HYDROLOGICAL SCIENCES JOURNAL – JOURNAL DES SCIENCES HYDROLOGIQUES, 2016
VOL. 61, NO. 3, 519–526
http://dx.doi.org/10.1080/02626667.2015.1051980

Bilinear surface smoothing for spatial interpolation with optional incorporation of an explanatory variable. Part 1: Theory

Nikolaos Malamos [a] and Demetris Koutsoyiannis [b]

[a]Department of Agricultural Technology, Technological Educational Institute of Western Greece, Amaliada, Greece; [b]Department of Water Resources and Environmental Engineering, School of Civil Engineering, National Technical University of Athens, Zographou, Greece

ABSTRACT
Bilinear surface smoothing is an alternative concept that provides flexible means for spatial interpolation. Interpolation is accomplished by means of fitting a bilinear surface into a regression model with known break points and adjustable smoothing terms. Additionally, as an option, the incorporation, in an objective manner, of the influence of an explanatory variable available at a considerable denser dataset is possible. The parameters involved in each case (with or without an explanatory variable) are determined by a nonparametric approach based on the generalized cross-validation (GCV) methodology. A convenient search technique for the smoothing parameters was achieved by transforming them in terms of tension parameters, with values restricted in the interval [0, 1). The mathematical framework, the computational implementation and details concerning both versions of the methodology, as well as practical aspects of their application are presented and discussed. In a companion paper, examples using both synthesized and real-world (hydrological) data are presented to illustrate the methodology. The proposed mathematical framework constitutes a simple alternative to existing spatial interpolation methodologies.

ARTICLE HISTORY
Received 19 July 2014
Accepted 14 April 2015

EDITOR
Z.W. Kundzewicz

ASSOCIATE EDITOR
A. Carsteanu

KEYWORDS
bilinear surface smoothing; spatial interpolation; explanatory variable; generalized cross-validation; linear smoothers; Cholesky decomposition

Introduction

In multidimensional interpolation, we seek estimates of the dependent variables at points placed inside the analysis space that forms regular or irregular sized grids. In order to achieve such an objective, various techniques have been deployed; many of them can be applied to perform spatial interpolation of environmental variables that are usually collected from point measurements.

These methodologies fall into three main categories (Li and Heap 2008):

(1) non-geostatistical methods, such as splines, thin plate splines (Craven and Wahba 1978, Wahba and Wendelberger 1980) and regression methods (Davis 1986);
(2) geostatistical methods including different approaches to kriging, such as ordinary and universal kriging, kriging with an external drift or co-kriging (Goovaerts 1997, Burrough and McDonnell 1998); and
(3) combined methods, such as trend surface analysis combined with kriging (Wang *et al.* 2005), regression kriging (Hengl *et al.* 2007) and stochastic interpolation (Sauquet *et al.* 2000).

Koutsoyiannis (2000) presented the so-called broken line smoothing (BLS) as a simple alternative to numerical smoothing and interpolating methods, related to piecewise linear regression and to smoothing splines. The idea was to approximate a smooth curve that may be drawn for the data points (x_i, y_i) with a broken line or open polygon which can be numerically estimated by means of a least squares fitting procedure. The abscissae of the vertices of the broken line did not necessarily coincide with x_is but they formed a series of points with some chosen, lower or higher, resolution.

Malamos and Koutsoyiannis (2014) extended the previous method by utilizing the combination of two broken lines into a regression model with known break points and adjustable weights (BLSI). The first broken line was fitted to the available data points, while the second incorporated, in an objective manner, the influence of an explanatory variable available at a considerably denser dataset. The objective was to improve the accuracy of interpolation across the data points.

The concept, for both methodologies, was the trade-off between the two objectives of minimizing the fitting error and the roughness of the broken lines. The larger the relative weight of the second objective is, the smoother the broken lines resulting from the fitting procedure will be.

In the present study the method is generalized for the case of two-dimensional data. The main idea, presented as bilinear surface smoothing (BSS), is to approximate a surface that may be drawn for the data points (x_i, y_i) with consecutive bilinear surfaces which can be estimated by means of a least squares fitting procedure into a surface regression model with known break points and adjustable weights. The concept was, once

CONTACT Nikolaos Malamos nmalamos@teimes.gr

more, a trade-off between the two objectives of minimizing the fitting error and the roughness of the bilinear surface.

Additionally, a second version of the methodology (BSSE) is presented, which is focused on the combination of two bilinear surfaces into the same regression model, in order to improve the interpolation accuracy across the data points. The first surface is fitted to the available data points while the second incorporates, in an objective manner, the influence of an explanatory variable available at a spatially denser dataset.

The estimation of parameters, i.e. the number of surface segments and the values of the corresponding smoothing parameters, is accomplished by a nonparametric approach based on the generalized cross-validation (GCV) methodology (Craven and Wahba 1978, Wahba and Wendelberger 1980) and the linear smoothers theory (Buja *et al.* 1989). The simplified but efficient parameter estimation technique was established after numerical investigation and contributed to performance enhancement and accuracy of the mathematical framework.

Mathematical framework

Bilinear surface smoothing interpolation (BBS)

Let $z_i(x_i, y_i)$ be a set of n points in the three-dimensional space (x, y, z) for $i = 1, \ldots, n$. Also, let cx_l, $l = 0, \ldots, \mathrm{mx}$, be $\mathrm{mx} + 1$ points on the x-axis and cy_k, $k = 0, \ldots, \mathrm{my}$, be $\mathrm{my} + 1$ points on the y-axis, so that the rectangle with vertices $(\mathrm{cx}_0, \mathrm{cy}_0)$, $(\mathrm{cx}_{\mathrm{mx}}, \mathrm{cy}_0)$, $(\mathrm{cx}_0, \mathrm{cy}_{\mathrm{my}})$ and $(\mathrm{cx}_{\mathrm{mx}}, \mathrm{cy}_{\mathrm{my}})$ contains all (x_i, y_i). For simplicity we will assume that the points on both axes are equidistant, i.e. $\mathrm{cx}_l - \mathrm{cx}_{l-1} = \delta_x$ and $\mathrm{cy}_k - \mathrm{cy}_{k-1} = \delta_y$.

We wish to find the $m + 1$ values of d_j, where $j = 0, \ldots, m$ and $m = (\mathrm{mx} + 1)(\mathrm{my} + 1) - 1$, on the three-dimensional space (x, y, z), so that the bilinear surface defined by the $m + 1$ points $(\mathrm{cx}_l, \mathrm{cy}_k, d_j)$ "fits" the set of points $z_i(x_i, y_i)$. This fit is defined in terms of minimizing the total square error among the set of original points $z_i(x_i, y_i)$ and the fitted bilinear surface:

$$p = \sum_{i=1}^{n} (z_i - \hat{z}_i)^2 \tag{1}$$

where $\hat{z}_i$ is the estimate given by the bilinear surface for each known z_i.

In matrix form, this can be written as:

$$p = \|\boldsymbol{z} - \hat{\boldsymbol{z}}\|^2 \tag{2}$$

where $\boldsymbol{z} = [z_1, \ldots, z_n]^{\mathrm{T}}$ is the vector of known applicates of the given data points with size n (the superscript T denotes the transpose of a matrix or vector) and $\hat{\boldsymbol{z}} = [\hat{z}_1, \ldots, \hat{z}_n]^{\mathrm{T}}$ is the vector of estimates with size n.

The general estimation function will be:

$$\hat{z}_u = d_u \tag{3}$$

where u refers to a point on the (x, y) plane, while d_u is the value of the bilinear surface at that point (Fig. 1).

