



HAL
open science

Cubic and bicubic spline interpolation in Python

Quentin Agrapart, Alain Batailly

► **To cite this version:**

Quentin Agrapart, Alain Batailly. Cubic and bicubic spline interpolation in Python. [0] École Polytechnique de Montréal. 2020. ⟨hal-03017566v2⟩

HAL Id: hal-03017566

<https://hal.science/hal-03017566v2>

Submitted on 6 Apr 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization



date : **November 20, 2020**

authors : **Quentin Agrapart** (quentin.agrapart@polymtl.ca)
Alain Batailly (alain.batailly@polymtl.ca)

online resources :

wiki: <https://lava-wiki.meca.polymtl.ca/public/notespline/accueil>

Cubic and bicubic spline interpolation in Python

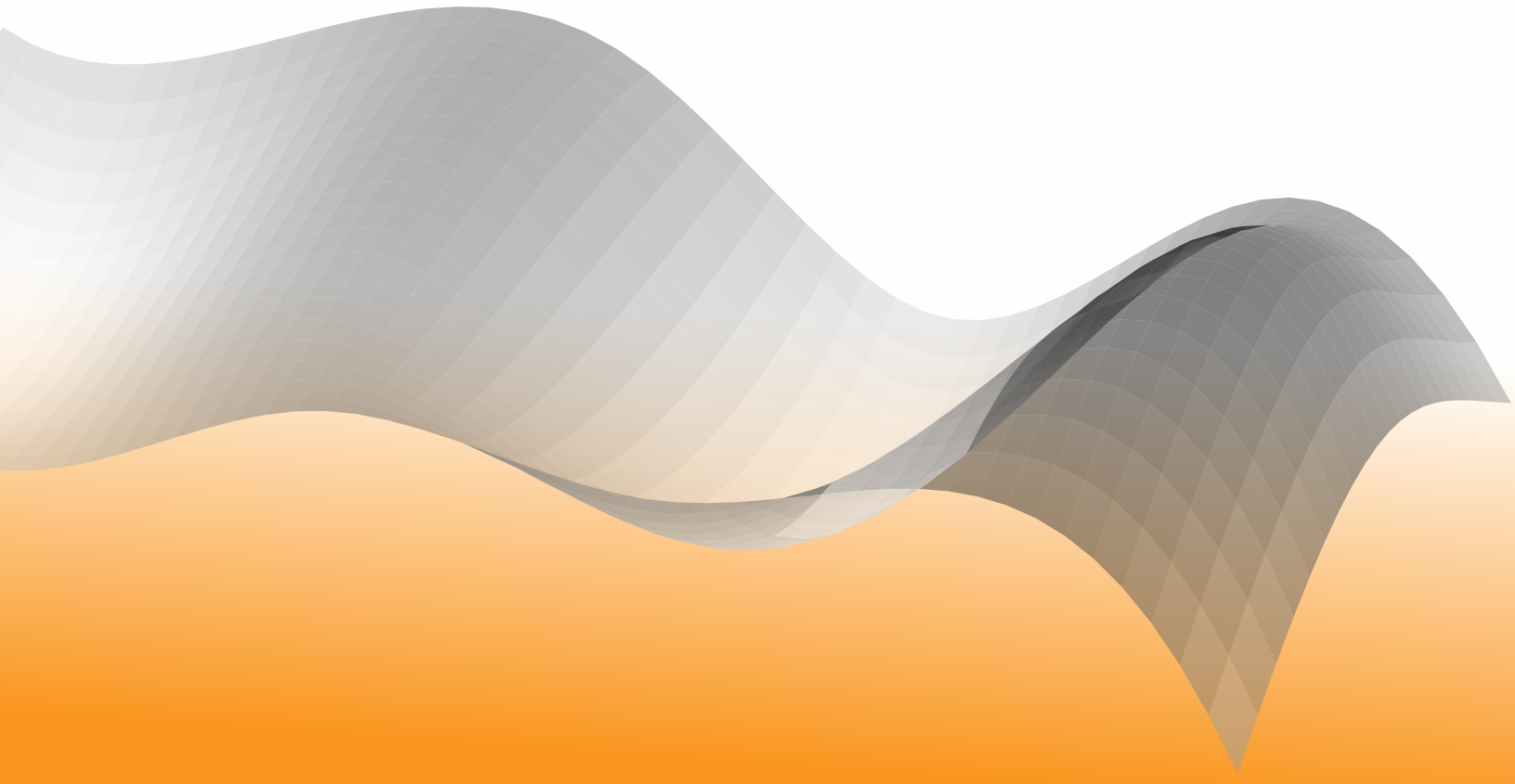


Table of contents

1	Two-dimensional cubic spline	3
1.1	Definition	3
1.2	Influence of control points	4
1.3	Interpolation of cubic spline	10
1.4	End conditions applied to cubic splines	11
1.4.1	free end condition	12
1.4.2	Imposed tangent direction	12
1.4.3	end-to-end tangent continuity	14
1.4.4	not-a-knot end condition	15
1.5	Positioning of control points	16
1.6	Plotting the cubic spline	16
1.6.1	free end condition	17
1.6.2	Imposed tangent direction	17
1.6.3	end-to-end tangent continuity	19
1.6.4	not-a-knot end condition	19
2	Three-dimensional bicubic parametric surface	19
2.1	Definition	19
2.2	Influence of control points	22
2.3	Interpolation equations for a parametric surface	26
2.4	End conditions for parametric surfaces	29
2.4.1	Free end condition	29
2.4.2	Cylindrical closed parametric surface	31
2.4.3	Spherical shaped parametric surface	35
2.4.4	not-a-knot end conditions	36
2.5	Position of control points	40
2.6	Plotting bicubic parametric surfaces	41
2.6.1	Free end condition	41
2.6.2	Cylindrical closed parametric surface	42
2.6.3	Spherical shaped parametric surface	42
2.6.4	not-a-knot end conditions	43
	Appendices	46

Introduction

Cubic and bicubic spline interpolations are widely used in a variety of domains. Nonetheless, there are limited resources available to help students or professionals who wish to implement these tools within a computer program. Be it for visualization purposes or for use within sophisticated algorithms, building a 2D or a 3D spline may not be a straightforward process.

In this context, the present technical note provides a brief theoretical description of both splines and bicubic splines but also focuses on the practical implementation of both concepts with an emphasis on the various types of boundary conditions that may be used. In particular, different configurations featuring free end conditions, *not-a-knot* end conditions or particular tangent orientations to build both open and closed parametric curves and surfaces are detailed.

Several source codes—written in Python 3.8—are provided with the intent to facilitate the reproduction of presented results.

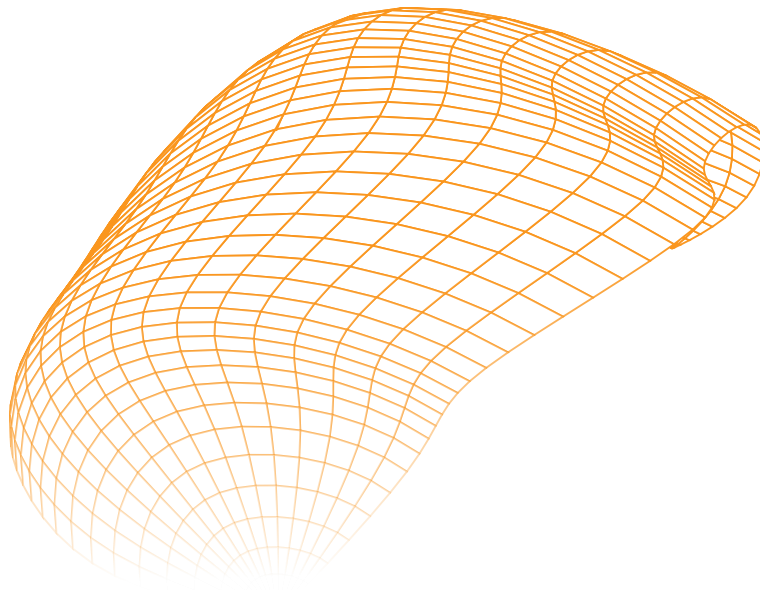


Figure 1. example of a bicubic spline surface

1 Two-dimensional cubic spline

1.1 Definition

A spline is a piecewise polynomial representation of a smooth curve which connects a set of knots. Each piece of the spline between two consecutive knots is called a *patch* in the following. On each patch, the spline is represented by a polynomial function of degree d . Within the framework of cubic spline, the focus is made on third order polynomials, with $d = 3$.

The polynomials on patches are written as a function of an intrinsic parameter t which follows the curve of the spline. Given a set of n knots t_i delimiting the ends of the patches, $n - 1$ patches can be defined between knots t_1 and t_n . The intrinsic parametrization of the space is such that the set of knots t_i are equidistant and belong to a same axis \vec{f} . A uniform knot vector \mathbf{t} can be defined which combines the successive values t_i of the intrinsic parameter which fix the ends of the successive patches.

$$\mathbf{t} = [t_1 \quad t_2 \quad t_3 \quad \dots \quad t_{n-1} \quad t_n] \quad (1)$$

Consequently, each interval $[t_i, t_{i+1}]$ fixes the ends of an elementary patch and all patches have the same length. This knot vector \mathbf{t} can also be chosen as an increasing list of integers where values reached at each knot.

$$\mathbf{t} = [0 \quad 1 \quad 2 \quad 3 \quad \dots \quad n-2 \quad n-1] \quad (2)$$

The shape of each patch of the spline is conditioned by the coefficients of a polynomial function. On one patch, the number of coefficients to be evaluated represents the order of the spline that is one greater than the polynomial degree, so 4 for cubic spline. The coefficients of successive patches are necessarily linked to each other since continuity position conditions at patch ends have to be verified. Also by definition, a cubic spline has continuous first and second derivatives which must be respected at distinct knots, except at the ends.

In practice, a patch is built as a linear combination of basis functions (B-spline functions) whose influence is driven by a set of neighboring control points \mathbf{Q}_i , see equation (3).

$$\mathbf{c}(t) = \sum_i^{i+d} \mathbf{Q}_i B_{i,d}(t) \quad (3)$$

with:

- t is the intrinsic parameter of a spatial system that follows the spline curve,
- $\mathbf{c}(t) = [x_c(t) \quad y_c(t)]$ are the spline curve coordinates evaluated at t in the Cartesian coordinate system $\langle \vec{x}; \vec{y} \rangle$.
- \mathbf{Q}_i are the control point coordinates,
- n is the number of fit knots,
- d is the degree of the parametric curve,
- $B_{i,d}(t)$ are the d -degree blending functions or the d -degree B-spline functions that can be defined by the Cox-de-Boor recursive algorithm.

1.2 Influence of control points

A polynomial function can only be described by one unique combination of B-spline functions of the same order. It means that a given set of control points implies unique shapes to patches. For example in figure 2, a single cubic spline (—) can be plotted from a given arbitrary set of $n + 2$ control points (\bullet). Each control point \mathbf{Q}_i is defined as a coordinates vector $\mathbf{Q}_i = [\hat{x}_i \ \hat{y}_i]$. To draw the entire curve $\mathbf{c}(t)$ built on control points, the intrinsic parameter t evolves in the interval $[0, n - 1]$. Patch ends are displayed in (\bullet) and values of t reached along the parametric curve are detailed at those knots.

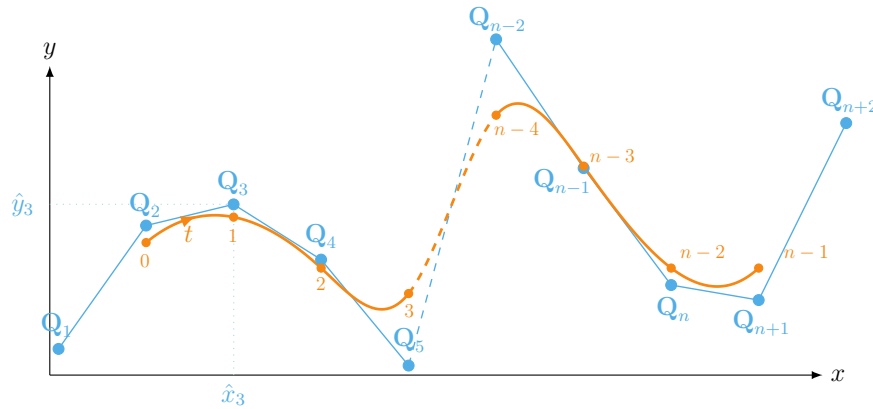


Figure 2. parametric curve $\mathbf{c}(t)$ (—) driven by an arbitrary set of control points (\bullet)

The control points are then representative of the polynomial coefficients which set the shape of patches. However, their influences are restricted to a limited number of intervals related to the degree of the spline. This restriction is only due to the domain of definition of B-splines functions. Conventionally, degree d B-splines functions $B_{i,d}(t)$, also called degree d blending functions, are defined by the Cox-de-Boor algorithm:

$$B_{i,0}(t) = \begin{cases} 1 & \text{if } t \in [t_i, t_{i+1}] \\ 0 & \text{if } t \notin [t_i, t_{i+1}] \end{cases} \quad (4)$$

$$B_{i,d}(t) = \frac{t - t_i}{t_{i+d} - t_i} B_{i,d-1}(t) + \frac{t_{i+d+1} - t}{t_{i+d+1} - t_{i+1}} B_{i+1,d-1}(t) \quad (5)$$

The Cox-de-Boor algorithm successively calls on lower degree B-splines functions, by additionally involving one or more following control points depending on its degree. A degree 0 blending function $B_{i,0}(t)$, define a step function equal to 1 on the interval $t \in [t_i, t_{i+1}]$. Otherwise, the equation is zero for all other values. B-spline functions are successively evaluated in higher degree functions up to $B_{i,3}(t)$ in the case of cubic spline. Thereby, equation (4) is used to calculate the linear B-spline function (with $d = 1$) formulated in equation (6) and developed in equation (7). Proper linear B-spline functions are then replaced in equation (8) for the calculation of quadratic B-spline functions ($d = 2$). On the same principle, cubic B-spline functions ($d = 3$) are obtained in equation (11).

linear B-spline function:

$$B_{i,1}(t) = \frac{t - t_i}{t_{i+1} - t_i} B_{i,0}(t) + \frac{t_{i+2} - t}{t_{i+2} - t_{i+1}} B_{i+1,0}(t) \quad (6)$$

$$B_{i,1}(t) = \begin{cases} \frac{t-t_i}{t_{i+1}-t_i} & \text{if } t \in [t_i, t_{i+1}] \\ \frac{t_{i+2}-t}{t_{i+2}-t_{i+1}} & \text{if } t \in [t_{i+1}, t_{i+2}] \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

quadratic B-spline function:

$$B_{i,2}(t) = \frac{t - t_i}{t_{i+2} - t_i} B_{i,1}(t) + \frac{t_{i+3} - t}{t_{i+3} - t_{i+1}} B_{i+1,1}(t) \quad (8)$$

$$B_{i,2}(t) = \begin{cases} \frac{t-t_i}{t_{i+2}-t_i} \frac{t-t_i}{t_{i+1}-t_i} & \text{if } t \in [t_i, t_{i+1}] \\ \frac{t-t_i}{t_{i+2}-t_i} \frac{t_{i+2}-t}{t_{i+2}-t_{i+1}} + \frac{t_{i+3}-t}{t_{i+3}-t_{i+1}} \frac{t-t_{i+1}}{t_{i+2}-t_{i+1}} & \text{if } t \in [t_{i+1}, t_{i+2}] \\ \frac{t_{i+3}-t}{t_{i+3}-t_{i+1}} \frac{t_{i+3}-t}{t_{i+3}-t_{i+2}} & \text{if } t \in [t_{i+2}, t_{i+3}] \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

cubic B-spline function:

$$B_{i,3}(t) = \frac{t - t_i}{t_{i+3} - t_i} B_{i,2}(t) + \frac{t_{i+4} - t}{t_{i+4} - t_{i+1}} B_{i+1,2}(t) \quad (10)$$

$$B_{i,3}(t) = \begin{cases} \frac{t-t_i}{t_{i+3}-t_i} \frac{t-t_i}{t_{i+2}-t_i} \frac{t-t_i}{t_{i+1}-t_i} & \text{if } t \in [t_i, t_{i+1}] \\ \frac{t-t_i}{t_{i+3}-t_i} \left[\frac{t-t_i}{t_{i+2}-t_i} \frac{t_{i+2}-t}{t_{i+2}-t_{i+1}} + \frac{t_{i+3}-t}{t_{i+3}-t_{i+1}} \frac{t-t_{i+1}}{t_{i+2}-t_{i+1}} \right] + \frac{t_{i+4}-t}{t_{i+4}-t_{i+1}} \frac{t-t_{i+1}}{t_{i+3}-t_{i+1}} \frac{t-t_{i+1}}{t_{i+2}-t_{i+1}} & \text{if } t \in [t_{i+1}, t_{i+2}] \\ \frac{t-t_i}{t_{i+3}-t_i} \frac{t_{i+3}-t}{t_{i+3}-t_{i+1}} \frac{t_{i+3}-t}{t_{i+3}-t_{i+2}} + \frac{t_{i+4}-t}{t_{i+4}-t_{i+1}} \left[\frac{t-t_{i+1}}{t_{i+3}-t_{i+1}} \frac{t_{i+3}-t}{t_{i+3}-t_{i+2}} + \frac{t_{i+4}-t}{t_{i+4}-t_{i+2}} \frac{t-t_{i+2}}{t_{i+3}-t_{i+2}} \right] & \text{if } t \in [t_{i+2}, t_{i+3}] \\ \frac{t_{i+4}-t}{t_{i+4}-t_{i+1}} \frac{t_{i+4}-t}{t_{i+4}-t_{i+2}} \frac{t_{i+4}-t}{t_{i+4}-t_{i+3}} & \text{if } t \in [t_{i+3}, t_{i+4}] \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

As the degree d of the B-spline function increases, the number of intervals on which the function is non-zero also increases. Domains of influence of B-spline functions are highlighted in figure 3. B-spline functions are plotted over the interval $t \in [t_i, t_i + 4]$ with different degrees d from 0 to 3, respectively in figures 3a to 3d. As evidenced by a non-zero cubic B-spline function on 4 intervals, the higher the degree, the larger the number of control points that will affect the shape of the spline.

The Cox-de-Boor algorithm and the links between degree d B-spline functions are schematized in figure 4. An arbitrary sequence of 7 patches is represented in (—). The knot vector here starts at index t_i

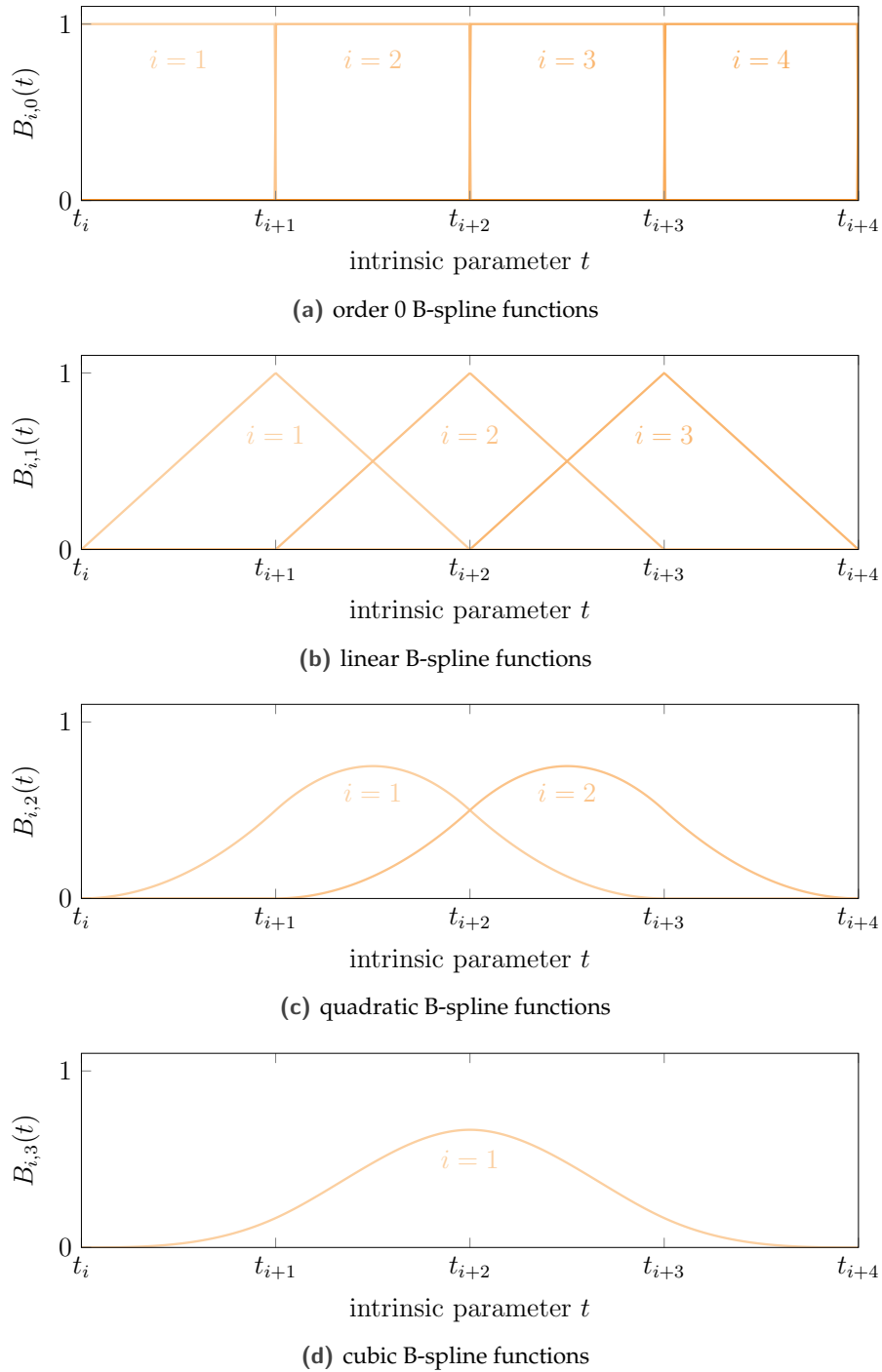


Figure 3. B-spline functions with different degree d over $t \in [t_i, t_i + 4]$ and links between d degree B-spline functions

and ends at t_{i+7} , but it is assumed that the spline continues hypothetically beyond the ends. The number of control points needed for setting up one patch is therefore also equal to the order of the spline. For

cubic B-spline functions, one control point can affect up to four patches. The same can be observed in equation (11) with the presence of four non-zero intervals defining the function $B_{i,3}(t)$. By taking the problem backwards, it is highlighted that one patch is controlled by 4 control points.

