Radial Basis Functions Interpolation

José Vicente Lorenzo Gómez *

June 12, 2018

Radial Basis Functions can be used to interpolate scattered data in a d-dimensional space. The main idea is to construct an interpolation expression $F : \mathbb{R}^d \to \mathbb{R}$ as a weighted sum of radial functions $\varphi : \mathbb{R} \to \mathbb{R}$ (see Tables 1 and 2),

$$F\left(\mathbf{x}\right) = \sum_{j=1}^{N} \gamma_j \varphi\left(\|\mathbf{x} - \mathbf{x}_{s,j}\|_{\mathbb{R}^d}\right) + p\left(\mathbf{x}\right), \tag{1}$$

where $p\left(x\right)$ is a polynomial term whose coefficients are unknown, and $\mathbf{x}_{s,j}$, with $1 \leq j \leq N$, are the source points where the function value is known. The weights γ_j and the coefficients in the polynomial p are obtained by imposing the known values at the source points,

$$F\left(\mathbf{x}_{s,i}\right) = f_{s,i}, \tag{2}$$

and the orthogonality conditions for every polynomial q with degree less or equal than that of p,

$$\sum_{j=1}^{N} \gamma_j q\left(\mathbf{x}_{s,j}\right) = 0. \tag{3}$$

The minimal degree of polynomials p depends on the choice of the radial functions. A unique interpolant is given if the basis function is a conditionally positive definite function.

Definition 1. [1] An even function $\phi : \mathbb{R}^d \to \mathbb{R}$ is said to be conditionally positive definite of order m if for all $N \in \mathbb{N}$, all sets of pairwise distinct centers $X = \{\mathbf{x}_1, ..., \mathbf{x}_N\} \subseteq \mathbb{R}^d$ and all $\gamma \in \mathbb{R}^N \setminus \{0\}$ that satisfy (3) for all polynomials q of degree less than m, the quadratic form

$$\sum_{i=1}^{N} \sum_{j=1}^{N} \gamma_i \gamma_j \phi\left(\|\mathbf{x}_j - \mathbf{x}_i\|_{\mathbb{R}^d}\right) \tag{4}$$

is positive.

Theorem 2. *[1, 2] Suppose ϕ is conditionally positive definite of order m. Suppose further that the set of centers $X = \{\mathbf{x}_1, ..., \mathbf{x}_N\} \subseteq \mathbb{R}^d$ has the property that the zero polynomial is the only polynomial of degree less than m that vanishes on it completely. Then there exists exactly one function F of the form* (1) *that satisfies both* (2) *and* (3)*.*

If the basis functions are conditionally positive definite of order $m \leq 2$, a linear polynomial can be used [3].

*Universidad Autónoma de Madrid.

For the sake of clarity, we will consider a linear polynomial and known values $f_{s,i}$ at the source nodes in a 2-dimensional domain. Thus,

$$f_{s,i} = \sum_{j=1}^{N} \gamma_j \varphi \left(\|\mathbf{x}_{s,i} - \mathbf{x}_{s,j}\|_{\mathbb{R}^d} \right) + \beta_x x_{s,i} + \beta_y y_{s,i} + \beta_0, \quad 1 \leq i \leq N. \tag{5}$$

The system is closed with the orthogonality conditions in (3) that provide $d+1$ additional equations, as the following relations must hold,

$$\begin{cases} \sum_{j=1}^{N} \gamma_j = 0, \\ \sum_{j=1}^{N} \gamma_j x_{s,j} = 0, \\ \sum_{j=1}^{N} \gamma_j y_{s,j} = 0. \end{cases} \tag{6}$$

The above system can be expressed in matrix form by introducing

$$\left[M_{ij} \right]_{N \times N} = \varphi \left(\|\mathbf{x}_{s,i} - \mathbf{x}_{s,j}\|_{\mathbb{R}^d} \right), \tag{7}$$

and

$$\mathbf{P}_s = \begin{pmatrix} 1 & x_{s,1} & y_{s,1} \\ \vdots & \vdots & \vdots \\ 1 & x_{s,i} & y_{s,i} \\ \vdots & \vdots & \vdots \\ 1 & x_{s,N} & y_{s,N} \end{pmatrix}, \quad \gamma = \begin{pmatrix} \gamma_1 \\ \vdots \\ \gamma_i \\ \vdots \\ \gamma_N \end{pmatrix}, \quad \mathbf{f}_s = \begin{pmatrix} f_{s,1} \\ \vdots \\ f_{s,i} \\ \vdots \\ f_{s,N} \end{pmatrix}, \quad \beta = \begin{pmatrix} 1 \\ \beta_x \\ \beta_y \end{pmatrix}. \tag{8}$$

