Radial Basis Functions: Meshless Interpolation and Approximation Methods

Vaclav Skala
University of West Bohemia, Pilsen
Czech Republic
www.VaclavSkala.eu

Problem formulations:

Interpolation or approximation – **explicit functions**

$y = f(x_1, \dots, x_d)$, resp. $\boldsymbol{y} = \boldsymbol{f}(x_1, \dots, x_d)$

scalar field case vector field case, e.g. flow

Interpolation or approximation – **implicit functions** – finding a curve in E^2 or a surface in E^3.

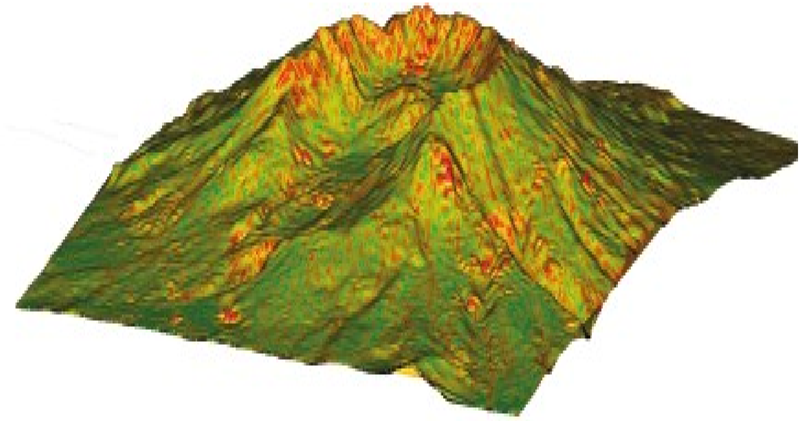
$\boldsymbol{x} = [x_1, \dots, x_d]^T$ $F(x_1, \dots, x_d) = 0$ $F(\boldsymbol{x}) = 0$

Generally, points $\boldsymbol{x}_i$, $i = 1, \dots N$ are scattered in the domain.

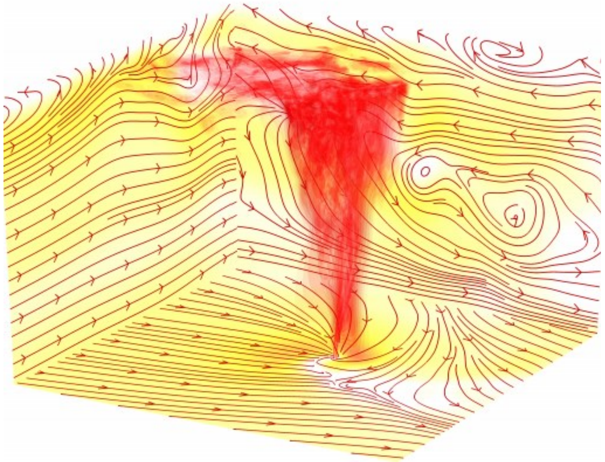
Task

Find an analytical function $f(\boldsymbol{x})$, resp. $F(\boldsymbol{x})$, which interpolates, resp. approximate the given **scattered** data, generally for d-dimensional case.

In the following, meshless methods for explicit functions interpolation or approximation based on Radial Basis Functions (RBF) will be shortly introduced.

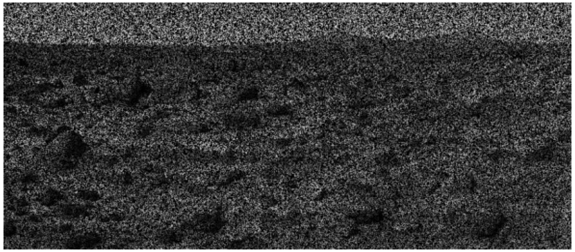


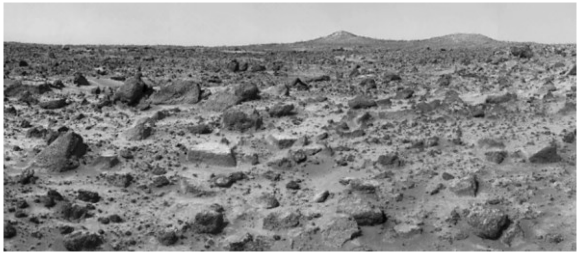
Surface approximation
Mount Saint Helens dataset $6.7\ 10^6$. points



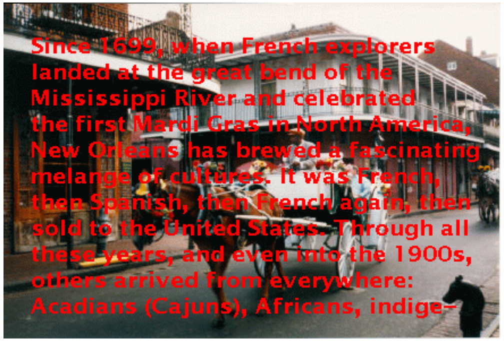
Tornado - Vector fields (speed) approximation
$5.5\ 10^6$ points - after $1{:}\ 7 * 10^3$ compression
with an error around 1%

Image reconstruction

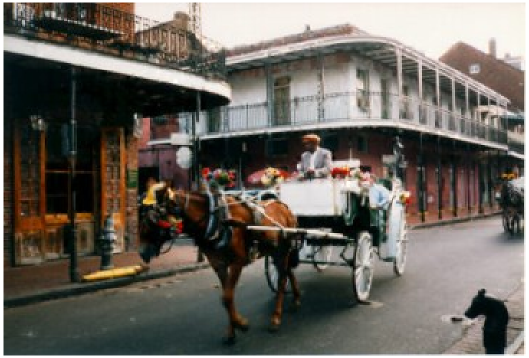

Original image with 60% missing pixels


Reconstructed image

Inpainting and cracks removal


Since 1699, when French explorers landed at the great bend of the Mississippi River and celebrated the first Mardi Gras in North America, New Orleans has brewed a fascinating melange of cultures. It was French, then Spanish, then French again, then sold to the United States. Through all these years, and even into the 1900s, others arrived from everywhere: Acadians (Cajuns), Africans, indige–

Original image

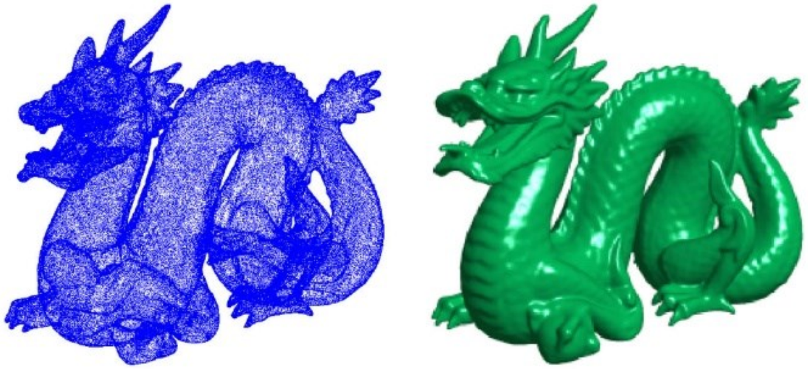

Reconstructed image without inpainting

- Interpolation and approximation of color images – RGB is not the best color system

Surface reconstruction from points

Different problem

- Given only points in the space without associated values.



Surface reconstruction (438 000 points) [Carr et al. 2001]

Some additional conditions, presumption must be specified in order to reconstruct a surface from a cloud of points

- Application – reverse engineering (CAD/CAM systems etc.)