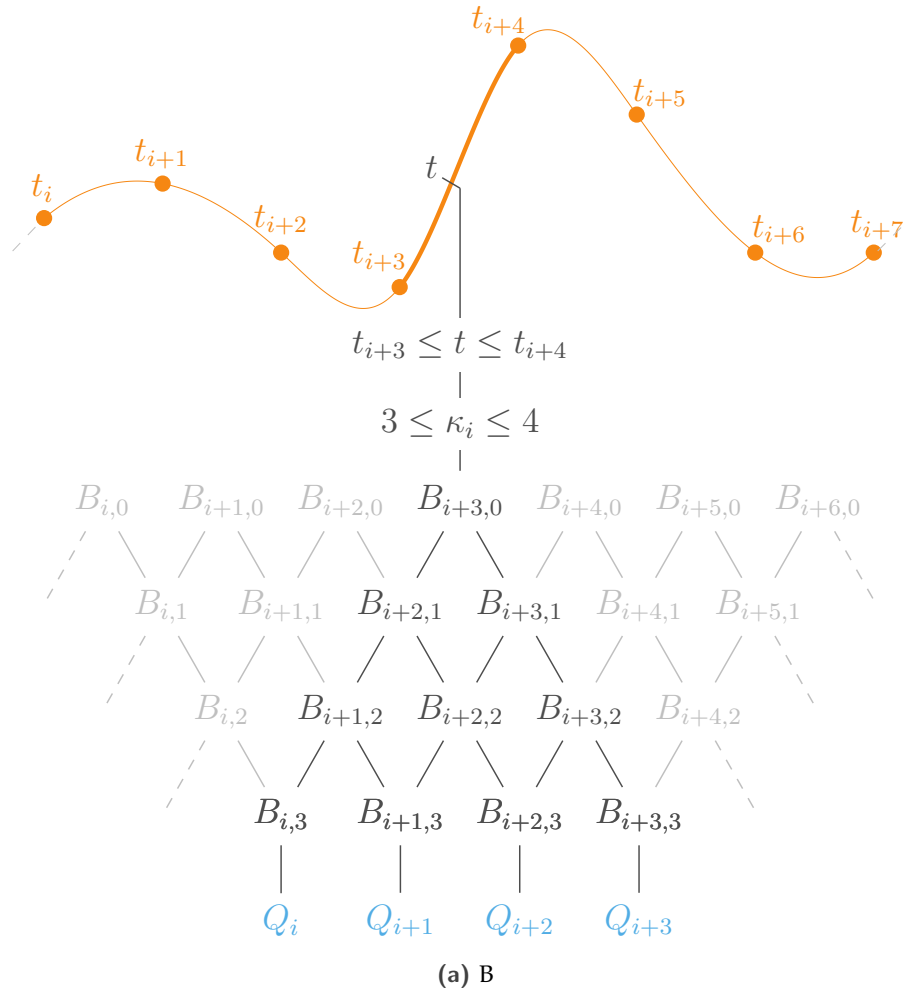


Figure 4. Schematization of the Cox-de-Boor algorithm and links between d degree B-spline functions

The calculation of the spline coordinates $\mathbf{c}(t)$, see equation (3), therefore requires the evaluation of $B_{i,3}$, $B_{i+1,3}$, $B_{i+2,3}$ and $B_{i+3,3}$. These functions are easily obtained by incrementing the index i in equation (11). To define the entire cubic spline, the intrinsic parameter t evolves in the interval $[0, n - 1]$ while index i needs to be adjusted accordingly to the patch number to which belongs t . Some simplifications are made at this stage to ensure that each patch has an elementary parameterization:

- a change of variables,

$$\kappa_i = t - t_i \tag{12}$$

- the proper use of homogeneous properties of the knot vector \mathbf{t} ,

$$t_{i+1} - t_i = 1 \quad (13)$$

- as κ_i is also a function of i , equation (14) is also true.

$$\kappa_{i+1} = \kappa_i - 1 \quad (14)$$

It comes:

$$\mathbf{c}(\kappa_i) = \mathbf{Q}_i B_{i,3}(\kappa_i) + \mathbf{Q}_{i+1} B_{i+1,3}(\kappa_i) + \mathbf{Q}_{i+2} B_{i+2,3}(\kappa_i) + \mathbf{Q}_{i+3} B_{i+3,3}(\kappa_i) \quad (15)$$

with cubic B-spline functions obtained from equation (11) rewrites:

$$B_{i,3}(\kappa_i) = \frac{1}{6} \begin{cases} \kappa_i^3 & \text{if } \kappa_i \in [0, 1] \\ -3\kappa_i^3 + 12\kappa_i^2 - 12\kappa_i + 4 & \text{if } \kappa_i \in [1, 2] \\ 3\kappa_i^3 - 24\kappa_i^2 + 60\kappa_i - 44 & \text{if } \kappa_i \in [2, 3] \\ (4 - \kappa_i)^3 & \text{if } \kappa_i \in [3, 4] \\ 0 & \text{otherwise} \end{cases} \quad (16)$$

$$B_{i+1,3}(\kappa_i) = \frac{1}{6} \begin{cases} (\kappa_i - 1)^3 & \text{if } \kappa_i \in [1, 2] \\ -3(\kappa_i - 1)^3 + 12(\kappa_i - 1)^2 - 12(\kappa_i - 1) + 4 & \text{if } \kappa_i \in [2, 3] \\ 3(\kappa_i - 1)^3 - 24(\kappa_i - 1)^2 + 60(\kappa_i - 1) - 44 & \text{if } \kappa_i \in [3, 4] \\ (4 - \kappa_i - 1)^3 & \text{if } \kappa_i \in [4, 5] \\ 0 & \text{otherwise} \end{cases} \quad (17)$$

$$B_{i+2,3}(\kappa_i) = \frac{1}{6} \begin{cases} (\kappa_i - 2)^3 & \text{if } \kappa_i \in [2, 3] \\ -3(\kappa_i - 2)^3 + 12(\kappa_i - 2)^2 - 12(\kappa_i - 2) + 4 & \text{if } \kappa_i \in [3, 4] \\ 3(\kappa_i - 2)^3 - 24(\kappa_i - 2)^2 + 60(\kappa_i - 2) - 44 & \text{if } \kappa_i \in [4, 5] \\ (4 - \kappa_i - 2)^3 & \text{if } \kappa_i \in [5, 6] \\ 0 & \text{otherwise} \end{cases} \quad (18)$$

$$B_{i+3,3}(\kappa_i) = \frac{1}{6} \begin{cases} (\kappa_i - 3)^3 & \text{if } \kappa_i \in [3, 4] \\ -3(\kappa_i - 3)^3 + 12(\kappa_i - 3)^2 - 12(\kappa_i - 3) + 4 & \text{if } \kappa_i \in [4, 5] \\ 3(\kappa_i - 3)^3 - 24(\kappa_i - 3)^2 + 60(\kappa_i - 3) - 44 & \text{if } \kappa_i \in [5, 6] \\ (4 - \kappa_i - 3)^3 & \text{if } \kappa_i \in [6, 7] \\ 0 & \text{otherwise} \end{cases} \quad (19)$$

The intrinsic parameter κ_i allows to control simultaneously the position covered on the spline and the patch indexing. It is important to remember that patch indexing controls the intervals where B-spline functions are defined, also linked to the control points indexing. The interval $\kappa_i \in [3, 4]$, highlighted in (■) in equations (16) to (19), is the only one for which all functions are defined and not zero at the same time. This mean that all 4 control points Q_i, Q_{i+1}, Q_{i+2} and Q_{i+3} simultaneously impact the shape of the spline curve for $\kappa_i \in [3, 4]$. Note that this statement depends to the degree of the spline and to the relative indexing chosen between the knots vector and the control points.

A last change of variables is operated so that an elementary system where the intrinsic parameter evolves between 0 and 1 is obtained on each patch:

$$\varkappa_i = \kappa_i - 3 \quad (20)$$

Finally, for $\varkappa_i \in [0, 1]$ and after polynomial expansions from equation (16) to (19), it remains:

$$B_{i,3}(\varkappa_i) = \frac{1}{6}(1 - \varkappa_i)^3 \quad (21)$$

$$B_{i+1,3}(\varkappa_i) = \frac{1}{6}(3\varkappa_i^3 - 6\varkappa_i^2 + 4) \quad (22)$$

$$B_{i+2,3}(\varkappa_i) = \frac{1}{6}(-3\varkappa_i^3 + 3\varkappa_i^2 + 3\varkappa_i + 1) \quad (23)$$

$$B_{i+3,3}(\varkappa_i) = \frac{1}{6}\varkappa_i^3 \quad (24)$$

Then, the formulation of the parametric curve takes the form of equation (25). \varkappa_i denotes the elementary intrinsic parameter of the i -th patch of the spline. Between two knots, \varkappa_i always evolves between 0 et 1 and only the control point indexes accordingly change. $\mathbf{c}(\varkappa_i)$ has to be evaluated patch by patch with its respective elementary intrinsic parameter to draw the entire cubic spline.

$$\mathbf{c}(\varkappa_i) = \frac{1}{6} [(-\varkappa_i^3 + 3\varkappa_i^2 - 3\varkappa_i + 1)\mathbf{Q}_i + (3\varkappa_i^3 - 6\varkappa_i^2 + 4)\mathbf{Q}_{i+1} + (-3\varkappa_i^3 + 3\varkappa_i^2 + 3\varkappa_i + 1)\mathbf{Q}_{i+2} + \varkappa_i^3\mathbf{Q}_{i+3}] \quad (25)$$

Equation (25) can also be written in a well-known matrix form:

$$\mathbf{c}(\varkappa_i) = \frac{1}{6} \begin{bmatrix} \varkappa_i^3 & \varkappa_i^2 & \varkappa_i & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_i \\ \mathbf{Q}_{i+1} \\ \mathbf{Q}_{i+2} \\ \mathbf{Q}_{i+3} \end{bmatrix} \quad (26)$$

However, the position of the control points must be known to use this formula. We will see later how to best position them according to a combination of interpolation equations and end condition constraints. Equation (26) will serve as a basis to fix the position of control points in the following.

1.3 Interpolation of cubic spline

A spline interpolation consists in connecting a set of fit knots with a parametric curve. Given a set of n fit knots \mathbf{P}_i to be interpolated, with index $i \in [1, n]$. Each fit knots \mathbf{P}_i is defined as a coordinate vector $\mathbf{P}_i = \begin{bmatrix} x_i & y_i \end{bmatrix}$ in the Cartesian coordinate system $\langle \vec{x}; \vec{y} \rangle$. It is assumed that the spline curve passes through the fit knots consistently with the value of their index i . In practice, this means conditioning the shape of the spline by a certain number of interpolation equations. To find these constraint equations, the parametric curve has to be evaluated to locally correspond to the coordinates of the fit knots. Hence, the following system of interpolation equations therefore summarizes the problem to be solved considering that the ends of all patches have to match consecutive fit knots:

$$\begin{cases} \mathbf{c}(z_i = 0) = \mathbf{P}_i & \text{for } 1 \leq i \leq (n-1) \\ \mathbf{c}(z_{n-1} = 1) = \mathbf{P}_n \end{cases} \quad (27)$$

For fit knots \mathbf{P}_1 to \mathbf{P}_{n-1} , interpolation equations are simplified below for $1 \leq i \leq n-1$:

$$\mathbf{P}_i = \mathbf{c}(z_i = 0) = \frac{1}{6} \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_i \\ \mathbf{Q}_{i+1} \\ \mathbf{Q}_{i+2} \\ \mathbf{Q}_{i+3} \end{bmatrix} = \frac{1}{6} \begin{bmatrix} 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_i \\ \mathbf{Q}_{i+1} \\ \mathbf{Q}_{i+2} \\ \mathbf{Q}_{i+3} \end{bmatrix} \quad (28)$$

And, for the last fit knots \mathbf{P}_n :

$$\mathbf{P}_n = \mathbf{c}(z_{n-1} = 1) = \frac{1}{6} \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{n-1} \\ \mathbf{Q}_{n-1+1} \\ \mathbf{Q}_{n-1+2} \\ \mathbf{Q}_{n-1+3} \end{bmatrix} = \frac{1}{6} \begin{bmatrix} 0 & 1 & 4 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{n-1} \\ \mathbf{Q}_n \\ \mathbf{Q}_{n+1} \\ \mathbf{Q}_{n+2} \end{bmatrix} \quad (29)$$

As the same expressions are found after simplification, the interpolation problem can be generalized for all indexes $1 \leq i \leq n$:

$$\mathbf{P}_i = \frac{1}{6}(\mathbf{Q}_i + 4\mathbf{Q}_{i+1} + \mathbf{Q}_{i+2}) \quad (30)$$

Expressed in matrix form, the system of interpolation equations becomes:

$$\begin{bmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \vdots \\ \mathbf{P}_{n-1} \\ \mathbf{P}_n \end{bmatrix} = \frac{1}{6} \begin{bmatrix} 1 & 4 & 1 & 0 & \dots & \dots & \dots & \dots & 0 \\ 0 & 1 & 4 & 1 & 0 & & & & \vdots \\ \vdots & 0 & 1 & 4 & 1 & 0 & & & \vdots \\ \vdots & & 0 & \ddots & \ddots & \ddots & 0 & & \vdots \\ \vdots & & & 0 & 1 & 4 & 1 & 0 & \vdots \\ \vdots & & & & 0 & 1 & 4 & 1 & 0 \\ 0 & \dots & \dots & \dots & \dots & 0 & 1 & 4 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_1 \\ \mathbf{Q}_2 \\ \mathbf{Q}_3 \\ \vdots \\ \mathbf{Q}_n \\ \mathbf{Q}_{n+1} \\ \mathbf{Q}_{n+2} \end{bmatrix} \quad (31)$$

Finally, the interpolation of the parametric curve $\mathbf{c}(z_i)$ is driven by equation (32) where \mathbf{P} is the fit knot coordinates vector, \mathbf{Q} is the control point coordinates vector and Φ is the passing matrix from \mathbf{Q}_i to \mathbf{P}_i .

$$\mathbf{P} = \frac{1}{6} \Phi \mathbf{Q} \quad (32)$$

Vectors \mathbf{P} and \mathbf{Q} do not have the same dimensions so the problem is not invertible. Indeed, $n + 2$ unknowns appear (length of vector \mathbf{Q}) when only n equations are available (number of rows in the matrix Φ). In order to determine \mathbf{Q}_i knowing fit knot coordinates, end conditions have to be formulated to complete the interpolation equations.

1.4 End conditions applied to cubic splines

Interpolation equations alone do not make it possible to establish a sufficient number of mathematical relation between fit knots and control points. As seen before, two equations are missing for the inverse calculation of all the control points coordinates. Additional equations have to be introduced in the form of end condition constraints while adjusting the shape of the curve at both ends. These boundary conditions result in considering relations between several control points or by adding local mathematical properties to the parametric curve. In practice, the missing relations are appended by completing the vector of fit knots \mathbf{P} and by adding rows to the matrix Φ . \mathbf{P}^* et Φ^* respectively denote the completed formulation of \mathbf{P} and Φ considering the two end boundary conditions.

Details for implementing different types of end conditions are given in the following paragraphs. This section deals with:

free end condition involving the symmetry of 3 control points at both ends,

imposed tangent direction where the slope at both ends is user defined,

end-to-end tangent continuity which ensures the same tangent directions and same curvatures at both ends,

not-a-knot end condition defined by the continuity of the third derivative at the junction of the first two patches.

1.4.1 free end condition

Free end conditions are the easiest to set up since based on the choice of relations between several control points at both ends. The first 3 control points must be aligned and separated two by two by the same distance. This choice of end conditions results in the following relations:

- the control point \mathbf{Q}_2 is the middle of the segment $[\mathbf{Q}_1, \mathbf{Q}_3]$,

$$\mathbf{Q}_2 = \frac{1}{2}(\mathbf{Q}_1 + \mathbf{Q}_3) \tag{33}$$

- the control point \mathbf{Q}_{n+1} is the middle of the segment $[\mathbf{Q}_n, \mathbf{Q}_{n+2}]$.

$$\mathbf{Q}_{n+1} = \frac{1}{2}(\mathbf{Q}_n + \mathbf{Q}_{n+2}) \tag{34}$$

which also take the form:

$$\mathbf{Q}_1 - 2\mathbf{Q}_2 + \mathbf{Q}_3 = 0 \tag{35}$$

$$\mathbf{Q}_n - 2\mathbf{Q}_{n+1} + \mathbf{Q}_{n+2} = 0 \tag{36}$$

Equations (35) and (36) are directly added in the matrix system from equation (31). Added rows are highlighted in red in equation (37). The positions of rows are not important as long as it is consistent in vector \mathbf{P}^* and matrix Φ^* . So, the first row corresponds to the left end constraint equation and the last one to the right end of the spline.

$$\begin{bmatrix} 0 \\ \mathbf{P}_1 \\ \mathbf{P}_2 \\ \vdots \\ \mathbf{P}_{n-1} \\ \mathbf{P}_n \\ 0 \end{bmatrix} = \frac{1}{6} \begin{bmatrix} 1 & -2 & 1 & 0 & 0 & 0 & \dots & 0 \\ 1 & 4 & 1 & 0 & \dots & \dots & \dots & 0 \\ 0 & 1 & 4 & 1 & 0 & & & \vdots \\ \vdots & 0 & 1 & 4 & 1 & 0 & & \vdots \\ \vdots & & 0 & \ddots & \ddots & \ddots & 0 & \vdots \\ \vdots & & & 0 & 1 & 4 & 1 & 0 \\ \vdots & & & & 0 & 1 & 4 & 1 & 0 \\ 0 & \dots & \dots & \dots & 0 & 1 & 4 & 1 \\ 0 & \dots & \dots & 0 & 0 & 0 & 1 & -21 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_1 \\ \mathbf{Q}_2 \\ \mathbf{Q}_3 \\ \vdots \\ \mathbf{Q}_n \\ \mathbf{Q}_{n+1} \\ \mathbf{Q}_{n+2} \end{bmatrix} \tag{37}$$

1.4.2 Imposed tangent direction

Fixing the tangent direction at both ends allows to fit the parametric curve on particular slope orientations. The goal is to establish a constraint equation which concerns the direction of the tangent at one end and

translate it into a relation between control points. The first step is to calculate the first derivative of the parametric curve in equation (38), valid on all patches between indexes $i \in [0, n - 1]$.

$$\frac{\partial \mathbf{c}(\varkappa_i)}{\partial \varkappa_i} = \frac{1}{6} \begin{bmatrix} 3\varkappa_i^2 & 2\varkappa_i & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_i \\ \mathbf{Q}_{i+1} \\ \mathbf{Q}_{i+2} \\ \mathbf{Q}_{i+3} \end{bmatrix} \quad (38)$$

This equation can then be evaluated at both ends, see equations (39) and (40). Here, \mathbf{T}_1 and \mathbf{T}_n describe the tangent vectors used from either side. These define the orientation given locally to the tangent.

$$\frac{\partial \mathbf{c}(\varkappa_1 = 0)}{\partial \varkappa_1} = \mathbf{T}_1 = [T_{x1}, T_{y1}] \quad (39)$$

$$\frac{\partial \mathbf{c}(\varkappa_{n-1} = 1)}{\partial \varkappa_{n-1}} = \mathbf{T}_n = [T_{xn}, T_{yn}] \quad (40)$$

After equations development, following relations between control points remain as results.

$$\mathbf{T}_1 = \frac{1}{6}(-3\mathbf{Q}_1 + 3\mathbf{Q}_3) \quad (41)$$

$$\mathbf{T}_n = \frac{1}{6}(-3\mathbf{Q}_n + 3\mathbf{Q}_{n+2}) \quad (42)$$

Finally, equations relative to the tangent direction are added to the interpolation equations:

$$\begin{bmatrix} \mathbf{T}_1 \\ \mathbf{P}_1 \\ \mathbf{P}_2 \\ \vdots \\ \mathbf{P}_{n-1} \\ \mathbf{P}_n \\ \mathbf{T}_n \end{bmatrix} = \frac{1}{6} \begin{bmatrix} -3 & 0 & 3 & 0 & \dots & 0 & 0 & 0 & 0 \\ 1 & 4 & 1 & 0 & \dots & \dots & \dots & \dots & 0 \\ 0 & 1 & 4 & 1 & 0 & & & & \vdots \\ \vdots & 0 & 1 & 4 & 1 & 0 & & & \vdots \\ \vdots & & 0 & \ddots & \ddots & \ddots & 0 & & \vdots \\ \vdots & & & 0 & 1 & 4 & 1 & 0 & \vdots \\ \vdots & & & & 0 & 1 & 4 & 1 & 0 \\ 0 & \dots & \dots & \dots & 0 & 1 & 4 & 1 & \\ 0 & 0 & 0 & 0 & \dots & 0 & -3 & 0 & 3 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_1 \\ \mathbf{Q}_2 \\ \mathbf{Q}_3 \\ \vdots \\ \mathbf{Q}_n \\ \mathbf{Q}_{n+1} \\ \mathbf{Q}_{n+2} \end{bmatrix} \quad (43)$$

1.4.3 end-to-end tangent continuity

Previous relations established on tangent orientations relate independently to the two ends of the spline. Linking the parametric curves at both ends may be considered to represent closed circular geometries or cyclic boundary conditions. Then, end-to-end connection conditions ensure the tangent continuity and a continuous parametric function of class C^2 at the fitting. In this case, the continuity of the first and second derivative must be verified at both ends of the spline. The first derivative of the parametric curve has already been detailed in equation (38). The second derivated is easily obtained in equation (44) by derivating one more time by variable \varkappa_i .

