Multidimensional Scattered Time-varying Scattered Data Meshless Interpolation for Sensor Networks

Vaclav Skala(✉)

Faculty of Applied Sciences, Department of Computer Science,
University of West Bohemia, 301 00 Pilsen, Czech Republic
skala@kiv.zcu.cz
https://www.vaclavskala.eu/

Abstract. Interpolation and approximation of scattered scalar and vector data is a part of a solution of many engineering problems. The methods are based mostly on some triangulation of the data domain, usually limited to 2D or 3D data, followed by an interpolation or an approximation to obtain a smooth result. This contribution presents a meshless approach based on the Radial Basis Functions (RBF). It is nearly dimensionless and it enables interpolation of time varying data, i.e. interpolation of scattered spatio-temporal varying data, i.e. interpolation in space-time domain without "time-frames". The meshless methods for scattered spatio-temporal data can be used for interpolation, approximation and evaluation of data acquired from buoys, sensor networks, sensors for tsunami, chemical and radiation detectors, ships and submarines detection, weather forecast, 3D vector fields compression and visualization, etc.

Keywords: Meshless method · meshfree method · Radial Basis Functions · scattered data · time-varying data · approximation · interpolation · sensor network · military application · tsunami detection · weather forecast · 3D vector fields · radiation situation · chemical situation

1 Introduction

Data interpolation and approximation is a frequent task in many areas. Usually the interpolation is used for data sets $h_i = f(\mathbf{x_i})$, where h_i is a value at $\mathbf{x_i} \in \Omega$, where Ω is the data domain in E^d and $d = 2, 3$ mostly. The data domain is somehow tessellated, but not necessarily by the Delaunay triangulation(DT). The values h_i might be scalar or vector values, e.g. a wind velocity (v_x, v_y, v_z). In the case of spatio-temporal data, the "framing" is implicitly expected with the fixed known correspondence of points Petrik [33,34]. It leads to spatio-temporal meshes with the fixed connectivity of points in frames t_i and t_{i+1}, if the domain

Research partially supported by the University of West Bohemia - Institutional research support.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2023
O. Gervasi et al. (Eds.): ICCSA 2023, LNCS 13956, pp. 99–112, 2023.
https://doi.org/10.1007/978-3-031-36805-9_7

tessellation is used. It should be noted that if the spatial domain $\Omega \in E^3$ then in the spatio-temporal case a tessellation for E^4 is to be used.

A usual tessellation technique is the Delaunay triangulation(DT).[1]. However, the computational complexity is $O(n^{\lceil d/2+1 \rceil})$, i.e. $O(n^2)$ for $d = 2$, $O(n^3)$ for $d = 3, 4$, see Smolik [44], WiKi [52]. Also, due to numerical robustness issues, the complexity of the DT implementation grows significantly with the dimensionality. The smoothness of the final interpolation of the h_i values is a fundamental requirement, which is not an easy task if triangular or tetrahedral meshes are used to represent the data domain. This approach can be used in the spatio-temporal case, when the data are "framed" for the given "time-slice" t_i and with fixed and known connectivity of points, see Petrik [33,35].

However, there are many areas when an interpolation of scattered spatio-temporal "non-framed" data are required, e.g. sensor networks, floating buoys with sensors on sea, when data sources are not constantly on-line and sending data occasionally only, e.g. tsunami detection, ships and submarine identification. Such sensors are connected only shortly. It leads to energy savings and hard detection of those.

2 Spatio-temporal Data Classification

Interpolation and approximation is usually made for the "ordered" data domains, e.g. rectangular, triangular and tetrahedral meshes, etc. In the CAD systems, the parametric space is used for interpolation, e.g. for parametric curves and surfaces. The data domain can be classified as:

- ordered
  - structured
    - * regular, e.g. a rectangular mesh where all elements have same size, triangular meshes with a constant vertex valency,
    - * irregular, e.g. a rectangular mesh, but elements have different size triangular meshes with non-constant vertex valencies,
  - unstructured, e.g. general triangular or tetrahedral meshes,
- unordered
  - clustered - points form clusters in the data domain,
  - scattered - points are scattered across the domain, generally.

Table 1. Classification of spatio-temporal data sets

		Temporal property	
		Static	Dynamic
Spatial property	Static	$h = f(\mathbf{x})$	$h = f(\mathbf{x}, t)$
	Dynamic	$h = f(\mathbf{x}(t))$	$h = f(\mathbf{x}(t), t)$

[1] For non DT tessellation, see Smolik [45].

The domain data can also be classified as static or dynamic in space and time, see Table 1. It can be seen that the case $h = f(\mathbf{x}(t), t)$ represents interpolation of a scalar value h on the d-dimensional domain of $\mathbf{x}(t) \in \Omega(t)$, where the position $\mathbf{x}(t)$ is changing within time t. It should be noted that the $\Omega(t)$ is not constant in time, generally.

