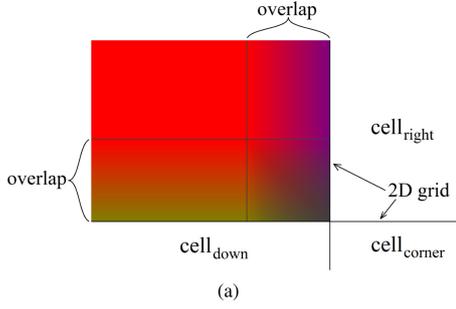


Fig. 5. Bilinear interpolation between cells for the overlapped areas. Red part of color represents the coefficient for the main cell value, green part of color represents the coefficient for the down cell value and blue part of color represents the coefficient for the right cell value. The value for the corner cell is calculated as $1 - (\text{red} + \text{green} + \text{blue})$.

Chebyshev metric, which is defined as

$$\text{distance}(P, Q) = \max_i (|p_i - q_i|), \quad (10)$$

where $P = [p_1, \dots, p_k]^T$ and $Q = [q_1, \dots, q_k]^T$ are two points in k -dimensional space.

The final coefficients α_i are computed using Eq. (9) as

$$\alpha_i = \frac{\alpha_i'}{\sum_{j=1}^{2^k} \alpha_j'}, \quad (11)$$

where k is the dimension, i.e. $k = 2$ for $2\frac{1}{2}D$ or $k = 3$ for $3\frac{1}{2}D$ input data. The visual representation of coefficients is shown in Fig. 5.

Knowing all the coefficients α_i and all function values from the RBF interpolations of cells, we can compute the final value of the proposed radial basis function interpolation algorithm for large scattered data interpolation.

$$f(\mathbf{x}) = \sum_{i=1}^{2^k} \alpha_i \left(\sum_{j=1}^{M_i} \lambda_j \varphi(\|\mathbf{x} - \mathbf{x}_j^{(i)}\|) \right), \quad (12)$$

where k is the dimension, i.e. $k = 2$ or $k = 3$, M_i is the number of points in the i -th cell, α_i is the coefficient

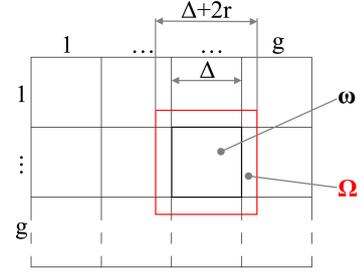


Fig. 6. Visualization of a grid.

from Eq. (11) and $\mathbf{x}_j^{(i)}$ are interpolation points in a cell.

During the blending phase we perform the interpolation between the interpolations of cells. The result of the blending phase is thus again the interpolation of all input points, as the resulting function passes through all input points.

3.4. Speed-up of the proposed approach (interpolation)

The proposed approach uses space subdivision to speed-up the calculation of radial basis function interpolation and to reduce the needed memory as well. In the following, we will use the notation shown in Fig. 6.

The value g is equal to the number of divisions in each dimension, k is the dimension, Δ is the size of one domain, r is the size of the overlap for each cell and is equal to the radius of the RBF.

The number of points n in the area ω can be estimated in the case of uniform distribution as

$$n = \frac{N}{g^k}, \quad (13)$$

where N is the total number of points for the interpolation and g is equal to the number of divisions in each dimension. Every domain ω is enlarged by the overlap r , see Fig. 3, at every side of the domain; thus the enlargement of the domain is equal to

$$\xi = \frac{\Delta + 2r}{\Delta} = 1 + \frac{2r}{\Delta}. \quad (14)$$

The average number of points in the enlarged cell Ω is equal to

$$m = \frac{N}{g^k} \xi^k. \quad (15)$$

When computing the RBF interpolation, we need to solve a system of linear equations (LSE). Let us assume that solving an LSE of size $N \times N$ has the time

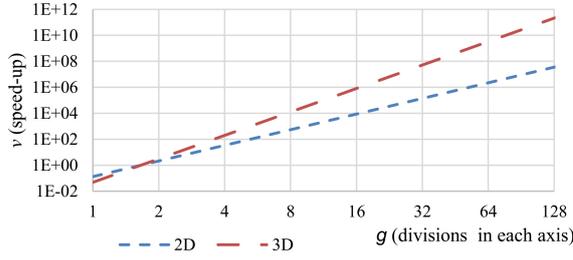


Fig. 7. Expected speed-up of the proposed algorithm according to Eq. (17) for different numbers g , i.e. resolution of the grid, for $\Delta = 1$ and the overlap $r = 0.2$.

complexity $O(N^3)$. The time complexity of our proposed interpolation for one enlarged cell ω , i.e. the cell Ω , is

$$O\left(\left(\frac{N}{g^k} \xi^k\right)^3\right). \quad (16)$$

Therefore, the expected speed-up of the proposed algorithm compared to the standard one is

$$\begin{aligned} \nu &= \frac{O(N^3)}{O\left(g^k \left(\frac{N}{g^k} \xi^k\right)^3\right)} = O\left(\frac{N^3}{g^k \left(\frac{N}{g^k} \xi^k\right)^3}\right) \\ &= O\left(\left(\frac{g^2}{\xi^3}\right)^k\right), \end{aligned} \quad (17)$$

where $\nu \gg 1$ for the most grid resolutions, as can be seen in Fig. 7, which was generated for $\Delta = 1$ and the overlap $r = 0.2$, i.e. 20% overlap at each side of every domain. It should be noted, that the axis for ν is in logarithmic scaling.

The time complexity of our proposed approach for the RBF interpolation is

$$O\left(g^k \left(\frac{N}{g^k} \xi^k\right)^3\right) = O\left(\frac{N}{n} (n\xi^k)^3\right), \quad (18)$$

where n and ξ can be constants. Then the only variable in Eq. (18) is N . Thus, the time complexity of the proposed approach is $O(N)$, but only in cases when the data points are uniformly distributed. Otherwise the worst time complexity of the proposed approach is $O(N^3)$.

3.5. Speed-up of the proposed approach (function evaluation)

The proposed approach does not speed-up only the RBF interpolation calculation, but it also speed-up the

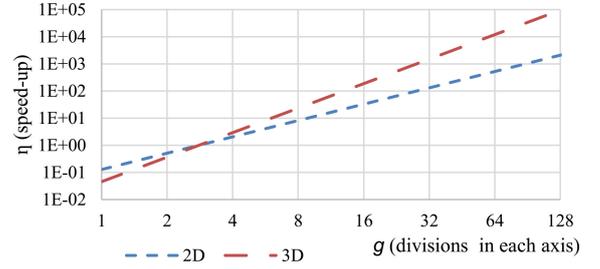


Fig. 8. Expected speed-up of function evaluation using the proposed algorithm according to Eq. (21) for different numbers g , i.e. resolution of the grid, for $\Delta = 1$ and the overlap $r = 0.2$.

evaluation of the interpolation function as well. The time complexity of the function evaluation for the standard RBF is

$$O(N). \quad (19)$$

The time complexity of the function evaluation for our proposed approach for the RBF interpolation is

$$O\left(2^k \frac{N}{g^k} \xi^k\right). \quad (20)$$

Using Eqs (19) and (20), we can compute the speed-up of our proposed algorithm when computing one function value of the RBF interpolation:

$$\begin{aligned} \eta &= \frac{O(N)}{O\left(2^k \frac{N}{g^k} \xi^k\right)} = O\left(\frac{N}{2^k \frac{N}{g^k} \xi^k}\right) \\ &= O\left(\left(\frac{g}{2\xi}\right)^k\right). \end{aligned} \quad (21)$$

For most grid resolutions the speed-up $\eta \gg 1$, as can be seen in Fig. 8, which was generated for $\Delta = 1$ and the overlap $r = 0.2$, i.e. 20% overlap at each side of every domain. It should be noted that the axis for η is in logarithmic scaling.

4. Results

In this section we show the results of our proposed approach. This approach for RBF interpolation is especially convenient for large data set interpolation. However, in the first sub-section we test it for the case of its simplicity only with small synthetically generated data sets to show some basic results of the proposed method for RBF interpolation.

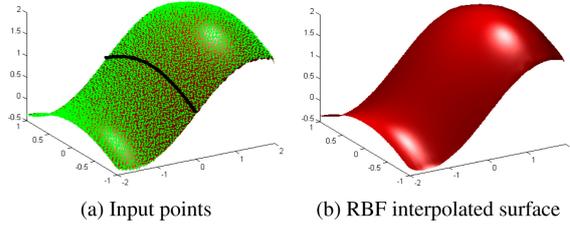


Fig. 9. 10^4 input points were used to test the proposed RBF interpolation.

In the second sub-section we tested our approach with real data sets. The second example is a data set containing more than 6×10^6 points, which is much more than the standard RBF interpolation is able to handle and compute on an ordinary computer.

Any of the CSRBFs in Eq. (4) can be used for the proposed RBF interpolation. However, in the tests we present results for one basis function, namely

$$\varphi_5(r) = (1 - er)_+^4(4er + 1). \quad (22)$$

We tested the proposed approach also with global radial basis functions, specifically with thin plate spline (TPS) and Gauss function. The results for global RBFs are very similar to those when using CSRBFs.

The implementation of the RBF interpolation was performed in MATLAB and tested on a PC with the following configuration:

- CPU: Intel® Core™ i7-920
(4×2.67 GHz + hyper-threading),
- memory: 22 GB RAM,
- operation system: Microsoft Windows 8 64 bit.

4.1. $2\frac{1}{2}D$ synthetic data

We first tested the proposed RBF interpolation on a synthetic data set of points using the function

$$f(x, y) = \sin(x) + \cos(y). \quad (23)$$

We sampled the function at 10^4 random positions with a Halton distribution (A.1 in [26]) where $x \in [-2; 2]$ and $y \in [-1; 1]$, see Fig. 9a. We used a grid of the size 2×1 and $\epsilon = 5$ and 10% of overlapping. The result of the proposed interpolation can be seen in Fig. 9b. The result is continuous.

We measured the difference of function values of the two RBF interpolations of two cells on their common border before the blending phase, see Fig. 9a. We should note that for this test, we did not blend these two

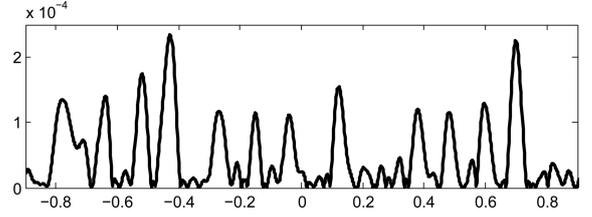


Fig. 10. Absolute difference in function values along the common border between the RBF interpolations of two cells without the blending phase, i.e. without the linear interpolation between cells.

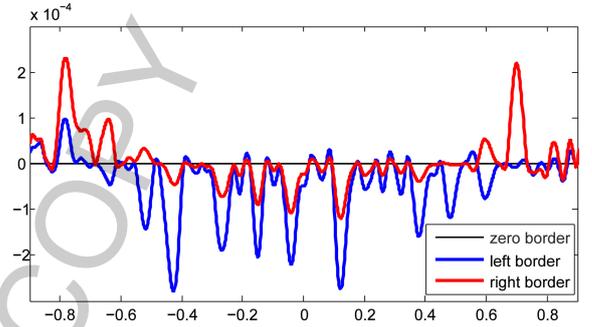


Fig. 11. Difference of function values along the common border between the interpolation of each cell without the blending phase, i.e. without the linear interpolation between cells, and the original function Eq. (23).

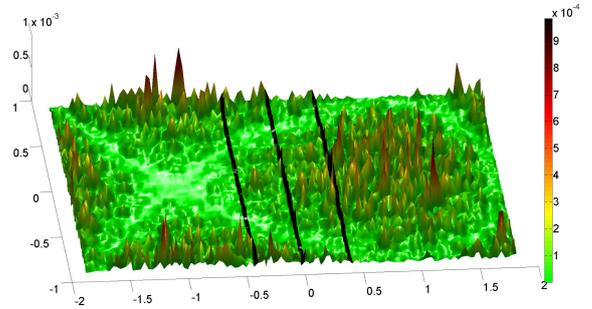


Fig. 12. Difference in function values between the proposed RBF interpolation and the original function Eq. (23).

RBF interpolations. The absolute difference between those two cells along the border is visualized in Fig. 10.

We measured the difference of function values between each cell RBF interpolation and the original function Eq. (23) on the common border before the blending phase. The difference between each cell and the original function is visualized in Fig. 11. We should note that for this test, we did not blend these two interpolations in any way.

Two cells are interpolated using RBF interpolation independently and then blend together. We measured

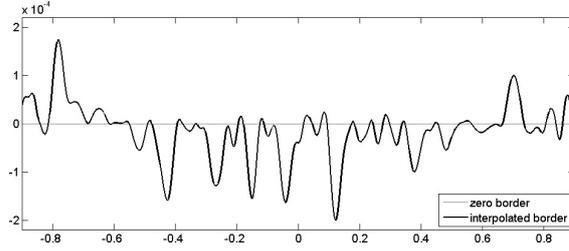


Fig. 13. Difference in function value between the proposed RBF interpolation with blending phase and the original function Eq. (23).

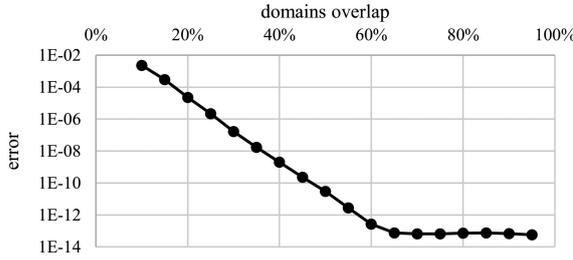


Fig. 14. Absolute difference in function values along the common border between interpolations of the two cells for different sizes of overlapping parts (for 100% error = 0).

the interpolation error between blended cells and the original function Eq. (23). The results are visualized in Fig. 12. The proposed interpolation is continuous, without any disparity between domains.

We measured the interpolation error between the proposed RBF interpolation and the original function Eq. (23) on the common border. The error is visualized in Fig. 13. It can be seen that the error has a behavior similar to that represented in Fig. 12.

The same measurement as in Fig. 10 was done for a different percentage of cells overlapping, see Fig. 14. It can be seen that the error decreases and for 100% overlapping this error is 0, as both the RBF interpolations use all points for the interpolation of their cell. It means that the proposed RBF interpolation is continuous, i.e. waterproof.

However, we need also to measure the quality of this RBF interpolation. For this purpose we compare our proposed method using the space subdivision with the standard RBF interpolation method (2.2 in [26]) using 2×10^4 randomly sampled points with the uniform distribution of the function [26]:

$$\begin{aligned}
 f(x, y) = & 3(1-x)^2 e^{-(x^2-(y+1)^2)} \\
 & - 10 \left(\frac{x}{5} - x^3 - y^5 \right) e^{-(x^2-y^2)} \\
 & - \frac{1}{3} e^{-(x+1)^2-y^2}, \quad (24)
 \end{aligned}$$

Table 1
Average interpolation error of the proposed approach and the standard RBF interpolation. The interpolation error difference between both measured methods is only 0.03%

	Proposed approach	Standard RBF interpolation
Mean absolute error	$3.1371 \cdot 10^{-4}$	$3.1362 \cdot 10^{-4}$

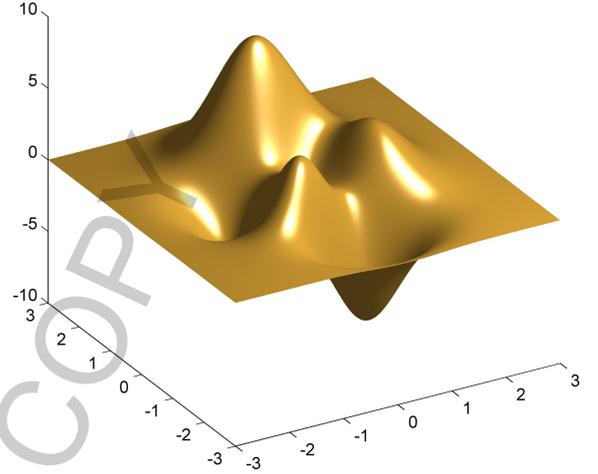


Fig. 15. The result of RBF interpolation using the proposed method with space subdivision.

where $x \in [-3; 3]$ and $y \in [-3; 3]$.

We used a grid of size 4×4 and the shape parameter with the size 20% of the domain edge length. The result of this interpolation is presented in Fig. 15. The standard RBF interpolation used the same points, the same basis function and the same shape parameter for interpolation.

To evaluate the quality of the interpolation we generated 1.5×10^5 randomly sampled points with Halton distribution where $x \in [-3; 3]$ and $y \in [-3; 3]$. Then we computed function values of both the interpolations and evaluate the absolute error of each interpolation. For each point $P_i = [x_i, y_i]^T$ we compute absolute error

$$Err_i = \|RBF(P_i) - f(x_i, y_i)\|_2, \quad (25)$$

where $RBF(P_i)$ is the interpolated value at point P_i using standard RBF interpolation on the whole dataset and RBF interpolation of the proposed approach, $f(x_i, y_i)$ is the function value of Eq. (24). The Fig. 16 presents distribution histograms of the interpolation errors.

As both histograms in Fig. 16 are visually identical, we created a difference histogram between the two his-

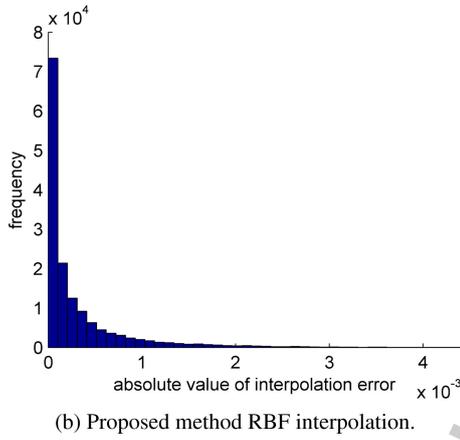
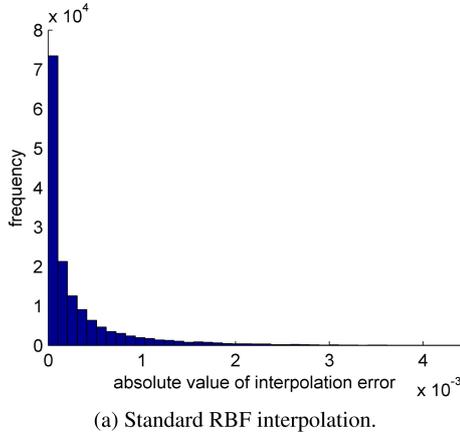


Fig. 16. Histograms of interpolation errors.

tograms. In Fig. 17 it can be seen that the interpolation errors distribution is almost identical. The difference in both histograms differs only slightly, see Fig. 17. Thus both interpolations have almost the same quality.

We also computed the average interpolation error for each RBF interpolation. The result is in Table 1. We can see that both average interpolation errors are almost the same, there is only a difference of 0.03%. Knowing all results from quality measurements we can say that our proposed RBF interpolation has almost identical quality as the standard RBF interpolation.

4.2. Real data set

The proposed approach is mainly suited for large data interpolation. For this reason we chose to use a real data set. The LiDAR data of Mount Saint Helens¹

¹<http://www.liblas.org/samples/>.

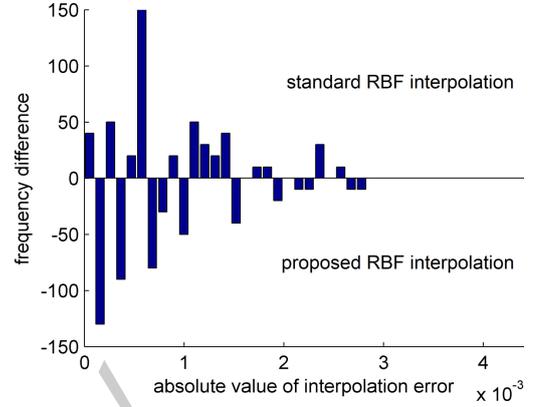


Fig. 17. Difference of histograms in Fig. 16. Positive values mean that the standard RBF interpolation has more errors with the specific absolute value of interpolation error and the negative values mean the same for the proposed method for RBF interpolation.

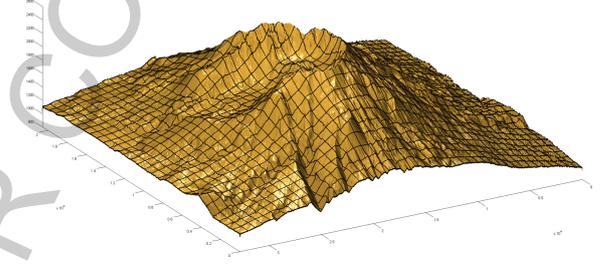


Fig. 18. Visualization of the interpolated terrain produced only as a visualization of each domain separately. The orthogonal grid used for the space subdivision with resolution of 29×46 is visualized on the terrain as well.

in Skamania County, Washington, contains scanned height data. The data set consists of 6,743,176 $2D$ points with associated heights, i.e. $2\frac{1}{2}D$ data.

We chose to divide the input data set into a regular grid in a way such that the inside of a domain is going to be on average 5,000 points. To make square domains, we created a grid of the size 29×46 , as the data range is around $2.1 \cdot 10^4 \text{ ft} \times 3.3 \cdot 10^4 \text{ ft}$, i.e. $6.4 \cdot 10^3 \text{ m} \times 1.0 \cdot 10^4 \text{ m}$, in x and y coordinates. The visualization of the created grid domains is in Fig. 18.

To perform the RBF interpolation, we needed to choose the shape parameter ϵ of the CSRBF. We tested different values of the shape parameter and selected the best shape parameter which has the size of 20% of the domain edge length. Each cell will therefore contain approximately

$$5,000 \times (1 + 2 \cdot 0.2)^2 = 9,800 \quad (26)$$

points. The number of points inside the cell is almost double times more than number of points inside the do-

Table 2

Parallel speed-up of the proposed method compared to the serial version of this method

Number of threads	1	2	4
Speed-up	1	1.791	3.172

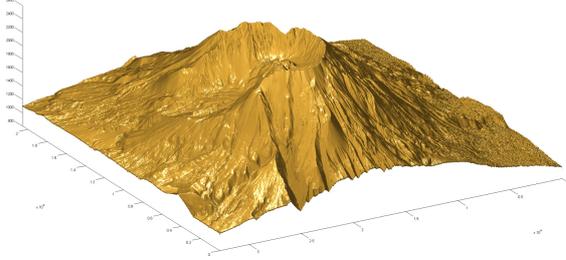


Fig. 19. Visualization of the final result of the proposed method for large scattered data interpolation with the RBF and space subdivision.

main, but the final speed-up will still be very high. For clarity we can estimate the speed-up of the proposed algorithm compared to the standard one as:

$$speed-up = \frac{6,743,176^3}{29 \times 46 \times 9,800^3} \approx 2.4 \times 10^5. \quad (27)$$

It can be seen that the speed-up is significant and we save a lot of calculations. The expected speed-up of the function evaluation is

$$speed-up = \frac{6,743,176}{2^2 \times 9,800} \approx 172. \quad (28)$$

This means that each RBF function computation for a given \mathbf{x} is approximately 172 times faster.

Moreover, the standard algorithm for RBF interpolation would require around 330 TB to save the full interpolation matrix to the memory when double precision is used.

The data set divided into cells was interpolated one cell after another. We used one RBF interpolated cell to reconstruct the terrain inside one domain of the grid without blending step. The result can be seen together with the grid of domains in Fig. 18.

Figure 19 presents the result of the proposed RBF interpolation method. We used Eq. (12) to compute interpolation of the height values of the terrain for the visualization. This terrain does not have any discontinuity because of the proposed blending procedure.

The proposed algorithm can be easily parallelized as the RBF interpolation of each cell of the grid can be done separately and thus in parallel. We measured the running time of the interpolation when using 1 or 2 or

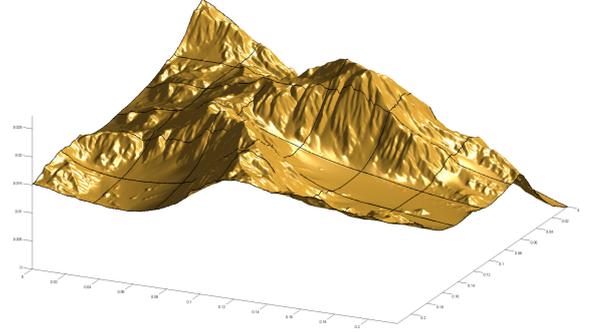


Fig. 20. Visualization of the interpolated terrain produced only as a visualization of each domain separately. The orthogonal grid used for the space subdivision with resolution of 6×6 is visualized on the terrain as well.

4 threads. The resulting speed-up in MATLAB is in the Table 2. It can be seen that the speed-up is high because the threads do not have to wait for any synchronization and are independent of each other.

We tested our proposed approach with another data set too. We chose a model of the terrain² which contains 131,044 points with associated heights, i.e. $2\frac{1}{2}D$ data.

We divided the input data set to a regular grid so that a domain contains 3,000 points in average. We created a grid of the size 6×6 , with the data range is around $0.2172 \text{ miles} \times 0.2172 \text{ miles}$ in x and y coordinates. The visualization of the created grid of domains is in Fig. 20.

For the shape parameter, we used the size of 20% of the domain edge length. Therefore, each cell contains around

$$3,000 \times (1 + 2 \cdot 0.2)^2 = 5,880 \quad (29)$$

points. It is almost double times more, but the final speed-up will still be very high. We can estimate the speed-up of the proposed algorithm compared to the standard one:

$$speed-up = \frac{131,044^3}{6 \times 6 \times 5,880^3} \approx 3 \times 10^2. \quad (30)$$

It can be seen that the speed-up is significant and will save us a lot of calculations. The speed-up of interpolating function evaluation is

$$speed-up = \frac{131,044}{2^2 \times 5,880} \approx 5.6. \quad (31)$$

²<http://www.badking.com.au/site/shop/environment/mountain-terrain/>.

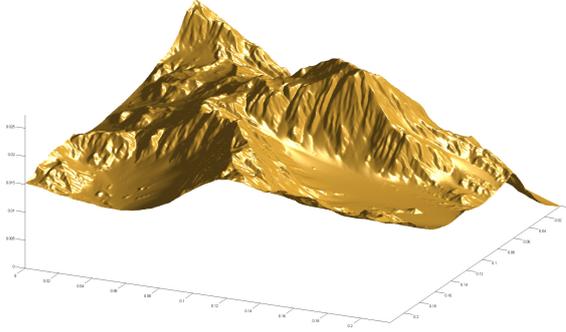


Fig. 21. Visualization of the final result of the proposed method for large scattered data interpolation with the RBF and space subdivision.

Moreover, the standard algorithm for the RBF interpolation needs around 128 GB to save the full interpolation matrix to the memory when double precision is used.

The data set divided into cells was interpolated one cell after another. We did a visualization of this RBF interpolation without doing any blending procedure. We used one RBF interpolated cell to reconstruct the terrain inside each domain of the grid. The result can be seen in Fig. 20, together with a visualization of the grid.

Figure 21 presents the result of the proposed RBF interpolation method. We used Eq. (12) to compute the height values of the terrain for the visualization. This terrain is continuous and does not have any discontinuity because of the proposed blending procedure.

If CSRBF is used, many elements in the interpolation matrix are equal to zero, as the matrix is sparse in general. To decrease the memory requirements and be able to solve large interpolation matrices we can use a sparse matrix data structure. There are several existing sparse matrix representations. e.g. [37,45,46]. The main difference among existing storage formats is the sparsity pattern, or the structure of nonzero elements, for they are best suited. In our implementation, the coordinate format is used, which is briefly described in the following.

The coordinate (COO) format [47] is the simplest storage scheme. The sparse matrix is represented by three arrays: *data*, where the nonzero values are stored, *row*, where the row index of each nonzero element is kept, and *col*, where the column indices of the nonzero values are stored. The benefit of this format is its generality, i.e. an arbitrary sparse matrix can be represented by the COO format and the required storage is always proportional to the number of nonzero values. The disadvantage of the COO format is that both row and col-

Table 3

Speed-up of the proposed approach for large scattered data interpolation compared to the standard RBF interpolation. Both methods are using sparse matrix with COO format and kd-tree structure

Grid resolution	4×4	6×6	8×8	10×10	12×12
Speed-up	1.69	1.83	2.06	2.28	2.57

Table 4

Memory requirements for our proposed method and for the standard RBF interpolation method. The proposed method was tested with full matrix data structure and also using sparse matrix with COO format together with kd-tree structure. The standard algorithm for RBF interpolation uses sparse matrix with COO format together with kd-tree structure

Grid size	Proposed method		Standard method
	kd-tree and sparse matrix	Full matrix	kd-tree and sparse matrix
4×4	590 MB	6,900 MB	
6×6	290 MB	2,300 MB	
8×8	180 MB	1,000 MB	8,800 MB
10×10	125 MB	500 MB	
12×12	95 MB	400 MB	

umn indices are stored explicitly, which reduces the efficiency of memory transactions (e.g. read operations).

Moreover, note that the elements in the interpolation matrix are zero for far away points, when CSRBFs are used. Therefore, we do not need to compute the elements for all pairs of points, so the kd-tree (A.2 in [26]) is used for computing the interpolation matrix.

As the proposed approach also needs to be compared with the standard one for RBF interpolation; we used the dataset in Fig. 21 which contains 131,044 points for interpolation. For the shape parameter for RBF interpolation we used the size of $1/30$ of the data range. We measured the running times of our algorithm running in sequential version for different grid resolutions and computed the speed-up compared to the standard algorithm for RBF interpolation, see Table 3. Both methods are using sparse matrix with COO format and kd-tree structure. We also measured the memory requirements and the results are in Table 4.

According to the results in Table 3, the proposed algorithm is faster than the standard one and the speed-up is increasing with increasing of the grid resolution; both methods used the COO sparse matrix structure. We could not compute the speed-up when using the full matrix data structure as we were unable to fit such large data into the available memory for the standard algorithm.

According to the results in Table 4, the proposed approach has much lower memory requirements than the standard one. Therefore our approach enables to compute the RBF interpolation for very large datasets even on computers with standard memory size.

5. Conclusion

We presented a new approach for radial basis function interpolation of scattered data. It computes the interpolation on partly overlapping cells and then blends these interpolations together to create the final interpolation of the whole data set. This approach is especially efficient for large scattered data interpolation, as it reduces the memory required. It significantly speeds up computation of the interpolated value. The proposed approach is suitable for parallelization and it was tested on synthetic and large real data sets. It proved its robustness and high performance.

In future the proposed approach will be used for vector fields interpolation of large data sets based on [48,49] and considering also vector field characteristics. We plan to modify the proposed method for 3D scattered data interpolation and approximation. In the case of 3D, point data will be split into overlapping cubes according to the grid. The joining phase, i.e. blending, will be almost the same as when blending 2D cells.

Acknowledgments

The authors would like to thank their colleagues at the University of West Bohemia, Plzen, for their discussions and suggestions, especially to Zuzana Majdisova, and anonymous reviewers for their valuable comments and hints provided. The research was supported by projects Czech Science Foundation (GACR) No. 17-05534S and SGS 2016-013.

References

- [1] Davis PJ. Interpolation and approximation. Courier Corporation 1975.
- [2] O'Rourke J, Mallinckrodt AJ, et al. Computational geometry in C. Computers in Physics 1995; 9(1): 55-55.
- [3] Yang J, Wang Z, Zhu C, Peng Q. Implicit surface reconstruction with radial basis functions. In: International Conference on Computer Vision and Computer Graphics, Springer 2007; 5-12.
- [4] Ohtake Y, Belyaev A, Alexa M, Turk G, Seidel HP. Multi-level partition of unity implicits. In: ACM Siggraph 2005 Courses, ACM 2005; 463-470.
- [5] Ohtake Y, Belyaev AG, Seidel H. 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) 2003; 153-164, 292.
- [6] Yokota R, Barba LA, Knepley MG. PetRBF-A parallel O(N) algorithm for radial basis function interpolation with Gaussians. Computer Methods in Applied Mechanics and Engineering 2010; 199(25): 1793-1804.
- [7] Saad Y, Schultz MH. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM Journal on Scientific and Statistical Computing 1986; 7(3): 856-869.
- [8] Cai XC, Sarkis M. A restricted additive Schwarz preconditioner for general sparse linear systems. SIAM Journal on Scientific Computing 1999; 21(2): 792-797.
- [9] Cuomo S, Galletti A, Giunta G, Starace A. Surface reconstruction from scattered point via RBF interpolation on GPU. In: Computer Science and Information Systems (FedCSIS), 2013 Federated Conference on IEEE 2013; 433-440.
- [10] Süßmuth J, Meyer Q, Greiner G. Surface reconstruction based on hierarchical floating radial basis functions. Computer Graphics Forum 2010; 29(6): 1854-1864.
- [11] Haase G, Martin D, Offner G. Towards RBF Interpolation on Heterogeneous HPC Systems. In: Large-Scale Scientific Computing – 10th International Conference, LSSC 2015; 2015: 182-190.
- [12] Beatson RK, Light WA, Billings SD. Fast Solution of the Radial Basis Function Interpolation Equations: Domain Decomposition Methods. SIAM J Scientific Computing 2001; 22(5): 1717-1740.
- [13] Farrell P, Pestana J. Block preconditioners for linear systems arising from multiscale collocation with compactly supported RBFs. Numerical Lin Alg with Applic 2015; 22(4): 731-747.
- [14] Ling L, Kansa EJ. Preconditioning for radial basis functions with domain decomposition methods. Mathematical and Computer Modelling 2004; 40(13): 1413-1427.
- [15] Majdisova Z, Skala V. A radial basis function approximation for large datasets. In: Proceedings of SIGRAD 2016; (127): 9-14.
- [16] Smolik M, Skala V, Nedved O. A Comparative Study of LOWESS and RBF Approximations for Visualization. In: Computational Science and Its Applications – ICCSA 2016 – 16th International Conference, Part II 2016; 405-419.
- [17] Mallet JL. Discrete smooth interpolation. ACM Transactions on Graphics (TOG) 1989; 8(2): 121-144.
- [18] Bui TQ, Nguyen TN, Nguyen-Dang H. A moving Kriging interpolation-based meshless method for numerical simulation of Kirchhoff plate problems. International Journal for Numerical Methods in Engineering 2009; 77(10): 1371-1395.
- [19] Ma YZ, Royer JJ, Wang H, Wang Y, Zhang T. Factorial kriging for multiscale modelling. Journal of the Southern African Institute of Mining and Metallurgy 2014; 114(8): 651-659.
- [20] Knopf GK, Sangole A. Interpolating scattered data using 2D self-organizing feature maps. Graphical Models 2004; 66(1): 50-69.
- [21] Marinov M, Kobbelt L. Optimization methods for scattered data approximation with subdivision surfaces. Graphical Models 2005; 67(5): 452-473.
- [22] Katz JO, Rohlf FJ. Function-point cluster analysis. Systematic Biology 1973; 22(3): 295-301.
- [23] Lounsbery M, DeRose TD, Warren J. Multiresolution analysis for surfaces of arbitrary topological type. ACM Transactions on Graphics (TOG) 1997; 16(1): 34-73.
- [24] Eck M, DeRose T, Duchamp T, Hoppe H, Lounsbery M, Stuetzle W. Multiresolution analysis of arbitrary meshes. In: Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques ACM 1995; 173-182.
- [25] Pan R, Skala V. A two-level approach to implicit surface modeling with compactly supported radial basis functions. Engineering with Computers 2011; 27(3): 299-307.
- [26] Fasshauer GE. Meshfree approximation methods with MATLAB. vol. 6. World Scientific; 2007.

- [27] Skala V. Meshless interpolations for computer graphics, visualization and games. In: Eurographics 2015 – Tutorials 2015.
- [28] Larsson E, Fornberg B. A numerical study of some radial basis function based solution methods for elliptic PDEs. *Computers & Mathematics with Applications* 2003; 46(5): 891-902.
- [29] Zhang X, Song KZ, Lu MW, Liu X. Meshless methods based on collocation with radial basis functions. *Computational Mechanics* 2000; 26(4): 333-343.
- [30] Uhlir K, Skala V. Reconstruction of damaged images using radial basis functions. In: Signal Processing Conference, 2005 13th European, IEEE 2005; 1-4.
- [31] Karim A, Adeli H. Radial basis function neural network for work zone capacity and queue estimation. *Journal of Transportation Engineering* 2003; 129(5): 494-503.
- [32] Ghosh-Dastidar S, Adeli H, Dadmehr N. Principal component analysis-enhanced cosine radial basis function neural network for robust epilepsy and seizure detection. *IEEE Transactions on Biomedical Engineering* 2008; 55(2): 512-518.
- [33] Yingwei L, Sundararajan N, Saratchandran P. Performance evaluation of a sequential minimal radial basis function (RBF) neural network learning algorithm. *IEEE Transactions on Neural Networks* 1998; 9(2): 308-318.
- [34] Adeli H, Karim A. Fuzzy-wavelet RBFNN model for freeway incident detection. *Journal of Transportation Engineering* 2000; 126(6): 464-471.
- [35] Karim A, Adeli H. Comparison of fuzzy-wavelet radial basis function neural network freeway incident detection model with California algorithm. *Journal of Transportation Engineering* 2002; 128(1): 21-30.
- [36] Hsu CF, Lin CM, Yeh RG. Supervisory adaptive dynamic RBF-based neural-fuzzy control system design for unknown nonlinear systems. *Applied Soft Computing* 2013; 13(4): 1620-1626.
- [37] Majdisova Z, Skala V. Big geo data surface approximation using radial basis functions: A comparative study. *Computers & Geosciences* 2017; 109: 51-58.
- [38] Prakash G, Kulkarni M, Sripathi U. Using RBF Neural Networks and Kullback-Leibler distance to classify channel models in Free Space Optics. In: Optical Engineering (ICOE), 2012 International Conference on IEEE 2012; 1-6.
- [39] Schagen IP. Interpolation in two dimensions – a new technique. *IMA Journal of Applied Mathematics* 1979; 23(1): 53-59.
- [40] Fornberg B, Piret C. On choosing a radial basis function and a shape parameter when solving a convective PDE on a sphere. *J Comput Physics* 2008; 227(5): 2758-2780.
- [41] Majdisova Z, Skala V. Radial basis function approximations: Comparison and applications. *Applied Mathematical Modelling* 2017; 51: 728-743.
- [42] Skala V. RBF interpolation with CSRBF of large data sets. *Procedia Computer Science* 2017; 108: 2433-2437.
- [43] Wendland H. Computational aspects of radial basis function approximation. *Studies in Computational Mathematics* 2006; 12: 231-256.
- [44] Hardy RL. Multiquadric equations of topography and other irregular surfaces. *Journal of Geophysical Research* 1971; 76(8): 1905-1915.
- [45] Bell N, Garland M. Implementing sparse matrix-vector multiplication on throughput-oriented processors. In: Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis, ACM 2009; 18: 1-11.
- [46] Simecek I. Sparse matrix computations using the quadtree storage format. In: Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), 2009 11th International Symposium on IEEE 2009; 168-173.
- [47] Gilbert JR, Moler C, Schreiber R. Sparse matrices in MATLAB: Design and implementation. *SIAM Journal on Matrix Analysis and Applications* 1992; 13(1): 333-356.
- [48] Smolik M, Skala V. Vector Field Interpolation with Radial Basis Functions. In: Proceedings of SIGRAD 2016, Linköping University Electronic Press 2016; (127): 15-21.
- [49] Smolik M, Skala V. Classification of Critical Points Using a Second Order Derivative. *Procedia Computer Science* 2017; 108: 2373-2377.

4.4 Efficient simple large scattered 3D vector fields Radial basis functions approximation using space subdivision

Vector field data sets can be very large and the standard techniques for approximation may not be able to process such large data sets. The paper [Smolik and Skala, 2019] focuses on approximation of vector fields using Radial Basis Functions. Standard RBF technique is able to approximate a few millions of vectors, which is for example a vector field on a grid of Full HD resolution. However, the 3D vector fields are mostly much larger than a few millions of vectors.

This paper presents a new approach for approximation of large scattered 3D vector fields. This approach is an extension of [Smolik and Skala, 2018] and it uses the space subdivision technique to reduce the memory requirements and the computational complexity, too. The 3D vector field is divided into several overlapping 3D cells. Each cell is approximated using RBF, where the radial basis functions are placed according to the properties of the vector field. All the RBF approximations are then blended together to obtain a final approximated vector field.

The proposed approach was tested on a large 3D vector field representing an EF5 tornado with around $5.5 \cdot 10^8$ vectors. The resulting approximated vector field proved low approximation error, while maintaining high compression ratio. Moreover, the proposed approach is capable to process even much larger data sets than the testing data set.

Citation:

- Michal Smolik and Vaclav Skala. Efficient simple large scattered 3D vector fields radial basis functions approximation using space subdivision. In *Computational Science and Its Applications – ICCSA 2019*, pages 337–350. Springer, 2019.



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 tetrahedralization, 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.

© Springer Nature Switzerland AG 2019
S. Misra et al. (Eds.): ICCSA 2019, LNCS 11619, pp. 337–350, 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 field 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 [23,24] 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 [13,14]. 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.

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 Algorithms 1 and 2 present relevant pseudocodes.

