mathematics

MDPI

Article

Automatic Handling of C^0-G^0 Continuous Rational Bézier Elements Produced from T-Splines Through Bézier Extraction

Christopher Provatidis [1,*] and Ioannis Dimitriou [2]

[1] School of Mechanical Engineering, National Technical University of Athens, 15780 Zografou, Greece
[2] Department of Mechanical Design and Control Systems, National Technical University of Athens, 15780 Zografou, Greece; idimitriou@mail.ntua.gr
* Correspondence: cprovat@mail.ntua.gr; Tel.: +30-210-7721520

Abstract: This paper shows that at a certain time-point in the analysis procedure, the accuracy of T-spline based isogeometric analysis (IGA) may be substantially improved by increasing the multiplicity of the inner knots up to the polynomial degree. This task can be performed by considering the Bézier extraction operator matrix elementwise, and thus an increased number of updated control points are easily received in the geometrical and computational models. Nevertheless, after the determination of the unique control points, the Bézier elements near the T-junctions may not be well shaped, and thus minor automatic interventions are required to ensure full (i.e., C^0 and G^0) compatibility. The improved IGA-based solution may be used as a reference to determine the a posteriori error estimations in the T-spline elements of the domain, and thus may be a useful tool for IGA adaptation. The methodology is shown in BVPs dominated by Laplace–Poisson equations in rectangular and curvilinear domains, while eigenvalues and eigenvectors were extracted in a rectangular acoustic cavity.

Keywords: isogeometric analysis; T-splines; Bézier extraction operator; C^0-G^0 continuity; potential problems; Laplace equation; eigenvalue problem

MSC: 65N06; 65B99



Academic Editors: Sheng-Wei Chi and Fernando Simoes

Received: 20 November 2024
Revised: 9 January 2025
Accepted: 20 January 2025
Published: 24 January 2025

Citation: Provatidis, C.; Dimitriou, I. Automatic Handling of C^0-G^0 Continuous Rational Bézier Elements Produced from T-Splines Through Bézier Extraction. *Mathematics* **2025**, *13*, 377. https://doi.org/10.3390/math13030377

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Isogeometric analysis is today's research frontier in computational methods for the solution of BVPs and eigenvalue (Sturm–Liouville) problems. According to this method, the domain is decomposed into smaller patches which are based on tensor-product NURBS approximation [1]. Later, the method was implemented in conjunction with T-spline approximation [2]. While in both of these cases the basis functions may be computed as tensor-products which are based on either global or local knot vectors, it has been proposed instead to implement the Bézier extraction (BEXT) operator [3,4].

The main advantage of this approach is that Bézier extraction, compared to the original implementation of IGA given in [1], is such that apart from the coefficients of the extraction operator, the basis functions are identical for all elements in the mesh, as is the case for classical finite elements. Therefore, there is no need to implement B-spline (basis) function evaluation routines; these are costly, from a numerical point of view. Another practical reason for using BEXT is that it reduces the communication errors in the data transfer between the geometric model and the analysis.

On the other hand, the BEXT operator is a matrix which, when multiplied by the original control points of the T-mesh, provides a greater number of new (updated) control

points; these have not been exploited to date. In this paper, we propose the utilization of the new control points, which form Bézier elements of C^0-continuity, to automatically solve a parallel BVP. The latter solution is generally more accurate for the same number of elements as found in the original T-spline, and therefore may be used as a reference in an a posteriori error estimator. The next step is obviously a refined T-spline classical isogeometric analysis.

2. Conventional IGA Procedures

2.1. Tensor-Product of Local Knot Vectors (MODEL-1)

It is well known that for two given global knot vectors associated with a patch, the tensor-product B-spline (or NURBS) functions can be determined using Cox–de Boor recursion [5,6]. The same can be performed in T-spline approximation, with the only exception being that the tensor-product is based on local knot vectors associated with anchors.

For the sake of simplicity, we limit ourselves to cubic T-splines (i.e., those with polynomial degree $p = 3$). Therefore, in a usual T-spline it is sufficient to construct a matrix, let us call it U_{vec}, which for each anchor 'α' determines the knots of those other anchors to the left and the right of that under consideration, and thus forming the local knot vector $\Xi_\alpha = \{\xi_1, \xi_2, \xi_3, \xi_4, \xi_5\}$. Similarly, we need another matrix, let us call it V_{vec}, which determines the knots of those anchors at the bottom and the top of that under consideration, and thus forming the local knot vector $H_\alpha = \{\eta_1, \eta_2, \eta_3, \eta_4, \eta_5\}$. Again, for $p = 3$, the matrix U_{vec} (and similarly V_{vec}) is of the size $n_\alpha \times 5$, where n_α is the number of anchors in the T-patch. Based on these two matrices as input, for each couple (ξ, η) of the T-patch, we can determine the tensor-product of B-splines, $N_{\alpha,p}(\xi)N_{\alpha,p}(\eta)$.