Now, the case of the scattered spatio-temporal domain is dynamic in both, i.e. $h(t) = f(\mathbf{x}(t), t)$, can be classified further as:

- framed in time - points lie on a hyperplane $\rho \in E^d$ for the given time slice t_i,
- framed in space - all points for the given slice in the given E^d space are given (limited to the (x, t) case, i.e. $x \in R^1$),
- unframed in space and time - just an unordered "heap of points" scattered in space-time.

Also, the points in the Ω domain might also be with known mutual point correspondences, i.e. a geometrical trajectory of a point can be reconstructed, e.g. using the buoys ID, or without any similar information, i.e. the buoys ID is not available. In the following, a general approach to interpolation is described and the radial basis functions (RBF) are used for interpolation.

3 Radial Basis Functions

The Radial Basis Functions (RBF) interpolation is based on the mutual distances of points in the data domain Ω, see Hardy [18,19], Schaback [37], Fasshauer [14], Wendland [50,51], Franke [16], Iske [20], Floater [15], Wu [54], Belytschko [6], etc. RBFs are also used in many applications, e.g. Biancolini [7], Cuomo [11], Sarra [36] and Zhang [55].

The RBF interpolation is given in the form:

$$h(\mathbf{x}) = \sum_{j=1}^{N} \lambda_j \varphi \|\mathbf{x} - \mathbf{x}_j\|) = \sum_{j=1}^{N} \lambda_j \varphi(r_j) \tag{1}$$

where r_j is the distance from a point $\mathbf{x}$ to the point $\mathbf{x}_j$. As the parameter r of the function $\varphi(r)$ is a distance of two points in the d-dimensional space, the interpolation is non-separable by a dimension. The RBF function $\varphi(r)$ will be described in detailed later on.

For each point, $\mathbf{x}_i$ the interpolating function has to have the value h_i. It leads to a system of linear equations:

$$h(\mathbf{x}_i) = \sum_{j=1}^{N} \lambda_j \varphi(\|\mathbf{x}_i - \mathbf{x}_j\|) = \sum_{j=1}^{N} \lambda_j \varphi(r_{ij}) \tag{2}$$

where λ_j are unknown weights for each radial basis function, N is the number of given points and $\varphi(r)$ is the radial basis function itself.

The Eq. 2 can be written if the matrix form as:

$$\mathbf{A}\boldsymbol{\lambda} = \mathbf{h} \tag{3}$$

or using $\varphi_{ij} = \varphi(r_{ij})$ as:

$$\begin{bmatrix} \varphi_{11} & \cdots & \varphi_{1j} & \cdots & \varphi_{1N} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \varphi_{i1} & \cdots & \varphi_{ij} & \cdots & \varphi_{iN} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \varphi_{N1} & \cdots & \varphi_{Nj} & \cdots & \varphi_{NN} \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_i \\ \vdots \\ \lambda_N \end{bmatrix} = \begin{bmatrix} h_1 \\ \vdots \\ h_i \\ \vdots \\ h_N \end{bmatrix} \tag{4}$$

The interpolated value at a point $\mathbf{x}$ is computed using the Eq. 1. However, due to numerical robustness and stability, additional polynomial conditions are usually added Fasshauer [14], Skala [39,40].

It can be seen, that the size of the matrix is nearly independent of the dimension and matrix size is $\approx (N \times N)$ only.[2]

In the case of an additional polynomial $P_k(\mathbf{x}_i)$ of a degree k, we obtain:

$$h(\mathbf{x}_i) = \sum_{j=1}^{N} \lambda_j \varphi(||\mathbf{x}_i - \mathbf{x}_j||) + P_k(\mathbf{x}_i) \tag{5}$$

In the case of a bilinear polynomial $P_1(x, y)$:

$$P_1(x, y) = a_0 + a_1 x + a_2 y + a_3 xy \tag{6}$$

the additional orthogonal conditions are to be used:

$$\sum_{j=1}^{N} \lambda_j = 0 \quad \sum_{j=1}^{N} \lambda_j x_j = 0 \quad \sum_{j=1}^{N} \lambda_j y_j = 0 \quad \sum_{j=1}^{N} \lambda_j x_j y_j = 0 \tag{7}$$

Then the RBF interpolation with the additional orthogonal conditions can be rewritten as in a more compact way:

$$\begin{bmatrix} \mathbf{A} & \mathbf{P} \\ \mathbf{P^T} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \mathbf{a} \end{bmatrix} = \begin{bmatrix} \mathbf{h} \\ \mathbf{0} \end{bmatrix} \tag{8}$$

where the matrix $\mathbf{P}$ represents the polynomial, $\boldsymbol{\lambda}$ is the vector of the RBF weights, the vector $\mathbf{a}$ contains resulting the polynomial coefficients and $\mathbf{h}$ are given values at the given points. The matrix $\mathbf{P^T}$ represents the additional orthogonal conditions, see Eq. 7. From the geometrical point of view, the polynomial $P_k(\mathbf{x})$ actually represents a rough approximation of the given data.

It should be noted that if the polynomial $P_k(\mathbf{x})$ is used:

- the RBF interpolation is not invariant to rotation and translation,
- interpolation, i.e. $\boldsymbol{\lambda}$ and $\mathbf{a}$, depends on physical units used for the vector $\mathbf{x}$,
- it might be actually counterproductive in the case of large range of domain data Jäger [21], Skala [39,40].

