CHAPTER

Spatiotemporal interpolation methods for air pollution

7

Lixin Li[a], **Weitian Tong**[b], **Reinhard Piltner**[c]

[a]*Department of Computer Science, Georgia Southern University, Statesboro, GA, United States*
[b]*Department of Computer Science, Eastern Michigan University, Ypsilanti, MI, United States*
[c]*Department of Mathematical Sciences, Georgia Southern University, Statesboro, GA, United States*

7.1 Introduction

Spatial interpolation methods have been well developed to estimate values at unknown locations based upon values that are spatially sampled in Geographic Information System (GIS). These methods assume a stronger correlation among points that are closer than those farther apart. They are characterized as either deterministic or stochastic depending on whether statistical properties are utilized. Deterministic interpolation methods determine an unknown value using mathematical functions with predefined parameters such as distances in inverse distance weighting (IDW) (Robichaud and Ménard, 2014; Shepard, 1968) and areas or volumes in shape function (SF)-based methods (Li and Revesz, 2004; Zienkiewics and Taylor, 2000). Many previous studies have used deterministic interpolation methods, such as radial basis functions (RBFs) (Franke and Schaback, 1998), spline (de Boor, 2001), natural neighbor (Sibson, 1981), and trend surfaces (Zurflueh, 1967). Stochastic interpolation methods such as Kriging (Krige, 1966) investigate the spatial autocorrelation and give estimates of model errors. Stochastic interpolation methods have been used to handle areal interpolation uncertainty (Geddes et al., 2013), model-data fusion (sometimes called *analysis*) (Blond and Vautard, 2004; Pagowski et al., 2010), and optimal interpolation (Robichaud et al., 2016).

Although spatial interpolation methods have been widely adopted in various GIS applications, many critical problems remain unsolved. One of them is that traditional spatial interpolation methods tend to treat space and time separately when interpolation needs to be conducted in a continuous space-time domain. The primary strategy identified from the literature is to reduce spatiotemporal interpolation problems to a sequence of snapshots of spatial interpolations (Liao et al., 2006). In order to interpolate at an unsampled time instance, temporal interpolation can then be conducted based on the spatial interpolation results at each location (Borak and Jasinski, 2009).

Spatiotemporal Analysis of Air Pollution and Its Application in Public Health. https://doi.org/10.1016/B978-0-12-815822-7.00007-8

Integrating space and time simultaneously is shown to yield better interpolation results than treating them separately for certain typical GIS applications (Li et al., 2012). Unfortunately, there are relatively fewer models for spatiotemporal interpolation compared with spatial interpolation, especially in the application of air pollution over a large geographic area. The first exception is a study that investigated the Kriging-based spatiotemporal interpolation approaches for daily mean PM_{10} concentrations (Gräler et al., 2013). The methods used included separate daily variogram estimates, temporally evolving variograms, the metric model, the separable covariance model, and the product-sum model, and are combined with multiple linear regression. These methods were applied to daily mean rural background PM_{10} concentrations across Europe for the year 2005. The second exception is IDW-based spatiotemporal interpolation methods in Li et al. (2014) and Tong et al. (2019b) that extended the traditional spatial IDW to interpolate daily $PM_{2.5}$ concentrations at the centroids of census block groups and counties across the contiguous United States. In these studies, various IDW-based spatiotemporal interpolation methods with different parameter configurations were evaluated and compared by cross-validation. Parallel programming techniques and an advanced data structure, named k-d tree, were adapted to address the computational challenges. The third exception is an SF-based spatiotemporal interpolation method in Li et al. (2016b) that extends the popular SFs in engineering applications such as finite element algorithms. This study compared the SF-based method with the IDW-based methods in Li et al. (2014) using the same $PM_{2.5}$ data and combined interpolation results with population data to estimate the population exposure to $PM_{2.5}$ in the contiguous United States. Furthermore, Zou et al. (2011) reviewed some air pollution exposure assessment methods utilized in epidemiological studies and the use of GIS for resolving problems with spatiotemporal attributes. In summary, in the era of big data, there is a need to develop and evaluate spatiotemporal methods that produce good interpolation results with computational efficiency to handle the increasing amount of air pollution data over a large geographic area.

Some recent research on spatiotemporal data interpolation and/or health effects include Xu et al. (2014), Amato and Vecchia (2018), Mei et al. (2016, 2017), Yang et al. (2017), Dunea et al. (2016), Susanto et al. (2016), Liang et al. (2017), Delikhoon et al. (2018), Bruno et al. (2016), Li et al. (2016a), Chen et al. (2017), Singh and Toshniwal (2019), Nyhan et al. (2016), Lassman et al. (2017), Safaie et al. (2017), and Wang et al. (2019). It is worth noting that there is a new line of research that applies machine learning to solve spatiotemporal interpolation problems (Brokamp et al., 2018; Chen et al., 2018; Fan et al., 2017; Gupta and Christopher, 2009; Hu et al., 2017; Qi et al., 2018; Reid et al., 2015; Tong et al., 2019a; Wang and Song, 2018; Zou et al., 2015).

In this chapter, we will focus on deterministic spatiotemporal interpolation methods based on SFs, IDW, and RBFs. First, we introduce two SF-based spatiotemporal interpolation methods in Section 7.2 and show how to implement the algorithms. In Chapter 8, we will show how to apply and compare these specific SF-based interpolation methods on a set of real-time air pollution data over a large geographic

area. Second, we introduce IDW-based spatiotemporal interpolation methods in Section 7.3. Third, we introduce RBF-based spatiotemporal interpolation methods in Section 7.4.