The relation of d_u to its four surrounding points, $d_1, \ldots, d_4$, as presented in Figure 2, is simply an application of bilinear interpolation (Press *et al.* 2002):

$$d_u = \frac{1}{\delta_x \delta_y}\left[d_1(\mathrm{cx}_l - x)(\mathrm{cy}_k - y) + d_2(x - \mathrm{cx}_{l-1})(\mathrm{cy}_k - y) + d_3(x - \mathrm{cx}_{l-1})(y - \mathrm{cy}_{k-1}) + d_4(\mathrm{cx}_l - x)(y - \mathrm{cy}_{k-1})\right] \tag{4}$$

where cy_k, cy_{k-1}, cx_l, cx_{l-1} are the coordinates of the four points and x, y are the corresponding coordinates of d_u. Notice that the bilinear function in (4) is not actually linear with respect to x and y as it contains products thereof.

Assuming that a point $z_i(x_i, y_i)$, lies in the two-dimensional space $([\mathrm{cx}_{l-1}, \mathrm{cx}_l] \times [\mathrm{cy}_{k-1}, \mathrm{cy}_k])$ for some cx_l, $(\mathrm{cx}_{l-1} \le x_i \le \mathrm{cx}_l)$ and some cy_k, $(\mathrm{cy}_{k-1} \le y_i \le \mathrm{cy}_k)$, then obviously the $\hat{z}_i$ estimate is given by:

$$\hat{z}_i(x_i, y_i) = \frac{1}{\delta_x \delta_y}\left[d_1(\mathrm{cx}_l - x_i)(\mathrm{cy}_k - y_i) + d_2(x_i - \mathrm{cx}_{l-1})(\mathrm{cy}_k - y_i) + d_3(x_i - \mathrm{cx}_{l-1})(y_i - \mathrm{cy}_{k-1}) + d_4(\mathrm{cx}_l - x_i)(y_i - \mathrm{cy}_{k-1})\right] \tag{5}$$

If we apply equation (5) for $i = 1, \ldots, n$, we get:

$$\hat{z}_1 = \frac{1}{\delta_x \delta_y}\left[d_1(\mathrm{cx}_1 - x_1)(\mathrm{cy}_1 - y_1) + d_2(x_1 - \mathrm{cx}_0)(\mathrm{cy}_1 - y_1) + d_3(x_1 - \mathrm{cx}_0)(y_1 - \mathrm{cy}_0) + d_4(\mathrm{cx}_1 - x_1)(y_1 - \mathrm{cy}_0)\right]$$

$$\vdots$$

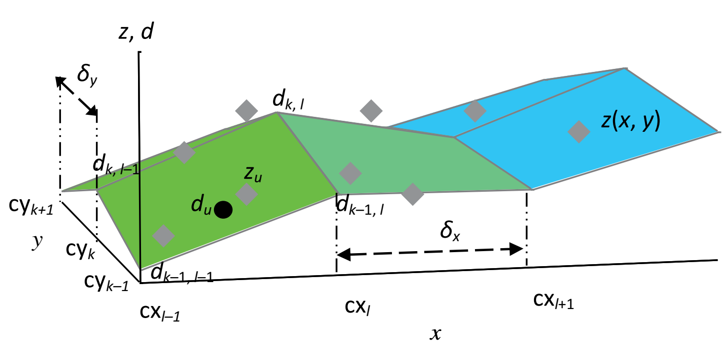

Figure 1. Definition sketch for bilinear surface d, similar for bilinear surface e.

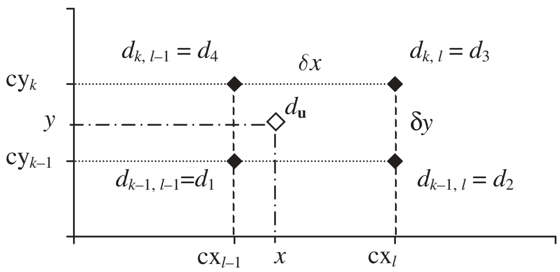

Figure 2. Definition sketch for the d_u calculation.

$$\begin{aligned}\hat{z}_n = \frac{1}{\delta_x\delta_y}[&d_{k-1,l-1}(\mathrm{cx}_l - x_n)(\mathrm{cy}_k - y_n)\\ &+ d_{k,l-1}(x_n - \mathrm{cx}_{l-1})(\mathrm{cy}_k - y_n)\\ &+ d_{k,l}(x_n - \mathrm{cx}_{l-1})(y_n - \mathrm{cy}_{k-1})\\ &+ d_{k-1,l}(\mathrm{cx}_l - x_n)(y_n - \mathrm{cy}_{k-1})]\end{aligned} \tag{6}$$

The above equations can be more concisely written in the form:

$$\hat{\boldsymbol{z}} = \boldsymbol{\Pi}\boldsymbol{d} \tag{7}$$

where $\hat{\boldsymbol{z}} = [\hat{z}_1, \ldots, \hat{z}_n]^{\mathrm{T}}$ is the vector of estimates with size n, $\boldsymbol{d} = [d_0, \ldots, d_m]^{\mathrm{T}}$ is the vector of the unknown applicates of the bilinear surface d with size $m + 1$ ($m = (\mathrm{mx} + 1)(\mathrm{my} + 1) - 1$) and $\boldsymbol{\Pi}$ is a matrix with size $n \times (m + 1)$ whose ijth entry (for $i = 1, \ldots, n; j = 0, \ldots, m$) is:

$$\pi_{ij} = \begin{cases} \frac{(\mathrm{cx}_l - x_i)(\mathrm{cy}_k - y_i)}{\delta_x\delta_y}, \text{ when } \mathrm{cx}_{l-1} < x_i \le \mathrm{cx}_l \text{ and } \mathrm{cy}_{k-1} < y_i \le \mathrm{cy}_k \\ \frac{(\mathrm{cx}_l - x_i)(y_i - \mathrm{cy}_{k-1})}{\delta_x\delta_y}, \text{ when } \mathrm{cx}_{l-1} < x_i \le \mathrm{cx}_l \text{ and } cy_k \le y_i < \mathrm{cy}_{k+1} \\ \frac{(x_i - \mathrm{cx}_{l-1})(y_i - \mathrm{cy}_{k-1})}{\delta_x\delta_y}, \text{ when } \mathrm{cx}_l \le x_i < \mathrm{cx}_{l+1} \text{ and } cy_k \le y_i < \mathrm{cy}_{k+1} \\ \frac{(x_i - \mathrm{cx}_{l-1})(\mathrm{cy}_k - y_i)}{\delta_x\delta_y}, \text{ when } \mathrm{cx}_l \le x_i < \mathrm{cx}_{l+1} \text{ and } cy_{k-1} < y_i \le \mathrm{cy}_k \\ 0, \qquad \text{otherwise} \end{cases} \tag{8}$$

In order to acquire the amount of smoothness of the bilinear surface d and to assure a unique solution of the fitting problem, we introduced the difference of slopes between two consecutive segments of the bilinear surface according to the x direction, for each cy_k point on the y-axis, by taking into account the fact that cx_ls are equidistant, as:

$$\frac{1}{\delta_x}\left(2d_{l,k} - d_{l-1,k} - d_{l+1,k}\right) \tag{9}$$

Likewise for the y direction, for each cx_l point on the x-axis, by taking into account the fact that cy_ks are equidistant, the slope difference will be:

$$\frac{1}{\delta_y}\left(2d_{k,l} - d_{k-1,l} - d_{k+1,l}\right) \tag{10}$$

Therefore, the following expressions constitute adequate smoothing terms of the bilinear surface for both directions:

$$q_{\mathrm{dx}} = \sum_{k=0}^{\mathrm{my}}\sum_{l=1}^{\mathrm{mx}-1}\left(2d_{l,k} - d_{l-1,k} - d_{l+1,k}\right)^2 \tag{11}$$

and

$$q_{\mathrm{dy}} = \sum_{l=0}^{\mathrm{mx}}\sum_{k=1}^{\mathrm{my}-1}\left(2d_{k,l} - d_{k-1,l} - d_{k+1,l}\right)^2 \tag{12}$$