It can be seen that the RBF interpolation leads to a linear system of equations $\mathbf{Ax} = \mathbf{b}$.

[2] In the case of high N using block matrices might help in solving large systems of linear equations, see Majdisova [28] - a system of 6.7 10^6 points was solved.

3.1 Distance

The $r_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|$ is the distance between points $\mathbf{x}_i$ and $\mathbf{x}_j$, i.e. in the case of:

– spatial data (time independent) - usually the Euclidean norm is used: [3]

$$r_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\| = \sqrt{\sum_{k=1}^{d} ({}^k x_i - {}^k x_j)^2} \tag{9}$$

where ${}^k x_i$ means the kth element of the vector $\mathbf{x}_i$,

– spatio-temporal data (time varying)

$$r_{ij} = \|(\mathbf{x}, t)_i - (\mathbf{x}, t)_j\| = \sqrt{\sum_{k=1}^{d} ({}^k x_i - {}^k x_j)^2 + \beta^2 (t_i - t_j)^2} \tag{10}$$

where ${}^k x_i$ means the kth element of the vector $\mathbf{x}_i$.
The coefficient β has physical unit $[m/s]$ and reflects the speed of the physical phenomena, e.g. speed of sound in water, speed of light, etc.

Another possibility is to use the $\varphi(r)$ RBF with a multiplicative exponential time term as:

$$\phi(r(t), t) = \varphi(r)\ e^{-kt} \qquad \text{or} \qquad \phi(r(t), t) = \varphi(r)\ e^{(-k_1 t^2 - k_2 t)}$$

where k, k_1 and k_2 are some positive constants, see Ku [23].

It should be noted, that if the points $\mathbf{x}$ are not static, i.e. $\mathbf{x} = \mathbf{x}(t)$ then the mutual distances of points are not constant, i.e. $r = r(t)$.

The above-mentioned radial basis functions are used for interpolation and approximation, solution of ordinary differential equations (ODE) and partial differential equations (PDE), etc.

3.2 Squared Normalized RBF

Normalized RBF is another modification of the "standard" RBF. The Normalized RBF (N-RBF) is given in the form:

$$h(\mathbf{x}) = \frac{\sum_{j=1}^{N} \lambda_j \varphi \|\mathbf{x} - \mathbf{x}_j\|)}{\sum_{j=1}^{N} \varphi \|\mathbf{x} - \mathbf{x}_j\|} = \frac{\sum_{j=1}^{N} \lambda_j \varphi(r_j)}{\sum_{j=1}^{N} \varphi(r_j)} \tag{11}$$

where r_j is the distance from a point $\mathbf{x}$ to the point $\mathbf{x}_j$.

The N-RBF are used especially in the RBF neural networks applications, see WiKi [53].

[3] The distance defined as $\|.\|_1$ is used in fuzzy approach, see Perfilieva [32].

However, some functions $\varphi(r)$ used for the interpolation and approximation are not strictly positive, e.g. $r^2 \ln(r)$ (Thin-Plate Spline - TPS) [4], which is negative on the interval $(0, 1)$, the Euclidean norm should be used for robustness of computation, see Eq. 12.

The Squared Normalized RBF (SN-RBF) is given in the form:

$$h(\mathbf{x}) = \frac{\sum_{j=1}^{N} \lambda_j \varphi(\|\mathbf{x} - \mathbf{x}_j\|)}{\sqrt{\sum_{j=1}^{N} \varphi^2(\|\mathbf{x} - \mathbf{x}_j\|)}} = \frac{\sum_{j=1}^{N} \lambda_j \varphi(r_j)}{\sqrt{\sum_{j=1}^{N} \varphi^2(r_j)}} \tag{12}$$

This is actually the Euclidean normalization of each row of the matrix $\mathbf{A}$ in Eq. 4.

$$\sum_{j=1}^{N} \lambda_j \varphi(\|\mathbf{x}_i - \mathbf{x}_j\|) = h(\mathbf{x}) \sqrt{\sum_{j=1}^{N} \varphi^2(r_{ij})} \quad i = 1, \ldots, N \tag{13}$$

where $r_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|$.

It should be noted, that $O(N^2)$ division operations are replaced by $O(N)$ multiplications; the given values $h(\mathbf{x}_i)$ are just multiplied by the values $\sqrt{\sum_{j=1}^{N} \varphi^2(r_{ij})}$. It leads to higher robustness of computation for high N.

In the case of interpolation a polynomial of a degree k $P_k(\mathbf{x})$ can be added:

$$h(\mathbf{x}) = \frac{\sum_{j=1}^{N} \lambda_j \varphi \|\mathbf{x} - \mathbf{x}_j\|)}{\sqrt{\sum_{j=1}^{N} \varphi^2(r_j)}} + P_k(\mathbf{x}) \tag{14}$$

and some orthogonal conditions have to be added as well, see Eq. 7.