7.2 SF-based spatiotemporal interpolation

7.2.1 Extension approach for SF-based spatiotemporal interpolation

In order to integrate space and time simultaneously, we developed an "extension approach" to conduct the SF-based spatiotemporal interpolation. This approach treats time as another dimension in space and therefore, extending the spatiotemporal interpolation problem into a higher-dimensional spatial interpolation problem (Li, 2003). Some applications using the extension approach can be found in Li and Revesz (2004) and Li et al. (2006, 2008, 2011, 2012).

Using the extension approach of SF-based interpolation method, we treat time as the imaginary third dimension z in space. Therefore, SF-based spatiotemporal interpolation method for two-dimensional (2D) space and one-dimensional (1D) time problems can be summarized as

$$\begin{aligned} w(x,y,t) = {} & N_1(x,y,t)w_1 + N_2(x,y,t)w_2 \\ & + N_3(x,y,t)w_3 + N_4(x,y,t)w_4 \end{aligned} \tag{7.1}$$

where N_1, N_2, N_3, and N_4 are the following SFs.

$$\begin{aligned} N_1(x,y,t) &= \frac{\mathcal{V}_1}{\mathcal{V}}, N_2(x,y,t) = \frac{\mathcal{V}_2}{\mathcal{V}} \\ N_3(x,y,t) &= \frac{\mathcal{V}_3}{\mathcal{V}}, N_4(x,y,t) = \frac{\mathcal{V}_4}{\mathcal{V}} \end{aligned} \tag{7.2}$$

$\mathcal{V}_1$, $\mathcal{V}_2$, $\mathcal{V}_3$, and $\mathcal{V}_4$ are the volumes of the four subtetrahedra $ww_2w_3w_4$, $w_1ww_3w_4$, $w_1w_2ww_4$, and $w_1w_2w_3w$, respectively; and $\mathcal{V}$ is the volume of the outside tetrahedron $w_1w_2w_3w_4$ as shown in Fig. 7.1.

Alternatively, the SFs can be computed as

$$\begin{aligned} N_1(x,y,t) &= \frac{a_1 + b_1x + c_1y + d_1t}{6\mathcal{V}} \\ N_2(x,y,t) &= \frac{a_2 + b_2x + c_2y + d_2t}{6\mathcal{V}} \\ N_3(x,y,t) &= \frac{a_3 + b_3x + c_3y + d_3t}{6\mathcal{V}} \\ N_4(x,y,t) &= \frac{a_4 + b_4x + c_4y + d_4t}{6\mathcal{V}} \end{aligned} \tag{7.3}$$

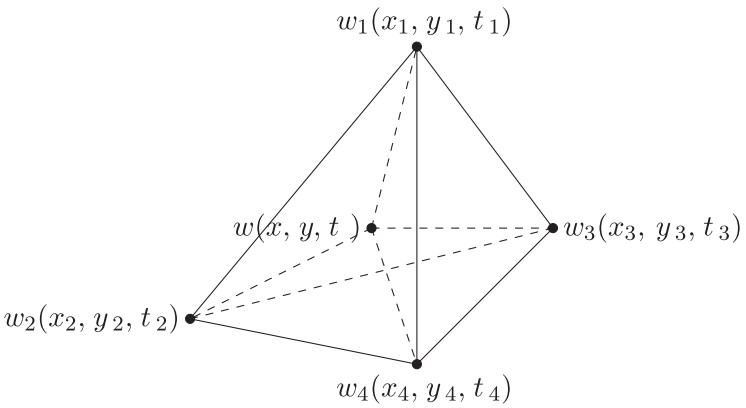

FIG. 7.1

A tetrahedral element. Computing 3D shape functions by tetrahedral volume divisions. w_1, w_2, w_3, and w_4 are measured values, while the value w at the location (x, y, t) is unknown and needs to be interpolated.

The volume $\mathcal{V}$ can be computed using the corner coordinates (x_i, y_i, t_i) $(i = 1, 2, 3, 4)$ in the determinant of a matrix as

$$\mathcal{V} = \frac{1}{6} \det \begin{bmatrix} 1 & x_1 & y_1 & t_1 \\ 1 & x_2 & y_2 & t_2 \\ 1 & x_3 & y_3 & t_3 \\ 1 & x_4 & y_4 & t_4 \end{bmatrix} \tag{7.4}$$

By expanding the other relevant determinants into their cofactors, we have

$$a_1 = \det \begin{bmatrix} x_2 & y_2 & t_2 \\ x_3 & y_3 & t_3 \\ x_4 & y_4 & t_4 \end{bmatrix} \quad b_1 = -\det \begin{bmatrix} 1 & y_2 & t_2 \\ 1 & y_3 & t_3 \\ 1 & y_4 & t_4 \end{bmatrix}$$

$$c_1 = -\det \begin{bmatrix} x_2 & 1 & t_2 \\ x_3 & 1 & t_3 \\ x_4 & 1 & t_4 \end{bmatrix} \quad d_1 = -\det \begin{bmatrix} x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \\ x_4 & y_4 & 1 \end{bmatrix}$$

with the other constants defined by cyclic interchange of the subscripts in the order of 4, 1, 2, 3 (Zienkiewics and Taylor, 2000).

