Constructing Cartesian Splines

H.R.N. van Erp[*,1], R.O. Linger[2] and P.H.A.J.M. van Gelder[1]

[1]*TU Delft Structural Hydraulic and Probabilistic Design, TU Delft, Delft, The Netherlands*

[2]*Medical Sciences, RuG, Groningen, The Netherlands*

Abstract: We introduce here Cartesian splines or, for short, C-splines. C-splines are piecewise polynomials which are defined on adjacent Cartesian coordinate systems and are C^r continuous throughout. The C^r continuity is enforced by constraining the coefficients of the polynomial to lie in the null-space of some smoothness matrix H. The matrix-product of the null-space of the smoothness matrix H and the original polynomial base results in a new base, the so-called C-spline base, which automatically enforces C^r continuity throughout. In this article we give a derivation of this C-spline base as well as an algorithm to construct C-spline models.

Keywords: Splines, cartesian, regression analysis, explicit base.

1. INTRODUCTION

We introduce here Cartesian splines or, for short, C-splines. C-splines are piecewise polynomials which are defined on adjacent Cartesian coordinate systems and are C^r continuous throughout. The C^r continuity is enforced by constraining the coefficients of the polynomial to lie in the null-space of some smoothness matrix H. The matrix-product of the null-space of the smoothness matrix H and the original polynomial base results in a new base, the so-called C-spline base, which automatically enforces C^r continuity throughout. The idea of using the null-space of some smoothness matrix H has been taken from the B-spline literature, where piecewise polynomials are defined on adjacent triangular Barycentric coordinate systems [1-3]. It turns out that C-spline bases have a particular simple form. This makes it possible to give an explicit formulation of general C-spline bases. In this article we will give a general outline how to enforce continuity constraints by way of the smoothness matrix H. We then show how these constraints lead us to the C-spline base. Then we will give the explicit algorithm for constructing a bivariate C-spline base and show how to use this base to construct a C-spline model.

2. PIECEWISE POLYNOMIALS

We start with the bivariate Cartesian x, y-coordinate system. We partition this initial coordinate system with origin $O = (0,0)$ in two adjacent coordinate systems, each with its own origin, $O = (0,0)$ and $\tilde{O} = (0,0)$. The geometry in terms of x and y may be depicted as:
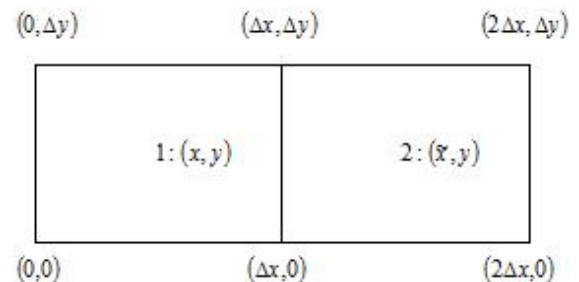
*Address correspondence to this author at the Structural Hydraulic and Probabilistic Design, TU Delft, Delft, The Netherlands; Tel: 06-30883099; E-mail: H.R.N.vanErp@tudelft.nl

Fig. (1). Geometry of the domain of two piecewise polynomials in terms of x and y..

where Δx and Δy are some constants. Likewise, the geometry in terms of $\tilde{x}$ and y may be depicted as:
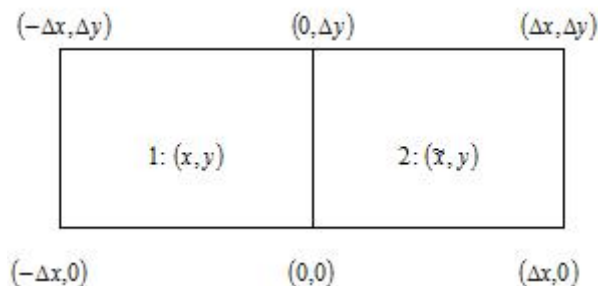

Fig. (2). Geometry of the domain of two piecewise polynomials in terms of $\tilde{x}$ and y.

where Δx and Δy are the same constants as used in Fig. (**1**).