$$\frac{\partial^2 \mathbf{c}(\varkappa_i)}{\partial \varkappa_i^2} = \frac{1}{6} \begin{bmatrix} 6\varkappa_i & 2 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_i \\ \mathbf{Q}_{i+1} \\ \mathbf{Q}_{i+2} \\ \mathbf{Q}_{i+3} \end{bmatrix} \quad \text{with } i \in [0, n-1] \quad (44)$$

Following equalities have to be asserted to verify end-to-end connection conditions:

$$\frac{\partial \mathbf{c}(\varkappa_1 = 0)}{\partial \varkappa_1} = \frac{\partial \mathbf{c}(\varkappa_{n-1} = 1)}{\partial \varkappa_{n-1}} \quad (45)$$

$$\frac{\partial^2 \mathbf{c}(\varkappa_1 = 0)}{\partial \varkappa_1^2} = \frac{\partial^2 \mathbf{c}(\varkappa_{n-1} = 1)}{\partial \varkappa_{n-1}^2} \quad (46)$$

Relations between control points are established by replacing each derivative by its expression evaluated at the right end.

$$\frac{1}{6}(-3\mathbf{Q}_1 + 3\mathbf{Q}_3) = \frac{1}{6}(-3\mathbf{Q}_n + 3\mathbf{Q}_{n+2}) \quad (47)$$

$$\frac{1}{6}(6\mathbf{Q}_1 - 12\mathbf{Q}_2 + 6\mathbf{Q}_3) = \frac{1}{6}(6\mathbf{Q}_n - 12\mathbf{Q}_{n+1} + 6\mathbf{Q}_{n+2}) \quad (48)$$

Two equations finally remain:

$$0 = \frac{1}{6}(-3\mathbf{Q}_1 + 3\mathbf{Q}_3 + 3\mathbf{Q}_n - 3\mathbf{Q}_{n+2}) \quad (49)$$

$$0 = \frac{1}{6}(6\mathbf{Q}_1 - 12\mathbf{Q}_2 + 6\mathbf{Q}_3 - 6\mathbf{Q}_n + 12\mathbf{Q}_{n+1} - 6\mathbf{Q}_{n+2}) \quad (50)$$

After formatting in matrix form in equation (51), the coordinates of the control points can be evaluated with the same inverse method.

$$\begin{bmatrix} 0 \\ \mathbf{P}_1 \\ \mathbf{P}_2 \\ \vdots \\ \mathbf{P}_{n-1} \\ \mathbf{P}_n \\ 0 \end{bmatrix} = \frac{1}{6} \begin{bmatrix} -3 & 0 & 3 & 0 & \dots & 0 & 3 & 0 & -3 \\ 1 & 4 & 1 & 0 & \dots & \dots & \dots & \dots & 0 \\ 0 & 1 & 4 & 1 & 0 & \dots & \dots & \dots & \vdots \\ \vdots & 0 & 1 & 4 & 1 & 0 & \dots & \dots & \vdots \\ \vdots & 0 & \ddots & \ddots & \ddots & 0 & \dots & \dots & \vdots \\ \vdots & \dots & 0 & 1 & 4 & 1 & 0 & \dots & \vdots \\ \vdots & \dots & \dots & 0 & 1 & 4 & 1 & 0 & \vdots \\ 0 & \dots & \dots & \dots & 0 & 1 & 4 & 1 & \vdots \\ 6 & -12 & 6 & 0 & \dots & 0 & -6 & 12 & -6 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_1 \\ \mathbf{Q}_2 \\ \mathbf{Q}_3 \\ \vdots \\ \mathbf{Q}_n \\ \mathbf{Q}_{n+1} \\ \mathbf{Q}_{n+2} \end{bmatrix} \quad (51)$$

1.4.4 not-a-knot end condition

As its name suggests, *not-a-knot* end condition offers a plot of spline extremities with smooth curvature. Considering *not-a-knot* end condition ensures the continuity of the 3rd derivative of the spline curve at the second and penultimate fit knots. The class of the curve has to be C^3 at fit knots P_2 et P_{n-1} . At both ends, it also means that the first two patches are defined by the same parametric curves. *not-a-knot* end conditions are so based on the calculation of the third derivatives of $\mathbf{c}(\varkappa_i)$, see equation (52).

$$\frac{\partial^3 \mathbf{c}(\varkappa_i)}{\partial \varkappa_i^3} = \begin{bmatrix} -1 & 3 & -3 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_i \\ \mathbf{Q}_{i+1} \\ \mathbf{Q}_{i+2} \\ \mathbf{Q}_{i+3} \end{bmatrix} \quad \text{with } i \in [0, n - 1] \quad (52)$$

Firstly, problem at fit knot P_2 is investigated. The third derivative at the right end of patch 1 (reached when parameter $\varkappa_1 = 1$) has to be equal to the third derivative at the left end of patch 2 (for $\varkappa_2 = 0$). There comes the equality:

$$\frac{\partial^3 \mathbf{c}(\varkappa_1 = 1)}{\partial \varkappa_1^3} = \frac{\partial^3 \mathbf{c}(\varkappa_2 = 0)}{\partial \varkappa_2^3} \quad (53)$$

$$-\mathbf{Q}_1 + 3\mathbf{Q}_2 - 3\mathbf{Q}_3 + \mathbf{Q}_4 = -\mathbf{Q}_2 + 3\mathbf{Q}_3 - 3\mathbf{Q}_4 + \mathbf{Q}_5 \quad (54)$$

$$0 = -\mathbf{Q}_1 + 4\mathbf{Q}_2 - 6\mathbf{Q}_3 + 4\mathbf{Q}_4 - \mathbf{Q}_5 \quad (55)$$

On the other end, *not-a-knot* end condition written on fit knot \mathbf{P}_{n-1} involves equality of the third derivative for patch $n - 2$ at $\varkappa_{n-2} = 1$ and patch $n - 1$ at $\varkappa_{n-1} = 0$.

$$\frac{\partial^3 \mathbf{c}(\varkappa_{n-2} = 1)}{\partial \varkappa_{n-2}^3} = \frac{\partial^3 \mathbf{c}(\varkappa_{n-1} = 0)}{\partial \varkappa_{n-1}^3} \tag{56}$$

$$-\mathbf{Q}_{n-2} + 3\mathbf{Q}_{n-1} - 3\mathbf{Q}_n + \mathbf{Q}_{n+1} = -\mathbf{Q}_{n-1} + 3\mathbf{Q}_n - 3\mathbf{Q}_{n+1} + \mathbf{Q}_{n+2} \tag{57}$$

$$0 = -\mathbf{Q}_{n-2} + 4\mathbf{Q}_{n-1} - 6\mathbf{Q}_n + 4\mathbf{Q}_{n+1} - \mathbf{Q}_{n+2} \tag{58}$$

This end condition therefore links the first five and last five control points. The matrix relation between fit knots and control points is finally completed by adding equation (55) and (58).

$$\begin{bmatrix} 0 \\ \mathbf{P}_1 \\ \mathbf{P}_2 \\ \vdots \\ \mathbf{P}_{n-1} \\ \mathbf{P}_n \\ 0 \end{bmatrix} = \frac{1}{6} \begin{bmatrix} -1 & 4 & -6 & 4 & -1 & 0 & \dots & 0 \\ 1 & 4 & 1 & 0 & \dots & \dots & \dots & 0 \\ 0 & 1 & 4 & 1 & 0 & \dots & \dots & \vdots \\ \vdots & 0 & 1 & 4 & 1 & 0 & \dots & \vdots \\ \vdots & 0 & \ddots & \ddots & \ddots & 0 & \dots & \vdots \\ \vdots & \dots & 0 & 1 & 4 & 1 & 0 & \vdots \\ \vdots & \dots & \dots & 0 & 1 & 4 & 1 & 0 \\ 0 & \dots & \dots & \dots & 0 & 1 & 4 & 1 \\ 0 & \dots & \dots & 0 & -1 & 4 & -6 & 4 & -1 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_1 \\ \mathbf{Q}_2 \\ \mathbf{Q}_3 \\ \vdots \\ \mathbf{Q}_n \\ \mathbf{Q}_{n+1} \\ \mathbf{Q}_{n+2} \end{bmatrix} \tag{59}$$

1.5 Positioning of control points

A unique set of control points can be evaluated once enough equations have been written from interpolation equations and end condition constraints. Regardless of the type of end conditions, vector \mathbf{P}^* , \mathbf{Q} and matrix Φ^* now have consistent dimensions to find a unique solution to control points. The control point coordinates are calculated by inverting equation (32) in equation (60).

$$\mathbf{Q} = 6\Phi^{*-1}\mathbf{P}^* \tag{60}$$

1.6 Plotting the cubic spline

Once the position of the control points is known, the parametric curve can be plotted patch by patch referring to equation (25). As seen before, the different patches will be drawn by changing the index i while the intrinsic parameter evolves from 0 to 1 on each patch. Different configurations are considered

in the following paragraphs bringing the different types of end conditions out. The same arbitrary distribution of eight fit points (\bullet), given in figure 5, will be used throughout the examples. Calculated control point coordinates are displayed in (\bullet) while cubic splines are represented in (—). In order to provide an open source and reproducible work base, figures are accompanied by hyperlinks which give access to their source code. Provided routines are written in Python language and summarize the whole procedure to plot a cubic spline from a set of fit knots with different types of end conditions.

1.6.1 free end condition

The shape of the parametric cubic spline displayed in (—) in figure 5 depends on the interpolation on fit knots (\bullet) while considering free end conditions at both extremities. Due to the choice of free end conditions, control points Q_1 , Q_2 and Q_3 are well aligned. As a consequence, the direction of the tangent is imposed by the line passing through these three control points. It is also apparent that the fit knot P_1 and the control point Q_2 are confused. The same behavior is observed at the other end. The source code used to plot a similar figure is provided as an hyperlink just below.

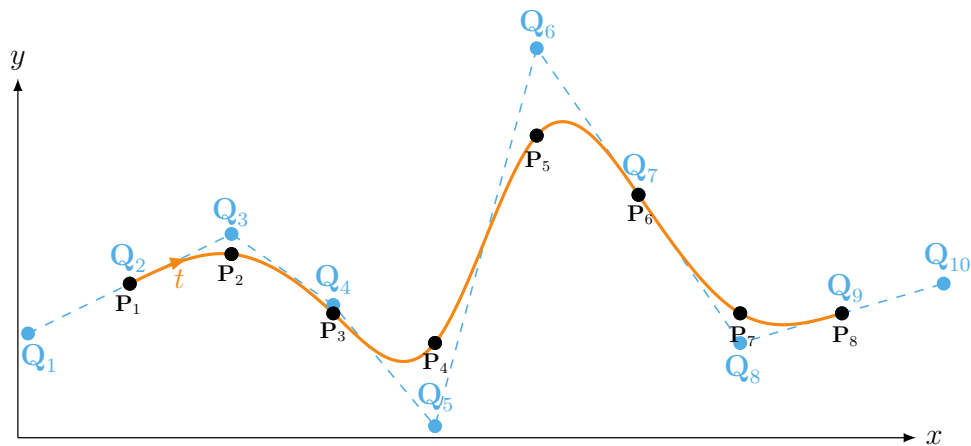


Figure 5. parametric cubic spline interpolated on fit knots with free end condition



The Python source code for generating the cubic spline interpolation of a given set of fit knots with **free end conditions** is available here: https://lava-wiki.meca.polymtl.ca/_media/public/notespline/cubic_spline_free_end.py

1.6.2 Imposed tangent direction

Different parametric cubic splines are plotted in figure 6 generated with different tangent angles as end conditions: (—) 0 , (—) $\pm\pi/4$ and (—) $\pm\pi/2$. Tangent vector are the same at both ends, and successively take the values $T = [0, -1]$, $[1, -1]$, $[1, 0]$, $[1, 1]$ and $[0, 1]$. For the sake of readability, only one set of control points (\bullet) is plotted corresponding to the tangent of zero angle. It is obvious that the position of the control points also changes with the tangent.

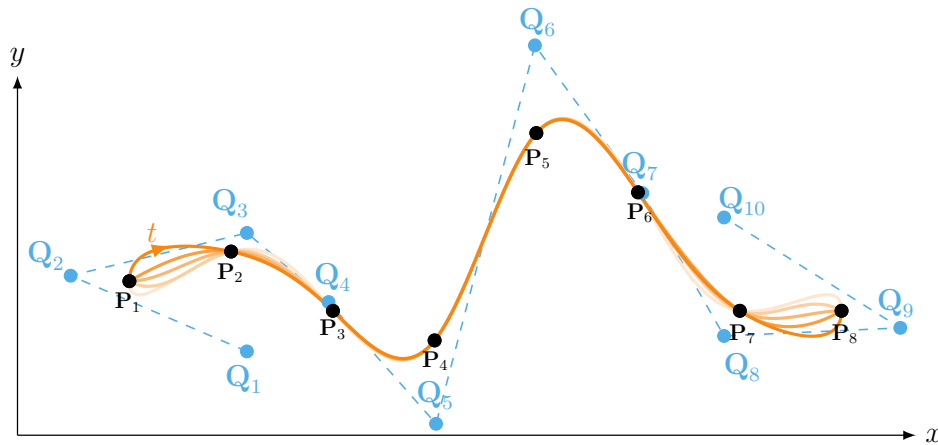


Figure 6. parametric cubic spline interpolated on fit knots with different tangent angles: $\pi/2$ (—), $\pi/4$ (—), 0 (—), $-\pi/4$ (—), $-\pi/2$ (—)

Considering a fixed direction, $\pi/2$ as example, the length of the tangent vector also have an impact on the shape of the parametric curve. Indeed, the coordinates of the last an antepenultimate control points also change related to the norm of the tangent vector. Figure 6 displays the different parametric curves obtained while successively doubling the length of the reference tangent vector $T = [0, 1]$ to $[0, 2]$, $[0, 4]$ and $[0, 8]$.

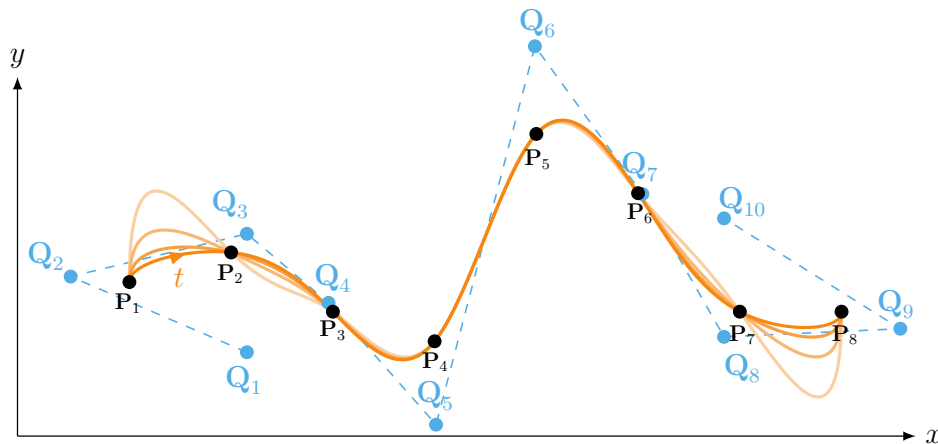


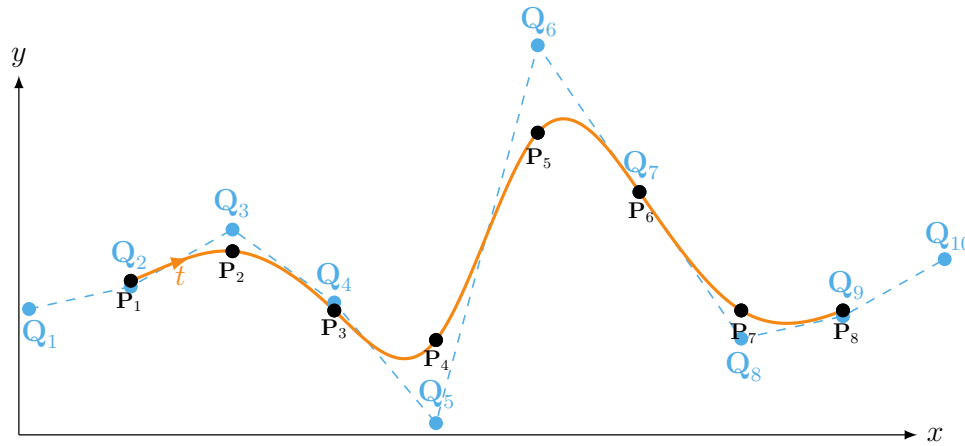
Figure 7. parametric cubic spline interpolated on fit knots with different tangent vectors: $T = [0, 1]$ (—), $T = [0, 2]$ (—), $T = [0, 4]$ (—) and $T = [0, 8]$ (—), control points (•) for $T = [0, 1]$ only



The Python source code for generating the cubic spline interpolation of a given set of fit knots with **imposed tangent direction** is available here: https://lava-wiki.meca.polymtl.ca/_media/public/notespline/cubic_spline_imposed_tangent_direction.py

1.6.3 end-to-end tangent continuity

Fit knot (●), control points (●) and parametric curve (—) with end-to-end connexion condition are displayed in figure 8. The continuity of the tangent is well observed with the same slope at fit knots P_1 and P_8 . It is also highlighted that the relative positions of the first three controls are the same on each side. This ensures a complementary generation of the spline patches at the ends which is the expression of the first and second derivative continuities.



(a) raccord conditions

Figure 8. parametric cubic spline interpolated on fit knots with end-to-end tangent continuity



The Python source code for generating the cubic spline interpolation of a given set of fit knots with **end-to-end tangent continuity** is available here: https://lawa-wiki.meca.polymtl.ca/_media/public/notespline/cubic_spline_end-to-end_tangent_continuity.py

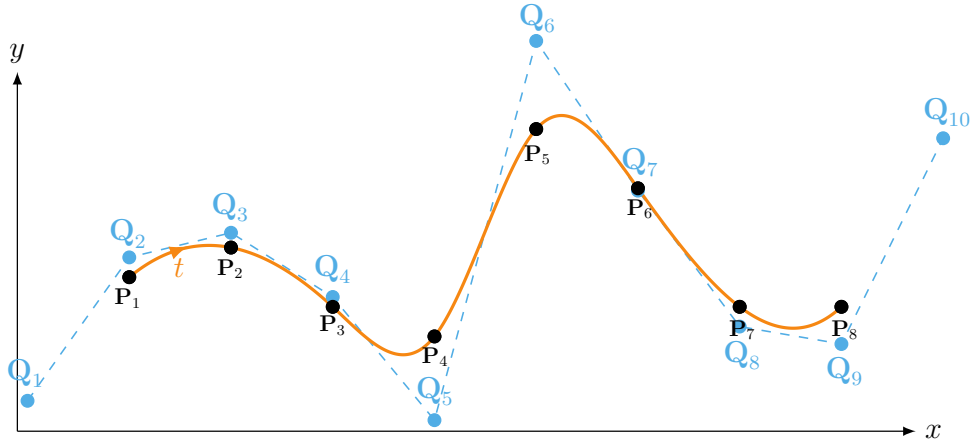
1.6.4 not-a-knot end condition

The shape of the cubic spline (—) obtained by solving equation (59) is plotted in figure 9. Here, control point coordinates (●) simultaneously respect the interpolation on fit knots (●) and *not-a-knot* end conditions. If specific conditions are not requested or if the user does not know what to impose, *not-a-knot* end conditions are an interesting solution to fix smooth shapes at the spline extremities.

2 Three-dimensional bicubic parametric surface

2.1 Definition

A parametric surface is an extension of section 1 applied on a two-dimensional grid of knots. Thereby, third order polynomials, with the same degree $d = 3$, are used in both directions to shape a bicubic parametric surface. The only assumption concerns the particular organization of the knots along both directions \vec{x} and \vec{y} . Indeed, having an organized mesh grid of knots is essential to build a parametric surface. A grid of knots is given as example in (●) in figure 10a displayed in the orthonormal coordinate



(a) not-a-knot

Figure 9. parametric cubic spline interpolated on fit knots with *not-a-knot* end conditions



The Python source code for generating the cubic spline interpolation of a given set of fit knots with **not-a-knot end conditions** is available here: https://lava-wiki.meca.polymtl.ca/_media/public/notespline/cubic_spline_not-a-knot.py

system $\langle \vec{x}; \vec{y}; \vec{z} \rangle$. m is the number of knots along direction \vec{x} while n is the number of knots along direction \vec{y} . This distribution of knots exhibits a discretization of the parametric surface problem in patches. One patch is defined between four knots at its corners. The parametric surface is then composed of $m - 1$ and $n - 1$ patches in their respective directions.