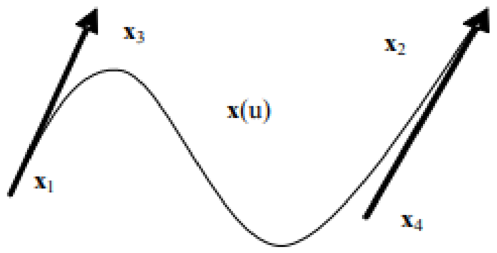
Interpolation and approximation of structured data

- Explicit functions - polynomial regression, etc.

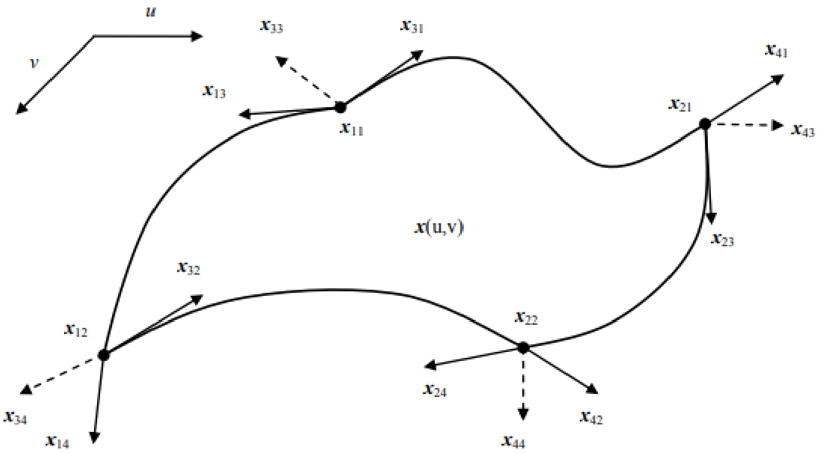
$$y = f(\boldsymbol{x}) \qquad f(x,y) = a_0 + a_1 \log x + a_2 x \log x + a_3 x^2 + a_4 x^3 \log x + a_5 xy + a_6 xy^2 + \cdots$$

- Parametric curves and surfaces

$$x = f_x(t), \;\; y = f_y(t), \;\; z = f_z(t)$$
$$x(u,v) = f_x(u,v), \;\; y(u,v) = f_y(u,v), \;\; z = f_z(u,v)$$

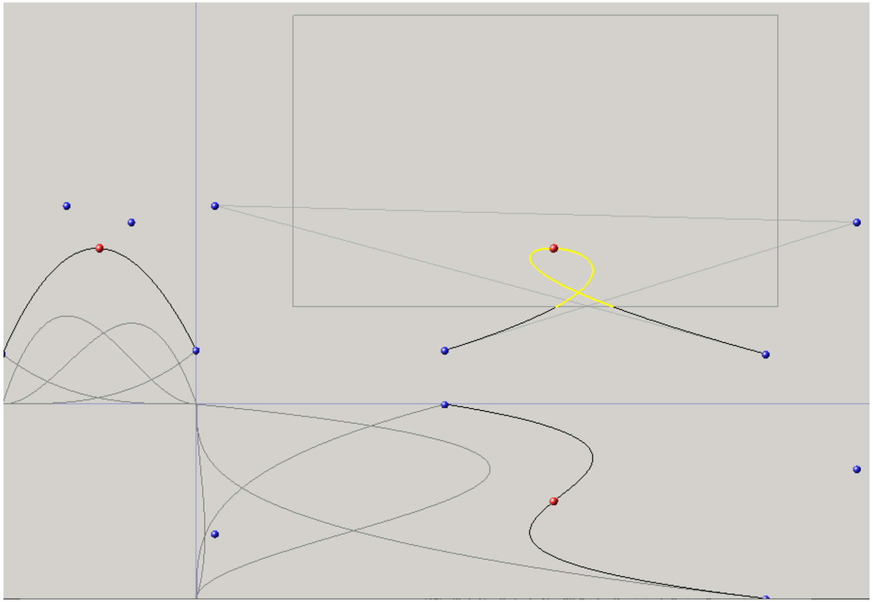


$$\boldsymbol{P}(t) = \boldsymbol{P}^T \boldsymbol{M}_H\, t$$

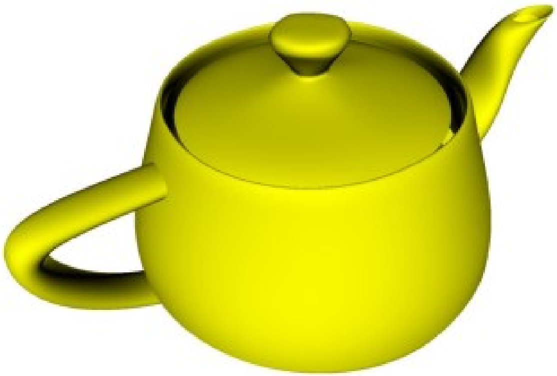


$$\boldsymbol{x}(u,v) = \boldsymbol{u}^T \boldsymbol{M}_H^T\, \boldsymbol{X}\, \boldsymbol{M}_H\, v$$

Examples



Bezier parametric curve

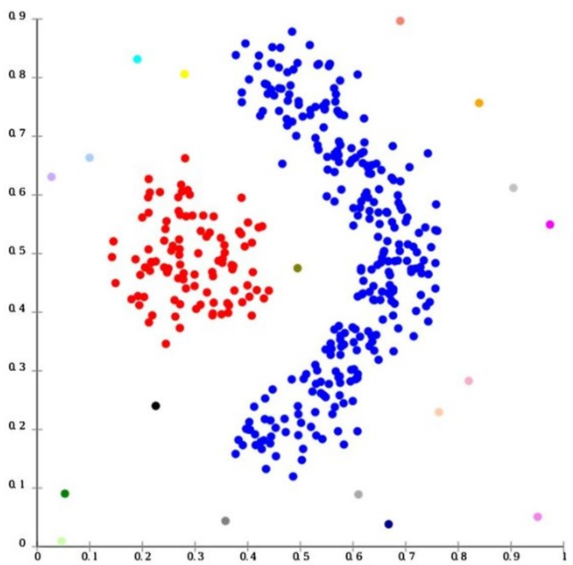


Tea-Pot

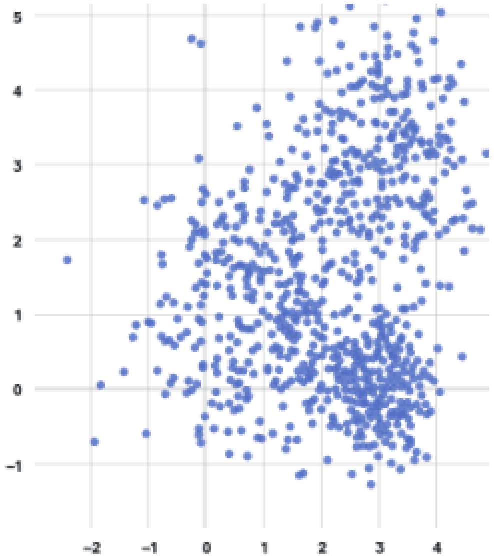
Advantages:

- Simple computation, resp. evaluation

Unstructured data



Clustered data example

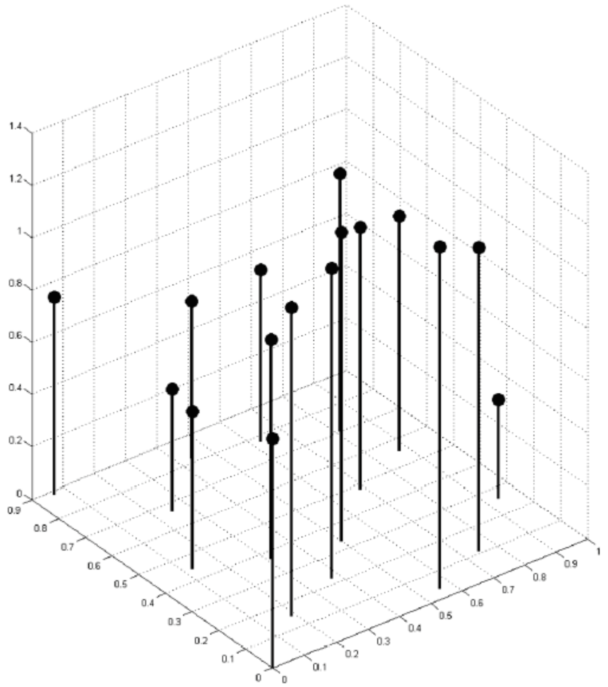


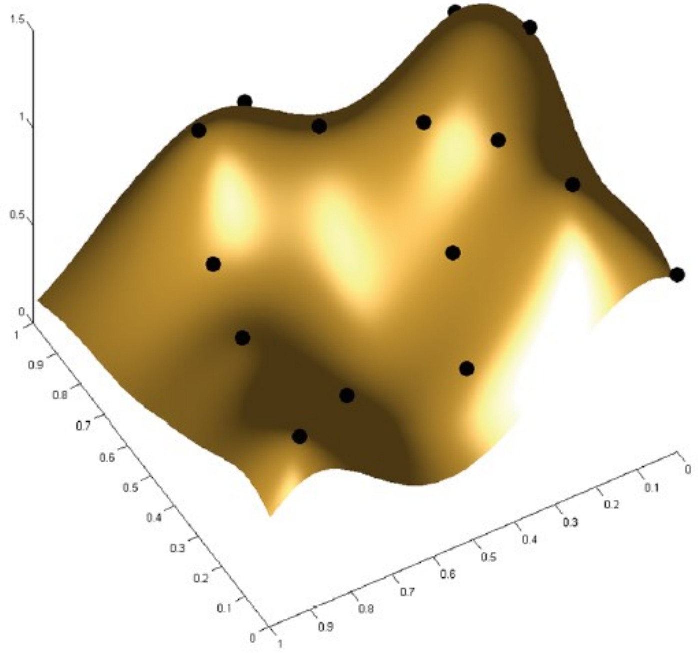
Scattered data example

Generally, with each point a scalar data are associated, e.g. temperature, pressure, or in the case of vector data – speed, acceleration or rotation etc.

Scattered data

Scalar data interpolation or approximation





Scalar values at different scattered positions

Approximation of the given data

This is the case of $z = f(x, y)$, i.e. two dimensional data domain case with associated scalar value.

Possible solution and problems:

- Tessellation of the domain to 2D or 3D meshes, e.g. using Delaunay triangulation/tetraheronization.
  Problem – computational complexity

$$O(N^{\lfloor (d^2+1)/2 \rfloor})$$

which is for high N prohibitive (usual size of $10^4 - 10^8$ elements processed)

- The Delaunay triangulation is not numerically stable if points are closed do regular grid and also some computational methods prefer different kinds of tessellation as it produces long thin triangles/tetrahedrons.

- Smooth interpolation of the associated values, i.e. height, temperature, speed etc., over triangular or tetrahedral mesh is not easy

- Approximation, i.e. reduction is very difficult as it must take also associated values, e.g. of the physical phenomena, not only as simplification of the definition domain

Radial Basis Functions

The Radial Basis Function (RBF) interpolation is based on computing of the distance of two points in the d-dimensional space and it is defined by a function:

$$f(\boldsymbol{x}) = \sum_{j=1}^{M} \lambda_j \, \varphi(\|\boldsymbol{x} - \boldsymbol{x}_j\|) = \sum_{j=1}^{M} \lambda_j \, \varphi(r_j)$$

where: $r_j = \|\boldsymbol{x} - \boldsymbol{x}_j\|_2 \overset{\text{def}}{=} \sqrt{(x - x_j)^2 + (y - y_j)^2}$

It means that for the given data set $\{\langle \boldsymbol{x}_i, h_i \rangle\}_1^M$, where h_i are associated values to be interpolated and $\boldsymbol{x}_i$ are domain coordinates, we obtain a linear system of equations:

$$h_i = f(\boldsymbol{x}_i) = \sum_{j=1}^{M} \lambda_j \, \varphi(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|) \; + P_k(\boldsymbol{x}_i) \qquad i = 1, \dots, M \qquad \boldsymbol{x} = [x, y, 1]^T$$

Due to some stability issues, usually a polynomial $P_k(\boldsymbol{x})$ of a degree *k* is added.

For a practical use, the polynomial of the 1st degree is used, i.e. linear polynomial $P_1(\boldsymbol{x}) = \boldsymbol{a}^T \boldsymbol{x}$ in many applications.

Therefore, the interpolation function has the form:

$$f(\boldsymbol{x}_i) = \sum_{j=1}^{M} \lambda_j \, \varphi(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|) + \boldsymbol{a}^T \boldsymbol{x_i} = \sum_{j=1}^{M} \lambda_j \, \varphi_{i,j} + \boldsymbol{a}^T \boldsymbol{x_i} \qquad h_i = f(\boldsymbol{x}_i) \qquad i = 1, \dots, M$$

and additional conditions are to be applied:

$$\sum_{j=1}^{M} \lambda_i \boldsymbol{x}_i = \boldsymbol{0} \qquad \text{i.e.} \qquad \sum_{j=1}^{M} \lambda_i x_i = 0 \qquad \sum_{j=1}^{M} \lambda_i y_i = 0 \qquad \sum_{j=1}^{M} \lambda_i = 0$$

It can be seen that for d-dimensional case a system of $(M + d + 1)$ LSE has to be solved, where *M* is a number of points in the dataset and d is the dimensionality of data.

For $d = 2$ vectors $\boldsymbol{x}_i$ and $\boldsymbol{a}$ are in the form $\boldsymbol{x}_i = [x_i, y_i, 1]^T$ and $\boldsymbol{a} = [a_x, a_y, a_0]^T$, we can write :

$$\begin{bmatrix} \varphi_{1,1} & .. & \varphi_{1,M} & x_1 & y_1 & 1 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ \varphi_{M,1} & .. & \varphi_{M,M} & x_M & y_M & 1 \\ x_1 & .. & x_M & 0 & 0 & 0 \\ y_1 & .. & y_M & 0 & 0 & 0 \\ 1 & .. & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_M \\ a_x \\ a_y \\ a_0 \end{bmatrix} = \begin{bmatrix} h_1 \\ \vdots \\ h_M \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

This can be rewritten in the matrix form as:

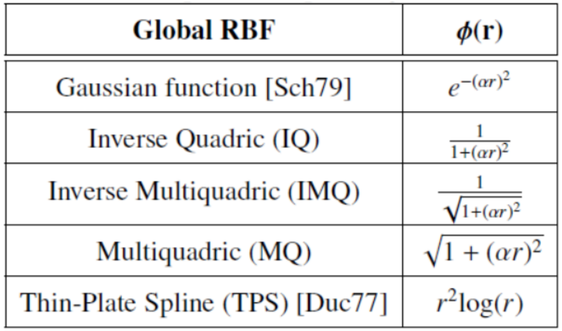
$$\begin{bmatrix} \boldsymbol{B} & \boldsymbol{P} \\ \boldsymbol{P}^T & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{a} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{0} \end{bmatrix} \qquad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \qquad \boldsymbol{a}^T \boldsymbol{x}_i = a_x\, x_i + a_y\, y_i + a_0$$

For the two-dimensional case and M points given a system of $(M + 3)$ linear equations has to be solved.