Algorithm 1. Pseudocode of the proposed approach for RBF approximation.

```

1: procedure RBF(Points P) ▷  $P_i = \{\mathbf{x}_i, \mathbf{v}_i\}$ 
2:   for all cells in grid do
3:     Enlarge cell for approximation by  $\Psi$ 
4:      $p \leftarrow$  Points in enlarged cell
5:      $\xi \leftarrow$  RBF centers in enlarged cell
6:     Compute RBF approximation of  $p$ 

```

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

```

1: procedure RBF(Point p) ▷  $p = \{x, y, z\}$ 
2:   Find neighboring cells
3:   Determine distances to cells
4:   Compute approximated RBF values for all cells
5:   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 using 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.

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

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

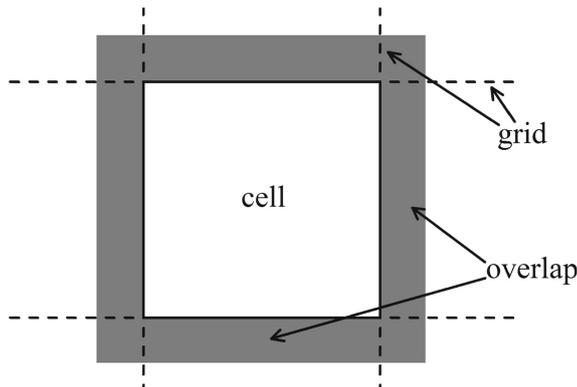


Fig. 1. 3D regular orthogonal grid (2D analogy) of one cell. Each cell has points that are inside the domain plus points from the overlapping parts (gray color).

Now, the given points are split 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.

$$\mathbf{v}_i = \mathbf{v}(\mathbf{x}_i) = \sum_{j=1}^M \lambda_j \varphi(\|\mathbf{x}_i - \boldsymbol{\xi}_j\|),$$

$$\text{for } \forall i \in \{1, \dots, N\} \quad (2)$$

where $\mathbf{v}_i = [v_i^{(x)}, v_i^{(y)}, v_i^{(z)}]$, M is the number of the RBF centers. Solution of the linear system of equations is a vector $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \dots, \lambda_M]^T$, where $\lambda_i = [\lambda_i^x, \lambda_i^y, \lambda_i^z]^T$. These values will be used later. However, the matrix for the RBF computation can be discarded as it will not be needed any more.

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. ($2\psi = \Psi$).

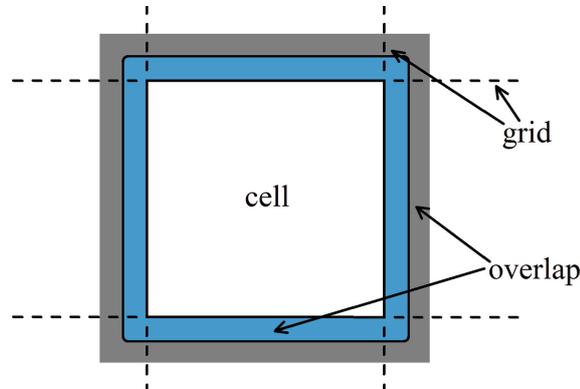


Fig. 2. Visualization of the overlap part used for blending (blue color). (Color figure online)

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

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

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

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

where $i = \{1, \dots, 2^k\}$ 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. 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 \mathbf{x} .

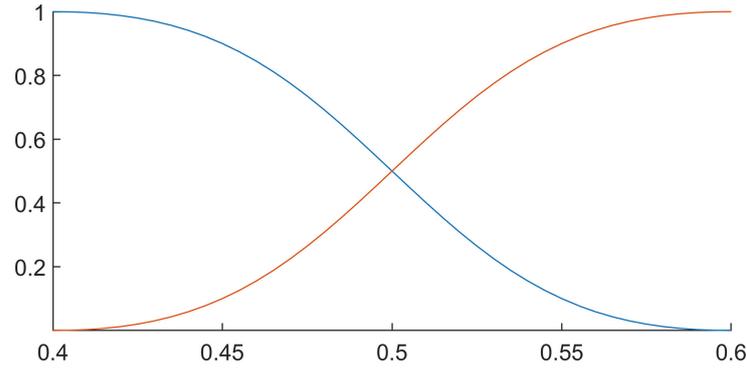


Fig. 3. Blending functions for blending of two approximations.

The space subdivision is used to speed-up the computation of vector field radial basis function approximation, i.e. computation of λ 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

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

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

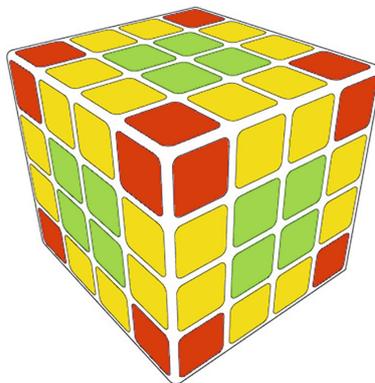


Fig. 4. Visualization of different type of cells according to the number of points inside the enlarged cell.

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 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

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

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.

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

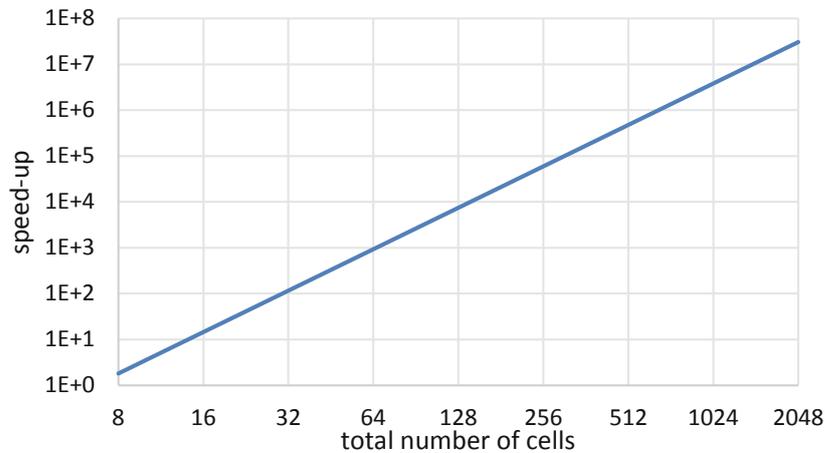


Fig. 5. Expected speed-up of the proposed approach of vector field RBF approximation compared to the standard one (note that the axes are logarithmic).

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

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

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:

$$O(M). \quad (12)$$

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

$$O(2^3 m), \quad (13)$$

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:

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

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.

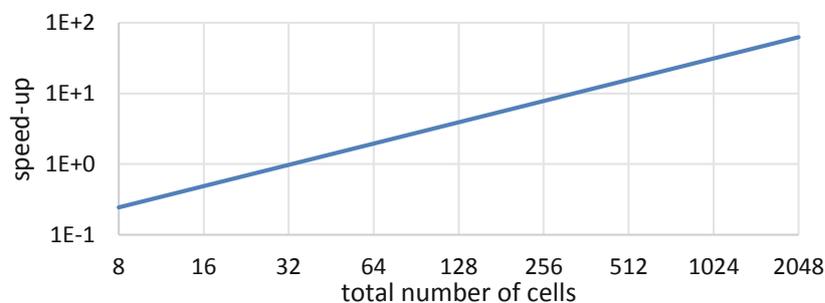


Fig. 6. Expected speed-up of function evaluation of the proposed approach for vector field RBF approximation compared to the standard one (note that the axes are logarithmic).

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.

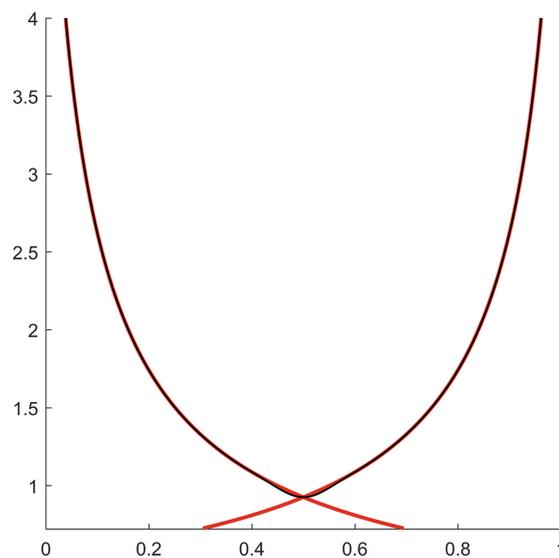


Fig. 7. Blending of two functions (red) and the result after blending (black). (Color figure online)

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.

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.

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

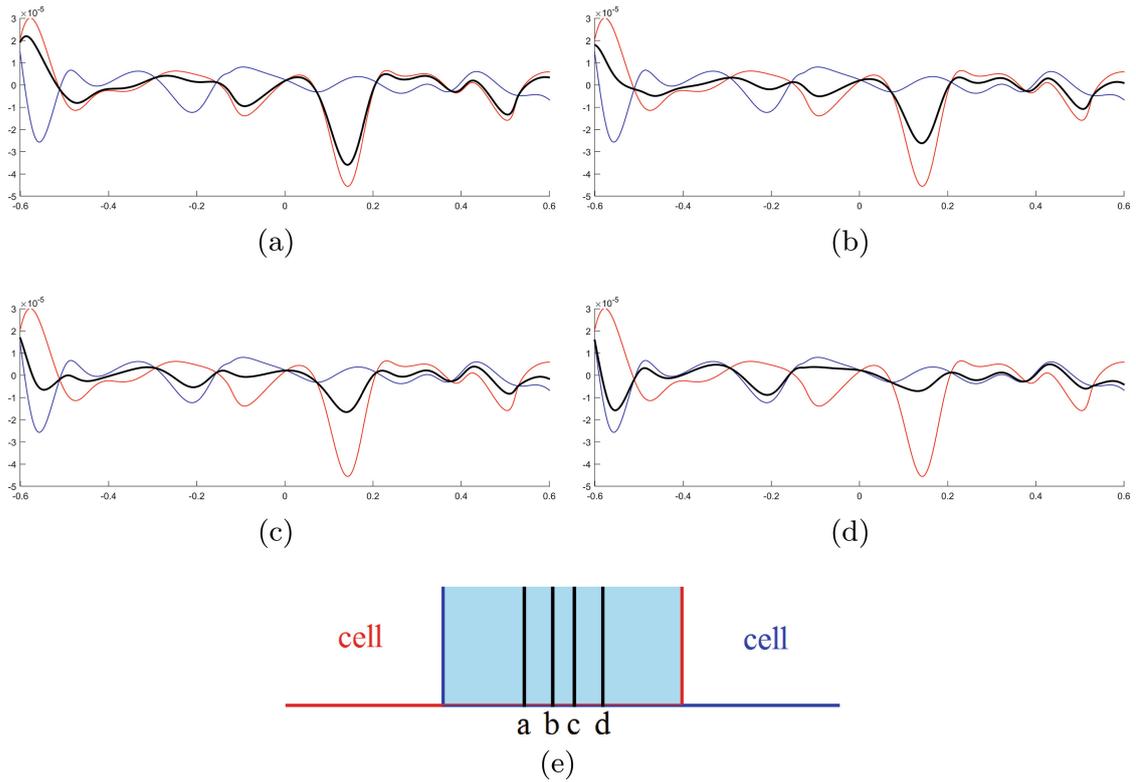


Fig. 8. Blending of two $2\frac{1}{2}D$ functions together. Visualization of blending for different “cut” of $2\frac{1}{2}D$ function (a–d), see (e) for “cut” location.

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 though 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.

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

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

where $\bar{\mathbf{v}}_i$ is the original vector, \mathbf{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.

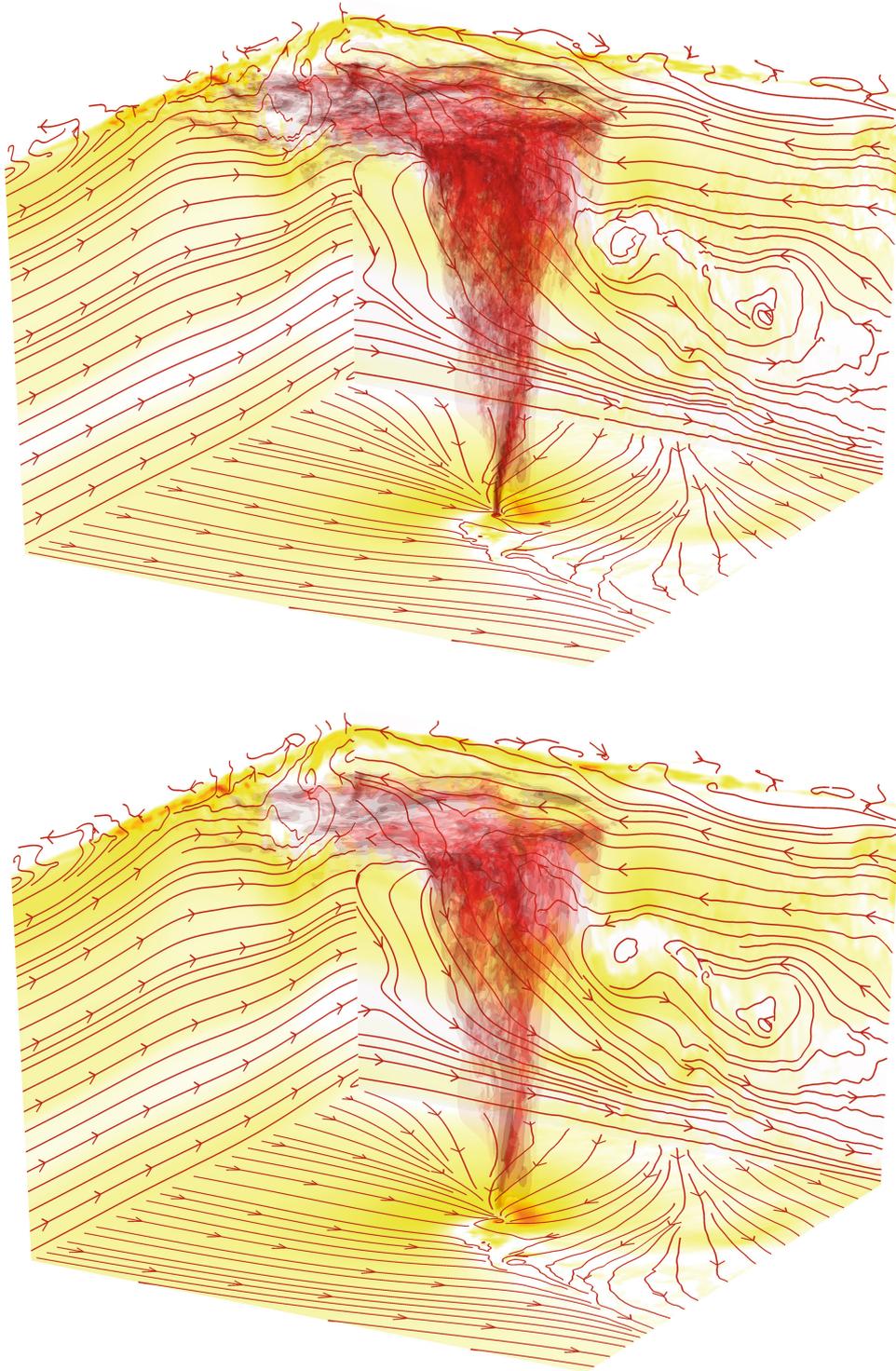


Fig. 9. Visualization of the 3D tornado vector field data set. Red central part represents the shape of tornado vortex and the yellow color on faces represents the speed of vector field. The original vector field (top) and the RBF approximated vector field (bottom). (Color figure online)

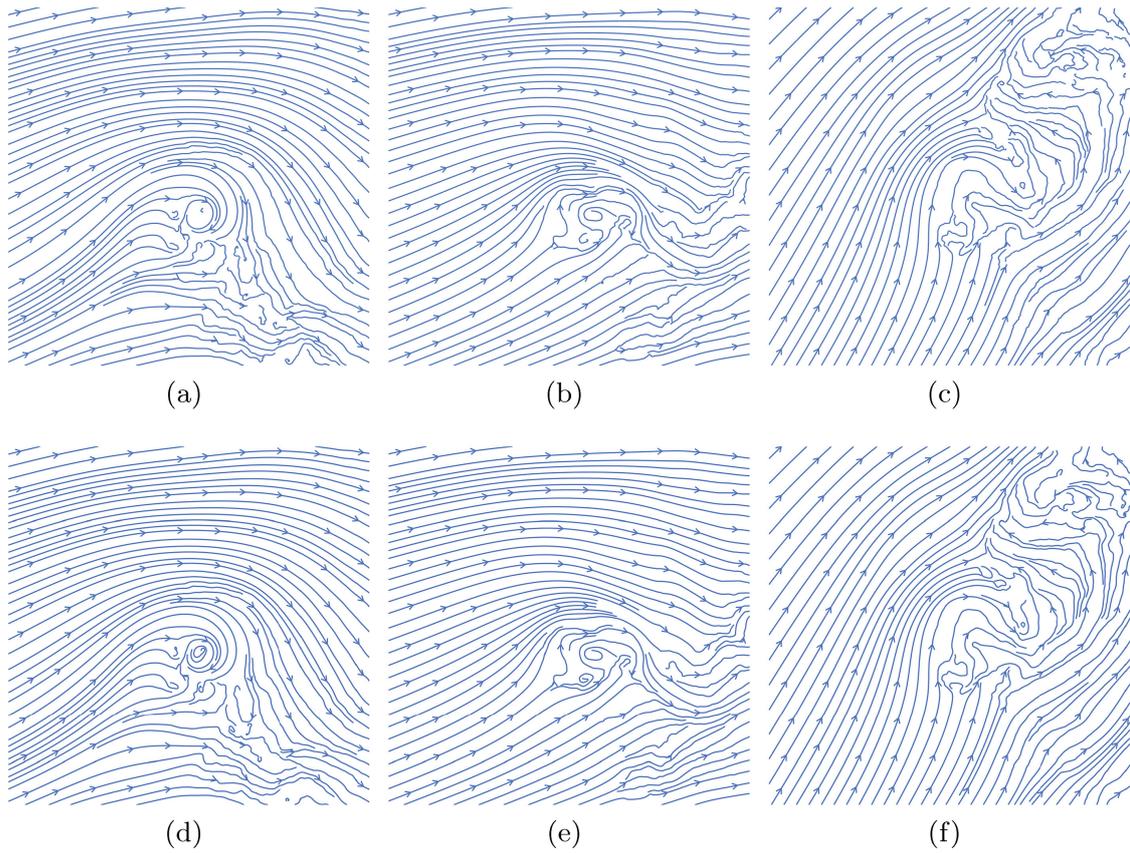


Fig. 10. Visualization of three 2D vector field slices. The top row represents the original vector field (a–c) and the bottom row represents the approximated vector field (d–f).

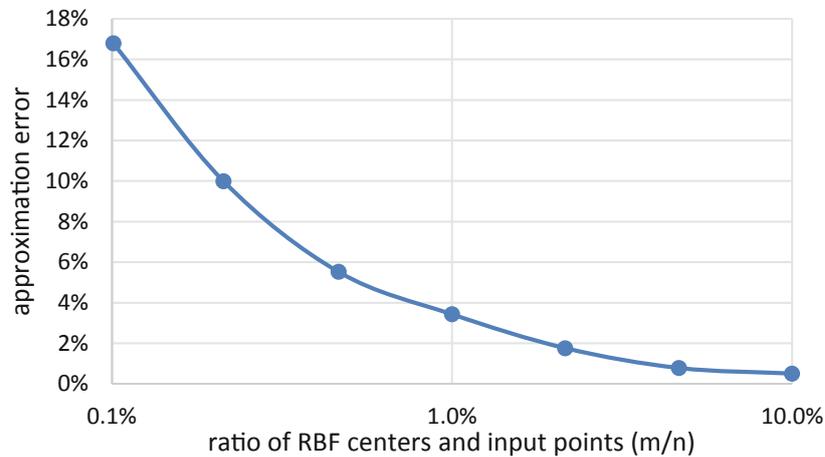


Fig. 11. The approximation error for different number of RBF centers.

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. λ values, and the final RBF evaluation as well.

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. Beatson, R.K., Light, W.A., Billings, S.D.: Fast solution of the radial basis function interpolation equations: domain decomposition methods. *SIAM J. Sci. Comput.* **22**(5), 1717–1740 (2001)
2. Cabrera, D.A.C., Gonzalez-Casanova, P., Gout, C., Juárez, L.H., Reséndiz, L.R.: Vector field approximation using radial basis functions. *J. Comput. Appl. Math.* **240**, 163–173 (2013)
3. Cai, X.-C., Sarkis, M.: A restricted additive Schwarz preconditioner for general sparse linear systems. *SIAM J. Sci. Comput.* **21**(2), 792–797 (1999)
4. Dey, T.K., Levine, J.A., Wenger, R.: A Delaunay simplification algorithm for vector fields. In: 15th Pacific Conference on Computer Graphics and Applications, PG 2007, pp. 281–290. IEEE (2007)
5. Duchon, J.: Splines minimizing rotation-invariant semi-norms in Sobolev spaces. In: Schempp, W., Zeller, K. (eds.) *Constructive Theory of Functions of Several Variables*, pp. 85–100. Springer, Heidelberg (1977). <https://doi.org/10.1007/BFb0086566>
6. Golub, G.H., Van Loan, C.F.: *Matrix computations*, vol. 3. JHU Press (2012)
7. Haase, G., Martin, D., Offner, G.: Towards RBF interpolation on heterogeneous HPC systems. In: Lirkov, I., Margenov, S.D., Waśniewski, J. (eds.) *LSSC 2015*. LNCS, vol. 9374, pp. 182–190. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-26520-9_19
8. Halton, J.H.: Algorithm 247: radical-inverse quasi-random point sequence. *Commun. ACM* **7**(12), 701–702 (1964)
9. Householder, A.S.: Unitary triangularization of a nonsymmetric matrix. *J. ACM (JACM)* **5**(4), 339–342 (1958)
10. Koch, S., Kasten, J., Wiebel, A., Scheuermann, G., Hlawitschka, M.: 2D vector field approximation using linear neighborhoods. *Vis. Comput.* **32**(12), 1563–1578 (2016)
11. Laramee, R.S., Hauser, H., Zhao, L., Post, F.H.: Topology-based flow visualization, the state of the art. In: Hauser, H., Hagen, H., Theisel, H. (eds.) *Topology-Based Methods in Visualization*, pp. 1–19. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-70823-0_1

12. Ling, L., Kansa, E.J.: Preconditioning for radial basis functions with domain decomposition methods. *Math. Comput. Model.* **40**(13), 1413–1427 (2004)
13. Majdisova, Z., Skala, V.: A radial basis function approximation for large datasets. *Proc. SIGRAD* **2016**(127), 9–14 (2016)
14. Majdisova, Z., Skala, V.: Big geo data surface approximation using radial basis functions: a comparative study. *Comput. Geosci.* **109**, 51–58 (2017)
15. Majdisova, Z., Skala, V.: Radial basis function approximations: comparison and applications. *Appl. Math. Model.* **51**, 728–743 (2017)
16. Ohtake, Y., Belyaev, A., Alexa, M., Turk, G., Seidel, H.-P.: Multi-level partition of unity implicits. In: *ACM Siggraph 2005 Courses*, pp. 463–470. ACM (2005)
17. Ohtake, Y., Belyaev, A.G., Seidel, H.: 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)*, pp. 153–164, 292 (2003)
18. Orf, L., Wilhelmson, R., Wicker, L.: Visualization of a simulated long-track ef5 tornado embedded within a supercell thunderstorm. *Parallel Comput.* **55**, 28–34 (2016)
19. Skala, V.: Fast interpolation and approximation of scattered multidimensional and dynamic data using radial basis functions. *WSEAS Trans. Math.* **12**(5), 501–511 (2013)
20. Skala, V., Smolik, M.: A new approach to vector field interpolation, classification and robust critical points detection using radial basis functions. In: Silhavy, R. (ed.) *CSOC2018 2018. AISC*, vol. 765, pp. 109–115. Springer, Cham (2019). https://doi.org/10.1007/978-3-319-91192-2_12
21. Smolik, M., Skala, V.: Classification of critical points using a second order derivative. *Procedia Comput. Sci.* **108**, 2373–2377 (2017)
22. Smolik, M., Skala, V.: Large scattered data interpolation with radial basis functions and space subdivision. *Integr. Comput.-Aided Eng.* **25**(1), 49–62 (2018)
23. Smolik, M., Skala, V., Majdisova, Z.: 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. Smolik, M., Skala, V., Majdisova, Z.: Vector field radial basis function approximation. *Adv. Eng. Softw.* **123**(1), 117–129 (2018)
25. Süßmuth, J., Meyer, Q., Greiner, G.: Surface reconstruction based on hierarchical floating radial basis functions. *Comput. Graph. Forum* **29**(6), 1854–1864 (2010)
26. Tricoche, X., Scheuermann, G., Hagen, H.: A topology simplification method for 2D vector fields. In: *Visualization 2000, Proceedings*, pp. 359–366. IEEE (2000)
27. Weinkauff, T., Theisel, H., Shi, K., Hege, H.-C., Seidel, H.-P.: Extracting higher order critical points and topological simplification of 3D vector fields. In: *Visualization, 2005, VIS 2005*, pp. 559–566. IEEE (2005)
28. Yang, J., Wang, Z., Zhu, C., Peng, Q.: Implicit surface reconstruction with radial basis functions. In: Braz, J., Ranchordas, A., Araújo, H.J., Pereira, J.M. (eds.) *VISIGRAPP 2007. CCIS*, vol. 21, pp. 5–12. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-89682-1_1
29. Yokota, R., Barba, L.A., Knepley, M.G.: PetRBF-a parallel $O(N)$ algorithm for radial basis function interpolation with gaussians. *Comput. Methods Appl. Mech. Eng.* **199**(25), 1793–1804 (2010)

4.5 Vector Field Radial Basis Functions Approximation with Streamlines Curvature

During the RBF vector field approximation, the approximation error occurs. The paper [Smolik and Skala, ttedb] proposes an approach for decreasing the approximation error at important locations of the vector field. The important places of the vector fields are where the streamlines change their direction the most. According to the amount of streamlines directions change can be determined the importance of vectors.

The amount of streamlines directions change can be expressed by computing the streamlines curvature. This curvature is then used for placing of radial basis functions. This functions are mostly placed at locations with high streamlines curvature, i.e. the density of spacing of the radial basis functions is proportional to the curvature. Moreover, during the approximation of vector fields, some vectors have higher importance and some have lower importance. This paper proposes a solution for weighted RBF approximation of the vector field, where the weights are determined based on the streamlines curvature.

The result of this proposed approach is an RBF approximation of a vector field, which preserves the important behavior of the vector field. This approach controls the amount of allowed approximation error in different locations of vector field.

Citation:

- Michal Smolik and Vaclav Skala. Vector field radial basis functions approximation with streamlines curvature. *Advances in Engineering Software*, 2019. (IF = 4.194) (submitted)

Vector Field Radial Basis Functions Approximation with Streamlines Curvature

Michal Smolik · Vaclav Skala

Abstract Vector fields are usually very large and complex data sets. In order to reduce the size of vector field data sets, approximation methods are used. In this paper we propose a new approach for vector field approximation with radial basis functions (RBF). This approach computes the streamlines curvature and uses it for optimal distribution of radial basis functions and for computing the weighted approximation of the vector field. The proposed approach is tested on $2D$ and $3D$ vector fields with different complexity of the flow. The results proved the precise vector field approximation compared to other methods and, moreover, the resulting RBF approximation is in the form of an analytic function, which can be used for further processing of the vector field.

Keywords Vector field · Streamlines curvature · Weighted approximation · Radial basis functions · Compression · RBF approximation · Visualization

1 Introduction

Vector fields data are originated mostly from numerical simulations, i.e. from computation fluid dynamics (CFD) [2], [24] and finite element methods (FEM) [20], [4]. Nowadays, the resulting vector fields are very large data sets that are difficult to process and visualize. Approximation [6], [25] is usually used to compress the

vector field. Many techniques for vector field approximation are used to compress and simplify the vector field.

The papers [34] and [35] describe continuous simplification of the topology of planar vector fields and thus creation of a compressed vector field. The method does not change the grid while simplifying the vector field by removing pairs of critical points. Higher order critical points are extracted in [38] and the topology of a $3D$ vector field is simplified. A convex surface is placed around the area of interest, i.e. on area with a lot of critical points, and then is this area simplified. This method is mostly used to find a simplified visual representation of clusters of critical points in complex $3D$ vector fields. The paper [37] performs topology-based smoothing of $2D$ scalar fields with C^1 -continuity. The method allows filtering out spurious features that arise due to noise. To do that, the feature significance is rated according to topological persistence. The paper [33] combines two approaches, i.e. a topological simplification technique and topology preserving compression. The important features are preserved during the compression, while the unimportant features are allowed to collapse and disappear. The approach is able to significantly decrease the mesh size of a vector field and thus increase the compression ratio. The paper [18] presents a technique for visualization of multi-level topology of vector fields. For each level of details a set of critical points is selected, that defines the global character of the flow and approximates the vector field. A detailed state of the art [17] describes the topology-based visualization. When extracting the topology of the vector field, we end up with a compact representation of the flow and can use it as an approximation or data compression. The papers [26] and [27] present a robust vector field simplification method.

M. Smolik
Department of Computer Science and Engineering, Faculty of Applied Sciences, University of West Bohemia, Plzen, Czech Republic

V. Skala
Department of Computer Science and Engineering, Faculty of Applied Sciences, University of West Bohemia, Plzen, Czech Republic

The approach tries to remove pairs of critical points based on robustness measurement. The removal process creates a hierarchical simplification scheme, so the user can select the required compression. The paper [12] performs scalar field denoising based on global energy optimization. This approach preserves extremes of the scalar field while smoothing the rest of the data set. Vector field approximation based on segmentation is presented in [15]. Each segmentation region of a vector field is approximated using an affine linear function which is presented in [36] and the final approximation has defined some maximal error below the user-defined value. The paper [16] discusses the impact of vector field approximation on visualization techniques using the example of Finite-Time Lyapunov Exponent computations. The paper [39] presents a filtering technique based on rotation of the vector fields to eliminate the less interesting and sometimes sporadic critical points. The Delaunay triangulation is used in [7] to simplify the vector field. From the Delaunay triangulation, some vortices are removed based on a local error metric and the final approximated vector field is reconstructed using piece-wise linear approximation.

Vector fields can be approximated using Radial basis functions (RBF) [3]. The global RBF collocation methods are mostly used due to smoothness; however, the paper [3] performs a numerical comparison between global and local RBFs for vector field approximation and shows that local RBFs are suitable for vector field approximation as well. The paper [23] performs easy and simple RBF approximation, which is suitable for parallelism. The paper [31] presents an algorithm for critical points reduction. Critical points are removed from the vector field based on the critical point influence. Only the important critical points are preserved in the final RBF approximation of the vector field. This technique is also used in [30] to approximate 3D vector fields. Vector field on sphere approximation using RBF is presented in [28]. The paper presents experimental results and recommends the setting for vector field RBF approximation on a sphere.

2 Proposed Approach

We propose a new approach for 2D and 3D vector field approximation using Radial basis functions. The proposed approach uses the streamlines curvature to determine the location of centers for RBF approximation and it is used for weighting the RBF approximation as well. The places, where the streamlines curvature is high, are important, as in those locations there are critical points or the vector field changes significantly.

First, we propose the formulas for 2D and 3D streamlines curvature calculation. Next, we define the weighted RBF approximation of a vector field and finally, we select the best centers of radial basis functions.

2.1 Streamlines Curvature

Curvature formulas for implicit curves and surfaces are derived from the classical curvature formulas in Differential Geometry for parametric curves and surfaces [11], [1]. For 2D implicit planar curves $F(x, y) = 0$, the curvature depends only on the gradient ∇F and the hessian $H(F)$. The gradient ∇F is computed as

$$\nabla F = \left[\frac{\partial F}{\partial x}, \frac{\partial F}{\partial y} \right] = [F_x, F_y], \quad (1)$$

where F_x and F_y of vector field streamlines have the following meaning:

$$\begin{aligned} F_x &= v_y \\ F_y &= -v_x, \end{aligned} \quad (2)$$

as the tangential vector $\mathbf{v} = [v_x, v_y]$ is orthogonal to the vector $[F_x, F_y]$. The hessian $H(F)$ is computed as

$$H(F) = \begin{bmatrix} \frac{\partial^2 F}{\partial x^2} & \frac{\partial^2 F}{\partial x \partial y} \\ \frac{\partial^2 F}{\partial y \partial x} & \frac{\partial^2 F}{\partial y^2} \end{bmatrix} = \begin{bmatrix} -\frac{\partial v_y}{\partial x} & -\frac{\partial v_y}{\partial y} \\ \frac{\partial v_x}{\partial x} & \frac{\partial v_x}{\partial y} \end{bmatrix}. \quad (3)$$

The curvature of vector field streamlines is computed using the following equation:

$$k = -\frac{\mathbf{v} \cdot \mathbf{H}(\mathbf{F}) \cdot \mathbf{v}^T}{\|\mathbf{v}\|^3}. \quad (4)$$

Substituting (1), (2) and (3) into (4), we get the following equation:

$$k = -\frac{[v_x, v_y] \cdot \begin{bmatrix} -\frac{\partial v_y}{\partial x} & -\frac{\partial v_y}{\partial y} \\ \frac{\partial v_x}{\partial x} & \frac{\partial v_x}{\partial y} \end{bmatrix} \cdot [v_x, v_y]^T}{(v_x^2 + v_y^2)^{\frac{3}{2}}}. \quad (5)$$

In some cases, there can be some numerical problems. If $\|\mathbf{v}\| = 0$, then there would be a division by zero. In this case, we can replace the curvature k with a value of infinity. Another problem that can arise is when $\mathbf{v} \cdot \mathbf{H}(\mathbf{F}) \cdot \mathbf{v}^T = 0$. In this case, the curvature is zero, but for further processing we need to replace it with some ϵ value that is close to zero.

Streamlines curvature in 3D can be computed in a way similar to that used in 2D. There is no implicit formula for description of a line in 3D, thus we need to derive the curvature from a parametric curve, i.e. a

streamline. At each point the streamline has a directional vector

$$\mathbf{v} = [v_x, v_y, v_z]. \quad (6)$$

To compute the curvature of streamlines in 3D, we need to compute the following formula:

$$k = \frac{\|(\mathbf{v} \cdot \nabla \mathbf{v}) \times \mathbf{v}\|}{\|\mathbf{v}\|^3}, \quad (7)$$

where $\nabla \mathbf{v}$ is the gradient of \mathbf{v} computed as

$$\nabla \mathbf{v} = \begin{bmatrix} \frac{\partial v_x}{\partial x} & \frac{\partial v_y}{\partial x} & \frac{\partial v_z}{\partial x} \\ \frac{\partial v_x}{\partial y} & \frac{\partial v_y}{\partial y} & \frac{\partial v_z}{\partial y} \\ \frac{\partial v_x}{\partial z} & \frac{\partial v_y}{\partial z} & \frac{\partial v_z}{\partial z} \end{bmatrix}. \quad (8)$$

Substituting (6), (8) into (7), we get the following formula for the curvature of 3D streamlines:

$$k = \frac{\left\| \left([v_x, v_y, v_z] \begin{bmatrix} \frac{\partial v_x}{\partial x} & \frac{\partial v_y}{\partial x} & \frac{\partial v_z}{\partial x} \\ \frac{\partial v_x}{\partial y} & \frac{\partial v_y}{\partial y} & \frac{\partial v_z}{\partial y} \\ \frac{\partial v_x}{\partial z} & \frac{\partial v_y}{\partial z} & \frac{\partial v_z}{\partial z} \end{bmatrix} \right) \times [v_x, v_y, v_z] \right\|}{(v_x^2 + v_y^2 + v_z^2)^{\frac{3}{2}}}. \quad (9)$$

The same numerical problems as when computing the curvature of 2D streamlines can arise. However, those problems are solved in the same way as for 2D streamlines curvature calculation.

2.2 Vector Field Approximation

Vector field in 2D and 3D can be approximated using Radial basis functions [31], [28]. The proposed approach focuses on maintaining the important details of the vector field, where the curvature of streamlines is high. It is necessary to reformulate the approximation scheme to give some places higher importance than to others. When computing the standard RBF approximation, the final solution has the following formula:

$$\mathbf{v}(\mathbf{x}) = [v_x(\mathbf{x}), v_y(\mathbf{x}), v_z(\mathbf{x})] = \sum_{j=1}^M \lambda_j \phi(\|\mathbf{x} - \boldsymbol{\xi}_j\|), \quad (10)$$

where $\boldsymbol{\xi}_j$ are centers of RBF, M is the number of RBF centers, $\lambda_j = [\lambda_j^{(x)}, \lambda_j^{(y)}, \lambda_j^{(z)}]$ are weights of RBF and $\phi(r)$ is the radial basis function. There are many RBFs; however, in our experiments we used the Thin plate spline (TPS) function

$$\phi(r) = r^2 \log r. \quad (11)$$

The TPS is shape parameter free function, which has a great advantage over other radial basis functions.

To solve the RBF approximation, we need to solve the following over-determined system of linear equations:

$$\mathbf{v}_i = \sum_{j=1}^M \lambda_j \phi(\|\mathbf{x}_i - \boldsymbol{\xi}_j\|), \quad \text{for } \forall i \in \{1, \dots, N\}, \quad (12)$$

where $\{\mathbf{x}_i, \mathbf{v}_i\}_{i=0}^N$ is the input data set.

We can rewrite RBF approximation (12) using the matrix notation

$$\mathbf{A}\boldsymbol{\lambda} = \mathbf{v}. \quad (13)$$

The system of linear equations (13) is over-determined. Each one linear equation has the same weight in the system of linear equations. If we multiply the i -th line in matrix \mathbf{A} and the corresponding elements in \mathbf{v} by some number k , the solution of (13) will give more importance to the i -th linear equation. The reason for this is simple. If we minimize the square root of

$$\mathit{err} = [(\mathbf{A}\boldsymbol{\lambda}_x - \mathbf{v}_x)^2, (\mathbf{A}\boldsymbol{\lambda}_y - \mathbf{v}_y)^2, (\mathbf{A}\boldsymbol{\lambda}_z - \mathbf{v}_z)^2] \quad (14)$$

and elements in the i -th row of \mathbf{A} and \mathbf{v} are multiplied by some number k , then the error for the i -th row will be higher by k^2 . To minimize the square root error, the solution must attract the final approximation closer to the i -th condition given by the i -th linear equation and thus give the i -th condition higher importance.

In the proposed approach, we want to approximate the vector field more precisely in locations where streamlines have high curvature. This can be achieved by multiplying all linear equations with some positive values that depend on the curvature. For locations with high curvature, we want to multiply the linear equations with some high positive values and for locations with low curvature, we want to multiply the linear equations with some lower positive values. We can compute curvature at all input points, i.e. we get a set of curvatures

$$\mathbf{k} = \{k_i\}_{i=1}^N. \quad (15)$$

We need to crop the maximal and minimal values. We find k_{10min} as 10% of \mathbf{k} are smaller than k_{10min} and k_{10max} as 10% of \mathbf{k} are larger than k_{10max} . The set \mathbf{k} can contain numbers close to zero as well as numbers much larger than 1, i.e. even larger than 10^5 . This kind of value cannot be directly used to multiply the rows of matrices \mathbf{A} and \mathbf{v} . We transform this set to a new one

using the logarithm of curvature [10], [41]. The result is as the following:

$$\bar{k} = 1 + 4 \frac{\{\log(k_i)\}_{i=1}^N}{\log(k_{10max}) - \log(k_{10min})} - \log(k_{10min}), \quad (16)$$

where the coefficients 1 and 4 are selected because the set of weights $\bar{k} = \{\bar{k}_i\}_{i=1}^N$ needs to be cropped to the interval [1; 5]. This interval was experimentally selected as the most appropriate for all data sets that we tested. The final vector of weights \bar{k} for the system of linear equations is

$$\bar{k}_i = \begin{cases} 1 & \bar{k}_i < 1 \\ \bar{k}_i & \bar{k}_i \in [1; 5] \\ 5 & \bar{k}_i > 5 \end{cases} \quad \text{for } \forall i \in \{1, \dots, N\}. \quad (17)$$

Now, the final over-determined weighted system of linear equations has the following form:

$$\bar{k}_i \mathbf{v}_i = \bar{k}_i \sum_{j=1}^M \lambda_j \phi(\|\mathbf{x}_i - \boldsymbol{\xi}_j\|), \quad \text{for } \forall i \in \{1, \dots, N\}, \quad (18)$$

where \bar{k}_i are elements of \bar{k} .

We can rewrite the RBF approximation (18) using the matrix notation as

$$\bar{\mathbf{A}}\boldsymbol{\lambda} = \bar{\mathbf{v}}, \quad (19)$$

where rows in $\bar{\mathbf{A}}$ and $\bar{\mathbf{v}}$ are multiplied by elements from \bar{k} , as explained in (18).

The important locations for RBF approximation are centers of radial basis functions. These centers should be mostly located in the locations with high curvature of streamlines, as we want the approximated vector field more precisely at those locations.

The initial locations of RBF centers are randomly generated points with Halton distribution (A.1 in [9], [40]). To put more points in the locations with high curvature and fewer points in the locations with low curvature, we need to move all the points in the gradient direction of the curvature. The amount of how much to move the points was experimentally selected as 12.5% of the gradient path to the local extreme of the curvature.

The proposed approach for vector field approximation can be used for 2D as well as 3D vector fields. The only difference between 2D and 3D is the calculation of streamlines curvature.

3 Experimental Results

In this section we verify the proposed approach for vector field approximation. We test the proposed approach with different types of vector fields: from a simple and smooth synthetic 2D vector field up to a large and complex real 3D vector field. The results of the proposed vector field approximation are also compared with a different approach for vector field approximation.

The first example of a vector field for testing our proposed approach is a synthetic vector field (see Fig. 1) described by the following formula:

$$\begin{aligned} v_x &= \frac{1}{2} [x(x^2 + 1) - y(2x - (y - 2)y - 1)] \\ v_y &= \frac{1}{2} [x^2y - x(y^2 - 2y + 1) + y - 2], \end{aligned} \quad (20)$$

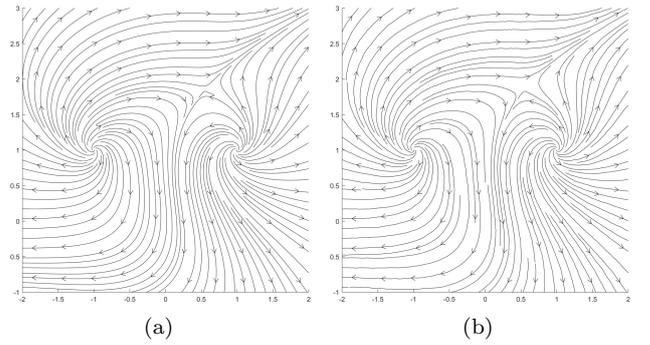