The abovementioned B-spline functions, $N_{\alpha,p}(\xi)$ and $N_{\alpha,p}(\eta)$, can be easily calculated using the function `spcol` included in MATLAB® R2018, but open-source software is also available in many repositories and books (e.g., [7]).

Moreover, in the case of $p = 3$, for any set of five nondecreasing knots $\{x_1, x_2, x_3, x_4, x_5\}$, which may be a single row from the abovementioned matrix U_{vec}, there is a single piecewise cubic B-spline function $N_\alpha(x)$, which analytically is given by the following (see, e.g., [8]):

$$N_\alpha(x) = \begin{cases} \frac{(x-x_1)^3}{(x_2-x_1)(x_4-x_1)(x_3-x_1)}, & 0 < x \le x_2 \\ \frac{(x-x_1)^2(x_3-x)}{(x_3-x_2)(x_4-x_1)(x_3-x_1)} + \frac{(x_4-x)(x-x_1)(x-x_2)}{(x_3-x_2)(x_4-x_2)(x_4-x_1)} + \frac{(x_5-x)(x-x_2)^2}{(x_3-x_2)(x_5-x_2)(x_4-x_2)}, & x_2 < x \le x_3 \\ \frac{(x-x_1)(x_4-x)^2}{(x_4-x_3)(x_4-x_2)(x_4-x_1)} + \frac{(x_5-x)(x_4-x)(x-x_2)}{(x_4-x_3)(x_5-x_2)(x_4-x_2)} + \frac{(x_5-x)^2(x-x_3)}{(x_4-x_3)(x_5-x_3)(x_5-x_2)}, & x_3 < x \le x_4 \\ \frac{(x_5-x)^3}{(x_5-x_4)(x_5-x_3)(x_5-x_2)}, & x_4 < x \le x_5 \\ 0, & x \le x_1 \text{ or } x > x_5 \end{cases} \quad (1)$$

In general, the independent variable x and the knots x_1 to x_5 shown in Equation (1) may represent either of the parameters ξ or η, as well as the knots involved in the abovementioned local knot vectors $\Xi_\alpha = \{\xi_1, \xi_2, \xi_3, \xi_4, \xi_5\}$ and $H_\alpha = \{\eta_1, \eta_2, \eta_3, \eta_4, \eta_5\}$. Therefore, for each anchor 'α', Equation (1) applied to Ξ_α determines one B-spline $N_\alpha(\xi)$, and applied to H_α produces $N_\alpha(\eta)$. The range of the non-vanishing tensor-product bivariate basis function $N_{\alpha,p}(\xi)N_{\alpha,p}(\eta)$ is the rectangle $[\xi_1, \xi_5] \times [\eta_1, \eta_5]$. In general, a weight w_α is associated to the anchor $\mathbf{P}_\alpha$.

Based on the above B-spline functions $N_\alpha(\xi)$ and $N_\alpha(\eta)$, for each anchor 'α' we can calculate the bivariate blending function $B_\alpha(\xi, \eta) = N_\alpha(\xi)N_\alpha(\eta)$. When the patch is

curvilinear, each anchor is also associated to a weight w_α. Following the notation in [2], if 'A' is the set of all the anchors in the patch, we define the blending functions as follows:

$$R_\alpha(\xi,\eta) = \frac{w_\alpha B_\alpha(\xi,\eta)}{\sum_{\beta\in A} w_\beta B_\beta(\xi,\eta)} \tag{2}$$

Clearly, the numerator of Equation (2) represents a weighted tensor-product of the local B-spline functions associated to the anchor 'α'. The denominator $W(\xi,\eta)$ represents the sum of the weighted tensor-products due to all the contributed anchors, and is set to ensure the partition of unity property at every point (ξ,η) of the parametric space:

$$\sum_{\alpha\in A} R_\alpha(\xi,\eta) = 1,\ \forall(\xi,\eta) \in [0,1]\times[0,1] \tag{3}$$