Now, we may define on both coordinate systems a polynomial of order d:

$$P_d(x,y) = \sum_{0 \le p+q \le d} c_{pq} x^p y^q \tag{1}$$

We start with the most simple case, that is, we set $d = 1$. The polynomial equations for both coordinate systems then become:

$$z_1(x,y) = c_{11} + c_{12}x + c_{13}y, \ 0 \le x \le \Delta x, \ 0 \le y \le \Delta y$$
$$z_2(\tilde{x},y) = c_{21} + c_{22}\tilde{x} + c_{23}y, \ 0 \le \tilde{x} \le \Delta x, \ 0 \le y \le \Delta y \quad (2)$$

If we look at Fig. (**2**), we see that

$$\tilde{x} = x - \Delta x \quad (3)$$

Combining (2) and (3) we get:

$$z_1(x,y) = c_{11} + c_{12}x + c_{13}y, \ 0 \le x \le \Delta x, \quad 0 \le y \le \Delta y$$
$$z_2(x,y) = c_{21} + c_{22}(x - \Delta x) + c_{23}y, \ \Delta x \le x \le 2\Delta x, \ 0 \le y \le \Delta y \quad (4)$$

Let

$$\mathbf{z} = \left(\ z_1(x,y) \quad z_2(x,y) \ \right)^T$$

be the outcome vector. Then (4) may be rewritten as the matrix-vector product of the polynomial base

$$B = \begin{pmatrix} 1 & x & y & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x - \Delta x & y \end{pmatrix} \quad (5)$$

and the coefficient vector

$$\mathbf{c} = \left(\ c_{11} \quad c_{12} \quad c_{13} \quad c_{21} \quad c_{22} \quad c_{23} \ \right)^T \quad (6)$$

that is,

$$\mathbf{z} = B\mathbf{c}$$

Note that the (x,y)-values that fall in the first quadrant of Fig. (**1**) are assigned to the first row of the polynomial base B, while (x,y)-values in the second quadrant are assigned to the second row.

3. ENFORCING ZEROTH ORDER CONTINUITY

In order for the two polynomials (4) to connect at the boundary, that is, in order to have C^0 continuity, we must have that

$$z_1(\Delta x, y) = z_2(\Delta x, y) \quad (7)$$

for any y. Substituting (4) in (7), we find

$$c_{11} + c_{12}\Delta x + c_{13}y = c_{21} + c_{23}y$$

or, equivalently,

$$c_{11} + c_{12}\Delta x + c_{13}y - c_{21} - c_{23}y = 0 \quad (8)$$

We have that (8) is a constraint on the $\mathbf{c}$ coefficients. The coefficients $\mathbf{c}$ must all lie in the null-space of the smoothness ``matrix'' H, where

$$H = \left(\ 1 \quad \Delta x \quad y \quad -1 \quad 0 \quad -1 \ \right) \quad (9)$$

The null-space [4], of H is

$$H_0 = \begin{pmatrix} y & 0 & 1 & -y & -\Delta x \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (10)$$

and it may be checked that

$$HH_0 = \mathbf{0}$$

where $\mathbf{0}$ is the 1×5 zero vector. It follows that the matrix product of H with any linear combination of the columns in H_0 must give a zero value, that is,

$$HH_0\mathbf{c}_0 = 0$$

where $\mathbf{c}_0$ is an arbitrary 5×1 vector. Stated differently, any linear combination of the columns of H_0 gives us an 6×1 vector that satisfies the constraint (7) or, equivalently, constraint (8).