It is assumed that there is an intrinsic parameterization of the space such as the set of knots belong to a unique plane $\langle \vec{u}; \vec{v} \rangle$. The position covered on the parametric surface is driven by the two intrinsic parameters u and v . \mathbf{u} and \mathbf{v} denotes the uniform knot vectors allowing the construction of the parametric surface on knots. The value u_i is reached as crossing the i -th row of knots, see figure 10b. According to the same logic, v_j is reached as crossing the j -th column of fit knots.

$$\mathbf{u} = [u_1 \ u_2 \ u_3 \ \dots \ u_{n-1} \ u_n] \tag{61}$$

$$\mathbf{v} = [v_1 \ v_2 \ v_3 \ \dots \ v_{n-1} \ v_n] \tag{62}$$

Consequently, one elementary patch is defined between intervals $u \in [u_i, u_{i+1}]$ and $v \in [v_j, v_{j+1}]$. The limits of such intervals match the patch corner coordinates, and all patches have the same area in the parametric space. Each patch describes a rectangular domain within knots, which are now regularly spaced. Knot vectors u and v can also be chosen as two increasing lists of integers where values, respectively reach as passing through a row or a column in the mesh grid of knots. Note that parameters

u and v start at zero and reach respectively to $m - 1$ and $n - 1$ at the opposite end of the parametric surface.

$$\mathbf{u} = [0 \ 1 \ 2 \ 3 \ \dots \ (m - 2) \ m - 1] \tag{63}$$

$$\mathbf{v} = [0 \ 1 \ 2 \ 3 \ \dots \ (n - 2) \ n - 1] \tag{64}$$

The general formulation of a parametric surface using bicubic spline is given in equation 65.

$$\mathbf{S}(u, v) = \sum_i^{i+d} \sum_j^{j+d} \mathbf{Q}_{ij} B_{i,d}(u) B_{j,d}(v) \tag{65}$$

where:

- u is the intrinsic parameter of a spatial system that follows the parametric surface in the first direction,
- v is the intrinsic parameter of a spatial system that follows the parametric surface in the second direction,
- $\mathbf{S}(u, v) = [x_S(u, v) \ y_S(u, v) \ z_S(u, v)]$ are the parametric surface coordinates in the Cartesian coordinate system evaluated at (u, v) ,
- d is the degree of the spline in both direction,
- $\mathbf{Q}_{i,j}$ are the control point coordinates,
- $B_{i,d}(u)$ and $B_{j,d}(v)$ are d -degree B-spline functions for their respective intrinsic parameters.

$\mathbf{S}(u, v)$ is a function of class C^1 since bicubic splines guarantee the continuity of the first derivatives in all of the parametric surface, except at the borders. Unlike the two-dimensional case, the second derivatives could be discontinuous but cross-derivative is still true.

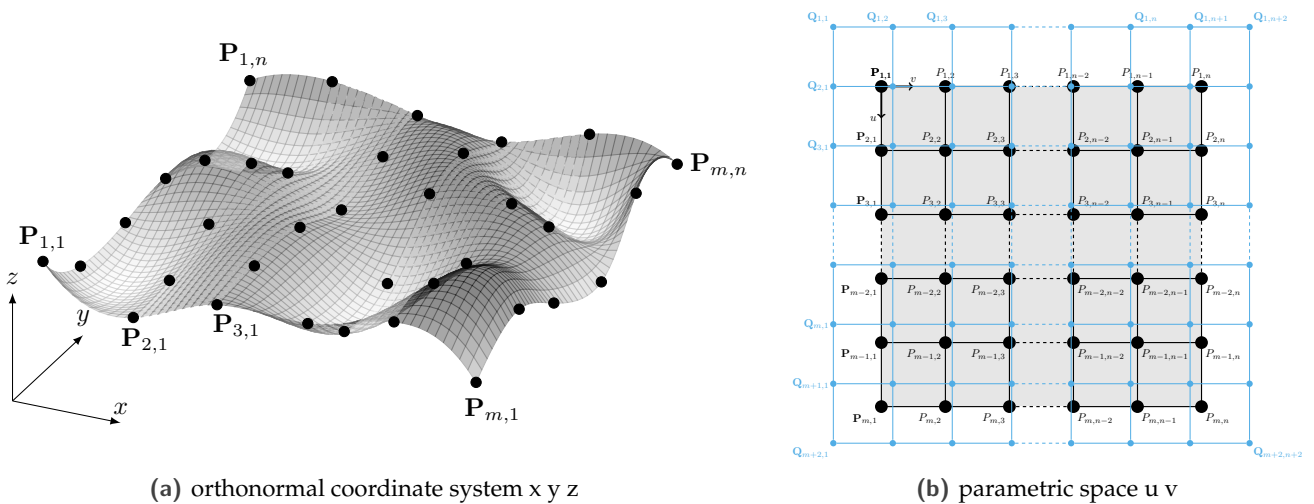


Figure 10. bi-spline surface interpolated on fit knots (•)

2.2 Influence of control points

Each patch of a parametric surface is built as a linear combination of basis functions (B-spline functions) whose influence is driven by a set of neighboring control points $\mathbf{Q}_{i,j}$. A unique combination of directional B-spline functions of the same order gives a unique shape to a patch. The shape of the parametric surface using bicubic splines is so conditioning by the choice of a control point vector \mathbf{Q} . The dimensions of \mathbf{Q} are $(m + 2) \times (n + 2)$ and depend on the number of knots in both directions. Each control point $\mathbf{Q}_{i,j}$ corresponds to a coordinate set $[\hat{x}_{i,j} \ \hat{y}_{i,j} \ \hat{z}_{i,j}]$. Likewise, control points are arranged as a two-dimensional regular grid in the plane $\langle \vec{u}; \vec{v} \rangle$. By construction, \mathbf{Q} is also a uniform organized mesh grid regarding the control point indexes. This space transformation is represented in figure 10b by superimposing control points (•) to square shaped patches delimited by knots at each corner.

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_{1,1} \\ \mathbf{Q}_{1,2} \\ \mathbf{Q}_{1,3} \\ \vdots \\ \mathbf{Q}_{m+2,n} \\ \mathbf{Q}_{m+2,n+1} \\ \mathbf{Q}_{m+2,n+2} \end{bmatrix} \quad (66)$$

Even for parametric surfaces, control points are representative of the polynomial coefficients which set the shape of the surface patches. However, their influences are restricted to a limited number of intervals by definition of B-splines functions. The parametric surface equation is developed to be solved—patch by patch—as an elementary problem. Thus, it is possible to identify the local contribution of the control points on the shape of one particular patch.

Back to equation (65), intrinsic parameters u and v are independent since they operate separately in the calculations of functions $B_{i,3}(u)$ and $B_{j,3}(v)$. The cubic B-spline functions defined by the Cox-de-Boor algorithm in equation (11) can be reused by only changing the notation of the intrinsic parameter. Hence, equations (67) and (68) are defined by their respective intrinsic parameters u and v over various intervals.

$$B_{i,3}(u) = \begin{cases} \frac{u-u_i}{u_{i+3}-u_i} \frac{u-u_i}{u_{i+2}-u_i} \frac{u-u_i}{u_{i+1}-u_i} & \text{if } u \in [u_i, u_{i+1}] \\ \frac{u-u_i}{u_{i+3}-u_i} \left[\frac{u-u_i}{u_{i+2}-u_i} \frac{u_{i+2}-u}{u_{i+2}-u_{i+1}} + \frac{u_{i+3}-u}{u_{i+3}-u_{i+1}} \frac{u-u_{i+1}}{u_{i+2}-u_{i+1}} \right] + \frac{u_{i+4}-u}{u_{i+4}-u_{i+1}} \frac{u-u_{i+1}}{u_{i+3}-u_{i+1}} \frac{u-u_{i+1}}{u_{i+2}-u_{i+1}} & \text{if } u \in [u_{i+1}, u_{i+2}] \\ \frac{u-u_i}{u_{i+3}-u_i} \frac{u_{i+3}-u}{u_{i+3}-u_{i+1}} \frac{u_{i+3}-u}{u_{i+3}-u_{i+2}} + \frac{u_{i+4}-u}{u_{i+4}-u_{i+1}} \left[\frac{u-u_{i+1}}{u_{i+3}-u_{i+1}} \frac{u_{i+3}-u}{u_{i+3}-u_{i+2}} + \frac{u_{i+4}-u}{u_{i+4}-u_{i+2}} \frac{u-u_{i+2}}{u_{i+3}-u_{i+2}} \right] & \text{if } u \in [u_{i+2}, u_{i+3}] \\ \frac{u_{i+4}-u}{u_{i+4}-u_{i+1}} \frac{u_{i+4}-u}{u_{i+4}-u_{i+2}} \frac{u_{i+4}-u}{u_{i+4}-u_{i+3}} & \text{if } u \in [u_{i+3}, u_{i+4}] \\ 0 & \text{otherwise} \end{cases} \quad (67)$$

$$B_{j,3}(v) = \begin{cases} \frac{v-v_j}{v_{j+3}-v_j} \frac{v-v_j}{v_{j+2}-v_j} \frac{v-v_j}{v_{j+1}-v_j} & \text{if } v \in [v_j, v_{j+1}] \\ \frac{v-v_j}{v_{j+3}-v_j} \left[\frac{v-v_j}{v_{j+2}-v_j} \frac{v_{j+2}-v}{v_{j+2}-v_{j+1}} + \frac{v_{j+3}-v}{v_{j+3}-v_{j+1}} \frac{v-v_{j+1}}{v_{j+2}-v_{j+1}} \right] + \frac{v_{j+4}-v}{v_{j+4}-v_{j+1}} \frac{v-v_{j+1}}{v_{j+3}-v_{j+1}} \frac{v-v_{j+1}}{v_{j+2}-v_{j+1}} & \text{if } v \in [v_{j+1}, v_{j+2}] \\ \frac{v-v_j}{v_{j+3}-v_j} \frac{v_{j+3}-v}{v_{j+3}-v_{j+1}} \frac{v_{j+3}-v}{v_{j+3}-v_{j+2}} + \frac{v_{j+4}-v}{v_{j+4}-v_{j+1}} \left[\frac{v-v_{j+1}}{v_{j+3}-v_{j+1}} \frac{v_{j+3}-v}{v_{j+3}-v_{j+2}} + \frac{v_{j+4}-v}{v_{j+4}-v_{j+2}} \frac{v-v_{j+2}}{v_{j+3}-v_{j+2}} \right] & \text{if } v \in [v_{j+2}, v_{j+3}] \\ \frac{v_{j+4}-v}{v_{j+4}-v_{j+1}} \frac{v_{j+4}-v}{v_{j+4}-v_{j+2}} \frac{v_{j+4}-v}{v_{j+4}-v_{j+3}} & \text{if } v \in [v_{j+3}, v_{j+4}] \\ 0 & \text{otherwise} \end{cases} \quad (68)$$

Following a similar approach, some simplifications are made to set up an elementary problem for each patch identified with indexes i, j :

- changes of variables,

$$\mu_i = u - u_i \quad (69)$$

$$v_j = v - v_j \quad (70)$$

- homogeneous properties of the two intrinsic knot vector,

$$u_{i+1} - u_i = 1 \quad (71)$$

$$v_{j+1} - v_j = 1 \quad (72)$$

- uniform properties of parameters μ_i and v_i ,

$$\mu_{i+1} = \mu_i - 1 \quad (73)$$

$$v_{j+1} = v_j - 1 \quad (74)$$

Following the study carried out in section 1.2 related to the intervals of definition of cubic B-spline functions, the formulation for a patch of indexes i, j of a bicubic parametric surface becomes:

$$\mathbf{S}(\mu_i, v_j) = \sum_i^{i+3} \sum_j^{j+3} \mathbf{Q}_{i,j} B_{i,3}(\mu_i) B_{j,3}(v_j) \quad (75)$$

Writing as in equation 76, the bicubic parametric surface formulation shows that up to 16 control points can affect the shape of a patch.

$$\begin{aligned}
 \mathbf{S}(\mu_i, v_j) = & B_{j,3}(v_j) [B_{i,3}(\mu_i) \mathbf{Q}_{i,j} + B_{i+1,3}(\mu_{i+1}) \mathbf{Q}_{i+1,j} + B_{i+2,3}(\mu_{i+2}) \mathbf{Q}_{i+2,j} + B_{i+3,3}(\mu_{i+3}) \mathbf{Q}_{i+3,j}] \\
 & + B_{j+1,3}(v_{j+1}) [B_{i,3}(\mu_i) \mathbf{Q}_{i,j+1} + B_{i+1,3}(\mu_{i+1}) \mathbf{Q}_{i+1,j+1} + B_{i+2,3}(\mu_{i+2}) \mathbf{Q}_{i+2,j+1} + B_{i+3,3}(\mu_{i+3}) \mathbf{Q}_{i+3,j+1}] \\
 & + B_{j+2,3}(v_{j+2}) [B_{i,3}(\mu_i) \mathbf{Q}_{i,j+2} + B_{i+1,3}(\mu_{i+1}) \mathbf{Q}_{i+1,j+2} + B_{i+2,3}(\mu_{i+2}) \mathbf{Q}_{i+2,j+2} + B_{i+3,3}(\mu_{i+3}) \mathbf{Q}_{i+3,j+2}] \\
 & + B_{j+3,3}(v_{j+3}) [B_{i,3}(\mu_i) \mathbf{Q}_{i,j+3} + B_{i+1,3}(\mu_{i+1}) \mathbf{Q}_{i+1,j+3} + B_{i+2,3}(\mu_{i+2}) \mathbf{Q}_{i+2,j+3} + B_{i+3,3}(\mu_{i+3}) \mathbf{Q}_{i+3,j+3}]
 \end{aligned}$$

(76)

Cubic B-spline functions are resumed from the equations 16 to 19 with the proper changes of variables.

$$B_{i,3}(\mu_i) = \frac{1}{6} \begin{cases} \mu_i^3 & \text{if } \mu_i \in [0, 1] \\ -3\mu_i^3 + 12\mu_i^2 - 12\mu_i + 4 & \text{if } \mu_i \in [1, 2] \\ 3\mu_i^3 - 24\mu_i^2 + 60\mu_i - 44 & \text{if } \mu_i \in [2, 3] \\ (4 - \mu_i)^3 & \text{if } \mu_i \in [3, 4] \\ 0 & \text{otherwise} \end{cases} \quad (77)$$

$$B_{j,3}(v_j) = \frac{1}{6} \begin{cases} v_j^3 & \text{if } v_j \in [0, 1] \\ -3v_j^3 + 12v_j^2 - 12v_j + 4 & \text{if } v_j \in [1, 2] \\ 3v_j^3 - 24v_j^2 + 60v_j - 44 & \text{if } v_j \in [2, 3] \\ (4 - v_j)^3 & \text{if } v_j \in [3, 4] \\ 0 & \text{otherwise} \end{cases}$$

$$B_{i+1,3}(\mu_i) = \frac{1}{6} \begin{cases} (\mu_i - 1)^3 & \text{if } \mu_i \in [1, 2] \\ -3(\mu_i - 1)^3 + 12(\mu_i - 1)^2 - 12(\mu_i - 1) + 4 & \text{if } \mu_i \in [2, 3] \\ 3(\mu_i - 1)^3 - 24(\mu_i - 1)^2 + 60(\mu_i - 1) - 44 & \text{if } \mu_i \in [3, 4] \\ (4 - \mu_i - 1)^3 & \text{if } \mu_i \in [4, 5] \\ 0 & \text{otherwise} \end{cases} \quad (78)$$

$$B_{j+1,3}(v_j) = \frac{1}{6} \begin{cases} (v_j - 1)^3 & \text{if } v_j \in [1, 2] \\ -3(v_j - 1)^3 + 12(v_j - 1)^2 - 12(v_j - 1) + 4 & \text{if } v_j \in [2, 3] \\ 3(v_j - 1)^3 - 24(v_j - 1)^2 + 60(v_j - 1) - 44 & \text{if } v_j \in [3, 4] \\ (4 - v_j - 1)^3 & \text{if } v_j \in [4, 5] \\ 0 & \text{otherwise} \end{cases}$$

$$B_{i+2,3}(\mu_i) = \frac{1}{6} \begin{cases} (\mu_i - 2)^3 & \text{if } \mu_i \in [2, 3] \\ -3(\mu_i - 2)^3 + 12(\mu_i - 2)^2 - 12(\mu_i - 2) + 4 & \text{if } \mu_i \in [3, 4] \\ 3(\mu_i - 2)^3 - 24(\mu_i - 2)^2 + 60(\mu_i - 2) - 44 & \text{if } \mu_i \in [4, 5] \\ (4 - \mu_i - 2)^3 & \text{if } \mu_i \in [5, 6] \\ 0 & \text{otherwise} \end{cases} \quad (79)$$

$$B_{j+2,3}(v_j) = \frac{1}{6} \begin{cases} (v_j - 2)^3 & \text{if } v_j \in [2, 3] \\ -3(v_j - 2)^3 + 12(v_j - 2)^2 - 12(v_j - 2) + 4 & \text{if } v_j \in [3, 4] \\ 3(v_j - 2)^3 - 24(v_j - 2)^2 + 60(v_j - 2) - 44 & \text{if } v_j \in [4, 5] \\ (4 - v_j - 2)^3 & \text{if } v_j \in [5, 6] \\ 0 & \text{otherwise} \end{cases}$$

$$B_{i+3,3}(\mu_i) = \frac{1}{6} \begin{cases} (\mu_i - 3)^3 & \text{if } \mu_i \in [3,4] \\ -3(\mu_i - 3)^3 + 12(\mu_i - 3)^2 - 12(\mu_i - 3) + 4 & \text{if } \mu_i \in [4,5] \\ 3(\mu_i - 3)^3 - 24(\mu_i - 3)^2 + 60(\mu_i - 3) - 44 & \text{if } \mu_i \in [5,6] \\ (4 - \mu_i - 3)^3 & \text{if } \mu_i \in [6,7] \\ 0 & \text{otherwise} \end{cases} \quad (80)$$

$$B_{j+3,3}(\nu_j) = \frac{1}{6} \begin{cases} (\nu_j - 3)^3 & \text{if } \nu_j \in [3,4] \\ -3(\nu_j - 3)^3 + 12(\nu_j - 3)^2 - 12(\nu_j - 3) + 4 & \text{if } \nu_j \in [4,5] \\ 3(\nu_j - 3)^3 - 24(\nu_j - 3)^2 + 60(\nu_j - 3) - 44 & \text{if } \nu_j \in [5,6] \\ (4 - \nu_j - 3)^3 & \text{if } \nu_j \in [6,7] \\ 0 & \text{otherwise} \end{cases}$$

The interval highlighted in (■) in equations 77 to 80 describes the domain where the shape of the parametric surface is affected by a maximum of control points. Indeed, all cubic B-spline functions are simultaneously defined and non-zero for parameters $\mu_i \in [3,4]$ and $\nu_j \in [3,4]$. Due to the degree of the spline and indexing choices, it is therefore the interval to consider to observe the influence of the 16 control points on the profile of the patch of indexes i, j . One last change of variables is operated to set up an elementary system where the intrinsic parameters μ_i and ν_i evolve from 0 to 1.

$$\mu_i = \mu_i - 3 \quad (81)$$

$$\nu_j = \nu_j - 3 \quad (82)$$

Cubic B-spline functions are finally written for $\mu_i \in [0,1]$ and $\nu_j \in [0,1]$ in equations 83 to 86. As part of this development, it is also the opportunity to express them in matrix form:

$$B_{i,3}(\mu_i) = \frac{1}{6}(1 - \mu_i)^3 = \frac{1}{6} \begin{bmatrix} -1 & 3 & -3 & 1 \end{bmatrix} \begin{bmatrix} \mu_i^3 \\ \mu_i^2 \\ \mu_i \\ 1 \end{bmatrix} \quad (83)$$

$$B_{j,3}(\nu_j) = \frac{1}{6}(1 - \nu_j)^3 = \frac{1}{6} \begin{bmatrix} -1 \\ 3 \\ -3 \\ 1 \end{bmatrix} \begin{bmatrix} \nu_j^3 & \nu_j^2 & \nu_j & 1 \end{bmatrix}$$

$$B_{i+1,3}(\mu_i) = \frac{1}{6}(3\mu_i^3 - 6\mu_i^2 + 4) = \frac{1}{6} \begin{bmatrix} 3 & -6 & 0 & 4 \end{bmatrix} \begin{bmatrix} \mu_i^3 \\ \mu_i^2 \\ \mu_i \\ 1 \end{bmatrix} \quad (84)$$

$$B_{j+1,3}(\nu_j) = \frac{1}{6}(3\nu_j^3 - 6\nu_j^2 + 4) = \frac{1}{6} \begin{bmatrix} \nu_j^3 & \nu_j^2 & \nu_j & 1 \end{bmatrix} \begin{bmatrix} 3 \\ -6 \\ 0 \\ 4 \end{bmatrix}$$

$$\begin{aligned}
 B_{i+2,3}(\mu_i) &= \frac{1}{6}(-3\mu_i^3 + 3\mu_i^2 + 3\mu_i + 1) = \frac{1}{6} \begin{bmatrix} -3 & 3 & 3 & 1 \end{bmatrix} \begin{bmatrix} \mu_i^3 \\ \mu_i^2 \\ \mu_i \\ 1 \end{bmatrix} \\
 B_{j+2,3}(\nu_j) &= \frac{1}{6}(-3\nu_j^3 + 3\nu_j^2 + 3\nu_j + 1) = \frac{1}{6} \begin{bmatrix} \nu_j^3 & \nu_j^2 & \nu_j & 1 \end{bmatrix} \begin{bmatrix} -3 \\ 3 \\ 3 \\ 1 \end{bmatrix}
 \end{aligned} \tag{85}$$

$$\begin{aligned}
 B_{i+3,3}(\mu_i) &= \frac{1}{6}\mu_i^3 = \frac{1}{6} \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mu_i^3 \\ \mu_i^2 \\ \mu_i \\ 1 \end{bmatrix} \\
 B_{j+3,3}(\nu_j) &= \frac{1}{6}\nu_j^3 = \frac{1}{6} \begin{bmatrix} \nu_j^3 & \nu_j^2 & \nu_j & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}
 \end{aligned} \tag{86}$$

Last but not least, the parametric surface formulation $\mathbf{S}(\mu_i, \nu_j)$ for the patch of indexes i, j is written in a well-known matrix form in equation 87. This matrix formulation will serve as a basis for the problem of interpolation of a bicubic parametric surface and the definition of end conditions in the following sections.

$$\mathbf{S}(\mu_i, \nu_j) = \frac{1}{36} \begin{bmatrix} \nu_j^3 & \nu_j^2 & \nu_j & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{i,j} & \mathbf{Q}_{i+1,j} & \mathbf{Q}_{i+2,j} & \mathbf{Q}_{i+3,j} \\ \mathbf{Q}_{i,j+1} & \mathbf{Q}_{i+1,j+1} & \mathbf{Q}_{i+2,j+1} & \mathbf{Q}_{i+3,j+1} \\ \mathbf{Q}_{i,j+2} & \mathbf{Q}_{i+1,j+2} & \mathbf{Q}_{i+2,j+2} & \mathbf{Q}_{i+3,j+2} \\ \mathbf{Q}_{i,j+3} & \mathbf{Q}_{i+1,j+3} & \mathbf{Q}_{i+2,j+3} & \mathbf{Q}_{i+3,j+3} \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 0 & 4 \\ -3 & 3 & 3 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mu_i^3 \\ \mu_i^2 \\ \mu_i \\ 1 \end{bmatrix} \tag{87}$$

2.3 Interpolation equations for a parametric surface

This section deals with the interpolation of a bicubic parametric surface from a set of fit knots. Given a set of $m \times n$ fit knots $\mathbf{P}_{i,j}$ to be interpolated, with index $i \in [1, m]$ and $j \in [1, n]$. Each fit knots $\mathbf{P}_{i,j}$ corresponds to a coordinate vector $\mathbf{P}_{i,j} = \begin{bmatrix} x_{i,j} & y_{i,j} & z_{i,j} \end{bmatrix}$ in the space $\langle \vec{x}; \vec{y}; \vec{z} \rangle$. The parametric surface $\mathbf{S}(\mu_i, \nu_j)$ is said to be interpolated if passing through all the fit knots. As seen before, the only assumption concerns the

particular organization of the fit knots vector \mathbf{P} of length $m \times n$:

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{1,1} \\ \mathbf{P}_{1,2} \\ \mathbf{P}_{1,3} \\ \vdots \\ \mathbf{P}_{m,n-2} \\ \mathbf{P}_{m,n-1} \\ \mathbf{P}_{m,n} \end{bmatrix} \quad (88)$$

where:

- m is the number of fit knots along direction \vec{x} ,
- n is the number of fit knots along direction \vec{y} .