The extension method is based on three-dimensional (3D) Delaunay triangulation. During implementation, we first generate a 3D Delaunay triangulation from a set of input points with (x, y, t) coordinates. The triangulation process generates an index for the containing tetrahedron in the Delaunay mesh, as well as barycentric coordinates associated with the query point within its containing tetrahedron. Barycentric coordinates are equivalent to the N_1, N_2, N_3, and N_4 coefficients in Eqs. (7.1), (7.2).

Table 7.1 Algorithm 1 using the SF-based extension method: *interpolate (A, B, dtMesh)*

Input:	*A*—a matrix representing the **known** data in the form [*t x y w*], where *t* is the time and *x* and *y* are the location coordinates, and *w* is the known measured value;
	B—a matrix of **query** points in the form [*t x y*], where *t* is the time and *x* and *y* are the location coordinates;
	dtMesh (optional)—the triangulation used to perform the interpolation;
Output:	*res*—a matrix containing the query point coordinates and the interpolated value at each query point in the form [*t x y w*].
1.	Check format of input arguments;
	// if the *dtMesh* is not provided, calculate one
2.	**if** *dtMesh* is null
	dtMesh = calculate *Delaunay Triangulation* using *t*, *x*, *y* of *A*;
	end
	// *pl* is the coordinates of the simplex containing the query point
	// *bc* are the barycentric coordinates
3.	[*pl*, *bc*] = *dtMesh.pointLocation*(B);
4.	*triVals* = extract *w* from *A* at each corner identified in *pl*;
	//calculate the interpolated values at each query point
5.	*Vq* = *bc* ·*triVals*;
6.	Set *Vq* to *NaN* if no containing tetrahedron was found;
7.	*res* = append *Vq* to *B*.

The interpolated value at the query point is then calculated as the dot product of the value at each corner of the containing tetrahedron with the barycentric coordinates. Error handling is incorporated to handle an edge case where query points are not located within the convex hull of the Delaunay mesh. The algorithm of the extension method is presented in Table 7.1.

7.2.2 Reduction approach for SF-based spatiotemporal interpolation

Alternatively, we developed a "reduction approach." This approach reduces the spatiotemporal interpolation problem to a regular spatial interpolation case using two steps. First, we interpolate (using any 1D interpolation in time) the value of interest overtime at each sample point. Second, by substituting the desired time instant into some regular 2D spatial interpolation functions, we can get spatiotemporal interpolation results (Li, 2003).

Assume the value at node i at time t_1 is w_{i1}, and at time t_2 the value is w_{i2}. The value at the node i at any time between t_1 and t_2 can be approximated using a 1D time SF in the following way as

$$w_i(t) = \frac{t_2 - t}{t_2 - t_1} w_{i1} + \frac{t - t_1}{t_2 - t_1} w_{i2} \tag{7.5}$$

The reduction method is based on 2D Delaunay triangulation. We first generate a 2D Delaunay triangulation using a set of monitoring site locations with (x, y) coordinates. Then, we need to find the containing triangle of a query point. The SF-based interpolation result at a query point (x, y) located inside the triangle can be obtained by using the measurement values w_1, w_2, and w_3 at the three corner vertices as below (Li and Revesz, 2004):

$$w(x,y) = N_1(x,y)w_1 + N_2(x,y)w_2 + N_3(x,y)w_3 \tag{7.6}$$

where N_1, N_2, and N_3 are the following linear SFs (Li and Revesz, 2002):

$$N_1(x,y) = \frac{\mathcal{A}_1}{\mathcal{A}}, N_2(x,y) = \frac{\mathcal{A}_2}{\mathcal{A}}, N_3(x,y) = \frac{\mathcal{A}_3}{\mathcal{A}} \tag{7.7}$$

$\mathcal{A}_1$, $\mathcal{A}_2$, and $\mathcal{A}_3$ are the areas of the three subtriangles ww_2w_3, w_1ww_3, and w_1w_2w, respectively; and $\mathcal{A}$ is the area of the outside triangle $w_1w_2w_3$.

Using Eqs. (7.5), (7.6), the interpolation function for any point constraint to a triangular element with corner vertices at any time between t_1 and t_2 can be expressed as follows (Li and Revesz, 2002):

$$N_1(x,y) = \frac{\mathcal{A}_1}{\mathcal{A}}, N_2(x,y) = \frac{\mathcal{A}_2}{\mathcal{A}}, N_3(x,y) = \frac{\mathcal{A}_3}{\mathcal{A}} \tag{7.8}$$

Since only 2D coordinate data are used in the reduction method, in order to deal with data at missing times, a linear time interpolation derived from Eq. (7.5) is used in Eq. (7.8).

Similar to the extension method, query points must be within the convex hull of the triangulation when using the reduction method. Since the triangulation in 2D geographic space only addresses the location coordinates, an additional restriction applies to the reduction method in the time domain. A query point must request a time that is between known measurements at each corner of the containing 2D triangle. If this is not the case, then values at the containing triangle corners can be extrapolated in the time dimension (prior to interpolating the value of the query point in the 2D coordinate dimensions using SFs). The algorithm of the reduction method is presented in Table 7.2.

7.3 IDW-based spatiotemporal interpolation

Similar as SF-based interpolation methods, IDW interpolation assumes that each measured point has a local influence that diminishes with distance. Instead of forming triangular or tetrehedral meshes as in SF-based interpolation methods, IDW needs to find nearest neighbors. Points at a shorter distance are given high weights, whereas points at a far distance are given small weights.