which can easily be expressed in matrix form as follows:

$$q_{\mathrm{dx}} = \boldsymbol{d}^{\mathrm{T}}\boldsymbol{\Psi}_x^{\mathrm{T}}\boldsymbol{\Psi}_x\boldsymbol{d} \tag{13}$$

$$q_{\mathrm{dy}} = \boldsymbol{d}^{\mathrm{T}}\boldsymbol{\Psi}_y^{\mathrm{T}}\boldsymbol{\Psi}_y\boldsymbol{d} \tag{14}$$

where $\boldsymbol{\Psi}_x$ and $\boldsymbol{\Psi}_y$ are matrices with size $(m - 1) \times (m + 1)$ (for $i = 1, \ldots, m - 1$ and $j = 0, \ldots, m$). As explained in the Appendix, their ijth entry is:

$$\Psi_{x\,i,j} = \begin{cases} 2, & \text{when } i = j \text{ and } i - k(\mathrm{mx}+1) \notin \{1, \mathrm{mx}+1\} \\ -1, & \text{when } |i-j| = 1 \text{ and } i - k(\mathrm{mx}+1) \notin \{1, \mathrm{mx}+1\} \\ 0, & \text{otherwise} \end{cases} \tag{15}$$

where $k = 0, \ldots, \mathrm{my}$, while

$$\Psi_{y\,i,j} = \begin{cases} 2, & \text{when } i = j \text{ and } i - l(\mathrm{my}+1) \notin \{1, \mathrm{my}+1\} \\ -1, & \text{when } |i-j| = 1 \text{ and } i - l(\mathrm{my}+1) \notin \{1, \mathrm{my}+1\} \\ 0, & \text{otherwise} \end{cases} \tag{16}$$

with $l = 0, \ldots, \mathrm{mx}$. It is noted that matrices $\boldsymbol{\Psi}_x$ and $\boldsymbol{\Psi}_y$ are identical when mx = my.

Combining equations (2), (7), (13) and (14) and introducing dimensionless multipliers for both x and y directions in order to control the smoothness of the bilinear surface, we form the generalized objective function to be minimized:

$$\begin{aligned} f(\boldsymbol{d}) &= p + \lambda_x q_{\mathrm{dx}} + \lambda_y q_{\mathrm{dy}} \\ &= \|\boldsymbol{z} - \hat{\boldsymbol{z}}\|^2 + \lambda_x \boldsymbol{d}^{\mathrm{T}}\boldsymbol{\Psi}_x^{\mathrm{T}}\boldsymbol{\Psi}_x\boldsymbol{d} + \lambda_y \boldsymbol{d}^{\mathrm{T}}\boldsymbol{\Psi}_y^{\mathrm{T}}\boldsymbol{\Psi}_y\boldsymbol{d} \end{aligned} \tag{17}$$

where $\lambda_x \ge 0$ for q_{dx} and $\lambda_y \ge 0$ for q_{dy}.

Differentiation of equation (17) with respect to $\boldsymbol{d}$, by applying the typical rules of derivatives involving matrices and equating to zero, yields:

$$\frac{\partial f}{\partial \boldsymbol{d}} = -2\boldsymbol{z}^{\mathrm{T}}\boldsymbol{\Pi} + 2\boldsymbol{d}^{\mathrm{T}}\boldsymbol{\Pi}^{\mathrm{T}}\boldsymbol{\Pi} + 2\lambda_x\boldsymbol{d}^{\mathrm{T}}\boldsymbol{\Psi}_x^{\mathrm{T}}\boldsymbol{\Psi}_x + 2\lambda_y\boldsymbol{d}^{\mathrm{T}}\boldsymbol{\Psi}_y^{\mathrm{T}}\boldsymbol{\Psi}_y = 0 \tag{18}$$

and consequently:

$$\left(\boldsymbol{\Pi}^{\mathrm{T}}\boldsymbol{\Pi} + \lambda_x\boldsymbol{\Psi}_x^{\mathrm{T}}\boldsymbol{\Psi}_x + \lambda_y\boldsymbol{\Psi}_y^{\mathrm{T}}\boldsymbol{\Psi}_y\right)\boldsymbol{d} = \boldsymbol{\Pi}^{\mathrm{T}}\boldsymbol{z} \tag{19}$$

Finally, the solution of equation (19) that minimizes equation (17) has the following form:

$$\boldsymbol{d} = \left(\boldsymbol{\Pi}^{\mathrm{T}}\boldsymbol{\Pi} + \lambda_x\boldsymbol{\Psi}_x^{\mathrm{T}}\boldsymbol{\Psi}_x + \lambda_y\boldsymbol{\Psi}_y^{\mathrm{T}}\boldsymbol{\Psi}_y\right)^{-1}\boldsymbol{\Pi}^{\mathrm{T}}\boldsymbol{z} \tag{20}$$

The vector of estimates, $\hat{\boldsymbol{z}}$, is obtained from equation (7), once vector $\boldsymbol{d}$ is calculated from equation (20). Also, from equation (5), we can estimate the applicate $\hat{z}$ of any point that lies in the two-dimensional interval ($[\mathrm{cx}_0, \mathrm{cx}_{\mathrm{mx}}] \times [\mathrm{cy}_0, \mathrm{cy}_{\mathrm{my}}]$).

The minimum number of $m + 1$ points required to solve equation (20) is 6, according to equations (11) and (12). This is illustrated in Fig. 1 since the minimum number of points needed to define the bilinear surface, d, is the number of points that define two consecutive planes oriented according to either x or y direction.

Bilinear surface smoothing interpolation with the incorporation of explanatory variable (BSSE)

The incorporation of an explanatory variable available at a considerably denser dataset than the initial main variable constitutes a distinct interpolation method that extends the above presented mathematical framework. The methodology is based on the one-dimensional implementation presented by Malamos and Koutsoyiannis (2014).

Let $z_i(x_i, y_i)$ be the same set of n points in the three-dimensional space (x, y, z) for $i = 1, \ldots, n$, as already defined in the previously presented case.

In addition, we assume that for every (x, y) value we know the value of an explanatory variable t. Therefore, for each point $z_i(x_i, y_i)$ there corresponds a value $t(x_i, y_i)$ and for point $(\mathrm{cx}_l, \mathrm{cy}_k)$ there corresponds a value $t(\mathrm{cx}_l, \mathrm{cy}_k)$, for $l = 0, \ldots, \mathrm{mx}$, and $k = 0, \ldots, \mathrm{my}$.

We wish to find the $m + 1$ values of d_j and e_j, where $j = 0, \ldots, m$ and $m = (\mathrm{mx} + 1)(\mathrm{my} + 1) - 1$, on the three-dimensional space (x, y, z), so that the bilinear surface defined by the $m + 1$ points $[\mathrm{cx}_l, \mathrm{cy}_k, d_j + t(\mathrm{cx}_l, \mathrm{cy}_k) \times e_j]$ fits the set of points $z_i(x_i, y_i)$. This fit is meant in terms of minimizing the total square error among the set of original points $z_i(x_i, y_i)$ and the fitted bilinear surface as already presented in equations (1) and (2).

In this case, the general estimation function will be:

$$\hat{z}_u = d_u + t_u e_u \tag{21}$$

where u refers to a point on the (x, y) plane, while d_u, e_u are the values of the two bilinear surfaces at that point and t_u is the corresponding value of the explanatory variable. This is not a global linear relationship but a local linear one as the quantities d_u and e_u change with x and y.