Then the extension to d-dimensional case is straightforward.

RBF function selection

The RBF interpolation was originally introduced by multiquadric method in 1971, which was called Radial Basis Function (RBF) method. Since then many different RFB interpolation schemes have been developed with some specific properties, e.g. uses $\varphi(r) = r^2 lg\, r$, which is called Thin-Plate Spline (TPS), a function $\varphi(r) = e^{-(\epsilon r)^2}$, was proposed.



Global RBF	$\boldsymbol{\phi(r)}$
Gaussian function [Sch79]	$e^{-(\alpha r)^2}$
Inverse Quadric (IQ)	$\frac{1}{1+(\alpha r)^2}$
Inverse Multiquadric (IMQ)	$\frac{1}{\sqrt{1+(\alpha r)^2}}$
Multiquadric (MQ)	$\sqrt{1+(\alpha r)^2}$
Thin-Plate Spline (TPS) [Duc77]	$r^2\log(r)$

- If “global” functions, e.g. TPS ($\varphi(r) = r^2 lg\, r$, are used, then the matrix $\boldsymbol{B}$ is “full”,
- If “local” functions (Compactly Supported RBF – CSRBF) are used, the matrix $\boldsymbol{B}$ can be sparse.

The **Chyba! Nenalezen zdroj odkazů.**Compactly Supported Radial Basis Functions (CSRBFs) were introduced as:

$$\varphi(r) = \begin{cases} (1-r)^q \, P(r), & 0 \le r \le 1 \\ 0, & r > 1 \end{cases} ,$$

where: $P(r)$ is a polynomial function and q is a parameter.

- In the case of global functions, the linear system of equations is becoming ill conditioned and problems with convergence can be expected.

On the other hand

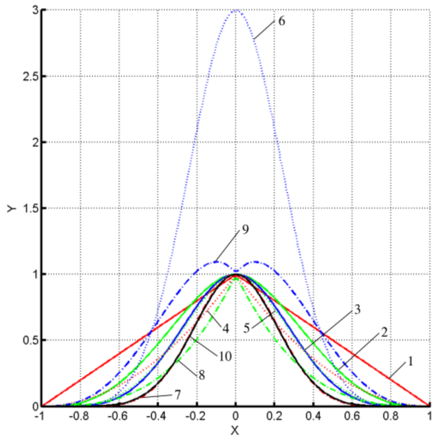
- If the CSRBFs are taken, the matrix $\boldsymbol{A}$ is becoming relatively sparse, i.e. computation of the LSE will be faster, but we need to carefully select the scaling factor α (which can be “tricky”) and the final function might tend to be “blobby” shaped.

However, the matrix $\boldsymbol{A}$ is still very large which causes numerical and computational robustness problems.

Examples of Compactly Supported Radial Basis Functions

Typical examples of "local" functions - CSRBF

ID	Function	ID	Function
1	$(1-r)_+$	6	$(1-r)_+^6(35r^2+18r+3)$
2	$(1-r)_+^3(3r+1)$	7	$(1-r)_+^8(32r^3+25r^2+8r+3)$
3	$(1-r)_+^5(8r^2+5r+1)$	8	$(1-r)_+^3$
4	$(1-r)_+^2$	9	$(1-r)_+^3(5r+1)$
5	$(1-r)_+^4(4r+1)$	10	$(1-r)_+^7(16r^2+7r+1)$



Geometrical properties of CSRBF

The compactly supported RBFs are defined for the interval $r \in \langle 0\ ,\ 1\ \rangle$, but for the practical use a scaling is used, i.e. the value r is multiplied by a scaling factor α, where $\alpha > 0$.

Meshless techniques are primarily based on approaches mentioned above. They are used in engineering problem solutions, nowadays, e.g. partial differential equations, surface modeling, surface reconstruction of scanned objects, reconstruction of corrupted images, etc. More generally, meshless object representation is based on specific interpolation or approximation techniques.

The resulting matrix $\boldsymbol{A}$ tends to be large and ill-conditioned. Therefore, some specific numerical methods have to be taken to increase robustness of a solution, like preconditioning methods or parallel computing on GPU, etc. In addition, subdivision or hierarchical methods are used to decrease sizes of computations and increase robustness.

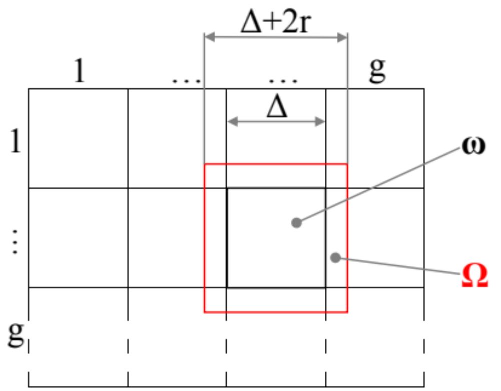
Computational complexity of meshless methods actually covers complexity of tessellation itself and interpolation and approximation methods. This results into problems with large data set processing, i.e. numerical stability and memory requirements, etc.

If global RBF functions are considered, the RBF matrix is full and in the case of 10^6 of points, the RBF matrix is of the size approx.$10^6 \times 10^6$! On the other hand, if CSRBF used, the relevant matrix is sparse and computational and memory requirements can be decreased significantly using special data structures.

On the other hand, in the case of physical phenomena visualization, data received by simulation, computation or obtained by experiments usually are oversampled in some areas and also numerically more or less precise. It seems possible to apply approximation methods to decrease computational complexity significantly by adding virtual points in the place of interest and use analogy of the least square method modified for the RBF case.

Due to CSRBF representation the space of data can be subdivided, interpolation, resp. approximation can be split to independent parts and computed more or less independently. This process can be also parallelized and if appropriate computational architecture is used, e.g. GPU etc. It will lead to faster computation as well. This approach was experimentally verified for scalar and vector data used in visualization of physical phenomena.

A possible solution is application of the space subdivision using cells with some overlap with blending on the cell border

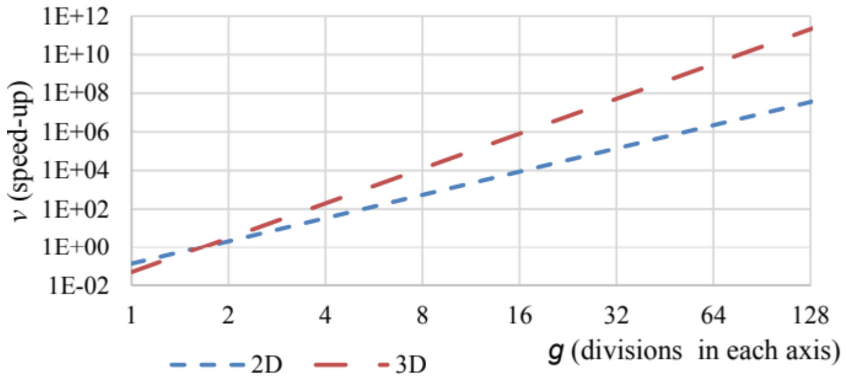


Advantages:

- Significantly decreases the size of the matrix $\boldsymbol{A}$
- Simplifies the function computation itself as it has significantly less elements in the sum

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

- Speed up is significant;
  speed up is in LOG scale !!
  (over 6 10^6 points)