Table 7.2 Algorithm 2 using the SF-based reduction method: *interpolate2D (A, B, Da, T)*

Input:	*A*—a matrix representing the **known** data in the form [*t x y w*], where *t* is the time and *x* and *y* are the location coordinates, and *w* is the known measured value;
	B—a matrix of **query** points in the form [*t x y*], where *t* is the time and *x* and *y* are the location coordinates;
	dtMesh (optional)—the triangulation used to perform the interpolation;
	Da (optional)—a matrix containing the time interpolated values for all locations in *A*;
	T (optional)—a vector containing all times in *A*;
Output:	*res*—a matrix containing the query point coordinates and the interpolated value at each query point in the form [*t x y w*]
	dtMesh—the triangulation used to perform the interpolation;
	Da—a matrix containing the time interpolated values for all locations in *A*;
	T—a vector containing all distinct times in *A*.
1.	Check format of input arguments;
	// if the *dtMesh*, *Da*, *T* are not provided, calculate them
2.	**if** *dtMesh, Da,* or *T* is null
	dtMesh = calculate *Delaunay Triangulation* using *t*, *x*, *y* of *A*;
	Da = calculate time-interpolated value at the Cartesian product of all locations in *A* with all times in *T*
	T = collect unique times in *A*;
	end
	// *pl* is the coordinates of the simplex containing the query point
	// *bc* are the barycentric coordinates
3.	[*pl*, *bc*] = *dtMesh.pointLocation*(B);
4.	*triVals* = extract *w* from *A* at each corner identified in *pl*;
	//calculate the interpolated values at each query point
5.	*Vq* = *bc* ·*triVals*;
6.	Set *Vq* to *NaN* if no containing tetrahedron was found;
7.	*res* = append *Vq* to *B*.

According to Johnston et al. (2001), the general formula of IDW interpolation is the following:

$$w(x,y)=\sum_{i=1}^{N}\lambda_i w_i, \lambda_i=\frac{\left(\frac{1}{d_i}\right)^p}{\sum_{k=1}^{N}\left(\frac{1}{d_k}\right)^p} \tag{7.9}$$

where $w(x, y)$ is the predicted value at location (x, y), N is the number of nearest known points surrounding (x, y), λ_i are the weights assigned to each known point

value w_i at location (x_i, y_i), d_i are the Euclidean distances between each (x_i, y_i) and (x, y), and p is the exponent, which influences the weighting of w_i on w.

IDW-based reduction and extension approaches to 2D problem have been introduced in Li and Revesz (2004).

7.3.1 Extension approach for IDW-based spatiotemporal interpolation

Since this method treats time as a third dimension, the formula of IDW-based spatiotemporal interpolation using the extension approach is

$$w(x,y,t)=\sum_{i=1}^{N}\lambda_i w_i, \lambda_i=\frac{\left(\frac{1}{d_i}\right)^p}{\sum_{k=1}^{N}\left(\frac{1}{d_k}\right)^p} \tag{7.10}$$

with $d_i=\sqrt{(x_i-x)^2+(y_i-y)^2+(t_i-t)^2}$.

7.3.2 Reduction approach for IDW-based spatiotemporal interpolation

Assume we are interested in the value of the unsampled point at location (x, y) and time t. The IDW-based reduction method first finds the nearest neighbors of each unsampled point and calculates the corresponding weights λ_i using 2D Euclidean distance $d_i=\sqrt{(x_i-x)^2+(y_i-y)^2}$. Then, it calculates for each neighbor the value at time t by some temporal interpolation method. If we use 1D time SF (Eq. 7.5) for the temporal interpolation, the formula of IDW-based reduction method can be expressed as

$$w(x,y,t)=\sum_{i=1}^{N}\lambda_i w_i(t), \lambda_i=\frac{\left(\frac{1}{d_i}\right)^p}{\sum_{k=1}^{N}\left(\frac{1}{d_k}\right)^p} \tag{7.11}$$

where

$$w_i(t)=\frac{t_{i2}-t}{t_{i2}-t_{i1}}w_{i1}+\frac{t-t_{i1}}{t_{i2}-t_{i1}}w_{i2} \tag{7.12}$$

Each neighbor may have different beginning and ending times t_{i1} and t_{i2} in Eq. (7.12) if each points are sampled at different times.

7.4 RBF-based spatiotemporal interpolation

In the recent decades, the use of RBFs has gained popularity for interpolating data and for approximating solutions of partial differential equations (Buhmann, 2003; Fasshauer, 2007; Franke and Schaback, 1998; Kansa, 1990; Liu and Gu, 2005; Piltner, 2019; Wendland, 2005; Wu, 1995). RBFs are also useful in computer graphics and medical imaging, RBFs showed some advantages in modeling complicated surfaces (Morse et al., 2001; Schaback, 1995).

In the context of spatiotemporal air pollution modeling, Losser et al. (2014) explored how to design and implement RBF-based spatiotemporal interpolation methods to assess the trend of daily $PM_{2.5}$ concentrations for the contiguous United States. In this study, time values are calculated with the help of a factor under the assumption that spatial and temporal dimensions are equally important when interpolating a continuous changing phenomenon in the space-time domain. Various RBF-based spatiotemporal interpolation methods were evaluated by leave-one-out cross-validation. Using various meteorological data and land-use information, Zou et al. (2015) presented a neural network based on RBF to estimate $PM_{2.5}$ concentrations in Texas. Most recently, Piltner (2018, 2019) used RBFs satisfying the 2D biharmonic equation $\Delta\Delta w = 0$ and a quadruharmonic equation $\Delta\Delta\Delta\Delta w = 0$ (see Table 7.5). The motivation for the use of biharmonic and polyharmonic solution functions is to avoid oscillating behavior of functions.