Following the methodology presented above, we obtain the relation that provides the second bilinear surface, e_u, which is:

$$\begin{aligned} e_u = \frac{1}{\delta_x \delta_y} &[e_1(\mathrm{cx}_l - x)(\mathrm{cy}_k - y) + e_2(x - \mathrm{cx}_{l-1})(\mathrm{cy}_k - y) \\ &+ e_3(x - \mathrm{cx}_{l-1})(y - \mathrm{cy}_{k-1}) + e_4(\mathrm{cx}_l - x)(y - \mathrm{cy}_{k-1})] \end{aligned} \tag{22}$$

Assuming that a point $z_i(x_i, y_i)$, lies in the two-dimensional interval $([\mathrm{cx}_{l-1}, \mathrm{cx}_l] \times [\mathrm{cy}_{k-1}, \mathrm{cy}_k])$ for some cx_l, $(\mathrm{cx}_{l-1} \leq x_i \leq \mathrm{cx}_l)$ and some cy_k, $(\mathrm{cy}_{k-1} \leq y_i \leq \mathrm{cy}_k)$, then the estimate is given by:

$$\begin{aligned} \hat{z}_i(x_i, y_i) = \frac{1}{\delta_x \delta_y} &\{[d_1(\mathrm{cx}_l - x_i)(\mathrm{cy}_k - y_i) + d_2(x_i - \mathrm{cx}_{l-1})(\mathrm{cy}_k - y_i) \\ &+ d_3(x_1 - \mathrm{cx}_{l-1})(y_i - \mathrm{cy}_{k-1}) + d_4(\mathrm{cx}_l - x_i)(y_i - \mathrm{cy}_{k-1})] \\ &+ t(x_i, y_i)[e_1(\mathrm{cx}_l - x_i)(\mathrm{cy}_k - y_i) + e_2(x_i - \mathrm{cx}_{l-1})(\mathrm{cy}_k - y_i) \\ &+ e_3(x_i - \mathrm{cx}_{l-1})(y_i - \mathrm{cy}_{k-1}) + e_4(\mathrm{cx}_l - x_i)(y_i - \mathrm{cy}_{k-1})]\} \end{aligned} \tag{23}$$

If we apply equation (23) for $i = 1, \ldots, n$, we obtain the following form, analogous to equation (6):

$$\begin{aligned} \hat{z}_1 = \frac{1}{\delta_x \delta_y} &\{[d_1(\mathrm{cx}_1 - x_1)(\mathrm{cy}_1 - y_1) \\ &+ d_2(x_1 - \mathrm{cx}_0)(\mathrm{cy}_1 - y_1) \\ &+ d_3(x_1 - \mathrm{cx}_0)(y_1 - \mathrm{cy}_0) \\ &+ d_4(\mathrm{cx}_1 - x_1)(y_1 - \mathrm{cy}_0)] \\ &+ t(x_1, y_1)[e_1(\mathrm{cx}_1 - x_1)(\mathrm{cy}_1 - y_1) \\ &+ e_2(x_1 - \mathrm{cx}_0)(\mathrm{cy}_1 - y_1) \\ &+ e_3(x_1 - \mathrm{cx}_0)(y_1 - \mathrm{cy}_0) \\ &+ e_4(\mathrm{cx}_1 - x_1)(y_1 - \mathrm{cy}_0)] \\ &\vdots \\ \hat{z}_n = \frac{1}{\delta_x \delta_y} &\{[d_{k-1,l-1}(\mathrm{cx}_l - x_n)(\mathrm{cy}_k - y_n) \\ &+ d_{k,l-1}(x_n - \mathrm{cx}_{l-1})(\mathrm{cy}_k - y_n) \\ &+ d_{k,l}(x_n - \mathrm{cx}_{l-1})(y_n - \mathrm{cy}_{k-1}) \\ &+ d_{k-1,l}(\mathrm{cx}_l - x_n)(y_n - \mathrm{cy}_{k-1})] \\ &+ t(x_n, y_n)[e_{k-1,l-1}(\mathrm{cx}_1 - x_n)(\mathrm{cy}_1 - y_n) \\ &+ e_{k,l-1}(x_n - \mathrm{cx}_0)(\mathrm{cy}_1 - y_n) \\ &+ e_{k,l}(x_n - \mathrm{cx}_{l-1})(y_n - \mathrm{cy}_{k-1}) \\ &+ e_{k-1,l}(\mathrm{cx}_l - x_n)(y_n - \mathrm{cy}_{k-1})]\} \end{aligned} \tag{24}$$

This can be more concisely written in matrix form as:

$$\hat{\boldsymbol{z}} = \boldsymbol{\Pi}\,\boldsymbol{d} + \boldsymbol{T}\boldsymbol{\Pi}\boldsymbol{e} \tag{25}$$

where $\hat{\boldsymbol{z}} = [\hat{z}_1, \ldots, \hat{z}_n]^{\mathrm{T}}$ is the vector of estimates with size n; $\boldsymbol{d} = [d_0, \ldots, d_m]^{\mathrm{T}}$ is the vector of the unknown applicates of the bilinear surface d, with size $m + 1$ $(m = (\mathrm{mx} + 1) \times (\mathrm{my} + 1) - 1)$; $\boldsymbol{e} = [e_0, \ldots, e_m]^{\mathrm{T}}$ is the vector of the unknown applicates of the bilinear surface e, with size $m + 1$; $\boldsymbol{T}$ is a $n \times n$ diagonal matrix:

$$\boldsymbol{T} = \mathrm{diag}(t(x_1, y_1), \ldots, t(x_n, y_n)) \tag{26}$$

with its elements $t(x_1, y_1), \ldots, t(x_n, y_n)$ being the values of the explanatory variable at the given data points; and $\boldsymbol{\Pi}$ is a matrix with size $n \times (m + 1)$ as defined in equation (8).

In order to incorporate the amount of smoothness of the second bilinear surface e and following the procedure presented in equations (9)–(12), we conclude with the following expressions for the smoothness of the bilinear surface e for the x and y directions:

$$q_{\mathrm{ex}} = \sum_{k=0}^{\mathrm{my}} \sum_{l=1}^{\mathrm{mx}-1} (2e_{l,k} - e_{l-1,k} - e_{l+1,k})^2 \tag{27}$$

and

$$q_{\mathrm{ey}} = \sum_{l=0}^{\mathrm{my}} \sum_{k=1}^{\mathrm{mx}-1} (2e_{k,l} - e_{k-1,l} - e_{k+1,l})^2 \tag{28}$$

In matrix form, equations (27) and (28), along with equations (11) and (12), express the amount of smoothness of the bilinear surfaces, d, e, for the BSSE case, as follows:

$$q_{\mathrm{dx}} = \boldsymbol{d}^{\mathrm{T}}\,\boldsymbol{\Psi}_x^{\mathrm{T}}\,\boldsymbol{\Psi}_x\boldsymbol{d}, \quad q_{\mathrm{dy}} = \boldsymbol{d}^{\mathrm{T}}\,\boldsymbol{\Psi}_y^{\mathrm{T}}\,\boldsymbol{\Psi}_y\boldsymbol{d} \tag{29}$$

$$q_{\mathrm{ex}} = \boldsymbol{e}^{\mathrm{T}}\,\boldsymbol{\Psi}_x^{\mathrm{T}}\,\boldsymbol{\Psi}_x\boldsymbol{e}, \quad q_{\mathrm{ey}} = \boldsymbol{e}^{\mathrm{T}}\,\boldsymbol{\Psi}_y^{\mathrm{T}}\,\boldsymbol{\Psi}_y\boldsymbol{e} \tag{30}$$

where $\boldsymbol{\Psi}_x$ and $\boldsymbol{\Psi}_y$ are matrices with size $(m-1)\times(m+1)$ (for $i = 1,\ldots, m-1$ and $j = 0,\ldots, m$) and ijth entry as in equations (15) and (16), respectively (see Appendix).