Based on the above blending functions $R_\alpha(\xi,\eta)$ within a curvilinear patch, and given the control points $\mathbf{P}_\alpha$, the T-spline in physical space is described by the following:

$$S(\xi,\eta) = \sum_{\alpha\in A} \mathbf{P}_a R_\alpha(\xi,\eta) \tag{4}$$

It is instructive to recall that in the special case of a NURBS tensor-product of degree $p = 3$, the abovementioned set 'A' consists of exactly $(p+1)^2$, i.e., sixteen, nonzero bivariate basis functions which fulfil Equation (3). In contrast, in a sub-patch of a T-spline, there may be more than 16 non-vanishing bivariate blending functions. Regarding an annulus, Ref. [4] shows that most T-elements are influenced by 16 (and a few of them by 17) blending functions.

As discussed above, regarding a certain T-spline patch, it is possible that a part of it (an element) is influenced by 16 non-vanishing blending functions, whereas another element might be influenced by 17. Moreover, another possibility is that the same number of 16 functions might be in evidence, but different blending functions may dominate in different elements of the same patch. In all these cases, we talk about "elements with reduced continuity", and to make these elements suitable for isogeometric analysis, it becomes necessary to subdivide them with continuity reduction lines (extended T-mesh), and thus an increased number of T-elements is produced. As a result of this process, we have an "Analysis Suitable T-spline" (AST) patch.

After the creation of analysis-suitable T-elements, the blending functions of Equation (2) may be used to estimate the mass and stiffness matrices associated with the T-patch under consideration.

2.2. The Standard Utilization of the Bézier Extraction Operator (MODEL-2)

Despite the ease in the implementation of Equation (1), and although numerical evaluations may be easily performed using open-source libraries or proprietary libraries, the tendency is to utilize the Bézier extraction (BEXT) version. The basic theory is given below.

In the theory of computer-aided geometric design (CAGD), it is well known that knot insertion does not alter the shape of a curvilinear patch, either in shape or parametrically [7]. For one-dimensional interpolation of a curve $C(\xi)$ with $p = 3$, initially we have C^2-continuity at the inner knots, and after one knot insertion it decreases to C^1-continuity. Continuing with a second knot insertion at the inner knots (thus, three equal knots in total, i.e., equal to degree p), the continuity further decreases to C^0. Therefore, when the multiplicity λ of inner knots becomes equal to the polynomial degree p, we obtain C^0-continuity at the inner knots.

Moreover, for a two-dimensional NURBS patch, the above procedure is again applicable, but then the multiplicity of the inner knot should become $\lambda = p^2$, so that C^0-continuity is obtained in both directions. As shown in [3], the column-vector including the non-zero

B-spline (basis) functions $\mathbf{N}_e$ is related to the extracted Bernstein polynomials stored in the column-vector $\mathbf{B}_e$ per element, through the following formula:

$$\mathbf{N}_e(\xi, \eta) = \mathbf{C}_e \cdot \mathbf{B}_e(\xi, \eta), \tag{5}$$

where $\mathbf{C}_e$ is the Bézier extraction operator matrix, which corresponds to the e-th B-spline tensor-product element. In general, in a NURBS patch, $\mathbf{C}_e$ is a matrix of size $(p+1)^2 \times (p+1)^2$, whereas $\mathbf{N}_e$ and $\mathbf{B}_e$ are column-vectors of size $(p+1)^2 \times 1$, due to the local-support property of B-splines.

In a similar way, for a two-dimensional T-spline patch, Scott et al. [4] presented a pseudocode and showed that multiple knot insertion leads to the "Bézier extraction operator matrix", $\mathbf{C}_e$, which may be used to estimate the C^2-continuous blending functions of the T-spline element assembly. It is clarified that $\mathbf{C}_e$ comes from the same concept as the one underlying the expression in Equation (5) for NURBS patches, but is now applied locally for the T-spline patches.

In more detail, given the initial control points $\mathbf{P}_e$ and their weights $\mathbf{w}_e$ of the C^2-continuous e-th T-spline element, the abovementioned multiple knot insertion alters the initial weights according to the following formula:

$$\mathbf{w}_e^b = \mathbf{C}_e^{\mathrm{T}} \mathbf{w}_e. \tag{6}$$

In Equation (6), $\mathbf{w}_e^b$ denotes the updated element weights after the multiple knot insertion, whereas $\mathbf{w}_e$ are the initial given element weights which are associated to the initial control points $\mathbf{P}_e$.