By using RBFs, it became possible to deal with higher-dimensional problems in a similar way as dealing with 2D and 3D problems. After choosing N points in a domain under consideration, we can easily construct N linearly independent basis functions by using distances measured from the chosen points. In 2D, the distance r from a point (x_q, y_q) is used: $r = \sqrt{(x-x_q)^2 + (y-y_q)^2}$. For a 3D problem involving spatial coordinates x, y, and z, the distance $r = \sqrt{(x-x_q)^2 + (y-y_q)^2 + (z-z_q)^2}$ is used. In the case of an N-dimensional problem with points $\mathbf{p} = (p_1, p_2, \ldots, p_N)$ and $\mathbf{q} = (q_1, q_2, \ldots, q_N)$, the Euclidian distance is $r = \| \mathbf{p} - \mathbf{q} \| = \sqrt{(\mathbf{p}-\mathbf{q}) \cdot (\mathbf{p}-\mathbf{q})}$.

If we want to interpolate time-dependent 2D spatial data, we can use the following coordinate for the RBFs:

$$r = \sqrt{(x-x_q)^2 + (y-y_q)^2 + c^2(t-t_q)^2} \tag{7.13}$$

The parameter c has the unit [distance/time]. Depending on the given data with spatial and time units, an appropriate numerical value for the velocity parameter has to be chosen. This could involve a few numerical experiments to find a suitable value for c.

For an interpolation task, we first choose an RBF $\varphi(r)$ from a catalog of RBFs and then add some low-order polynomial functions in P to get the interpolation function (Fasshauer, 2007; Morse et al., 2001):

$$w(\mathbf{x}_p) = \sum_{q=1}^{N} \lambda_q \varphi(\| \mathbf{x}_p - \mathbf{x}_q \|) + P(\mathbf{x}_p) \tag{7.14}$$

For a 3D problem, the polynomial part can be chosen as $P = a + bx + cy + dz$.

Some of the most commonly used RBFs with global support are listed in Table 7.3. In order to obtain a banded matrix for the linear system of equations, RBFs with local support can be used. In Table 7.4, RBFs for compactly supported cases derived by Wendland (2005) are listed.

For 2D spatial interpolation problems, Piltner suggested in Piltner (2019) the use of compactly supported RBFs which satisfy biharmonic or polyharmonic differential equations (see Table 7.5). From the bending of elastic plates governed by a biharmonic partial differential equation, we know that the bending functions have a nice looking smooth behavior.

Table 7.3 Global RBFs.

Gaussian	$\varphi(r) = e^{-cr^2}$
Multiquadric	$\varphi(r) = \sqrt{r^2 + c^2}$
Inverse multiquadric	$\varphi(r) = 1/\sqrt{r^2 + c^2}$
Thin plate spline	$\varphi(r) = r^2 \log r$
Polyharmonic	$\varphi(r) = r^{2n} \log r$

Table 7.4 Selected Wendland RBFs for $0 \leq r \leq 1$.

$\varphi(r)$	Continuity
$\varphi(r) = (1-r)_+^2$	C^0
$\varphi(r) = (1-r)_+^4(4r + 1)$	C^2
$\varphi(r) = (1-r)_+^6(35r^2 + 18r + 3)$	C^4
$\varphi(r) = (1-r)_+^8(32r^3 + 25r^2 + 8r + 1)$	C^6

Table 7.5 Compactly supported 2D Trefftz RBFs (see Piltner, 2019) with C^1 continuity $\left(\text{support radius} = a,\ R = \frac{r}{a}\right)$.

RBF	PDE
$\Phi(R) = 2R^2 \ln(R) + (1 - R^2)$	$\Delta\Delta\Phi = 0$
$w = \frac{1}{288}R^6 \ln(R) - \frac{1}{216}R^6 + \frac{1}{64}R^4$ $-\frac{1747}{1728}R^2 + 1 + \frac{857}{432}R^2 \ln(R)$	$\Delta\Delta\Delta\Delta w = 0$ and $\Delta\Delta w = \frac{1}{a^4}\Phi(R)$

For each data point $\mathbf{x} = \mathbf{x_i}$ (where $i = 1, \ldots, N$), we prescribe the function value for w as

$$w(\mathbf{x_i}) = \sum_j^N \lambda_j \varphi(\| \mathbf{x_i} - \mathbf{x_j} \|) + P(\mathbf{x_i}) = w_i \tag{7.15}$$

In this way, we generate N linear equations. For the example of a 3D spatial interpolation problem where the polynomial part involves four additional unknown parameters a, b, c, and d, we have to add four constraint equations to obtain the following system of $(N + 4)$ equations:

$$\lambda_1 \varphi(\| \mathbf{x_i} - \mathbf{x_1} \|) + \cdots + \lambda_N \varphi(\| \mathbf{x_i} - \mathbf{x}_N \|) + a + bx_i + cy_i + dz_i = w_i \tag{7.16}$$

$$\sum_j^N \lambda_j = 0$$

$$\sum_j^N \lambda_j x_j = 0$$

$$\sum_j^N \lambda_j y_j = 0$$

$$\sum_j^N \lambda_j z_j = 0$$

In matrix notation, the system of equations can be written as

$$\begin{bmatrix} \varphi_{11} & \cdots & \varphi_{1N} & 1 & x_1 & y_1 & z_1 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \varphi_{N1} & \cdots & \varphi_{NN} & 1 & x_N & y_N & z_N \\ 1 & \cdots & 1 & 0 & 0 & 0 & 0 \\ x_1 & \cdots & x_N & 0 & 0 & 0 & 0 \\ y_1 & \cdots & y_N & 0 & 0 & 0 & 0 \\ z_1 & \cdots & z_N & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_N \\ a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} w_1 \\ \vdots \\ w_N \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{7.17}$$

where

$$\varphi_{ij} = \varphi(\| \mathbf{x_i} - \mathbf{x_j} \|) \tag{7.18}$$

The coefficient matrix in the linear system of equations is symmetric.