Thus,

$$\begin{pmatrix} \mathbf{M} & \mathbf{P}_s \\ \mathbf{P}_s^T & \mathbf{0} \end{pmatrix} \begin{pmatrix} \gamma \\ \beta \end{pmatrix} = \begin{pmatrix} \mathbf{f}_s \\ \mathbf{0} \end{pmatrix}. \tag{9}$$

Once the linear system in (9) is solved, the obtained parameters can be used to interpolate the known data with (1) at a set of points $\mathbf{x}_i$, $1 \leq i \leq M$. This can be easily done by building the matrices

$$\left[\hat{M}_{ij} \right]_{N \times N} = \varphi \left(\|\mathbf{x}_i - \mathbf{x}_{s,j}\|_{\mathbb{R}^d} \right), \tag{10}$$

and

$$\hat{\mathbf{P}} = \begin{pmatrix} 1 & x_1 & y_1 \\ \vdots & \vdots & \vdots \\ 1 & x_i & y_i \\ \vdots & \vdots & \vdots \\ 1 & x_M & y_M \end{pmatrix}, \tag{11}$$

and finally performing the matrix-vector product

$$\mathbf{f} = \begin{pmatrix} \hat{\mathbf{M}} & \hat{\mathbf{P}} \end{pmatrix} \begin{pmatrix} \gamma \\ \beta \end{pmatrix}. \tag{12}$$

It is easy to check that the matrix in (9) is dense unless RBF with compact support are used. This is a major drawback of the RBF interpolation technique, as building and solving the associated matrix must be done at a high computational cost.

RBF name	**Abbreviation**	$\varphi(\mathbf{r})$
Spline type	R	$(\epsilon r)^n$, n odd
Thin plate spline	TPS	$(\epsilon r)^n \log(\epsilon r)$, n even
Multiquadric biharmonics	MQB	$\sqrt{1+(\epsilon r)^2}$
Quadric biharmonics	QB	$1+(\epsilon r)$
Inverse multiquadric biharmonics	IMQB	$\dfrac{1}{\sqrt{1+(\epsilon r)^2}}$
Inverse quadric biharmonics	IQB	$\dfrac{1}{1+(\epsilon r)^2}$
Gaussian	Gauss	$e^{-(\epsilon r)^2}$

Table 1: Radial Basis Functions with global support [4, 3].

In order to work with sparse matrices, RBF with compact support can be used instead, which have the following form,

$$\varphi\left(\xi = \frac{r}{R}\right) = \begin{cases} f(\xi), & 0 \leq \xi \leq 1 \\ 0, & \xi > 1 \end{cases} . \tag{13}$$

where $f(\xi) \geq 0$ and R is the support radius. When using a support radius, only the points at a distance lower than R to the source nodes are considered in the calculation of the matrix $\mathbf{M}$, so that the lower the support radius the greater the matrix sparsity.

RBF name	$\varphi(\xi)$
CP $\mathcal{C}^0$	$(1-\xi)^2$
CP $\mathcal{C}^2$	$(1-\xi)^4(4\xi+1)$
CP $\mathcal{C}^4$	$(1-\xi)^6\left(\frac{35}{3}\xi^2+6\xi+1\right)$
CP $\mathcal{C}^6$	$(1-\xi)^8(32\xi^3+25\xi^2+8\xi+1)$
CTPS $\mathcal{C}^0$	$(1-\xi)^5$
CTPS $\mathcal{C}^1$	$1+\frac{80}{3}\xi^2-40\xi^3+15\xi^4-\frac{8}{3}\xi^5+20\xi^2\log\xi$
CTPS $\mathcal{C}^2_a$	$1-30\xi^2-10\xi^3+45\xi^4-6\xi^5-60\xi^3\log\xi$
CTPS $\mathcal{C}^2_b$	$1-20\xi^2+80\xi^3-45\xi^4-16\xi^5+60\xi^4\log\xi$

Table 2: Radial Basis Functions with compact support [2].