Combining equations (2), (25), (29) and (30), and introducing dimensionless multipliers for both x and y directions in order to control the smoothness of the bilinear surfaces, we form the generalized objective function to be minimized:

$$\begin{aligned} f(\boldsymbol{d},\boldsymbol{e}) &:= p + \lambda_x q_{\mathrm{dx}} + \lambda_y q_{\mathrm{dy}} + \mu_x q_{\mathrm{ex}} + \mu_y q_{\mathrm{ey}} \\ &= \|\boldsymbol{z}-\hat{\boldsymbol{z}}\|^2 + \lambda_x \boldsymbol{d}^{\mathrm{T}} \boldsymbol{\Psi}_x^{\mathrm{T}} \boldsymbol{\Psi}_x \boldsymbol{d}, + \lambda_y \boldsymbol{d}^{\mathrm{T}} \boldsymbol{\Psi}_y^{\mathrm{T}} \boldsymbol{\Psi}_y \boldsymbol{d} \\ &\quad + \mu_x \boldsymbol{e}^{\mathrm{T}} \boldsymbol{\Psi}_x^{\mathrm{T}} \boldsymbol{\Psi}_x \boldsymbol{e} + \mu_y \boldsymbol{e}^{\mathrm{T}} \boldsymbol{\Psi}_y^{\mathrm{T}} \boldsymbol{\Psi}_y \boldsymbol{e} \end{aligned} \tag{31}$$

where $\lambda_x \geq 0$ for q_{dx}, $\lambda_y \geq 0$ for q_{dy} and $\mu_x \geq 0$ for q_{ex}, $\mu_y \geq 0$ for q_{ey}.

Differentiation of equation (31) with respect to $\boldsymbol{d}$ and $\boldsymbol{e}$, by applying the typical rules of derivatives involving matrices and equating them to zero, yields:

$$\begin{aligned} \frac{\partial f_1}{\partial \boldsymbol{d}} &= -2\boldsymbol{z}^{\mathrm{T}}\boldsymbol{\Pi} + 2\boldsymbol{d}^{\mathrm{T}}\boldsymbol{\Pi}^{\mathrm{T}}\boldsymbol{\Pi} + 2\boldsymbol{e}^{\mathrm{T}}\boldsymbol{\Pi}^{\mathrm{T}}\boldsymbol{T}^{\mathrm{T}}\boldsymbol{\Pi} \\ &\quad + 2\lambda_x \boldsymbol{d}^{\mathrm{T}}\boldsymbol{\Psi}_x^{\mathrm{T}}\boldsymbol{\Psi}_x + 2\lambda_y \boldsymbol{d}^{\mathrm{T}}\boldsymbol{\Psi}_y^{\mathrm{T}}\boldsymbol{\Psi}_y = 0 \end{aligned} \tag{32}$$

$$\begin{aligned} \frac{\partial f_2}{\partial \boldsymbol{e}} &= -2\boldsymbol{z}^{\mathrm{T}}\boldsymbol{T}\boldsymbol{\Pi} + 2\boldsymbol{d}^{\mathrm{T}}\boldsymbol{\Pi}^{\mathrm{T}}\boldsymbol{T}\boldsymbol{\Pi} + 2\boldsymbol{e}^{\mathrm{T}}\boldsymbol{\Pi}^{\mathrm{T}}\boldsymbol{T}^{\mathrm{T}}\boldsymbol{T}\boldsymbol{\Pi} \\ &\quad + 2\mu_x \boldsymbol{e}^{\mathrm{T}}\boldsymbol{\Psi}_x^{\mathrm{T}}\boldsymbol{\Psi}_x + 2\mu_y \boldsymbol{e}^{\mathrm{T}}\boldsymbol{\Psi}_y^{\mathrm{T}}\boldsymbol{\Psi}_y = 0 \end{aligned} \tag{33}$$

and consequently:

$$\begin{aligned} &[\boldsymbol{\Pi}^{\mathrm{T}}\boldsymbol{\Pi} + \lambda_x\boldsymbol{\Psi}_x^{\mathrm{T}}\boldsymbol{\Psi}_x + \lambda_y\boldsymbol{\Psi}_y^{\mathrm{T}}\boldsymbol{\Psi}_y]\boldsymbol{d} + \boldsymbol{\Pi}^{\mathrm{T}}\boldsymbol{T}\boldsymbol{\Pi}\boldsymbol{e} = \boldsymbol{\Pi}^{\mathrm{T}}\boldsymbol{z}\boldsymbol{\Pi}^{\mathrm{T}} + \boldsymbol{T}\boldsymbol{\Pi}\boldsymbol{d} \\ &\quad + [\boldsymbol{\Pi}^{\mathrm{T}}\boldsymbol{T}^{\mathrm{T}}\boldsymbol{T}\boldsymbol{\Pi} + \mu_x\boldsymbol{\Psi}_x^{\mathrm{T}}\boldsymbol{\Psi}_x + \mu_y\boldsymbol{\Psi}_y^{\mathrm{T}}\boldsymbol{\Psi}_y]\boldsymbol{e} = \boldsymbol{\Pi}^{\mathrm{T}}\boldsymbol{T}^{\mathrm{T}}\boldsymbol{z} \end{aligned} \tag{34}$$

Finally, the solution of the above set of equations that provides the unknown vectors $\boldsymbol{d}$, $\boldsymbol{e}$ that minimize equation (31) is:

$$\begin{aligned} &[\boldsymbol{\Pi}^{\mathrm{T}}\boldsymbol{\Pi} + \lambda_x\boldsymbol{\Psi}_x^{\mathrm{T}}\boldsymbol{\Psi}_x + \lambda_y\boldsymbol{\Psi}_y^{\mathrm{T}}\boldsymbol{\Psi}_y]\boldsymbol{d} + \boldsymbol{\Pi}^{\mathrm{T}}\boldsymbol{T}\boldsymbol{\Pi}\boldsymbol{e} = \boldsymbol{\Pi}^{\mathrm{T}}\boldsymbol{z}\boldsymbol{\Pi}^{\mathrm{T}} + \boldsymbol{T}\boldsymbol{\Pi}\boldsymbol{d} \\ &\quad + [\boldsymbol{\Pi}^{\mathrm{T}}\boldsymbol{T}^{\mathrm{T}}\boldsymbol{T}\boldsymbol{\Pi} + \mu_x\boldsymbol{\Psi}_x^{\mathrm{T}}\boldsymbol{\Psi}_x + \mu_y\boldsymbol{\Psi}_y^{\mathrm{T}}\boldsymbol{\Psi}_y]\boldsymbol{e} = \boldsymbol{\Pi}^{\mathrm{T}}\boldsymbol{T}^{\mathrm{T}}\boldsymbol{z} \end{aligned} \tag{35}$$

The vector of estimates, $\hat{\boldsymbol{z}}$, is obtained from equation (25), once vectors $\boldsymbol{d}$ and $\boldsymbol{e}$ are calculated from equation (35). Also, from equation (23), we can estimate the applicate $\hat{z}$ of any point that lies in the two-dimensional interval $[\mathrm{cx}_0, \mathrm{cx}_{\mathrm{mx}}] \times [\mathrm{cy}_0, \mathrm{cy}_{\mathrm{my}}]$.

We observe that from the four matrices with size $(m+1)\times(m+1)$ appearing in equations (20) and (35), i.e. $\boldsymbol{B} := \boldsymbol{\Pi}^{\mathrm{T}}\boldsymbol{\Pi}$, $\boldsymbol{C} := \boldsymbol{\Psi}_x^{\mathrm{T}}\boldsymbol{\Psi}_x$, $\boldsymbol{D} := \boldsymbol{\Pi}^{\mathrm{T}}\boldsymbol{T}^{\mathrm{T}}\boldsymbol{T}\boldsymbol{\Pi}$ and $\boldsymbol{E} := \boldsymbol{\Psi}_y^{\mathrm{T}}\boldsymbol{\Psi}_y$; $\boldsymbol{B}$ and $\boldsymbol{D}$ are symmetric block tridiagonal while $\boldsymbol{C}$ and $\boldsymbol{E}$ are block diagonal matrices. Furthermore, matrices $\boldsymbol{C}$ and $\boldsymbol{E}$ are always singular; however, when $\lambda_x, \mu_x > 0$ or $\lambda_y, \mu_y > 0$, the sums $\boldsymbol{B} + \lambda_x\boldsymbol{C} + \lambda_y\boldsymbol{E}$ and $\boldsymbol{D} + \mu_x\boldsymbol{C} + \mu_y\boldsymbol{E}$ are non-singular and thus their inverses exist.