References

Amato, U., Vecchia, B.D., 2018. On Shepard-Gupta-type operators. J. Inequal. Appl. 2018 (1), 232. https://doi.org/10.1186/s13660-018-1823-7.

Blond, N., Vautard, R., 2004. Three-dimensional ozone analyses and their use for short-term ozone forecasts. J. Geophys. Res. Atmos. 109 (D17). https://doi.org/10.1029/2004JD004515.

Borak, J.S., Jasinski, M.F., 2009. Effective interpolation of incomplete satellite-derived leaf-area index time series for the continental united states. Agric. For. Meteorol. 149 (2), 320–332.

Brokamp, C., Jandarov, R., Hossain, M., Ryan, P., 2018. Predicting daily urban fine particulate matter concentrations using a random forest model. Environ. Sci. Technol. 52 (7), 4173–4179.

Bruno, F., Cameletti, M., Franco-Villoria, M., Greco, F., Ignaccolo, R., Ippoliti, L., Valentini, P., Ventrucci, M., 2016. A survey on ecological regression for health hazard associated with air pollution. Spat. Statist. 18, 276–299. https://doi.org/10.1016/j.spasta.2016.05.003.

Buhmann, M.D., 2003. Radial Basis Functions: Theory and Implementations. Cambridge University Press, Cambridge, MA.

Chen, Y., Liu, X., Li, X., Liu, X., Yao, Y., Hu, G., Xu, X., Pei, F., 2017. Delineating urban functional areas with building-level social media data: a dynamic time warping (DTW) distance based k-medoids method. Landsc. Urban Plan. 160, 48–60. https://doi.org/10.1016/j.landurbplan.2016.12.001.

Chen, G., Li, S., Knibbs, L.D., Hamm, N., Cao, W., Li, T., Guo, J., Ren, H., Abramson, M.J., Guo, Y., 2018. A machine learning method to estimate $PM_{2.5}$ concentrations across China with remote sensing, meteorological and land use information. Sci. Total Environ. 636, 52–60.

de Boor, C., 2001. A Practical Guide to Splines. vol. 27 Springer, New York, NY.

Delikhoon, M., Fazlzadeh, M., Sorooshian, A., Baghani, A.N., Golaki, M., Ashournejad, Q., Barkhordari, A., 2018. Characteristics and health effects of formaldehyde and acetaldehyde in an urban area in Iran. Environ. Pollut. 242, 938–951. https://doi.org/10.1016/j.envpol.2018.07.037.

Dunea, D., Iordache, S., Pohoata, A., 2016. Fine particulate matter in urban environments: a trigger of respiratory symptoms in sensitive children. Int. J. Environ. Res. Public Health 13 (12). https://doi.org/10.3390/ijerph13121246.

Fan, J., Li, Q., Hou, J., Feng, X., Karimian, H., Lin, S., 2017. A spatiotemporal prediction framework for air pollution based on deep RNN. ISPRS Ann. J. Photogramm. Remote Sens. Spat. Inf. Sci. 4, 15.

Fasshauer, G.E., 2007. Meshfree approximation methods with MATLAB. Interdisciplinary Mathematical Sciences, World Scientific, Singapore.

Franke, C., Schaback, R., 1998. Solving partial differential equations by collocation using radial basis functions. Appl. Math. Comput. 93, 73–82.

Geddes, A., Elston, D.A., Hodgson, M.E.A., Birnie, R.V., 2013. Stochastic model-based methods for handling uncertainty in areal interpolation. Int. J. Geogr. Inf. Sci. 27 (4), 785–803. https://doi.org/10.1080/13658816.2012.722636.

Gräler, B., Rehr, M., Gerharz, L., Pebesma, E., 2013. Spatio-temporal analysis and interpolation of PM_{10} measurements in Europe for 2009, pp. 1–29. ETC/ACM Technical Paper.

Gupta, P., Christopher, S.A., 2009. Particulate matter air quality assessment using integrated surface, satellite, and meteorological products: 2. A neural network approach. J. Geophys. Res. Atmos. 114 (D20), 1–14.

Hu, X., Belle, J.H., Meng, X., Wildani, A., Waller, L.A., Strickland, M.J., Liu, Y., 2017. Estimating $PM_{2.5}$ concentrations in the conterminous United States using the random forest approach. Environ. Sci. Technol. 51 (12), 6936–6944.

Johnston, K., Ver Hoef, J.M.V., Krivoruchko, K., Lucas, N., 2001. Using ArcGIS Geostatistical Analyst. ESRI Press, Redlands, CA.

Kansa, E.J., 1990. Multiquadrics-A scattered data approximation scheme with applications to computational fluid-dynamics-I surface approximations and partial derivative estimates. Comput. Math. Appl. 66 (8–9), 127–145. https://doi.org/10.1016/0898-1221(90)90270-T.

Krige, D.G., 1966. Two dimensional weighted moving average trend surfaces for ore evaluation. J. Soc. Afr. Inst. Min. Metall. 66, 13–38.