Fig. 1 The original vector field computed from (20) (a) and the approximated vector field using the proposed approach (b). The number of centers for RBF approximation is 256.

After computing the streamlines curvature, we can create two visualizations, see Fig. 2. It can be seen that the highest absolute value of curvature is located around critical points and below the most left critical point, where the vector field changes its direction about 90°. On the other hand, the lowest value of \bar{k} (16) is as expected at locations where streamlines are straight lines or the direction of flow does not change significantly.

The centers for RBF approximation are mostly randomly distributed over the approximated domain. However, in our case, we distribute these points according to the streamlines curvature, see Fig. 3. It can be seen that the RBF centers are located with higher density at locations with higher streamlines curvature. On the other hand, the RBF centers are located with low density at locations with low streamlines curvature.

The final result of the proposed weighted RBF approximation of vector field (20) is visualized in Fig. 1b.

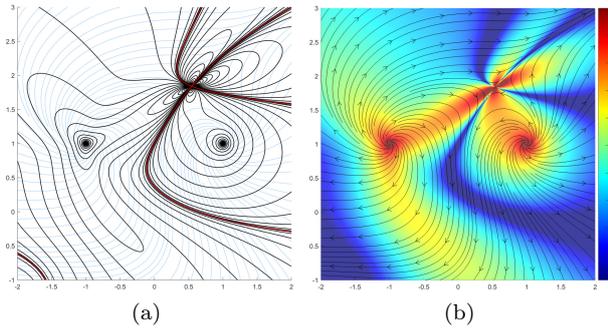


Fig. 2 Iso-contours of vector field curvature with highlighted (red color) value for curvature zero iso-line (a). Logarithm of absolute value of curvature visualized as a color plot (the value \bar{k} is cropped to the interval $[1; 5]$, see (16)) (b).

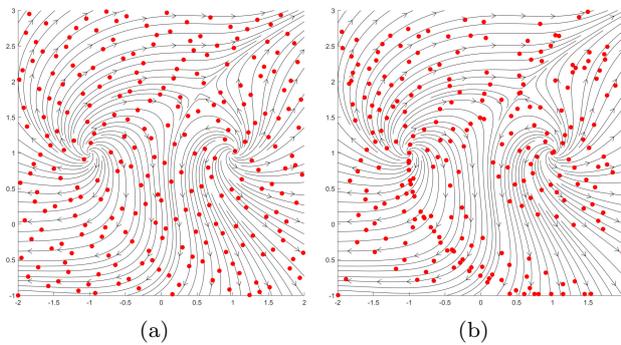


Fig. 3 The location of 256 centers for RBF approximation. Centers with Halton distribution (a) and centers distributed according to the streamlines curvature (b).

We can see that both vector fields in Fig. 1 are almost identical, i.e. the approximation preserves all the important details of the vector field.

Another example of a dynamical system is a chemostat [21], [14]. A chemostat is a bioreactor to which a fresh medium is continuously added, while culture liquid containing leftover nutrients, metabolic end products and microorganisms are continuously removed at the same rate to keep the culture volume constant. An example description of a chemostat dynamical system is

$$\begin{aligned} v_x &= 1 - x - ay \\ v_y &= -y + ay, \end{aligned} \quad (21)$$

where x is the concentration of the nutrient solution in the growth chamber, y is the density of culture population and a is [21], [14]

$$a = \frac{x}{\frac{1}{16} + \frac{x}{4} + x^2}. \quad (22)$$

Visualization of the dynamical system described by (21) is in Fig. 4.

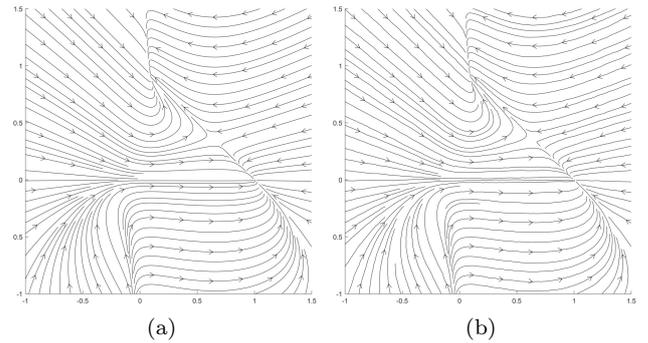


Fig. 4 The original vector field computed from (21) (a) and the approximated vector field using the proposed approach (b). The number of centers for RBF approximation is 256.

The visualization of streamlines curvature can be seen in Fig. 5. The vector field is more complex than (20) and thus the streamlines curvature is more variable.

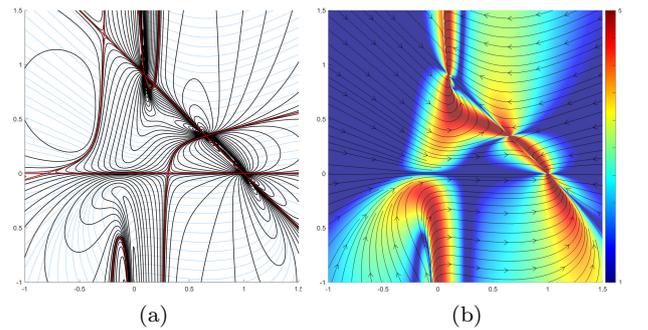


Fig. 5 Iso-contours of vector field curvature with highlighted (red color) value for curvature zero iso-line (a). Logarithm of absolute value of curvature visualized as a color plot (the value \bar{k} is cropped to the interval $[1; 5]$, see (16)) (b).

We can use the streamlines curvature to distribute the centers for RBF approximation as proposed in this approach. Again, it can be seen that the centers for RBF are distributed according to the streamlines curvature, see Fig. 6. The final proposed RBF approximation of the vector field (21) is presented in Fig. 4b.

The proposed approach needs to be tested with a real data set as well. We selected a numerical forecast wind data set¹ taken from [5]. This vector field data set contains $2.2 \cdot 10^4$ points with associated vector values and is visualized in Fig. 7a. The vector field is much more complex than the first two synthetic examples generated from (20) and (21).

¹ Data set of wind flow at a height of 10m over the surface of the Czech Republic, courtesy of the Institute of Computer Science of the Czech Academy of Sciences.

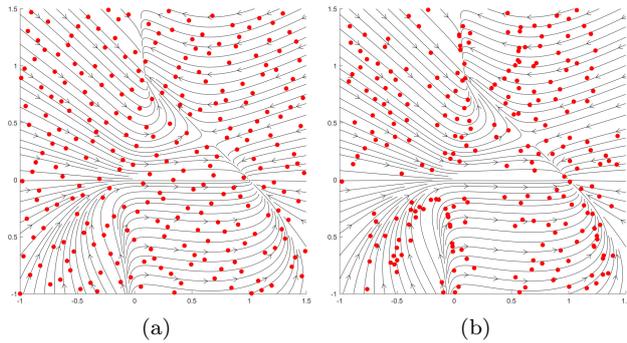


Fig. 6 The location of 256 centers for RBF approximation. Centers with Halton distribution (a) and centers distributed according to the streamlines curvature (b).

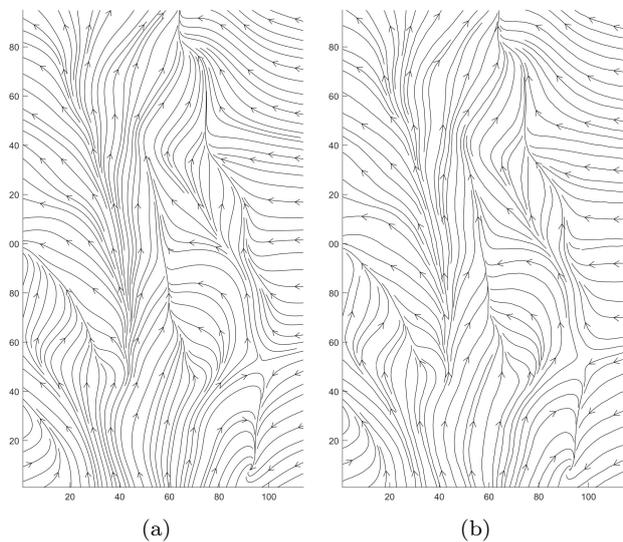


Fig. 7 The original vector field (a) and the approximated vector field using the proposed approach with the compression ratio 55.5 : 1 (b).

When computing the streamlines curvature, we end-up with the visualization in Fig. 8b. It can be seen that the highest streamlines curvature is at locations, where critical points occur and thus the flow direction is significantly changing the most at those locations. When computing the proposed vector field RBF approximation, we need to distribute RBF centers according to the streamlines curvature. The resulting location of those centers is visualized in Fig. 8a.

The result of the proposed approach for the forecast data set vector field approximation using streamlines curvature is visualized in Fig. 7b. It can be seen that the approximation preserves all the details of the original data set, while achieving a very high compression ratio (55.5 : 1).

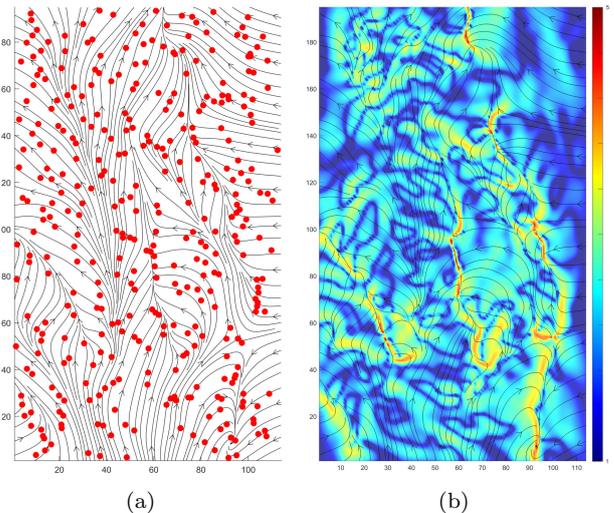


Fig. 8 The location of 400 centers for RBF approximation that are distributed according to the streamlines curvature (a). Logarithm of absolute value of curvature visualized as a color plot (the value k is cropped to the interval [1; 5], see (16)) (b).

In the last example we use the EF5 tornado data set² (from [22]), see Fig. 9a. This data set contains around $5.5 \cdot 10^8$ 3D points with associated 3D vectors.

According to the visualization in Fig. 9b, it can be seen that the highest streamlines curvature is around the main swirl of the tornado and at the top part of the data set. The highest density of RBF centers will be at these locations. When computing the final weighted RBF approximation, we need to use the approach proposed in [29], [30] for approximation of large vector fields using the space subdivision. The resulting approximation of the vector field is visualized in Fig. 9a and is almost identical with the original one. In this case, compression ratio $10^3 : 1$ was reached.

3.1 Comparison with Existing Approaches

The proposed approach needs to be compared with different approaches for vector field approximation. The first one for comparison and the most similar one is the standard RBF approximation.

$$\mathbf{v}(\mathbf{x}) = \sum_{j=1}^M \lambda_j \phi(\|\mathbf{x} - \boldsymbol{\xi}_j\|), \quad (23)$$

where $\boldsymbol{\xi}_j$ are randomly distributed centers of RBF (with Halton distribution (A.1 in [9], [40])), M is the number

² Data set of an EF5 tornado courtesy of Leigh Orf from Cooperative Institute for Meteorological Satellite Studies, University of Wisconsin, Madison, WI, USA.

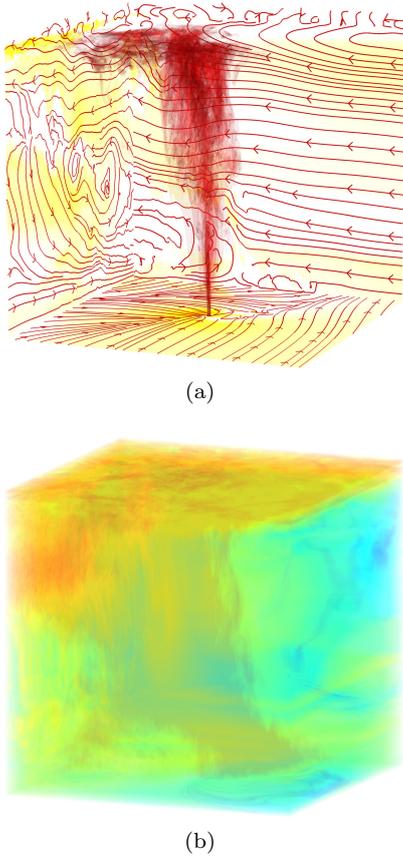


Fig. 9 The approximated vector field using the proposed approach with the compression ratio $10^3 : 1$ (a). Logarithm of absolute value of curvature visualized as a color plot (the value \bar{k} is cropped to the interval $[1; 5]$, see (16)) (b).

of RBF centers, λ_j are weights of RBF and $\phi(r)$ is the radial basis function [19], [32]. The radial basis function that is used for the standard RBF approximation is the same as the radial basis function used in the proposed approach, i.e. the following one:

$$\phi(r) = r^2 \log r. \quad (24)$$

The second approach for comparison is the Fourier transform approximation of a vector field [8], [13]. The vector field is transformed using the Fourier transform of the form

$$F(\alpha, \beta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} v(x, y) e^{-2\pi i(x\alpha + y\beta)} dx dy, \quad (25)$$

where α and β are frequencies of the vector field. To approximate the vector field we need to discard high frequencies and keep only some (according to the required compression ratio) low frequencies. The approx-

imated vector field can then be reconstructed using the following inverse Fourier transform:

$$v(x, y) = \frac{1}{(2\pi)^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(\alpha, \beta) e^{2\pi i(x\alpha + y\beta)} d\alpha d\beta. \quad (26)$$

To measure the quality of the approximation, we can compute the approximation error using the following formula:

$$Err = \frac{\sum_{i=1}^N \|\mathbf{v}_i - \bar{\mathbf{v}}_i\|}{\sum_{i=1}^N \|\bar{\mathbf{v}}_i\|}, \quad (27)$$

where \mathbf{v} is the approximated vector field and $\bar{\mathbf{v}}$ is the original vector field. This formula computes the relative difference in vector size. The approximation errors of vector fields Fig. 1b, Fig. 4b, Fig. 7b and Fig. 9a are summarized in the Table 1. Each data set was approximated with all three approximation approaches with the same compression ratio, i.e. the approximation errors are comparable.

Table 1 Vector field approximation error for different data sets and different approximation methods.

	Proposed RBF	Standard RBF	Fourier transform
Fig. 1	2.34%	3.16%	3.07%
Fig. 4	3.38%	4.59%	4.46%
Fig. 7	4.42%	6.24%	5.93%
Fig. 9	10.71%	15.37%	14.18%

It can be seen that the proposed approach for vector field approximation gives the best results for all tested 2D and 3D data sets. The proposed approach has about 35%–40% lower approximation error than the standard RBF vector field approximation and about 31%–34% lower approximation error than the Fourier transform approximation.

4 Conclusion

We propose a new approach for vector field approximation with radial basis functions. This approach computes the curvature of vector field streamlines and uses it for optimal placing of reference points and for computing the weighted RBF approximation. This approach is convenient for both 2D and 3D vector fields. The approximation results proved, this to be the better approximation of vector fields compared to the standard RBF approximation and to the Fourier transform approximation.

In future research, we plan to compare different radial basis functions with variable shape parameters. The optimal shape parameter for each radial basis function should be computed according to the streamlines curvature and data distribution.

Acknowledgments.

The authors would like to thank their colleagues at the University of West Bohemia, Plzen, for their discussions and valuable suggestions. The research was supported by projects Czech Science Foundation (GACR) No. 17-05534S and partially by SGS 2019-016.

References

- Aléssio, O.: Formulas for second curvature, third curvature, normal curvature, first geodesic curvature and first geodesic torsion of implicit curve in n-dimensions. *Computer Aided Geometric Design* **29**(3-4), 189–201 (2012)
- Blazek, J.: *Computational fluid dynamics: principles and applications*. Butterworth-Heinemann (2015)
- Cabrera, D.A.C., Gonzalez-Casanova, P., Gout, C., Juárez, L.H., Reséndiz, L.R.: Vector field approximation using radial basis functions. *Journal of Computational and Applied Mathematics* **240**, 163–173 (2013)
- Chung, E.T., Efendiev, Y., Lee, C.S.: Mixed generalized multiscale finite element methods and applications. *Multiscale Modeling & Simulation* **13**(1), 338–366 (2015)
- Corbet, L., Vlček, O., Eben, K., Liczki, J., Benešová, N., Modlák, M.: Regional air quality forecasting for the Czech Republic. In: 15th International Conference on Harmonisation within Atmospheric Dispersion Modelling for Regulatory Purposes, HARMO 2013, pp. 150–154 (2013)
- Davis, P.J.: *Interpolation and approximation*. Courier Corporation (1975)
- Dey, T.K., Levine, J.A., Wenger, R.: A delaunay simplification algorithm for vector fields. In: *Computer Graphics and Applications, 2007. PG'07. 15th Pacific Conference on*, pp. 281–290. IEEE (2007)
- Ebling, J., Scheuermann, G.: Clifford fourier transform on vector fields. *IEEE Transactions on Visualization and Computer Graphics* **11**(4), 469–479 (2005)
- Fasshauer, G.E.: *Meshfree approximation methods with MATLAB*, vol. 6. World Scientific (2007)
- Gobithaasan, R., Miura, K.T.: Logarithmic curvature graph as a shape interrogation tool. *Applied Mathematical Sciences* **8**(16), 755–765 (2014)
- Goldman, R.: Curvature formulas for implicit curves and surfaces. *Computer Aided Geometric Design* **22**(7), 632–658 (2005)
- Günther, D., Jacobson, A., Reininghaus, J., Seidel, H.P., Sorkine-Hornung, O., Weinkauff, T.: Fast and memory-efficiently topological denoising of 2D and 3D scalar fields. *IEEE transactions on visualization and computer graphics* **20**(12), 2585–2594 (2014)
- Hitzer, E.M.: Quaternion fourier transform on quaternion fields and generalizations. *Advances in Applied Clifford Algebras* **17**(3), 497–517 (2007)
- James, T.W.: Continuous culture of microorganisms. *Annual Reviews in Microbiology* **15**(1), 27–46 (1961)
- Koch, S., Kasten, J., Wiebel, A., Scheuermann, G., Hlawitschka, M.: 2D vector field approximation using linear neighborhoods. *The Visual Computer* **32**(12), 1563–1578 (2016)
- Koch, S., Volke, S., Scheuermann, G., Hagen, H., Hlawitschka, M.: Comparing finite-time lyapunov exponents in approximated vector fields. In: *Topological Methods in Data Analysis and Visualization*, pp. 267–281. Springer (2015)
- Laramee, R.S., Hauser, H., Zhao, L., Post, F.H.: Topology-based flow visualization, the state of the art. In: *Topology-based methods in visualization*, pp. 1–19. Springer (2007)
- de Leeuw, W., van Liere, R.: Multi-level topology for flow visualization. *Computers & Graphics* **24**(3), 325–331 (2000)
- Majdisova, Z., Skala, V.: Radial basis function approximations: Comparison and applications. *Applied Mathematical Modelling* **51**, 728–743 (2017)
- Molina-Aiz, F., Fatnassi, H., Boulard, T., Roy, J., Valera, D.: Comparison of finite element and finite volume methods for simulation of natural ventilation in greenhouses. *Computers and electronics in agriculture* **72**(2), 69–86 (2010)
- Novick, A., Szilard, L.: Description of the chemostat. *Science* **112**(2920), 715–716 (1950)
- Orf, L., Wilhelmson, R., Wicker, L.: Visualization of a simulated long-track ef5 tornado embedded within a supercell thunderstorm. *Parallel Computing* **55**, 28–34 (2016)
- Roy, T., Gout, C., Le Guyader, C., Lenglar, E.: Wind velocity field approximation from sparse data. In: *Geoscience and Remote Sensing Symposium (IGARSS), 2013 IEEE International*, pp. 1606–1609. IEEE (2013)
- Sanderse, B., Pijl, S., Koren, B.: Review of computational fluid dynamics for wind turbine wake aerodynamics. *Wind energy* **14**(7), 799–819 (2011)
- Schagen, I.P.: Interpolation in two dimensions - a new technique. *IMA Journal of Applied Mathematics* **23**(1), 53–59 (1979)
- Skraba, P., Wang, B., Chen, G., Rosen, P.: 2D vector field simplification based on robustness. In: *Visualization Symposium (PacificVis), 2014 IEEE Pacific*, pp. 49–56. IEEE (2014)
- Skraba, P., Wang, B., Chen, G., Rosen, P.: Robustness-based simplification of 2D steady and unsteady vector fields. *IEEE transactions on visualization and computer graphics* **21**(8), 930–944 (2015)
- Smolik, M., Skala, V.: Spherical RBF vector field interpolation: experimental study. In: *Applied Machine Intelligence and Informatics (SAMII), 2017 IEEE 15th International Symposium on*, pp. 000,431–000,434. IEEE (2017)
- Smolik, M., Skala, V.: Large scattered data interpolation with radial basis functions and space subdivision. *Integrated Computer-Aided Engineering* **25**(1), 49–26 (2018)
- Smolik, M., Skala, V., Majdisova, Z.: 3D vector field approximation and critical points reduction using radial basis functions. In: *International Conference on Applied Physics, System Science and Computers*. Springer (2018)
- Smolik, M., Skala, V., Majdisova, Z.: Vector field radial basis function approximation. *Advances in Engineering Software* **123**, 117–129 (2018)
- Smolik, M., Skala, V., Nedved, O.: A comparative study of LOWESS and RBF approximations for visualization. In: *International Conference on Computational Science and Its Applications*, pp. 405–419. Springer (2016)

33. Theisel, H., Rossel, C., Seidel, H.P.: Combining topological simplification and topology preserving compression for 2D vector fields. In: *Computer Graphics and Applications, 2003. Proceedings. 11th Pacific Conference on*, pp. 419–423. IEEE (2003)
34. Tricoche, X., Scheuermann, G., Hagen, H.: A topology simplification method for 2D vector fields. In: *Visualization 2000. Proceedings*, pp. 359–366. IEEE (2000)
35. Tricoche, X., Scheuermann, G., Hagen, H.: Continuous topology simplification of planar vector fields. In: *Proceedings of the conference on Visualization'01*, pp. 159–166. IEEE Computer Society (2001)
36. Volke, S., Koch, S., Hlawitschka, M.: Identifying linear vector fields on 2D manifolds (2016)
37. Weinkauff, T., Gingold, Y., Sorkine, O.: Topology-based smoothing of 2D scalar fields with c_1 -continuity. In: *Computer Graphics Forum*, vol. 29, pp. 1221–1230. Wiley Online Library (2010)
38. Weinkauff, T., Theisel, H., Shi, K., Hege, H.C., Seidel, H.P.: Extracting higher order critical points and topological simplification of 3D vector fields. In: *Visualization, 2005. VIS 05. IEEE*, pp. 559–566. IEEE (2005)
39. Wong, P.C., Foote, H., Leung, R., Jurrus, E., Adams, D., Thomas, J.: Vector fields simplification-a case study of visualizing climate modeling and simulation data sets. In: *Visualization 2000. Proceedings*, pp. 485–488. IEEE (2000)
40. Wong, T.T., Luk, W.S., Heng, P.A.: Sampling with hammersley and halton points. *Journal of graphics tools* **2**(2), 9–24 (1997)
41. Yoshida, N., Fukuda, R., Saito, T.: Logarithmic curvature and torsion graphs. In: *International Conference on Mathematical Methods for Curves and Surfaces*, pp. 434–443. Springer (2008)

4.6 Radial Basis Function and Multi-level 2D Vector Field Approximation

In many areas of visualization is very useful the technique called level of details. This technique, as well as the proposed approach in [Smolik and Skala, tteda], can be used to increase the transferred amount of data between data servers and visualization devices, or to give user a simplified visualization without small details.

The paper [Smolik and Skala, tteda] presents an approach for multi-level vector field approximation. The proposed approach uses the Radial basis functions to simplify and to approximate the vector field. The initial level of details approximation represents only the main global character of the vector field. The next level of details is represented as the offset to the previous level of details vector field approximation. This representation helps to increase the compression ratio.

The proposed approach proved its ability to a high compression ratio. It is especially convenient for visualization on mobile devices, as only some first level of details are downloaded and thus it is reduced the necessary data transfer.

Citation:

- Michal Smolik and Vaclav Skala. Radial basis function and multi-level 2D vector field approximation. *Mathematics and Computers in Simulation*, 2019. (IF = 1.409) (submitted)

Radial Basis Function and Multi-level 2D Vector Field Approximation

Michal Smolik^a and Vaclav Skala^a

^aFaculty of Applied Sciences, University of West Bohemia, Plzen, Czech Republic

ABSTRACT

We propose a new approach for meshless multi-level radial basis function (ML-RBF) approximation which offers data sensitive compression and progressive details visualization. It leads to an analytical description of compressed vector fields, too. The proposed approach approximates the vector field in multiple levels of details. The low level approximation removes minor flow patterns while the global character of the flow remains unchanged. And conversely, the higher level approximation contains all small details of the vector field. The ML-RBF has been tested with a numerical forecast data set to prove its ability to handle data with complex topology. Comparison with the Fourier vector field approximation has been made and significant advantages, i.e. high compression ratio, accuracy, extensibility to a higher dimension etc., of the proposed ML-RBF were proved.

KEYWORDS

Radial basis functions; adaptive shape parameter; vector field; approximation; Gaussian low-pass filter; Fourier transform

1. Introduction

In applied sciences, interpolation and approximation are very often used methods [8]. This paper propose a new approach for meshless multi-level radial basis function (ML-RBF) approximation which offers data sensitive compression, progressive details visualization and leads to analytical description of compressed vector fields. It is capable to handle vector data fields with complex topology as well.

A vector field is a function that assigns to each point a vector. Vector fields come mostly from numerical simulations, i.e. Computational Fluid Dynamics (CFD) [20], [21], [1], [15], [10] and Finite Element Method (FEM) [31], [6]. The analysys of the vector field can be done at any location of the vector field [18], [42], [44]. However the most important places of the vector field are so-called critical points [18].

Topology-based flow visualization is well known technique [25]. However, the result can be a cluttered image which is difficult to interpret, when the topology-based technique is used in complex and information-rich data sets. One solution of this problem is described in the paper [48], which optimizes the topology. The Multi-level topology visualization of vector field data sets is presented in [9]. The algorithm visualizes the topology without excessive cluttering while maintaining the global structure of the flow. Another approach [4] uses fully adaptive multiresolution schemes for strongly

The research was supported by projects Czech Science Foundation (GACR) No. 17-05534S and partially by SGS 2019-016.

degenerate parabolic equations with discontinuous flux. The paper [51] uses a multi-scale model for solute transport in a wavy-waled channel. This approach concerns steady flow and identifies conditions under which is the approximation uniformly valid in a full channel flow. The paper [40] simplifies the vector field using the reduction of critical points according to their stability measurement, which is computed as the minimum amount of vector field perturbation that is required to remove the critical point. After this, critical points can be hierarchically removed. The global energy change during simplification of topology is computed in [17]. This energy is optimized during the denoising of the vector field. The algorithm for topology-controlled denoising of scalar fields, which processes small patches of the domain independently, is presented in [17]. It is based on a global energy optimization and avoids the introduction of new critical points. The paper [5] describes a numerical comparison between RBF local and global methods and highlights the possible advantage of using local methods for the approximation of vector fields. The vector field approximation for two-dimensional vector fields that preserves their topology and significantly reduces the memory footprint is presented in [24]. This approximation is based on a segmentation and the flow within each segmentation region is approximated by an affine linear function. The paper [27] reduces the size degree of the complexity of density variations. This approach is compared with a phase-field method [23].

The Fourier transform decomposes a function into the frequencies that make it up. It can be used for vector field analysis and approximation or simplification. The Clifford Fourier transform in [11] allows a frequency analysis of vector fields and the behavior of vector-valued filters. In frequency space, vectors are transformed into general multivectors of the Clifford Algebra. Many basic vector-valued patterns, such as source, sink, saddle points, and potential vortices, can be described by a few multivectors in frequency space. A two-dimensional filtering operation, involving both curl and divergence, is applied in [30] to the $2D$ Clifford Fourier Transform in order to simultaneously enhance important features of a $2D$ vector field, such as vortices and pairs of sources and sinks. The approach [34] defines convolution operators on vector fields using geometric algebra. This includes a corresponding Clifford Fourier transform of a spatial vector or multivector data. This approach is used for the analysis of the fluid flow. There are also different approaches to transforming vector-valued data using a Fourier transform [3], [19] and [12].

The proposed (ML-RBF) vector field approximation method has variety of uses. The vector field data sets come mostly from the numerical simulations and contains very large number of sample points, i.e. the data set is very large. This data sets need to be stored for future use and backup. Thus, the approximation techniques are used to compress the vector field data sets. For this reason, we propose a new technique for vector field ML-RBF approximation. The ML-RBF technique is suitable for fast preview of the vector field data set as visualization can be done using only first level of details or few first levels of details. This is also useful for mobile devices as the data set does not need to be transferred whole at once and the data transfer can be reduced to only first level of details. The additional levels of details can be transferred additionally; one by one when they are needed. Another use of the proposed ML-RBF vector field approximation is the exploration and the insight of the vector field as the vector field is visualized without excessive cluttering while maintaining the global structure of the vector field. Next, the compressed vector field is in the form of an analytical description which can be used for further vector field analysis and symbolic manipulation.

2. Radial Basis Functions

Radial basis function (RBF) is a real-valued function whose value depends only on some distances. i.e. the RBF interpolation [32] and approximation [13], [47] of scattered data is invariant under all Euclidean transformations. The RBF interpolation and approximation is widely used in many scientific disciplines, e.g. for solution of partial differential equations [26], [53], image reconstruction [49], neural networks [22], [16], [52], GIS systems [28], optics [33], vector fields approximation [46], [45], [41], [43], etc.

The RBF interpolation or approximation leads to a system of linear equations $\mathbf{Ax} = \mathbf{b}$ which is to be solved. It should be noted, that if the RBF is used for interpolation or approximation of data with large span, additional numerical problems can be expected [37], [39], [38].

There exist two groups of radial basis functions according to their influence. The first group are “global” RBFs [36]. The second group are “local” RBFs. The global RBF function application leads to ill-conditioned linear system of equations, in general, especially in the case of large data sets with a large span [29], [39]. The following global RBFs will be used in our experiments:

$$\begin{aligned}
 \text{Thin Plate Spline (TPS)} & \quad \varphi_1(r) = r^2 \log r \\
 \text{Gauss function} & \quad \varphi_2(r) = e^{-(\epsilon r)^2} \\
 \text{Inverse Quadric (IQ)} & \quad \varphi_3(r) = \frac{1}{1 + (\epsilon r)^2} \\
 \text{Inverse Multiquadric (IMQ)} & \quad \varphi_4(r) = \frac{1}{\sqrt{1 + (\epsilon r)^2}} \\
 \text{Multiquadric (MQ)} & \quad \varphi_5(r) = \sqrt{1 + (\epsilon r)^2}
 \end{aligned} \tag{1}$$

“Local” RBFs were introduced in [50] as compactly supported RBF (CSRBF). They satisfy the following condition:

$$\begin{aligned}
 \varphi(r) &= (1 - r)_+^q P(r) \\
 &= \begin{cases} (1 - r)^q P(r) & 0 \leq r \leq 1 \\ 0 & r > 1 \end{cases}
 \end{aligned} \tag{2}$$

where $P(r)$ is a polynomial function and q is a parameter. The following local RBFs will be used in our experiments:

$$\begin{aligned}
 \varphi_6(r) &= (1 - r)_+ \\
 \varphi_7(r) &= (1 - r)_+^3 (3r + 1) \\
 \varphi_8(r) &= (1 - r)_+^5 (8r^2 + 5r + 1) \\
 \varphi_9(r) &= (1 - r)_+^2 \\
 \varphi_{10}(r) &= (1 - r)_+^4 (4r + 1) \\
 \varphi_{11}(r) &= (1 - r)_+^6 (35r^2 + 18r + 3) \\
 \varphi_{12}(r) &= (1 - r)_+^8 (32r^3 + 25r^2 + 8r + 1)
 \end{aligned} \tag{3}$$

3. Proposed approach

In this section we describe our new proposed approach for multi-level vector field approximation using radial basis functions. The algorithm is composed by three main steps. The first step is the calculation of approximation error, the second one is the use of a Gaussian low-pass filter and the last one is the approximation using RBF. The pseudo-code of the proposed approach is in Algorithm 1. The algorithm is iterative and runs until the maximum level of details is computed.

In the following, we describe the proposed method on a $2D$ vector field. However, this algorithm is easy to extend to higher dimensions.

Algorithm 1 The multi-level RBF approximation of vector field.