The polynomial $P_k(\mathbf{x})$ improves the conditionality of the matrix $\mathbf{A}$, see Eq. 3, and also roughly approximate the given data. The Eq. 14 can be modified similarly as the Eq. 13 to:

$$\sum_{j=1}^{N} \lambda_j \varphi \|\mathbf{x}_i - \mathbf{x}_j\|) + P_k(\mathbf{x}_i) \sqrt{\sum_{j=1}^{N} \varphi^2(r_{ij})} = h(\mathbf{x}_i) \sqrt{\sum_{j=1}^{N} \varphi^2(r_{ij})} \quad i = 1, \ldots, N \tag{15}$$

As the $q_i = \sqrt{\sum_{j=1}^{N} \varphi^2(r_{ij})}$ is constant for the ith row $(i = 1, \ldots, N)$ in the Eq. 4, the Eq. 15 can be simplified to:

$$\sum_{j=1}^{N} \lambda_j \varphi \|\mathbf{x}_i - \mathbf{x}_j\|) + q_i \; P_k(\mathbf{x}_i) = q_i \; h(\mathbf{x}_i) \quad i = 1, \ldots, N \tag{16}$$

[4] It should be noted that $r^2 \log(r) = \frac{1}{2} r^2 \log(r^2)$, i.e. actually no $\sqrt{r^2}$ computation is needed, only the values of the weights λ_j are doubled.

and in the matrix for the SN-RBF interpolation has form:

$$\begin{bmatrix} \varphi_{11} & \dots & \varphi_{1N} & q_1 & q_1x_1 & q_1y_1 & q_1x_1y_1 \\ \vdots & \ddots & \vdots & 1 & \vdots & \vdots & \vdots \\ \varphi_{N1} & \dots & \varphi_{NN} & q_N & q_Nx_N & q_Ny_N & q_Nx_ny_N \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ x_1 & \dots & x_N & 0 & 0 & 0 & 0 \\ y_1 & \dots & y_N & 0 & 0 & 0 & 0 \\ x_1y_1 & \dots & x_Ny_N & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_N \\ a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} q_1h_1 \\ \vdots \\ q_Nh_N \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{17}$$

or in a more compact way:

$$\begin{bmatrix} \mathbf{A} & \mathbf{QP} \\ \mathbf{P^T} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \mathbf{a} \end{bmatrix} = \begin{bmatrix} \mathbf{Qh} \\ \mathbf{0} \end{bmatrix} \tag{18}$$

where $\mathbf{Q} = diag[q_1, \dots, q_N]$ is a diagonal matrix and $q_i > 0$, $i = 1, \dots, N$ in the case of SN-RBFs.

However, in the approximation case, i.e. the matrix $\mathbf{A}$ is $(N \times M), N > M$, the polynomial part has to be handled differently and the Least Square Method (LSE) cannot be used directly, see Skala [42], Majdisova [26,29,30].

4 RBF Functions

There are several radial basis functions Fasshauer [14], Buhmann [8],Majdisova [28,29] and Smolik [46]. The RBFs can be divided into two major groups:

- "global" RBFs having global influence, e.g.
  - Polyharmonic spline:
    $$\varphi(r) = r^k, \quad k = 1, 3, 5, \dots$$
    $$\varphi(r) = r^k \ln r, \quad k = 2, 4, 6, \dots$$
  - Thin plate spline [TPS](a special polyharmonic spline): $\varphi(r) = r^2 \ln r$, [5]
  - Gaussian: $\varphi(r) = e^{-\alpha r^2}$
  - Multiquadric: $\sqrt{1 + \alpha r^2}$
  - Inverse quadratic: $\frac{1}{1+\alpha r^2}$
  - Inverse multiquadratic: $\frac{1}{\sqrt{1+\alpha r^2}}$

  where $\alpha > 0$ is a shape parameter, $r \in < 0, \infty)$.
  The RBF matrix is usually full and the matrix can be very ill conditioned.
- "local" RBFs - Compactly Supported RBF (CS-RBF) have a non-zero positive value on the interval $< 0, 1)$ only. The RBF matrix is usually sparse and it depends on the shape parameter α. Some function examples are listed in Table 2.
  There are other RBFs used, e.g. the "bumb CS-RBF" based on Gaussian:

$$\varphi(r) = \begin{cases} e^{-\frac{1}{1-r^2}} & r \in < 0, 1) \\ 0 & r \geq 1 \end{cases}$$

[5] In the actual implementation $\varphi(r) = r^2 \ln r^2$ should be used as $r^2 \ln r^2 = 2\ r^2 \ln r$. The λ weights are doubled and $\sqrt{\cdot}$ operation is not needed.

Buhmann [8] proposed CS-RBF in the form:

$$\varphi(r) = \begin{cases} \frac{1}{3} + r^2 - \frac{4}{3}r^3 + 2r^2 \log r & r \in < 0, 1) \\ 0 & r \geq 1 \end{cases}$$

and Manandro [31] proposed two new rational CS-RBF classes.