Matlab Code

```
function [y] = RBFinterp(xs, ys, x, RBFtype, R)

fPar = RBFparam(xs, ys, RBFtype, R);

if ~isempty(fPar)
    y = RBFeval(xs, x, fPar, RBFtype, R);
else
    y = [];
end

end
```

```
function [fPar, M] = RBFparam(xs, ys, RBFtype, R)

if size(xs, 1) == size(ys, 1)

    Ns = size(xs, 1);
    Ms = size(ys, 2);
    dim = size(xs, 2);

    ncol = 1 + dim;
    r = zeros(Ns);

    for i = 1:Ns
        for j = (i+1):Ns
            r(i, j) = norm( xs(i, :) - xs(j, :) );
            r(j, i) = r(i, j);
        end
    end
    P = [ones(Ns, 1), xs];
    M = radialFunction(r, RBFtype, R);

    A = [M, P; P', zeros(ncol)];
    b = [ys; zeros(ncol, Ms)];

    fPar = A\b;

else

    fPar = [];
    M = [];

end

end
```

```
function [y] = RBFeval(xs, x, fPar, RBFtype, R)

Ns = size(xs, 1);
dim = size(xs, 2);
```

```
if size(x, 2) == dim && size(fPar, 1) == Ns + dim + 1

    N = size(x, 1);

    r = zeros(N, Ns);
    for i = 1:N
        for j = 1:Ns
            r(i, j) = norm( x(i, :) - xs(j, :) );
        end
    end

    P = [ones(N, 1), x];
    M = radialFunction(r, RBFtype, R);

    y = [M, P]*fPar;

else

    y = [];

end

end
```

```
function [phi] = radialFunction(r, RBFtype, R)

r = r/R;

phi = zeros(size(r));

switch RBFtype
    case 'R1'
        phi = r;
    case 'R3'
        phi = r.^3;
    case 'TPS2'
        I = (r > 0);
        phi(I) = r(I).^2.*log(r(I));
    case 'Q'
        phi = 1 + r.^2;
    case 'MQ'
        phi = sqrt(1 + r.^2);
    case 'IMQ'
        phi = 1./sqrt(1 + r.^2);
    case 'IQ'
        phi = 1./(1 + r.^2);
    case 'GS'
        phi = exp(-r.^2);
    case 'CP_C0'
```

```
        I = (r < 1);
        phi(I) = (1 - r(I)).^2;
    case 'CP_C2'
        I = (r < 1);
        phi(I) = (1 - r(I)).^4.*(4*r(I) + 1);
    case 'CP_C4'
        I = (r < 1);
        phi(I) = (1 - r(I)).^6.*(35/3*r(I).^2 + 6*r(I) + 1);
    case 'CP_C6'
        I = (r < 1);
        phi(I) = (1 - r(I)).^8.*(32*r(I).^3 + 25*r(I).^2 + 8*r(I) +
           1);
    case 'CTPS_C0'
        I = (r < 1);
        phi(I) = (1 - r(I)).^5;
    case 'CTPS_C1'
        I = (r < 1 & r > 0);
        phi(I) = 1 + 80/3*r(I).^2 - 40*r(I).^3 + 15*r(I).^4 - 8/3*r(
           I).^5 + 20*r(I).^2.*log(r(I));
        phi(r == 0) = 1;
    case 'CTPS_C2a'
        I = (r < 1 & r > 0);
        phi(I) = 1 - 30*r(I).^2 - 10*r(I).^3 + 45*r(I).^4 - 6*r(I)
           .^5 - 60*r(I).^3.*log(r(I));
        phi(r == 0) = 1;
    case 'CTPS_C2b'
        I = (r < 1 & r > 0);
        phi(I) = 1 - 20*r(I).^2 + 80*r(I).^3 - 45*r(I).^4 -16*r(I)
           .^5 + 60*r(I).^4.*log(r(I));
        phi(r == 0) = 1;
    otherwise
        phi = radialFunction(r, 'R1', R);
end

end
```

References

[1] Beckert, Armin and Wendland, Holger. Multivariate interpolation for fluid-structure-interaction problems using radial basis functions. *Aerospace Science and Technology*, 5 (2), p. 125-134, 2001.

[2] Wendland, Holger. *Konstruktion und Untersuchung radialer Basisfunktionen mit kompaktem Träger*. PhD thesis, Göttingen, Georg-August-Universität zu Göttingen, Diss, 1996.

[3] De Boer, A and Van der Schoot, MS and Bijl, Hester. Mesh deformation based on radial basis function interpolation. *Computers & structures*, 85 (11-14), p. 784-795, 2007.

[4] Biancolini, Marco Evangelos. *Fast Radial Basis Functions for Engineering Applications*. Springer, 2018.