Now, if we take the matrix product of our original polynomial base, B, and the null-space of our smoothness matrix, H_0, we get the null-base B_0:

$$B_0 = BH_0$$

$$= \begin{pmatrix} y & 0 & 1 & 0 & x - \Delta x \\ y & x - \Delta x & 1 & 0 & 0 \end{pmatrix} \quad (11)$$

If we drop the zero column in (11) and rearrange the columns somewhat, we get the C-spline base, B_C:

$$B_C = \begin{pmatrix} 1 & y & x - \Delta x & 0 \\ 1 & y & 0 & x - \Delta x \end{pmatrix} \quad (12)$$

Let

$$\mathbf{b} = \left(\ b_1 \quad b_2 \quad b_3 \quad b_4 \ \right)^T$$

be an arbitrary coefficient vector. Then

$$\mathbf{z} = B_C\mathbf{b}$$

corresponds with the polynomial equations

$$z_1(x,y) = b_1 + b_2 y + b_3(x - \Delta x), \ 0 \le x \le \Delta x, \quad 0 \le y \le \Delta y$$
$$z_2(x,y) = b_1 + b_2 y + b_4(x - \Delta x), \ \Delta x \le x \le 2\Delta x, \ 0 \le y \le \Delta y \quad (13)$$

Now, if we substitute $x = \Delta x$ in (13) we have that for any choice of $\mathbf{b}$ constraint (7) is satisfied:

$$z_1(\Delta x, y) = z_2(\Delta x, y) = b_1 + b_2 y \quad (14)$$

It follows that B_C, (12), is the base that enforces zeroth order continuity.

We summarize, C^0 continuity between two piecewise polynomials results in a smoothness matrix H, (9). The coefficients $\mathbf{c}$, (6), defined on the original polynomial base B, (5), are constrained to lie within the null-space of this smoothness matrix. Stated differently, the coefficients $\mathbf{c}$ are constrained to be a linear combination of the columns of H_0, (10), which span the null space of H. By directly multiplying the null-space matrix H_0 with the the original polynomial base B we get the null-base B_0 [4], which contains redundant columns consisting of zero vectors. Dropping these zero vectors we obtain the C-spline base B_C,

(12), which has the C^0 constraint (7) build into its structure, as may be checked, (14).

4. ENFORCING FIRST ORDER CONTINUITY

In order for the partial derivatives of the two polynomials (4) to connect at the boundary, that is, in order to have C^1 continuity, we must have that the partial derivatives $\partial z_1 / \partial x$ and $\partial z_2 / \partial x$ are C^0 at their boundaries, that is

$$\left.\frac{\partial z_1(x,y)}{\partial x}\right|_{x=\Delta x} = \left.\frac{\partial z_2(x,y)}{\partial x}\right|_{x=\Delta x} \tag{15}$$

Substituting (4) in (15), we find

$$c_{12} = c_{22}$$

or, equivalently,

$$c_{12} - c_{22} = 0 \tag{16}$$

Adding constraint (16) to (9), the new smoothness matrix H and the corresponding null space H_0 become, respectively,

$$H = \begin{pmatrix} 1 & \Delta x & y & -1 & 0 & -1 \\ 0 & 1 & 0 & 0 & -1 & 0 \end{pmatrix}$$

and [4],

$$H_0 = \begin{pmatrix} y & -\Delta x & 1 & -y \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \tag{17}$$

Multiplying (17) with the original polynomial base (5) we get

$$B_0 = B H_0$$

$$= \begin{pmatrix} y & x - \Delta x & 1 & 0 \\ y & x - \Delta x & 1 & 0 \end{pmatrix}$$

Dropping the redundant zero column and rearranging the columns somewhat, we get the C-spline base:

$$B_C = \begin{pmatrix} 1 & y & x - \Delta x \\ 1 & y & x - \Delta x \end{pmatrix} \tag{18}$$

which, since Δx is a constant, is equivalent to the base

$$B_C = \begin{pmatrix} 1 & y & x \\ 1 & y & x \end{pmatrix} \tag{19}$$

From C-spline-base (19) it can be seen that the first order piecewise polynomials which have first order partial derivatives everywhere collapse to a global polynomial of

order $d = 1$ and C^1, which is just the base of a linear regression model having an intercept and predictors x and y.