```

1: vectorField  $\mathbf{v} = [\mathbf{v}_x, \mathbf{v}_y] = [\mathbf{0}, \mathbf{0}]$  ▷ Initialization
2:  $\sigma =$  initial value ▷ Standard deviation
3: procedure MULTI-LEVELRBF(Flow  $\bar{\mathbf{v}} = [\bar{v}_x, \bar{v}_y]$ )
4:   for  $i \leftarrow 1, LevelCount$  do
5:      $\mathbf{Err} = \bar{\mathbf{v}} - \mathbf{v};$  ▷ Error estimation
6:      $\mathbf{Err} = Gauss(\mathbf{Err}, \sigma)$  ▷ Gaussian low-pass filter
7:      $\mathbf{e} =$  Find extrema of  $\mathbf{Err}$ 
8:      $\mathbf{x}_0 =$  Find critical points of  $\mathbf{Err} + \mathbf{v}$ 
9:      $RBF =$  RBF approximation ( $\mathbf{e}, \mathbf{x}_0, \mathbf{Err}$ )
10:     $\mathbf{v} += RBF$  ▷ Update vector field
11:     $\sigma /= 2$  ▷ Decrease  $\sigma$ 
12:  end for

```

The first step of the proposed method is the error estimation. We need to compute the error for both components of a vector field

$$\begin{aligned} \mathbf{Err}_x &= \bar{v}_x - v_x, \\ \mathbf{Err}_y &= \bar{v}_y - v_y, \end{aligned} \tag{4}$$

where \mathbf{Err}_x and \mathbf{Err}_y are error vectors, \bar{v}_x and \bar{v}_y are the x and y components of input flow field vectors, v_x and v_y are the x and y components of actual flow field approximation vectors. For the 0 (zero) level vector field approximation, $v_x = \mathbf{0}$ and $v_y = \mathbf{0}$.

Vector fields can be very complex data sets with very large number of critical points. The multi-level RBF vector field approximation aims to approximate vector field in several levels of details. The lowest level of details should describe only the main global character of the flow. Each additional level of details should add some more details into the approximation. Thus with higher levels of details, the approximated vector field will contain more and more critical points and smaller flow details as well.

The next step of the proposed multi-level RBF vector field approximation is filtering the data set to obtain a simplified one. In our case, we filter the $2\frac{1}{2}D$ error data from (4), i.e. $2D$ function with associated errors. The low-pass Gaussian filter is used to filter the $2\frac{1}{2}D$ data. The Gaussian filter can have different scope and thus filter either small perturbations of the flow or large changes of the flow. The Gaussian filter has the formula

$$G(x, y) = \frac{1}{2\pi\sigma^2}g(x, y), \tag{5}$$

where x and y are the location coordinates, σ is the standard deviation of the Gaussian distribution and $g(x, y)$ is defined as

$$g(x, y) = e^{-\frac{x^2+y^2}{2\sigma^2}}. \quad (6)$$

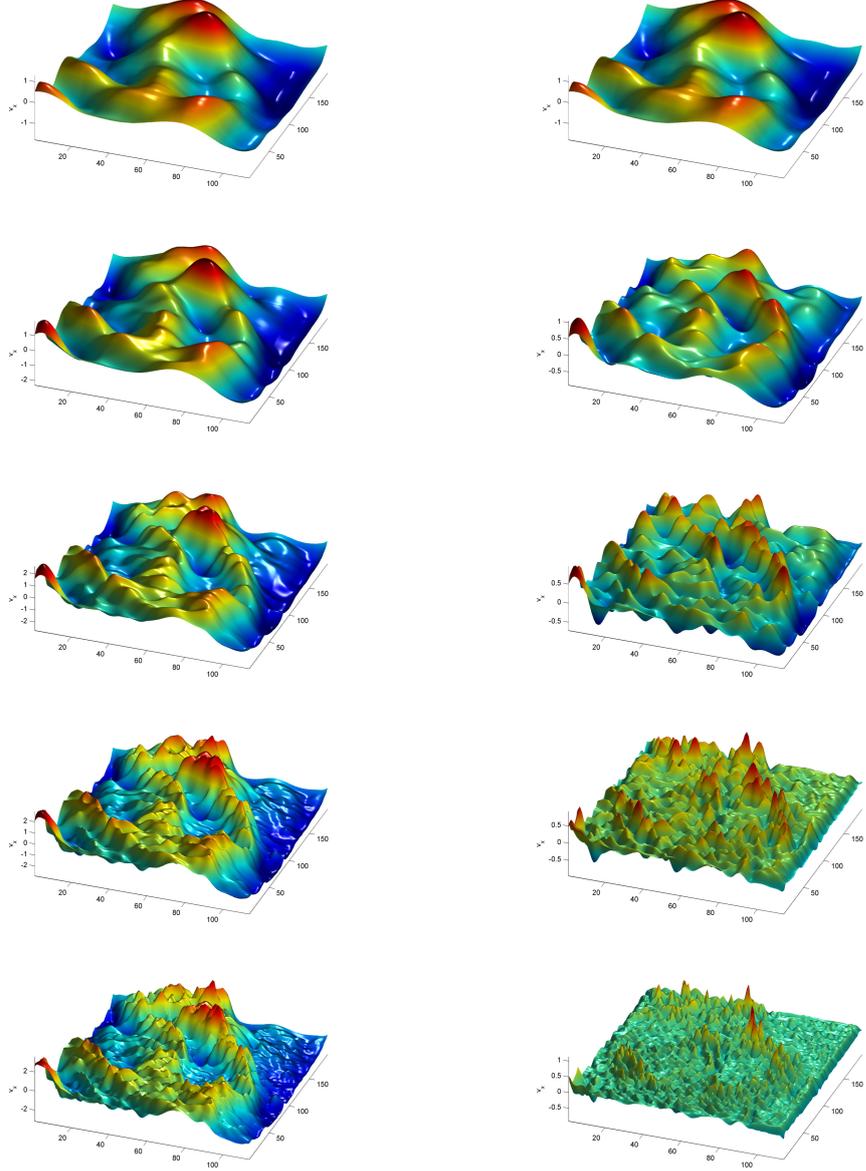


Figure 1. The right column represents Err_x of a vector field for different levels of details, i.e. from top to bottom: $\sigma = 10, 5, 2.5, 1.25$ and 0.625 . The left column represents the v_x component of a vector field for different levels of details, i.e. v_x is the sum of Err_x from the previous levels of details.

For each level of approximation, a different value of σ is to be chosen. For the first level of approximation we need to set up the initial value of σ . The value of σ in every

following level will always be half of the value σ from the previous level. The initial value of standard deviation, i.e. σ_1 , can be selected as

$$\sigma_1 = s_\sigma \cdot \text{MIN}(x_{\min} - x_{\max}, y_{\min} - y_{\max}), \quad (7)$$

where $x_{\min}, x_{\max}, y_{\min}, y_{\max}$ are minimal and maximal values of x and y coordinates, and s_σ is a constant reflecting the data size and sampling density. A pragmatical choice is $s_\sigma = \frac{1}{10}$. If this value s_σ is smaller, then the first level vector field approximation will be even more simplified, and conversely. An example of an application of a Gaussian low-pass filter on a vector field data set¹ [7] is in Figure 1.

The next step of the proposed method is the RBF approximation, for which we need the locations of radial basis functions, i.e. the centers. The centers need to be in the location of critical points, i.e.

$$\mathbf{x}_0 = \text{Find_critical_points}(\mathbf{Err} + \mathbf{v}). \quad (8)$$

Moreover at the extremes of \mathbf{v}_x , resp. \mathbf{v}_y , are located the additional centers of radial basis functions. The number of extremes will increase with increasing the level of approximation.

The radial basis function used for the RBF approximation is $\varphi_{10}(r)$ and it was used to demonstrate the proposed approach. It was selected due to continuity properties, computational complexity and it is the most adequate radial basis function according the tests in chapter 5.1. The RBF function $\varphi_{10}(r)$ is defined as

$$\varphi(r)_{10} = (1 - \epsilon r)_+^4 (4\epsilon r + 1), \quad (9)$$

where ϵ is the shape parameter of the radial basis function. The shape parameter is different for every level of approximation to capture different levels of details of the vector field. The shape parameter should be selected in a way that (9) has a similar shape as the Gaussian filter (6), i.e. the absolute difference of these two functions is minimal. We performed tests to select the best shape parameter, see Figure 6 in “Results” section. For different standard deviations σ in (6), the best shape parameter is

$$\epsilon = \frac{0.2694}{\sigma}. \quad (10)$$

Now, the RBF approximation is to be computed for each x and y component separately. To approximate \mathbf{Err}_x , the centers of radial basis functions will be locations of critical points from (8) and extremes of \mathbf{Err}_x , similarly for \mathbf{Err}_y .

After the RBF approximation, we need to update the actual level vector field

$$\begin{aligned} \mathbf{v}_x &= \mathbf{v}_x + \mathbf{RBF}_x, \\ \mathbf{v}_y &= \mathbf{v}_y + \mathbf{RBF}_y, \end{aligned} \quad (11)$$

The algorithm is repeated until the required number of levels of details is reached.

¹Data set of wind flow at a height of 10m over the surface of the Czech Republic courtesy of the Institute of Computer Science of the Czech Academy of Sciences.

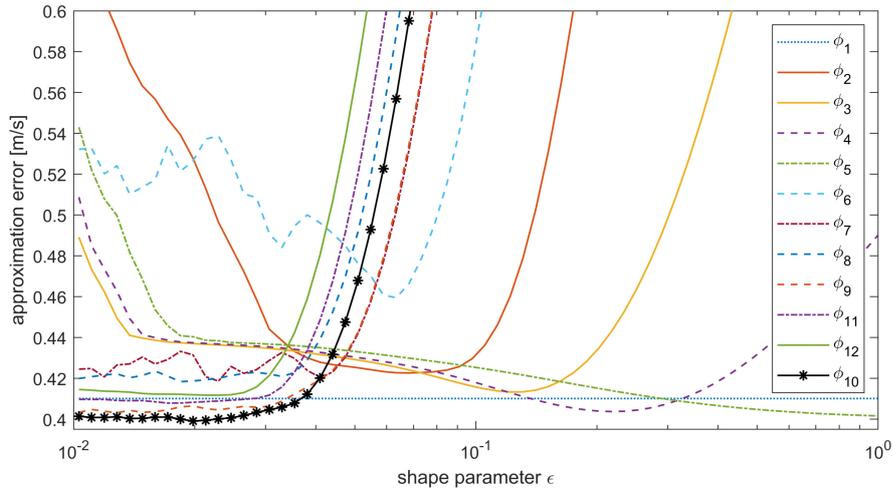


Figure 2. Approximation errors for different shape parameters.

With every additional level of details, the vector field RBF approximation is more accurate.

In the following chapter experimental results of the proposed multi-level RBF method are presented.

4. Results

The multi-level RBF (ML-RMB) vector field approximation is especially convenient for visualization purposes and vector field data understanding. The numerical forecast data set taken from [7] was used to prove the multi-level RBF approximation properties and for comparison with approximation based on the Fourier transform. The data set consists of around $2.2 \cdot 10^4$ points.

4.1. Selection of the RBF

One of the most important and critical part in the RBF approximation is the selection of the most adequate radial basis function [35], [29]. We tested the radial basis functions in (1) and (3). We selected around 275 centers of RBF, so that the compression ratio of the RBF approximation is 80 : 1. For each RBF in (1) and (3) we tested the approximation error for different shape parameters. The approximation error is computed using (16) and the results are presented in Figure 2. It can be seen that the “local” RBFs have all similar behavior and the approximation error is low when using lower values of shape parameter. In our case, we select a local RBF function, as the approximation matrix will be sparse and better conditioned. Also we will be able to approximate larger data sets compared to the case, when using “global” RBFs which leads to ill-conditioned full matrices, in general. The best choice according to the results in Figure 2 is the radial basis function $\varphi_{10}(r)$ as it has the lowest approximation error and is C_2 continuous.

We also compared the selected RBF $\varphi_{10}(r)$ with other RBFs and computed the dif-

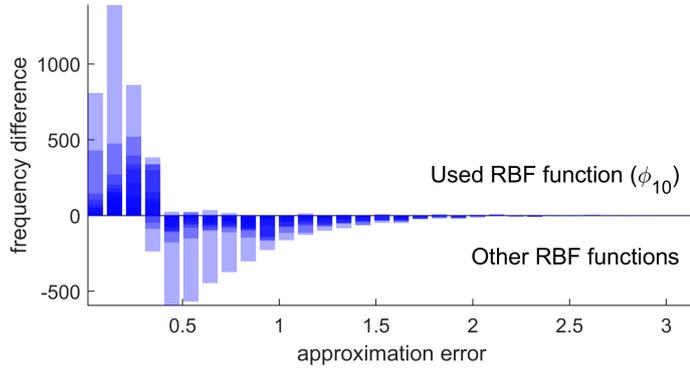


Figure 3. Difference histograms of approximation errors. All difference histograms are blended over each other.

ference histogram of approximation error, see Figure 3. The test was performed always for the best shape parameter for each RBF. The positive values for small approximation errors mean that the approximation with the selected RBF has much more smaller approximation errors. The negative values for larger approximation errors mean that the approximation with the selected RBF has much less larger approximation errors. This test clearly confirmed the selection of $\varphi_{10}(r)$ as the most adequate RBF for our proposed approach.

4.2. Multi-level RBF Approximation

To compute the RBF approximation, we need to find the centers of radial basis functions for every level of details. We tested the number of centers and summarized this in Table 1.

Table 1. The number of centers for RBF approximation at every added level of details.

Level number	σ	# of extreme points in \mathbf{v}_x	# of extreme points in \mathbf{v}_y	# of critical points
1	10	28	27	2
2	5	58	54	2
3	2.5	161	176	10
4	1.25	517	539	25
5	0.625	1238	1249	49

It can be seen that even when computation is done until the last level of details, we need for x and y components of the vector field approximately $2.1 \cdot 10^3$ centers of radial basis functions. This is approximately 9.5% of the input data set and the resulting vector field approximation is very similar to the original one.

At each level of RBF approximation, we approximate the $2\frac{1}{2}D$ functions \mathbf{Err}_x and \mathbf{Err}_y , see Figure 4a,d,g,j,m for \mathbf{Err}_x and Figure 5a,d,g,j,m for \mathbf{Err}_y . To find the location of radial basis functions, we use a Gauss filter for smoothing and then locate extremes of the resulting $2\frac{1}{2}D$ function, see Figure 4b,e,h,k,n for \mathbf{Err}_x smoothing and Figure 5b,e,h,k,n for \mathbf{Err}_y smoothing. These $2\frac{1}{2}D$ functions have the same global character as the original $2\frac{1}{2}D$ functions \mathbf{Err}_x and \mathbf{Err}_y , but they do not contain tiny details.

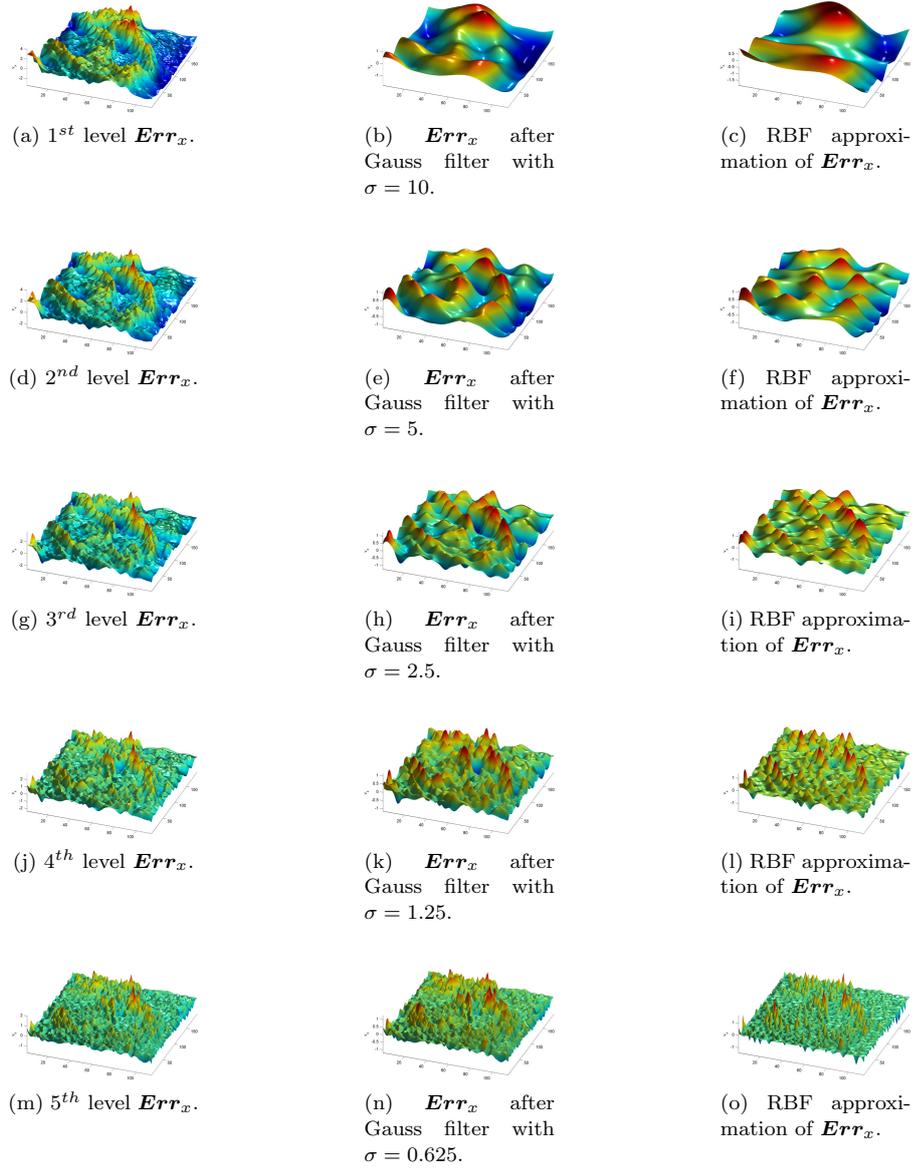


Figure 4. Err_x function for each level of details, filtered Err_x functions and the RBF approximation of Err_x .

The shape parameter is different for every level of approximation to capture different levels of details of the vector field. The shape parameter should be selected in a way that (9) has a similar shape as the Gaussian filter (6), i.e. the absolute difference of these two functions is minimal. We performed tests to select the best shape parameter. The results in Figure 6 are for a Gaussian filter with $\sigma = 1$.

The best shape parameter is $\epsilon = 0.2694$, see Figure 6. For different standard deviations σ , the best shape parameter is

$$\epsilon = \frac{0.2694}{\sigma}. \quad (12)$$

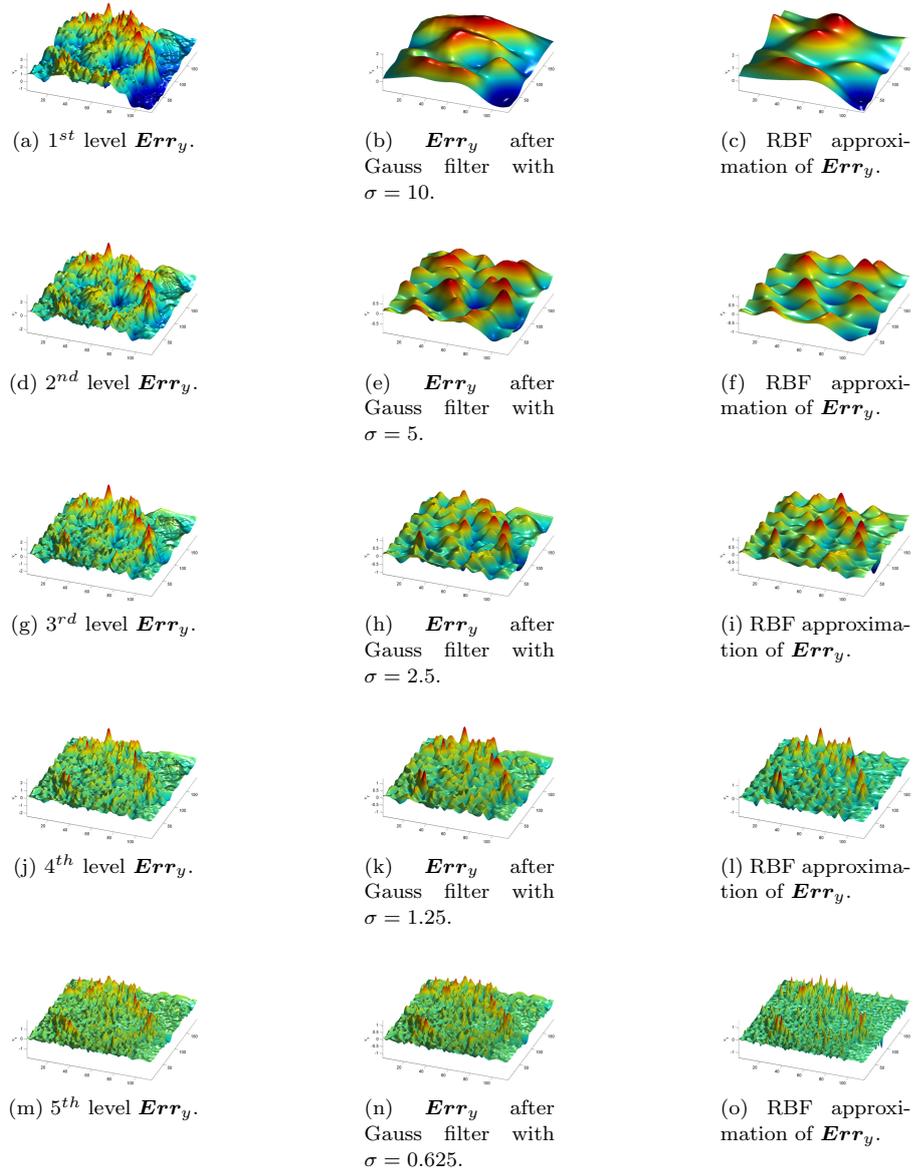


Figure 5. Err_y function for each level of details, filtered Err_y functions and the RBF approximation of Err_y .

The RBF approximations of Err_x are in Figure 4c,f,i,l,o and approximations of Err_y are in Figure 5c,f,i,l,o. The RBF approximations of Err_x and Err_y are very close to the filtered $2\frac{1}{2}D$ functions of Err_x and Err_y , as the placement of radial basis function centers and the shape parameter of radial basis functions are very well chosen.

The resulting multi-level RBF approximation of the vector field is visualized in Figure 7. It can be seen that with every additional level of details the approximated vector field is closer to the original one. Even the first level approximation has the same global characteristics as the original vector field.

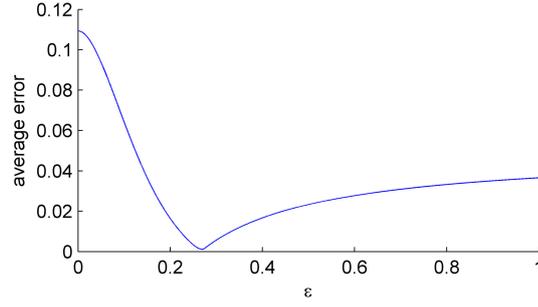


Figure 6. The average absolute difference of (5) with $\sigma = 1$ and (9) for different values of shape parameter ϵ and $r \in [0; 1/\epsilon)$.

To measure the quality of the vector field approximation, we compute the approximation error at every point of the vector field using the following formula

$$error^{(i)} = \sqrt{\left(v_x^{(i)} - \bar{v}_x^{(i)}\right)^2 + \left(v_y^{(i)} - \bar{v}_y^{(i)}\right)^2}, \quad (13)$$

where $v_x^{(i)}$ and $v_y^{(i)}$ are approximated values, $\bar{v}_x^{(i)}$ and $\bar{v}_y^{(i)}$ are the original values of the vector field and $error^{(i)}$ is the approximation error at the i^{th} point. The approximation error is color-coded in Figure 8. It can be seen that the approximation error is lower with every additional level of details.

4.3. Comparison with Existing Approach

The proposed method needs to be compared with an other existing approach. We selected the mostly used Fourier transform [14], [2]. The vector field is approximated with the Fourier transform using the following formula

$$F(\alpha, \beta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} v(x, y) e^{-2\pi i(x\alpha + y\beta)} dx dy, \quad (14)$$

where α and β are frequencies (see Figure 9). According to the required accuracy, only some of the most important frequencies are selected to approximate the vector field. This approximated vector field represented by a list of frequencies can be transformed back to the vector field using the following inverse Fourier transform

$$v(x, y) = \frac{1}{(2\pi)^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(\alpha, \beta) e^{2\pi i(x\alpha + y\beta)} d\alpha d\beta. \quad (15)$$

To compare the proposed approach for multi-level vector field approximation with the Fourier transform vector field approximation, we need to compute and compare the vector field approximation errors. We can compute the average difference approximation error using the following formula

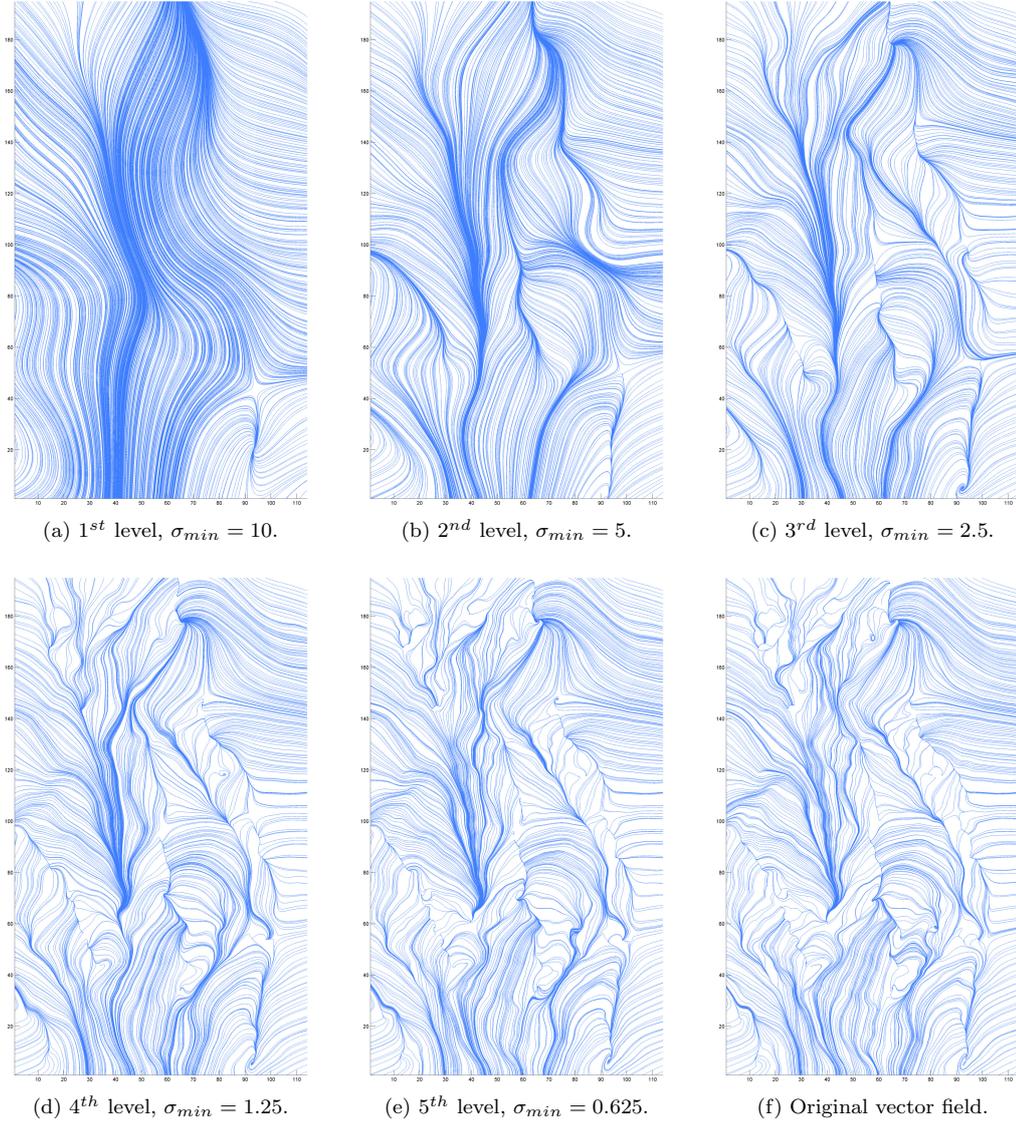


Figure 7. The vector field approximation for different levels of details (a-e) and the original vector field (f).

$$Err = \frac{\sum_{i=1}^N \|\mathbf{v}_i - \bar{\mathbf{v}}_i\|}{N}, \quad (16)$$

where \mathbf{v}_i is the approximated vector, $\bar{\mathbf{v}}_i$ is the original vector and N is the number of points with associated vectors in the original dataset. This error shows how much the approximated vector field differs from the original one. The resulting error has the

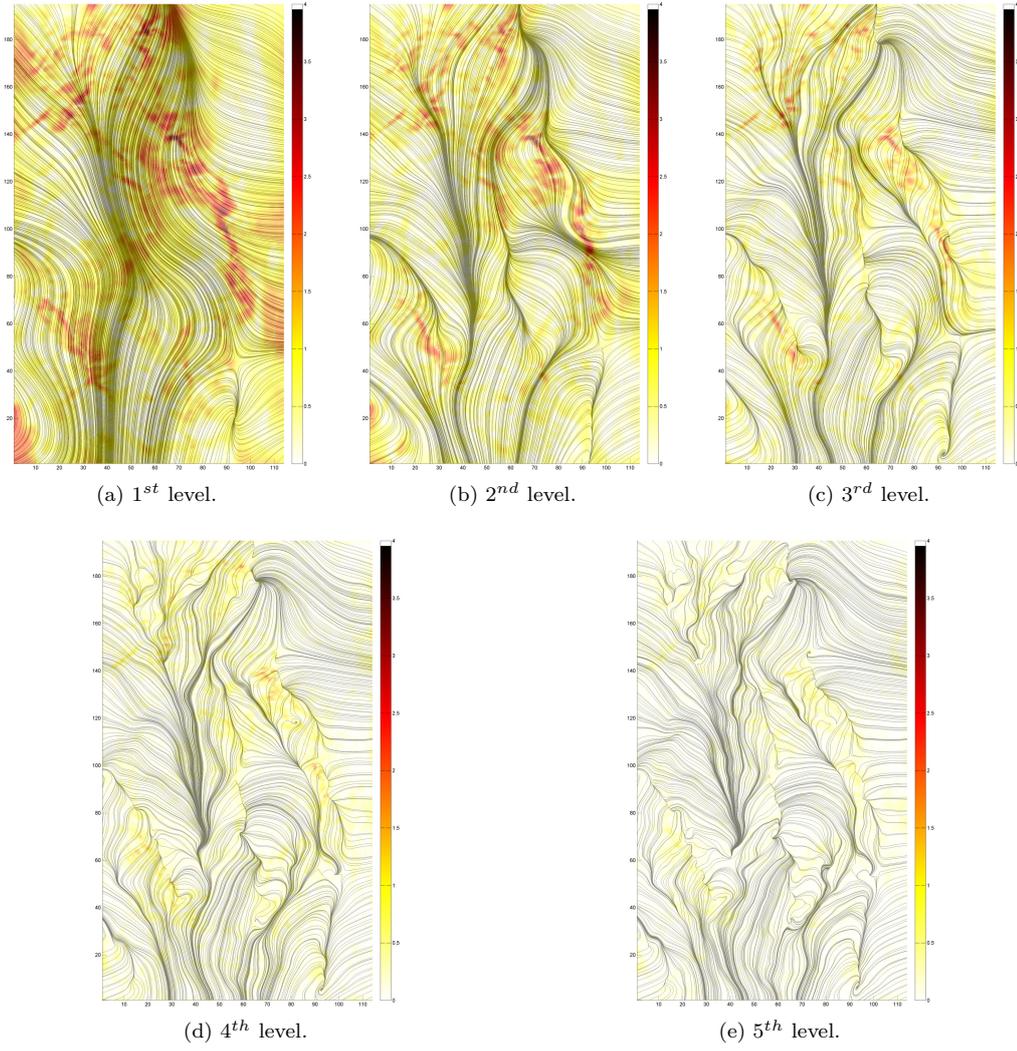


Figure 8. The vector field approximation error for different levels of details. The approximation error is computed using (13). All color bars have the same error range for better comparison.

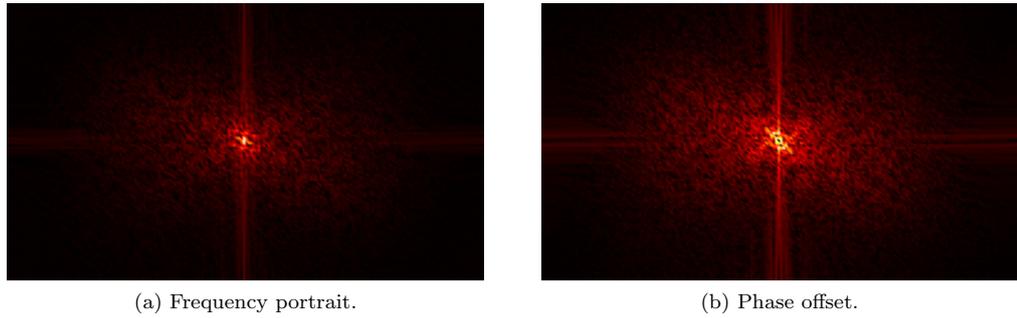
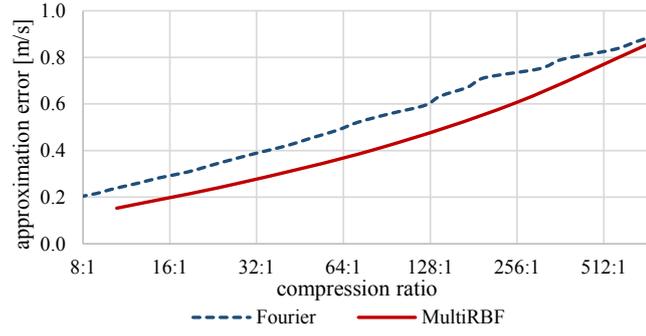


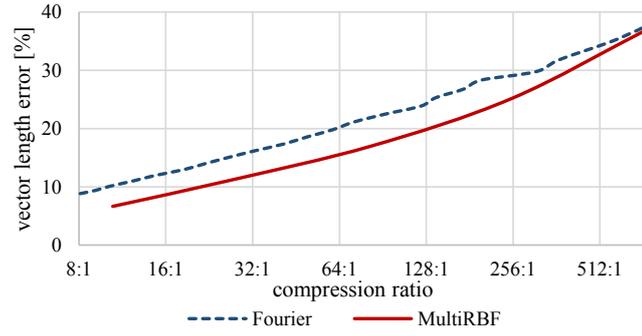
Figure 9. Approximation of the vector field using Fourier transform.

same units as the vector field. Next, we can compute another kind of approximation error. We can compute the average vector length error using the following formula

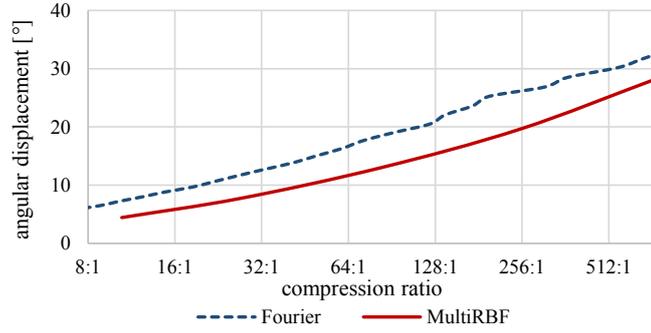
$$Err = \frac{\sum_{i=1}^N \|\mathbf{v}_i\| - \|\bar{\mathbf{v}}_i\|}{\sum_{i=1}^N \|\bar{\mathbf{v}}_i\|}. \quad (17)$$



(a) The average difference approximation error.



(b) The average vector length error.



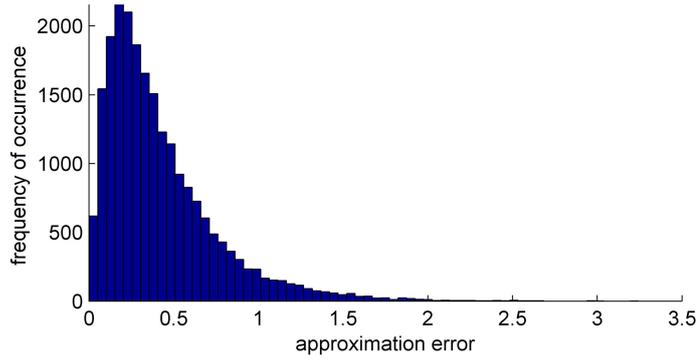
(c) The average angular displacement error.

Figure 10. Visualization of approximation errors for different compression ratios.

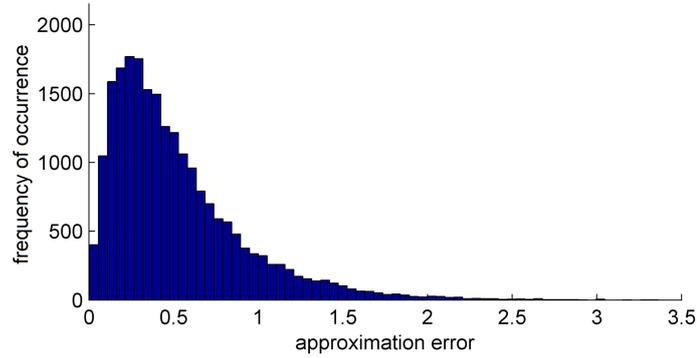
This formula computes some kind of relative vector length error. The standard formula for relative vector length error is

$$Err = \frac{1}{N} \sum_{i=1}^N \frac{|\|\mathbf{v}_i\| - \|\bar{\mathbf{v}}_i\||}{\|\bar{\mathbf{v}}_i\|}. \quad (18)$$

However, using this formula will give us incorrect result because of division by numbers close to zero or even equal to zero. For this reason we use (17) instead of the standard (18).



(a) Multilevel RBF approximation.



(b) Fourier approximation.

Figure 11. Histograms of vector field approximation error for compression ratio 80 : 1.

Last we will compute one more kind of approximation error, namely, the average angular displacement error. This error is computed using the following formula

$$Err = \frac{\sum_{i=1}^N \text{acos} \left(\frac{\mathbf{v}_i \cdot \bar{\mathbf{v}}_i}{\|\mathbf{v}_i\| \|\bar{\mathbf{v}}_i\|} \right)}{N}. \quad (19)$$

We compared our proposed method with the Fourier method using three types of approximation errors for different compression ratios. Computed approximation errors for our proposed multi-level vector field approximation and for the Fourier vector field approximation are visualized in Figure 10. It can be seen, that for all three approximation errors computations, the proposed method has lower approximation error for all compression ratios. To analyze the approximation error more closely, we selected compression ratio 80 : 1 and computed histograms of approximation error using (13), see Figure 11. It can be seen that most approximation errors are low and only a few errors are high.

Moreover to compare the two methods for vector field approximation, we computed the difference histogram, see Figure 12. It can be seen, that the proposed method has much more lower approximation errors and much less higher approximation errors than the Fourier approximation method. The experiments made also proved similar behavior for other compression ratios.

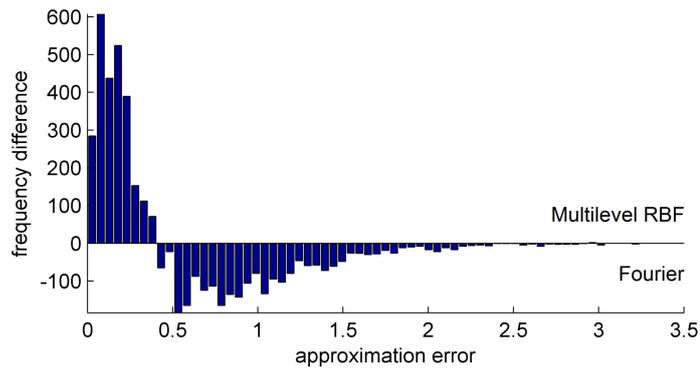


Figure 12. Difference histograms of approximation errors (“Multilevel RBF” - “Fourier”), see Figure 11, for compression ratio 80 : 1.

5. Conclusion

We proposed a new approach for multi-level vector field approximation. The vector field is approximated in several levels of details, where each level of details adds some additional information and refine the vector field approximation. The approach uses Radial basis function for approximation of vector field. The centers of radial basis functions are placed according to the distribution of approximation error of the previous level of detail vector field approximation. The proposed approach is especially convenient for approximation and visualization of large and complex data sets, i.e. only needed levels of details of the vector field can be transferred and visualized on the devices (mobile phone, web browser, etc.) with high compression ratio. Another advantage over existing approaches is the final analytical description of the approximated vector field, which can be used for further processing.

In the future, the proposed approach for multi-level vector field approximation will be extended to approximate the $3D$ vector fields as well. This extension should be straightforward and easy to implement.

Acknowledgment

The authors express thanks to their colleagues at the University of West Bohemia, Plzen, for their discussions, suggestions and hints; colleagues at the Institute of Computer Science, the Czech Academy of Sciences, for their constructive comments and for providing testing data sets.

This research was supported by projects Czech Science Foundation (GACR) No. GA 17-05534S and partially by SGS 2019-016.

References

- [1] J. Blazek, *Computational fluid dynamics: principles and applications*, Butterworth-Heinemann, 2015.
- [2] R.N. Bracewell and R.N. Bracewell, *The Fourier transform and its applications*, Vol. 31999, McGraw-Hill New York, 1986.

- [3] T. Bülow, M. Felsberg, and G. Sommer, *Non-commutative hypercomplex Fourier transforms of multidimensional signals*, in *Geometric computing with Clifford algebras*, Springer, 2001, pp. 187–207.
- [4] R. Bürger, R. Ruiz, K. Schneider, and M.A. Sepúlveda, *Fully adaptive multiresolution schemes for strongly degenerate parabolic equations with discontinuous flux*, *Journal of Engineering Mathematics* 60 (2008), pp. 365–385.
- [5] 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 (2013), pp. 163–173.
- [6] E.T. Chung, Y. Efendiev, and C.S. Lee, *Mixed generalized multiscale finite element methods and applications*, *Multiscale Modeling & Simulation* 13 (2015), pp. 338–366.
- [7] L. Corbet, O. Vlček, K. Eben, J. Liczki, N. Benešová, and M. Modlík, *Regional air quality forecasting for the Czech Republic*, in *15th International Conference on Harmonisation within Atmospheric Dispersion Modelling for Regulatory Purposes, HARMO 2013*. 2013, pp. 150–154.
- [8] P.J. Davis, *Interpolation and approximation*, Courier Corporation, 1975.
- [9] W. de Leeuw and R. van Liere, *Multi-level topology for flow visualization*, *Computers & Graphics* 24 (2000), pp. 325–331.
- [10] W. Dou, L. Zhang, G. Chen, and W. Zhu, *A combined radial basis function based interpolation method for fluid-structure interaction problems and its application on high-speed trains*, *Advances in Engineering Software* (2019).
- [11] J. Ebling and G. Scheuermann, *Clifford Fourier transform on vector fields*, *IEEE Transactions on Visualization and Computer Graphics* 11 (2005), pp. 469–479.
- [12] T.A. Ell and S.J. Sangwine, *Hypercomplex Fourier transforms of color images*, *IEEE Transactions on image processing* 16 (2007), pp. 22–35.
- [13] G.E. Fasshauer, *Meshfree approximation methods with MATLAB*, Vol. 6, World Scientific, 2007.
- [14] J. Fourier, *Theorie analytique de la chaleur, par M. Fourier*, Chez Firmin Didot, père et fils, 1822.
- [15] F. Gagliardi and K.C. Giannakoglou, *A two-step radial basis function-based cfd mesh displacement tool*, *Advances in Engineering Software* 128 (2019), pp. 86–97.
- [16] S. Ghosh-Dastidar, H. Adeli, and N. Dadmehr, *Principal component analysis-enhanced cosine radial basis function neural network for robust epilepsy and seizure detection*, *IEEE Transactions on Biomedical Engineering* 55 (2008), pp. 512–518.
- [17] D. Günther, A. Jacobson, J. Reininghaus, H.P. Seidel, O. Sorkine-Hornung, and T. Weinkauff, *Fast and memory-efficiently topological denoising of 2d and 3d scalar fields*, *IEEE transactions on visualization and computer graphics* 20 (2014), pp. 2585–2594.
- [18] J. Helman and L. Hesselink, *Representation and display of vector field topology in fluid flow data sets*, *IEEE Computer* 22 (1989), pp. 27–36.
- [19] E.M. Hitzer and B. Mawardi, *Clifford Fourier transform on multivector fields and uncertainty principles for dimensions $n=2 \pmod{4}$ and $n=3 \pmod{4}$* , *Advances in Applied Clifford Algebras* 18 (2008), pp. 715–736.
- [20] E.J. Kansa, *Multiquadrics - A scattered data approximation scheme with applications to computational fluid-dynamics—i surface approximations and partial derivative estimates*, *Computers & Mathematics with applications* 19 (1990), pp. 127–145.
- [21] E.J. Kansa, *Multiquadrics - A scattered data approximation scheme with applications to computational fluid-dynamics—ii solutions to parabolic, hyperbolic and elliptic partial differential equations*, *Computers & mathematics with applications* 19 (1990), pp. 147–161.
- [22] A. Karim and H. Adeli, *Radial basis function neural network for work zone capacity and queue estimation*, *Journal of Transportation Engineering* 129 (2003), pp. 494–503.
- [23] J. Kim, *Phase-field models for multi-component fluid flows*, *Communications in Computational Physics* 12 (2012), pp. 613–661.

- [24] S. Koch, J. Kasten, A. Wiebel, G. Scheuermann, and M. Hlawitschka, *2d vector field approximation using linear neighborhoods*, *The Visual Computer* 32 (2016), pp. 1563–1578.
- [25] 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*, Springer, 2007, pp. 1–19.
- [26] E. Larsson and B. Fornberg, *A numerical study of some radial basis function based solution methods for elliptic PDEs*, *Computers & Mathematics with Applications* 46 (2003), pp. 891–902.
- [27] H.G. Lee and J. Kim, *A comparison study of the boussinesq and the variable density models on buoyancy-driven flows*, *Journal of Engineering Mathematics* 75 (2012), pp. 15–27.
- [28] Z. Majdisova and V. Skala, *Big geo data surface approximation using radial basis functions: A comparative study*, *Computers & Geosciences* 109 (2017), pp. 51–58.
- [29] Z. Majdisova and V. Skala, *Radial basis function approximations: Comparison and applications*, *Applied Mathematical Modelling* 51 (2017), pp. 728–743.
- [30] H. Mohammadzade and L.T. Bruton, *A simultaneous div-curl 2D Clifford Fourier transform filter for enhancing vortices, sinks and sources in sampled 2D vector field images*, in *Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on*. IEEE, 2007, pp. 821–824.
- [31] F. Molina-Aiz, H. Fatnassi, T. Boulard, J. Roy, and D. Valera, *Comparison of finite element and finite volume methods for simulation of natural ventilation in greenhouses*, *Computers and electronics in agriculture* 72 (2010), pp. 69–86.
- [32] R. Pan and V. Skala, *A two-level approach to implicit surface modeling with compactly supported radial basis functions*, *Engineering with Computers* 27 (2011), pp. 299–307.
- [33] G. Prakash, M. Kulkarni, and U. Sripathi, *Using RBF Neural Networks and Kullback-Leibler distance to classify channel models in Free Space Optics*, in *Optical Engineering (ICOE), 2012 International Conference on*. IEEE, 2012, pp. 1–6.
- [34] W. Reich and G. Scheuermann, *Analyzing real vector fields with clifford convolution and clifford-fourier transform*, in *Geometric Algebra Computing*, Springer, 2010, pp. 121–133.
- [35] H. Rocha, *On the selection of the most adequate radial basis function*, *Applied Mathematical Modelling* 33 (2009), pp. 1573–1583.
- [36] I.P. Schagen, *Interpolation in two dimensions - a new technique*, *IMA Journal of Applied Mathematics* 23 (1979), pp. 53–59.
- [37] V. Skala, *High dimensional and large span data least square error: Numerical stability and conditionality*, *International Journal of Applied Physics and Mathematics* 7 (2017), pp. 148–156.
- [38] V. Skala, *Least Square Method Robustness of Computations: What is not usually considered and taught*, in *Proceedings of the 2017 Federated Conference on Computer Science and Information Systems FedCSIS*, *Annals of Computer Science and Information Systems* Vol. 11. IEEE, 2017, pp. 537–541.
- [39] V. Skala, *RBF interpolation with CSRBF of large data sets*, *Procedia Computer Science (ICCS)* 108 (2017), pp. 2433–2437.
- [40] P. Skraba, B. Wang, G. Chen, and P. Rosen, *2D vector field simplification based on robustness*, in *Visualization Symposium (PacificVis), 2014 IEEE Pacific*. IEEE, 2014, pp. 49–56.
- [41] M. Smolik and V. Skala, *Vector field interpolation with radial basis functions*, in *Proceedings of SIGRAD 2016, May 23rd and 24th, Visby, Sweden*. Linköping University Electronic Press, 127, 2016, pp. 15–21.
- [42] M. Smolik and V. Skala, *Classification of critical points using a second order derivative*, *Procedia Computer Science (ICCS)* 108 (2017), pp. 2373–2377.
- [43] M. Smolik and V. Skala, *Spherical RBF vector field interpolation: experimental study*, in *Applied Machine Intelligence and Informatics (SAMII), 2017 IEEE 15th International Symposium on*. IEEE, 2017, pp. 431–434.

- [44] M. Smolik and V. Skala, *Vector Field Second Order Derivative Approximation and Geometrical Characteristics*, in *International Conference on Computational Science and Its Applications (ICCSA)*. Springer, 2017, pp. 148–158.
- [45] 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 (APSAC)*. Springer, 2018, pp. 1–6.
- [46] M. Smolik, V. Skala, and Z. Majdisova, *Vector field radial basis function approximation*, *Advances in Engineering Software* 123 (2018), pp. 117–129.
- [47] M. Smolik, V. Skala, and O. Nedved, *A Comparative Study of LOWESS and RBF Approximations for Visualization*, in *International Conference on Computational Science and Its Applications (ICCSA)*. Springer, 2016, pp. 405–419.
- [48] J.I. Toivanen, R.A. Mäkinen, and J. Haslinger, *Topology optimization in bernoulli free boundary problems*, *Journal of Engineering Mathematics* 80 (2013), pp. 173–188.
- [49] K. Uhlir and V. Skala, *Reconstruction of damaged images using radial basis functions*, in *Signal Processing Conference, 2005 13th European*. IEEE, 2005, pp. 1–4.
- [50] H. Wendland, *Computational aspects of radial basis function approximation*, *Studies in Computational Mathematics* 12 (2006), pp. 231–256.
- [51] H. Woollard, J. Billingham, O. Jensen, and G. Lian, *A multi-scale model for solute transport in a wavy-walled channel*, *Journal of Engineering Mathematics* 64 (2009), p. 25.
- [52] L. Yingwei, N. Sundararajan, and P. Saratchandran, *Performance evaluation of a sequential minimal radial basis function (RBF) neural network learning algorithm*, *IEEE Transactions on neural networks* 9 (1998), pp. 308–318.
- [53] X. Zhang, K.Z. Song, M.W. Lu, and X. Liu, *Meshless methods based on collocation with radial basis functions*, *Computational mechanics* 26 (2000), pp. 333–343.

4.7 Classification of critical points using a second order derivative

The vector field classification is usually made using the Jacobian matrix, i.e. there are distinguished saddle points, sinks, sources, etc. However, the classification of critical points can be done even more precisely. The paper [Smolik and Skala, 2017a] presents an approach for classification of critical points using the second order derivative.

The second order derivative approximation can be represented using the Hessian matrix. This matrix can be further use to determine the curvature of the main axes of critical points. It can be used to determine the shape of the vector field and to classify it more closely.

Citation:

- Michal Smolik and Vaclav Skala. Classification of critical points using a second order derivative. *Procedia Computer Science*, 108:2373–2377, 2017.

Classification of Critical Points Using a Second Order Derivative

Michal Smolik¹ and Vaclav Skala¹

¹*University of West Bohemia, Plzen, Czech Republic*
{smolik, skala}@kiv.zcu.cz

Abstract

This article presents a new method for classification of critical points. A vector field is usually classified using only a Jacobian matrix of the approximated vector field. This work shows how an approximation using a second order derivative can be used for more detailed classification. An algorithm for calculation of the curvature of main axes is also presented.

Keywords: vector field; critical point; Hessian matrix; curvature

1 Introduction

The visualization of vector field topology is a problem that arises naturally when studying the qualitative structure of flows. The knowledge of the data at a single point would be of little help when the goal is to obtain knowledge and understanding of the whole vector field. The individual numbers can be of little interest. It is the connection between them which is important.

Helman and Hesselink [1] introduced the concept of the topology of a planar vector field to the visualization community. They extracted critical points and classified them into sources, sinks and saddles, and integrated certain stream lines called separatrices from the saddles in the directions of the eigenvectors of the Jacobian matrix. Later, topological methods have been extended to higher order critical points [6], boundary switch points [3], and closed separatrices [14]. In addition, topological methods using classification have been applied to simplify [7], smooth and compress [2] vector fields.

Theisel [10] presents a summary of vector field curvatures. Weinkauff and Theisel [12] present the theory of curvature measures of 3D vector fields. The curvature measurements are used to measure the distance between streamlines in vector fields [4].

All of the published research uses for classification of critical points and vector field description only linear approximation of the vector field. None of it uses an approximation with second order partial derivatives, i.e. a Hessian matrix. This approximation gives a more detailed description of the vector field around a critical point and can be used for a more detailed classification. Use of an approximation with a Hessian matrix will be described in this paper.

2 Approximation Using a Hessian Matrix

Vector fields are approximated using only linear approximation to determine the local behavior of the vector field [1]. However, linearization gives us only a basic classification of the critical points and basic information about the flow around them; the approximation using second order derivatives will give us some more information.

An approximation of the vector field around a critical point using a second order derivative must be written for each vector component (v_x and v_y) separately; see the following equation for v_x :

$$v_x = \begin{bmatrix} \frac{\partial v_x}{\partial x} \\ \frac{\partial v_x}{\partial y} \end{bmatrix}^T \cdot \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} + \frac{1}{2} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}^T \cdot \begin{bmatrix} \frac{\partial^2 v_x}{\partial x^2} & \frac{\partial^2 v_x}{\partial x \partial y} \\ \frac{\partial^2 v_x}{\partial y \partial x} & \frac{\partial^2 v_x}{\partial y^2} \end{bmatrix} \cdot \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \mathbf{J}_x \cdot \Delta \mathbf{x} + \frac{1}{2} \Delta \mathbf{x}^T \cdot \mathbf{H}_x \cdot \Delta \mathbf{x} \quad (1)$$

where $\Delta x = x - x_0$, $\Delta y = y - y_0$, $\Delta \mathbf{x} = [\Delta x, \Delta y]^T$, \mathbf{H}_x is Hessian matrix, \mathbf{J}_x is the first row of a Jacobian matrix. Equation for v_y is similar as for v_x .

A Hessian matrix is a square matrix of second-order partial derivatives of a scalar-valued function, or scalar field. It describes the local curvature of a function of many variables.

An approximation of a vector field using (1) is a bit more computationally expensive than the standard linear approximation, but gives us a more detailed description than a linear approximation of a vector field, see Fig. 2. The approximation in Fig. 2 ($t \neq 0$) gives us the same information as in Fig. 2 ($t = 0$), although we can see the curvature of the two main axes for the saddle.

3 Curvature of a Vector Field

An approximated vector field using (1) is not only linear but contains the Hessian matrices that describe the local curvature of the vector field. In this section, an approach for computing the local curvature of a vector field that is approximated with Jacobian and Hessian matrices is introduced.

Using an approximation of the vector field with second order derivatives gives us the opportunity to compute a Jacobian matrix \mathbf{J}_ε in the neighborhood of a critical point from approximated vector field (1):

$$\mathbf{J}_\varepsilon = \begin{bmatrix} \left. \frac{\partial v_x}{\partial x} \right|_{x_0+\varepsilon} & \left. \frac{\partial v_x}{\partial y} \right|_{x_0+\varepsilon} \\ \left. \frac{\partial v_y}{\partial x} \right|_{x_0+\varepsilon} & \left. \frac{\partial v_y}{\partial y} \right|_{x_0+\varepsilon} \end{bmatrix}, \quad (2)$$

where $\boldsymbol{\varepsilon} = [e_x, e_y]^T$ is an arbitrary direction vector pointing from the critical point \mathbf{x}_0 . The matrix \mathbf{J}_ε (2×2) in (2) can be rewritten using elements of \mathbf{J} , \mathbf{H}_x (1) and \mathbf{H}_y as:

$$\begin{bmatrix} \left. \frac{\partial v_x}{\partial x} \right|_{x_0} + \frac{\partial^2 v_x}{\partial x^2} \Big|_{x_0} \varepsilon_x + \frac{1}{2} \left(\left. \frac{\partial^2 v_x}{\partial x \partial y} \right|_{x_0} + \left. \frac{\partial^2 v_x}{\partial y \partial x} \right|_{x_0} \right) \varepsilon_y & \left. \frac{\partial v_x}{\partial y} \right|_{x_0} + \frac{\partial^2 v_x}{\partial y^2} \Big|_{x_0} \varepsilon_x + \frac{1}{2} \left(\left. \frac{\partial^2 v_x}{\partial x \partial y} \right|_{x_0} + \left. \frac{\partial^2 v_x}{\partial y \partial x} \right|_{x_0} \right) \varepsilon_x \\ \left. \frac{\partial v_y}{\partial x} \right|_{x_0} + \frac{\partial^2 v_y}{\partial x^2} \Big|_{x_0} \varepsilon_x + \frac{1}{2} \left(\left. \frac{\partial^2 v_y}{\partial x \partial y} \right|_{x_0} + \left. \frac{\partial^2 v_y}{\partial y \partial x} \right|_{x_0} \right) \varepsilon_y & \left. \frac{\partial v_y}{\partial y} \right|_{x_0} + \frac{\partial^2 v_y}{\partial y^2} \Big|_{x_0} \varepsilon_x + \frac{1}{2} \left(\left. \frac{\partial^2 v_y}{\partial x \partial y} \right|_{x_0} + \left. \frac{\partial^2 v_y}{\partial y \partial x} \right|_{x_0} \right) \varepsilon_x \end{bmatrix}. \quad (3)$$

Assuming that \mathbf{v} has continuous second partial derivatives at any given point, the mixed derivatives of v_x and v_y in the Hessian matrix are commutative. Equation (3) can be simplified and rewritten as

$$\mathbf{J}_\varepsilon = \begin{bmatrix} \frac{\partial v_x}{\partial x} \Big|_{x_0} & \frac{\partial v_x}{\partial y} \Big|_{x_0} \\ \frac{\partial v_y}{\partial x} \Big|_{x_0} & \frac{\partial v_y}{\partial y} \Big|_{x_0} \end{bmatrix} + \begin{bmatrix} \frac{\partial^2 v_x}{\partial x^2} \Big|_{x_0} & \frac{\partial^2 v_x}{\partial x \partial y} \Big|_{x_0} \\ \frac{\partial^2 v_y}{\partial x^2} \Big|_{x_0} & \frac{\partial^2 v_y}{\partial x \partial y} \Big|_{x_0} \end{bmatrix} \varepsilon_x + \begin{bmatrix} \frac{\partial^2 v_x}{\partial x \partial y} \Big|_{x_0} & \frac{\partial^2 v_x}{\partial y^2} \Big|_{x_0} \\ \frac{\partial^2 v_y}{\partial x \partial y} \Big|_{x_0} & \frac{\partial^2 v_y}{\partial y^2} \Big|_{x_0} \end{bmatrix} \varepsilon_y \quad (4)$$

Elements of \mathbf{J} , \mathbf{H}_x (1) and \mathbf{H}_y are all elements used to calculate \mathbf{J}_ε in (3) and (4), so there is no need to compute any additional derivatives than those in (1). Note that the Jacobian matrix \mathbf{J}_ε is for $\varepsilon = [0,0]^T$ equal to Jacobian matrix \mathbf{J} . The Jacobian matrix \mathbf{J}_ε can be computed for any point $(\mathbf{x}_0 + \varepsilon)$. Therefore, we start by computing the eigenvectors of Jacobian matrix \mathbf{J} in a critical point \mathbf{x}_0 .