Table 2. Wendland's Compactly Supported RBF (CS-RBF) $(x)_+$ means, that the value of the expression is zero for $r \geq 1$

ID	RBF	Function	ID	RBF	Function
1	$\varphi_{1,0}$	$(1-r)_+$	2	$\varphi_{1,1}$	$(1-r)_+^3(3r+1)$
3	$\varphi_{1,2}$	$(1-r)_+^5(8r^2+5r+1)$	4	$\varphi_{3,0}$	$(1-r)_+^2$
5	$\varphi_{3,1}$	$(1-r)_+^4(4r+1)$	6	$\varphi_{3,2}$	$(1-r)_+^6(35r^2+18r+3)$
7	$\varphi_{3,3}$	$(1-r)_+^8(32r^3+25r^2+8r+3)$	8	$\varphi_{5,0}$	$(1-r)_+^3$
9	$\varphi_{5,1}$	$(1-r)_+^3(5r+1)$	10	$\varphi_{5,2}$	$(1-r)_+^7(16r^2+7r+1)$

5 RBF Interpolation Example

For demonstration of the RBF spatio-temporal scattered data interpolation properties, the Rainfall data of the Peninsular Malaysia from the Malaysian Meteorology Department in 2007 [4] were taken. The data are static from the spatial point of view and dynamic from the temporal one.

5.1 Algorithm Based on Triangulation and Timmer Patches

One possible approach is to make the domain tessellation, then subdivide the mesh and then smooth it. Ali [4] used Delaunay triangulation on 25 major meteorological stations positions and used cubic Timmer triangular patches, see Ali [2,3,5], for the triangular surface representation and interpolation. The Ali's algorithm is presented by Algorithm 1 and the final triangular surface is presented in Table 3.

It should be noticed, that additional operations are need in order to obtain a smooth surface over the triangular mesh generated by the Ali's algorithm.

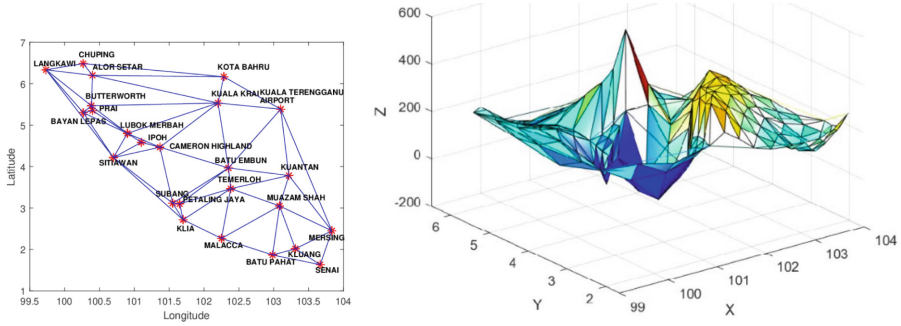
In the following, a solution based on the RBF interpolation is presented.

5.2 RBF Interpolation

The RBF approach leads to smooth surface automatically, however, interpolation on borders might not be reliable. Examples of the data interpolation are presented in figures in Table 4[6].

[6] The SN-RBF with the Gauss function and bilinear polynomial was used; parameters $\alpha = 0.255$ and $\beta = 1$ in Eq. 10.

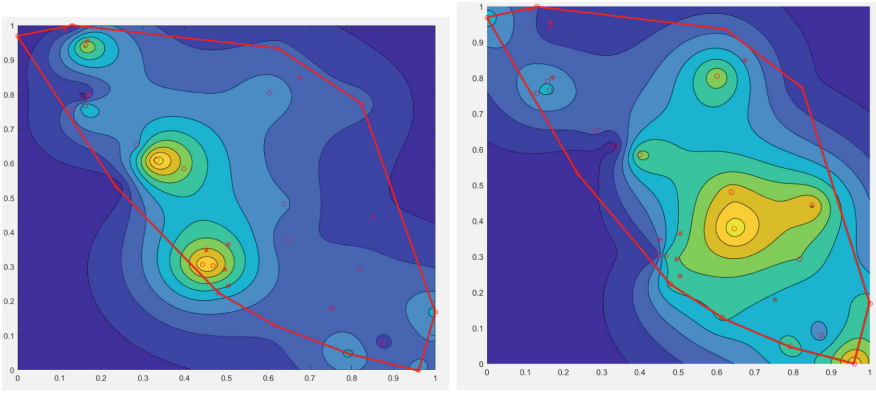
Table 3. Tessellation of the Peninsular Malaysia and Triangular surface generated by Ali's algorithm for May-2007; courtesy Ali [4]



Algorithm 1. Ali's Construction scattered data interpolation algorithm

1: Input: Data points
2: Triangulate the domain by using Delaunay triangulation.
3: Specify the derivatives at the data points using [17] then assign Timmer ordinates values for each triangular patch.
4: Generate the triangular patches of the surfaces by using cubic Timmer triangular patches.
5: Output: Surface reconstruction.

Table 4. Interpolation of the Rainfall data at Peninsular Malaysia in 2007; the polygon represents a convex hull of the meteorological stations positions



March 2007 May 2007

The 3D views of the Rainfall interpolated data are presented in Fig. 1 and Fig. 2. It should be noted that the α shape parameter has to be set *reasonably*[7].