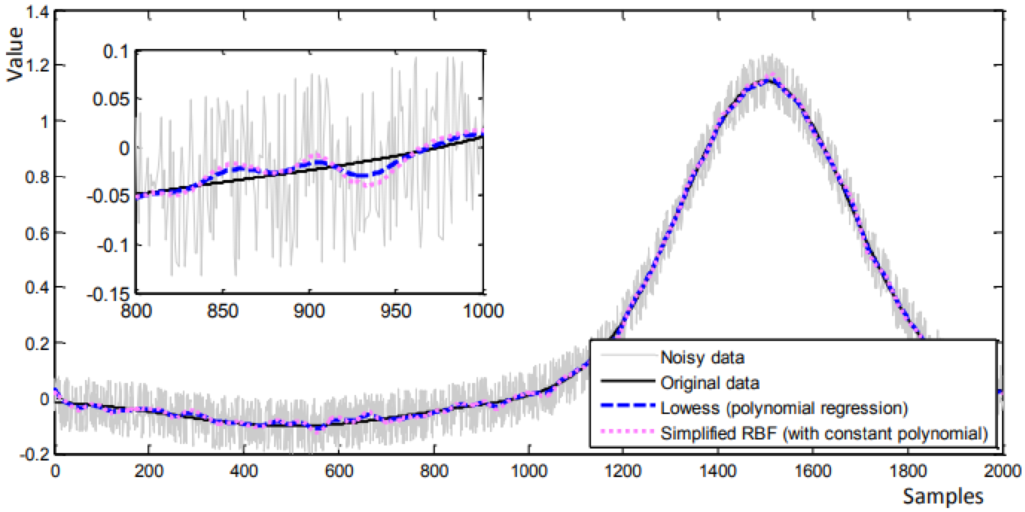


Interpolation x approximation?

The question:

How the data can be reduced significantly but still having a good precision?

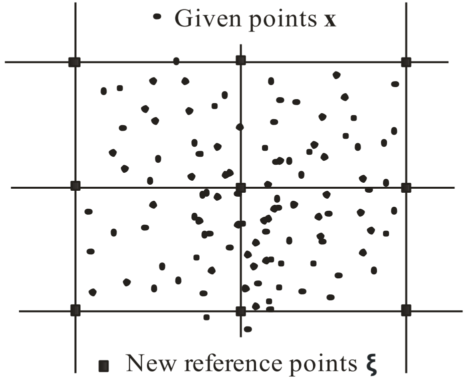
This leads to a question, how the radial basis functions can be used for approximation



Meshless approximation

The RBF interpolation relies on solution of a LSE $\boldsymbol{Ax} = \boldsymbol{b}$ of the size $M \times M$ in principle, where M is a number of the data to be processed. If “global” functions are used, the matrix $\boldsymbol{A}$ is full, while if “local” functions are used (CSRBF), the matrix $\boldsymbol{A}$ is sparse.

However, in visualization applications, it is necessary to compute the final function $f(\boldsymbol{x})$ many many times and even for already computed λ_i values, the computation of $f(\boldsymbol{x})$ is too expensive. Therefore, it is reasonable to significantly “reduce” the dimensionality of the relevant LSE $\boldsymbol{Ax} = \boldsymbol{b}$. Of course, we are now changing the interpolation property of the RBF to RBF approximation, i.e. the values computed do not pass the given values exactly.


• Given points $\mathbf{x}$
■ New reference points $\boldsymbol{\xi}$

RBF approximation and points’ reduction

Simple approach

Probably the best way is to formulate the problem using the Least Square Error approximation. Let us consider the modified formulation of the RBF interpolation

$$f(\boldsymbol{x}_i) = \sum_{j=1}^{M} \lambda_j \, \varphi\left(\left\|\boldsymbol{x}_i - \boldsymbol{\xi}_j\right\|\right) \qquad\qquad h_i = f(\boldsymbol{x}_i) \qquad i = 1, \dots, N$$

where: $\boldsymbol{\xi}_j$ are not given points, but points in a pre-defined "virtual mesh" (in positions of area of interest etc.) as only coordinates are needed (there is no tessellation needed). This "virtual mesh" can be irregular, regular or adaptive etc. For a simplicity (just for explanation purposes), let us consider a two-dimensional squared (orthogonal) mesh in the following example. Then the $\boldsymbol{\xi}_j$ coordinates are the corners of this mesh. It means that the given scattered data will be actually "re-sampled", e.g. to the squared mesh.

In many applications the given data sets are heavily over sampled. For fast previews, we can afford to "down sample" the given data set, e.g. for data visualization, WEB applications

Let us consider that for the visualization purposes we want to represent the final potential field by P values instead of M and $P \ll M$. The reason is very simple as if we need to compute the function $f(\boldsymbol{x})$ in many points, the formula above needs to be evaluated many times. We can expect that the number of evaluation Q can be easily requested at $10^2\ M$ of points (new points) used for visualization.

If we consider that $Q \geq 10^2\, M$ and $M \geq 10^2\, P$ then

the speed up factor in evaluation can be easily about $\mathbf{10^4}$!

The formulation above leads to a solution of an over determined system of linear equations $\boldsymbol{Ax} = \boldsymbol{b}$ where number of rows $M \gg P$, number of unknown $\boldsymbol{\lambda} = [\lambda_1, \dots, \lambda_P]^T$. The linear system of equations $\boldsymbol{Ax} = \boldsymbol{b}$. It can be solved by the Least Square Method (LSM) as $\boldsymbol{A}^T\boldsymbol{Ax} = \boldsymbol{A}^T\boldsymbol{b}$ or by factorization using Gram-Schmidt orthogonalization or Householder transformation etc.

$$\begin{bmatrix} \varphi_{1,1} & \cdots & \varphi_{1,P} \\ \vdots & \ddots & \vdots \\ \varphi_{i,1} & \cdot\cdot & \varphi_{i,P} \\ \vdots & \ddots & \vdots \\ \varphi_{M,1} & \cdots & \varphi_{M,P} \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_P \end{bmatrix} = \begin{bmatrix} h_1 \\ \vdots \\ \vdots \\ \vdots \\ h_M \end{bmatrix} \qquad \boldsymbol{Ax} = \boldsymbol{b}$$

When the system of LSE is solved, computation of a function value $f(x)$ will be sped-up by a factor

$$\nu = {}^{M}/_{P}.$$

RBF with Lagrange Multipliers

Let us assume again:

$$f(\boldsymbol{x}_i) = \sum_{j=1}^{M} \lambda_j \, \varphi(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|) \qquad i = 1, \dots, N \qquad \boldsymbol{A\lambda} = \boldsymbol{f}$$

where $M \leq N$. We want to determine $\boldsymbol{\lambda} = [\lambda_1, \dots, \lambda_M]^T$ minimizing a quadratic form $\frac{1}{2}\boldsymbol{\lambda}^T\boldsymbol{Q\lambda}$ with a linear constrains $-\boldsymbol{f} = \boldsymbol{0}$, where $\boldsymbol{Q}$ is a positive symmetric matrix. This can be solved using Lagrange multipliers $\boldsymbol{\xi} = [\xi_1, \dots, \xi_N]^T$, i.e. minimizing the expression:

$$\frac{1}{2}\boldsymbol{\lambda}^T\boldsymbol{Q\lambda} - \boldsymbol{\xi}^{\boldsymbol{T}}(\boldsymbol{A\lambda} - \boldsymbol{f}) \qquad \text{i.e } \boldsymbol{\lambda} = ? \text{ and } \boldsymbol{\xi} = ?$$