Here the given framework for deriving C-spline bases may be generalized to d th order piecewise polynomials with r th order continuity, $0 \le r \le d$, on arbitrary geometries. If one does this then it is found that the C-spline base, B_C, has a relatively simple structure. This simple structure allows us to directly construct B_C without first having to compute the null matrix H_0 and then taking its matrix product with the original base B. This makes C-spline modeling, as given in the next section, computationally efficient. It will be seen that the computational burden of constructing a C-spline is equivalent to that of performing an ordinary regression analysis.

5. AN ALGORITHM TO CONSTRUCT C-SPLINES

We give here the algorithm for the construction of C-splines for bivariate geometries, partitioned into $I \times J$ adjacent Cartesian domains on which d th order piecewise polynomials with r th order continuity everywhere are defined.

5.1. The Geometry

First we define a partitioning of the Cartesian (x, y) plane. Then we number the resulting partitionings. In the region of interest the x values take on values from a_x to b_x and the y values take on values from a_y to b_y. If we partition the x-axis in I adjacent axes with equal lengths Δx and the y-axis in J adjacent axes with equal lengths Δy. Then this results in $K = IJ$ partitionings.

Now, we may number each partitioning in the following manner. For $i = 1$ we number the partitionings of the y-axis from $k = 1, \ldots, J$, for $i = 2$ we number the partitionings of the y-axis from $k = J + 1, \ldots, 2J$, etc… We then have that the (i, j) th partitioning is numbered as

$$k = (i-1)J + j, \qquad 1 \le i \le I, \qquad 1 \le j \le J \tag{20}$$

where $k = 1, \ldots, K$ and $K = IJ$.

In the next paragraph we will construct our C-spline base. The geometry, as given in (20), is non-trivial in that the Cartesian coordinate system having coordinates (i, j) corresponds with the k th row of this C-spline base.

5.2. Constructing the C-Spline Base

First we construct the building blocks of our base. Let

$$u_i = \begin{cases} (x - a_x) - i\Delta x, & i = 1, \ldots, I-1 \\ (x - a_x) - (i-1)\Delta x, & i = I \end{cases} \tag{21}$$

Then the x-columns of the building blocks are:

$$u_{ki} = \begin{cases} u_i, & k = 1,\ldots,iJ, \\ 0, & else \end{cases} \quad \{\ i = 1,\ldots,I \qquad (22)$$

where $k = 1,\ldots,K$ and $K = IJ$. Likewise, let

$$v_j = \begin{cases} (y - a_y) - j\Delta y, & j = 1,\ldots,J-1 \\ (y - a_y) - (j-1)\Delta y, & j = J \end{cases} \qquad (23)$$

Then the y-columns of the building blocks are:

$$v_{kj} = \begin{cases} v_j, & k = 1,\ldots,j + (i-1)\times J, \\ 0, & else \end{cases} \quad \begin{cases} j = 1,\ldots,J \\ i = 1,\ldots,I \end{cases} \quad (24)$$

where $k = 1,\ldots,K$ and $K = IJ$.

Using the building blocks (22) and (24), we may now construct the C-spline base B_C. Our polynomial is of order d, that is, let p and q be the powers of x and y, respectively, then $0 \le p + q \le d$. Let

$$U_p = \begin{cases} u_{kI}^p, & p \le r \\ \{u_{k1}^p,\ldots,u_{kI}^p\}, & p > r \end{cases} \qquad (25)$$

$$V_q = \begin{cases} v_{kJ}^q, & q \le r \\ \{v_{k1}^q,\ldots,v_{kJ}^q\}, & q > r \end{cases} \qquad (26)$$