There are two eigenvectors (\mathbf{u}_1 and \mathbf{u}_2) for a Jacobian matrix in $2D$. In the case that the vector field is circular around the critical point, we will use only the real part of the eigenvectors, i.e.

$$\begin{aligned} \operatorname{Re}(a + bi) &= a \\ \operatorname{Im}(a + bi) &= b \end{aligned} \quad a, b \in \mathbb{R}. \quad (5)$$

To calculate the curvature of the vector field we need to compute the eigenvectors in the near surroundings of the critical point as will be explained later. First, we need to compute vectors pointing from \mathbf{x}_0 in the direction of the main axes of the vector field, i.e.

$$\boldsymbol{\varepsilon}_{1L} = -\frac{\operatorname{Re}(\mathbf{u}_1)}{\|\operatorname{Re}(\mathbf{u}_1)\|} h \quad \boldsymbol{\varepsilon}_{2L} = -\frac{\operatorname{Re}(\mathbf{u}_2)}{\|\operatorname{Re}(\mathbf{u}_2)\|} h \quad \boldsymbol{\varepsilon}_{1R} = \frac{\operatorname{Re}(\mathbf{u}_1)}{\|\operatorname{Re}(\mathbf{u}_1)\|} h \quad \boldsymbol{\varepsilon}_{2R} = \frac{\operatorname{Re}(\mathbf{u}_2)}{\|\operatorname{Re}(\mathbf{u}_2)\|} h, \quad (6)$$

where h is some small number (e.g. $h = 10^{-3}$ for the vector field in Fig. 2).

In the next step, we calculate Jacobian matrix \mathbf{J}_ε for all vectors computed in (6), i.e. the Jacobian matrix at points $(\mathbf{x}_0 + \boldsymbol{\varepsilon}_{**})$:

$$\mathbf{J}_{1L} = \mathbf{J}_\varepsilon(\boldsymbol{\varepsilon}_{1L}) \quad \mathbf{J}_{2L} = \mathbf{J}_\varepsilon(\boldsymbol{\varepsilon}_{2L}) \quad \mathbf{J}_{1R} = \mathbf{J}_\varepsilon(\boldsymbol{\varepsilon}_{1R}) \quad \mathbf{J}_{2R} = \mathbf{J}_\varepsilon(\boldsymbol{\varepsilon}_{2R}). \quad (7)$$

For each Jacobian matrix in (7) we need to calculate the real parts of both eigenvectors and determine which one is pointing in almost the same direction as the original eigenvector $\operatorname{Re}(\mathbf{u}_1)$ for \mathbf{J}_{1L} and \mathbf{J}_{1R} and determine \mathbf{u}_{1L} and \mathbf{u}_{1R} , respectively; similarly eigenvector $\operatorname{Re}(\mathbf{u}_2)$ for \mathbf{J}_{2L} and \mathbf{J}_{2R} and determine \mathbf{u}_{2L} and \mathbf{u}_{2R} . This test can be done using the dot product between original eigenvector $\operatorname{Re}(\mathbf{u}_i)$, where $i = \{1, 2\}$, and both real parts of eigenvectors for Jacobian matrix \mathbf{J}_{i*} in (7). The closest two vectors, i.e. vectors with a minimal angle between them, have the greatest dot product. Therefore, for each directional vector in (6) we get one vector, thus four vectors \mathbf{u}_{1L} , \mathbf{u}_{1R} , \mathbf{u}_{2L} and \mathbf{u}_{2R} , i.e., for example, \mathbf{u}_{1L} is computed as the following procedure:

$$\{\mathbf{u}_{1L}, \mathbf{u}_{2L}\} = \operatorname{Re}(\operatorname{eigenvectors}(\mathbf{J}_{1L})) \quad \mathbf{u}_{1L} = \begin{cases} \mathbf{u}_{1L} & \mathbf{u}_{1L} \cdot \operatorname{Re}(\mathbf{u}_1) > \mathbf{u}_{2L} \cdot \operatorname{Re}(\mathbf{u}_1) \\ \mathbf{u}_{2L} & \mathbf{u}_{2L} \cdot \operatorname{Re}(\mathbf{u}_1) > \mathbf{u}_{1L} \cdot \operatorname{Re}(\mathbf{u}_1) \end{cases}. \quad (8)$$

The curvature vectors of a vector field are computed as follows:

$$\mathbf{c}_1 = \frac{1}{2h} \left(\frac{\mathbf{u}_{1R}}{\|\mathbf{u}_{1R}\|} - \frac{\mathbf{u}_{1L}}{\|\mathbf{u}_{1L}\|} \right) \quad \mathbf{c}_2 = \frac{1}{2h} \left(\frac{\mathbf{u}_{2R}}{\|\mathbf{u}_{2R}\|} - \frac{\mathbf{u}_{2L}}{\|\mathbf{u}_{2L}\|} \right). \quad (9)$$

This is a discrete formula for curvature calculation using the difference between two unit vectors. The important property of the curvature vectors in (9) is the perpendicularity to $\operatorname{Re}(\mathbf{u}_1)$, resp. $\operatorname{Re}(\mathbf{u}_2)$, i.e.

$$\mathbf{c}_1 \cdot \operatorname{Re}(\mathbf{u}_1) = 0 \quad \mathbf{c}_2 \cdot \operatorname{Re}(\mathbf{u}_2) = 0. \quad (10)$$

The length of the curvature vectors in (9) is a number that characterizes how much each of the main axes of the vector field is curved. In the case that both curvatures are equal to zero, then matrices \mathbf{H}_x and \mathbf{H}_y must be zero matrices; otherwise at least one of the curvatures is nonzero.

4 Example of Vector Field Curvature

The vector field around a critical point can be classified as one of the vector types [1]. This section presents examples of how the vector field approximated with the same Jacobian matrix changes when changing the Hessian matrices used to approximate the vector field around a critical point.

An example of the vector field around a saddle point can be characterized with the following approximation:

$$\begin{aligned} v_x &= \begin{bmatrix} 1.2 \\ 1.4 \end{bmatrix}^T \cdot \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} + \frac{1}{2} t \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}^T \cdot \begin{bmatrix} 1.2 & 0.84 \\ 0.84 & 1.2 \end{bmatrix} \cdot \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \\ v_y &= \begin{bmatrix} 0.7 \\ -0.9 \end{bmatrix}^T \cdot \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} + \frac{1}{2} t \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}^T \cdot \begin{bmatrix} -2 & 0.6 \\ 0.6 & 2 \end{bmatrix} \cdot \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}, \end{aligned} \quad (11)$$

where $t \in \mathbb{R}$ is a parameter. If we continuously change the parameter t , the vector field will change continuously as well, i.e. there will be no discontinuity.

As an example, the parameter $t \in (-1; 1)$ was changed and both curvatures of the main axes of the vector field were calculated. It can be seen that both the curvatures change continuously (see results in Fig. 1). For parameter $t = 0$, the vector field is approximated only with the Jacobian matrix part of (11), i.e.

$$v_x = \begin{bmatrix} 1.2 & 1.4 \end{bmatrix} \cdot \begin{bmatrix} \Delta x & \Delta y \end{bmatrix}^T \quad v_y = \begin{bmatrix} 0.7 & -0.9 \end{bmatrix} \cdot \begin{bmatrix} \Delta x & \Delta y \end{bmatrix}^T \quad (12)$$

and both the curvatures are thus equal to 0.

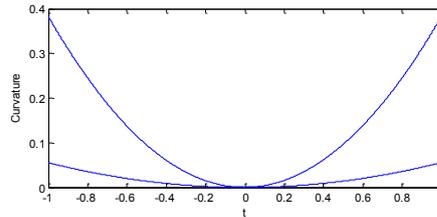


Fig. 1. Progress of both curvatures when changing parameter $t \in (-1; 1)$ in (11). One curvature grows faster with a greater absolute value of t . This means that one main axis is more curved than the other.

The approximated vector field represented by (11) can be seen for different values of parameter t in Fig. 2. It can be seen that the vector field has a different phase portrait for different values t ; however, all of them have the same description using a linear approximation of the vector field around a critical point.

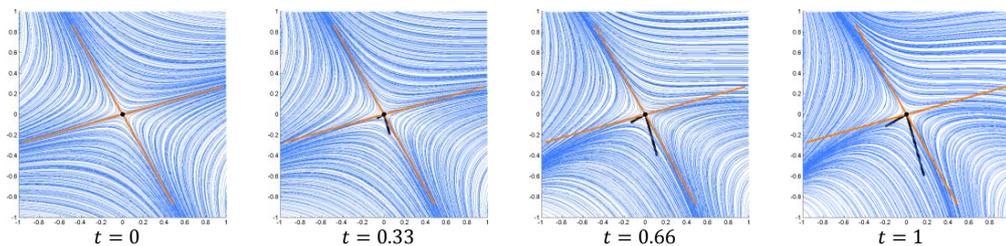


Fig. 2. Vector fields and their curvatures for different parameters t in (11). The orange lines visualize the main axes of vector fields obtained from the linear part of the approximation. The black lines visualize vectors of the curvature of the main axes (note that they are perpendicular to the main orange axes).

5 Conclusion

A new and easy algorithm for calculating the curvature of vector fields has been presented. It uses second order partial derivatives for an approximation of a vector field around a critical point. This approximation gives us a more detailed description of the vector field close to the critical point than the standardly used linear approximation.

The algorithm was presented for 2D flow fields because of their simplicity and clarity, but it can in general be used with no modifications for N dimensional vector flow fields.

Acknowledgments

The authors would like to thank their colleagues at the University of West Bohemia, Plzen, for their discussions and suggestions, and anonymous reviewers for their valuable comments and hints provided. The research was supported by projects Czech Science Foundation (GACR) No. 17-05534S and partly by SGS 2016-013.

References

- [1] J. Helman and L. Hesselink. "Representation and display of vector field topology in fluid flow data sets," IEEE Computer, Vol. 22, No. 8, pp. 27-36, 1989.
- [2] S. Koch, J. Kasten, A. Wiebel, G. Scheuermann and M. Hlawitschka, "2D Vector field approximation using linear neighborhoods," The Visual Computer, pp. 1-16, 2015.
- [3] W. de Leeuw and R. van Liere, "Collapsing flow topology using area metrics," Proc. IEEE Visualization '99, pp. 349-354, 1999.
- [4] K. Lu, A. Chaudhuri, T. Y. Lee, H. W. Shen and P. C. Wong. "Exploring vector fields with distribution-based streamline analysis," PacificVis, pp. 257-264, 2013.
- [5] P. A. Philippou, and R. N. Strickland, "Vector field analysis and synthesis using three dimensional phase portraits," Graph. Models Image Process., Vol. 59, No. 6, pp. 446-462, 1997.
- [6] G. Scheuermann, H. Krüger, M. Menzel, and A. Rockwood, "Visualizing non-linear vector field topology," IEEE Transactions on Visualization and Computer Graphics, Vol. 4, No. 2, pp. 109-116, 1998.
- [7] P. Skraba, B. Wang, G. Chen, and P. Rosen, "2D vector field simplification based on robustness," Pacific Visualization Symposium (PacificVis), IEEE, pp. 49-56, 2014.
- [8] M. Smolik, V. Skala, "Spherical RBF Vector Field Interpolation: Experimental Study," SAMI 2017, pp. 431-434, Slovakia, 2017.
- [9] M. Smolik, V. Skala, "Vector Field Interpolation with Radial Basis Functions," SIGRAD 2016, pp. 15-21, Sweden, 2016
- [10] H. Theisel, "Vector field curvature and applications," PhD dissertation, University of Rostock, Germany, 1995.
- [11] US GFS global weather model. National Centers for Environmental Information (NCEI), <https://www.ncdc.noaa.gov/data-access/model-data/model-datasets/global-forecast-system-gfs>, [downloaded: 9.11.2016].
- [12] T. Weinkauff and H. Theisel. "Curvature measures of 3D vector fields and their applications", Journal of WSCG, Vol. 10, No. 2, 2002.
- [13] T. Weinkauff, H. Theisel, K. Shi, H.-C. Hege, and H.-P. Seidel, "Extracting higher order critical points and topological simplification of 3D vector fields," Proc. IEEE Visualization 2005, pp. 559-566, Minneapolis, U.S.A., 2005.
- [14] T. Wischgoll and G. Scheuermann, "Detection and visualization of closed streamlines in planar flows," IEEE Transactions on Visualization and Computer Graphics, Vol. 7, No. 2, pp. 165-172, 2001.

4.8 Vector field second order derivative approximation and geometrical characteristics

The paper [Smolik and Skala, 2017a] proposed a description of a vector field using the second order derivative. This second order derivative description, i.e. the Hessian matrix and Jacobian matrix, has some geometrical characteristic. Using both matrices (Jacobian and Hessian matrix), one can reformulate it into a different matrix. This matrix is a description of a conic section.

The paper [Smolik and Skala, 2017c] describes the relation between the conic sections and different types of critical points. Moreover, the conic section can be used to locate additional critical points in the vector field, as different types of conic sections can have different number of intersections, where each intersection represents a critical point.

Citation:

- Michal Smolik and Vaclav Skala. Vector field second order derivative approximation and geometrical characteristics. In *International Conference on Computational Science and Its Applications*, pages 148–158. Springer, 2017.

Vector Field Second Order Derivative Approximation and Geometrical Characteristics

Michal Smolik¹, Vaclav Skala¹

¹Department of Computer Science and Engineering, Faculty of Applied Sciences
University of West Bohemia, Plzen, Czech Republic
{smolik, skala}@kiv.zcu.cz

Abstract. Vector field is mostly linearly approximated for the purpose of classification and description. This approximation gives us only basic information of the vector field. We will show how to approximate the vector field with second order derivatives, i.e. Hessian and Jacobian matrices. This approximation gives us much more detailed description of the vector field. Moreover, we will show the similarity of this approximation with conic section formula.

Keywords: Vector field; Critical point; Geometry; Conic section; Hessian matrix.

1 Introduction

The visualization of vector field topology is a problem that arises naturally when studying the qualitative structure of flows that are tangential to some surface. The knowledge of the data in a single point would be of little help when the goal is to obtain knowledge and understanding of the whole vector field. The individual numbers can be of little interest. It is the connection between them, which is important.

Helman and Hesselink [6] introduced the concept of the topology of a planar vector field to the visualization community. They extracted critical points and classified them into sources, sinks and saddles, and integrated certain stream lines called separatrices from the saddles in the directions of the eigenvectors of the Jacobian matrix. Later, topological methods have been extended to higher order critical points [14], boundary switch points [10], and closed separatrices [21]. In addition, topological methods using classification have been applied to simplify [16], [15], smooth [20], compress [1], [7], compare [11] and design vector fields.

The published research methods use for classification of critical points and vector field description only linear approximation of the vector field. None of it uses an approximation with second order partial derivatives, i.e. Hessian matrix. This approximation gives a more detailed description of the vector field around a critical point and can be used for a more detailed classification. Use of the approximation with Hessian matrix will be described in this paper.

2 Vector Field Approximation

Vector fields [18] on surfaces [17] are important objects, which appear frequently in scientific simulation in CFD (Computational Fluid Dynamics) [2], [12] or modelling by FEM (Finite Element Method). To be visualized [5], [8], such vector fields are usually linearly approximated for the sake of simplicity and performance considerations. Other possible approximations are [3], [4], [9].

The vector field can be easily analyzed when having an approximation of the vector field near some location point. The important places to be analyzed are so called critical points. Analyzing the vector field behavior near these points gives us the information about the characteristic of the vector field.

2.1 Critical Point

Critical points (\mathbf{x}_0) of the vector field are points at which the magnitude of the vector vanishes

$$\frac{d\mathbf{x}}{dt} = \mathbf{v}(\mathbf{x}) = \mathbf{0}, \quad (1)$$

i.e. all components are equal to zero

$$\begin{bmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \quad (2)$$

A critical point is said to be isolated, or simple, if the vector field is non-vanishing in an open neighborhood around the critical point. Thus for all surrounding points \mathbf{x}_ε of the critical point \mathbf{x}_0 the equation (1) does not apply, i.e.

$$\frac{d\mathbf{x}_\varepsilon}{dt} \neq \mathbf{0}. \quad (3)$$

At critical points, the direction of the field line is indeterminate, and they are the only points in the vector field where field lines can intersect (asymptotically). The terms singular point, null point, neutral point or equilibrium point are also frequently used to describe critical points.

These points are important because together with the nearby surrounding vectors, they have more information encoded in them than any such group in the vector field, regarding the total behavior of the field.

2.2 Linearization of Vector Field

Critical points can be characterized according to the behavior of nearby tangent curves. We can use a particular set of these curves to define a skeleton that characterizes the global behavior of all other tangent curves in the vector field. An important feature of differential equations is that it is often possible to determine the

local stability of a critical point by approximating the system by a linear system. These approximations are aimed at studying the local behavior of a system, where the nonlinear effects are expected to be small. To locally approximate a system, the Taylor series expansion must be utilized locally to find the relation between \mathbf{v} and position \mathbf{x} , supposing the flow \mathbf{v} to be sufficiently smooth and differentiable. In such case, the expansion of \mathbf{v} around the critical points \mathbf{x}_0 is

$$\mathbf{v}(\mathbf{x}) = \mathbf{v}(\mathbf{x}_0) + \frac{\partial \mathbf{v}}{\partial \mathbf{x}} (\mathbf{x} - \mathbf{x}_0). \quad (4)$$

As $\mathbf{v}(\mathbf{x}_0)$ is according to (1) equal zero for critical points, we can rewrite equation (4) using matrix notation

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} \frac{\partial v_x}{\partial x} & \frac{\partial v_x}{\partial y} \\ \frac{\partial v_y}{\partial x} & \frac{\partial v_y}{\partial y} \end{bmatrix} \cdot \begin{bmatrix} x - x_0 \\ y - y_0 \end{bmatrix} \quad (5)$$

$$\mathbf{v} = \mathbf{J} \cdot (\mathbf{x} - \mathbf{x}_0), \quad (6)$$

where \mathbf{J} is called Jacobian matrix and characterizes the vector field behavior around a critical point \mathbf{x}_0 .

2.3 Approximation Using Hessian Matrix

Vector fields are approximated using only linear approximation to determine the local behavior of the vector field. However, linearization gives as basic classification of the critical points and about the flow around them, the approximation using second order derivatives will give us some more information.

The approximation of vector field around a critical point using the second order derivative must be written for each vector component (v_x and v_y) separately, see the following equations

$$v_x = \begin{bmatrix} \frac{\partial v_x}{\partial x} \\ \frac{\partial v_x}{\partial y} \end{bmatrix}^T \cdot \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} + \frac{1}{2} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}^T \cdot \begin{bmatrix} \frac{\partial^2 v_x}{\partial x^2} & \frac{\partial^2 v_x}{\partial x \partial y} \\ \frac{\partial^2 v_x}{\partial y \partial x} & \frac{\partial^2 v_x}{\partial y^2} \end{bmatrix} \cdot \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \quad (7)$$

$$v_y = \begin{bmatrix} \frac{\partial v_y}{\partial x} \\ \frac{\partial v_y}{\partial y} \end{bmatrix}^T \cdot \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} + \frac{1}{2} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}^T \cdot \begin{bmatrix} \frac{\partial^2 v_y}{\partial x^2} & \frac{\partial^2 v_y}{\partial x \partial y} \\ \frac{\partial^2 v_y}{\partial y \partial x} & \frac{\partial^2 v_y}{\partial y^2} \end{bmatrix} \cdot \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}, \quad (8)$$

where $\Delta x = x - x_0$ and $\Delta y = y - y_0$. These two equations can be written in matrix notation as well

$$v_x = J_x \cdot (\mathbf{x} - \mathbf{x}_0) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_0)^T \cdot \mathbf{H}_x \cdot (\mathbf{x} - \mathbf{x}_0) \quad (9)$$

$$v_y = J_y \cdot (\mathbf{x} - \mathbf{x}_0) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_0)^T \cdot \mathbf{H}_y \cdot (\mathbf{x} - \mathbf{x}_0), \quad (10)$$

where \mathbf{H}_x and \mathbf{H}_y are Hessian matrices, J_x is the first row of Jacobian matrix and J_y is the second row of Jacobian matrix.

The Hessian matrix is a square matrix of second-order partial derivatives of a scalar-valued function, or scalar field. It describes the local curvature of a function of many variables.

Approximation of vector field using (7) and (8) gives us more detailed description than approximation of vector field using (5), see Fig. 1. The approximation in Fig. 1 (right) gives us the same information like in Fig. 1 (left), although we can see the curvature of the two main axis for the saddle.

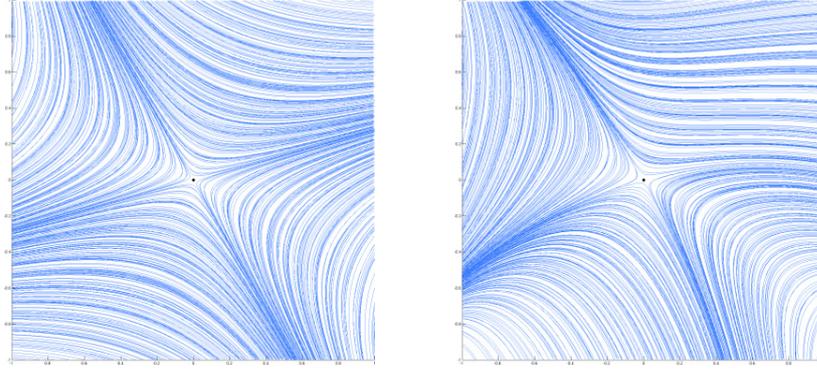


Fig. 1. Comparison between the phase portraits for the vector field approximated using linear approximation (left) and using second order derivative (right).

Equations (7) and (8) can be rewritten in different formulas as follows

$$v_x = \frac{1}{2} \begin{bmatrix} \Delta x & \Delta y & 1 \end{bmatrix} \cdot \begin{bmatrix} \frac{\partial^2 v_x}{\partial x^2} & \frac{\partial^2 v_x}{\partial x \partial y} & \frac{\partial v_x}{\partial x} \\ \frac{\partial^2 v_x}{\partial y \partial x} & \frac{\partial^2 v_x}{\partial y^2} & \frac{\partial v_x}{\partial y} \\ \frac{\partial v_x}{\partial x} & \frac{\partial v_x}{\partial y} & 0 \end{bmatrix} \cdot \begin{bmatrix} \Delta x \\ \Delta y \\ 1 \end{bmatrix} \quad (11)$$

$$v_y = \frac{1}{2} [\Delta x \quad \Delta y \quad 1] \cdot \begin{bmatrix} \frac{\partial^2 v_y}{\partial x^2} & \frac{\partial^2 v_y}{\partial x \partial y} & \frac{\partial v_y}{\partial x} \\ \frac{\partial^2 v_y}{\partial y \partial x} & \frac{\partial^2 v_y}{\partial y^2} & \frac{\partial v_y}{\partial y} \\ \frac{\partial v_y}{\partial x} & \frac{\partial v_y}{\partial y} & 0 \end{bmatrix} \cdot \begin{bmatrix} \Delta x \\ \Delta y \\ 1 \end{bmatrix}. \quad (12)$$

These two equations have some geometrical background. When v_x and v_y are equal zero, each equation describes some conic section.

Approximation of the vector field using Hessian matrix, i.e. using second order derivatives, is a bit more computationally expensive than the standard linear approximation but gives us more detailed description of the vector field as will be seen in the following chapters.

Conic Section.

A conic is the curve obtained as the intersection of a plane, called the cutting plane, with a double cone, see Fig. 2. Planes that pass through the vertex of the cone will intersect the cone in a point, a line or a pair of intersecting lines. These are called degenerate conics and some authors do not consider them to be conics at all.

There are three types of non-degenerated conics, the ellipse, parabola, and hyperbola, see Fig. 2. The circle is a special kind of ellipse. The circle and the ellipse arise when the intersection of the cone and plane is a closed curve. The circle is obtained when the cutting plane is parallel to the plane of the generating circle of the cone, this means that the cutting plane is perpendicular to the symmetry axis of the cone. If the cutting plane is parallel to exactly one generating line of the cone, then the conic is unbounded and is called a parabola. In the remaining case, the figure is a hyperbola. In this case, the plane will intersect both halves of the cone, producing two separate unbounded curves.

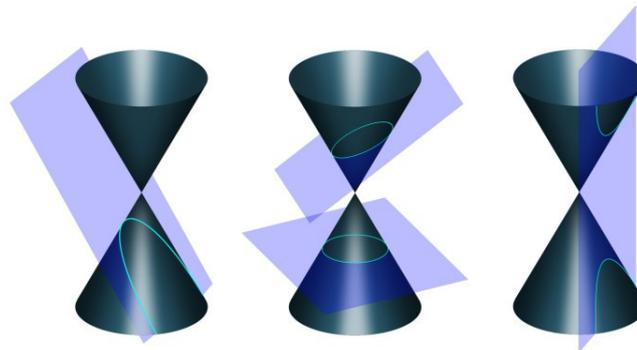


Fig. 2. Types of conic sections, i.e. parabola, circle and ellipse, and hyperbola.

A conic section is described by the following implicit equation

$$\begin{bmatrix} x & y & 1 \end{bmatrix} \cdot \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0. \quad (13)$$

where $a_{ij}, i, j \in \{1, 2, 3\}$ are coefficients of conic section. Depending on these values, we can classify the types of conic sections. To do that, we need to compute two determinants

$$\Omega = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} \quad (14)$$

$$\omega = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}. \quad (15)$$

When knowing determinants Ω and ω we can easily classify the type of conic section using the following table

Table 1: Classification of conic section.

	$\omega \neq 0$		$\omega = 0$
$\Omega \neq 0$	$\omega > 0$	$\omega < 0$	parabola
	ellipse	hyperbola	
$\Omega = 0$	pair of intersecting lines		pair of parallel lines

Equations (11) and (12) are the same as (13) when $v_x = 0$ and $v_y = 0$ and therefore they geometrically represent conic sections.

3 Classification of Critical Points

There exist a finite set of fundamentally different critical points, defined by the number of inflow and outflow directions, spiraling structures etc., and combinations of these. Since the set is finite, each critical point can be classified. Such a classification defines the field completely in a close neighborhood around the critical point. By knowing the location and classification of critical points in a vector field, the topology of the field is known in small areas around these. Assuming a smooth transition between these areas, one can construct a simplified model of the whole vector field. Such a simplified representation is useful, for instance, in compressing vector field data into simpler building blocks [13].

The critical points are classified based on the vector field around that point. The information derived from the classification of critical points aids the information selection process when it comes to visualizing the field. By choosing seed points for field lines based on the topology of critical points, field lines encoding important information is ensured. A more advanced approach is to connect critical points, and use the connecting lines and surfaces to separate areas of different flow topology [1], [19].

3.1 Standard Classification Using a Linear Approximation

The fact that a linear model can be used to study the behavior of a nonlinear system near a critical point is a powerful one [1]. We can use the Jacobian matrix to characterize the vector field and the behavior of nearby tangent curves, for nondegenerate critical point.

The eigenvalues and eigenvectors of Jacobian matrix are very important for vector field classification and description (see Fig. 3). A real eigenvector of the Jacobian matrix defines a direction such that if we move slightly from the critical point in that direction, the field is parallel to the direction we moved. Thus, at the critical point, the real eigenvectors are tangent to the trajectories that end on the point. The sign of the corresponding eigenvalue determines whether the trajectory is outgoing (repelling) or incoming (attracting) at the critical point. The imaginary part of an eigenvalue denotes circulation about the point.

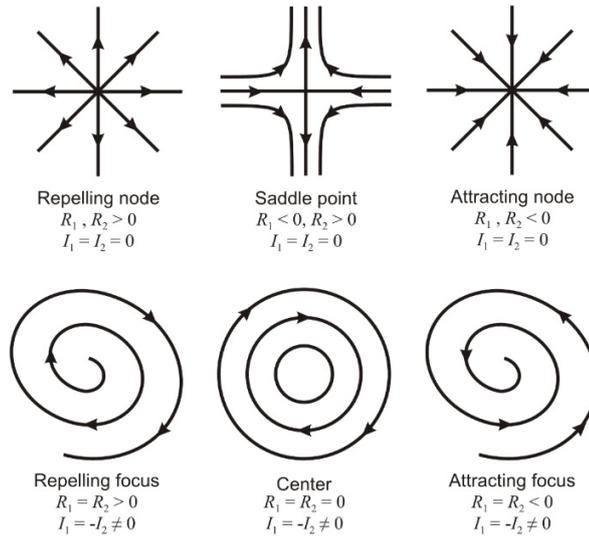


Fig. 3. Classification of 2D first order critical points. R_1, R_2 denote the real parts of the eigenvalues of the Jacobian matrix while I_1, I_2 denote their imaginary parts (from [1]).

3.2 Classification Using Description of Conic Sections

Each vector field can be approximated at a critical point with the approximation that uses the second order derivatives, i.e. Hessian matrix. One such example of approximated vector field around a critical point $\mathbf{x}_0 = [0, 0]^T$ can be

$$v_x = \frac{1}{2} \begin{bmatrix} \Delta x & \Delta y & 1 \end{bmatrix} \cdot \begin{bmatrix} -1 & 1 & 1 \\ 1 & -1 & 2 \\ 1 & 2 & 0 \end{bmatrix} \cdot \begin{bmatrix} \Delta x \\ \Delta y \\ 1 \end{bmatrix}, \quad (16)$$

$$v_y = \frac{1}{2} [\Delta x \quad \Delta y \quad 1] \cdot \begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & 1.5 \\ -1 & 1.5 & 0 \end{bmatrix} \cdot \begin{bmatrix} \Delta x \\ \Delta y \\ 1 \end{bmatrix}. \quad (17)$$

Equation (16) represents for $v_x = 0$ a parabola and (17) for $v_y = 0$ a line. This approximated vector field can be seen in Fig. 4.

Now, we showed conic sections that have only one intersection point at $[0, 0]^T$. Two conic sections can have up to four intersections. Each intersection defines a critical point. Therefore, we can approximate a vector field around one critical point and some more critical points in the neighborhood will be included in this approximation.

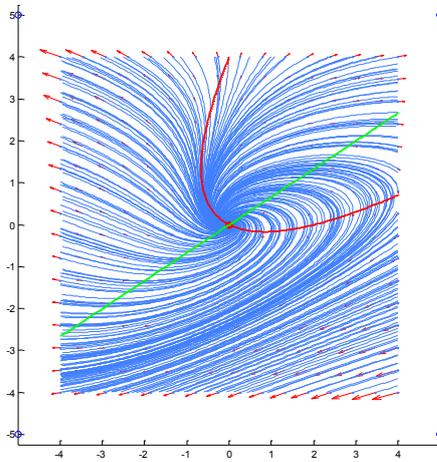


Fig. 4. Vector field approximated as (16) and (17). The zero iso-lines are a line and a parabola.

Vector fields around a focus critical point can be for some real vector field approximated for example as

$$v_x = \frac{1}{2} [\Delta x \quad \Delta y \quad 1] \cdot \begin{bmatrix} 1 & -3 & 1 \\ -3 & 1 & 2 \\ 1 & 2 & 0 \end{bmatrix} \cdot \begin{bmatrix} \Delta x \\ \Delta y \\ 1 \end{bmatrix} \quad (18)$$

$$v_y = \frac{1}{2} [\Delta x \quad \Delta y \quad 1] \cdot \begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & 1.5 \\ -1 & 1.5 & 0 \end{bmatrix} \cdot \begin{bmatrix} \Delta x \\ \Delta y \\ 1 \end{bmatrix}$$

$$v_x = \frac{1}{2} [\Delta x \ \Delta y \ 1] \cdot \begin{bmatrix} -0.5 & 0.5 & 1 \\ 0.5 & -0.5 & 2 \\ 1 & 2 & 0 \end{bmatrix} \cdot \begin{bmatrix} \Delta x \\ \Delta y \\ 1 \end{bmatrix}$$

$$v_y = \frac{1}{2} [\Delta x \ \Delta y \ 1] \cdot \begin{bmatrix} -1 & 1 & -1 \\ 1 & -1 & 1.5 \\ -1 & 1.5 & 0 \end{bmatrix} \cdot \begin{bmatrix} \Delta x \\ \Delta y \\ 1 \end{bmatrix}$$
(19)

This both approximations of vector fields describe behavior around a focus critical point at $[0, 0]^T$. Both of them contain one more critical point, which is a saddle critical point. These saddle critical points do not have to be real critical points of the approximated vector field, but they can be present in the vector field. Therefore, this approximation can give us some information about other possible critical points in the neighborhood of approximated critical point \mathbf{x}_0 . When locating all critical points in the vector field, we can use this information to increase the probability of finding all critical points.

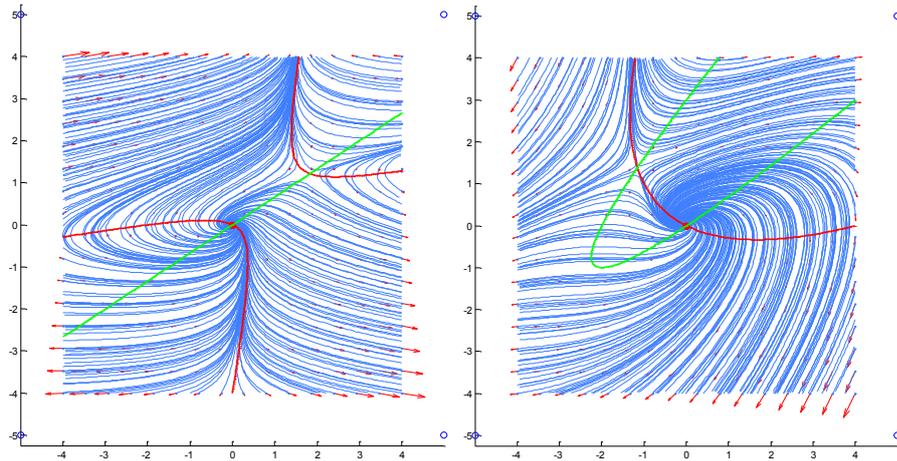


Fig. 5. Vector field approximated as (18) (left) and (19) (right). The zero iso-lines are a line and a hyperbola (left), or two parabolas (right).

The maximal number of two conic sections intersection points is four. In the next example, we will show it. Let us have a vector field, which can be approximated at point \mathbf{x}_0 for example as

$$v_x = \frac{1}{2} [\Delta x \ \Delta y \ 1] \cdot \begin{bmatrix} -0.25 & 0 & 1 \\ 0 & -1 & 2 \\ 1 & 2 & 0 \end{bmatrix} \cdot \begin{bmatrix} \Delta x \\ \Delta y \\ 1 \end{bmatrix}$$

$$v_y = \frac{1}{2} [\Delta x \ \Delta y \ 1] \cdot \begin{bmatrix} -1 & 1 & -1 \\ 1 & -1 & 1.5 \\ -1 & 1.5 & 0 \end{bmatrix} \cdot \begin{bmatrix} \Delta x \\ \Delta y \\ 1 \end{bmatrix}$$
(20)

$$v_x = \frac{1}{2} [\Delta x \ \Delta y \ 1] \cdot \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 2 \\ 1 & 2 & 0 \end{bmatrix} \cdot \begin{bmatrix} \Delta x \\ \Delta y \\ 1 \end{bmatrix} \quad (21)$$

$$v_y = \frac{1}{2} [\Delta x \ \Delta y \ 1] \cdot \begin{bmatrix} 1 & -1 & -1 \\ -1 & 1.5 & 1.5 \\ -1 & 1.5 & 0 \end{bmatrix} \cdot \begin{bmatrix} \Delta x \\ \Delta y \\ 1 \end{bmatrix}$$

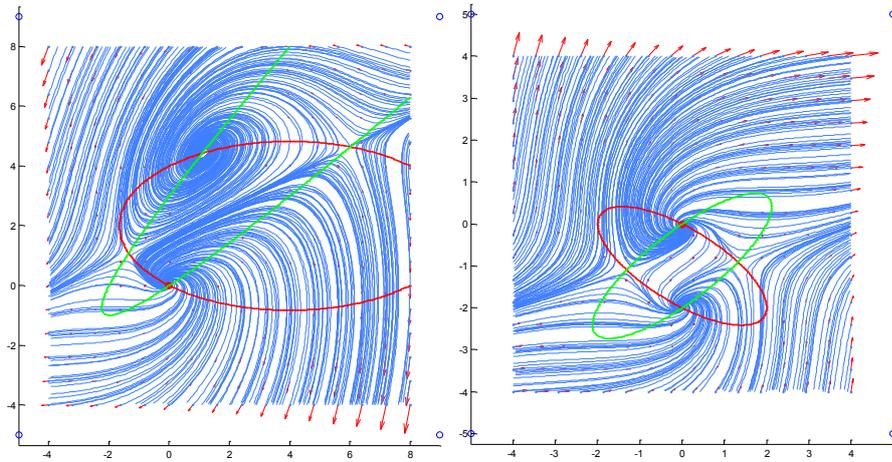


Fig. 6. Vector field approximated as (20) (left) and (21) (right). The zero iso-lines are a parabola and an ellipse (left), or two ellipses (right).

These two approximations (20) and (21) of vector fields are visualized in Fig. 6. It can be seen, that each approximation contains four critical points, i.e. one critical point where the vector field was approximated and three more critical points.

4 Conclusion

A new vector field critical points description using the second order derivatives approximation is described. The approximation can be rewritten in a matrix form of a conic section formula. We proved, that approximation using Hessian matrix, rather than only Jacobian matrix, gives us better representation of a vector field and it can help with localization of critical points in a vector field.

Acknowledgments.

The authors would like to thank their colleagues at the University of West Bohemia, Plzen, for their comments and suggestions, their valuable comments and hints provided. The research was supported by projects Czech Science Foundation (GACR) No. 17-05534S and partly by SGS 2016-013.

References

- [1] A. Agranovsky, D. Camp, K. I. Joy, H. Childs: Subsampling-based compression and flow visualization. In IS&T/SPIE Electronic Imaging, International Society for Optics and Photonics, 2015.
- [2] F. Balduzzi, A. Bianchini, R. Maleci, G. Ferrara and L. Ferrari. "Critical issues in the CFD simulation of Darrieus wind turbines," *Renewable Energy*, Vol. 85, pp. 419-435, 2016.
- [3] M. N. Benbourhim. A. Bouhamidi. "Approximation of vectors fields by thin plate splines with tension," *Journal of Approximation Theory*, Vol. 136, No. 2, pp. 198-229, 2005.
- [4] D. A. Cervantes Cabrera, P. González-Casanova, Ch. Gout, L. H. Juárez, L. R. Reséndiz. "Vector field approximation using radial basis functions," *Journal of Computational and Applied Mathematics*, Vol. 240, pp. 163-173, 2013.
- [5] A. Forsberg, J. Chen and D. Laidlaw. "Comparing 3d vector field visualization methods: A user study," *IEEE Transactions on Visualization and Computer Graphics*, Vol. 15, No. 6, pp. 1219-1226, 2009.
- [6] J. Helman and L. Hesselink. "Representation and display of vector field topology in fluid flow data sets," *IEEE Computer*, Vol. 22, No. 8, pp. 27-36, 1989.
- [7] S. Koch, J. Kasten, A. Wiebel, G. Scheuermann and M. Hlawitschka, "2D Vector field approximation using linear neighborhoods," *The Visual Computer*, pp. 1-16, 2015.
- [8] D. H. Laidlaw, R. M. Kirby, C. D. Jackson, J. S. Davidson, T. S. Miller, M. Da Silva, W. H. Warren and M. J. Tarr. "Comparing 2D vector field visualization methods: A user study," *IEEE Transactions on Visualization and Computer Graphics*, Vol. 11, No. 1, pp. 59-70, 2005.
- [9] M. Lage, F. Petronetto, A. Paiva, H. Lopes, T. Lewiner and G. Tavares, "Vector field reconstruction from sparse samples with applications," 19th Brazilian Symposium on Computer Graphics and Image Processing, SIBGRAPI, 2006.
- [10] W. de Leeuw and R. van Liere, "Collapsing flow topology using area metrics," *Proc. IEEE Visualization '99*, pp. 349-354, 1999.
- [11] K. Lu, A. Chaudhuri, T. Y. Lee, H. W. Shen and P. C. Wong. "Exploring vector fields with distribution-based streamline analysis," *PacificVis*, pp. 257-264, 2013.
- [12] C. Peng, Y. Teng, B. Hwang, Z. Guo and L. P. Wang. "Implementation issues and benchmarking of lattice Boltzmann method for moving rigid particle simulations in a viscous flow," *Computers & Mathematics with Applications*, Vol. 72, No. 2, pp. 349-374, 2016.
- [13] P. A. Philippou, and R. N. Strickland, "Vector field analysis and synthesis using three-dimensional phase portraits," *Graph. Models Image Process.* Vol. 59, No. 6, pp. 446-462, 1997.
- [14] G. Scheuermann, H. Krüger, M. Menzel, and A. Rockwood, "Visualizing non-linear vector field topology," *IEEE Transactions on Visualization and Computer Graphics*, Vol. 4, No. 2, pp. 109-116, 1998.
- [15] P. Skraba, P. Rosen, B. Wang, G. Chen, H. Bhatia and V. Pascucci: Critical Point Cancellation in 3D Vector Fields: Robustness and Discussion. *IEEE Transactions on Visualization and Computer Graphics*, 2016.

- [16] P. Skraba, B. Wang, G. Chen, and P. Rosen, "2D vector field simplification based on robustness," Pacific Visualization Symposium (PacificVis), IEEE, pp. 49-56, 2014.
- [17] M. Smolik, V. Skala, "Spherical RBF Vector Field Interpolation: Experimental Study," SAMI 2017, pp. 431-434, Slovakia, 2017.
- [18] M. Smolik, V. Skala, "Vector Field Interpolation with Radial Basis Functions," SIGRAD 2016, pp. 15-21, Sweden, 2016
- [19] T. Weinkauff, H. Theisel, K. Shi, H.-C. Hege, and H.-P. Seidel, "Extracting higher order critical points and topological simplification of 3D vector fields," Proc. IEEE Visualization 2005, pp. 559-566, Minneapolis, U.S.A., 2005.
- [20] R. Westermann, C. Johnson and T. Ertl. "Topology-preserving smoothing of vector fields," IEEE Transactions on Visualization and Computer Graphics, Vol. 7, No. 3, pp. 222-229, 2001.
- [21] T. Wischgoll and G. Scheuermann, "Detection and visualization of closed streamlines in planar flows," IEEE Transactions on Visualization and Computer Graphics, Vol. 7, No. 2, pp. 165-172, 2001.

4.9 Vector field interpolation with Radial basis functions

The main vector field behavior can be described using critical points. The paper [Smolik and Skala, 2016a] presents an approach for RBF vector field approximation. This approach selects important locations and vectors in the vector field and use it for the RBF interpolation to represent the approximation of the input vector field. The selected locations and vectors are located at the position of critical points and then in the surrounding of those critical points.

The interpolation of only a few selected vectors results in a very high compression of the vector field, while still preserving the main flow of the vector field.

Citation:

- Michal Smolik and Vaclav Skala. Vector field interpolation with radial basis functions. In *Proceedings of SIGRAD 2016, May 23rd and 24th, Visby, Sweden*, number 127, pages 15–21. Linköping University Electronic Press, 2016.

Vector Field Interpolation with Radial Basis Functions

M. Smolik¹ and V. Skala¹

¹Faculty of Applied Sciences, University of West Bohemia, Plzen, Czech Republic

Abstract

This paper presents a new approach for the Radial Basis Function (RBF) interpolation of a vector field. Standard approaches for interpolation randomly select points for interpolation. Our approach uses the knowledge of vector field topology and selects points for interpolation according to the critical points location. We present the results of interpolation errors on a vector field generated from an analytical function.

Categories and Subject Descriptors (according to ACM CCS): G.1.1 [Numerical analysis]: Interpolation—Interpolation formulas

1. Introduction

Interpolation is probably the most frequent operation used in computational methods. Several methods have been developed for data interpolation, but they expect some kind of data "ordering", e.g. structured mesh, rectangular 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 scattered data are tessellated using triangulation but this approach is quite prohibitive for the case of d -dimensional data interpolation because of the computational cost.