In general, the RBF interpolation benefits from the following properties:

- the final interpolation is naturally smooth,
- if CS-RBFs are used, the RBF matrix is sparse,
- computational complexity is nearly independent of the dimensionality, depends on the number of points,
- an explicit formal analytical formula of the final interpolation is obtained, see Eq. 12, Eq. 14 and Eq. 15,
- if the RBF function $\varphi(r)$ is positive definite, iterative methods for solving linear system of equations can be used,
- the SN-RBF increases the numerical stability as it actually normalizes each row of the RBF matrix Skala [38,41], Cervenka [10],
- as a solution of linear system of equations is equivalent to the outer product (extended cross product) use, the standard symbolic operations can be used for further processing *without need of the numerical evaluation*,
- the block matrix decomposition might be used in the case of large data, see Majdisova [27,28],
- the domain decomposition can be applied, which leads to faster computation, see Smolik [46–48].

It should be noted, that the RBFs are also used in solution of partial differential equations (PDEs) or surface reconstruction of the acquired data Carr [9], Drake [13] Cuomo [11,12] and Macedo [24,25].

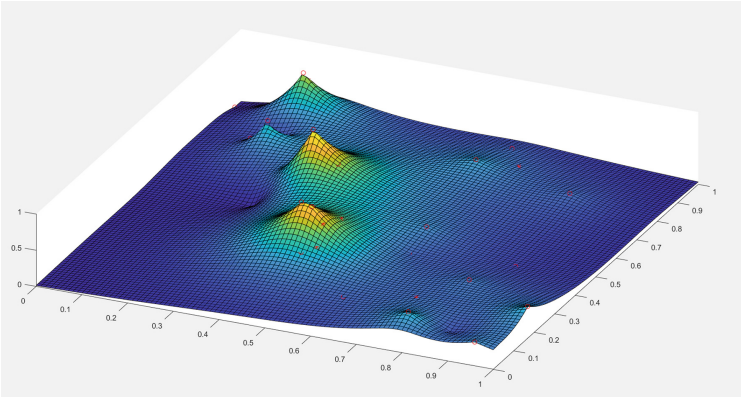


Fig. 1. Interpolation of the Rainfall data March 2007 the Gauss function used, shape parameter $\alpha = 0.255$

[7] There are several suboptimal shape parameters, see Afiatdoust [1], Skala [43], Karageorghis [22] and Wang [49].

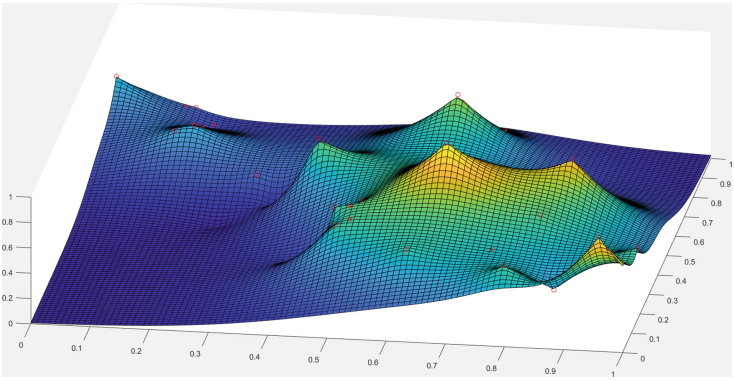


Fig. 2. Interpolation of the Rainfall data May 2007 the Gauss function used, shape parameter $\alpha = 0.2550$

6 Conclusion

This contribution briefly presents a new form of normalized RBF, the Squared Normalized RBF SN-RBF, for the spatio-temporal data and a perspective of use in sensor networks, when sensors do not have a fixed position and data are not synchronized in time, i.e. in the time-slots. The proposed SN-RBF formulation leads to better RBF matrix conditionality, too.

The presented SN-RBF interpolation method is especially convenient for scattered spatio-temporal interpolation of data cases, e.g. when the sensors are transmitting data only in the case, when physical phenomena reach values outside the expected range and/or changing their positions, e.g. surveillance sensors in the sea, etc.

Future work is to be targeted to analysis of the β parameter setting and its sensitivity, to the TPS $(r^2 \ln r)$ function applicability for large data sets, conditionality of the resulting RBF matrices and to methods for visualization of spatio-temporal data for 2D+T and 3D+T data as they cannot be visualized using methods like contour plots or 3D projections.

Acknowledgments. The author thanks to colleagues at the Shandong University(Jinan) and Zhejiang University(Hangzhou) China, University of West Bohemia, Pilsen for their critical comments, discussions, especially to colleagues Michal Smolik, Zuzana Majdisova, Martin Cervenka, Mariia Martynova and Jan Kasak for producing some images and for numerical verification.