As we mentioned earlier, within a T-mesh there may be some sub-patches of reduced continuity, and this has been a matter of intensive research [9]. To ensure uniqueness and then perform accurate numerical integration, it becomes necessary to elongate some edges of the T-mesh and thus obtain an analysis-suitable T-mesh. Nevertheless, although this procedure leads to an increased number of analysis-suitable T-elements, a standard number of non-vanishing blending functions is not always ensured.

As we shall see later, in the benchmark test of the annulus (Example 3), which has been studied in [4], most of the elements are influenced by $(p+1)^2 = 16$ nonzero basis functions, but also there are two elements which are influenced by 17 functions. This in turn means that, for most elements, the Bézier extraction operator matrix $\mathbf{C}_e$ will be of size 16×16, but in some of them, it will be of the size 17×16.

For the sake of completeness, the computation of the blending functions is performed as follows:

From Equation (2) one may observe that the bivariate blending functions $R(\xi, \eta)$ are tensor-products of local univariate basis functions, properly normalized, as follows:

$$R_\alpha(\xi, \eta) = \frac{N_{\alpha,p}(\xi) N_{\alpha,p}(\eta) w_\alpha}{W(\xi, \eta)}, \tag{7}$$

where the denominator $W(\xi, \eta)$ is the so-called "weighting function". Each blending function $R_\alpha(\xi, \eta)$ is associated to an anchor $\mathbf{P}_\alpha$, which for the polynomial degree $p = 3$ is a true control point.

At a certain point (ξ, η) of the T-patch, there are a number of 'n' nonzero bivariate basis functions, in sum approximating the value $(p+1)^2$ (e.g., $n = 16, 17, \ldots$).

Dropping out the subscript 'e' denoting the element under consideration, if $\mathbf{W}$ is the diagonal matrix which includes all the above 'n' involved weights,

$$\mathbf{W} = \begin{bmatrix} w_1 & & & \\ & w_2 & & \\ & & \ddots & \\ & & & w_n \end{bmatrix}, \tag{8}$$

and $\mathbf{N}(\xi, \eta)$ is the column-vector (of size $n \times 1$) of all the bivariate B-spline functions affecting the element under consideration, then Equation (7) is rewritten in a matrix form which provides the rational column-vector $\mathbf{R}(\xi, \eta)$ [of size $n \times 1$],

$$\mathbf{R}(\xi, \eta) = \frac{1}{W(\xi, \eta)} \mathbf{W}\mathbf{N}(\xi, \eta) \tag{9}$$

To obtain the blending functions, we still need to calculate the weighting function $W(\xi, \eta)$, which is involved in the denominator of Equation (9). The latter is written as

$$W(\xi, \eta) = \sum_{\alpha=1}^{n} N_{\alpha,p}(\xi, \eta)\, w_\alpha = \mathbf{w}^T \mathbf{N}(\xi, \eta) \tag{10}$$

Substituting the B-spline column-vector $\mathbf{N}$ from Equation (5) into Equation (10), the latter becomes

$$W(\xi, \eta) = \mathbf{w}^T \mathbf{N}(\xi, \eta) = \mathbf{w}^T \mathbf{C}\mathbf{B}(\xi, \eta) \equiv \left(\mathbf{C}^T \mathbf{w}\right)^T \mathbf{B}(\xi, \eta) \tag{11}$$

Substituting Equation (6) into Equation (11), we receive the following:

$$W(\xi, \eta) = \left(\mathbf{w}^b\right)^T \mathbf{B}(\xi, \eta) = W_b(\xi, \eta), \tag{12}$$

where $\mathbf{w}^b = \mathbf{C}^T \mathbf{w}$ is a column-vector of standard size 16×1, including the weights associated to the Bézier–Bernstein basis functions. Substituting Equations (5) and (12) into Equation (9), we eventually receive the following:

$$\mathbf{R}(\xi, \eta) = \frac{1}{W^b(\xi, \eta)} \mathbf{W}\mathbf{C}\mathbf{B}(\xi, \eta), \tag{13}$$

where $\mathbf{W}$ is the diagonal matrix including the initial weights of the control points in the C^2-continuous model (cf. Equation (8)). Equation (13) dictates that it is not necessary to directly calculate the B-spline functions, either through the recursive (Curry–Schoenberg 1966) formulas or by using the analytical ones given by Equation (1), but only the extraction matrix $\mathbf{C}$, which depends only on the knot vectors, while the expression of the tensor-product Bernstein polynomials $\mathbf{B}(\xi, \eta)$ is trivial.