Then we take the outer product of U_p and V_q to get $B_{p,q}$, the C-spline equivalent of the polynomial term $x^p y^q$:

$$B_{p,q} = U_p \otimes V_q = \begin{cases} u_{kI}^p v_{kJ}^q, & p \le r, q \le r \\ \{u_{kI}^p v_{k1}^q, u_{kI}^p v_{k2}^q,\ldots,u_{kI}^p v_{kJ}^q\}, & p \le r, q > r \\ \{u_{k1}^p v_{kJ}^q, u_{k2}^p v_{kJ}^q,\ldots,u_{kI}^p v_{kJ}^q\}, & p > r, q \le r \\ \{u_{k1}^p v_{k1}^q, u_{k1}^p v_{k2}^q,\ldots,u_{kI}^p v_{kJ}^q\}, & p > r, q > r \end{cases} \quad (27)$$

Just as the collection of terms $\{x^p y^q\}_{0 \le p+q \le d}$ span the polynomial P_d, (1), So the collection of column vectors

$$B_C(x,y) = \{B_{p,q}\}_{0 \le p+q \le d} \qquad (28)$$

span the piecewise polynomials that make up the C-spline.

Note that for the geometry $I = 2$, $J = 1$, polynomial order $d = 1$ and continuity order $r = 0$, the C-spline base (28) will differ from (12) by one column permutation. Both bases may be considered equivalent though in that they both enforce constraint (7).

5.3. Assigning Data Points to the C-Spline Base

We have N observed data points in the Cartesian (x,y)-plane that are related to some observed point on the z-axis through the unknown function f, that is

$$f(x_n, y_n) = z_n, \qquad n = 1,\ldots,N. \qquad (29)$$

by using base (28), we approximate the unknown function f with a collection of piecewise polynomials of degree d that are C^r continuous everywhere. To do this we first have to assign each data point (x_n, y_n) to its corresponding partitioning. The x- and y-axes of each partitioning have, see paragraph 5.1, lengths of $\Delta x = \dfrac{b_x - a_x}{I}$, $\Delta y = \dfrac{b_y - a_y}{J}$

we then have that for the data point (x_n, y_n) which lies in the partitioning having coordinates (i,j):

$$a_x + (i-1)\Delta x \le x_n \le a_x + i\Delta x,$$
$$a_y + (j-1)\Delta y \le y_n \le a_y + j\Delta y \quad \text{or, equivalently,}$$

$(i-1) \le \dfrac{x_n - a_x}{\Delta x} \le i$, $(j-1) \le \dfrac{y_n - a_y}{\Delta y} \le j$ it follows that the coordinates of the partitioning in which the data point (x_n, y_n) lies may be found as

$$i = ceil\left(\frac{x_n - a_x}{\Delta x}\right), \quad j = ceil\left(\frac{y_n - a_y}{\Delta y}\right) \qquad (30)$$

where $ceil(x)$ is the function that gives the smallest integer that is greater than or equal to x. Substituting these coordinates in (20), we may assign the data point (x_n, y_n) to its corresponding piecewise polynomial, or, equivalently, to its corresponding row k in the base (28).

Example

Say, we use the C-spline base as given in (12)

$$B_C(x,y) = \begin{pmatrix} 1 & y & x - \Delta x & 0 \\ 1 & y & 0 & x - \Delta x \end{pmatrix}$$

where the first and second row of B_C correspond, respectively, with the first and second partitioning of Fig. (**1**). Now, say we have a small dataset of $N = 5$ observations (x_n, y_n) having values of

$$(x_1, y_1) = (1.1\Delta x, 0.3\Delta y)$$
$$(x_2, y_2) = (1.2\Delta x, 0.7\Delta y)$$
$$(x_3, y_3) = (0.1\Delta x, 0.3\Delta y)$$
$$(x_4, y_4) = (0.5\Delta x, 0.1\Delta y)$$
$$(x_5, y_5) = (1.7\Delta x, 0.8\Delta y)$$

where Δx and Δy are some constants. Then, using (20) and (30), the points (x_3, y_3) and (x_4, y_4) are assigned to the first