Choice of parameters

Transformation of smoothing parameters

It is apparent that the number of the adjustable parameters for each of the two above presented versions of the methodology consists of the numbers of intervals, mx, my, and the smoothing parameters for the x, y directions.

Therefore, for the case of the bilinear surfaces interpolation (BSS) there are four adjustable parameters: the numbers of intervals, mx, my, and the smoothing parameters λ_x and λ_y corresponding to vector $\boldsymbol{d}$. The incorporation of the explanatory variable, for the BSSE case, adds two more adjustable parameters: the smoothing parameters μ_x and μ_y corresponding to vector $\boldsymbol{e}$.

The choice of parameters can be done by using an efficient, but standard, objective way as described by the following analysis.

A convenient search of the smoothing parameters, in terms of computational time, can be achieved by transforming λ and μ in terms of tension parameters τ_λ and τ_μ, whose values are restricted in the interval [0, 1), for both directions. The formulation is based on the expressions presented by Koutsoyiannis (2000), as well as Malamos and Koutsoyiannis (2014), and was established after a numerical investigation of the method on several examples. The proposed equations have the form:

$$\lambda_x = \left(10^{\varepsilon} m \frac{\log \tau_{\mathrm{m}}}{\log \tau_{\lambda x}}\right)^{\kappa_\lambda}, \quad \lambda_y = \left(10^{\varepsilon} m \frac{\log \tau_{\mathrm{m}}}{\log \tau_{\lambda y}}\right)^{\kappa_\lambda} \tag{36}$$

for the BSS case, while for BSSE the extra smoothing parameters μ_x and μ_y are set to:

$$\mu_x = \left(10^{\theta} m \frac{\log \tau_{\mathrm{m}}}{\log \tau_{\mu x}}\right)^{\kappa_\mu}, \quad \mu_y = \left(10^{\theta} m \frac{\log \tau_{\mathrm{m}}}{\log \tau_{\mu y}}\right)^{\kappa_\mu} \tag{37}$$

where $\tau_{\mathrm{m}} = 0.99$ is the maximum allowed tension, corresponding to the upper bound of λ and μ, set for numerical stability equal to:

$$\lambda_{\mathrm{m}} = \frac{\mathrm{trace}(\boldsymbol{B})}{\mathrm{trace}(\boldsymbol{C}+\boldsymbol{E})} 10^9, \mu_{\mathrm{m}} = \frac{\mathrm{trace}(\boldsymbol{D})}{\mathrm{trace}(\boldsymbol{C}+\boldsymbol{E})} 10^9 \tag{38}$$

The exponents κ_λ, κ_μ in equations (36) and (37) are determined by the relations:

$$\begin{aligned} \kappa_\lambda &= \frac{\log \lambda_{\mathrm{m}}}{\log(10^{\varepsilon} m)}, \quad \kappa_\mu = \frac{\log \mu_{\mathrm{m}}}{\log(10^{\theta} m)}, \\ m &= (\mathrm{mx}+1)(\mathrm{my}+1) - 1 \end{aligned} \tag{39}$$

which are obtained by combining equations (36) or (37) with equation (38). The exponents ε, θ in equations (36), (37) and (39) are set to:

$$\varepsilon = \max(1, \lfloor \log[\mathrm{trace}(\boldsymbol{B})] \rfloor) \tag{40}$$

and

$$\theta = \max(1, \lfloor \log[\mathrm{trace}(\boldsymbol{D})] \rfloor) \tag{41}$$

with $\varepsilon, \theta \in \mathbb{Z}^+$. The minimum allowed value of λ_x, λ_y, μ_x, μ_y is 0.

Estimation of smoothing parameters

Combining equations (7) and (20) for the BSS case, we obtain:

$$\hat{\boldsymbol{z}} = \boldsymbol{A}\boldsymbol{z} \tag{42}$$

where $\boldsymbol{A}$ is a $n \times n$ symmetric and positive-definite matrix given by:

$$A = \Pi\left(\Pi^{\mathrm{T}}\Pi + \lambda_x \Psi_x^{\mathrm{T}}\Psi_x + \lambda_y \Psi_y^{\mathrm{T}}\Psi_y\right)^{-1}\Pi^{\mathrm{T}} \tag{43}$$

while combining equations (25) and (35) for the case with explanatory variable (BSSE), we obtain the same relationship as equation (42) with A being a $n \times n$ symmetric and positive-definite matrix now given by:

$$A = \Pi T \Pi \begin{bmatrix} \Pi^{\mathrm{T}}\Pi + \lambda_x \Psi_x^{\mathrm{T}}\Psi_x + \lambda_y \Psi_y^{\mathrm{T}}\Psi_y & \Pi^T T \Pi \\ \Pi^{\mathrm{T}} T \Pi & \Pi^T T^T T \Pi + \mu_x \Psi_x^{\mathrm{T}}\Psi_x + \mu_y \Psi_y^{\mathrm{T}}\Psi_y \end{bmatrix}^{-1} (\Pi T \Pi)^{\mathrm{T}} \tag{44}$$

while equation (45) is used for estimating GCV for the BSSE method.

Based on the above presented analysis, for a given combination of segments mx, my, the minimization of GCV results in the optimum values of $\tau_{\lambda x}$, $\tau_{\lambda y}$ and $\tau_{\mu x}$, $\tau_{\mu y}$. This can be repeated for several trial combinations of mx, my values, until the global minimum of GCV is reached.

equations (43) and (44) depend on all adjustable parameters: mx, my, $\tau_{\lambda x}$, $\tau_{\lambda y}$ and $\tau_{\mu x}$, $\tau_{\mu y}$.

The parameter estimation is based on the generalized cross-validation (Craven and Wahba 1978) methodology, defined by:

$$\mathrm{GCV} = \frac{\frac{1}{n}\|(I - A)z\|^2}{\left[\frac{1}{n}\mathrm{trace}(I - A)\right]^2} \tag{45}$$

where matrix A is called the "influence" or "smoother" matrix, while the quantity:

$$\mathrm{trace}(I - A) \tag{46}$$

in the denominator of equation (45) describes the "residual degrees of freedom" of the fitted smoother used by nonparametric regression methods (Buja *et al.* 1989, Wahba 1990, Carmack *et al.* 2012).

Based on the literature, there are two alternative definitions for residual degrees of freedom under independence in the context of symmetric linear smoothers, namely:

$$\mathrm{trace}\left(I - AA^{\mathrm{T}}\right) \tag{47}$$

$$\mathrm{trace}\left[I - \left(2A - AA^{\mathrm{T}}\right)\right] \tag{48}$$

with $0 \le \mathrm{trace}[I - (2A - AA^{\mathrm{T}})] \le \mathrm{trace}(I - A) \le \mathrm{trace}(I - AA^{\mathrm{T}}) \le n$ (Buja *et al.* 1989, Carmack *et al.* 2012).

For exploration purposes, we analysed the methods' performance against all three definitions. The results showed that when matrix A is defined by equation (43), the best results were obtained when the residual degrees of freedom were defined by equation (48). However, when matrix A is defined by equation (44), the best results were obtained when the residual degrees of freedom were defined by equation (46), which is the standard definition of the generalized cross-validation, as already presented in equation (45). The degrees of freedom definition presented by equation (47) did not perform as well as the previously mentioned expressions and thus it was excluded from the methods' implementation.