Order and indexes given to $\mathbf{P}_{i,j}$ are important since they ensure the mapping of the parametric surface on the grid of fit knots. The interpolation problem is addressed patch by patch as position continuity conditions. One patch must be defined between four fit knots $\mathbf{P}_{i,j}$, $\mathbf{P}_{i+1,j}$, $\mathbf{P}_{i+1,j+1}$ and $\mathbf{P}_{i,j+1}$. In practice, the patch corners have to locally correspond to fit knot positions, and this must be true for any couple of indexes $i \in [1, m]$ and $j \in [1, n]$. The shape of the parametric surface is so conditioned by interpolation equations which allow this condition to be verified. The interpolation of a bicubic parametric surface is translated into equations 89 to 92 depending on the fit knot to be treated and its position on the studied patch. Notations u^- , u^+ , v^- , v^+ are introduced to identify the different borders of a single patch, with reference to their relative position in the parametrized coordinate system $\langle \vec{u}; \vec{v} \rangle$.

fit knots $\mathbf{P}_{i,j}$ for $i \in [1, m-1]$ and $j \in [1, n-1]$ matching patch i, j at corner u^-v^- :

$$\mathbf{P}_{i,j} = \mathbf{S}(\mu_i = 0, \nu_j = 0) \quad (89)$$

fit knots $\mathbf{P}_{i,n}$ for $i \in [1, m-1]$ matching patch $i, n-1$ at corner u^-v^+ :

$$\mathbf{P}_{i,n} = \mathbf{S}(\mu_i = 0, \nu_{n-1} = 1) \quad (90)$$

fit knots $\mathbf{P}_{m,j}$ for $j \in [1, n-1]$ matching patch $m-1, j$ at corner u^+v^- :

$$\mathbf{P}_{m,j} = \mathbf{S}(\mu_{m-1} = 1, \nu_j = 0) \quad (91)$$

fit knots $\mathbf{P}_{m,n}$ matching patch $m-1, n-1$ at corner u^+v^+ :

$$\mathbf{P}_{m,n} = \mathbf{S}(\mu_i = 1, \nu_j = 1) \quad (92)$$

The development of the system of equations leads to a simple formulation of the interpolation problem,

which is generalizable for all fit points $\mathbf{P}_{i,j}$ with $i \in [1, m]$ and $j \in [1, n]$:

$$\left\{ \begin{array}{l} \mathbf{P}_{1,1} = \frac{1}{36} (\mathbf{Q}_{1,1} + 4\mathbf{Q}_{1,2} + \mathbf{Q}_{1,3} + 4\mathbf{Q}_{2,1} + 16\mathbf{Q}_{2,2} + 4\mathbf{Q}_{2,3} + \mathbf{Q}_{3,1} + 4\mathbf{Q}_{3,2} + \mathbf{Q}_{3,3}) \\ \mathbf{P}_{1,2} = \frac{1}{36} (\mathbf{Q}_{1,2} + 4\mathbf{Q}_{1,3} + \mathbf{Q}_{1,4} + 4\mathbf{Q}_{2,2} + 16\mathbf{Q}_{2,3} + 4\mathbf{Q}_{2,4} + \mathbf{Q}_{3,2} + 4\mathbf{Q}_{3,3} + \mathbf{Q}_{3,4}) \\ \mathbf{P}_{1,3} = \frac{1}{36} (\mathbf{Q}_{1,3} + 4\mathbf{Q}_{1,4} + \mathbf{Q}_{1,5} + 4\mathbf{Q}_{2,3} + 16\mathbf{Q}_{2,4} + 4\mathbf{Q}_{2,5} + \mathbf{Q}_{3,3} + 4\mathbf{Q}_{3,4} + \mathbf{Q}_{3,5}) \\ \vdots \\ \mathbf{P}_{i,j} = \frac{1}{36} (\mathbf{Q}_{i,j} + 4\mathbf{Q}_{i,j+1} + \mathbf{Q}_{i,j+2} + 4\mathbf{Q}_{i+1,j} + 16\mathbf{Q}_{i+1,j+1} + 4\mathbf{Q}_{i+1,j+2} + \mathbf{Q}_{i+2,j} + 4\mathbf{Q}_{i+2,j+1} + \mathbf{Q}_{i+2,j+2}) \\ \vdots \\ \mathbf{P}_{m,n-2} = \frac{1}{36} (\mathbf{Q}_{m,n-2} + 4\mathbf{Q}_{m,n-1} + \mathbf{Q}_{m,n} + 4\mathbf{Q}_{m+1,n-2} + 16\mathbf{Q}_{m+1,n-1} + 4\mathbf{Q}_{m+1,n} + \mathbf{Q}_{m+2,n-2} + 4\mathbf{Q}_{m+2,n-1} + \mathbf{Q}_{m+2,n}) \\ \mathbf{P}_{m,n-1} = \frac{1}{36} (\mathbf{Q}_{m,n-1} + 4\mathbf{Q}_{m,n} + \mathbf{Q}_{m,n+1} + 4\mathbf{Q}_{m+1,n-1} + 16\mathbf{Q}_{m+1,n} + 4\mathbf{Q}_{m+1,n+1} + \mathbf{Q}_{m+2,n-1} + 4\mathbf{Q}_{m+2,n} + \mathbf{Q}_{m+2,n+1}) \\ \mathbf{P}_{m,n} = \frac{1}{36} (\mathbf{Q}_{m,n} + 4\mathbf{Q}_{m,n+1} + \mathbf{Q}_{m,n+2} + 4\mathbf{Q}_{m+1,n} + 16\mathbf{Q}_{m+1,n+1} + 4\mathbf{Q}_{m+1,n+2} + \mathbf{Q}_{m+2,n} + 4\mathbf{Q}_{m+2,n+1} + \mathbf{Q}_{m+2,n+2}) \end{array} \right. \quad (93)$$

Also, this system can be put in matrix form $\mathbf{P} = \Phi\mathbf{Q}$. An example is given in equation (94) restricted to the case $n = 4$ and $m = 4$. A specific color code is used to replace values in the matrix.

color code : $\square = 0$, $\blacksquare = 1$, $\blacksquare = 4$ and $\blacksquare = 16$

$$\begin{bmatrix} \mathbf{P}_{1,1} \\ \mathbf{P}_{1,2} \\ \mathbf{P}_{1,3} \\ \vdots \\ \mathbf{P}_{4,3} \\ \mathbf{P}_{4,4} \end{bmatrix} = \frac{1}{36} \begin{bmatrix} \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{1,1} \\ \mathbf{Q}_{1,2} \\ \vdots \\ \mathbf{Q}_{6,5} \\ \mathbf{Q}_{6,6} \end{bmatrix} \quad (94)$$

As it stands, the dimensions of vectors \mathbf{P} and \mathbf{Q} are not homogeneous and it is not possible to find a unique solution to the problem. Indeed, the length of \mathbf{P} is still $m \times n$ while \mathbf{Q} is $(m + 2) \times (n + 2)$.

However, adding end condition equations to constrain the shape of the parametric surface will be sufficient to calculate every control point coordinates $\mathbf{Q}_{i,j}$ by inversion.

2.4 End conditions for parametric surfaces

On the side of interpolation equations, end condition constraints are requested for the calculation of the control point coordinates. As shown in equation (94), $2m + 2n + 4$ relations are missing for the inverse calculation of all \mathbf{Q}_i . One option is to translate end conditions into enough equations to make the solution unique. The easiest way is to find relations between several control points at the neighborhood of the parametric surface edges. If not trivial, these relations can also be established, notably by adding local mathematical properties to the parametric surface. Details for implementing different types of end conditions for the three-dimensional case are given in the following paragraphs. This section deals with:

free end condition involving the symmetry of 3 control points on all the 4 edges,

cylindrical closed parametric surface connecting two opposite edges with continuous first and second derivatives,

spherical closed parametric surface obtained from the cylindrical configuration by adding extra continuous first derivative continuity to completely close a parametric surface,

not-a-knot end condition defined by the continuity of the third derivatives at the junction of the first two patches facing the 4 edges,

2.4.1 Free end condition

Free end conditions are also easy to set up on parametric surfaces since directly transposable from the two-dimensional case. Indeed, relations between control points written at the spline extremities are the same to consider at the parametric surface edges. The first 3 control points on the four edges must be aligned and separated two by two by the same distance. This choice of end conditions for a parametric surface results in the following relations:

edge u^- : for $2 \leq j \leq n + 1$, the control point $\mathbf{Q}_{2,j}$ is the middle of the segment $[\mathbf{Q}_{1,j}, \mathbf{Q}_{3,j}]$,

$$\mathbf{Q}_{1,j} - 2\mathbf{Q}_{2,j} + \mathbf{Q}_{3,j} = 0 \quad (95)$$

edge u^+ : for $2 \leq j \leq n + 1$, the control point $\mathbf{Q}_{m+1,j}$ is the middle of the segment $[\mathbf{Q}_{m,j}, \mathbf{Q}_{m+2,j}]$,

$$\mathbf{Q}_{m,j} - 2\mathbf{Q}_{m+1,j} + \mathbf{Q}_{m+2,j} = 0 \quad (96)$$

edge v^- : for $2 \leq i \leq m + 1$, the control point $\mathbf{Q}_{i,2}$ is the middle of the segment $[\mathbf{Q}_{i,1}, \mathbf{Q}_{i,3}]$,

$$\mathbf{Q}_{i,1} - 2\mathbf{Q}_{i,2} + \mathbf{Q}_{i,3} = 0 \quad (97)$$

edge v^+ : for $2 \leq i \leq m + 1$, the control point $\mathbf{Q}_{i,n+1}$ is the middle of the segment $[\mathbf{Q}_{i,n}, \mathbf{Q}_{i,n+2}]$,

$$\mathbf{Q}_{i,n} - 2\mathbf{Q}_{i,n+1} + \mathbf{Q}_{i,n+2} = 0 \quad (98)$$

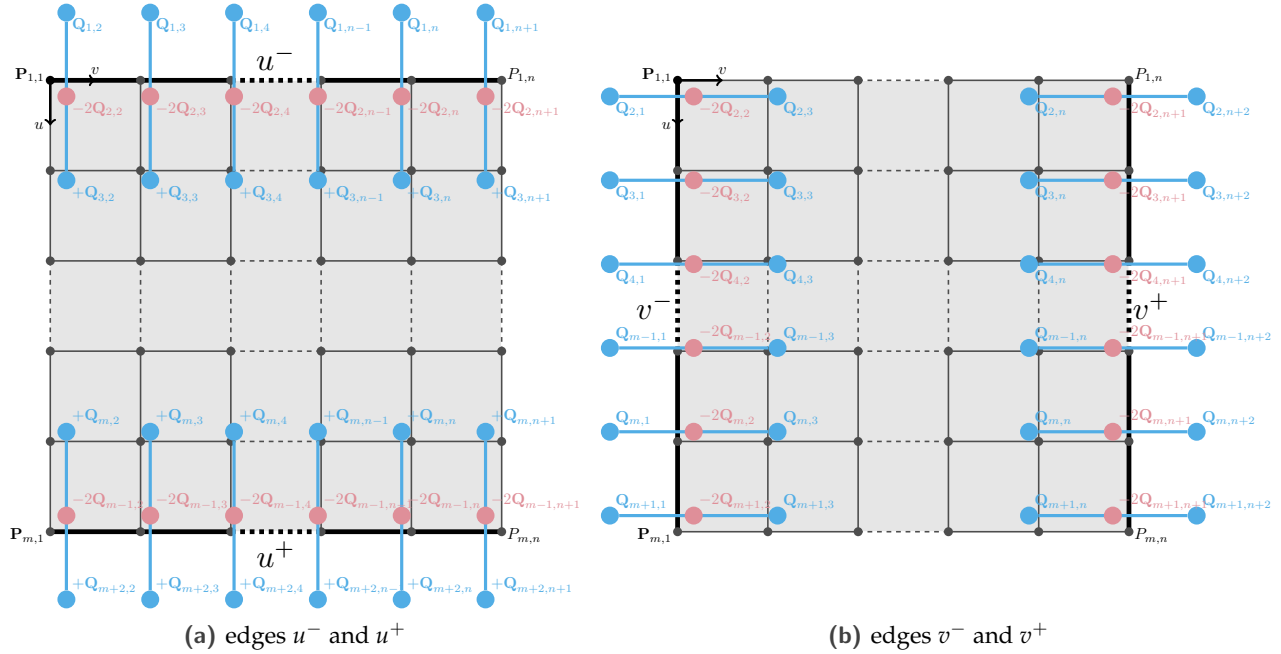


Figure 11. equations between control points for free end conditions written at opposite edges

These relations are schematised in figures 11a and 11b respectively, for both couples of opposite edges. The background represents the bicubic surface in the parametric space. Edges concerned with the free end conditions are highlighted in thicker line (—). Edge u^- refers to the upper end, edge u^+ to the lower end, edge v^- to the left side end, edge v^+ to the right side end. End condition equations are displayed over by continuous lines connecting control points. One line is equivalent to a unique equation. Specific colors are used for each control point with reference to the weight they play in end condition relations.

We notice at this stage that 4 equations are missing. Missing relations necessarily concern the diagonal directions and are obtained in equations (99) to (102) by writing free end conditions at the 4 corners of the parametric surface. Free end condition equations, identified at corners noted u^-v^- , u^-v^+ , u^+v^- and u^+v^+ , are displayed in figures 12.

corner u^-v^- : the control point $Q_{2,2}$ is the middle of the segment $[Q_{1,1}, Q_{3,3}]$,

$$Q_{1,1} - 2Q_{2,2} + Q_{3,3} = 0 \quad (99)$$

corner u^-v^+ : the control point $Q_{2,n+1}$ is the middle of the segment $[Q_{3,n}, Q_{1,n+2}]$,

$$Q_{3,n} - 2Q_{2,n+1} + Q_{1,n+2} = 0 \quad (100)$$

corner u^+v^- : the control point $Q_{m+1,2}$ is the middle of the segment $[Q_{m+2,1}, Q_{m,3}]$,

$$Q_{m+2,1} - 2Q_{m+1,2} + Q_{m,3} = 0 \quad (101)$$

corner u^+v^+ : the control point $\mathbf{Q}_{m+1,n+1}$ is the middle of the segment $[\mathbf{Q}_{m,n}, \mathbf{Q}_{m+2,n+2}]$,

$$\mathbf{Q}_{m,n} - 2\mathbf{Q}_{m+1,n+1} + \mathbf{Q}_{m+2,n+2} = 0 \quad (102)$$

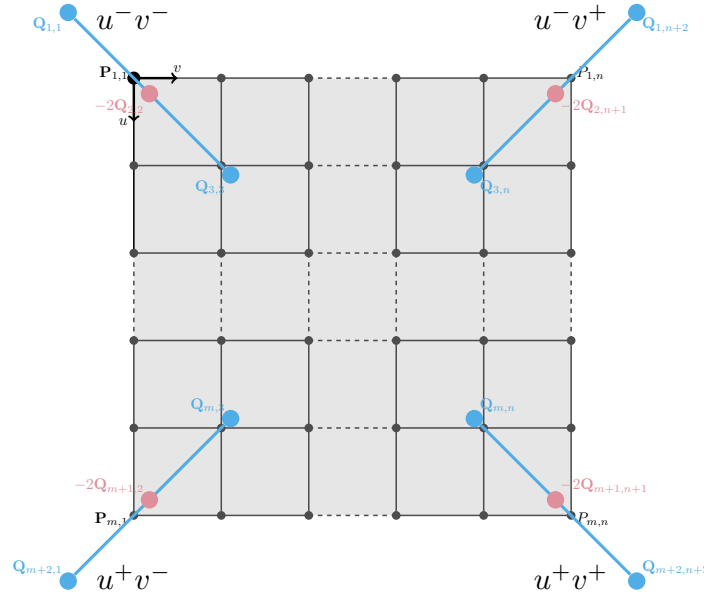


Figure 12. equations between control points for free end conditions written at corners

2.4.2 Cylindrical closed parametric surface

A parametric surface is said to be closed if all the points constituting two opposite edges are confused. In this section only the connection between edges u^- and u^+ is investigated that results in the continuity condition in equation (103).

$$\mathbf{S}(\mu_1 = 0, \nu_j) = \mathbf{S}(\mu_{m-1} = 1, \nu_j) \quad \forall j \in [1, n-1] \quad \text{and} \quad \nu_j \in [0, 1] \quad (103)$$

This equation cannot be used as end conditions since already taken into account in the interpolation problem. Indeed, a position-type end condition is first of all possible only if the fit knots are perfectly superimposed two by two on the studied edges. The construction of the mesh grid of fit knots has to carefully respect the following condition:

$$\mathbf{P}_{1,j} = \mathbf{P}_{m,j} \quad \forall j \in [1, n] \quad (104)$$

However, end conditions should be added between edges u^- and u^+ to replenish properties of a class C^2 differentiable function. Indeed, the first derivative has to be continuous between fitting edges to avoid discontinuities from curve breaking. For further, smoothness of the parametric surface near the fitting zone is relative to the continuity of the second derivative between edges which ensures identical curvatures on both sides. Derivatives are calculated with respect to the direction of the parametric system

which allows to go from one edge to the other. The calculations of the first and second derivatives with respect to μ_i are so detailed in equations (105) and (106) respectively.

$$\frac{\partial \mathbf{S}(\mu_i, \nu_j)}{\partial \mu_i} = \frac{1}{36} \begin{bmatrix} \nu_j^3 & \nu_j^2 & \nu_j & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{i,j} & \mathbf{Q}_{i+1,j} & \mathbf{Q}_{i+2,j} & \mathbf{Q}_{i+3,j} \\ \mathbf{Q}_{i,j+1} & \mathbf{Q}_{i+1,j+1} & \mathbf{Q}_{i+2,j+1} & \mathbf{Q}_{i+3,j+1} \\ \mathbf{Q}_{i,j+2} & \mathbf{Q}_{i+1,j+2} & \mathbf{Q}_{i+2,j+2} & \mathbf{Q}_{i+3,j+2} \\ \mathbf{Q}_{i,j+3} & \mathbf{Q}_{i+1,j+3} & \mathbf{Q}_{i+2,j+3} & \mathbf{Q}_{i+3,j+3} \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 \\ 3 & -6 & 0 \\ -3 & 3 & 3 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 3\mu_i^2 \\ 2\mu_i \\ 1 \end{bmatrix} \quad (105)$$

$$\frac{\partial^2 \mathbf{S}(\mu_i, \nu_j)}{\partial \mu_i^2} = \frac{1}{36} \begin{bmatrix} \nu_j^3 & \nu_j^2 & \nu_j & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{i,j} & \mathbf{Q}_{i+1,j} & \mathbf{Q}_{i+2,j} & \mathbf{Q}_{i+3,j} \\ \mathbf{Q}_{i,j+1} & \mathbf{Q}_{i+1,j+1} & \mathbf{Q}_{i+2,j+1} & \mathbf{Q}_{i+3,j+1} \\ \mathbf{Q}_{i,j+2} & \mathbf{Q}_{i+1,j+2} & \mathbf{Q}_{i+2,j+2} & \mathbf{Q}_{i+3,j+2} \\ \mathbf{Q}_{i,j+3} & \mathbf{Q}_{i+1,j+3} & \mathbf{Q}_{i+2,j+3} & \mathbf{Q}_{i+3,j+3} \end{bmatrix} \begin{bmatrix} -1 & 3 \\ 3 & -6 \\ -3 & 3 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 6\mu_i \\ 2 \end{bmatrix} \quad (106)$$

Equalities of the first two derivatives at specific fit knots are detailed below:

- continuity of the first derivative with respect to μ_i at fit knots $\mathbf{P}_{1,j}$ and $\mathbf{P}_{m,j}$ for $j \in [1, n-1]$:

$$\frac{\partial \mathbf{S}(\mu_1 = 0, \nu_j = 0)}{\partial \mu_1} = \frac{\partial \mathbf{S}(\mu_{m-1} = 1, \nu_j = 0)}{\partial \mu_{m-1}} \quad (107)$$

$$\begin{aligned} -3\mathbf{Q}_{1,j} + 3\mathbf{Q}_{3,j} - 12\mathbf{Q}_{1,j+1} + 12\mathbf{Q}_{3,j+1} - 3\mathbf{Q}_{1,j+2} + 3\mathbf{Q}_{3,j+2} + 3\mathbf{Q}_{m,j} - 3\mathbf{Q}_{m+2,j} \\ + 12\mathbf{Q}_{m,j+1} - 12\mathbf{Q}_{m+2,j+1} + 3\mathbf{Q}_{m,j+2} - 3\mathbf{Q}_{m+2,j+2} = 0 \end{aligned} \quad (108)$$