Interpolating scattered vector data on a surface becomes frequent in applied problem solutions. There are applications for vector field decomposition [EJF09], for vector field design system for surfaces that allows the user to control the number of singularities in the vector field and their placement [ZMT06]. [MZT*14] uses the vector field interpolation for estimating robust point correspondences between two sets of points.

2. Vector Field

Vector fields on surfaces are important objects, which appear frequently in scientific simulation in CFD (Computational Fluid Dynamics) or modeling by FEM (Finite Element Method). To be visualized, such vector fields are usually linearly approximated for the sake of simplicity and performance considerations.

The vector field can be easily analyzed when having an approximation of the vector field near some location point.

The important places to be analyzed are so called critical points. Analyzing the vector field behavior near these points gives us the information about the characteristic of the vector field.

2.1. Critical Point

Critical points \mathbf{x}_0 of the vector field are points at which the magnitude of the vector vanishes

$$\frac{d\mathbf{x}}{dt} = \mathbf{v}(\mathbf{x}) = \mathbf{0}, \quad (1)$$

i.e. all components are equal to zero

$$\begin{bmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \quad (2)$$

A critical point is said to be isolated, or simple, if the vector field is non vanishing in an open neighborhood around the critical point. Thus for all surrounding points \mathbf{x}_ϵ of the critical point \mathbf{x}_0 the equation (1) does not apply, i.e.

$$\frac{d\mathbf{x}_\epsilon}{dt} \neq \mathbf{0}, \quad (3)$$

At critical points, the direction of the field line is indeterminate, and they are the only points in the vector field where field lines can intersect (asymptotically). The terms singular point, null point, neutral point or equilibrium point are also frequently used to describe critical points.

These points are important because together with the

nearby surrounding vectors, they have more information encoded in them than any such group in the vector field, regarding the total behavior of the field.

2.2. Linearization of Vector Field

Critical points can be characterized according to the behavior of nearby tangent curves. We can use a particular set of these curves to define a skeleton that characterizes the global behavior of all other tangent curves in the vector field. An important feature of differential equations is that it is often possible to determine the local stability of a critical point by approximating the system by a linear system. These approximations are aimed at studying the local behavior of a system, where the nonlinear effects are expected to be small. To locally approximate a system, the Taylor series expansion must be utilized locally to find the relation between \mathbf{v} and position \mathbf{x} , supposing the flow \mathbf{v} to be sufficiently smooth and differentiable. In such case, the expansion of \mathbf{v} around the critical points \mathbf{x}_0 is

$$\mathbf{v}(\mathbf{x}) = \mathbf{v}(\mathbf{x}_0) + \frac{\partial \mathbf{v}}{\partial \mathbf{x}}(\mathbf{x} - \mathbf{x}_0). \quad (4)$$

As $\mathbf{v}(\mathbf{x}_0)$ is according to (1) equal zero for critical points, we can rewrite equation (4) using matrix notation

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} \frac{\partial v_x}{\partial x} & \frac{\partial v_x}{\partial y} \\ \frac{\partial v_y}{\partial x} & \frac{\partial v_y}{\partial y} \end{bmatrix} \begin{bmatrix} x - x_0 \\ y - y_0 \end{bmatrix} \quad (5)$$

$$\mathbf{v} = \mathbf{J} \cdot (\mathbf{x} - \mathbf{x}_0), \quad (6)$$

where \mathbf{J} is called Jacobian matrix and characterizes the vector field behavior around a critical point \mathbf{x}_0 .

2.3. Classification of Critical Points

There exist a finite set of fundamentally different critical points, defined by the number of inflow and outflow directions, spiraling structures etc., and combinations of these. Since the set is finite, each critical point can be classified. Such a classification defines the field completely in a close neighborhood around the critical point. By knowing the location and classification of critical points in a vector field, the topology of the field is known in small areas around these. Assuming a smooth transition between these areas, one can construct a simplified model of the whole vector field. Such a simplified representation is useful, for instance, in compressing vector field data into simpler building blocks [PS97].

The critical points are classified based on the vector field around these points. The information derived from the classification of critical points aids the information selection process when it comes to visualizing the field. By choosing seed points for field lines based on the topology of critical points, field lines encoding important information is ensured.

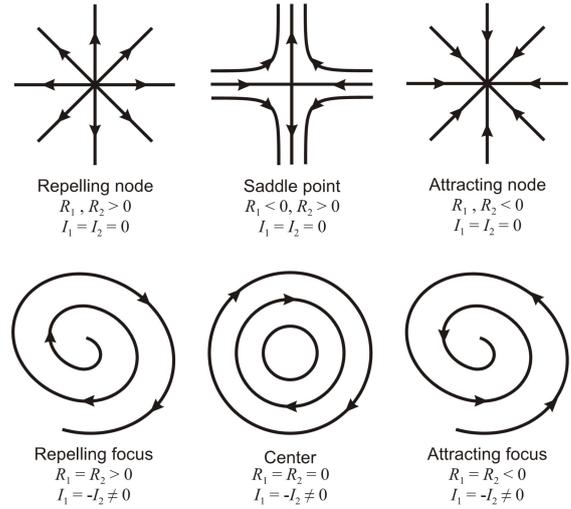


Figure 1: Classification of 2D first order critical points. R_1, R_2 denote the real parts of the eigenvalues of the Jacobian matrix while I_1, I_2 denote their imaginary parts (from [HH89]).

A more advanced approach is to connect critical points, and use the connecting lines and surfaces to separate areas of different flow topology [HH89], [WTS*05].

The fact that a linear model can be used to study the behavior of a nonlinear system near a critical point is a powerful one [HH89]. We can use the Jacobian matrix to characterize the vector field and the behavior of nearby tangent curves, for nondegenerate critical point.

The eigenvalues and the eigenvectors of Jacobian matrix are very important for vector field classification and description, see Figure 1. A real eigenvector of the Jacobian matrix defines a direction such that if we move slightly from the critical point in that direction, the field is parallel to the direction we moved. Thus, at the critical point, the real eigenvectors are tangent to the trajectories that end on the point. The sign of the corresponding eigenvalue determines whether the trajectory is outgoing (repelling) or incoming (attracting) at the critical point. The imaginary part of an eigenvalue denotes circulation about the point.

3. Radial Basis Functions

The Radial basis functions (RBF) is a technique for scattered data interpolation [PS11] and approximation [Fas07], [Ska15]. The RBF interpolation and approximation is computationally more expensive, because input data are not ordered and there is no known relation between them. Although the RBF has higher computational cost, it can be used for d -dimensional problem solution in many applications, e.g. solution of partial differential equations, image re-

construction, neural networks, fuzzy systems, GIS systems, optics etc.

The RBF is a function whose value depends only on the distance from some center point. Due to the use of the distance functions, the RBFs can be easily implemented to reconstruct the surface using scattered data in 2D, 3D or higher dimensional spaces. It should be noted that the RBF interpolation is not separable.

Radial function interpolants have a nice property of being invariant under all Euclidean transformations, i.e. translations, rotations and reflections. It means that it does not matter whether we first compute the RBF interpolation function and then apply a Euclidean transformation, or if we first transform all the data and then compute the radial function interpolants. This is result of the fact that Euclidean transformations are characterized by orthogonal transformation matrices and are therefore 2 norm invariant. Radial basis functions can be divided into two groups according to their influence. First group are "global" RBF [Sch79], for example:

Thin Plate Spline (TPS)	$\varphi(r) = r^2 \log r$	
Gauss function	$\varphi(r) = e^{-(\varepsilon r)^2}$	
Inverse Quadric (IQ)	$\varphi(r) = \frac{1}{1 + (\varepsilon r)^2}$	(7)
Inverse Multiquadric (IMQ)	$\varphi(r) = \frac{1}{\sqrt{1 + (\varepsilon r)^2}}$	
Multiquadric (MQ)	$\varphi(r) = \sqrt{1 + (\varepsilon r)^2}$	

where ε is the shape parameter of radial basis function [FP08].

The "local" RBF were introduced by [Wen06] as Compactly Supported RBF (CSRBF) and satisfy the following condition

$$\varphi(r) = (1 - r)_+^q P(r) = \begin{cases} (1 - r)^q P(r) & 0 \leq r \leq 1 \\ 0 & r > 1 \end{cases} \quad (8)$$

where $P(r)$ is a polynomial function and q is a parameter. Typical examples of CSRBF are

$$\begin{aligned} \varphi_1(r) &= (1 - \varepsilon r)_+ \\ \varphi_2(r) &= (1 - \varepsilon r)_+^3 (3\varepsilon r + 1) \\ \varphi_3(r) &= (1 - \varepsilon r)_+^5 (8(\varepsilon r)^2 + 5\varepsilon r + 1) \\ \varphi_4(r) &= (1 - \varepsilon r)_+^2 \\ \varphi_5(r) &= (1 - \varepsilon r)_+^3 (4\varepsilon r + 1) \\ \varphi_6(r) &= (1 - \varepsilon r)_+^6 (35(\varepsilon r)^2 + 18\varepsilon r + 3) \\ \varphi_7(r) &= (1 - \varepsilon r)_+^8 (32(\varepsilon r)^3 + 25(\varepsilon r)^2 + 8\varepsilon r + 1) \\ \varphi_8(r) &= (1 - \varepsilon r)_+^3 \\ \varphi_9(r) &= (1 - \varepsilon r)_+^3 (5\varepsilon r + 1) \\ \varphi_{10}(r) &= (1 - \varepsilon r)_+^7 (16(\varepsilon r)^2 + 7\varepsilon r + 1) \end{aligned} \quad (9)$$

where ε is the shape parameter of radial basis function, see Figure 2 for visualization of (9).

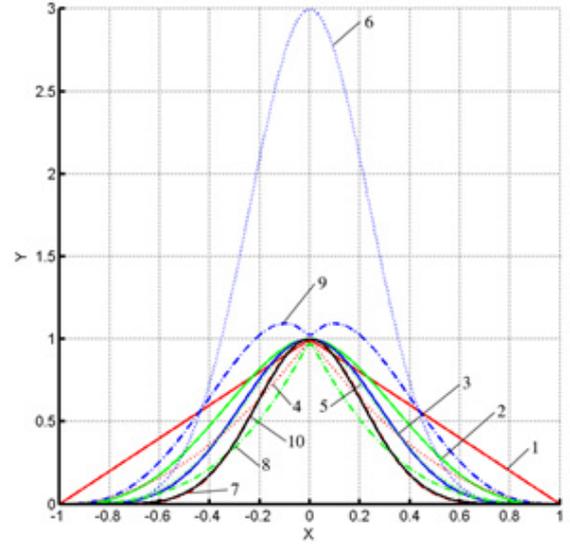


Figure 2: Examples of CSRBF (from [US04])

3.1. Radial Basis Function Interpolation

The RBF interpolation was originally introduced by [Har71] and is based on computing the distance of two points in the k -dimensional space and is defined by a function

$$f(\mathbf{x}) = \sum_{j=1}^M \lambda_j \varphi(\|\mathbf{x} - \mathbf{x}_j\|) \quad (10)$$

where λ_j are weights of the RBFs, M is the number of the radial basis functions, i.e. the number of interpolation points, and φ is the radial basis function. For a given dataset of points with associated values, i.e. in the case of scalar values $\{\mathbf{x}_i, h_i\}_1^M$, the following linear system of equations is obtained

$$h_i = f(\mathbf{x}_i) = \sum_{j=1}^M \lambda_j \varphi(\|\mathbf{x}_i - \mathbf{x}_j\|) \quad \text{for } \forall i \in \{1, \dots, M\} \quad (11)$$

where λ_j are weights to be computed, see Figure 3 for visual interpretation of (10) or (11) for a $2\frac{1}{2}D$ function.

Equation (11) can be rewritten in a matrix form as

$$\mathbf{A}\boldsymbol{\lambda} = \mathbf{h} \quad (12)$$

where matrix \mathbf{A} is symmetrical, as $\|\mathbf{x}_i - \mathbf{x}_j\| = \|\mathbf{x}_j - \mathbf{x}_i\|$.

The RBF interpolation can be done using "global" or "local" functions. When using "global" radial basis functions the matrix \mathbf{A} will be full, but when using "local" radial basis

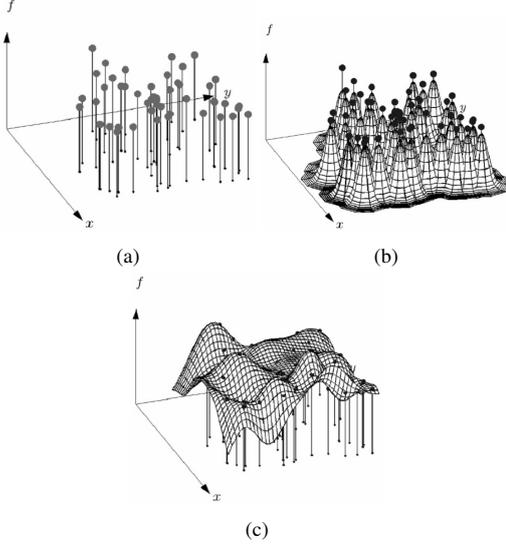


Figure 3: Data values $\{\mathbf{x}_i, \mathbf{h}_i\}_1^M$ (Figure 3a), the RBF collocation functions (Figure 3b), the resulting interpolant (Figure 3c). (From [FW09]).

functions the matrix \mathbf{A} will be sparse, which can be beneficial when solving the system of linear equations $\mathbf{A}\boldsymbol{\lambda} = \mathbf{h}$.

In the case of the vector data, i.e. $\{\mathbf{x}_i, \mathbf{h}_i\}_1^M$ values \mathbf{h}_i are actually vectors, the RBF is to be performed for each coordinate of \mathbf{h}_i .

4. Vector Field RBF Approximation

Vector fields are results of numerical simulations or data measuring process. This kind of vector field data has discrete representation, but an analytical formula describing the vector field is much more useful. We will show how to approximate a vector field using radial basis functions.

A very important feature of a vector field are its critical points. The interpolation must preserve positions and types of all critical points. Thus, the RBF interpolation should interpolate the vector field at all positions of critical points to preserve their positions. To preserve their types, we should include few more points in the neighborhood of each critical point to the interpolation. The number of points in the neighborhood was experimentally chosen to be 4, as more points does not improve the interpolation in any significant way. Points in the neighborhood of a critical point $\mathbf{x}_0 = [x_0, y_0]^T$ are chosen using the following formula

$$\begin{bmatrix} P_x^{(k)} \\ P_y^{(k)} \end{bmatrix} = \begin{bmatrix} x_0 + r \sin(k\frac{\pi}{2}) \\ y_0 + r \cos(k\frac{\pi}{2}) \end{bmatrix}. \quad (13)$$

where $k \in \{0, 1, 2, 3\}$ and r is a small number depending on

the distance of critical points, where the distance to the nearest critical point should be $\gg r$.

This set of critical points together with their neighborhood points can be interpolated using RBF (11), note that each component of vectors $\mathbf{v} = [v_x, v_y]^T$ is interpolated separately. This interpolation will preserve the location of critical points together with their types.

To get more accurate interpolation formula of a vector field at points $\mathbf{x} \in [x_{min}, x_{max}] \times [y_{min}, y_{max}]$ we can include some more random points from this interval into the interpolation. The improvement of quality depending on the number of additionally included points will be shown in the following chapter.

5. Results

The results will be demonstrated on an analytical vector field, as we can measure the interpolation errors precisely. The analytical vector field, that we choose as an example, is described with the following equation

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} x(\frac{1}{2}x^2 + \frac{1}{2}) + y(-x + (\frac{1}{2}y - 1)y + \frac{1}{2}) \\ \frac{1}{2}x^2y + x(-\frac{1}{2}y^2 + y - \frac{1}{2}) + \frac{1}{2}y - 1 \end{bmatrix} \quad (14)$$

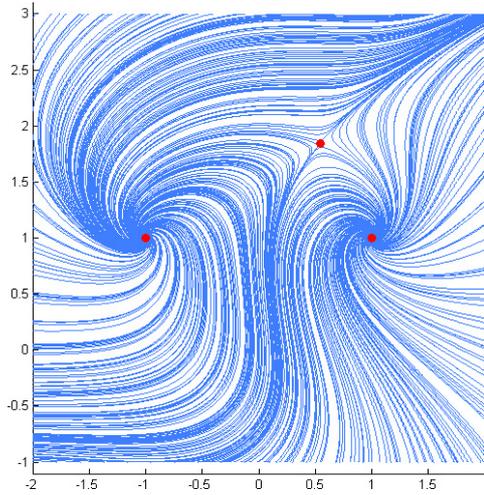
this vector field (14) has three critical points \mathbf{x}_0

$$\begin{aligned} \text{source location: } \quad \mathbf{x}_0 &= [-1, 1]^T \\ \text{source location: } \quad \mathbf{x}_0 &= [1, 1]^T \\ \text{saddle location: } \quad \mathbf{x}_0 &= [0.543689, 1.83929]^T. \end{aligned} \quad (15)$$

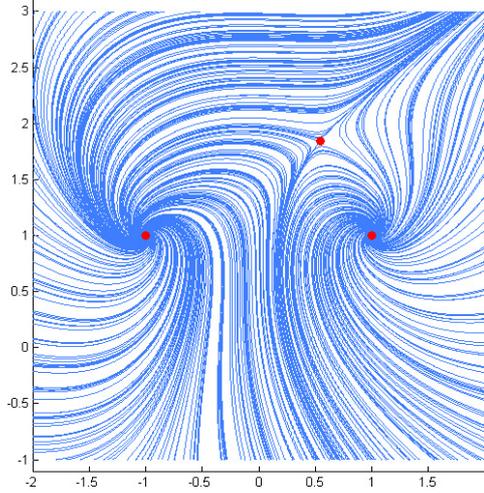
The vector field (14) will be interpolated and tested on interval $[-2, 2] \times [-1, 3]$, as all important features will be visible. The RBF function used for interpolation is a Gauss radial basis function and the shape parameter ϵ was experimentally selected as $\epsilon = 1$.

Vector field (14) can be interpolated using 3 critical point positions and 12 more neighborhood points, i.e. 4 neighborhood points for each critical point. The neighborhood points are computed with (13) and the parameter $r = 0.1$. The v_x component of the vector field is interpolated with one RBF and the v_y component of the vector field is interpolated with one RBF as well. The phase portrait of original analytical vector field (14) is visualized in Figure 4a and the phase portrait of RBF interpolated vector field is visualized in Figure 4b. It can be seen, that both phase portraits look very similar and have the same vector field topology. Moreover, the critical points location is identical, as the average length of displacement error for all critical points is $7.0283 \cdot 10^{-8}$, which is only a numerical error of the critical points location algorithm.

We computed the interpolation error for v_x and v_y and visualized it in Figure 5. It can be seen that the interpolation error is getting higher as the distance from critical



(a)

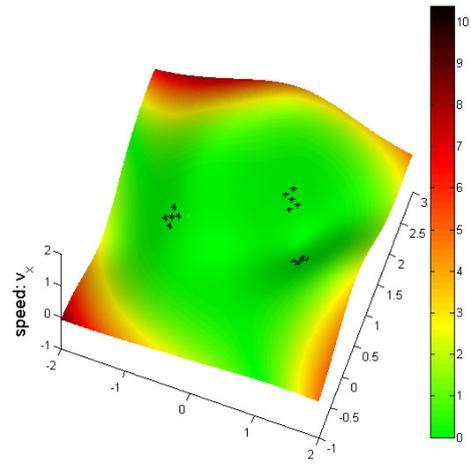


(b)

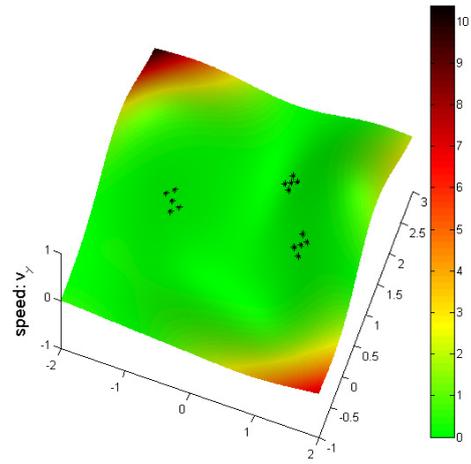
Figure 4: Phase portrait of the vector field (14) (Figure 4a) and phase portrait of a RBF interpolation using only 15 reference points (3 critical points plus three times 4 neighborhood points) (Figure 4b).

points increases. The average error of vector length at interval $[-2, 2] \times [-1, 3]$ is 1.7943 (the vector length varies from 0 to 12.6194) and the average error of vector angular displacement is 0.1966 [rad].

The vector field (14) was interpolated using 3 critical points locations plus three times 4 neighborhood points. We can include few more randomly distributed points into the interpolation to reduce the distance error from (14). We choose to generate additional 85 points from interval $[-2, 2] \times [-1, 3]$, so the interpolation of vector field will contain 10^2 points in total. This interpolation of vector field is



(a)



(b)

Figure 5: Interpolation error of RBF interpolation using only 15 reference points (3 critical points plus three times 4 neighborhood points). Interpolation error of v_x (Figure 5a) and interpolation error of v_y (Figure 5b).

visualized in a phase portrait, see Figure 6 and Figure 4a for comparison with original phase portrait.

We computed the interpolation error for v_x and v_y and visualized it in Figure 7. It can be seen that the interpolation error is close to zero except for locations on the border. The average error of vector length at interval $[-2, 2] \times [-1, 3]$ is 0.0549 (note that the vector length varies from 0 to 12.6194) and the average error of vector angular displacement is 0.0065 [rad].

The average vector length error and the average vector angular displacement error were measured for different number of interpolated points. A number of points k is used as

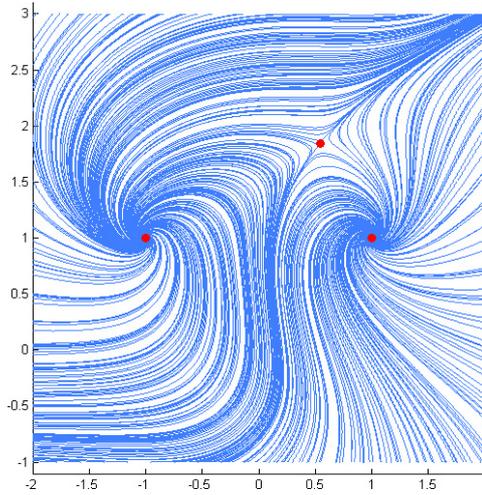


Figure 6: Phase portrait of a vector field RBF interpolation of (14) using 100 reference points (3 critical points plus three times 4 neighborhood points plus 85 randomly distributed points).

added points for the RBF interpolation, thus the RBF interpolation uses $(k + 3 + 3 \cdot 4)$ points for interpolation of vector field, i.e. k randomly distributed points from interval $[-2, 2] \times [-1, 3]$ plus 3 critical points plus three times 4 neighborhood points. Number k was tested from 0 to 400 fifty times for each k with step $\Delta k = 1$ and results are visualized in Figure 8.

It can be seen that both errors in Figure 8 decrease with increasing number k of added points for the interpolation of vector field. According to the required accuracy of the interpolation, the user can select the minimal necessary number of added points and perform the interpolation according to the algorithm proposed.

6. Conclusions

We presented a new and easy to implement approach for the vector field approximation using radial basis functions. In general, it can be used in any d -dimensional space, although the results were presented only for 2D vector field. The proposed RBF interpolation proved the ability to approximate a vector field when preserving the location of critical points and the vector field topology as well.

The proposed approach offers not only analytical description of the discrete data of vector field, but also a significant data compression. This might be a significant feature for "progressive vector field visualization" approach.

In future, the proposed approach will be deeply explored for t -varying data sets together with other aspects for very

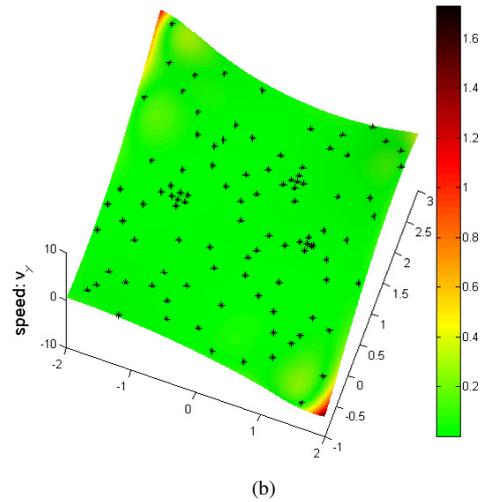
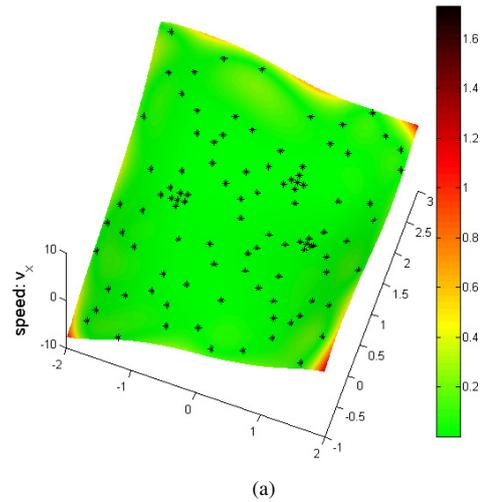


Figure 7: Interpolation error of RBF interpolation using 100 reference points (3 critical points plus three times 4 neighborhood points plus 85 randomly distributed points). Interpolation error of v_x (Figure 7a) and interpolation error of v_y (Figure 7b).

large vector field data set interpolation. The more sophisticated placement of interpolation points around critical points will be deeply explored as well.

Acknowledgement

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

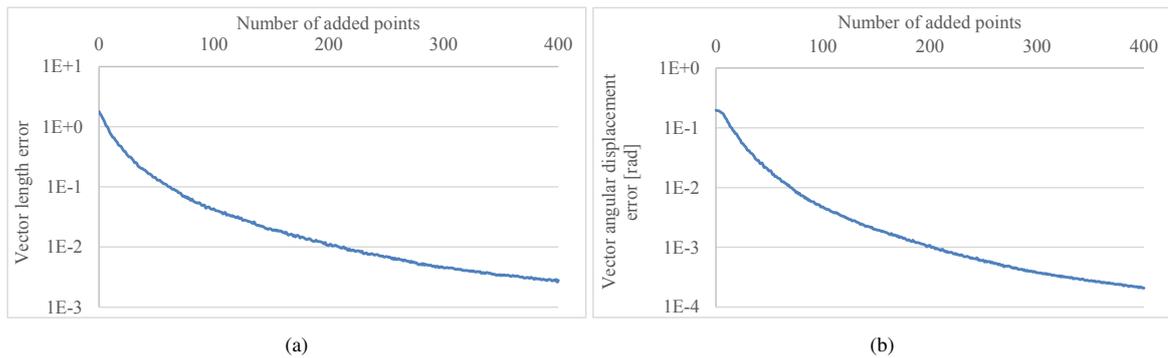


Figure 8: Average errors of the RBF interpolation of vector field (14) using k added reference points, i.e. 3 critical points plus three times 4 neighborhood points plus k randomly distributed points, where $k \in \{0, \dots, 400\}$. The vector field length error, note that the vector length varies from 0 to 12.6194 (Figure 8a) and the vector field angular displacement error (Figure 8b).

References

- [EJF09] EDWARD J. FUSELIER G. B. W.: Stability and error estimates for vector field interpolation and decomposition on the sphere with rbfs. *SIAM Journal on Numerical Analysis* 47, 5 (2009), 3213–3239. 1
- [Fas07] FASSHAUER G. E.: *Meshfree approximation methods with MATLAB*, vol. 6. World Scientific, 2007. 2
- [FP08] FORNBERG B., PIRET C.: On choosing a radial basis function and a shape parameter when solving a convective PDE on a sphere. *J. Comput. Physics* 227, 5 (2008), 2758–2780. 3
- [FW09] FLYER N., WRIGHT G. B.: A radial basis function method for the shallow water equations on a sphere. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* (2009), The Royal Society, pp. rspa–2009. 4
- [Har71] HARDY R. L.: Multiquadric equations of topography and other irregular surfaces. *Journal of geophysical research* 76, 8 (1971), 1905–1915. 3
- [HH89] HELMAN J., HESSELINK L.: Representation and display of vector field topology in fluid flow data sets. *IEEE Computer* 22, 8 (1989), 27–36. 2
- [MZT*14] MA J., ZHAO J., TIAN J., YUILLE A. L., TU Z.: Robust point matching via vector field consensus. *IEEE Transactions on Image Processing* 23, 4 (2014), 1706–1721. 1
- [PS97] PHILIPPOU P. A., STRICKLAND R. N.: Vector field analysis and synthesis using three-dimensional phase portraits. *CVGIP: Graphical Model and Image Processing* 59, 6 (1997), 446–462. 2
- [PS11] PAN R., SKALA V.: A two-level approach to implicit surface modeling with compactly supported radial basis functions. *Engineering with Computers* 27, 3 (2011), 299–307. 2
- [Sch79] SCHAGEN I.: Interpolation in two dimensions—a new technique. *IMA Journal of Applied Mathematics* 23, 1 (1979), 53–59. 3
- [Ska15] SKALA V.: Meshless interpolations for computer graphics, visualization and games. In *Eurographics 2015 - Tutorials, Zurich, Switzerland, May 4-8, 2015* (2015). 2
- [US04] UHLIR K., SKALA V.: Radial basis function use for the restoration of damaged images. In *International Conference on Computer Vision and Graphics, ICCVG 2004, Warsaw, Poland, September 2004, Proceedings* (2004), pp. 839–844. 3
- [Wen06] WENDLAND H.: Computational aspects of radial basis function approximation. *Studies in Computational Mathematics* 12 (2006), 231–256. 3
- [WTS*05] WEINKAUF T., THEISEL H., SHI K., HEGE H., SEIDEL H.: Extracting higher order critical points and topological simplification of 3d vector fields. In *16th IEEE Visualization Conference (VIS 2005), 23-28 October 2005, Minneapolis, MN, USA* (2005), p. 71. 2
- [ZMT06] ZHANG E., MISCHAIKOW K., TURK G.: Vector field design on surfaces. *ACM Trans. Graph.* 25, 4 (2006), 1294–1326. 1

4.10 Vector field RBF interpolation on a sphere

The wind flow on the surface of the Earth is basically the flow on a sphere. Traditional methods Radial basis functions interpolation or approximation on a sphere use the standard Euclidean distance. However, the paper [Smolik and Skala, 2016b] propose a different approach. When interpolating the vector field on a sphere, it is more accurate to use as the distance for radial basis functions, the distance of the shortest path over the surface of the sphere.

The resulting vector field is interpolated more accurately when using the shortest distance over the surface of a sphere compared to the standard Euclidean distance. Moreover, this approach is much more natural and correct.

Citation:

- Michal Smolik and Vaclav Skala. Vector field RBF interpolation on a sphere. In *The Computer Graphics, Visualization, Computer Vision and Image Processing (CGVCVIP)*, pages 352–354. IADIS Press, 2016.

VECTOR FIELD RBF INTERPOLATION ON A SPHERE

Michal Smolik, Vaclav Skala

Faculty of Applied Sciences, University of West Bohemia
Univerzitni 8, CZ 30614 Plzen, Czech Republic

ABSTRACT

This paper presents a new approach for Radial Basis Function (RBF) interpolation on a sphere. Standard approaches use the Euclidian metrics for the distance calculation of two points. However, for interpolation on a sphere, more naturally is computation of the distance as the shortest distance over the surface on a sphere, i.e. spherical distance of two points is more natural for interpolation on a sphere. We present the results on synthetic and real wind vector datasets on a globe.

KEYWORDS

Vector field, Radial Basis Functions, interpolation on sphere, visualization, spherical distance.

1 INTRODUCTION

Interpolation is probably the most frequent operation used in computational methods. Several methods have been developed for data interpolation, but they expect some kind of data “ordering”. Usually, in technical applications, the scattered data are tessellated using triangulation, but this approach is quite prohibitive for the case of k -dimensional data interpolation because of the computational cost.

Interpolating scattered vector data on a surface becomes frequent in applied problem solutions [Turk, G., O'Brien, J.F., 2002]. When the underlying manifold is a sphere, there are applications to geodesy [Aguilar, F. J., et al, 2005], meteorology [Eldrandaly, K. A., Abu-Zaid, M. S., 2011], astrophysics, geophysics, geosciences [Flyer, N. et al, 2014], and other areas. Radial basis function interpolation on a sphere [Golitschek, M. V., Light, W. A., 2001], [Baxter, B. J., Hubbert, S., 2001] has the advantage of having a continuous interpolant all over the sphere, as there are no borders.

2 RADIAL BASIS FUNCTIONS ON A SPHERE

Radial basis functions (RBF) is a technique for scattered data interpolation [Pan, R. and Skala, V., 2011] and approximation [Fasshauer, G.E., 2007].

Radial basis function interpolation can be computed on a sphere and has some advantages. There are no non-physical boundaries and there are no problems with interpolation on the poles, i.e. the sphere has no boundaries, and the vector field can be interpolated on the whole sphere surface at once. The other advantage is that there are no coordinate singularities and the maximal distance of any two points has an upper limit.

The calculation of the distance r between two points \mathbf{x}_1 and \mathbf{x}_2 on a sphere can be computed as the Euclidian distance between these two points

$$r = \|\mathbf{x}_1 - \mathbf{x}_2\| = \sqrt{(\mathbf{x}_1 - \mathbf{x}_2)^T \cdot (\mathbf{x}_1 - \mathbf{x}_2)}. \quad (1)$$

In cases where both points lie on a unit sphere, then $r \in \langle 0; 2 \rangle$.

Another possibility is to compute the distance as the shortest distance between two points \mathbf{x}_1 and \mathbf{x}_2 on the surface of a sphere, measured along the surface of the sphere. The distance is computed using

$$r = \|\mathbf{x}_1 - \mathbf{x}_2\|_{spherical} = \cos^{-1}(\mathbf{n}_1 \cdot \mathbf{n}_2), \quad (2)$$

where $r \in \langle 0; \pi \rangle$ and

$$\mathbf{n}_1 = \frac{\mathbf{x}_1}{\|\mathbf{x}_1\|} \quad \mathbf{n}_2 = \frac{\mathbf{x}_2}{\|\mathbf{x}_2\|}. \quad (3)$$

The distance r in (2) is measured in radians. When the sphere has a radius equal to one, the computed distance in radians is equal to the distance measured along the surface of the sphere.

The RBF interpolation on a sphere is computed using the same formula as standard RBF. The only difference compared to the standard equation for RBF interpolation is when computing the distance between two points, as both of these approaches can be used.

2.1 Example of Vector Field on Sphere on Synthetic data

An example of a vector field on a sphere can be described analytically. This analytical description must fulfill one criteria, which is that this function is continuous all over the sphere. For this purpose, we can use goniometric functions that have a period equal to 2π , i.e.

$$\sin \alpha = \sin(\alpha + k \cdot 2\pi) \quad \cos \alpha = \cos(\alpha + k \cdot 2\pi), \quad (4)$$

where k is an integer, i.e. $k \in \mathbb{Z}$.

The first example of a vector field on a sphere is described using the following equations:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sin 4\delta \\ \cos 4\theta \end{bmatrix} \quad \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sin 3\delta + \cos 4\delta \cdot \cos 3\delta \\ \cos 4\theta - \sin 4\theta \cdot \sin 3\delta \end{bmatrix}. \quad (5)$$

where δ is an azimuth angle, i.e. $\delta \in (-\pi; \pi)$ and θ is a zenith angle, i.e. $\theta \in (0; \pi)$. Data $[u, v]^T$ represents the direction vector on the surface of the sphere at point $[P_x, P_y, P_z]^T$:

$$[P_x \ P_y \ P_z]^T = [\sin \theta \cos \delta \ \sin \theta \sin \delta \ \cos \theta]^T. \quad (6)$$

The vector fields (5) were discretized on uniformly distributed 10 000 points on the sphere and then interpolated using RBF on the sphere with CSRBF with a shape parameter equal to 1:

$$\varphi(r) = (1 - r)_+^4(4r + 1). \quad (7)$$

The interpolation, when using (2) to compute the distance r for basis function $\varphi(r)$, can be seen in Figure 2(a, b). This visualization was created with ray-tracing and line integral convolution on the sphere.

To measure the quality of the interpolation, we can compute the mean error of speed and the mean error of angular displacement of vectors. The mean errors were computed for 10^6 randomly generated positions on the sphere. The results for both equations (5) and both ways of calculating the distance between two points can be seen in Table 1. Note that both vectors $[u, v]^T$ in (5) are computed in $[ms^{-1}]$.

Table 1. Errors of RBF interpolated vector fields (5) on a sphere for both ways of computing distance between two points

		Speed error $[ms^{-1}]$	Angular displacement error $[rad]$
Euclidian distance	vector field (5 left)	$2.452 \cdot 10^{-4}$	$4.233 \cdot 10^{-4}$
	vector field (5 right)	$1.884 \cdot 10^{-3}$	$2.672 \cdot 10^{-3}$
Spherical distance	vector field (5 left)	$1.686 \cdot 10^{-4}$	$3.074 \cdot 10^{-4}$
	vector field (5 right)	$1.379 \cdot 10^{-3}$	$1.906 \cdot 10^{-3}$

It can be seen that the RBF interpolation when using spherical distance gives better results for both vector fields, i.e. more accurate speed and more accurate orientation at every location on the sphere on average, see Table 1. The RBF interpolation is less accurate for the vector field (5 right) than for the vector field (5 left). The reason is that the vector field (5 right) is significantly more complicated than (5 left). The distribution of speed errors and angular displacement errors is visualized in Figure 1. Histograms were created from 10^6 samples and data were grouped into 71 bins.

2.2 Real Example of Vector Field on Sphere on Experiment Data

Numerical forecasts can predict weather as well as wind velocity and direction. For this example, one such prediction of the wind vector field for the whole world [US GFS global weather model] was used. This data contains information about wind speed and wind direction every one degree in latitude and longitude. Therefore, the resolution of the numerically computed dataset is 360×180 , which is 64 800 vectors in total.

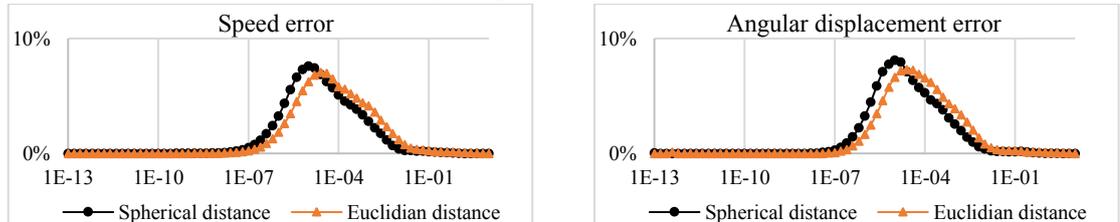


Figure 1. Histogram of speed error distribution (a) and displacement error distribution (b) for a vector field (5 left).

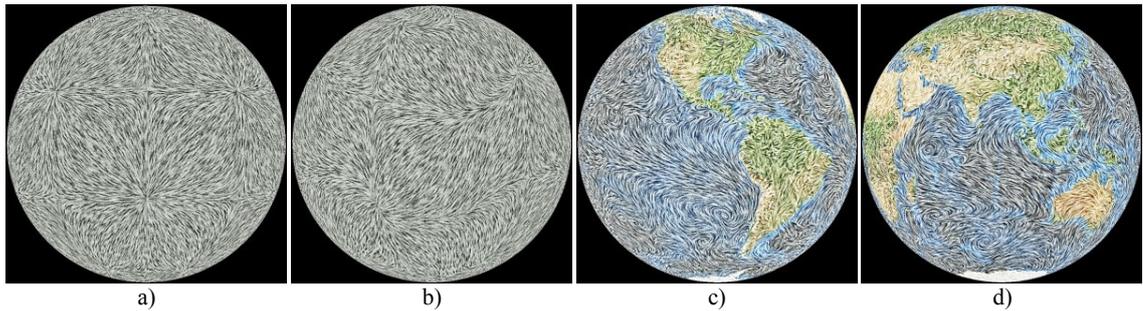


Figure 2. Visualization of examples of vector fields. All vector fields were interpolated using RBF and visualized as LIC images on a sphere. Equations (5 left) (a) and (5 right) (b). Sources, resp. sinks, and saddles are clearly seen in both images. Both images (a, b) are visually identical to the ones with an original analytical description. Visualization of an RBF interpolated wind vector field from numerical simulation (c, d).

Some reduction of this dataset was done, as for the North or South Pole only one vector is needed and for locations near these two poles, the computed vectors can be reduced as well. After the reduction, there were 62 742 vectors. This wind data were interpolated using RBF with CSRBF (7) and shape parameter $\varepsilon = 1$. The RBF interpolation was used to create the visualization of wind vector field on the sphere, Figure 2(c, d).

CONCLUSION

Two approaches for interpolation on a sphere using Radial Basis Functions were presented. The new approach uses the spherical distance as the parameter for the radial basis function computation. The proposed approach gives better results for interpolation on a sphere in comparison to the original standard approach using the Euclidian distance. The proposed method was verified on synthetic analytical datasets and non-trivial real wind datasets of a weather forecast [US GFS global weather model]. In future, the proposed approach will be explored more deeply for t-varying datasets together with aspects of implementation for very large dataset processing.

ACKNOWLEDGEMENT

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

REFERENCES

- Aguilar, F. J., et al., 2005. Effects of terrain morphology, sampling density, and interpolation methods on grid DEM accuracy. *Photogrammetric Engineering & Remote Sensing*, Vol. 71, No. 7, pp. 805-816.
- Baxter, B. J., Hubbert, S., 2001. Radial basis functions for the sphere. *Recent Progress in Multivariate Approximation*, pp. 33-47, Birkhäuser Basel.
- Eldrandaly, K. A., Abu-Zaid, M. S., 2011. Comparison of Six GIS-Based Spatial Interpolation Methods for Estimating Air Temperature in Western Saudi Arabia. *Journal of Environmental Informatics*, Vol. 18, No. 1.
- Fasshauer, G.E., 2007. *Meshfree Approximation Methods with MATLAB*. World Scientific Publ. Co., Inc., NJ, USA.
- Flyer, N. et al., 2014. Radial basis function-generated finite differences: A mesh-free method for computational geosciences. *Handbook of Geomathematics*. Springer, Berlin.
- Golitschek, M. V., Light, W. A., 2001. Interpolation by polynomials and radial basis functions on spheres. *Constructive Approximation*, Vol. 17, No. 1, pp. 1-18.
- Pan, R. and Skala, V., 2011. A Two-level Approach to Implicit Surface Modeling with Compactly Supported Radial Basis Functions. *In Eng. Comput. (Lond.)*, Vol. 27, No. 3, pp. 299-307.
- Turk, G., O'Brien, J. F., 2002. Modelling with implicit surfaces that interpolate. *ACM Transactions on Graphics*, Vol. 21, No. 4, pp. 855-873.
- US GFS global weather model. *National Centers for Environmental Information*, <https://www.ncdc.noaa.gov/data-access/model-data/model-datasets/global-forecast-system-gfs>, [downloaded: 13.1.2016].

4.11 Spherical RBF vector field interpolation – experimental study

The paper [Smolik and Skala, 2017b] is an extension of [Smolik and Skala, 2016b]. It compares different approximation errors and the condition number of interpolation matrix while using different shape parameters for RBF interpolation on a unit sphere. The approach is tested using the local RBF $\varphi_{2,1}$, which is C^2 continuous [Wendland, 1995].

When increasing the number of interpolation points by factor ν , one can increase the shape parameter by factor $\sqrt{\nu}$ to maintain the same ratio between interpolation error and the condition number of interpolation matrix. The optimal shape parameter for RBF interpolation depends on the required accuracy of the interpolation and on the required condition number of the RBF interpolation matrix. However, each point should be covered by some number of radial basis functions, i.e. more than 50 radial basis functions for each point according to results in [Smolik and Skala, 2017b].

Citation:

- Michal Smolik and Vaclav Skala. Spherical RBF vector field interpolation: experimental study. In *2017 IEEE 15th International Symposium on Applied Machine Intelligence and Informatics (SAMi)*, pages 431–434. IEEE, 2017.

Spherical RBF Vector Field Interpolation: Experimental Study

Michal Smolik*, Vaclav Skala*

* University of West Bohemia/Faculty of Applied Sciences, Plzen, Czech Republic
smolik@kiv.zcu.cz, www.vaclavskala.eu

Abstract—The Radial Basis Function (RBF) interpolation is a common technique for scattered data interpolation. We present and test an approach of RBF interpolation on a sphere which uses the spherical distance on the surface of the sphere instead of the Euclidian distance. We show how the interpolation of vector field data depends on the value of shape parameter of RBF and find the optimal shape parameter for our experiments.

Keywords—Radial basis functions; vector field; interpolation; spherical distance; shape parameter.

I. INTRODUCTION

Interpolation is frequently applied operation used in computational methods. Several methods have been developed for data interpolation, but they expect some kind of data “ordering”, e.g. structured mesh, rectangular mesh, unstructured mesh etc. However, in many engineering problems, data are not ordered and they are scattered in k -dimensional space, in general. Often, in technical applications, the scattered data are tessellated using triangulation but this approach is quite prohibitive for the case of k -dimensional data interpolation because of the computational cost, i.e. if data is large.

Interpolating scattered vector data on a surface becomes frequent in applied problem solutions. When the underlying manifold is a sphere, there are applications to geodesy [1], meteorology [3], astrophysics, geophysics, geosciences [4] and other areas. The radial basis function interpolation on a sphere [5] has the advantage of continuous interpolant all over the sphere, as there are no borders.

II. VECTOR FIELD

Vector fields on surfaces are important objects which appear frequently in scientific simulation in CFD (Computational Fluid Dynamics) [10] or modelling by FEM (Finite Element Method). To be visualized [6], such vector fields are usually linearly approximated for the sake of simplicity and performance considerations.