As the matrix $\boldsymbol{Q}$ is positive and symmetric, we obtain

$$\frac{\partial}{\partial\boldsymbol{\lambda}}\left(\frac{1}{2}\boldsymbol{\lambda}^T\boldsymbol{Q\lambda} - \boldsymbol{\xi}^{\boldsymbol{T}}(\boldsymbol{A\lambda} - \boldsymbol{f})\right) = \boldsymbol{Q\lambda} - \boldsymbol{A}^T\boldsymbol{\xi} = \boldsymbol{0} \qquad \frac{\partial}{\partial\boldsymbol{\xi}}\left(\frac{1}{2}\boldsymbol{\lambda}^T\boldsymbol{Q\lambda} - \boldsymbol{\xi}^{\boldsymbol{T}}(\boldsymbol{A\lambda} - \boldsymbol{f})\right) = \boldsymbol{A}^T\boldsymbol{\lambda} - \boldsymbol{f} = \boldsymbol{0}$$

In more compact matrix form we can write

$$\begin{bmatrix} \boldsymbol{Q} & -\boldsymbol{A}^T \\ \boldsymbol{A} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\xi} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{f} \end{bmatrix}$$

As the matrix $\boldsymbol{Q}$ is positive definite, block in matrix operations can be applied and we get:

$$\boldsymbol{\lambda} = \boldsymbol{Q}^{-1}\boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{Q}^{-1}\boldsymbol{A}^T)^{-1}\boldsymbol{f} \qquad \boldsymbol{\xi} = (\boldsymbol{A}\boldsymbol{Q}^{-1}\boldsymbol{A}^T)^{-1}\boldsymbol{f}$$

If $\boldsymbol{A} = \boldsymbol{A}^T$ and invertible, computation can be further simplified. This approach is more robust, however also more computationally expensive.

It should be noted, that if the Least Square Method (LSM) is used directly, i.e. $\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^T\boldsymbol{b}$ is to be solved directly, the $\boldsymbol{A}^T\boldsymbol{A}$ matrix is ill conditioned and for large M the system of linear equations is difficult to solve. In addition, selection of the $\boldsymbol{Q}$ matrix elements is not fully determined and depends on a user, actually. The advantage of this approach is that values of the matrix $\boldsymbol{A}$ have only a linear influence. It should be noted that the matrix size is $2Mx2M$, where M is a number of points. It means that the memory requirements are no acceptable even for medium data sets. Also a cost of the value computation, i.e. computation of a value $f(\boldsymbol{x})$ for the given $\boldsymbol{x}$ is doubled. For real applications of the RBF approximation, we need to decrease memory requirements significantly.

Least Square Method with a Polynomial Reproduction

Let us consider again the overdetermined system:

$$f(\boldsymbol{x}_i) = \sum_{j=1}^{M} \lambda_j\, \varphi\left(\left\|\boldsymbol{x}_i - \boldsymbol{\xi}_j\right\|\right) + \boldsymbol{a}^T \boldsymbol{x}_i = \sum_{j=1}^{M} \lambda_j\, \varphi_{i,j} + \boldsymbol{a}^T \boldsymbol{x}_{\boldsymbol{i}}$$

It can be rewritten in the matrix form as

$$\boldsymbol{A\lambda} + \boldsymbol{Pa} = \boldsymbol{f}$$

Now, we can define an error r of a solution as

$$r^2 = \|\boldsymbol{A\lambda} + \boldsymbol{Pa} - \boldsymbol{f}\|^2 = (\boldsymbol{A\lambda} + \boldsymbol{Pa} - \boldsymbol{f})^T(\boldsymbol{A\lambda} + \boldsymbol{Pa} - \boldsymbol{f})$$

where:

$$\boldsymbol{Pa} = \begin{bmatrix} x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots \\ x_m & y_m & 1 \end{bmatrix} \begin{bmatrix} a_x \\ a_y \\ a_0 \end{bmatrix}$$

To minimize the error r the following conditions must be valid:

$$\frac{\partial r^2}{\partial \boldsymbol{\lambda}} = \boldsymbol{A}^T\boldsymbol{A\lambda} + \boldsymbol{A}^T\boldsymbol{Pa} - \boldsymbol{A}^T\boldsymbol{f} = \boldsymbol{0} \qquad \frac{\partial r^2}{\partial \boldsymbol{a}} = \boldsymbol{P}^T\boldsymbol{A\lambda} + \boldsymbol{P}^T\boldsymbol{Pa} - \boldsymbol{P}^T\boldsymbol{f} = \boldsymbol{0}$$

or in a matrix form as $\boldsymbol{Mx} = \boldsymbol{y}$, i.e.

$$\begin{bmatrix} \boldsymbol{A}^T\boldsymbol{A} & \boldsymbol{A}^T\boldsymbol{P} \\ \boldsymbol{P}^T\boldsymbol{A} & \boldsymbol{P}^T\boldsymbol{P} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{a} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}^T\boldsymbol{f} \\ \boldsymbol{P}^T\boldsymbol{f} \end{bmatrix}$$

The above presented formula leads to correct results. However, it can be seen, that the values m_{ij} of the matrix $\boldsymbol{M}$ are influenced by:

- elements of the matrix $\boldsymbol{A}^T\boldsymbol{A}$, i.e. by used radial basis function and mutual positions of the given points
- elements of the matrix $\boldsymbol{P}^T\boldsymbol{P}$, i.e. by coordinates of the given points.

This is a significant problem if large data sets are to be processed and the interval of $\boldsymbol{x}$ values, i.e. x, y is high as the value is squared due to $\boldsymbol{P}^T\boldsymbol{P}$ submatrix etc.

Let us analyze this property more in detail, now, in order to be able to estimate problems in real application use.

Decomposition of RBF Interpolation

The RBF interpolation can be described in the matrix form as

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{P} \\ \boldsymbol{P}^T & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{a} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{0} \end{bmatrix} \qquad \boldsymbol{a}^T \boldsymbol{x}_i = a_x x_i + a_y y_i + a_0$$

where $\boldsymbol{x} = [x, y, 1]^T$, the matrix $\boldsymbol{A}$ is symmetrical and semidefinite positive (or strictly positive) definite. Let us consider the Schur's complement (validity of all operation is expected)

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{C}\boldsymbol{A}^{-1} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{A} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M}/\boldsymbol{A} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{A}^{-1}\boldsymbol{B} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \qquad \boldsymbol{M}/\boldsymbol{A} \overset{\text{def}}{=} \boldsymbol{D} - \boldsymbol{C}\boldsymbol{A}^{-1}\boldsymbol{B}$$

where $\boldsymbol{M}/\boldsymbol{A}$ is the Schur complement.
Then the inversion matrix $\boldsymbol{M}^{-1}$

$$\boldsymbol{M}^{-1} = \begin{bmatrix} \boldsymbol{I} & -\boldsymbol{A}^{-1}\boldsymbol{B} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{A}^{-1} & \boldsymbol{0} \\ \boldsymbol{0} & (\boldsymbol{M}/\boldsymbol{A})^{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ -\boldsymbol{C}\boldsymbol{A}^{-1} & \boldsymbol{I} \end{bmatrix}$$

Now, the Schur complement can be applied to the RBF interpolation. As the matrix $\boldsymbol{M}$ is nonsingular, inversion of the matrix $\boldsymbol{M}$ can be used. Using the Schur complement (as the matrix $\boldsymbol{D} = \boldsymbol{0}$)

$$\boldsymbol{M}^{-1} = \begin{bmatrix} \boldsymbol{I} & -\boldsymbol{A}^{-1}\boldsymbol{P} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{A}^{-1} & \boldsymbol{0} \\ \boldsymbol{0} & (\boldsymbol{M}/\boldsymbol{A})^{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ -\boldsymbol{P}^T\boldsymbol{A}^{-1} & \boldsymbol{I} \end{bmatrix} \qquad \boldsymbol{M}/\boldsymbol{A} \overset{\text{def}}{=} \boldsymbol{P}^T\boldsymbol{A}^{-1}\boldsymbol{P}$$

Then $\det(\boldsymbol{M}) \neq 0$, $\det(\boldsymbol{M}/\boldsymbol{A}) \neq 0$ and $\det(\boldsymbol{M}/\boldsymbol{A}) \neq 0$ as the matrices are nonsingular.