- continuity of the first derivative with respect to μ_i at the last fit knots $\mathbf{P}_{1,n}$ and $\mathbf{P}_{m,n}$

$$\frac{\partial \mathbf{S}(\mu_1 = 0, \nu_{n-1} = 1)}{\partial \mu_1} = \frac{\partial \mathbf{S}(\mu_{m-1} = 1, \nu_{n-1} = 1)}{\partial \mu_{m-1}} \quad (109)$$

$$\begin{aligned} -3\mathbf{Q}_{1,n} + 3\mathbf{Q}_{3,n} - 12\mathbf{Q}_{1,n+1} + 12\mathbf{Q}_{3,n+1} - 3\mathbf{Q}_{1,n+2} + 12\mathbf{Q}_{3,n+2} + 3\mathbf{Q}_{m,n} - 3\mathbf{Q}_{m+2,n} \\ + 12\mathbf{Q}_{m,n+1} - 12\mathbf{Q}_{m+2,n+1} + 3\mathbf{Q}_{m,n+2} - 3\mathbf{Q}_{m+2,n+2} = 0 \end{aligned} \quad (110)$$

- continuity of the second derivative with respect to μ_i at fit knots $\mathbf{P}_{1,j}$ and $\mathbf{P}_{m,j}$ for $j \in [1, n-1]$:

$$\frac{\partial^2 \mathbf{S}(\mu_1 = 0, \nu_j = 0)}{\partial \mu_1^2} = \frac{\partial^2 \mathbf{S}(\mu_{m-1} = 1, \nu_j = 0)}{\partial \mu_{m-1}^2} \quad (111)$$

$$\begin{aligned}
 &6\mathbf{Q}_{1,j} - 12\mathbf{Q}_{2,j} + 6\mathbf{Q}_{3,j} + 24\mathbf{Q}_{1,j+1} - 48\mathbf{Q}_{2,j+1} + 24\mathbf{Q}_{3,j+1} + 6\mathbf{Q}_{1,j+2} - 12\mathbf{Q}_{2,j+2} + 6\mathbf{Q}_{3,j+2} \\
 &- 6\mathbf{Q}_{m,j} + 12\mathbf{Q}_{m+1,j} - 6\mathbf{Q}_{m+2,j} - 24\mathbf{Q}_{m,j+1} + 48\mathbf{Q}_{m+1,j+1} - 24\mathbf{Q}_{m+2,j+1} - 6\mathbf{Q}_{m,j+2} \\
 &+ 12\mathbf{Q}_{m+1,j+2} - 6\mathbf{Q}_{m+2,j+2} = 0
 \end{aligned} \tag{112}$$

- continuity of the second derivative with respect to μ_i at the last fit knots $\mathbf{P}_{1,n}$ and $\mathbf{P}_{m,n}$

$$\frac{\partial^2 \mathbf{S}(\mu_1 = 0, \nu_{n-1} = 1)}{\partial \mu_1^2} = \frac{\partial^2 \mathbf{S}(\mu_{m-1} = 1, \nu_{n-1} = 1)}{\partial \mu_{m-1}^2} \tag{113}$$

$$\begin{aligned}
 &6\mathbf{Q}_{1,n} - 12\mathbf{Q}_{2,n} + 6\mathbf{Q}_{3,n} + 24\mathbf{Q}_{1,n+1} - 48\mathbf{Q}_{2,n+1} + 24\mathbf{Q}_{3,n+1} + 6\mathbf{Q}_{1,n+2} - 12\mathbf{Q}_{2,n+2} + 6\mathbf{Q}_{3,n+2} \\
 &- 6\mathbf{Q}_{m,n} + 12\mathbf{Q}_{m+1,n} - 6\mathbf{Q}_{m+2,n} - 24\mathbf{Q}_{m,n+1} + 48\mathbf{Q}_{m+1,n+1} - 24\mathbf{Q}_{m+2,n+1} - 6\mathbf{Q}_{m,n+2} \\
 &+ 12\mathbf{Q}_{m+1,n+2} - 6\mathbf{Q}_{m+2,n+2} = 0
 \end{aligned} \tag{114}$$

It is therefore shown that equations (108) and (110), then (113) and (114), are respectively generalizable for any couple of fit knots $\mathbf{P}_{1,j}$ and $\mathbf{P}_{m,j}$ for $j \in [1, n]$. In summary, simple relations between control points emerge as end conditions for a cylindrical closed parametric surface, see equations (115) and (116).

first derivative continuity between edges u^- and u^+ :

$$\begin{aligned}
 &-\mathbf{Q}_{1,j} + \mathbf{Q}_{3,j} - 4\mathbf{Q}_{1,j+1} + 4\mathbf{Q}_{3,j+1} - \mathbf{Q}_{1,j+2} + \mathbf{Q}_{3,j+2} + \mathbf{Q}_{m,j} - \mathbf{Q}_{m+2,j} + 4\mathbf{Q}_{m,j+1} \\
 &- 4\mathbf{Q}_{m+2,j+1} + \mathbf{Q}_{m,j+2} - \mathbf{Q}_{m+2,j+2} = 0 \quad \text{for } j \in [1, n]
 \end{aligned} \tag{115}$$

second derivative continuity between edges u^- and u^+ :

$$\begin{aligned}
 &\mathbf{Q}_{1,j} - 2\mathbf{Q}_{2,j} + \mathbf{Q}_{3,j} + 4\mathbf{Q}_{1,j+1} - 8\mathbf{Q}_{2,j+1} + 4\mathbf{Q}_{3,j+1} + \mathbf{Q}_{1,j+2} - 2\mathbf{Q}_{2,j+2} + \mathbf{Q}_{3,j+2} - \mathbf{Q}_{m,j} + 2\mathbf{Q}_{m+1,j} \\
 &- \mathbf{Q}_{m+2,j} - 4\mathbf{Q}_{m,j+1} + 8\mathbf{Q}_{m+1,j+1} - 4\mathbf{Q}_{m+2,j+1} - \mathbf{Q}_{m,j+2} + 2\mathbf{Q}_{m+1,j+2} - \mathbf{Q}_{m+2,j+2} = 0 \quad \text{for } j \in [1, n]
 \end{aligned} \tag{116}$$

End condition equations for closing a cylindrical shaped parametric surface are schematized in figure 13. The bicubic surface in the parametric space is shown unrolled but it is important to remember that highlighted edges u^- and u^+ in (—) are perfectly confused. In the parametric space, one end condition equation is represented by a solid line and connects many control points from opposite edges. Only two equations are displayed in figures 13a and 13b for both derivatives. Shown patterns only correspond to cases $j = 1$ and $j = n$. The whole set of end condition equations would be obtained by translating the patterns along the axis \vec{v} .

As for the two remaining edges v^- and v^+ , free end conditions are still used. By definition of closed surface, the concept of corners no longer exists. Thereby, free end condition equations written in diagonal directions cannot be reused. Instead, the same end condition is used all over edges v^- and v^+ , even for control point of indexes $i = 1$ and $i = m + 2$:

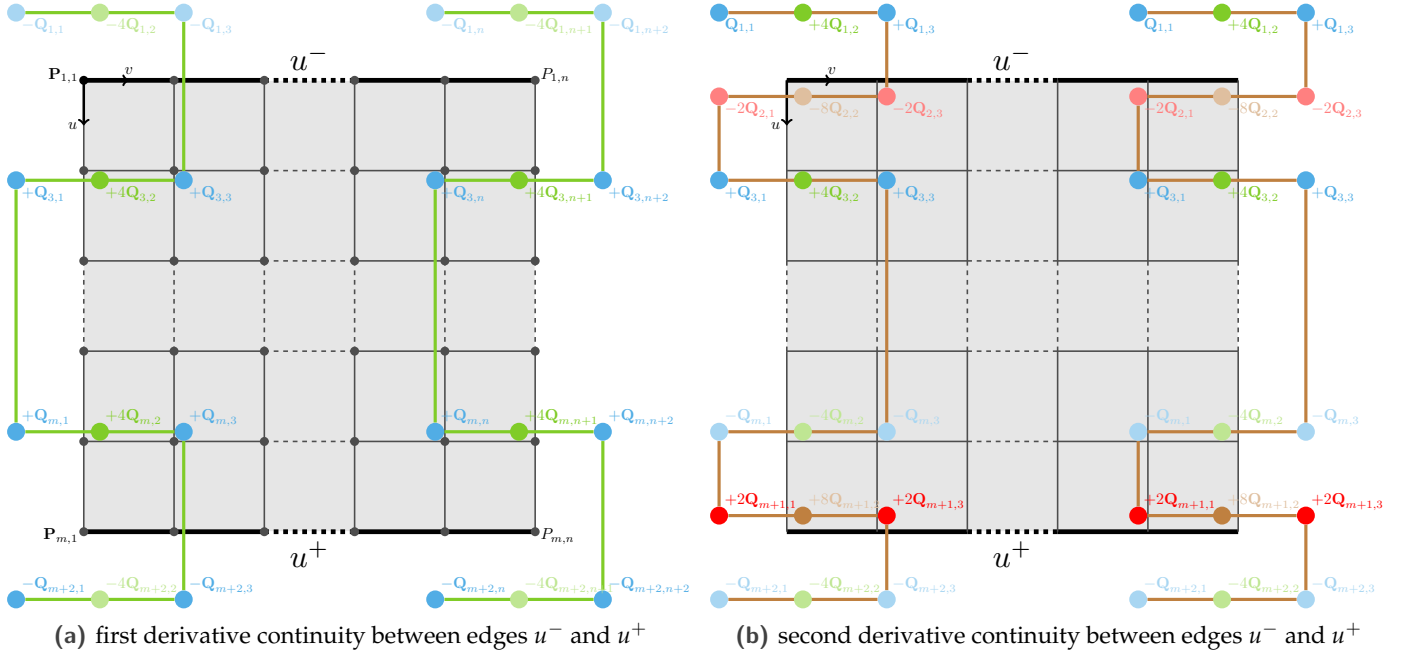


Figure 13. equations between control points for cylindrical closed surface

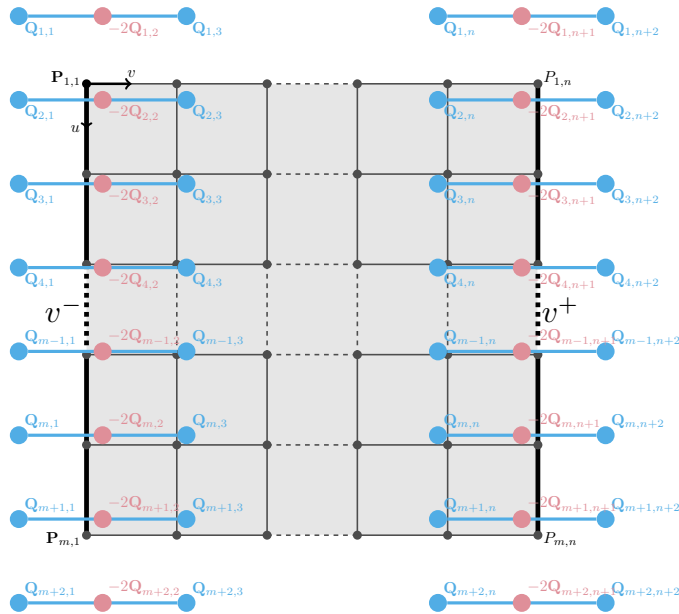


Figure 14. equations between control points for free end conditions at edges v^- and v^+

edge v^- :

$$\mathbf{Q}_{i,1} - 2\mathbf{Q}_{i,2} + \mathbf{Q}_{i,3} = 0 \quad \text{for } i \in [1, m + 2] \quad (117)$$

edge v^+ :

$$\mathbf{Q}_{i,n} - 2\mathbf{Q}_{i,n+1} + \mathbf{Q}_{i,n+2} = 0 \quad \text{for } i \in [1, m+2] \quad (118)$$

Free end conditions on edges v^- and v^+ are finally schematized in figure 14. Note that, end condition equations would be easily transposable to the other direction if the connection between edges v^- and v^+ is preferred for the closed parametric surface.

2.4.3 Spherical shaped parametric surface

Keeping the connection between edges u^- and u^+ detailed in section 2.4.2, a fully closed parametric surface can be obtained by merging edges v^- and v^+ into nodal problems. To this end, all the fit knots on edges v^- have been moved up to a single point with given coordinates, same on the edge v^+ . Considering a mesh grid of fit knots arranged in a spherical shape, the following condition has to be respected:

$$\mathbf{P}_{1,1} = \mathbf{P}_{i,1} \quad \forall i \in [1, m] \quad (119)$$

$$\mathbf{P}_{1,n} = \mathbf{P}_{i,n} \quad \forall i \in [1, m] \quad (120)$$

Regarding end conditions, the tangent at fitting points v^- and v^+ must be consistent for all of the juxtaposed patches. A null first derivative with respect to parameter v_1 , respectively v_{n-1} at the other side, satisfies this condition. The calculation of the first derivative with respect to v_j is detailed in equations (105).

$$\frac{\partial \mathbf{S}(\mu_i, v_j)}{\partial v_j} = \frac{1}{36} \begin{bmatrix} 3v_j^2 & 2v_j & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{i,j} & \mathbf{Q}_{i+1,j} & \mathbf{Q}_{i+2,j} & \mathbf{Q}_{i+3,j} \\ \mathbf{Q}_{i,j+1} & \mathbf{Q}_{i+1,j+1} & \mathbf{Q}_{i+2,j+1} & \mathbf{Q}_{i+3,j+1} \\ \mathbf{Q}_{i,j+2} & \mathbf{Q}_{i+1,j+2} & \mathbf{Q}_{i+2,j+2} & \mathbf{Q}_{i+3,j+2} \\ \mathbf{Q}_{i,j+3} & \mathbf{Q}_{i+1,j+3} & \mathbf{Q}_{i+2,j+3} & \mathbf{Q}_{i+3,j+3} \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 0 & 4 \\ -3 & 3 & 3 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mu_i^3 \\ \mu_i^2 \\ \mu_i \\ 1 \end{bmatrix} \quad (121)$$

While checking that the first derivatives are equal to zero, the relations between control points are established at the two merging points:

- first derivative with respect to v_1 at fit knots $\mathbf{P}_{i,1}$ for $i \in [1, m]$:

$$\frac{\partial \mathbf{S}(\mu_i = 0, v_1 = 0)}{\partial v_1} = 0 \quad (122)$$

$$-3\mathbf{Q}_{i,1} + 3\mathbf{Q}_{i,3} - 12\mathbf{Q}_{i+1,1} + 12\mathbf{Q}_{i+1,3} - 3\mathbf{Q}_{i+2,1} + 3\mathbf{Q}_{i+2,3} = 0 \quad (123)$$

- first derivative with respect to v_{n-1} at fit knots $\mathbf{P}_{i,n}$ for $i \in [1, m]$:

$$\frac{\partial \mathbf{S}(\mu_i = 1, v_{n-1} = 1)}{\partial v_{n-1}} = 0 \quad (124)$$

$$-3\mathbf{Q}_{i,m+2} + 3\mathbf{Q}_{i,m} - 12\mathbf{Q}_{i+1,m+2} + 12\mathbf{Q}_{i+1,m} - 3\mathbf{Q}_{i+2,m+2} + 3\mathbf{Q}_{i+2,m} = 0 \quad (125)$$

The relations between control points for the fully merged edges v^- and v^+ are schematized in figure 15. Only one equation is plotted at both edges for the sake of readability. An index shifting in the direction \vec{u} would allow to plot all the end conditions. We might think that 4 conditions are still missing, but these are brought by the continuity established between edges u^- and u^+ . The case treated on the edge v^+ is a clear illustration of this. End conditions at edges u^- and u^+ is exactly the same as in figure 13, so it is not redetailed.

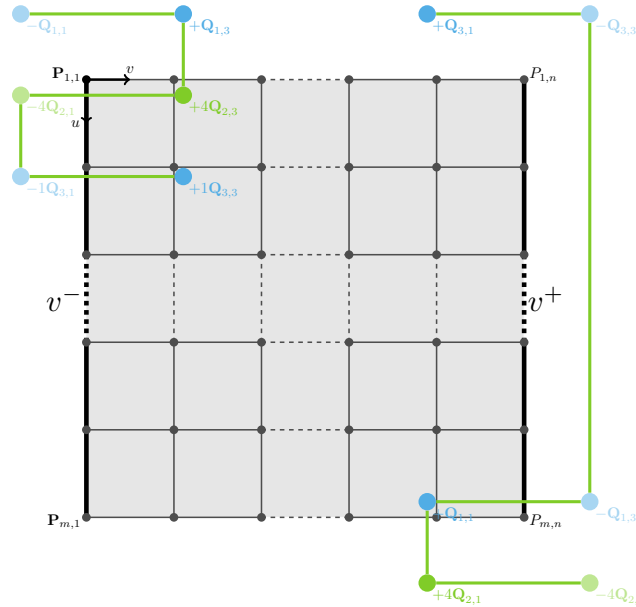


Figure 15. equations between control points for merged edges v^- and v^+

2.4.4 not-a-knot end conditions

not-a-knot end conditions ensure the continuity of the third derivative of the bicubic parametric surface on the second row of fit knots on all four edges. On these fit knots, the parametric surface has to be C^3 . This means that juxtaposed patches are defined by the same parametric functions. In the parametric space $(\vec{\mu}_i; \vec{\nu}_j)$, the third derivatives of $\mathbf{S}(\mu_i, \nu_j)$ with respect to both directions are calculated, see equation (126) and (142). The mixed derivatives equality does not need to be studied because always true for bicubic parametric surface.

$$\frac{\partial^3 \mathbf{S}(\mu_i, \nu_j)}{\partial \mu_i^3} = \frac{1}{6} \begin{bmatrix} \nu_j^3 & \nu_j^2 & \nu_j & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{ij} & \mathbf{Q}_{i+1,j} & \mathbf{Q}_{i+2,j} & \mathbf{Q}_{i+3,j} \\ \mathbf{Q}_{ij+1} & \mathbf{Q}_{i+1,j+1} & \mathbf{Q}_{i+2,j+1} & \mathbf{Q}_{i+3,j+1} \\ \mathbf{Q}_{ij+2} & \mathbf{Q}_{i+1,j+2} & \mathbf{Q}_{i+2,j+2} & \mathbf{Q}_{i+3,j+2} \\ \mathbf{Q}_{ij+3} & \mathbf{Q}_{i+1,j+3} & \mathbf{Q}_{i+2,j+3} & \mathbf{Q}_{i+3,j+3} \end{bmatrix} \begin{bmatrix} -1 \\ 3 \\ -3 \\ 1 \end{bmatrix} \quad (126)$$

$$\frac{\partial^3 \mathbf{S}(\mu_i, \nu_j)}{\partial \nu_j^3} = \frac{1}{6} \begin{bmatrix} -1 & 3 & -3 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{i,j} & \mathbf{Q}_{i+1,j} & \mathbf{Q}_{i+2,j} & \mathbf{Q}_{i+3,j} \\ \mathbf{Q}_{i,j+1} & \mathbf{Q}_{i+1,j+1} & \mathbf{Q}_{i+2,j+1} & \mathbf{Q}_{i+3,j+1} \\ \mathbf{Q}_{i,j+2} & \mathbf{Q}_{i+1,j+2} & \mathbf{Q}_{i+2,j+2} & \mathbf{Q}_{i+3,j+2} \\ \mathbf{Q}_{i,j+3} & \mathbf{Q}_{i+1,j+3} & \mathbf{Q}_{i+2,j+3} & \mathbf{Q}_{i+3,j+3} \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 0 & 4 \\ -3 & 3 & 3 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mu_i^3 \\ \mu_i^2 \\ \mu_i \\ 1 \end{bmatrix} \quad (127)$$

$2m + 2n$ equations are so defined from the *not-a-knot* end conditions when written on the proper fit knots:

edge u^- at fit knots $P_{2,j}$:

$$\frac{\partial^3 \mathbf{S}(\mu_1 = 1, \nu_j = 0)}{\partial \mu_1^3} = \frac{\partial^3 \mathbf{S}(\mu_2 = 0, \nu_j = 0)}{\partial \mu_2^3} \quad \text{for } j \in [1, n-1] \quad (128)$$

$$\frac{\partial^3 \mathbf{S}(\mu_1 = 1, \nu_{n-1} = 1)}{\partial \mu_1^3} = \frac{\partial^3 \mathbf{S}(\mu_2 = 0, \nu_{n-1} = 1)}{\partial \mu_2^3} \quad (129)$$

Equations are detailed for cases $j \in [1, n-1]$ but generalizable at all fit knots $P_{2,j}$ $j \in [1, n]$ because equation (129) gives the same result.