The vector field can be easily analyzed when having an approximation of the vector field near some location point. The important places to be analyzed are so called critical points. Analyzing the vector field behavior near these points gives us the information about the characteristic of the vector field.

A. Critical Point

Critical points (\mathbf{x}_0) of the vector field are points where the magnitude of the vector vanishes

$$\frac{d\mathbf{x}}{dt} = \mathbf{v}(\mathbf{x}) = \mathbf{0}, \quad (1)$$

i.e. all components are equal to zero

$$\begin{bmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \\ \frac{dz}{dt} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \quad (2)$$

A critical point is said to be isolated, or simple, if the vector field is non-vanishing in an open neighborhood around the critical point. Thus, for all surrounding points \mathbf{x}_ε of the critical point \mathbf{x}_0 the equation (1) does not apply, i.e.

$$\frac{d\mathbf{x}_\varepsilon}{dt} \neq \mathbf{0}. \quad (3)$$

At critical points, the direction of the field line is indeterminate, and they are the only points in the vector field where field lines, e.g. stream lines in a CFD dataset, can intersect (asymptotically). The terms singular point, null point, neutral point or equilibrium point are also frequently used to describe critical points.

Critical points deliver important information about the overall characteristics of a vector field because together with the nearby surrounding vectors, they have more information encoded in them than any such group in the vector field, regarding the total behavior of the field.

III. VECTOR FIELD INTERPOLATION

The RBF interpolation was originally introduced by [7] and is based on computing the distance of two points in the k -dimensional space and is defined by a function

$$f(\mathbf{x}) = \sum_{j=1}^M \lambda_j \varphi(\|\mathbf{x} - \mathbf{x}_j\|), \quad (4)$$

where λ_j are weights of the RBF, M is number of radial basis functions, i.e. number of reference points, and φ is the radial basis function.

The radial basis function interpolation can be computed on a sphere and has some advantages [2], [8]. There are any unphysical boundaries and there are no problems with interpolation on the poles, i.e. the sphere has no boundaries, and the vector field can be interpolated on the whole sphere surface at once compared to using only spherical coordinates and interpolation in $2D$. The other advantage is that there are no coordinate singularities and the maximal distance of any two points has an upper bound and the RBF interpolation does not need any mesh, i.e. triangulation, for interpolation.

The RBF interpolation interpolates scalar values on a sphere. However the vector field is not a scalar field, the RBF

interpolation can be used for vector fields as well. For each component of the vector, we need to compute one RBF interpolation separately but it should be noted that the interpolation matrices for all component of the vector are the same.

The calculation of the distance r between two points \mathbf{x}_1 and \mathbf{x}_2 on a sphere can be computed as the Euclidian distance between this two points

$$\begin{aligned} r &= \|\mathbf{x}_1 - \mathbf{x}_2\|_{Euclidian} \\ &= \sqrt{(\mathbf{x}_1 - \mathbf{x}_2)^T \cdot (\mathbf{x}_1 - \mathbf{x}_2)}. \end{aligned} \quad (5)$$

In the case that both points lie on a unit sphere then $r \in \langle 0; 2 \rangle$.

Or the distance can be computed as the shortest distance between two points \mathbf{x}_1 and \mathbf{x}_2 on the surface of a sphere, measured along the surface of the sphere. The distance is computed using

$$r = \|\mathbf{x}_1 - \mathbf{x}_2\|_{spherical} = \cos^{-1}(\mathbf{n}_1 \cdot \mathbf{n}_2), \quad (6)$$

where $r \in \langle 0; \pi \rangle$ and

$$\mathbf{n}_1 = \frac{\mathbf{x}_1}{\|\mathbf{x}_1\|} \quad \mathbf{n}_2 = \frac{\mathbf{x}_2}{\|\mathbf{x}_2\|} \quad (7)$$

The distance r in (6) is measured in radians. In the case that the sphere has radius equal to one, the computed distance in radians is equal to the distance measured along the surface of the sphere.

The RBF interpolation performs slightly better interpolation results when using spherical distance (6) compared to the RBF with the Euclidian distance calculation (5). For this reason we use only the spherical distance calculation for all our tests.

IV. RESULTS

For experimental verification of the proposed approach an analytical function is used.

An example of a vector field on a sphere can be described analytically. The analytical function has to fulfill a criterion which is that the function must be continuous all over the sphere including wrapping. For this purpose we can use goniometric functions that have periodicity equal to 2π , i.e. for example a vector field with the following formula

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sin 3\delta + \cos 4\delta \cdot \cos 3\delta \\ \cos 4\theta - \sin 4\theta \cdot \sin 3\delta \end{bmatrix}, \quad (8)$$

where δ is an azimuth angle, i.e. $\delta \in (-\pi; \pi)$ and θ is a zenith angle, i.e. $\theta \in \langle 0; \pi \rangle$. Vector $[u, v]^T$ represents a directional vector in the spherical coordinates on the surface of a sphere at point $\mathbf{P} = [P_x, P_y, P_z]^T$

$$\begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} = \begin{bmatrix} \sin \theta \cos \delta \\ \sin \theta \sin \delta \\ \cos \theta \end{bmatrix}. \quad (9)$$

The vector field (8) was discretized by 10 000 uniformly distributed points on a surface of a sphere and then interpolated using RBF on sphere with Compact-Support-RBFs (CSRBF) [9]

$$\varphi(r) = \begin{cases} (1 - \varepsilon r)^4 (4\varepsilon r + 1) & \varepsilon r \leq 1 \\ 0 & \varepsilon r > 1 \end{cases}, \quad (10)$$

where ε is a shape parameter and r is the distance measured over the surface of a sphere.

We computed the RBF interpolation on a sphere of the original vector field (8) using 10^3 , $5 \cdot 10^3$ and 10^4 sampling points for different shape parameters and measured the average vector length error and the average angular displacement error of interpolated vectors. The shape parameter ε cannot be less than $1/\pi$, as the CSRBF with $\varepsilon = 1/\pi$ covers the whole surface of a unit sphere, i.e. the CSRBF with shape parameter $\varepsilon > 1/\pi$ covers only a part of the sphere surface.

The vector length error is computed using the formula

$$err_l = \frac{\sum_{i=1}^N \|\tilde{\mathbf{v}}_i\| - \|\mathbf{v}_i\|}{\sum_{i=1}^N \|\mathbf{v}_i\|}, \quad (11)$$

where $\tilde{\mathbf{v}}_i$ is the interpolated vector and \mathbf{v}_i is the vector computed from the analytical function (8). The vector length error is visualized in Figure 1. It can be seen that the average error is almost identical for shape parameter $\varepsilon \in \langle 1/\pi; 4 \rangle$ for $5 \cdot 10^3$ and 10^4 sampling points and for larger shape parameters the error increases. The vector length error for 10^3 sampling points is slightly higher than for $5 \cdot 10^3$ and 10^4 sampling points and starts distinctly increasing for shape parameter $\varepsilon > 2$.

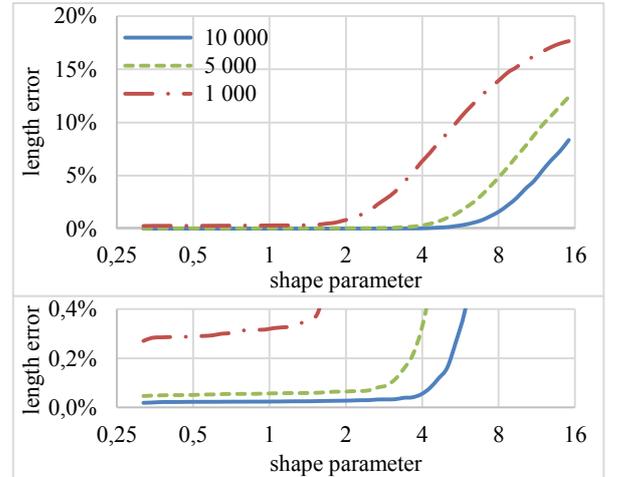


Figure 1: Average error of vector lengths of the RBF interpolation on a sphere for different shape parameters and different numbers of interpolated points.

The average angular displacement error is computed using the formula

$$err_\varphi = \frac{\sum_{i=1}^N \cos^{-1}(\tilde{\mathbf{v}}_i \cdot \mathbf{v}_i)}{N} \cdot \frac{180}{\pi}. \quad (12)$$

The results of the average angular displacement error are visualized in Figure 2. The progress of the error is similar to Figure 1 and, thus, the quality of the vector field interpolation is almost identical for shape parameters $\varepsilon \in \langle 1/\pi; 4 \rangle$ for $5 \cdot 10^3$ and 10^4 sampling points and for larger shape parameters the error increases. The angular displacement error for 10^3

sampling points is slightly higher than for $5 \cdot 10^3$ and 10^4 sampling points and starts distinctly increasing for shape parameter $\varepsilon > 2$.

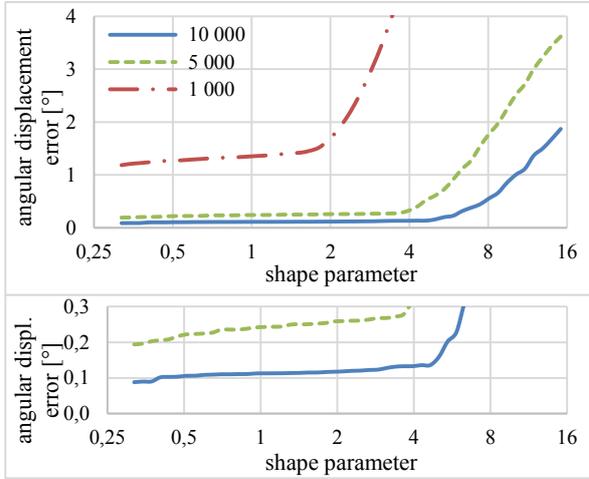


Figure 2: Average angular displacement error [°] of vectors of RBF interpolation on a sphere for different shape parameters and different numbers of interpolated points.

The CSRBF (10) is a “local” radial basis function, therefore, the RBF interpolation matrix is sparse. We varied the shape parameter and measured the occupancy of the interpolation matrix. The results can be seen in Figure 4. When the shape parameter is $\varepsilon > 2/\pi$ then more than half of the elements in the RBF interpolation matrix are equal zero.

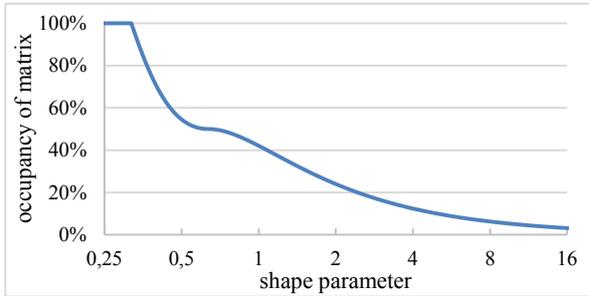


Figure 4: Occupancy of the interpolation matrix for the RBF interpolation on a unit sphere for different shape parameters.

The RBF interpolation matrix has different condition numbers for different shape parameters because the occupancy of the matrix changes for different shape parameters. The condition number of this matrix is visualized in Figure 5 and it can be seen that the matrix is better conditioned with increasing shape parameter. It is justified by the fact that the occupancy of the RBF interpolation matrix decreases for increasing shape parameter.

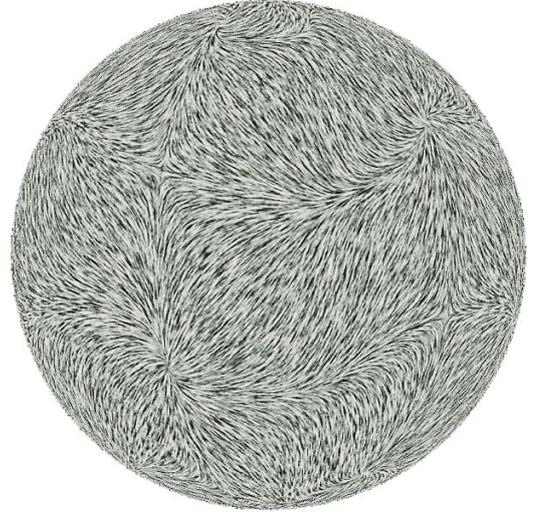


Figure 3: Line integral convolution visualization of the RBF interpolated vector field (8).

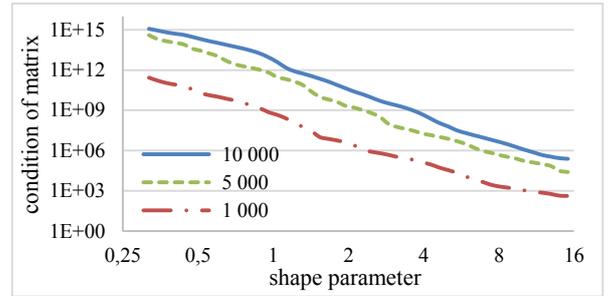


Figure 5: Condition of the RBF interpolation matrix for different shape parameters and different numbers of interpolated points.

We performed another series of tests with a different CSRBF function as well. The second CSRBF used for our tests is

$$\varphi(r) = \begin{cases} (1 - \varepsilon r)^3 (3\varepsilon r + 1) & \varepsilon r \leq 1 \\ 0 & \varepsilon r > 1 \end{cases}, \quad (13)$$

We performed the same tests as for (10). The results of average vector field length error are visualized in the following graph, i.e. Figure 6.

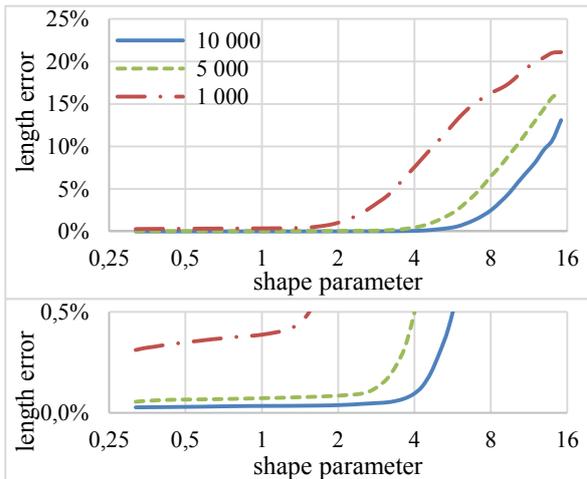


Figure 6: Average error of vector lengths of the RBF interpolation on a sphere for different shape parameters and different numbers of interpolated points.

The results of average angular displacement error are visualized in the following graph, i.e. Figure 7.

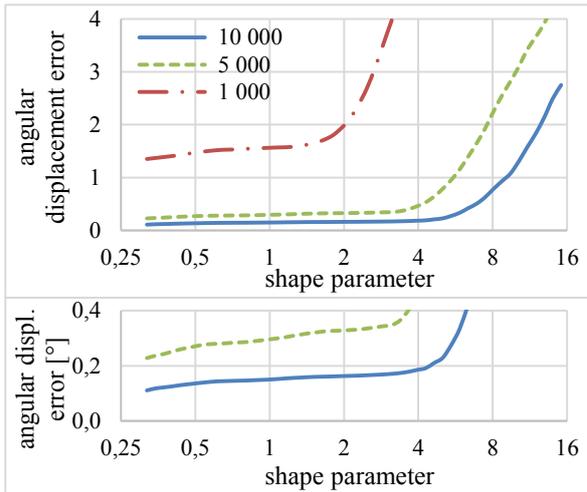


Figure 7: Average angular displacement error [°] of vectors of RBF interpolation on a sphere for different shape parameters and different numbers of interpolated points.

It can be seen, that the results for both CSRBF used in our tests are very similar. The only difference is that the RBF interpolation on a sphere using (10) performs slightly better than using (13).

Using all the previous results we can choose the best shape parameter to be $\varepsilon = 4$ for $5 \cdot 10^3$ and 10^4 sampling points and $\varepsilon = 2$ for 10^3 sampling points. For this parameters the interpolation errors are the smallest, the RBF interpolation matrix is sparse and has a rather small condition number. For $\varepsilon > 4$, resp. $\varepsilon > 2$, will increase both interpolation errors and for $\varepsilon < 4$, resp. $\varepsilon > 2$, will increase the occupancy and the condition number of RBF interpolation matrix.

The RBF interpolated vector field (8) on a unit sphere was visualized using the line integral convolution, see Figure 3. The important property of the interpolated vector field is that for all

shape parameters $\varepsilon < 4$ it preserves the type of all critical points in the vector field (8). And the location of all critical points in the interpolated vector field is almost identical to the locations of the critical points in the vector field (8). Thus the RBF interpolated vector field has the same topology as the vector field (8).

V. CONCLUSION

We presented an approach for vector field interpolation using radial basis functions on a sphere. The distance between two points is computed over the surface, as it is more natural and the interpolation is more accurate. The presented experiments showed how the interpolation error, the matrix occupancy, and the condition number of the interpolation matrix depends on the value of the shape parameter of the RBF.

VI. ACKNOWLEDGMENTS

The authors would like to thank their colleagues at the University of West Bohemia, Plzen, for their discussions and suggestions, and the anonymous reviewers for their valuable comments and hints provided. The research was supported by MSMT CR projects LH12181, SGS 2016-013 and GA17-05534S.

VII. REFERENCES

- [1] Aguilar, F. J., et al. Effects of terrain morphology, sampling density, and interpolation methods on grid DEM accuracy. *Photogrammetric Engineering & Remote Sensing*, Vol. 71, No. 7, pp. 805-816, 2005.
- [2] Baxter, B. J., Hubbert, S. Radial basis functions for the sphere. *Recent Progress in Multivariate Approximation*, pp. 33-47, Birkhäuser Basel, 2001.
- [3] Eldrandaly, K. A., Abu-Zaid, M. S. Comparison of Six GIS-Based Spatial Interpolation Methods for Estimating Air Temperature in Western Saudi Arabia. *Journal of environmental Informatics*, Vol. 18, No. 1, 2011.
- [4] Flyer, N. et al. Radial basis function-generated finite differences: A mesh-free method for computational geosciences. *Handbook of Geomathematics*. Springer, Berlin, 2014.
- [5] Golitschek, M. V., Light, W. A. Interpolation by polynomials and radial basis functions on spheres. *Constructive Approximation*, Vol. 17, No. 1, pp. 1-18, 2001.
- [6] Günther, T., Theisel, H. Inertial Steady 2D Vector Field Topology, *Computer Graphics Forum (Proc. Eurographics)*, Vol. 35, No. 2, 2016.
- [7] Hardy, R. L. Multiquadric equations of topography and other irregular surfaces. *Journal of geophysical research*, Vol. 76, No. 8, pp. 1905-1915, 1971.
- [8] Hubbert, S., Lê Gia, Q. T., Morton, T. M. Spherical radial basis functions, theory and applications. ISBN: 978-3-319-17938-4, Springer, 2015.
- [9] Wendland, H.: Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Advances in computational Mathematics*, Vol. 4, No. 1, pp. 389-396, 1995.
- [10] Zanino, R., et al. Computational Fluid Dynamics (CFD) analysis of the Helium inlet mock-up for the ITER TF superconducting magnets. *Applied Superconductivity, IEEE Transactions on*, Vol. 24, No. 3, pp. 1-5, 2014.

4.12 A comparative study of LOWESS and RBF approximations for visualization

A similar method for data approximation as the RBF is the LOWESS approximation. The paper [Smolik et al., 2016] compares this two approaches and gives recommendations when using each one from this two approximation methods.

The LOWESS approximation is more appropriate for computing the approximation value in only few locations. When using the RBF approximation in the same manner as the LOWESS, the RBF approximation gives worse results. However, when computing the approximation value in many locations, it is more useful to use the RBF approximation of all input points at once as the final approximation has lower approximation error compared to the use of LOWESS.

Moreover, in higher dimensions is the time complexity of LOWESS approximation increased by the time complexity of finding k -nearest neighbor points. Thus the RBF approximation is more convenient in higher dimensions.

Citation:

- Michal Smolik, Vaclav Skala, and Ondrej Nedved. A comparative study of LOWESS and RBF approximations for visualization. In *International Conference on Computational Science and Its Applications*, pages 405–419. Springer, 2016.

A Comparative Study of LOWESS and RBF Approximations for Visualization

Michal Smolik¹, Vaclav Skala¹ and Ondrej Nedved¹

¹ Faculty of Applied Sciences, University of West Bohemia,
Univerzitni 8, CZ 30614 Plzen, Czech Republic

Abstract. Approximation methods are widely used in many fields and many techniques have been published already. This comparative study presents a comparison of LOWESS (Locally weighted scatterplot smoothing) and RBF (Radial Basis Functions) approximation methods on noisy data as they use different approaches. The RBF approach is generally convenient for high dimensional scattered data sets. The LOWESS method needs finding a subset of nearest points if data are scattered. The experiments proved that LOWESS approximation gives slightly better results than RBF in the case of lower dimension, while in the higher dimensional case with scattered data the RBF method has lower computational complexity.

Keywords: Radial Basis Functions, LOWESS, Approximation

Notation used

D : dimension

K : k -nearest points

M : number of radial basis functions for approximation

N : number of all input points

R : number of points at which the approximation is calculated

ξ : point where to calculate the approximation

d : degree of polynomial

r : $r = d + 2$

q : $q = d + 1$

1 Introduction

Interpolation and approximation techniques are often used in data processing. Approximation methods of values y_i in the given $\{(x_i, y_i)\}_1^N$ data set lead to a smooth function which minimizes the difference between given data and the determined function [13]. It can be used for visualization of noisy data [1, 2], visualization of the basic shape of measured/calculated data [9], for prediction, and other purposes. Many methods have been described together with their properties. This paper describes LOWESS (Locally weighted scatterplot smoothing) and RBF (Radial basis functions) methods and their experimental comparison.

2 LOWESS

The locally weighted scatterplot smoothing method (LOWESS) [3] is often used, especially in statistical applications. The value of an approximated function at a point x_0 is calculated from the formula of a curve which minimizes a sum S in the k -nearest neighborhood (KNN) points of the given point ξ .

$$S = \sum_{i=1}^K \omega_i \cdot (y_i - P_{(d)}(x_i))^2, \quad (1)$$

where $P_{(d)}(x) = a_0 + a_1x + a_2x^2 + \dots + a_dx^d$ is a d degree of a polynomial function with unknown coefficients $\mathbf{a} = [a_0, a_1, a_2, \dots, a_d]^T$. We can rewrite the sum S in a matrix form as:

$$S = (\mathbf{b} - \mathbf{A}\mathbf{a})^T \cdot \mathbf{W} \cdot (\mathbf{b} - \mathbf{A}\mathbf{a}), \quad (2)$$

where $\mathbf{b} = [y_1, y_2, \dots, y_K]^T$ is a vector of function values, matrix \mathbf{A} is equal to:

$$\mathbf{A} = \begin{bmatrix} 1 & x_1 & \dots & x_1^d \\ 1 & x_2 & \dots & x_2^d \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_K & \dots & x_K^d \end{bmatrix} \quad (3)$$

and matrix \mathbf{W} is a diagonal matrix:

$$\mathbf{W} = \begin{bmatrix} \omega(\|x_1 - \xi\|) & & & 0 \\ & \omega(\|x_2 - \xi\|) & & \\ & & \ddots & \\ 0 & & & \omega(\|x_K - \xi\|) \end{bmatrix} = \begin{bmatrix} \omega_1 & & & 0 \\ & \omega_2 & & \\ & & \ddots & \\ 0 & & & \omega_K \end{bmatrix}, \quad (4)$$

where $\omega(r)$ are weighting functions, which have to satisfy the following conditions defined as:

$$\forall a, b \in [0; 1], a < b : \omega(a) \geq \omega(b) \wedge \omega(0) = 1 \wedge \forall c \geq 1 : \omega(c) = 0. \quad (5)$$

One such example of a weighting function ω can be the tricube function:

$$\omega(r = \|x_i - \xi\|) = \omega_i = \begin{cases} (1 - r^3)^3 & r \in \langle 0; 1 \rangle \\ 0 & r > 1 \end{cases}. \quad (6)$$

Equation (2) can be modified as:

$$\begin{aligned} S &= \mathbf{b}^T \mathbf{W} \mathbf{b} - \mathbf{b}^T \mathbf{W} \mathbf{A} \mathbf{a} - (\mathbf{A} \mathbf{a})^T \mathbf{W} \mathbf{b} + (\mathbf{A} \mathbf{a})^T \mathbf{W} \mathbf{A} \mathbf{a} \\ &= \mathbf{b}^T \mathbf{W} \mathbf{b} - \mathbf{b}^T \mathbf{W} \mathbf{A} \mathbf{a} - \mathbf{a}^T \mathbf{A}^T \mathbf{W} \mathbf{b} + \mathbf{a}^T \mathbf{A}^T \mathbf{W} \mathbf{A} \mathbf{a}. \end{aligned} \quad (7)$$

The sum S is minimal if the partial derivative of S with respect to \mathbf{a} is equal to zero:

$$\frac{\partial S}{\partial \mathbf{a}} = -(\mathbf{b}^T \mathbf{W} \mathbf{A})^T - \mathbf{A}^T \mathbf{W} \mathbf{b} + 2\mathbf{A}^T \mathbf{W} \mathbf{A} \mathbf{a} = \mathbf{0} \quad (8)$$

as $\mathbf{W} = \mathbf{W}^T$ and therefore:

$$\begin{aligned} \mathbf{A}^T \mathbf{W} \mathbf{A} \mathbf{a} &= \mathbf{A}^T \mathbf{W} \mathbf{b} \\ \mathbf{a} &= (\mathbf{A}^T \mathbf{W} \mathbf{A})^{-1} \mathbf{A}^T \mathbf{W} \mathbf{b}. \end{aligned} \quad (9)$$

The numerical stability of calculations is influenced by the position of the interval of the k -nearest neighborhood points of the point ξ . The LOWESS approximation is “locally” based, as only k -nearest points are used and thus r is actually computed as $r = \|x_i - \xi\|$. To solve problems with the numerical stability of calculations and independence of absolute position, we have to use relative position of all the k -nearest neighborhood points of the point ξ such that the matrix \mathbf{A} from (3) is defined as:

$$\mathbf{A} = \begin{bmatrix} 1 & (x_1 - \xi) & \cdots & (x_1 - \xi)^d \\ 1 & (x_2 - \xi) & \cdots & (x_2 - \xi)^d \\ \vdots & \vdots & \ddots & \vdots \\ 1 & (x_K - \xi) & \cdots & (x_K - \xi)^d \end{bmatrix} \quad (10)$$

2.1 LOWESS with linear regression

Linear regression, i.e. choosing $d = 1$, appears to strike a good balance between computational simplicity and the flexibility needed to reproduce patterns in the data. In such a case, we can rewrite (9) as:

$$\mathbf{a} = \begin{bmatrix} \sum_{i=1}^K \omega_i & \sum_{i=1}^K \omega_i x_i \\ \sum_{i=1}^K \omega_i x_i & \sum_{i=1}^K \omega_i x_i^2 \end{bmatrix}^{-1} \cdot \begin{bmatrix} \sum_{i=1}^K \omega_i y_i \\ \sum_{i=1}^K \omega_i x_i y_i \end{bmatrix} \quad (11)$$

and after some adjustments we can get a final formula for unknown coefficients \mathbf{a} :

$$\begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \frac{1}{(\sum_{i=1}^K \omega_i) \cdot (\sum_{i=1}^K \omega_i x_i^2) - (\sum_{i=1}^K \omega_i x_i)^2} \cdot \begin{bmatrix} \left(\sum_{i=1}^K \omega_i y_i \right) \left(\sum_{i=1}^K \omega_i x_i^2 \right) - \left(\sum_{i=1}^K \omega_i x_i \right) \left(\sum_{i=1}^K \omega_i x_i y_i \right) \\ - \left(\sum_{i=1}^K \omega_i y_i \right) \left(\sum_{i=1}^K \omega_i x_i \right) + \left(\sum_{i=1}^K \omega_i \right) \left(\sum_{i=1}^K \omega_i x_i y_i \right) \end{bmatrix} \quad (12)$$

2.2 LOWESS with constant regression

Constant regression, i.e. choosing $d = 0$, is the most computationally simple, but from a practical point of view, an assumption of local linearity seems to serve far better than an assumption of local constancy because the tendency is to plot variables that are related to one another. Thus, the linear LOWESS regression produces better results than

the constant LOWESS regression, which is very simple. In this case, we can rewrite it from (9) as:

$$a_0 = \frac{\sum_{i=1}^K \omega_i y_i}{\sum_{i=1}^K \omega_i}. \quad (13)$$

Comparing formulas from (13) and (12), it can be seen that LOWESS with constant regression is computationally much easier than LOWESS with linear regression.

3 Radial Basis Functions

Radial basis functions (RBF) [4, 11, 12] is based on distances, generally in D -dimensional space. The value of an approximated function at a point \mathbf{x} is calculated from the formula:

$$f(x) = \sum_{i=1}^M \lambda_i \Phi(\|x - \xi_i\|) + P_d(x), \quad (14)$$

where $P_{(d)}(x) = a_0 + a_1 x + a_2 x^2 + \dots + a_d x^d$ is a d degree polynomial function with unknown coefficients $\mathbf{a} = [a_0, a_1, a_2, \dots, a_d]^T$, M is the number of radial basis functions, and $\boldsymbol{\lambda} = [\lambda_1, \dots, \lambda_M]$ are weights of radial basis functions $\Phi(\|x - \xi_i\|)$. The function Φ is a real-valued function whose value depends only on the distance from some other point ξ_i , called a center, so that:

$$\Phi_i(x) = \Phi(\|x - \xi_i\|). \quad (15)$$

As the values $f(x_i)$ at a point x_i are known, equation (14) represents a system of linear equations that has to be solved in order to determine coefficients $\boldsymbol{\lambda}$ and \mathbf{a} , i.e.

$$f(x_j) = \sum_{i=1}^M \lambda_i \Phi(\|x_j - \xi_i\|) + P_d(x_j) \text{ for } \forall j \in \{1, \dots, N\}. \quad (16)$$

Using matrix notation we can rewrite (16) as:

$$\begin{bmatrix} \Phi(\|x_1 - \xi_1\|) & \dots & \Phi(\|x_1 - \xi_M\|) & 1 & x_1 & \dots & x_1^d \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ \Phi(\|x_N - \xi_1\|) & \dots & \Phi(\|x_N - \xi_M\|) & 1 & x_N & \dots & x_N^d \end{bmatrix} \cdot \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_M \\ a_0 \\ \vdots \\ a_d \end{bmatrix} = \begin{bmatrix} f(x_1) \\ \vdots \\ f(x_N) \end{bmatrix}. \quad (17)$$

We can create a “simple” RBF formula, see (18), using (17) with only one radial basis function, i.e. $M = 1$. This formula can be used in the same manner as the LOWESS method for calculating approximated value at the point ξ , using only the k -nearest neighborhood points of the point ξ , which is the center of radial basis function $\phi(\|x - \xi\|)$, too.

$$\begin{bmatrix} \Phi(\|x_1 - \xi\|) & 1 & x_1 & \cdots & x_1^d \\ \vdots & \vdots & \vdots & & \vdots \\ \Phi(\|x_K - \xi\|) & 1 & x_K & \cdots & x_K^d \end{bmatrix} \cdot \begin{bmatrix} \lambda_1 \\ a_0 \\ \vdots \\ a_d \end{bmatrix} = \begin{bmatrix} f(x_1) \\ \vdots \\ f(x_K) \end{bmatrix} \rightarrow \mathbf{A} \cdot \boldsymbol{\lambda} = \mathbf{f}. \quad (18)$$

The coefficients $\boldsymbol{\eta} = [\lambda_1, \mathbf{a}^T]^T$ in overdetermined system of linear equations (18) are computed by the least squares error method:

$$\boldsymbol{\eta} = (\mathbf{A}^T \mathbf{A})^{-1} \cdot (\mathbf{A}^T \mathbf{f}). \quad (19)$$

As the numerical stability of calculations is influenced by the position of the interval of the k -nearest neighborhood points of the point ξ and the RBF approximation is “locally” based, only k -nearest points are used. To solve problems with the numerical stability of calculations, we have to move all the k -nearest neighborhood points of the point ξ such that the matrix \mathbf{A} from (18) is defined as:

$$\mathbf{A} = \begin{bmatrix} \Phi(\|x_1 - \xi\|) & 1 & (x_1 - \xi) & \cdots & (x_1 - \xi)^d \\ \vdots & \vdots & \vdots & & \vdots \\ \Phi(\|x_K - \xi\|) & 1 & (x_K - \xi) & \cdots & (x_K - \xi)^d \end{bmatrix} \quad (20)$$

and $f(x)$ is defined as:

$$f(x) = \lambda_1 \Phi(\|x - \xi\|) + P_d(x - \xi). \quad (21)$$

For locally-based approximation, any compactly supported radial basis function (CSRBF) [8, 12] can be used. CSRBF is a function defined on $r \in \langle 0; 1 \rangle$, is equal to 0 for all $r > 1$, and has to satisfy the conditions in (5). In the tests presented here, the $\Phi(r)$ function was selected as:

$$\Phi(r) = \begin{cases} (1 - r^3)^3 & r \in \langle 0; 1 \rangle \\ 0 & r > 1 \end{cases}, \quad (22)$$

which is exactly the same function as weighting function (6) for LOWESS approximation.

3.1 Simplified RBF with a constant polynomial

Choosing $d = 0$, we will get a polynomial of zero degree which is only a constant, i.e.; $P_d = a_0$.

$$\begin{bmatrix} \Phi(\|x_1 - \xi\|) & 1 \\ \vdots & \vdots \\ \Phi(\|x_K - \xi\|) & 1 \end{bmatrix} \cdot \begin{bmatrix} \lambda_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} f(x_1) \\ \vdots \\ f(x_K) \end{bmatrix} \rightarrow \mathbf{A} \cdot \boldsymbol{\eta} = \mathbf{f}. \quad (23)$$

It leads to overdetermined system of linear equations. Using the method of least squares, we can calculate $\boldsymbol{\eta}$:

$$\boldsymbol{\eta} = (\mathbf{A}^T \mathbf{A})^{-1} \cdot (\mathbf{A}^T \mathbf{f}), \quad (24)$$

where $\boldsymbol{\eta} = [\lambda_1, a_0]^T$.

$$\begin{bmatrix} \lambda_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^K (\Phi(\|x_i - \xi\|))^2 & \sum_{i=1}^K \Phi(\|x_i - \xi\|) \\ \sum_{i=1}^K \Phi(\|x_i - \xi\|) & \sum_{i=1}^K 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} \sum_{i=1}^K \Phi(\|x_i - \xi\|) \cdot f(x_i) \\ \sum_{i=1}^K f(x_i) \end{bmatrix}, \quad (25)$$

where $\sum_{i=1}^K 1 = K$ and after adjustments:

$$\begin{bmatrix} \lambda_1 \\ a_0 \end{bmatrix} = \frac{1}{\left(\sum_{i=1}^K (\Phi(\|x_i - \xi\|))^2\right) \cdot K - \left(\sum_{i=1}^K \Phi(\|x_i - \xi\|)\right)^2} \cdot \begin{bmatrix} K & -\sum_{i=1}^K \Phi(\|x_i - \xi\|) \\ -\sum_{i=1}^K \Phi(\|x_i - \xi\|) & \sum_{i=1}^K (\Phi(\|x_i - \xi\|))^2 \end{bmatrix} \cdot \begin{bmatrix} \sum_{i=1}^K \Phi(\|x_i - \xi\|) \cdot f(x_i) \\ \sum_{i=1}^K f(x_i) \end{bmatrix}. \quad (26)$$

The value $f(\xi)$ is calculated as:

$$f(\xi) = \lambda_1 \Phi(\|\xi - \xi\|) + a_0 = \lambda_1 \Phi(0) + a_0. \quad (27)$$

3.2 Simplified RBF without a polynomial

In the case of using simplified RBF without polynomial P_d , we get the following equation:

$$\begin{bmatrix} \Phi(\|x_1 - \xi\|) \\ \vdots \\ \Phi(\|x_K - \xi\|) \end{bmatrix} \cdot [\lambda_1] = \begin{bmatrix} f(x_1) \\ \vdots \\ f(x_K) \end{bmatrix} \rightarrow \mathbf{A} \cdot \lambda_1 = \mathbf{f}, \quad (28)$$

where \mathbf{A} and \mathbf{f} are column vectors. Using the method of least squares, we can calculate λ_1 :

$$\lambda_1 = \frac{\mathbf{A}^T \cdot \mathbf{f}}{\mathbf{A}^T \mathbf{A}}. \quad (29)$$

Equation (29) can be rewritten as:

$$\lambda_1 = \frac{\sum_{i=1}^K \Phi(\|x_i - \xi\|) \cdot f(x_i)}{\sum_{i=1}^K (\Phi(\|x_i - \xi\|))^2}. \quad (30)$$

The value $f(\xi)$ is calculated as:

$$f(\xi) = \lambda_1 \Phi(\|\xi - \xi\|) = \lambda_1 \Phi(0). \quad (31)$$

4 Comparison of Time Complexity

In the following, a comparison of LOWESS and RBF will be made. The main criteria for comparison are:

- The computational complexity, which is critical if many points have to be approximated.
- The quality of the final approximation (see section 5).

4.1 LOWESS

The size of matrix \mathbf{A} is $k \times q$, where the number of used nearest points is k and q is equal to the degree of the polynomial plus 1. The size of diagonal matrix \mathbf{W} is $k \times k$, the size of vector \mathbf{b} is $k \times 1$ and the size of vector \mathbf{x} is $k \times 1$. The time complexity of LOWESS using equation (9) can be calculated in the following way:

$$\begin{aligned}
 \mathbf{A}^T \mathbf{W} \mathbf{A} &\rightarrow O(q^2 k + qk) \\
 (\mathbf{A}^T \mathbf{W} \mathbf{A})^{-1} &\rightarrow O(q^2 k + qk + q^3) \\
 \mathbf{A}^T \mathbf{W} \mathbf{b} &\rightarrow O(2qk) \\
 (\mathbf{A}^T \mathbf{W} \mathbf{A})^{-1} \mathbf{A}^T \mathbf{W} \mathbf{b} &\rightarrow O(k(q^2 + 3q) + q^3 + q^2)
 \end{aligned} \tag{32}$$

As the size k of matrix \mathbf{A} is much larger than the size q of matrix \mathbf{A} , the time complexity from (32) will become:

$$\begin{aligned}
 O(3qk) &\text{ for } q = \{1,2\} \\
 O(q^2 k) &\text{ for } q \geq 3
 \end{aligned} \tag{33}$$

The time complexity of LOWESS when calculating the approximation value in R points will become:

$$\begin{aligned}
 O(N \log N + R \cdot 3qk) &\text{ for } q = \{1,2\} \\
 O(N \log N + R \cdot q^2 k) &\text{ for } q \geq 3
 \end{aligned} \tag{34}$$

where N is the number of input points and $O(N \log N)$ is the time complexity of the sorting algorithm for $1\frac{1}{2}$ dimensional data. In the case of higher dimensions $D\frac{1}{2}$, i.e. $D > 1$, the total time complexity of selecting k -nearest points from N points increases (see section 7 for more details).

4.2 Simplified RBF

The size of matrix \mathbf{A} is $k \times r$, where the number of used nearest points is k and r is equal to the degree of the polynomial plus 2. The size of vector \mathbf{f} is $k \times 1$ and the size of vector $\boldsymbol{\eta} = [\boldsymbol{\lambda}^T, \mathbf{a}^T]^T$ is $k \times 1$. The time complexity of RBF using equation (24) can be calculated in the following way:

$$\begin{aligned}
\mathbf{A}^T \mathbf{A} &\rightarrow O(r^2 k) \\
(\mathbf{A}^T \mathbf{A})^{-1} &\rightarrow O(r^2 k + r^3) \\
\mathbf{A}^T \mathbf{f} &\rightarrow O(rk) \\
(\mathbf{A}^T \mathbf{A})^{-1} (\mathbf{A}^T \mathbf{f}) &\rightarrow O(k(r^2 + r) + r^3 + r^2)
\end{aligned} \tag{35}$$

As the size k of matrix \mathbf{A} is much larger than the size r of matrix \mathbf{A} , the time complexity from (35) will become:

$$O(r^2 k) \tag{36}$$

The time complexity of simplified RBF when calculating the approximation value in R points can be estimated:

$$O(N \log N + R \cdot r^2 k) \tag{37}$$

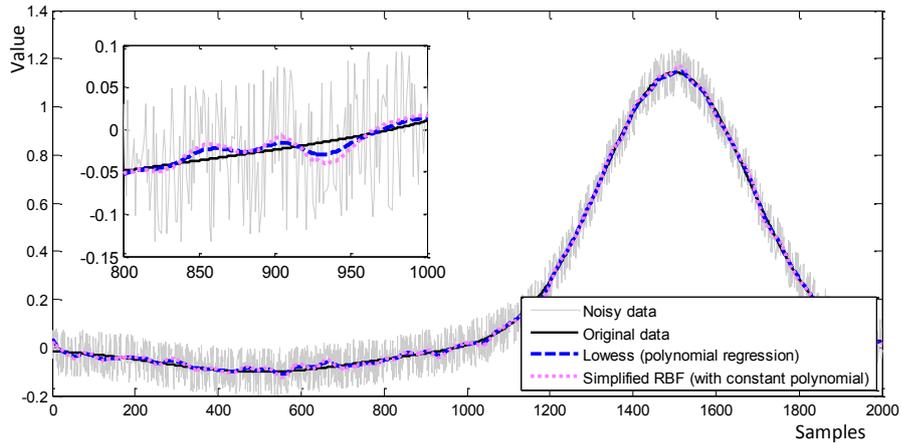
where N is the number of input points and $O(N \log N)$ is the time complexity of the sorting algorithm for $1\frac{1}{2}$ dimensional data.

5 Comparison of Measured Errors

For a demonstration of LOWESS and RBF approximation properties, the standard testing function, which is considered by Hickernell and Hon [5], has been selected:

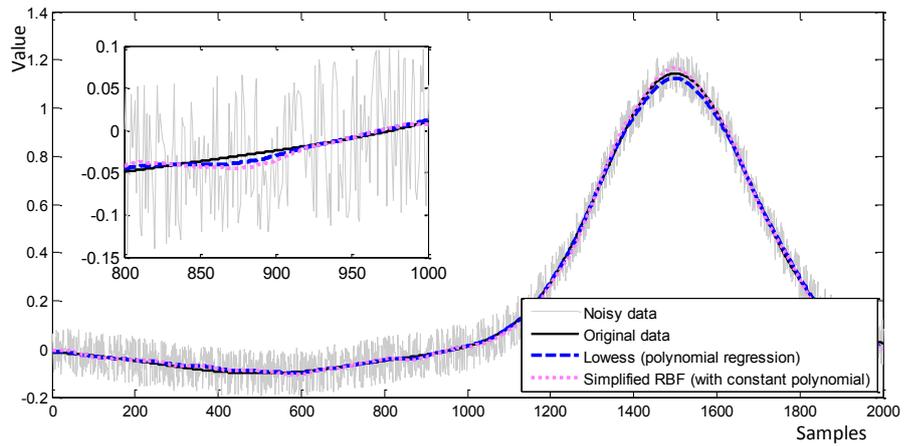
$$\tau(x) = e^{\left[-15\left(x-\frac{1}{2}\right)^2\right]} + \frac{1}{2}e^{\left[-20\left(x-\frac{1}{2}\right)^2\right]} - \frac{3}{4}e^{\left[-8\left(x+\frac{1}{2}\right)^2\right]} \tag{38}$$

This function was sampled at $\langle -1, 1 \rangle$. We added random noise with uniform distribution from interval $\langle -0.1, 0.1 \rangle$ and used it as input for both methods. The following graphs present the behavior of the LOWESS and RBF approximations.

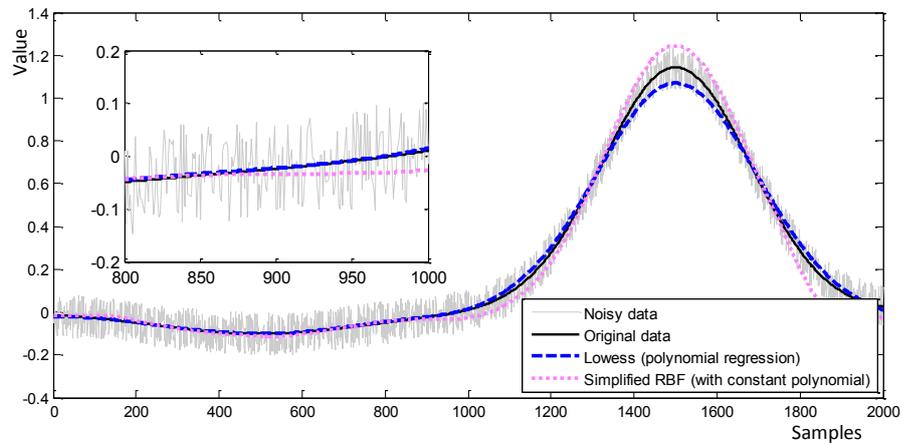


Graph 1: Comparison of LOWESS with Simplified RBF. 100 nearest samples out of 2000 total were used as values for approximation; sampled interval: $\langle -1, 1 \rangle$.

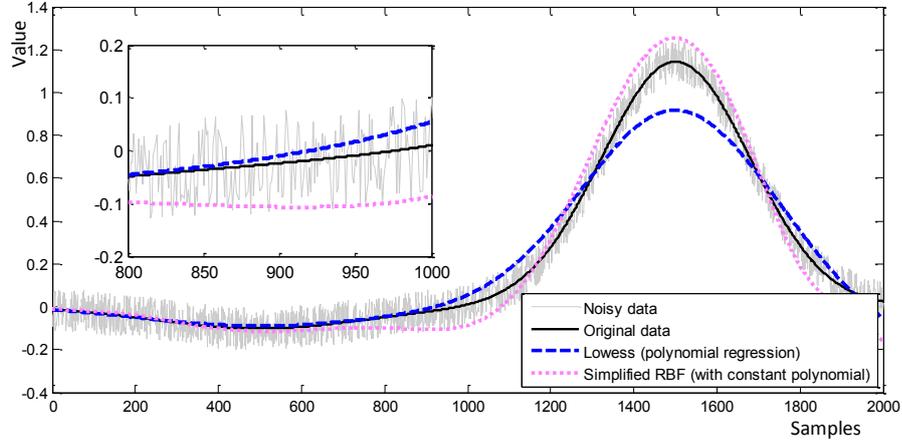
Function values are on the vertical axis; values on the horizontal axis are sample indices. Since the function is sampled at $(-1,1)$ and the number of samples is 2000, the sampling rate is 1000 samples per 1 unit.



Graph 2: Comparison of LOWESS with Simplified RBF. 200 nearest samples out of 2000 total were used as values for approximation; sampled interval: $(-1,1)$.



Graph 3: Comparison of LOWESS with Simplified RBF. 500 nearest samples out of 2000 total were used as values for approximation; sampled interval: $(-1,1)$.



Graph 4: Comparison of LOWESS with Simplified RBF. 1000 nearest samples out of 2000 total were used as values for approximation; sampled interval: $\langle -1,1 \rangle$.

The error of approximation can be measured in different ways. The first one is to measure the change of the first derivative, which is the curvature of the resulting curve. If the first derivative changes too much, then the curve is jagged; on the contrary, if the first derivative does not change too much, then the curve is smooth. The absolute error can be calculated using the formula:

$$E_c = \sum_{i=1}^N \|f''(x_i)\|, \quad (39)$$

where $f''(x_i)$ is calculated using the formula:

$$f''(x_i) = \frac{f(x_{i+1} - 2x_i + x_{i-1}))}{(x_{i+1} - x_i)(x_i - x_{i-1})}, \quad (40)$$

Let $\mathbf{p} = [x, f(x)]$ be the approximated point in current space ($2D$ for $1\&\frac{1}{2}$ dimensions) and $\mathbf{\kappa} = [x, \tau(x)]$ be a point of the sampled function (38), which is a set $K = \{\mathbf{\kappa}_1, \dots, \mathbf{\kappa}_N\} = \{[x_1, \tau(x_1)], \dots, [x_N, \tau(x_N)]\}$, then the distance error from the original curve without noise can be calculated as:

$$E_d = \sum_{i=1}^N \|\mathbf{p}_i - \xi_j\|, \text{ where } \|\mathbf{p}_i - \mathbf{\kappa}_j\| \text{ is minimal } \forall j \in \{1, \dots, N\} \text{ for given } i. \quad (41)$$

Let us note that the distance is not measured vertically to the curve but ‘‘orthogonally’’ to the curve. Using formulas (39) and (41), we can show the following table of calculated errors.

Tab. 1: Measured errors for graphs Graph 1 - Graph 4 (for $N = 2000$).

k -nearest samples	E_c		E_d	
	LOWESS	Simplified RBF	LOWESS	Simplified RBF
100	0.0721	1.4585	7.2997	12.5647
200	0.0212	0.7689	10.5378	14.7898
500	0.0132	0.3103	15.6759	40.5985
1 000	0.0091	0.1618	45.0717	70.8979

Some comparison results can be seen using (Tab. 1). The LOWESS approximation is always smoother (according to measured error E_c) and closer to the original data without noise (according to measured error E_d) when using the same k -nearest samples.

6 Global RBF Approximation

Global RBF approximation can be calculated using (14). In this case, the whole data set has to be processed at once. Compared to the simplified version of RBF approximation, we only get one λ vector for all input samples and thus we solve a linear system only once. Moreover, we do not need to sort the input points in any way, unlike LOWESS and simplified RBF approximations, which were presented in previous sections. The global RBF approximation is calculated using the following formula (from (17)):

$$\mathbf{A}\lambda = \mathbf{f} \rightarrow \lambda = (\mathbf{A}^T \mathbf{A})^{-1} \cdot (\mathbf{A}^T \mathbf{f}) \quad (42)$$

where the size of matrix \mathbf{A} is $N \times (M + d + 1)$, N is the number of input points, M is the number of radial basis functions, d is the degree of the polynomial, the size of vector \mathbf{f} is $N \times 1$, the size of vector $\lambda = [\lambda_1, \dots, \lambda_M, a_0, \dots, a_d]^T$ is $(M + d + 1) \times 1$. We can express the time complexity of the global RBF approximation calculation (42) as:

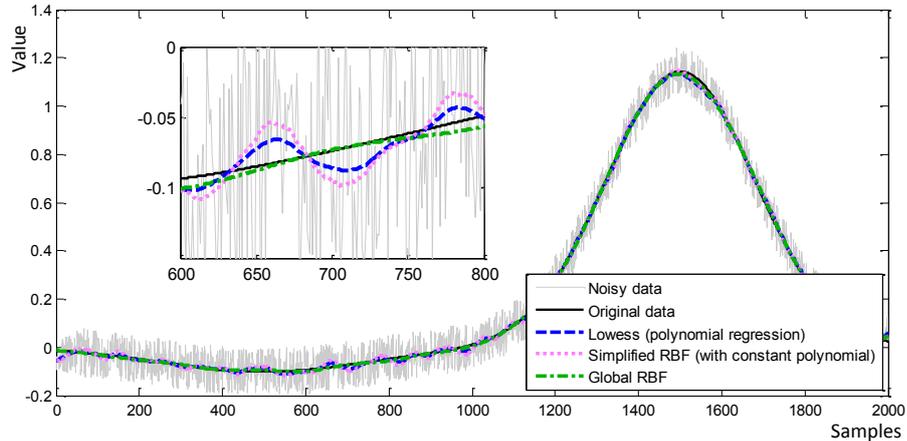
$$\begin{aligned} \mathbf{A}^T \mathbf{A} &\rightarrow O((M + d + 1)^2 N) \\ (\mathbf{A}^T \mathbf{A})^{-1} &\rightarrow O((M + d + 1)^2 N + (M + d + 1)^3) \\ \mathbf{A}^T \mathbf{f} &\rightarrow O((M + d + 1)N) \\ (\mathbf{A}^T \mathbf{A})^{-1} \cdot (\mathbf{A}^T \mathbf{f}) &\rightarrow O\left(N((M + d + 1)^2 + (M + d + 1)) + (M + d + 1)^3 + (M + d + 1)^2\right) \end{aligned} \quad (43)$$

and after leaving only the most complex part, the time complexity is:

$$O(M^2 N) \quad (44)$$

We sampled function (38), added random noise with uniform distribution from interval $(-0.1, 0.1)$, and used that data as input for both methods mentioned in previous sections (LOWESS and simplified RBF) and for global RBF approximation as well. The following graph presents the behavior of the LOWESS, simplified RBF and global RBF approximations. Function values are on the vertical axis, values on the horizontal