Based on the blending functions given by Equation (13), the computation of the mass and stiffness matrices is straightforward.

3. The Proposed Computational Procedure (MODEL-3 and MODEL-4)

3.1. General Theory (MODEL-3)

Although Ref. [4] has focused on the estimation of the C^2-continuous blending functions according to the BEXT Equation (13), for the purposes of the present paper we focus below on the updated control points as well.

In more detail, given the initial set of control points **P** and their associated weights **w** in the C^2-continuous T-spline model, the multiple knot insertion until the multiplicity becomes equal the degree p, provides new control points **Q** according to the following formula:

$$\mathbf{Q}_e^w = \mathbf{C}_e^{\mathrm{T}} \mathbf{P}_e^w \tag{14}$$

In Equation (14), the superscript 'w' indicates projective coordinates, i.e., Cartesian coordinates multiplied by the associated updated weight, whereas the superscript 'T' indicates the transpose of the BEXT matrix $\mathbf{C}_\mathrm{e}$. Moreover, as previously mentioned in Section 2.2, in Equation (6), $\mathbf{w}_e^b \equiv \mathbf{w}_e^Q$ denotes the updated weights of the new control points **Q** (after the multiple knot insertion), whereas $\mathbf{w}_e \equiv \mathbf{w}_e^P$ are the initially given weights associated to the initial control points **P**. Note that the superscripts $\{Q\}$ and $\{P\}$ are set only to clearly indicate the status after and before knot insertion, respectively.

Regarding the size of the arrays involved, we repeat that the element Bézier extraction operator matrix $\mathbf{C}_e$ includes as many rows as the number of the non-vanishing blending functions within the T-spline element (e.g., 16, 17), whereas the number of columns is standard, and equal to 16. Therefore, in general, $\mathbf{C}_e$ is a non-square matrix. Concerning the element vector of control points $\mathbf{P}_e$ and the associated vector of element weights $\mathbf{w}_e^Q$ in the initial C^2-continuous model, each of them has exactly as many rows as the matrix $\mathbf{C}_e$ (i.e., 16, 17, etc.). Interestingly, whatever the number of rows in $\mathbf{C}_e$ is, the resulting number of updated control points $\mathbf{Q}_e$ in the C^0-continuous Bézier element equals $(p+1)^2 = 16$.

The applied algorithm of the present paper is as follows:

- Based on the initial n_P control points **P** and the given index space, create n_{ele} analysis-suitable T-spline elements.
- Apply the BEXT matrix $\mathbf{C}_e$ in each of the above n_{ele} T-spline elements, and thus determine the following:
  - ○ The updated control points $\mathbf{Q}_e$;
  - ○ The associated weights $\mathbf{w}_e^Q$.
- Find the unique control points $(\mathbf{Q}_e)_{\text{unique}}$ among the above $\mathbf{Q}_e$ (initially including $n_{Q'} = 16 n_{ele}$ points), and thus determine the final number of n_Q control points, which fully defines the n_{ele} rational Bézier elements. An instructive example is presented in Section 3.3. Alternatively, one could implement a variation of the Oslo algorithm [10], which is a recursive procedure designed to simultaneously handle multiple knot insertion.
- For the abovementioned n_{ele} rational Bézier elements, calculate the element stiffness matrix (and mass matrix, if necessary). Note that, since in the present paper only cubic elements have been implemented (degree $p = 3$), there are always $n_{pts} = (p+1)^2 = 16$ control points in each rational Bézier element, and thus for potential problems (one degree of freedom per node) each element matrix has a size of 16×16. Obviously, if the same concept is extended to plane elasticity problems, the stiffness (or mass) matrix of each rational Bézier element will have a size of 32×32, and so on.
- Build the total matrices of the entire T-patch, apply the boundary conditions, and solve the equations system to determine the unknown coefficients.

As we demonstrate below in the five examples of the present paper, the final number, n_Q, of control points in the proposed C^0-continuous model is larger than the initial number of n_P control points in the C^2-continuous one. By analogy to the tensor-product NURBS model, a closed-form expression will be provided below for the quantity n_Q.