$$\begin{bmatrix} 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{1,j} & \mathbf{Q}_{2,j} & \mathbf{Q}_{3,j} & \mathbf{Q}_{4,j} \\ \mathbf{Q}_{1,j+1} & \mathbf{Q}_{2,j+1} & \mathbf{Q}_{3,j+1} & \mathbf{Q}_{4,j+1} \\ \mathbf{Q}_{1,j+2} & \mathbf{Q}_{2,j+2} & \mathbf{Q}_{3,j+2} & \mathbf{Q}_{4,j+2} \\ \mathbf{Q}_{1,j+3} & \mathbf{Q}_{2,j+3} & \mathbf{Q}_{3,j+3} & \mathbf{Q}_{4,j+3} \end{bmatrix} \begin{bmatrix} -1 \\ 3 \\ -3 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{2,j} & \mathbf{Q}_{3,j} & \mathbf{Q}_{4,j} & \mathbf{Q}_{5,j} \\ \mathbf{Q}_{2,j+1} & \mathbf{Q}_{3,j+1} & \mathbf{Q}_{4,j+1} & \mathbf{Q}_{5,j+1} \\ \mathbf{Q}_{2,j+2} & \mathbf{Q}_{3,j+2} & \mathbf{Q}_{4,j+2} & \mathbf{Q}_{5,j+2} \\ \mathbf{Q}_{2,j+3} & \mathbf{Q}_{3,j+3} & \mathbf{Q}_{4,j+3} & \mathbf{Q}_{5,j+3} \end{bmatrix} \begin{bmatrix} -1 \\ 3 \\ -3 \\ 1 \end{bmatrix} \quad (130)$$

$$0 = -\mathbf{Q}_{1,j} - 4\mathbf{Q}_{1,j+1} - \mathbf{Q}_{1,j+2} + 4\mathbf{Q}_{2,j} + 16\mathbf{Q}_{2,j+1} + 4\mathbf{Q}_{2,j+2} - 6\mathbf{Q}_{3,j} - 24\mathbf{Q}_{3,j+1} - 6\mathbf{Q}_{3,j+2} + \dots \\ \dots + 4\mathbf{Q}_{4,j} + 16\mathbf{Q}_{4,j+1} + 4\mathbf{Q}_{4,j+2} - \mathbf{Q}_{5,j} - 4\mathbf{Q}_{5,j+1} - \mathbf{Q}_{5,j+2} \quad (131)$$

edge u^+ at fit knots $P_{m-1,j}$:

$$\frac{\partial^3 \mathbf{S}(\mu_{m-2} = 1, \nu_j = 0)}{\partial \mu_{m-2}^3} = \frac{\partial^3 \mathbf{S}(\mu_{m-1} = 0, \nu_j = 0)}{\partial \mu_{m-1}^3} \quad \text{for } j \in [1, n-1] \quad (132)$$

$$\frac{\partial^3 \mathbf{S}(\mu_{m-2} = 1, \nu_{n-1} = 1)}{\partial \mu_{m-2}^3} = \frac{\partial^3 \mathbf{S}(\mu_{m-1} = 0, \nu_{n-1} = 1)}{\partial \mu_{m-1}^3} \quad (133)$$

Equations are detailed for cases $j \in [1, n - 1]$ but generalizable at all fit knots $P_{m-1,j}$ $j \in [1, n]$ because equation (133) gives the same result.

$$\begin{bmatrix} 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{m-2,j} & \mathbf{Q}_{m-1,j} & \mathbf{Q}_{m,j} & \mathbf{Q}_{m+1,j} \\ \mathbf{Q}_{m-2,j+1} & \mathbf{Q}_{m-1,j+1} & \mathbf{Q}_{m,j+1} & \mathbf{Q}_{m+1,j+1} \\ \mathbf{Q}_{m-2,j+2} & \mathbf{Q}_{m-1,j+2} & \mathbf{Q}_{m,j+2} & \mathbf{Q}_{m+1,j+2} \\ \mathbf{Q}_{m-2,j+3} & \mathbf{Q}_{m-1,j+3} & \mathbf{Q}_{m,j+3} & \mathbf{Q}_{m+1,j+3} \end{bmatrix} \begin{bmatrix} -1 \\ 3 \\ -3 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{m-1,j} & \mathbf{Q}_{m,j} & \mathbf{Q}_{m+1,j} & \mathbf{Q}_{m+2,j} \\ \mathbf{Q}_{m-1,j+1} & \mathbf{Q}_{m,j+1} & \mathbf{Q}_{m+1,j+1} & \mathbf{Q}_{m+2,j+1} \\ \mathbf{Q}_{m-1,j+2} & \mathbf{Q}_{m,j+2} & \mathbf{Q}_{m+1,j+2} & \mathbf{Q}_{m+2,j+2} \\ \mathbf{Q}_{m-1,j+3} & \mathbf{Q}_{m,j+3} & \mathbf{Q}_{m+1,j+3} & \mathbf{Q}_{m+2,j+3} \end{bmatrix} \begin{bmatrix} -1 \\ 3 \\ -3 \\ 1 \end{bmatrix} \quad (134)$$

$$\begin{aligned} 0 = & -\mathbf{Q}_{m-2,j} - 4\mathbf{Q}_{m-2,j+1} - \mathbf{Q}_{m-2,j+2} + 4\mathbf{Q}_{m-1,j} + 16\mathbf{Q}_{m-1,j+1} + 4\mathbf{Q}_{m-1,j+2} - 6\mathbf{Q}_{m,j} - 24\mathbf{Q}_{m,j+1} + \dots \\ & \dots - 6\mathbf{Q}_{m,j+2} + 4\mathbf{Q}_{m+1,j} + 16\mathbf{Q}_{m+1,j+1} + 4\mathbf{Q}_{m+1,j+2} - \mathbf{Q}_{m+2,j} - 4\mathbf{Q}_{m+2,j+1} - \mathbf{Q}_{m+2,j+2} \end{aligned} \quad (135)$$

edge v^- at fit knots $P_{i,2}$:

$$\frac{\partial^3 \mathbf{S}(\mu_i = 0, \nu_1 = 1)}{\partial \nu_1^3} = \frac{\partial^3 \mathbf{S}(\mu_i = 0, \nu_2 = 0)}{\partial \nu_2^3} \quad \text{for } i \in [1, m - 1] \quad (136)$$

$$\frac{\partial^3 \mathbf{S}(\mu_{m-1} = 1, \nu_1 = 1)}{\partial \nu_1^3} = \frac{\partial^3 \mathbf{S}(\mu_{m-1} = 1, \nu_2 = 0)}{\partial \nu_2^3} \quad (137)$$

Equations are detailed for cases $i \in [1, m - 1]$ but generalizable at all fit knots $P_{i,2}$ $i \in [1, m]$ because equation (137) gives the same result.

$$\begin{bmatrix} -1 & 3 & -3 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{i,1} & \mathbf{Q}_{i+1,1} & \mathbf{Q}_{i+2,1} & \mathbf{Q}_{i+3,1} \\ \mathbf{Q}_{i,2} & \mathbf{Q}_{i+1,2} & \mathbf{Q}_{i+2,2} & \mathbf{Q}_{i+3,2} \\ \mathbf{Q}_{i,3} & \mathbf{Q}_{i+1,3} & \mathbf{Q}_{i+2,3} & \mathbf{Q}_{i+3,3} \\ \mathbf{Q}_{i,4} & \mathbf{Q}_{i+1,4} & \mathbf{Q}_{i+2,4} & \mathbf{Q}_{i+3,4} \end{bmatrix} \begin{bmatrix} 1 \\ 4 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} -1 & 3 & -3 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{i,2} & \mathbf{Q}_{i+1,2} & \mathbf{Q}_{i+2,2} & \mathbf{Q}_{i+3,2} \\ \mathbf{Q}_{i,3} & \mathbf{Q}_{i+1,3} & \mathbf{Q}_{i+2,3} & \mathbf{Q}_{i+3,3} \\ \mathbf{Q}_{i,4} & \mathbf{Q}_{i+1,4} & \mathbf{Q}_{i+2,4} & \mathbf{Q}_{i+3,4} \\ \mathbf{Q}_{i,5} & \mathbf{Q}_{i+1,5} & \mathbf{Q}_{i+2,5} & \mathbf{Q}_{i+3,5} \end{bmatrix} \begin{bmatrix} 1 \\ 4 \\ 1 \\ 0 \end{bmatrix} \quad (138)$$

$$\begin{aligned} 0 = & -\mathbf{Q}_{i,1} + 4\mathbf{Q}_{i,2} - 6\mathbf{Q}_{i,3} + 4\mathbf{Q}_{i,4} - 1\mathbf{Q}_{i,5} - 4\mathbf{Q}_{i+1,1} + 16\mathbf{Q}_{i+1,2} - 24\mathbf{Q}_{i+1,3} + 16\mathbf{Q}_{i+1,4} - 4\mathbf{Q}_{i+1,5} + \dots \\ & \dots - \mathbf{Q}_{i+2,1} + 4\mathbf{Q}_{i+2,2} - 6\mathbf{Q}_{i+2,3} + 4\mathbf{Q}_{i+2,4} - 1\mathbf{Q}_{i+2,5} \end{aligned} \quad (139)$$

edge v^+ at fit knots $P_{i,n-1}$:

$$\frac{\partial^3 \mathbf{S}(\mu_i = 0, \nu_{n-2} = 1)}{\partial \nu_{n-2}^3} = \frac{\partial^3 \mathbf{S}(\mu_i = 0, \nu_{n-1} = 0)}{\partial \nu_{n-1}^3} \quad \text{for } i \in [1, m-1] \quad (140)$$

$$\frac{\partial^3 \mathbf{S}(\mu_{m-1} = 1, \nu_{n-2} = 1)}{\partial \nu_{n-2}^3} = \frac{\partial^3 \mathbf{S}(\mu_{m-1} = 1, \nu_{n-1} = 0)}{\partial \nu_{n-1}^3} \quad (141)$$

Equations are detailed for cases $i \in [1, m-1]$ but generalizable at all fit knots $P_{i,n-1}$ $i \in [1, m]$ because equation (141) gives the same result.

$$\begin{bmatrix} -1 & 3 & -3 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{i,n-2} & \mathbf{Q}_{i+1,n-2} & \mathbf{Q}_{i+2,n-2} & \mathbf{Q}_{i+3,n-2} \\ \mathbf{Q}_{i,n-1} & \mathbf{Q}_{i+1,n-1} & \mathbf{Q}_{i+2,n-1} & \mathbf{Q}_{i+3,n-1} \\ \mathbf{Q}_{i,n} & \mathbf{Q}_{i+1,n} & \mathbf{Q}_{i+2,n} & \mathbf{Q}_{i+3,n} \\ \mathbf{Q}_{i,n+1} & \mathbf{Q}_{i+1,n+1} & \mathbf{Q}_{i+2,n+1} & \mathbf{Q}_{i+3,n+1} \end{bmatrix} \begin{bmatrix} 1 \\ 4 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_{i,n-1} & \mathbf{Q}_{i+1,n-1} & \mathbf{Q}_{i+2,n+1} & \mathbf{Q}_{i+3,n-1} \\ \mathbf{Q}_{i,n} & \mathbf{Q}_{i+1,n} & \mathbf{Q}_{i+2,n} & \mathbf{Q}_{i+3,n} \\ \mathbf{Q}_{i,n+1} & \mathbf{Q}_{i+1,n+1} & \mathbf{Q}_{i+2,n+1} & \mathbf{Q}_{i+3,n+1} \\ \mathbf{Q}_{i,n+2} & \mathbf{Q}_{i+1,n+2} & \mathbf{Q}_{i+2,n+2} & \mathbf{Q}_{i+3,n+2} \end{bmatrix} \begin{bmatrix} 1 \\ 4 \\ 1 \\ 0 \end{bmatrix} \quad (142)$$

$$\begin{aligned} 0 = & -\mathbf{Q}_{i,n-2} + 4\mathbf{Q}_{i,n-1} - 6\mathbf{Q}_{i,n} + 4\mathbf{Q}_{i,n+1} - 1\mathbf{Q}_{i,n+2} - 4\mathbf{Q}_{i+1,n-2} + 16\mathbf{Q}_{i+1,n-1} - 24\mathbf{Q}_{i+1,n} + 16\mathbf{Q}_{i+1,n+1} \dots \\ & \dots - 4\mathbf{Q}_{i+1,n+2} - \mathbf{Q}_{i+2,n-2} + 4\mathbf{Q}_{i+2,n-1} - 6\mathbf{Q}_{i+2,n} + 4\mathbf{Q}_{i+2,n+1} - 1\mathbf{Q}_{i+2,n+2} \end{aligned} \quad (143)$$

These equations therefore establish relations between 15 control points near each fit knots concerned by the *not-a-knot* end conditions. Figures 16 provide a schematization of the *not-a-knot* end conditions. The set of control points that contribute to the *not-a-knot* end condition are connected by a solid line. For the sake of readability, only two equations from opposite edges are detailed in each figure. Figure 16a deals with fit knots $P_{1,2}$ and $P_{m,n-1}$ and figure 16b with $P_{2,1}$ and $P_{m-1,n}$. The coefficients relative to each $\mathbf{Q}_{i,j}$ are also reported with a specific color code. In order to obtain all of the other end condition equations, patterns have to be translated along the concerned edges, considering an index shifting.

Four equations are still missing to find a unique solution to the vector of control points \mathbf{Q} . Note that the control points ($\mathbf{Q}_{1,1}$, $\mathbf{Q}_{1,n+2}$, $\mathbf{Q}_{m+2,1}$ and $\mathbf{Q}_{m+2,n+2}$) —at the four corners of the parametric surface— are the least redundant unknowns in the system of equations. It is then proposed to add symmetry conditions to fix the four corners. Such end conditions, detailed in equations (144) to (147), have very slight influence on the final shape of the parametric surface.

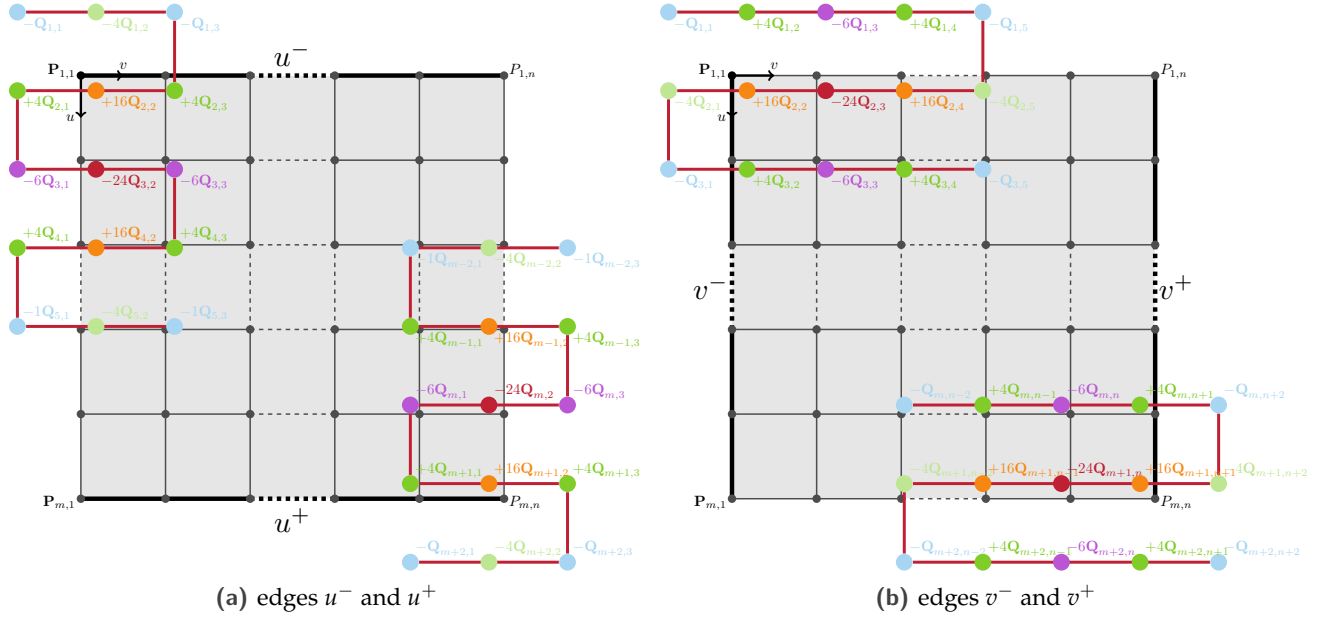


Figure 16. equations between control points for *not-a-knot* end condition on the 4 edges

corner u^-v^- :

$$Q_{1,1} - Q_{1,2} - Q_{2,1} + Q_{2,2} = 0 \quad (144)$$

corner u^-v^+ :

$$-Q_{1,n-1} + Q_{1,n} + Q_{2,n-1} - Q_{2,n} = 0 \quad (145)$$

corner u^+v^- :

$$-Q_{m-1,1} + Q_{m-1,2} + Q_{m,1} - Q_{m,2} = 0 \quad (146)$$

corner u^+v^+ :

$$Q_{m-1,n-1} - Q_{m-1,n} - Q_{m,n-1} + Q_{m,n} = 0 \quad (147)$$

Equations at the corners are schematized in figure 17.

2.5 Position of control points

The calculation of the control point coordinates follows the same approach as in the two-dimensional case. End condition equations are concatenated by adding rows to the passing matrix Φ while completing the fit knots vector \mathbf{P} with enough zeros. The fit knots vector \mathbf{P}^* and passing matrix Φ^* respectively denote the completed formulation of \mathbf{P} and Φ including end conditions. Once we have chosen one configuration of end conditions, a unique vector \mathbf{Q} is obtained while using the inverse calculation defined in equation (148).

$$\mathbf{Q} = 6\Phi^{*-1}\mathbf{P}^* \quad (148)$$

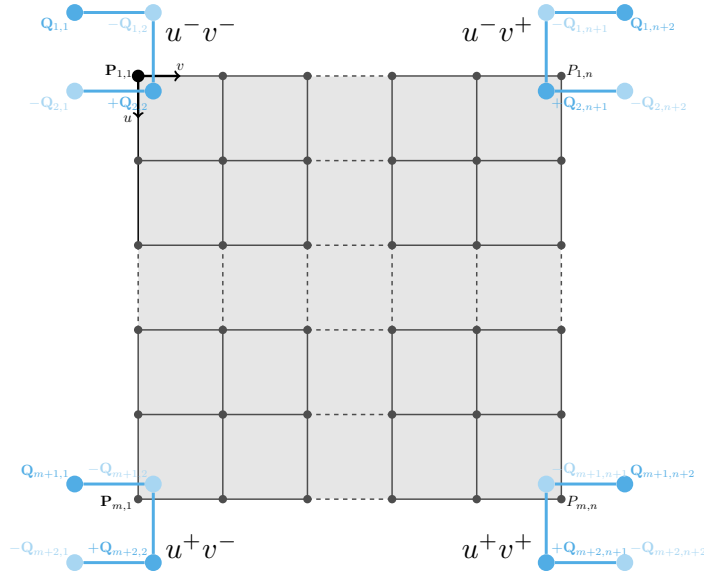


Figure 17. equations between control points at corners for *not-a-knot* end condition

Some examples of extended passing matrices Φ^* are given in appendices A to D for the case $m = 4$ and $n = 4$. This configuration is the smallest mesh grid of fit knots that can be computed with *not-a-knot* end conditions. Also, the dimensions are still realistic to write the entire matrices by hand. The content of matrices is highlighted by a specific color code ($\square = 0$, $\blacksquare = 1$, $\square = -1$, $\blacksquare = 2$, $\square = -2$, $\blacksquare = 4$, $\square = -4$, $\blacksquare = 8$, $\square = -8$, $\blacksquare = 16$, $\blacksquare = -24$) which has already been used to display the weights of control points in the schematization of end condition equations. Comments are added to identify which row is associated with which end conditions. These are a good support to visualize how to set up interpolation problem with free end conditions, cylindrical closed surface, spherical closed surface and knot-a-knot end conditions. Furthermore, these matrix forms correspond to the ones implemented in the Python routines.

2.6 Plotting bicubic parametric surfaces

Once the positions of the control points are known, the bicubic parametric surface can be plotted by evaluating equation 87, patch by patch, in their respective parametric space. As seen before, the different patches will be drawn by changing indexes i and j to navigate through rows and columns. Different configurations are considered in the following paragraph relating to the investigated end conditions. Fit points used for the interpolation are detailed on a case-by-case basis. With the perspective of an open source access, Python 3.8 routines are provided to reprogram by hand the interpolation of bicubic parametric surfaces from different sets of fit knots with different types of end conditions.

2.6.1 Free end condition

A parametric surface is obtained in figure 18 by interpolating an arbitrary set of fit knots, for case $m = 9$ and $n = 10$, with free end conditions. The particular shape obtained at the ends is well representative of the fact that the first three control points on all edges and corners are aligned and evenly spaced. The

Python 3.8 routine for plotting a similar parametric surface with free end conditions is given just below. In the provided example, fit knots positions are randomized around a regular mesh grid of dimensions $m \times n$.

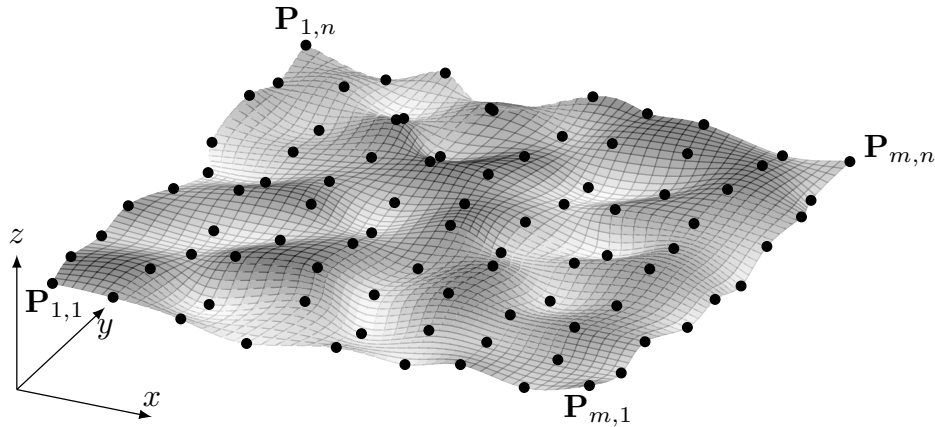


Figure 18. bicubic parametric surface interpolation for case $m = 9$ and $n = 10$ with free end condition



The Python source code for generating the bicubic parametric surface interpolation of a given set of fit knots with **free end conditions** is available here: https://lava-wiki.meca.polymtl.ca/_media/public/notespline/bicubic_parametric_surface_free_end.py

2.6.2 Cylindrical closed parametric surface

A closed parametric surface is displayed in figures 19 interpolated on a set of fit knots arranged on a randomized cylindrical geometry. The chosen mesh grid discretization is $m = 7$ and $n = 8$. The connexion between edges u^- and u^+ is highlighted in (—). Along this line, fit knots of opposite edges are superimposed and end conditions require that the first and second derivative are continuous. Both other edges are defined with free end conditions. The Hyperlink to a Python 3.8 routine to plot a similar cylindrical parametric surface is given below the figure.