Consequently, the relation that provides GCV for the BSS method is:

$$\mathrm{GCV} = \frac{\frac{1}{n}\|(I - A)z\|^2}{\left[\frac{1}{n}\mathrm{trace}\left[I - \left(2A - AA^{\mathrm{T}}\right)\right]\right]^2} \tag{49}$$

Computational implementation

In similar applications presented earlier by Koutsoyiannis (2000) and Malamos and Koutsoyiannis (2014), the implementation of the computational framework was made in Microsoft Excel, since it provides a direct means of data visualization and graphical exploration.

Since the block matrices involved in the systems of equations (20) and (35) have dimensions $(m + 1) \times (m + 1)$ and $(2m + 2) \times (2m + 2)$, respectively, a considerable computational effort, which could not be satisfied from Microsoft Excel alone, was required. This was tackled by the development of a dynamic link library in Object Pascal (Delphi) programming language, which was linked to Microsoft Excel.

In this context, an Excel array formula acts as the main interface, with its arguments being the available points' values and coordinates along with the unknown points' coordinates, the number of points on the x and y axis that form the bilinear surfaces and the smoothing parameter values.

The dynamic link library performs the following tasks:

(1) constructs the matrices involved in the systems of equations (20) or (35) depending on which of the two versions of the methodology is implemented;
(2) solves the system of equations, for each case, by implementing the "Cholesky decomposition", thus decomposing the symmetric and positive-definite matrix A into a lower triangular matrix whose transpose can itself serve as the upper triangular part. This method is about a factor 2 faster than a "LU decomposition" of A, where its symmetry is ignored (Press *et al.* 2002);
(3) finds the inverse matrices involved in equations (43) and (44) by a straightforward procedure based on the above mentioned "Cholesky decomposition" method (Press *et al.* 2002); and
(4) returns to Microsoft Excel, apart from the solution of the systems of equations (20) and (35), information concerning the above presented numerical procedure, such as the matrices B, C, D and E, along with the GCV and mean square estimation error. The latter is acquired from the numerators of equations (45) and (49).

Results and comments

The BSS and BSSE methods, with the mathematical formulation described in the previous sections, were derived from extending, in two dimensions, the broken line smoothing method described by Koutsoyiannis (2000) and the broken line smoothing with explanatory variable described by Malamos and Koutsoyiannis (2014).

The main difference between bilinear surface smoothing methods and other known interpolation methods is the introduction of the smoothness terms $\boldsymbol{\Psi}_x^{\mathrm{T}}\boldsymbol{\Psi}_x$ and $\boldsymbol{\Psi}_y^{\mathrm{T}}\boldsymbol{\Psi}_y$ in the corresponding problem formulation. Those terms control the overall smoothness of the bilinear surface through adjustable parameters according to the x or y direction.

It should be obvious from the above discourse that bilinear surface smoothing methods do not require linearity between the involved variables, namely x, y, z and the explanatory variable t, but two-dimensional local bilinearity is incorporated in the mathematical framework in a bilinear surface approach. Also, the functional dependence, in terms of vectors $\boldsymbol{d}$, $\boldsymbol{e}$, the number of segments, mx and my, and the tension parameters, is neither constant nor *a priori* known, but in each case is determined through the procedure of minimizing the generalized cross-validation (GCV).

Both implementations of bilinear surface smoothing require the minimization of generalized objective functions with respect to the total square error and the surface smoothness. The formulation of equation (42) allows the adaptation of the generalized cross-validation from the splines theory, allowing a standard and objective way to estimate the smoothness parameters and the number of bilinear surfaces involved in the interpolation procedure.

According to the classification presented by Li and Heap (2008), BSS and BSSE have the following features:

(1) They are both *local* and *global*. Their locality stems from the fact that they use the four surrounding points of the corresponding bilinear surface to derive the estimation of the included data point (Fig. 2). On the other hand, they are also global since they implement the GCV procedure to globally fit the consecutive bilinear surfaces to the available data points.
(2) They can be either *exact* or *inexact*. Specifically, they are able to generate an estimate that is the same as the observed value at a sampled point (exactness) if the minimum values of the smoothing parameters are used. On the other hand, when the GCV procedure is implemented along with strong smoothing, they are inexact.
(3) They are *stochastic* since the proposed mathematical framework, apart from estimations, also provides direct means of evaluating interpolation errors across the available data points from the numerators of equations (45) and (49), as already presented in the one-dimensional implementation (Malamos and Koutsoyiannis 2014).
(4) The surfaces that they produce can be either *gradual* or *abrupt* depending on the magnitude of the smoothing parameters, e.g. if their values are close to 1, the resulting surface will be smooth while the opposite will occur if their values are close to the lower limit. Also, the numbers of bilinear surfaces along the x and/or y directions, i.e. mx and my, contribute to the overall surface smoothness, thus acting as additional smoothing parameters. This derives also from the one-dimensional implementations (Koutsoyiannis 2000, Malamos and Koutsoyiannis 2014), where increased numbers of broken lines segments were associated with small values of the smoothing parameters.
(5) BSS is *univariate* since it implements only the primary variable in deriving the estimation, while BSSE is *multivariate* since it incorporates an explanatory variable available at a considerably denser dataset in the interpolation procedure.
(6) Both BSS and BSSE implement a regular grid, but this does not have to be square necessarily since the number of bilinear surfaces along the x direction does not have to coincide with the number of bilinear surfaces along the y direction.

Conclusions

A nonparametric innovative mathematical framework which can be utilized to perform various interpolation tasks is described. The technique incorporates smoothing terms with adjustable weights, defined by means of the angles formed by the consecutive bilinear surfaces into a piecewise surface regression model with known break points. The incorporation, in an objective manner, of an explanatory variable, available from measurements at a considerably denser dataset than the initial main variable, is presented in terms of an alternative implementation of the main methodology.

A notable property of the proposed framework is the fact that the resolution (number of consecutive bilinear surfaces) does not necessarily have to coincide with that of the given data points, but it can be either finer or coarser, depending on the specific requirements of the problem of interest. This is an important property which makes the method applicable and reliable even in the case of scarce datasets.

The proposed mathematical framework follows a parsimonious approach for fulfilling spatial interpolation tasks, without the need to make many decisions on parameters or complex concepts. Likewise, the computational implementation offers an almost automated procedure in achieving the final results.

Further research can be focused towards the incorporation of alternative techniques for acquiring the global minimum value of GCV, providing means for faster convergence to the optimal solution.

Application of the method in hydrological problems is given in a companion paper (Malamos and Koutsoyiannis 2015) along with comparisons to established interpolation methods.

Acknowledgements

We thank Associate Editor Alin Carsteanu, the reviewer Efraín Domínguez and an anonymous reviewer for their comments and suggestions which helped us to improve our paper.

Disclosure statement

No potential conflict of interest was reported by the authors.

ORCID

Nikolaos Malamos http://orcid.org/0000-0002-5292-2870
Demetris Koutsoyiannis http://orcid.org/0000-0002-6226-0241

References

Buja, A., Hastie, T., and Tibshirani, R., 1989. Linear smoothers and additive models. *The Annals of Statistics*, 17 (2), 453–510. doi:10.1214/aos/1176347115

Burrough, P.A. and McDonnell, R.A., 1998. *Principles of geographical information systems*. Oxford: Oxford University Press, 333pp.

Carmack, P.S., Spence, J.S., and Schucany, W.R., 2012. Generalised correlated cross-validation. *Journal of Nonparametric Statistics*, 24 (2), 269–282. doi:10.1080/10485252.2012.655733

Craven, P. and Wahba, G., 1978. Smoothing noisy data with spline functions. *Numerische Mathematik*, 31 (4), 377–403. doi:10.1007/BF01404567

Davis, J.C., 1986. *Statistics and data analysis in geology*. 2nd ed. New York: John Wiley & Sons.