3.2. Estimation of Control Points n_Q for C^0-Continuity

To establish the procedure for the estimation of the updated control points **Q** which are produced by the BEXT process, we resort to the conventional tensor-product B-spline approximation. For the sake of simplicity, we consider the piecewise cubic tensor-product B-spline patch shown in Figure 1a (of degree $p = 3$) for the following unidirectional knot vector:

$$\mathrm{U} = [0,0,0,0,\ 1/3,2/3,\ 1,1,1,1] \tag{15}$$

Based on trivial computer-aided geometric design (CAGD)-theory [7] (p.66), the number of control points per direction is given by

$$n = m - p - 1, \tag{16}$$

where m is the number of elements in the knot vector U. Since in Equation (15) there are $m = 10$ knots and $p = 3$, we have $n = 6$ per direction, which means $n_P = n^2 = 36$ control points in the entire two-dimensional patch.

After double knot insertion at the $n_{in} = 2$ inner knots, as required by BEXT [7] (p.161), the number of control points per direction becomes

$$n' = n + n_{in}(p - 1) = 10, \tag{17}$$

and therefore, the total number of the updated control points $\{Q\}$ after BEXT will be

$$n_Q = n'^2 = 10^2 = 100. \tag{18}$$

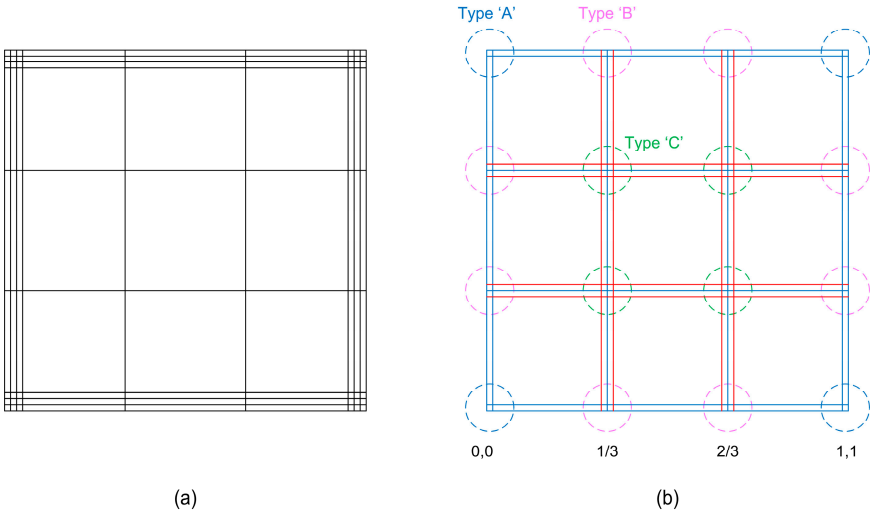


Figure 1. Tensor-product (**a**) B-spline in traditional form; (**b**) T-spline form after knot insertion.

Below, the same result will be derived, considering now the tensor-product B-spline patch as a T-spline patch. The conclusions of this approach will be extended to T-spline patches as well.

Obviously, the above two-dimensional B-spline patch, as described by Equation (15), is equivalent to a T-mesh with six anchors per direction, and thus forming the following index space:

$$A_{\mathrm{x}} = A_{\mathrm{y}} = \{0,\ 0,\ 1/3,\ 2/3,\ 1,\ 1\} = \{\xi_1, \xi_2, \xi_3, \xi_4, \xi_5, \xi_6\}. \tag{19}$$

The anchors in the set described in Equation (19) are denoted by blue-colored lines in Figure 1b. The BEXT procedure requires the knot insertion at the inner knots, such as $\xi_3 = 1/3$ and $\xi_4 = 2/3$, until the multiplicity becomes $\lambda = p = 3$. This demands the

introduction of the red-colored lines shown in Figure 1b, wherein one may distinguish three different types of knots, as follows:

- Type A: Knots related to the corners of the patch (four corners with four anchors per corner).
- Type B: Knots related to intermediate places along the four edges, not coinciding with the corners (six anchors per initial knot).
- Type C: Knots related to initial knots in the interior of the patch (nine anchors per initial knot).