Lassman, W., Ford, B., Gan, R.W., Pfister, G., Magzamen, S., Fischer, E.V., Pierce, J.R., 2017. Spatial and temporal estimates of population exposure to wildfire smoke during the Washington State 2012 wildfire season using blended model, satellite, and in situ data. GeoHealth 1 (3), 106–121. https://doi.org/10.1002/2017GH000049.

Li, L., 2003, May. Spatiotemporal Interpolation Methods in GIS (Ph.D. thesis), University of Nebraska-Lincoln, Lincoln, Nebraska.

Li, L., Revesz, P., 2002. A comparison of spatio-temporal interpolation methods. In: Proceedings of the Second International Conference on GIScience 2002. Lecture Notes in Computer Science, vol. 2478. Springer, pp. 145–160.

Li, L., Revesz, P., 2004. Interpolation methods for spatio-temporal geographic data. J. Comput. Environ. Urban Syst. 28 (3), 201–227.

Li, L., Zhang, X., Piltner, R., 2006. A spatiotemporal database for ozone in the conterminous U.S. In: Proceedings of the Thirteenth International Symposium on Temporal Representation and Reasoning. IEEE, Budapest, Hungary, pp. 168–176.

Li, L., Zhang, X., Piltner, R., 2008. An application of the shape function based spatiotemporal interpolation method on ozone and population exposure in the contiguous U.S. J. Environ. Inf. 12 (2), 120–128.

Li, L., Zhang, X., Holt, J., Tian, J., Piltner, R., 2011. Spatiotemporal interpolation methods for air pollution exposure. In: Proceedings of the Ninth Symposium on Abstraction, Reformulation, and Approximation. AAAI (Association for the Advancement of Artificial Intelligence), Parador de Cardona, Spain, pp. 75–81.

Li, L., Zhang, X., Holt, J.B., Tian, J., Piltner, R., 2012. Estimating population exposure to fine particulate matter in the conterminous U.S. using shape function-based spatiotemporal interpolation method: a county level analysis. GSTF Int. J. Comput. 1, 24–30.

Li, L., Losser, T., Yorke, C., Piltner, R., 2014. Fast inverse distance weighting-based spatiotemporal interpolation: a web-based application of interpolating daily fine particulate matter $PM_{2.5}$ in the contiguous U.S. using parallel programming and k-d tree. Int. J. Environ. Res. Public Health 11 (9), 9101–9141.

Li, H., Fan, H., Mao, F., 2016. A visualization approach to air pollution data exploration—a case study of air quality index (PM2.5) in Beijing, China. Atmosphere 7 (3). https://doi.org/10.3390/atmos7030035.

Li, L., Zhou, X., Kalo, M., Piltner, R., 2016. Spatiotemporal interpolation methods for the application of estimating population exposure to fine particulate matter in the contiguous U.S. and a real-time web application. Int. J. Environ. Res. Public Health 13 (8), 749. https://doi.org/10.3390/ijerph13080749. 20 p.

Liang, F., Gao, M., Xiao, Q., Carmichael, G.R., Pan, X., Liu, Y., 2017. Evaluation of a data fusion approach to estimate daily PM2.5 levels in North China. Environ. Res. 158, 54–60. https://doi.org/10.1016/j.envres.2017.06.001.

Liao, D., Peuquet, D.J., Duan, Y., Whitsel, E.A., Dou, J., Smith, R.L., Lin, H.-M., Chen, J.-C., Heiss, G., 2006. GIS approaches for the estimation of residential-level ambient PM concentrations. Environ. Health Perspect. 114 (9), 1374–1380.

Liu, G.-R., Gu, Y.-T., 2005. An Introduction to Meshfree Methods and Their Programming. Springer, New York, NY.

Losser, T., Li, L., Piltner, R., 2014. A spatiotemporal interpolation method using radial basis functions for geospatiotemporal big data. In: Proceedings of the 5th International Conference on Computing for Geospatial Research and Application. IEEE, Washington, D.C, pp. 17–24.

Mei, G., Xu, N., Xu, L., 2016. Improving GPU-accelerated adaptive IDW interpolation algorithm using fast kNN search. SpringerPlus 5 (1), 1389. https://doi.org/10.1186/s40064-016-3035-2.

Mei, G., Xu, L., Xu, N., 2017. Accelerating adaptive inverse distance weighting interpolation algorithm on a graphics processing unit. R. Soc. Open Sci. 4 (9), 170436. https://doi.org/10.1098/rsos.170436.

Morse, B.S., Yoo, T.S., Rheingans, P., Chen, D.T., Subramanian, K.R., 2001. Interpolating implicit surfaces from scattered surface data using compactly supported radial basis functions. In: SMI'01: Proceedings of the International Conference on Shape Modeling and Applications, pp. 89–98.

Nyhan, M., Grauwin, S., Britter, R., Misstear, B., McNabola, A., Laden, F., Barrett, S.R.H., Ratti, C., 2016. Exposure track-the impact of mobile-device-based mobility patterns on quantifying population exposure to air pollution. Environ. Sci. Technol. 50 (17), 9671–9681. https://doi.org/10.1021/acs.est.6b02385.

Pagowski, M., Grell, G.A., McKeen, S.A., Peckham, S.E., Devenyi, D., 2010. Three-dimensional variational data assimilation of ozone and fine particulate matter observations: some results using the weather research and forecasting—chemistry model and grid-point statistical interpolation. Q. J. R. Meteorol. Soc. 136 (653), 2013–2024. https://doi.org/10.1002/qj.700.