partitioning, or, equivalently, to the first row of B_C. Likewise, (x_1, y_1), (x_2, y_2) and (x_5, y_5) are assigned to the second partitioning, or, equivalently, to the second row of B_C:

$$
\tilde{B}_C = \begin{pmatrix} B_C^{(2)}(x_1, y_1) \\ B_C^{(2)}(x_2, y_2) \\ B_C^{(1)}(x_3, y_3) \\ B_C^{(1)}(x_4, y_4) \\ B_C^{(2)}(x_5, y_5) \end{pmatrix} = \begin{pmatrix} 1 & 0.3\Delta y & 0 & 0.1\Delta x \\ 1 & 0.7\Delta y & 0 & 0.2\Delta x \\ 1 & 0.3\Delta y & -0.9\Delta x & 0 \\ 1 & 0.1\Delta y & -0.5\Delta x & 0 \\ 1 & 0.8\Delta y & 0 & 0.7\Delta x \end{pmatrix} \quad (31)
$$

Note that use a tilde to signify a base B_C to which data points have been assigned.

5.4. Constructing a C-Spline

Let m be the number of columns of the C-spline base B_C, (28). Then, after we have assigned all N data points to the base B_C, we get the $N \times m$ matrix $\tilde{B}_C$, see (31). The unknown b coefficients of the C-spline, see (13), are found through least-squares estimation [5]:

$$
b = \left(\tilde{B}_C^T \tilde{B}_C \right)^{-1} \tilde{B}_C^T z \quad (32)
$$

where $z = \begin{pmatrix} z_1 & \cdots & z_N \end{pmatrix}$ is the vector with output values, (29).

Now, say we wish to get the C-spline estimate $\hat{z}_{N+1}$ of the data point (x_{N+1}, y_{N+1}). Then, using (20) and (30), we first determine the row k of the base B_C, (28), that corresponds with this data point and then plug in its value. This results in the $1 \times m$ row-vector

$$
\tilde{B}_C^{(N+1)} = B_C^{(k)}(x_{N+1}, y_{N+1}) \quad (33)
$$

The estimate $\hat{z}_{N+1}$ is then found by simply taking the inner product of (33) and (32):

$$
z = \tilde{B}_C^{(N+1)} \cdot b
$$

We see that constructing a C-spline is equivalent to performing a regression analysis.

6. DISCUSSION

We have introduced here Cartesian splines, or C-splines, for short. C-splines are piecewise polynomials which are defined on adjacent Cartesian coordinate systems and are C^r continuous throughout. We have given here an algorithm that allows one to construct C-spline bases without first having to find the null-space of the corresponding smoothness matrix H. This makes the construction of a given C-spline base computationally trivial since no null-space of H has to be evaluated. This means that for C-splines the computational burden lies solely, just as in any ordinary regression analysis, in the evaluation of the inverse of $\tilde{B}_C^T \tilde{B}_C$, where $\tilde{B}_C$ is the matrix with the independent variables. Note that the algorithm, equations (20) through (28), may be generalized relatively easy to construct C-splines for multivariate domains.

ACKNOWLEDGMENTS

REFERENCES

[1]   de Boor C. B-form basics. Geometric modeling: algorithms and new trends, In: Farin G, Ed. SIAM Publication 1987; pp. 131-48.

[2]   Lai MJ, Schumaker LL. Scattered data interpolation using piecewise polynomials of degree six. SIAM Numer Anal 1997; 34: 905-21.

[3]   Awanou G, Lai MJ, Wenston P. Multivariate splines for scattered data fitting and numerical solutions of partial differential equations. Wavelets and Splines. In: Chen G, Lai MJ, Eds. Athens 2005; pp. 24-75.

[4]   Lay DC. Linear algebra and its applications. Reading, Massachusetts: Addison Wesley 2005.

[5]   Zellner A. An introduction to Bayesian inference in econometrics. New York: J. Wiley & Sons 1971.