References

1. Afiatdoust, F., Esmaeilbeigi, M.: Optimal variable shape parameters using genetic algorithm for radial basis function approximation. Ain Shams Eng. J. **6**, 639–647 (2015)

2. Ali, F.A.M., Karim, S.A.A., Dass, S.C., Skala, V., Hasan, M.K., Hashim, I.: Efficient visualization of scattered energy distribution data by using cubic timmer triangular patches. In: Sulaiman, S.A. (ed.) Energy Efficiency in Mobility Systems, pp. 145–180. Springer, Singapore (2020). https://doi.org/10.1007/978-981-15-0102-9_8
3. Ali, F., et al.: New cubic timmer triangular patches with c1 and g1 continuity. Jurnal Teknologi **81**(6), 1–11 (2019)
4. Ali, F., et al.: Construction of cubic timmer triangular patches and its application in scattered data interpolation. Mathematics **8**, 159 (2020)
5. Ali, F.A.M., et al.: Visualizing the energy of scattered data by using cubic timmer triangular patches. J. Phys. **1366**(1), 012098 (2019)
6. Belytschko, T., Krongauz, Y., Organ, D., Fleming, M., Krysl, P.: Meshless methods: an overview and recent developments. Comput. Methods Appl. Mech. Eng. **139**(1–4), 3–47 (1996)
7. Biancolini, M.E.: Fast Radial Basis Functions for Engineering Applications, 1st edn. Springer, Heidelberg (2017). https://doi.org/10.1007/978-3-319-75011-8
8. Buhmann, M.: On quasi-interpolation with radial basis functions. J. Approx. Theory **72**(1), 103–130 (1993)
9. Carr, J., et al.: Reconstruction and representation of 3d objects with radial basis functions. In: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 2001, pp. 67–76 (2001)
10. Červenka, M., Skala, V.: Conditionality analysis of the radial basis function matrix. In: Gervasi, O., Murgante, B., Misra, S., Garau, C., Blečić, I., Taniar, D., Apduhan, B.O., Rocha, A.M.A.C., Tarantino, E., Torre, C.M., Karaca, Y. (eds.) ICCSA 2020. LNCS, vol. 12250, pp. 30–43. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58802-1_3
11. Cuomo, S., Galletti, A., Giunta, G., Marcellino, L.: Reconstruction of implicit curves and surfaces via rbf interpolation. Appl. Numer. Math. **116**, 157–171 (2017)
12. Cuomo, S., Galletti, A., Giunta, G., Starace, A.: Surface reconstruction from scattered point via rbf interpolation on gpu. In: 2013 Federated Conference on Computer Science and Information Systems, FedCSIS 2013, pp. 433–440 (2013)
13. Drake, K.P., Fuselier, E.J., Wright, G.B.: Implicit surface reconstruction with a curl-free radial basis function partition of unity method (2021)
14. Fasshauer, G.: Meshfree Approximation Methods with Matlab, 1st edn. World Scientific, Singapore (2007)
15. Floater, M.S., Iske, A.: Multistep scattered data interpolation using compactly supported radial basis functions. J. Comput. Appl. Math. **73**(1–2), 65–78 (1996)
16. Franke, R.: A critical comparison of some methods for interpolation of scattered data. Technical report. Naval Postgraduate School Monterey, CA (1979)
17. Goodman, T., Said, H., Chang, L.: Local derivative estimation for scattered data interpolation. Appl. Math. Comput. **68**(1), 41–50 (1995)
18. Hardy, R.L.: Multiquadric equations of topography and other irregular surfaces. J. Geophys. Res. **76**, 1905–1915 (1971)
19. Hardy, R.L.: Theory and applications of the multiquadric-biharmonic method 20 years of discovery 1968–1988. Comput. Math. Appl. **19**, 163–208 (1990)
20. Iske, A.: Multiresolution Methods in Scattered Data Modelling, vol. 37. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-642-18754-4
21. Jäger, J.: Advances in radial and spherical basis function interpolation. PhD thesis, Justus-Liebig-Universität, Otto-Behaghel-Str. 8, 35394 Gießen. PhD Thesis (2018)
22. Karageorghis, A., Tryfonos, P.: Shape parameter estimation in RBF function approximation. Int. J. Comput. Methods Exp. Meas. **7**, 246–259 (2019)