However, if RBF interpolation is used for larger data sets, there is a severe problem with robustness and numerical stability, i.e. numerical computability issues. Using the Schur's complement we can see, that

$$\det(\boldsymbol{M}) = \det(\boldsymbol{A})\ \det(\boldsymbol{M}/\boldsymbol{A})$$

therefore

$$\det(\boldsymbol{M}^{-1}) = \frac{1}{\det(\boldsymbol{A})}\,\frac{1}{\det(\boldsymbol{M}/\boldsymbol{A})} = \frac{1}{\det(\boldsymbol{A})}\,\frac{1}{\det(\boldsymbol{P}^T\boldsymbol{A}^{-1}\boldsymbol{P})}$$

Properties of the matrix $\boldsymbol{A}$ are determined by the RFB function used. The value of $\det(\boldsymbol{A})$ depends also on the mutual distribution of points. However, the influence of $\det(\boldsymbol{P}^T\boldsymbol{A}^{-1}\boldsymbol{P})$ is also significant as the value depends on the points mutual distribution due to the matrix $\boldsymbol{A}$ but also to points distribution in space, due to the matrix $\boldsymbol{P}$. It means that translation of points in space does have significant influence. Let us imagine for a simplicity that the matrix $\boldsymbol{A} = \boldsymbol{I}$ (it can happen if CSRBF is used and only one point is within the radius $r = 1$). Then the distance of a point from the origin has actually quadratic influence as the point position is in the matrices $\boldsymbol{P}^T$ and $\boldsymbol{P}$.

There is a direct significant consequence for the RBF interpolation.

$$f(\boldsymbol{x}) = \sum_{j=1}^{M} \lambda_j\, \varphi\left(\left\|x - \boldsymbol{x}_j\right\|\right) + P_k(\boldsymbol{x})$$

when the $P_k(\boldsymbol{x})$, $k = 1, 2$ is a quadratic polynomial in the form

$$P_1(\boldsymbol{x}) = a_0 + a_1 x + a_2 y \qquad \text{, resp.} \qquad P_2(\boldsymbol{x}) = a_0 + a_1 x + a_2 y + a_3 x^2 + a_4 xy + a_5 y^2$$

In the case of $\boldsymbol{A} = \boldsymbol{I}$, we get a matrix $\boldsymbol{P}^T\boldsymbol{P}$ of the size (3×3) and $\det(\boldsymbol{P}^T\boldsymbol{P})$ in the case of a linear polynomial $P_1(\boldsymbol{x})$ as:

$$\det(\boldsymbol{P}^T\boldsymbol{P}) = \begin{vmatrix} \sum_{i=1}^{M} x_i^2 & \sum_{i=1}^{M} x_i y_i & \sum_{i=1}^{M} x_i \\ \sum_{i=1}^{M} x_i y_i & \sum_{i=1}^{M} y_i^2 & \sum_{i=1}^{M} y_i \\ \sum_{i=1}^{M} x_i & \sum_{i=1}^{M} y_i & \sum_{i=1}^{M} 1 \end{vmatrix} = 1\left(\sum x_i^2 \sum y_i^2\right) - \sum y_i\,(\dots) + \sum y_i\,(\dots)$$

It means that points distribution in space and their distances from the origin play a significant role as the $\det(\boldsymbol{P}^T\boldsymbol{P})$ contains elements $\sum_{i=1}^{M} x_i^2$ and $\sum_{i=1}^{M} y_i^2$ in multiplicative etc. in the linear polynomial case. If a quadratic polynomial $P_2(\boldsymbol{x})$ is used, the matrix $\boldsymbol{P}^T\boldsymbol{P}$ is of the size (6×6):

$$\boldsymbol{P}^T\boldsymbol{P} = \begin{bmatrix} x_1^2 & \cdots & x_M^2 \\ y_1^2 & \cdots & y_1^2 \\ x_1 y_1 & \cdots & x_M y_M \\ x_1 & \cdots & x_M \\ y_1 & \cdots & y_M \\ 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} x_1^2 & y_1^2 & x_1 y_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_M^2 & y_M^2 & x_M y_M & x_M & y_M & 1 \end{bmatrix}$$

Then

$$\det(\boldsymbol{P}^T\boldsymbol{P}) = \det\begin{bmatrix} \sum_{i=1}^{M} x_i^4 & \sum_{i=1}^{M} x_i^2 y_i^2 & \sum_{i=1}^{M} x_i^3 y_i & \sum_{i=1}^{M} x_i^3 & \sum_{i=1}^{M} x_i^2 y_i & \sum_{i=1}^{M} x_i^2 \\ \sum_{i=1}^{M} x_i^2 y_i^2 & \sum_{i=1}^{M} y_i^4 & \sum_{i=1}^{M} x_i y_i^3 & \sum_{i=1}^{M} x_i y_i^2 & \sum_{i=1}^{M} y_i^3 & \sum_{i=1}^{M} y_i^2 \\ \sum_{i=1}^{M} x_i^3 y_i & \sum_{i=1}^{M} x_i y_i^3 & \sum_{i=1}^{M} x_i^2 y_i^2 & \sum_{i=1}^{M} x_i^2 y_i & \sum_{i=1}^{M} x_i y_i^2 & \sum_{i=1}^{M} x_i y_i \\ \sum_{i=1}^{M} x_i^3 & \sum_{i=1}^{M} x_i y_i^2 & \sum_{i=1}^{M} x_i^2 y_i & \sum_{i=1}^{M} x_i^2 & \sum_{i=1}^{M} x_i y_i & \sum_{i=1}^{M} x_i \\ \sum_{i=1}^{M} x_i^2 y_i & \sum_{i=1}^{M} y_i^3 & \sum_{i=1}^{M} x_i y_i^2 & \sum_{i=1}^{M} x_i y_i & \sum_{i=1}^{M} y_i^2 & \sum_{i=1}^{M} y_i \\ \sum_{i=1}^{M} x_i^2 & \sum_{i=1}^{M} y_i^2 & \sum_{i=1}^{M} x_i y_i & \sum_{i=1}^{M} x_i & \sum_{i=1}^{M} y_i & \sum_{i=1}^{M} 1 \end{bmatrix}$$

In the quadratic polynomial case, the $\det(\boldsymbol{P}^T\boldsymbol{P})$ contains elements $\sum_{i=1}^{M} x_i^4$, $\sum_{i=1}^{M} y_i^2$,…, $\sum_{i=1}^{M} 1$ in multiplicative, which brings even worst situation as the matrix $\boldsymbol{P}^T\boldsymbol{P}$ contains small and very high values. As a direct consequence, eigenvalues will have large span and therefore the linear system of equations will become ill-conditioned.