Goovaerts, P., 1997. *Geostatistics for natural resources evaluation*. New York: Oxford University Press, 483pp.

Hengl, T., Heuvelink, G.B.M., and Rossiter, D.G., 2007. About regression-kriging: from equations to case studies. *Computers & Geosciences*, 33 (10), 1301–1315. doi:10.1016/j.cageo.2007.05.001

Koutsoyiannis, D., 2000. Broken line smoothing: a simple method for interpolating and smoothing data series. *Environmental Modelling & Software*, 15 (2), 139–149. doi:10.1016/S1364-8152(99)00026-2

Li, J. and Heap, A.D., 2008. *A review of spatial interpolation methods for environmental scientists*. Canberra, ACT: Geoscience Australia.

Malamos, N. and Koutsoyiannis, D., 2014. Broken line smoothing for data series interpolation by incorporating an explanatory variable with denser observations: application to soil-water and rainfall data. *Hydrological Sciences Journal*. doi:10.1080/02626667.2014.899703

Malamos, N. and Koutsoyiannis, D., 2015. Bilinear surface smoothing for spatial interpolation with optional incorporation of an explanatory variable. Part 2: Application to synthesized and rainfall data. *Hydrological Sciences Journal*. doi:10.1080/02626667.2015.1080826

Press, W.H., *et al.*, 2002. *Numerical recipes in C*. 2nd ed. Cambridge: Cambridge University Press.

Sauquet, E., Gottschalk, L., and Leblois, E., 2000. Mapping average annual runoff: a hierarchical approach applying a stochastic interpolation scheme. *Hydrological Sciences Journal*, 45 (6), 799–815. doi:10.1080/02626660009492385

Wahba, G., 1990. Spline models for observational data. *Regional Conference Series in Applied Mathematics 59*. Philadelphia, PA, SIAM.

Wahba, G. and Wendelberger, J., 1980. Some new mathematical methods for variational objective analysis using splines and cross validation. *Monthly Weather Review*, 108 (8), 1122–1143. doi:10.1175/1520-0493(1980)108<1122:SNMMFV>2.0.CO;2

Wang, H., Liu, G., and Gong, P., 2005. Use of cokriging to improve estimates of soil salt solute spatial distribution in the Yellow River delta. *Acta Geographica Sinica*, 60 (3), 511–518.

Appendix

$\boldsymbol{\Psi}_x$ and $\boldsymbol{\Psi}_y$ matrix definition

If we apply equation (11) for the general case where $l = 1, \ldots, \mathrm{mx} - 1$ and $k = 0, \ldots, \mathrm{my}$, we obtain equation (A1):

$$\begin{aligned} q_{\mathrm{dx}} = & \left(2d_{1,0} - d_{0,0} - d_{2,0}\right)^2 + \left(2d_{2,0} - d_{1,0} - d_{3,0}\right)^2 + \cdots + \left(2d_{\mathrm{mx}-1,0} - d_{\mathrm{mx}-2,0} - d_{\mathrm{mx},0}\right)^2 \\ & + \left(2d_{1,1} - d_{0,1} - d_{2,1}\right)^2 + \left(2d_{2,1} - d_{1,1} - d_{3,1}\right)^2 + \cdots + \left(2d_{\mathrm{mx}-1,1} - d_{\mathrm{mx}-2,1} - d_{\mathrm{mx},1}\right)^2 \\ & + \\ & \vdots \\ & + \\ & + \left(2d_{1,\mathrm{my}} - d_{0,\mathrm{my}} - d_{2,\mathrm{my}}\right)^2 + \left(2d_{2,\mathrm{my}} - d_{1,\mathrm{my}} - d_{3,\mathrm{my}}\right)^2 + \cdots + \left(2d_{\mathrm{mx}-1,\mathrm{my}} - d_{\mathrm{mx}-2,\mathrm{my}} - d_{\mathrm{mx},\mathrm{my}}\right)^2 \end{aligned} \quad \text{(A1)}$$

which can easily be expressed in matrix form as follows:

$$q_{\mathrm{dx}} = (\boldsymbol{\Psi}_x \boldsymbol{d})^{\mathrm{T}} (\boldsymbol{\Psi}_x \boldsymbol{d}) = \boldsymbol{d}^{\mathrm{T}} \boldsymbol{\Psi}_x^{\mathrm{T}} \boldsymbol{\Psi}_x \boldsymbol{d} \quad \text{(A2)}$$

where $\boldsymbol{\Psi}_x\, \boldsymbol{d}$ is a vector of $(m - 1)$ elements and has the form:

$$\boldsymbol{\Psi}_x \boldsymbol{d} = \begin{bmatrix} 2d_{1,0} - d_{0,0} - d_{2,0} \\ 2d_{2,0} - d_{1,0} - d_{3,0} \\ \vdots \\ 2d_{\mathrm{mx}-1,0} - d_{\mathrm{mx}-2,0} - d_{\mathrm{mx},0} \\ 2d_{1,1} - d_{0,1} - d_{2,1} \\ 2d_{2,1} - d_{1,1} - d_{3,1} \\ \vdots \\ 2d_{\mathrm{mx}-1,1} - d_{\mathrm{mx}-2,1} - d_{\mathrm{mx},1} \\ 2d_{1,\mathrm{my}} - d_{0,\mathrm{my}} - d_{2,\mathrm{my}} \\ 2d_{2,\mathrm{my}} - d_{1,\mathrm{my}} - d_{3,\mathrm{my}} \\ \vdots \\ 2d_{\mathrm{mx}-1,\mathrm{my}} - d_{\mathrm{mx}-2,\mathrm{my}} - d_{\mathrm{mx},\mathrm{my}} \end{bmatrix} \quad \text{(A3)}$$

From equation (A3) can easily be derived that $\boldsymbol{\Psi}_x$ is a matrix with size $(m - 1) \times (m + 1)$ (for $i = 1, \ldots, m - 1$ and $j = 0, \ldots, m$) and ijth entry:

$$\boldsymbol{\Psi}_{x,\,i,j} = \begin{cases} 2, & \text{when } i = j \quad \text{and } i - k(\mathrm{mx}+1) \notin \{1, \mathrm{mx}+1\} \\ -1, & \text{when } |i - j| = 1 \quad \text{and } i - k(\mathrm{mx}+1) \notin \{1, \mathrm{mx}+1\} \\ 0, & \text{otherwise} \end{cases} \quad \text{(A4)}$$

where $k = 0, \ldots, \mathrm{my}$.

By following an equivalent procedure to the above presented, we concluded with the following expression for the smoothness of the bilinear surface according to the y direction:

$$q_{\mathrm{dy}} = \left(\boldsymbol{\Psi}_y \boldsymbol{d}\right)^{\mathrm{T}} \left(\boldsymbol{\Psi}_y \boldsymbol{d}\right) = \boldsymbol{d}^{\mathrm{T}} \boldsymbol{\Psi}_y^{\mathrm{T}} \boldsymbol{\Psi}_y \boldsymbol{d} \quad \text{(A5)}$$

where $\boldsymbol{\Psi}_y$ is a matrix with size $(m - 1) \times (m + 1)$ (for $i = 1, \ldots, m - 1$ and $j = 0, \ldots, m$) and ijth entry:

$$\boldsymbol{\Psi}_{y\ i,j} = \begin{cases} 2, & \text{when } i = j \quad \text{and } i - l(\mathrm{my}+1) \notin \{1, \mathrm{my}+1\} \\ -1, & \text{when } |i - j| = 1 \quad \text{and } i - l(\mathrm{my}+1) \notin \{1, \mathrm{my}+1\} \\ 0, & \text{otherwise} \end{cases} \quad \text{(A6)}$$

where $l = 0, \ldots, \mathrm{mx}$. We note that matrices $\boldsymbol{\Psi}_x$ and $\boldsymbol{\Psi}_y$ are identical when $\mathrm{mx} = \mathrm{my}$.