Piltner, R., 2018. Exploring new options for data interpolation with radial basis functions. In: MOBIMEDIA'18 Proceedings of the 11th EAI International Conference on Mobile Multimedia Communications, Workshop on Environmental Health and Air Pollution. EAI, Qingdao, People's Republic of China, pp. 228–231.

Piltner, R., 2019. Some remarks on Trefftz type approximations. Eng. Anal. Bound. Elem. 101, 102–112. https://doi.org/10.1016/j.enganabound.2018.12.010.

Qi, Z., Wang, T., Song, G., Hu, W., Li, X., Zhang, Z.M., 2018. Deep air learning: interpolation, prediction, and feature analysis of fine-grained air quality. IEEE Trans. Knowl. Data Eng 30 (12), 2285–2297.

Reid, C.E., Jerrett, M., Petersen, M.L., Pfister, G.G., Morefield, P.E., Tager, I.B., Raffuse, S.M., Balmes, J.R., 2015. Spatiotemporal prediction of fine particulate matter during the 2008 Northern California wildfires using machine learning. Environ. Sci. Technol. 49 (6), 3887–3896.

Robichaud, A., Ménard, R., 2014. Multi-year objective analyses of warm season ground-level ozone and $PM_{2.5}$ over North America using real-time observations and Canadian operational air quality models. Atmos. Chem. Phys. 14, 1769–1800.

Robichaud, A., Ménard, R., Zaïtseva, Y., Anselmo, D., 2016. Multi-pollutant surface objective analyses and mapping of air quality health index over North America. Air Qual. Atmos. Health, 1–17. https://doi.org/10.1007/s11869-015-0385-9.

Safaie, A., Dang, C., Qiu, H., Radha, H., Phanikumar, M.S., 2017. Manifold methods for assimilating geophysical and meteorological data in earth system models and their components. J. Hydrol. 544, 383–396. https://doi.org/10.1016/j.jhydrol.2016.11.009.

Schaback, R., 1995. Creating surfaces from scattered data using radial basis functions. In: Mathematical Methods for Curves and Surfaces. Vanderbilt University Press, Nashville, TN, pp. 477–496.

Shepard, D., 1968. A two-dimensional interpolation function for irregularly spaced data. In: Proceedings of the 23rd National Conference ACM. ACM, New York, NY, pp. 517–524.

Sibson, R., 1981. A brief description of natural neighbor interpolation. In: Barnett, V. (Ed.), Interpreting Multivariate Data. John Wiley & Sons, Inc., New York, NY, pp. 21–36.

Singh, B., Toshniwal, D., 2019. MOWM: multiple overlapping window method for RBF based missing value prediction on big data. Expert Syst. Appl. 122, 303–318. https://doi.org/10.1016/j.eswa.2018.12.060.

Susanto, F., De Souza, P., He, J., 2016. Spatiotemporal interpolation for environmental modelling. Sensors 16 (8). https://doi.org/10.3390/s16081245.

Tong, W., Li, L., Zhou, X., Hamilton, A., Zhang, K., 2019. Deep learning air pollution with bidirectional LSTM RNN. Air Qual. Atmos. Health 12 (4), 411–423.

Tong, W., Li, L., Zhou, X., Franklin, J., 2019. Efficient spatiotemporal interpolation with Spark machine learning. Earth Sci. Inf. 12 (1), 87–96.

Wang, J., Song, G., 2018. A deep spatial-temporal ensemble model for air quality prediction. Neurocomputing 314, 198–206.

Wang, J., Zhang, J., Feng, Y., 2019. Characterizing the spatial variability of soil particle size distribution in an underground coal mining area: an approach combining multi-fractal theory and geostatistics. CATENA 176, 94–103. https://doi.org/10.1016/j.catena.2019.01.011.

Wendland, H., 2005. Scattered Data Approximation. vol. 17 Cambridge University Press, Cambridge.

Wu, Z., 1995. Compactly supported positive definite radial functions. Adv. Comput. Math. 4 (1), 283–292. https://doi.org/10.1007/BF03177517.

Xu, M., Guo, Y., Zhang, Y., Westerdahl, D., Mo, Y., Liang, F., Pan, X., 2014. Spatiotemporal analysis of particulate air pollution and ischemic heart disease mortality in Beijing, China. Environ. Health 13 (1), 109. https://doi.org/10.1186/1476-069X-13-109.

Yang, W., Wang, G., Bi, C., 2017. Analysis of long-range transport effects on PM2.5 during a short severe haze in Beijing, China. Aerosol Air Qual. Res. 17 (6), 1610–1622. https://doi.org/10.4209/aaqr.2016.06.0220.

Zienkiewics, O.C., Taylor, R.L., 2000. Finite Element Method, vol. 1, The Basis. Butterworth-Heinemann, London.

Zou, B., Wilson, J.G., Zhan, F.B., Zeng, Y., 2011. Air pollution exposure assessment methods utilized in epidemiological studies. J. Environ. Monit. 11, 475–490.

Zou, B., Wang, M., Wan, N., Wilson, J.G., Fang, X., Tang, Y., 2015. Spatial modeling of $PM_{2.5}$ concentrations with a multifactoral radial basis function neural network. Environ. Sci. Pollut. Res. 22 (14), 10395–10404.

Zurflueh, E.G., 1967. Applications of two-dimensional linear wavelength filtering. Geophysics 32, 1015–1035.