2.6.3 Spherical shaped parametric surface

A fully closed parametric surface is plotted in figures 20. This shape is obtained while interpolating a mesh grid of fit knots ($m = 7$ and $n = 8$) arranged on a randomized spherical distribution. The set of fit knots is built as edges u^- and u^+ are still connected. Also, all the fit knots of edges v^- and v^+ are merged, meeting at both ends of the line highlighted in (—).

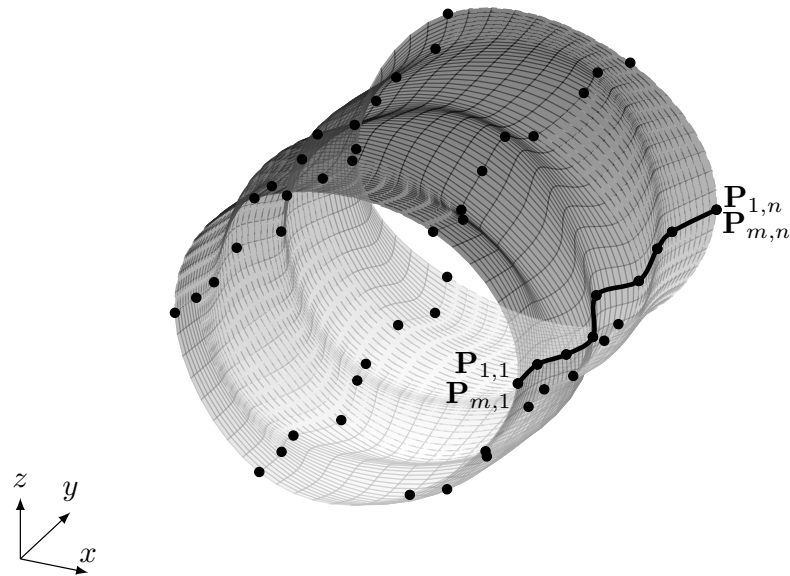


Figure 19. closed bicubic parametric surface interpolation on a randomized cylinder with $m = 7$ and $n = 8$



The Python source code for generating the closed bicubic parametric surface interpolation on a **randomized cylinder** is available here: https://lava-wiki.meca.polymtl.ca/_media/public/notespline/bicubic_parametric_cylinder.py

2.6.4 not-a-knot end conditions

Figure 21 represents a bicubic parametric surface interpolated on an arbitrary set of fit knots with *not-a-knot* end conditions. This configuration involves a $m = 9$ by $n = 10$ mesh grid of fit knots. This configuration can be reproduced for any parametric surface discretization with the provided routine. As for two-dimensional cubic splines, *not-a-knot* end conditions offer a smooth representation of the parametric surface extremities. To consider if specific conditions are not requested by the user.

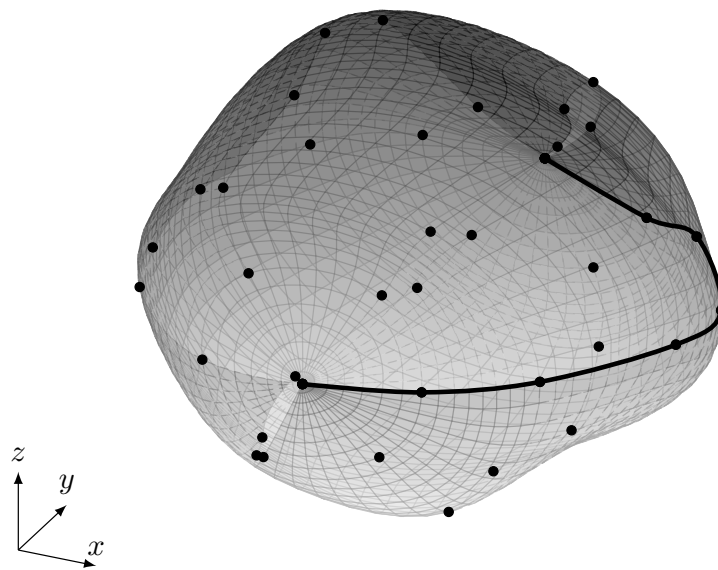


Figure 20. closed bicubic parametric surface interpolation on a randomized sphere with $m = 7$ and $n = 8$



The Python source code for generating the closed bicubic parametric surface interpolation on a **randomized sphere** is available here: https://lava-wiki.meca.polymtl.ca/_media/public/notespline/bicubic.parametric_sphere.py

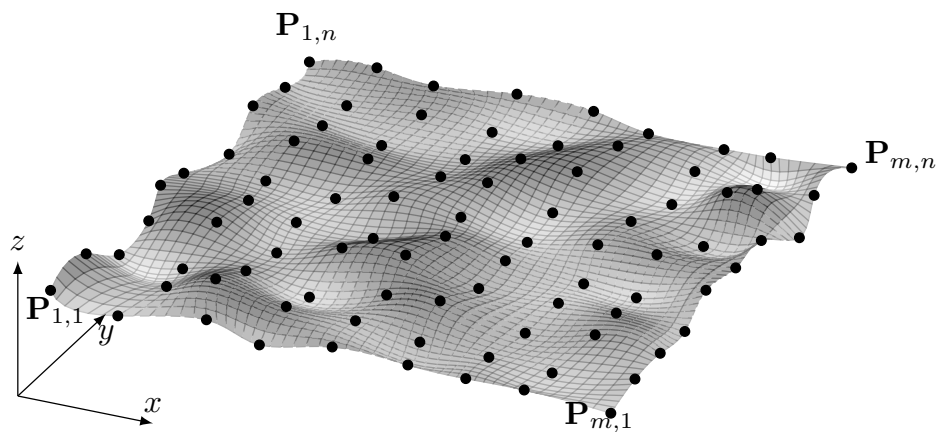


Figure 21. bicubic parametric surface interpolation for case $m = 9$ and $n = 10$ with *not-a-knot* end conditions

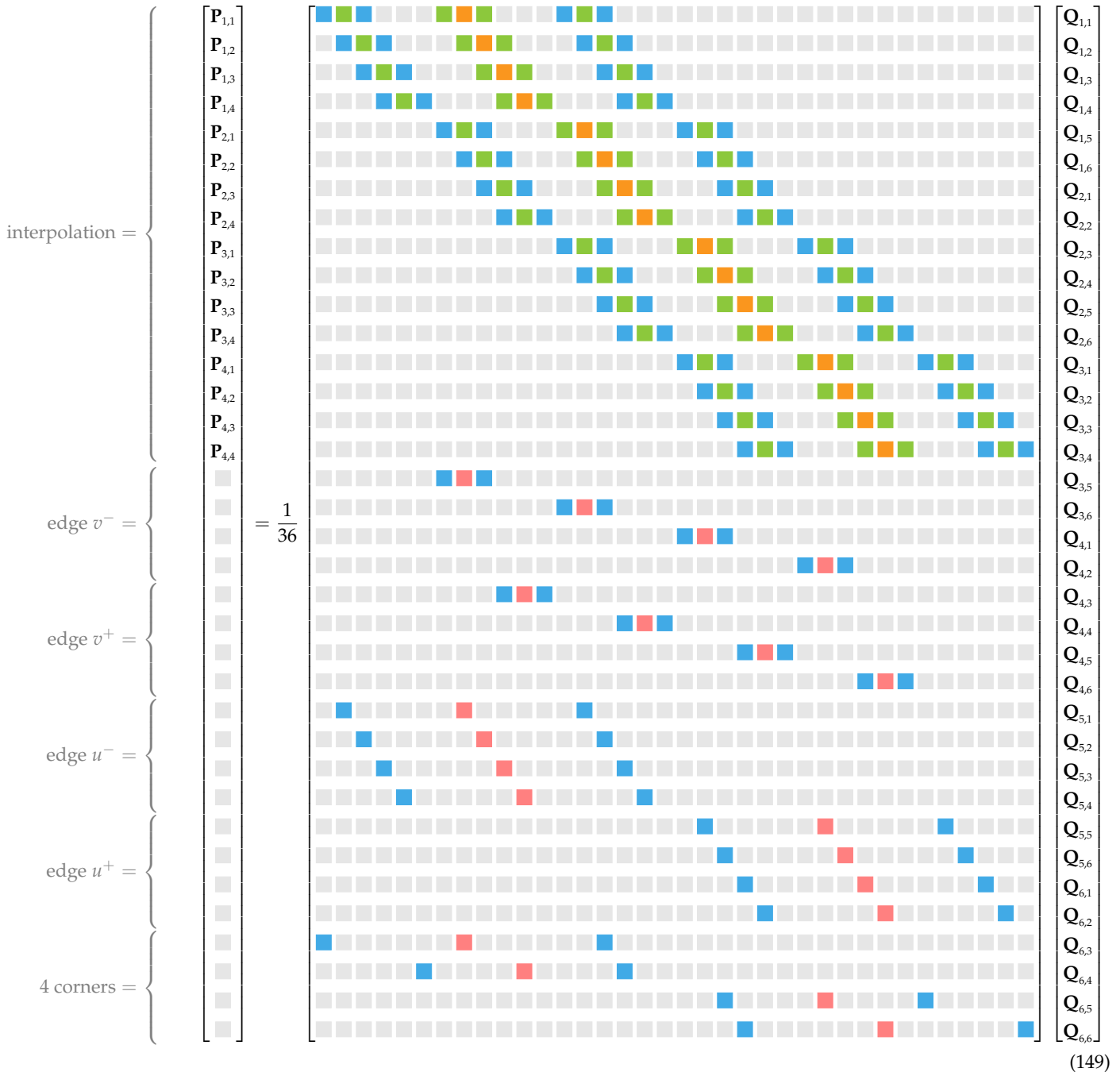


The Python source code for generating the bicubic parametric surface interpolation of a given set of fit knots with **not-a-knot end conditions** is available here: https://lava-wiki.meca.polymtl.ca/_media/public/notespline/bicubic_parametric_surface_not-a-knot.py

Appendices

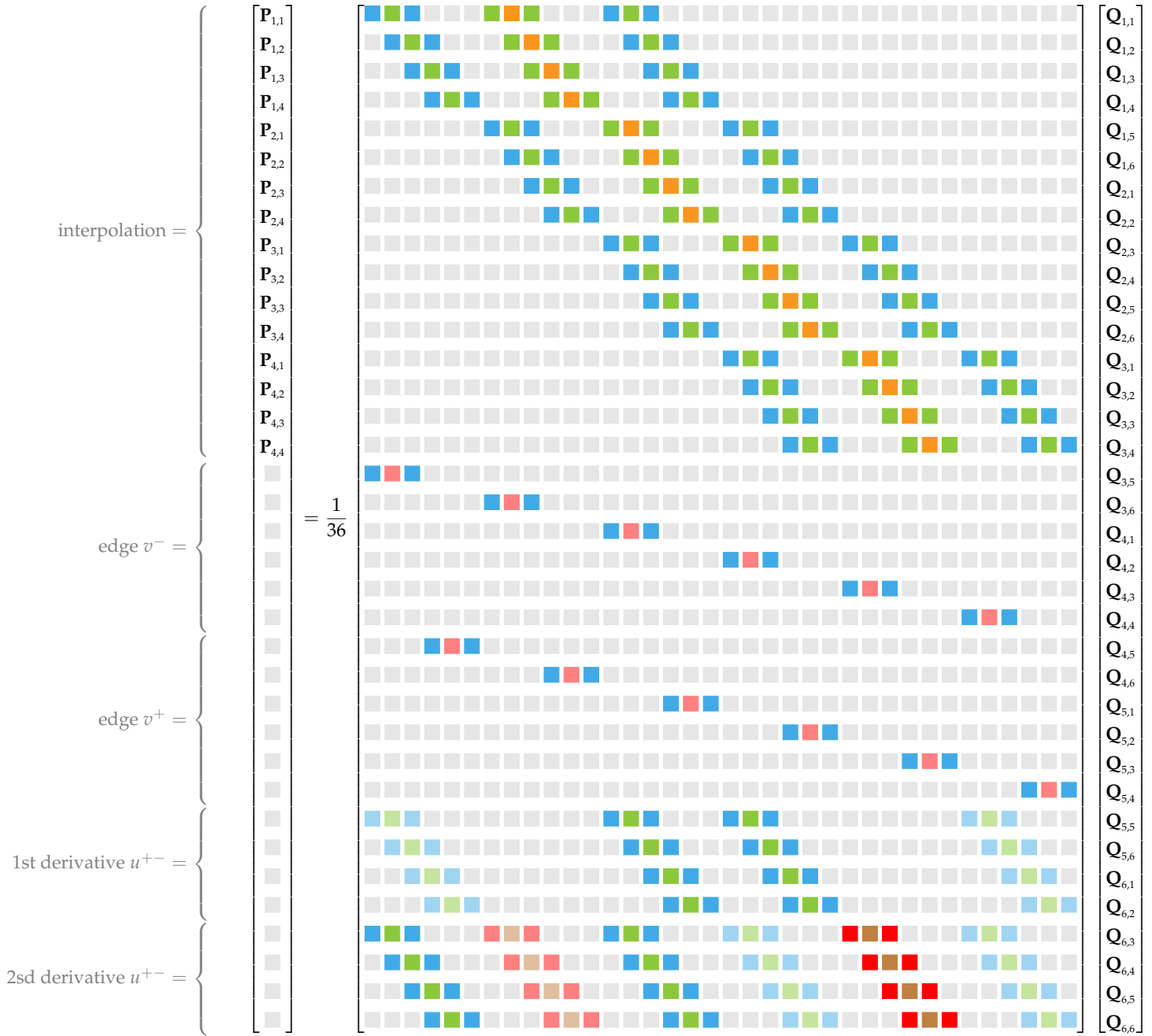
A Free end conditions: passing matrix from control points to fit knots

color code : $\square = 0$, $\blacksquare = 1$, $\blacksquare = -2$, $\blacksquare = 4$, $\blacksquare = 16$



B Cylindrical closed surface: passing matrix from control points to fit knots

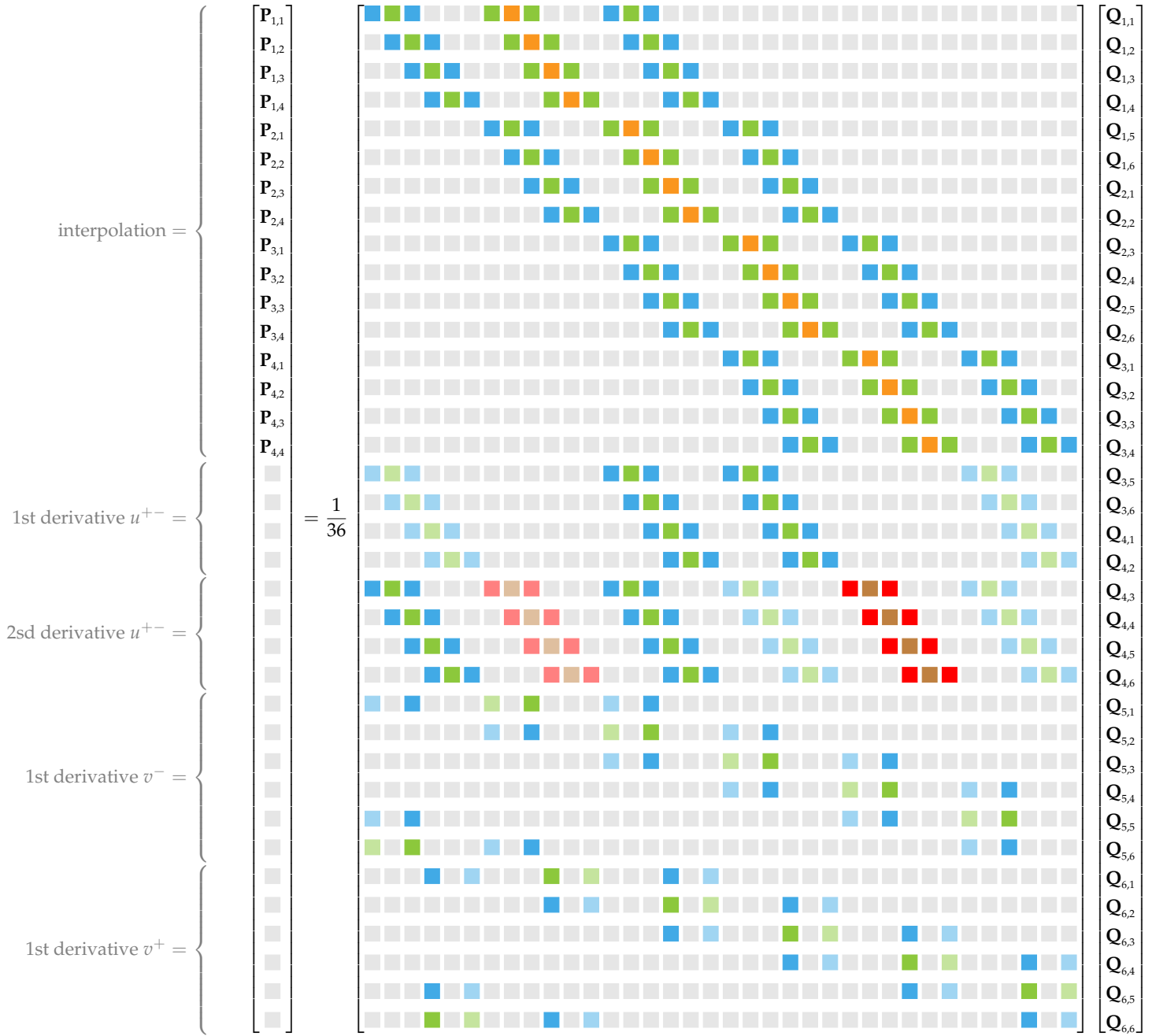
color code: $\square = 0$, $\square = 1$, $\square = -1$, $\square = 2$, $\square = -2$, $\square = 4$, $\square = -4$, $\square = 8$, $\square = -8$, $\square = 16$



(150)

C Spherical closed surface: passing matrix from control points to fit knots

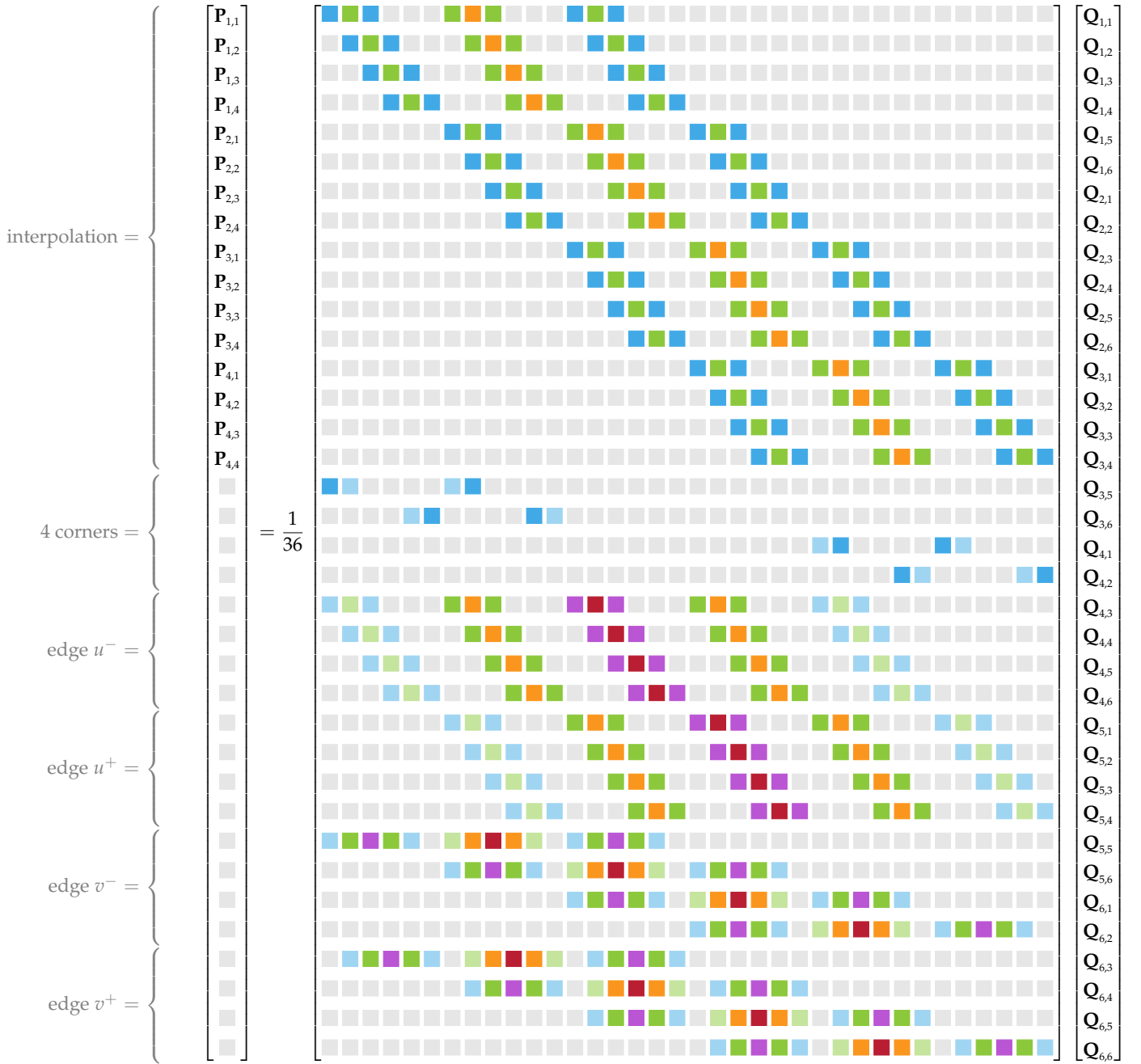
color code: $\square = 0$, $\square = 1$, $\square = -1$, $\square = 2$, $\square = -2$, $\square = 4$, $\square = -4$, $\square = 8$, $\square = -8$, $\square = 16$



(151)

D Not-a-knot end conditions: passing matrix from control points to fit knots

color code $\square = 0$, $\square = 1$, $\square = -1$, $\square = 4$, $\square = -4$, $\square = -6$, $\square = 16$, $\square = -24$



(152)