axis are sample indices. Since the function is sampled at $(-1,1)$ and the number of samples is 2000, the sampling rate is 1000 samples per 1 unit.



Graph 5: Comparison of LOWESS and Simplified RBF with global RBF. 100 nearest samples out of 2000 total were used as values for local approximation, which gives 20 pivots (lambdas) for global RBF; sampled interval: $(-1,1)$.

The following table presents calculated errors using formulas (39) and (41) (for Graph 5) for all approximation methods described in this paper.

Tab. 2: Measured errors for Graph 5.

E_c			E_d		
LOWESS	Simplified RBF	Global RBF	LOWESS	Simplified RBF	Global RBF
0.0718	1.5266	0.0168	10.6785	16.0734	6.0123

It can be seen, that global RBF approximation is closer to the original data and smoother than simple RBF or even LOWESS approximation. For the situation in Graph 5 and Tab. 2, the time complexity of global RBF is exactly the same as the time complexity of both other methods when calculating the approximation at all input points.

7 Approximation in Higher Dimensions

Let us assume that a scattered data approximation [6, 7] in $2\frac{1}{2}$ or $3\frac{1}{2}$ dimensions, i.e. $D\frac{1}{2}$ dimensions have to be made. In the following, we describe the expansion of LOWESS and RBF approximation algorithms into higher dimensions.

In higher than $1\frac{1}{2}$ dimensions, we have to deal with the fact that there is no ordering defined in general. Thus, we cannot sort all input points at once in the beginning and then choose k -nearest points with $O(1)$ time complexity. The time

complexity of selecting k -nearest points from N points is $O(N \log N)$, and thus the time complexity of LOWESS or Simple RBF approximation can be estimated as:

$$O\left(R \cdot \left(N \log N + \begin{cases} O_{LOWESS} \\ \text{or} \\ O_{RBF} \end{cases}\right)\right), \quad (45)$$

where O_{LOWESS} is the same time complexity as the time complexity of LOWESS approximation in $1\&\frac{1}{2}$ dimensions and O_{RBF} is the same time complexity as the time complexity of Simple RBF approximation in $1\&\frac{1}{2}$ dimensions.

7.1 LOWESS

In the case of $D\&\frac{1}{2}$ dimensional approximation, we have to change the notation in (1) as \mathbf{x} is a D -dimensional position vector:

$$S = \sum_{i=1}^N \omega_i \cdot \left(h_i - P^{(D)}(\mathbf{x}_i)\right)^2, \quad (46)$$

where $h = P^{(D)}(\mathbf{x})$ is a D dimensional hypersurface function with unknown coefficients $\mathbf{a} = [a_0, a_1, a_2, \dots, a_k]^T$. For $D = 2$, we can write $P^{(D)}(\mathbf{x})$, for example, like:

$$P^{(2)}(\mathbf{x}) = a_0 + a_1x + a_2y + a_3x^2 + a_4y^2 + a_5xy, \quad (47)$$

where $\mathbf{x} = [x, y]^T$. The matrix \mathbf{A} is then equal to:

$$\mathbf{A} = \begin{bmatrix} 1 & x_1 & y_1 & x_1^2 & y_1^2 & x_1y_1 \\ 1 & x_2 & y_2 & x_2^2 & y_2^2 & x_2y_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_N & y_N & x_N^2 & y_N^2 & x_Ny_N \end{bmatrix} \quad (48)$$

We can omit some coefficients a_i and corresponding columns in matrix \mathbf{A} , where $i \in \{0, 1, \dots, 5\}$. All other computations remain the same.

The computation complexity will increase as the size of matrix \mathbf{A} increases. However, if we use a constant hypersurface function with only one coefficient a_0 , then the time complexity does not change with different dimensions D .

7.2 Simplified RBF

The RBF approximation is formally independent from the dimension D . Therefore, all the computations remain the same as described above. The computation complexity increases slightly as the complexity of polynomial/hypersurface $P^{(D)}(\mathbf{x})$ increases. However, if we use a constant hypersurface function with only one coefficient a_0 , then the time complexity will not change with different dimensions D . The polynomial $P^{(D)}(\mathbf{x})$ is actually a data approximation using a basic function and $\sum_{i=1}^M \lambda_i \Phi_i(r)$ controls the perturbation from $P^{(D)}(\mathbf{x})$.

8 Conclusion

We have introduced the LOWESS method of approximation and modified RBF approximation, which is comparable with LOWESS. Both methods use the same number of nearest samples for approximation and the time complexity of both these methods is the same. We calculated the distance of approximated noisy data to the original data. In all cases, for the same number of nearest samples for approximation, LOWESS gives better results. Another comparison of both methods is calculation of the smoothness. The LOWESS approximation gives us smoother results than the Simple RBF approximation. However, both these methods use a different approach for approximation than global RBF approximation; we compared them with global RBF approximation as well. Using global RBF approximation we can achieve better results (closer distance to original data and smoother approximation) when having the same time complexity of calculation. Moreover, we get one simple continuous formula and not only function values at discrete points. On the other hand, both methods can be used in higher dimensions, but the time complexity will increase for both of them compared to the situation in $1\frac{1}{2}$ dimensions. Due to this fact, in higher dimensions, global RBF approximation has lower time complexity than either LOWESS or Simple RBF approximation due to necessity of finding k -nearest neighbor points. Therefore, the global RBF approximation is recommendable for approximation of scattered data in higher dimensions, i.e. $2\frac{1}{2}$ dimensions and higher.

All methods for approximation compared in this paper were implemented and tested in MATLAB.

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

References

- 1 Bellochino, F., Borghese, N. A., Ferrari, S., Piuri, V.: 3D surface Reconstruction, ISBN: 978-1-4614-5632-2, Springer, 2013.
- 2 Chen, L. M.: Digital Functions and Data Reconstruction, Springer, ISBN: 978-1-4614-5637-7, 2013.
- 3 Cleveland, W. S.: Robust Locally Weighted Regression and Smoothing Scatterplots, Journal of the American Statistical Association, Vol. 74, No. 368, pp. 829-836, 1979, dx.doi.org/10.2307/2286407.
- 4 Fasshauer, G. E.: Meshfree Approximation Methods with MATLAB, World Scientific Publ., ISBN 978-981-270-633-1, 2007.
- 5 Hickernell, F. J., Hon, Y. C.: Radial basis function approximation as smoothing splines, Appl. Math. Comput. 102 (1999) 1–24.

- 6 Lazzaro, D., Montefusco, L. B.: Radial basis functions for the multivariate interpolation of large scattered data sets, *Journal of Computational and Applied Mathematics* 1040, pp. 521-536, Elsevier, 2002.
- 7 Narcowich, F. J.: Recent developments in error estimates for scattered data interpolation via radial basis functions, *Numerical Algorithms*, Vo. 39, pp.307-315, Springer, 2005.
- 8 Pan, R., Skala, V.: A two level approach to implicit modeling with compactly supported radial basis functions, *Engineering and Computers*, Vol. 27. No. 3, pp. 299-307, ISSN: 0177-0667, Springer, 2011, [dx.doi.org/10.1007/s00366-010-0199-1](https://doi.org/10.1007/s00366-010-0199-1).
- 9 Pan, R., Skala, V.: Surface Reconstruction with higher-order smoothness, *The Visual Computer*, Vol. 28, No. 2., pp. 155-162, ISSN: 0178-2789, Springer, 2012.
- 10 Skala, V: Progressive RBF Interpolation, 7th Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction, Afrigraph 2010, pp. 17-20, ACM, ISBN: 978-1-4503-0118-3, 2010.
- 11 Skala, V., Pan, R., Nedved, O: Making 3D replicas using a flatbed scanner and a 3D printer, *ICCSA 2014*, pp. 76-86, ISBN: 978-3-319-09152-5, Springer, 2014.
- 12 Wendland, H.: Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree, *Advances in Computational Mathematics*, Vol. 4, Issue 1, pp 389-396, [dx.doi.org/10.1007/BF02123482](https://doi.org/10.1007/BF02123482).
- 11 Yao, X., Fu, B., Lü, Y., et al.: Comparison of Four Spatial Interpolation Methods for Estimating Soil Moisture in a Complex Terrain Catchment. *PLoS ONE*, Vol. 8, Issue 1, 2013, [dx.doi.org/10.1371/journal.pone.0054660](https://doi.org/10.1371/journal.pone.0054660).

5. Conclusion

This thesis presented new approaches and methods in the field of meshless vector fields approximation and compression that were researched and developed during the doctoral studies of the author of this thesis.

The main research topic was the vector field approximation using Radial basis functions. Many approaches were developed, where the focus always was to keep the important features of the vector field. The important features are for example critical points, however, some of them are not important and can be omitted as presented in one of the journal publications. Another important feature that has been preserved in the approximated vector fields is the global character of the vector field.

Some applications require small turbulence and small details to be preserved. The recently submitted journal paper presents an approach that deals with this problem using streamlines curvature and weighted RBF approximation.

The measured or numerically simulated vector fields are represent by very large data sets. Several approaches were developed how to approximate such large data sets. The approaches are using the space subdivision to speed-up the computation and to reduce the needed memory for approximation of such large data sets, that could not be even approximated by conventional meshless methods.

As the result of the vector field approximation is an analytical formula that can be used for further processing of the vector field. This is the advantage over other existing methods that are using the standard mesh approaches.

During the doctoral studies were also published many other publication that are not directly related to the main topic of this dissertation. This publications are listed in the Appendix A. Altogether there were published 6 journal papers and 22 conference papers.

5.1 Future work

Now, the solutions for topology-based unsteady flow remain incomplete, compared with the level of research achieved for steady flows. Incremental extensions of methods that work well for steady flows are proven not to be able to fully capture the behavior of time varying vector fields. Therefore, new approaches and methods are needed to be investigated in this area as well. Moreover, there needs to be thorough research of how to correctly solve the approximation of time varying data using the Radial basis functions. When computing the distance parameter for radial basis function, one needs to compute the distance in time and space, which has to be computed physically correct.

Another direction of research can be the compression of “lambda” values for RBF value computation. Those values are usually saved as double precision floating point numbers. However, the accuracy of “lambda” values may be reduced, while maintaining the low approximation error of RBF approximation.

The next, but not the last problem that can be solved is the selection of optimal centers for radial basis functions as well as the optimal shape parameters for each radial basis function. This problem is very hard to solve, as there exist many local optimal solutions. However, to compute the global optimum, it would be very time consuming if all time realistic.

A. Activities

A.1 Publications in Journals

Published journal papers

- Michal Smolik, Vaclav Skala, and Zuzana Majdisova. 3D vector field approximation and critical points reduction using radial basis functions. In *International journal of mechanics*, 13(1):100–103, 2019. (CiteScore = 1.00)
- Michal Smolik and Vaclav Skala. Large scattered data interpolation with radial basis functions and space subdivision. *Integrated Computer-Aided Engineering*, 25(1):49–62, 2018. (IF = 4.904)
- Michal Smolik, Vaclav Skala, and Zuzana Majdisova. Vector field radial basis function approximation. *Advances in Engineering Software*, 123(1):117–129, 2018. (IF = 4.194)
- Vaclav Skala, Zuzana Majdisova, and Michal Smolik. Space subdivision to speed-up convex hull construction in E3. *Advances in Engineering Software*, 91:12–22, 2016. (IF = 3.000)
- Vaclav Skala, Michal Smolik, and Lukas Karlicek. HS-Patch: A new Hermite smart bicubic patch modification. *International journal of mathematics and computers in simulation*, 8(1):292–299, 2014. (CiteScore = 0.66)

Accepted journal papers

- Zuzana Majdisova, Vaclav Skala, and Michal Smolik. Near optimal placement of reference points and choice of an appropriate variable shape parameter for RBF approximation. *Integrated Computer-Aided Engineering*, 2019. (IF = 4.904)

Journal papers in review

- Michal Smolik and Vaclav Skala. Radial basis function and multi-level 2D vector field approximation. *Mathematics and Computers in Simulation*, 2019. (IF = 1.409)
- Michal Smolik and Vaclav Skala. Vector field radial basis functions approximation with streamlines curvature. *Advances in Engineering Software*, 2019. (IF = 4.194)

A.2 Publications on International Conferences

- Michal Smolik and Vaclav Skala. Efficient simple large scattered 3D vector fields radial basis functions approximation using space subdivision. In *Computational Science and Its Applications – ICCSA 2019*, pages 337–350. Springer, 2019.
- Martin Cervenka, Michal Smolik, and Vaclav Skala. A new strategy for scattered data approximation using radial basis functions respecting points of inflection. In *Computational Science and Its Applications – ICCSA 2019*, pages 322–336. Springer, 2019.
- Vaclav Skala and Michal Smolik. Simple and fast $O_{exp}(N)$ algorithm for finding an exact maximum distance in E2 instead of $O(N^2)$ or $O(N \lg N)$. In *Computational Science and Its Applications – ICCSA 2019*, pages 367–380. Springer, 2019.
- Michal Smolik, Vaclav Skala, and Zuzana Majdisova. A new simple, fast and robust total least square error computation in E2 experimental comparison. In *International Conference on Advanced Engineering Theory and Applications*, pages 325–334. Springer, 2018.
- Michal Smolik, Vaclav Skala, and Zuzana Majdisova. Vector field RBF approximation multi-level and critical points reduction. In *Future Forces Forum (FFF) - Posters*. Ministry of Defense and others, 2018.
- Vaclav Skala and Michal 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.
- Vaclav Skala and Michal Smolik. A new formulation of Plücker coordinates using projective representation. In *2018 5th International Conference on Mathematics and Computers in Sciences and Industry (MCSI)*, pages 52–56. IEEE, 2018.
- Vaclav Skala, Michal Smolik, and Mariia Martynova. Geometric product for multidimensional dynamical systems - Laplace transform and geometric algebra. In *European Conference on Electrical Engineering & Computer Science*. IEEE, 2018.
- Zuzana Majdisova, Vaclav Skala, and Michal Smolik. Incremental meshfree approximation of real geographics data. In *International Conference on Applied Physics, System Science and Computers*, pages 222–228. Springer, 2018.
- Zuzana Majdisova, Vaclav Skala, and Michal Smolik. Determination of stationary points and their bindings in dataset using RBF methods. In *Proceedings of the Computational Methods in Systems and Software*, pages 213–224. Springer, 2018.
- Zuzana Majdisova, Vaclav Skala, and Michal Smolik. Geographic point clouds RBF approximation and compression. In *Future Forces Forum (FFF) - Posters*. Ministry of Defense and others, 2018.
- Michal Smolik and Vaclav Skala. Vector field second order derivative approximation and geometrical characteristics. In *International Conference on Computational Science and Its Applications*, pages 148–158. Springer, 2017.
- Michal Smolik and Vaclav Skala. Spherical RBF vector field interpolation: experimental study. In *2017 IEEE 15th International Symposium on Applied Machine Intelligence and Informatics (SAMII)*, pages 431–434. IEEE, 2017.

- Michal Smolik and Vaclav Skala. Classification of critical points using a second order derivative. *Procedia Computer Science*, 108:2373–2377, 2017.
- Michal Smolik and Vaclav Skala. Vector field RBF interpolation on a sphere. In *The Computer Graphics, Visualization, Computer Vision and Image Processing (CGVCVIP)*, pages 352–354. IADIS Press, 2016.
- Michal Smolik and Vaclav Skala. Vector field interpolation with radial basis functions. In *Proceedings of SIGRAD 2016, May 23rd and 24th, Visby, Sweden*, number 127, pages 15–21. Linköping University Electronic Press, 2016.
- Michal Smolik, Vaclav Skala, and Ondrej Nedved. A comparative study of LOWESS and RBF approximations for visualization. In *International Conference on Computational Science and Its Applications*, pages 405–419. Springer, 2016.
- Vaclav Skala, Michal Smolik, and Zuzana Majdisova. Reducing the number of points on the convex hull calculation using the polar space subdivision in E2. In *2016 29th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 40–47. IEEE, 2016.
- Michal Smolik and Vaclav Skala. Highly parallel algorithm for large data in-core and out-core triangulation in E2 and E3. *Procedia Computer Science*, 51:2613–2622, 2015.
- Vaclav Skala and Michal Smolik. A point in non-convex polygon location problem using the polar space subdivision in E2. In *International Conference on Image and Graphics*, pages 394–404. Springer, 2015.
- Michal Smolik and Vaclav Skala. In-core and out-core memory fast parallel triangulation algorithm for large data sets in E2 and E3. In *ACM SIGGRAPH 2014 Posters*, page 37. ACM, 2014.
- Michal Smolik and Vaclav Skala. Fast parallel triangulation algorithm of large data sets in E2 and E3 for in-core and out-core memory processing. In *International Conference on Computational Science and Its Applications*, pages 301–314. Springer, 2014.

A.3 Participation in Scientific Projects

- GA17-05534S – Meshless methods for large scattered spatio-temporal vector data visualization. The Czech Science Foundation (GAČR)
- LH12181 – Development of Algorithms for Computer Graphics and CAD/CAM systems. Czech Ministry of Education, Youth and Sports (MEYS)
- LG13047 – EURO: Activities within Eurographics Association. Czech Ministry of Education, Youth and Sports (MEYS)
- SGS-2019-016 – Synthesis and Analysis of Geometric and Computing Models. University of West Bohemia (UWB)
- SGS-2016-013 – Advanced Graphics and Computational Systems. University of West Bohemia (UWB)
- SGS-2013-029 – Advanced Computing and Information Systems. University of West Bohemia (UWB)

Bibliography

- [Agranovsky et al., 2015] Agranovsky, A., Camp, D., Joy, K. I., and Childs, H. (2015). Subsampling-based compression and flow visualization. In *Visualization and Data Analysis 2015*, volume 9397, page 93970J. International Society for Optics and Photonics.
- [Asimov, 1993] Asimov, D. (1993). Notes on the topology of vector fields and flows. Technical report, Technical report, NASA Ames Research Center, 1993. RNR-93-003.
- [Ben-Israel, 1966] Ben-Israel, A. (1966). A newton-raphson method for the solution of systems of equations. *Journal of Mathematical analysis and applications*, 15(2):243–252.
- [Bhatia et al., 2014] Bhatia, H., Gyulassy, A., Wang, H., Bremer, P.-T., and Pascucci, V. (2014). Robust detection of singularities in vector fields. In *Topological Methods in Data Analysis and Visualization III*, pages 3–18. Springer.
- [Blazek, 2015] Blazek, J. (2015). *Computational fluid dynamics: principles and applications*. Butterworth-Heinemann.
- [Cabrera et al., 2013] Cabrera, D. A. C., Gonzalez-Casanova, P., Gout, C., Juárez, L. H., and Reséndiz, L. R. (2013). Vector field approximation using radial basis functions. *Journal of Computational and Applied Mathematics*, 240:163–173.
- [Cao et al., 2015] Cao, L., Pan, H., Levine, J. A., and Wang, W. (2015). Anisotropic optimal delaunay triangulation for the simplification of 2D vector fields. *Journal of Information & Computational Science*, 12(13):4885–4896.
- [Chen et al., 2008] Chen, G., Mischaikow, K., Laramée, R. S., and Zhang, E. (2008). Efficient morse decompositions of vector fields. *IEEE Transactions on Visualization and Computer Graphics*, 14(4):848–862.
- [Chung et al., 2015] Chung, E. T., Efendiev, Y., and Lee, C. S. (2015). Mixed generalized multiscale finite element methods and applications. *Multiscale Modeling & Simulation*, 13(1):338–366.
- [Čomić and De Floriani, 2008] Čomić, L. and De Floriani, L. (2008). Cancellation of critical points in 2D and 3D morse and morse-smale complexes. In *International Conference on Discrete Geometry for Computer Imagery*, pages 117–128. Springer.
- [de Leeuw and van Liere, 2000] de Leeuw, W. and van Liere, R. (2000). Multi-level topology for flow visualization. *Computers & Graphics*, 24(3):325–331.
- [Dey et al., 2007] Dey, T. K., Levine, J. A., and Wenger, R. (2007). A delaunay simplification algorithm for vector fields. In *Computer Graphics and Applications, 2007. PG'07. 15th Pacific Conference on*, pages 281–290. IEEE.
- [Edelsbrunner et al., 2000] Edelsbrunner, H., Letscher, D., and Zomorodian, A. (2000). Topological persistence and simplification. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 454–463. IEEE.
- [Fasshauer, 2007] Fasshauer, G. E. (2007). *Meshfree approximation methods with MATLAB*, volume 6. World Scientific.

- [Forman, 1998] Forman, R. (1998). Combinatorial vector fields and dynamical systems. *Mathematische Zeitschrift*, 228(4):629–681.
- [Gardiner et al., 1992] Gardiner, J. D., Laub, A. J., Amato, J. J., and Moler, C. B. (1992). Solution of the sylvester matrix equation $axb + cx + dx = e$. *ACM Transactions on Mathematical Software (TOMS)*, 18(2):223–231.
- [Garth et al., 2004] Garth, C., Tricoche, X., and Scheuermann, G. (2004). Tracking of vector field singularities in unstructured 3D time-dependent datasets. In *Proceedings of the conference on Visualization'04*, pages 329–336. IEEE Computer Society.
- [Ghosh-Dastidar et al., 2008] Ghosh-Dastidar, S., Adeli, H., and Dadmehr, N. (2008). Principal component analysis-enhanced cosine radial basis function neural network for robust epilepsy and seizure detection. *IEEE Transactions on Biomedical Engineering*, 55(2):512–518.
- [Gjøystdal, 2004] Gjøystdal, K. (2004). *Classifying zeros in three dimensional vector fields*. PhD thesis, Master's thesis, The Norwegian University of Science and Technology.
- [Globus et al., 1991] Globus, A., Levit, C., and Lasinski, T. (1991). A tool for visualizing the topology of three-dimensional vector fields. In *Proceedings of the 2nd conference on Visualization'91*, pages 33–40. IEEE Computer Society Press.
- [Greene, 1992] Greene, J. M. (1992). Locating three-dimensional roots by a bisection method. *Journal of Computational Physics*, 98(2):194–198.
- [Gyulassy and Natarajan, 2005] Gyulassy, A. and Natarajan, V. (2005). Topology-based simplification for feature extraction from 3D scalar fields. In *VIS 05. IEEE Visualization, 2005.*, pages 535–542. IEEE.
- [Gyulassy et al., 2006] Gyulassy, A., Natarajan, V., Pascucci, V., Bremer, P.-T., and Hamann, B. (2006). A topological approach to simplification of three-dimensional scalar functions. *IEEE Transactions on Visualization and Computer Graphics*, 12(4):474–484.
- [Gyulassy et al., 2007] Gyulassy, A., Natarajan, V., Pascucci, V., and Hamann, B. (2007). Efficient computation of morse-smale complexes for three-dimensional scalar functions. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1440–1447.
- [Hardy, 1971] Hardy, R. L. (1971). Multiquadric equations of topography and other irregular surfaces. *Journal of geophysical research*, 76(8):1905–1915.
- [Heath, 2018] Heath, M. T. (2018). *Scientific computing: an introductory survey*, volume 80. SIAM.
- [Heckel et al., 1999] Heckel, B., Weber, G., Hamann, B., and Joy, K. I. (1999). Construction of vector field hierarchies. In *Proceedings of the conference on Visualization'99: celebrating ten years*, pages 19–25. IEEE Computer Society Press.
- [Helman and Hesselink, 1989] Helman, J. and Hesselink, L. (1989). Representation and display of vector field topology in fluid flow data sets. *IEEE Computer*, 22(8):27–36.
- [Helman and Hesselink, 1991] Helman, J. L. and Hesselink, L. (1991). Visualizing vector field topology in fluid flows. *IEEE Computer Graphics and Applications*, (3):36–46.
- [Jumarhon et al., 2000] Jumarhon, B., Amini, S., and Chen, K. (2000). The hermite collocation method using radial basis functions. *Engineering analysis with boundary elements*, 24(7-8):607–611.
- [Kansa, 1990a] Kansa, E. J. (1990a). Multiquadrics - A scattered data approximation scheme with applications to computational fluid-dynamics—i surface approximations and partial derivative estimates. *Computers & Mathematics with applications*, 19(8-9):127–145.
- [Kansa, 1990b] Kansa, E. J. (1990b). Multiquadrics - A scattered data approximation scheme with applications to computational fluid-dynamics—ii solutions to parabolic, hyperbolic and elliptic partial differential equations. *Computers & mathematics with applications*, 19(8):147–161.

- [Karim and Adeli, 2003] Karim, A. and Adeli, H. (2003). Radial basis function neural network for work zone capacity and queue estimation. *Journal of Transportation Engineering*, 129(5):494–503.
- [Katok and Hasselblatt, 1995] Katok, A. and Hasselblatt, B. (1995). *Introduction to the modern theory of dynamical systems*, volume 54. Cambridge university press.
- [Koch et al., 2016] Koch, S., Kasten, J., Wiebel, A., Scheuermann, G., and Hlawitschka, M. (2016). 2D vector field approximation using linear neighborhoods. *The Visual Computer*, 32(12):1563–1578.
- [Koch et al., 2013] Koch, S., Wiebel, A., Kasten, J., and Hlawitschka, M. (2013). Visualizing linear neighborhoods in non-linear vector fields. In *2013 IEEE Pacific Visualization Symposium (PacificVis)*, pages 249–256. IEEE.
- [Larsson and Fornberg, 2003] Larsson, E. and Fornberg, B. (2003). A numerical study of some radial basis function based solution methods for elliptic PDEs. *Computers & Mathematics with Applications*, 46(5):891–902.
- [Lawson and Hanson, 1995] Lawson, C. L. and Hanson, R. J. (1995). *Solving least squares problems*, volume 15. Siam.
- [Lodha et al., 2003] Lodha, S. K., Faaland, N. M., and Renteria, J. C. (2003). Topology preserving top-down compression of 2D vector fields using bintree and triangular quadtrees. *IEEE Transactions on Visualization and Computer Graphics*, 9(4):433–442.
- [Lodha et al., 2000] Lodha, S. K., Renteria, J. C., and Roskin, K. M. (2000). Topology preserving compression of 2D vector fields. In *Proceedings Visualization 2000. VIS 2000 (Cat. No. 00CH37145)*, pages 343–350. IEEE.
- [Majdisova and Skala, 2017a] Majdisova, Z. and Skala, V. (2017a). Big geo data surface approximation using radial basis functions: A comparative study. *Computers & Geosciences*, 109:51–58.
- [Majdisova and Skala, 2017b] Majdisova, Z. and Skala, V. (2017b). Radial basis function approximations: Comparison and applications. *Applied Mathematical Modelling*, 51:728–743.
- [Mann and Rockwood, 2002] Mann, S. and Rockwood, A. (2002). Computing singularities of 3D vector fields with geometric algebra. In *Proceedings of the conference on Visualization '02*, pages 283–290. IEEE Computer Society.
- [Max and Weinkauff, 2009] Max, N. and Weinkauff, T. (2009). Critical points of the electric field from a collection of point charges. In *Topology-Based Methods in Visualization II*, pages 101–114. Springer.
- [Molina-Aiz et al., 2010] Molina-Aiz, F., Fatnassi, H., Boulard, T., Roy, J., and Valera, D. (2010). Comparison of finite element and finite volume methods for simulation of natural ventilation in greenhouses. *Computers and electronics in agriculture*, 72(2):69–86.
- [Pan and Skala, 2011] Pan, R. and Skala, V. (2011). A two-level approach to implicit surface modeling with compactly supported radial basis functions. *Engineering with Computers*, 27(3):299–307.
- [Pan and Skala, 2012] Pan, R. and Skala, V. (2012). Surface reconstruction with higher-order smoothness. *The Visual Computer*, 28(2):155–162.
- [Perry and Chong, 1987] Perry, A. E. and Chong, M. S. (1987). A description of eddying motions and flow patterns using critical-point concepts. *Annual Review of Fluid Mechanics*, 19(1):125–155.
- [Platis and Theoharis, 2004] Platis, N. and Theoharis, T. (2004). Simplification of vector fields over tetrahedral meshes. In *Proceedings Computer Graphics International, 2004.*, pages 174–181. IEEE.
- [Prakash et al., 2012] Prakash, G., Kulkarni, M., and Sripathi, U. (2012). Using RBF neural networks and Kullback-Leibler distance to classify channel models in free space optics. In *Optical Engineering (ICOE), 2012 International Conference on*, pages 1–6. IEEE.
- [Press et al., 1989] Press, W. H., Flannery, B. P., Teukolsky, S. A., Vetterling, W. T., et al. (1989). Numerical recipes.

- [Reininghaus and Hotz, 2011] Reininghaus, J. and Hotz, I. (2011). Combinatorial 2D vector field topology extraction and simplification. In *Topological Methods in Data Analysis and Visualization*, pages 103–114. Springer.
- [Reininghaus et al., 2011a] Reininghaus, J., Kotava, N., Guenther, D., Kasten, J., Hagen, H., and Hotz, I. (2011a). A scale space based persistence measure for critical points in 2D scalar fields. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2045–2052.
- [Reininghaus et al., 2011b] Reininghaus, J., Lowen, C., and Hotz, I. (2011b). Fast combinatorial vector field topology. *IEEE Transactions on Visualization and Computer Graphics*, 17(10):1433–1443.
- [Schagen, 1979] Schagen, I. P. (1979). Interpolation in two dimensions - a new technique. *IMA Journal of Applied Mathematics*, 23(1):53–59.
- [Sipeki and Szymczak, 2014] Sipeki, L. and Szymczak, A. (2014). Simplification of morse decompositions using morse set mergers. In *Topological Methods in Data Analysis and Visualization III*, pages 39–53. Springer.
- [Skala, 2013] Skala, V. (2013). Fast interpolation and approximation of scattered multidimensional and dynamic data using radial basis functions. *WSEAS Transaction on Mathematics*, 12(5):501–511.
- [Skala, 2015] Skala, V. (2015). Meshless interpolations for computer graphics, visualization and games. In *Eurographics 2015 - Tutorials*.
- [Skala, 2017] Skala, V. (2017). RBF interpolation with CSRBF of large data sets. *Procedia Computer Science*, 108:2433–2437.
- [Skraba et al., 2016] Skraba, P., Rosen, P., Wang, B., Chen, G., Bhatia, H., and Pascucci, V. (2016). Critical point cancellation in 3D vector fields: Robustness and discussion. *IEEE transactions on visualization and computer graphics*, 22(6):1683–1693.
- [Skraba et al., 2014] Skraba, P., Wang, B., Chen, G., and Rosen, P. (2014). 2D vector field simplification based on robustness. In *Visualization Symposium (PacificVis), 2014 IEEE Pacific*, pages 49–56. IEEE.
- [Skraba et al., 2015] Skraba, P., Wang, B., Chen, G., and Rosen, P. (2015). Robustness-based simplification of 2D steady and unsteady vector fields. *IEEE transactions on visualization and computer graphics*, 21(8):930–944.
- [Smolik and Skala, 2016a] Smolik, M. and Skala, V. (2016a). Vector field interpolation with radial basis functions. In *Proceedings of SIGRAD 2016, May 23rd and 24th, Visby, Sweden*, number 127, pages 15–21. Linköping University Electronic Press.
- [Smolik and Skala, 2016b] Smolik, M. and Skala, V. (2016b). Vector field RBF interpolation on a sphere. In *The Computer Graphics, Visualization, Computer Vision and Image Processing (CGVCVIP)*, pages 352–354. IADIS Press.
- [Smolik and Skala, 2017a] Smolik, M. and Skala, V. (2017a). Classification of critical points using a second order derivative. *Procedia Computer Science*, 108:2373–2377.
- [Smolik and Skala, 2017b] Smolik, M. and Skala, V. (2017b). Spherical RBF vector field interpolation: experimental study. In *2017 IEEE 15th International Symposium on Applied Machine Intelligence and Informatics (SAMII)*, pages 431–434. IEEE.
- [Smolik and Skala, 2017c] Smolik, M. and Skala, V. (2017c). Vector field second order derivative approximation and geometrical characteristics. In *International Conference on Computational Science and Its Applications*, pages 148–158. Springer.
- [Smolik and Skala, 2018] Smolik, M. and Skala, V. (2018). Large scattered data interpolation with radial basis functions and space subdivision. *Integrated Computer-Aided Engineering*, 25(1):49–62.
- [Smolik and Skala, 2019] Smolik, M. and Skala, V. (2019). Efficient simple large scattered 3D vector fields radial basis functions approximation using space subdivision. In *Computational Science and Its Applications – ICCSA 2019*, pages 337–350. Springer.

- [Smolik and Skala, tteda] Smolik, M. and Skala, V. (2019 (submitted)a). Radial basis function and multi-level 2D vector field approximation. *Mathematics and Computers in Simulation*.
- [Smolik and Skala, ttedb] Smolik, M. and Skala, V. (2019 (submitted)b). Vector field radial basis functions approximation with streamlines curvature. *Advances in Engineering Software*.
- [Smolik et al., 2018] Smolik, M., Skala, V., and Majdisova, Z. (2018). Vector field radial basis function approximation. *Advances in Engineering Software*, 123(1):117–129.
- [Smolik et al., 2019] Smolik, M., Skala, V., and Majdisova, Z. (2019). 3D vector field approximation and critical points reduction using radial basis functions. *International journal of mechanics*, 13(1):100–103.
- [Smolik et al., 2016] Smolik, M., Skala, V., and Nedved, O. (2016). A comparative study of lowess and rbf approximations for visualization. In *International Conference on Computational Science and Its Applications*, pages 405–419. Springer.
- [Szymczak, 2013] Szymczak, A. (2013). Hierarchy of stable morse decompositions. *IEEE transactions on visualization and computer graphics*, 19(5):799–810.
- [Szymczak and Zhang, 2012] Szymczak, A. and Zhang, E. (2012). Robust morse decompositions of piecewise constant vector fields. *IEEE Transactions on Visualization and Computer Graphics*, 18(6):938–951.
- [Telea and Van Wijk, 1999] Telea, A. and Van Wijk, J. J. (1999). Simplified representation of vector fields. In *Proceedings of the conference on Visualization '99: celebrating ten years*, pages 35–42. IEEE Computer Society Press.
- [Theisel et al., 2003a] Theisel, H., Rossl, C., and Seidel, H.-P. (2003a). Combining topological simplification and topology preserving compression for 2D vector fields. In *11th Pacific Conference on Computer Graphics and Applications*, pages 419–423. IEEE.
- [Theisel et al., 2003b] Theisel, H., Rössl, C., and Seidel, H.-P. (2003b). Compression of 2D vector fields under guaranteed topology preservation. In *Computer Graphics Forum*, volume 22, pages 333–342. Wiley Online Library.
- [Tricoche et al., 2000] Tricoche, X., Scheuermann, G., and Hagen, H. (2000). A topology simplification method for 2D vector fields. In *Visualization 2000. Proceedings*, pages 359–366. IEEE.
- [Tricoche et al., 2001] Tricoche, X., Scheuermann, G., and Hagen, H. (2001). Continuous topology simplification of planar vector fields. In *Proceedings of the conference on Visualization '01*, pages 159–166. IEEE Computer Society.
- [Tricoche et al., 2002] Tricoche, X., Scheuermann, G., Hagen, H., and Clauss, S. (2002). Vector and tensor field topology simplification, tracking, and visualization. In *PhD. thesis, Schriftenreihe Fachbereich Informatik (3), Universität*. Citeseer.
- [Uhlir and Skala, 2005] Uhlir, K. and Skala, V. (2005). Reconstruction of damaged images using radial basis functions. In *Signal Processing Conference, 2005 13th European*, pages 1–4. IEEE.
- [Van Loan and Golub, 1983] Van Loan, C. F. and Golub, G. H. (1983). *Matrix computations*. Johns Hopkins University Press.
- [Wang et al., 2013] Wang, B., Rosen, P., Skraba, P., Bhatia, H., and Pascucci, V. (2013). Visualizing robustness of critical points for 2D time-varying vector fields. In *Computer Graphics Forum*, volume 32, pages 221–230. Wiley Online Library.
- [Wang et al.,] Wang, W., Wang, W., and Li, S. Detection and classification of critical points in piecewise linear vector fields. *Journal of Visualization*, pages 1–15.
- [Weinkauff et al., 2004] Weinkauff, T., Theisel, H., Hege, H.-C., and Seidel, H.-P. (2004). Topological construction and visualization of higher order 3D vector fields. In *Computer Graphics Forum*, volume 23, pages 469–478. Wiley Online Library.

- [Weinkauff et al., 2005] Weinkauff, T., Theisel, H., Shi, K., Hege, H.-C., and Seidel, H.-P. (2005). Extracting higher order critical points and topological simplification of 3D vector fields. In *Visualization, 2005. VIS 05. IEEE*, pages 559–566. IEEE.
- [Wendland, 1995] Wendland, H. (1995). Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Advances in computational Mathematics*, 4(1):389–396.
- [Wendland, 2006] Wendland, H. (2006). Computational aspects of radial basis function approximation. *Studies in Computational Mathematics*, 12:231–256.
- [Yingwei et al., 1998] Yingwei, L., Sundararajan, N., and Saratchandran, P. (1998). Performance evaluation of a sequential minimal radial basis function (RBF) neural network learning algorithm. *IEEE Transactions on neural networks*, 9(2):308–318.
- [Yuan et al., 2019] Yuan, Z., Zhao, Y., Chen, F., Reber, S., Lu, C.-C., and Chen, Y. (2019). Detail-preserving compression for smoke-based flow visualization. *Journal of Visualization*, 22(1):51–64.
- [Zhang et al., 2000] Zhang, X., Song, K. Z., Lu, M. W., and Liu, X. (2000). Meshless methods based on collocation with radial basis functions. *Computational mechanics*, 26(4):333–343.