23. Ku, C.-Y., Hong, L.-D., Liu, C.-Y., Xiao, J.-E.: Space-time polyharmonic radial polynomial basis functions for modeling saturated and unsaturated flows. Eng. Comput. (2021)
24. Macedo, I., Gois, J., Velho, L.: Hermite interpolation of implicit surfaces with radial basis functions. In: Proceedings of SIBGRAPI 2009–22nd Brazilian Symposium on Computer Graphics and Image Processing, pp. 1–8 (2009)
25. Macedo, I., Gois, J., Velho, L.: Hermite radial basis functions implicits. Comput. Graph. Forum **30**(1), 27–42 (2011)
26. Majdisova, Z., Skala, V.: A new radial basis function approximation with reproduction. In: CGVCVIP 2016, pp. 215–222 (2016)
27. Majdisova, Z., Skala, V.: A radial basis function approximation for large datasets. In: SIGRAD 2016, pp. 9–14 (2016)
28. Majdisova, Z., Skala, V.: Big geo data surface approximation using radial basis functions: a comparative study. Comput. Geosci. **109**, 51–58 (2017)
29. Majdisova, Z., Skala, V.: Radial basis function approximations: comparison and applications. Appl. Math. Model. **51**, 728–743 (2017)
30. Majdisova, Z., Skala, V., Smolik, M.: Algorithm for placement of reference points and choice of an appropriate variable shape parameter for the RBF approximation. Integr. Comput. Aided Eng. **27**, 1–15 (2020)
31. Menandro, F.: Two new classes of compactly supported radial basis functions for approximation of discrete and continuous data. Eng. Rep. **1**, e12028:1–30 (2019)
32. Perfilieva, I., Vlasanek, P., Wrublova, M.: Fuzzy Transform for Image Reconstruction, pp. 615–620. World Scientific, Singapore (2012)
33. Petrik, S., Skala, V.: Iso-contouring in time-varying meshes. In: SCCG 2007: 23rd Spring Conference on Computer Graphics, pp. 175–182 (2007)
34. Petrik, S., Skala, V.: Z-diamonds: a fast iso-surface extraction algorithm for dynamic meshes. MCCSIS **2007**(3), 67–74 (2007)
35. Petrik, S., Skala, V.: Space and time efficient isosurface extraction. Comput. Graph. (Pergamon) **32**(6), 704–710 (2008)
36. Sarra, S.A., Sturgill, D.: A random variable shape parameter strategy for radial basis function approximation methods. Eng. Anal. Bound. Elements **33**, 1239–1245 (2009)
37. Schaback, R.: Optimal geometric hermite interpolation of curves, mathematical methods for curves and surfaces. ii. Innov. Appl. Math., 417–428 (1998)
38. Skala, V.: High dimensional and large span data least square error: numerical stability and conditionality. Int. J. Appl. Phys. Math. **7**(3), 148–156 (2017)
39. Skala, V.: RBF interpolation with CSRBF of large data sets. Proceedia Comput. Sci. **108**, 2433–2437 (2017)
40. Skala, V.: RBF interpolation and approximation of large span data sets. In: MCSI 2017 - Corfu, pp. 212–218. IEEE (2018)
41. Skala, V.: Conditionality of linear systems of equation and matrices using projective geometric algebra. In: Gervasi, O., Murgante, B., Misra, S., Garau, C., Blečić, I., Taniar, D., Apduhan, B.O., Rocha, A.M.A.C., Tarantino, E., Torre, C.M., Karaca, Y. (eds.) ICCSA 2020. LNCS, vol. 12250, pp. 3–17. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58802-1_1
42. Skala, V., Kansa, E.: Why is the least square error method dangerous? Computacion y Sistemas **25**(1), 149–151 (2021)
43. Skala, V., Karim, S.A.A., Zabran, M.: Radial basis function approximation optimal shape parameters estimation. In: Krzhizhanovskaya, V.V., et al. (eds.) ICCS 2020. LNCS, vol. 12142, pp. 309–317. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-50433-5_24

44. Smolik, M., Skala, V.: Fast parallel triangulation algorithm of large data sets in E^2 and E^3 for in-core and out-core memory processing. In: Murgante, B., et al. (eds.) ICCSA 2014. LNCS, vol. 8580, pp. 301–314. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-09129-7_23
45. Smolik, M., Skala, V., In-core and out-core memory fast parallel triangulation algorithm for large data sets in E2 and E3. In: ACM SIGGRAPH,: Posters. SIGGRAPH 2014 (2014)
46. Smolik, M., Skala, V.: Large scattered data interpolation with radial basis functions and space subdivision. Integr. Comput. Aided Eng. **25**, 49–62 (2018)
47. Smolik, M., Skala, V.: Efficient simple large scattered 3D vector fields radial basis function approximation using space subdivision. In: Computational Science and Its Application, ICSSA 2019 proceedings, pp. 337–350 (2019)
48. Smolik, M., Skala, V.: Efficient speed-up of radial basis functions approximation and interpolation formula evaluation. In: Gervasi, O., et al. (eds.) ICCSA 2020. LNCS, vol. 12249, pp. 165–176. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58799-4_12
49. Wang, J., Liu, G.: On the optimal shape parameters of radial basis functions used for 2-d meshless methods. Comput. Methods Appl. Mech. Eng. **191**, 2611–2630 (2002)
50. Wendland, H.: Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. Adv. Comput. Math. **4**(1), 389–396 (1995)
51. Wendland, H.: Scattered Data Approximation. Cambridge University Press, Cambridge (2004)
52. Wikipedia contributors. Delaunay triangulation – Wikipedia, the free encyclopedia (2022). Accessed 9 June 2022
53. Wikipedia contributors. Radial basis function network – Wikipedia, the free encyclopedia (2022). Accessed 22 June 2022
54. Wu, Z.: Compactly supported positive definite radial functions. Adv. Comput. Math. **4**(1), 283–292 (1995)
55. Zhang, X., Song, K.Z., Lu, M.W., Liu, X.: Meshless methods based on collocation with radial basis functions. Comput. Mech. **26**, 333–343 (2000)