Decomposition of RBF approximation

Decomposition for RBF approximation is analogous to interpolation decomposition. Let us explore decomposition of the RBF approximation using the Schur complement. Let us consider the system of linear equation for the RBF approximation in the form $\boldsymbol{Mx} = \boldsymbol{y}$:

$$\begin{bmatrix} \boldsymbol{A}^T\boldsymbol{A} & \boldsymbol{A}^T\boldsymbol{P} \\ \boldsymbol{P}^T\boldsymbol{A} & \boldsymbol{P}^T\boldsymbol{P} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{a} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}^T\boldsymbol{f} \\ \boldsymbol{P}^T\boldsymbol{f} \end{bmatrix}$$

Let us consider again the Schur's complement (validity of operations is expected and the matrix $\boldsymbol{D} \neq \boldsymbol{0}$)

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{CA}^{-1} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{A} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M}/\boldsymbol{A} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{A}^{-1}\boldsymbol{B} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \qquad \boldsymbol{M}/\boldsymbol{A} \stackrel{\text{def}}{=} \boldsymbol{D} - \boldsymbol{CA}^{-1}\boldsymbol{B}$$

In this case, for the RBF approximation we obtain:

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{A}^T\boldsymbol{A} & \boldsymbol{A}^T\boldsymbol{P} \\ \boldsymbol{P}^T\boldsymbol{A} & \boldsymbol{P}^T\boldsymbol{P} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ (\boldsymbol{P}^T\boldsymbol{A})(\boldsymbol{A}^T\boldsymbol{A})^{-1} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{A}^T\boldsymbol{A} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M}/(\boldsymbol{A}^T\boldsymbol{A}) \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & (\boldsymbol{A}^T\boldsymbol{A})^{-1}(\boldsymbol{A}^T\boldsymbol{P}) \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}$$

Then the matrix $\boldsymbol{M}^{-1}$ using the Schur complement:

$$\boldsymbol{M}^{-1} = \begin{bmatrix} \boldsymbol{I} & -(\boldsymbol{A}^T\boldsymbol{A})^{-1}(\boldsymbol{A}^T\boldsymbol{P}) \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} (\boldsymbol{A}^T\boldsymbol{A})^{-1} & \boldsymbol{0} \\ \boldsymbol{0} & \left(\boldsymbol{M}/(\boldsymbol{A}^T\boldsymbol{A})\right)^{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ -(\boldsymbol{P}^T\boldsymbol{A}^{-1})(\boldsymbol{A}^T\boldsymbol{A})^{-1} & \boldsymbol{I} \end{bmatrix}$$

where

$$\boldsymbol{M}/(\boldsymbol{A}^T\boldsymbol{A}) = \boldsymbol{P}^T\boldsymbol{P} - (\boldsymbol{P}^T\boldsymbol{A})(\boldsymbol{A}^T\boldsymbol{A})^{-1}(\boldsymbol{A}^T\boldsymbol{P})$$

In the RBF approximation case, the matrix $\boldsymbol{D}$ is non-zero matrix $\boldsymbol{P}^T\boldsymbol{P}$. It can be seen, that if the matrix $\boldsymbol{A}^T\boldsymbol{A} \rightarrow \boldsymbol{I}$, then the Schur's complement $\boldsymbol{M}/(\boldsymbol{A}^T\boldsymbol{A}) \rightarrow \left(\boldsymbol{P}^T\boldsymbol{P} - (\boldsymbol{P}^T\boldsymbol{A})(\boldsymbol{A}^T\boldsymbol{P})\right) = \boldsymbol{P}^T\boldsymbol{P} - (\boldsymbol{A}^T\boldsymbol{P})^T(\boldsymbol{A}^T\boldsymbol{P})$. It means, that the whole matrix $\boldsymbol{M}$ tends to be singular. It can be seen that the $\det(\boldsymbol{P}^T\boldsymbol{P})$ contains elements $\sum_{i=1}^{M} x_i^4$, $\sum_{i=1}^{M} y_i^2$,…, $\sum_{i=1}^{M} 1$ in multiplicative.

This has a significant influence to the robustness of computation if small and high values of x_i and y_i occur in the data sets, or if they are from large interval span.

If the values $(x_i, y_i) \in \langle -10^5, 10^5 \rangle \times \langle -10^5, 10^5 \rangle$, the value of $\det(\boldsymbol{P}^T\boldsymbol{P}) > \sum_{i=1}^{M} x_i^2 \sum_{i=1}^{M} y_i^2 > 10^{20}$ and the value of $1/\det(\boldsymbol{P}^T\boldsymbol{P}) < 10^{-20}$, in the case of the linear polynomial. It results to a situation when the matrix $\boldsymbol{M}^{-1}$ will be "close" to singular.

In the case of quadratic polynomial $P_2(\boldsymbol{x})$, the situation gets even worst as $\det(\boldsymbol{P}^T\boldsymbol{P})$ contains elements $\sum x_i^4$ and $\sum y_i^4$ in multiplicative, i.e. $\det(\boldsymbol{P}^T\boldsymbol{P}) > \sum_{i=1}^{M} x_i^4 \sum_{i=1}^{M} y_i^4 > 10^{40}$. This should be considered as a significant disadvantage of the RBF approximation used for large data spans.

Conclusion

The RBF interpolation using compactly supported RBF (CSRBF) have several significant advantages over methods based on smooth interpolation made on triangulated space area. In this contribution some properties of the CSRBF interpolation and approximation methods have been presented from the “engineering” point of view and selected features related to robustness and stability of computation have been presented.

References

Relevant papers related to meshless with the PDFs can be found at

- http://afrodita.zcu.cz/~skala/Publication-RBF.htm
- http://afrodita.zcu.cz/~skala/publications.htm (all papers in general)

The latest results in the field of meshless methods are to be available at the
MESHFREE 2020 conference track of ICCS 2020, Amsterdam

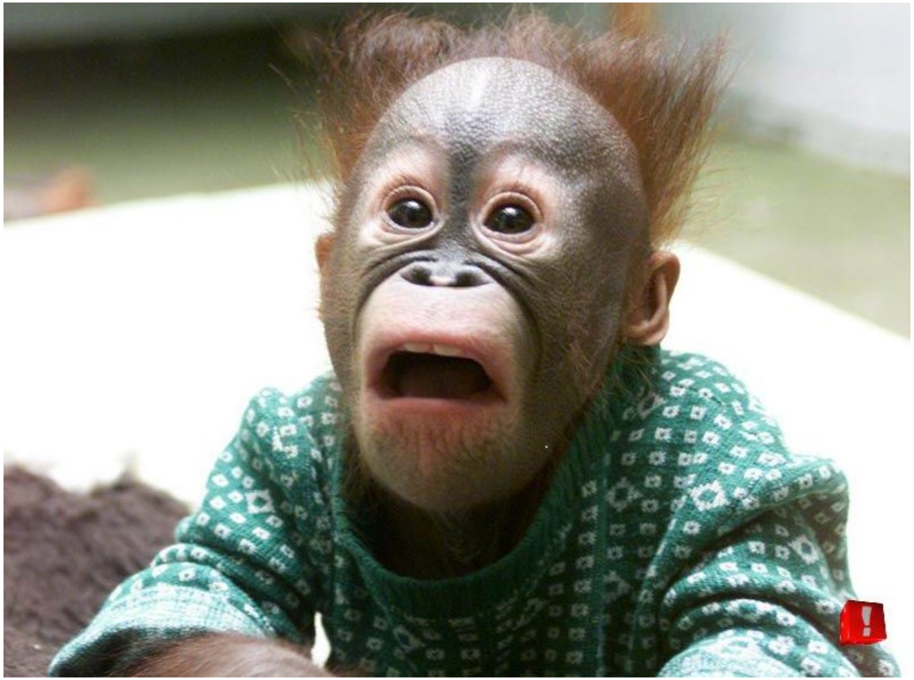
- http://meshfree.zcu.cz/ICCS2020/

Acknowledgments

Thanks belong to my collaborators and PhD students at the University of West Bohemia, especially to Zuzana Majdisova, Michal Smolik and Karel Uhlir for implementation of some algorithms and graph/images generation.
Some images were taken from recent common publications and from publicly available resources.

Research supported by the Czech Science Foundation GACR, No. GA17-05534S.

Questions



??????