Based on the above categorization, for the case shown in Figure 1b, in which there are four, six, and nine initial knots of categories A, B, and C, circled respectively, we have

$$\text{i.} \quad \text{Anchors at the corners:} \quad 4 \times 4 = 16 \tag{20a}$$

$$\text{ii.} \quad \text{Anchors along boundaries:} \quad 8 \times 6 = 48 \tag{20b}$$

$$\text{iii.} \quad \text{Anchors in interior:} \quad 4 \times 9 = 36 \tag{20c}$$

$$\text{SUM of anchors in the patch: } n_Q = 100, \text{ Q.E.D.} \tag{20d}$$

A procedure similar to that used to determine the number n_Q may be followed for a T-mesh, which is again divided into the above three categories, A, B, and C. The only minor exception concerns elements of reduced continuity, in which extra inner knots are added in the index space by the extension of existing connecting lines, as mentioned earlier. Depending on the location of these extra points, which may also be classified as Type C, the full number '9' or part of it may be applied to them.

3.3. On the Uniqueness of Control Points

Let us continue working with the nine tensor-product B-spline elements illustrated in Figure 1, which are based on the univariate knot vector of Equation (15).

For this configuration, the initial ($n_P = 36$) control points are shown in Figure 2a, whereas those ($n_Q = 100$) after multiple knot insertion are shown in Figure 2b. One may observe that, after the multiple knot insertion, each of the nine Bézier elements is occupied by $4 \times 4 = 16$ control points. Obviously, the number of the actual $n_Q = 100$ control points differs from the blind product $n_{Q'} = 9$ elements $\times$ 16 control points per element = 144.

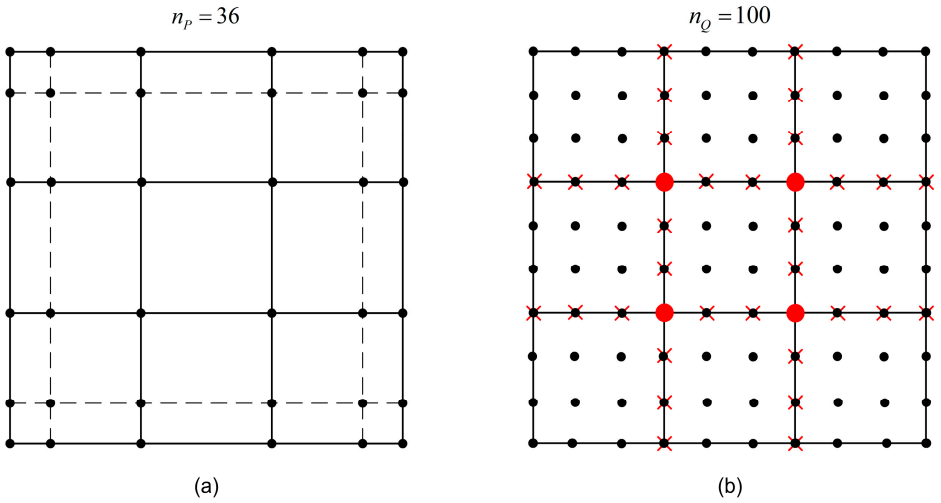


Figure 2. Control points (**a**) before and (**b**) after knot insertion (shared control points: at intersections •; along the inter-element boundaries ×).

The above discrepancy between the erroneous $n_{Q'} = 144$ and the correct $n_Q = 100$ is due to the following reasons:

- Each of the four control points at the intersections between horizontal and vertical inter-element boundaries (illustrated in Figure 2b by red circle, •) belongs to four elements, while they must be countered only once. Therefore, instead of the blind number $4 \times 4 = 16$, we must consider only four of them, which means that we have twelve additional fictitious points to subtract from 144.
- Each of the thirty-two control points along the inter-element boundaries (illustrated in Figure 2b by red cross, ×) belongs to two elements, and therefore a blind computation would result in $32 \times 2 = 64$ points. To obtain the exact number, half of them (i.e., 32) must be subtracted from 144.

Therefore, the total fictitious number of the additional control points is 12 + 32 = 44. When the latter (44) is subtracted from the blind number $n_{Q'} = 144$, we obtain the correct number, $n_Q = 100$.

3.4. Fixing C^0 and G^0-Incompatibilities (MODEL-4)

Although the procedure discussed in Section 3.1 (MODEL-3) is straightforward, sometimes there are minor shortcomings to be corrected before analysis is performed. In this context, it may happen that within a sub-patch there is a "discontinuity", and thus partitioning using an extension line is necessary for the accurate numerical integration (e.g., the dashed line GH shown in Figure 3a). For example, in the test case of [4] (p. 134), it was shown that among twenty-four T-spline elements, twenty-two of them are affected by 16 whereas the other two are affected by by 17 control points.

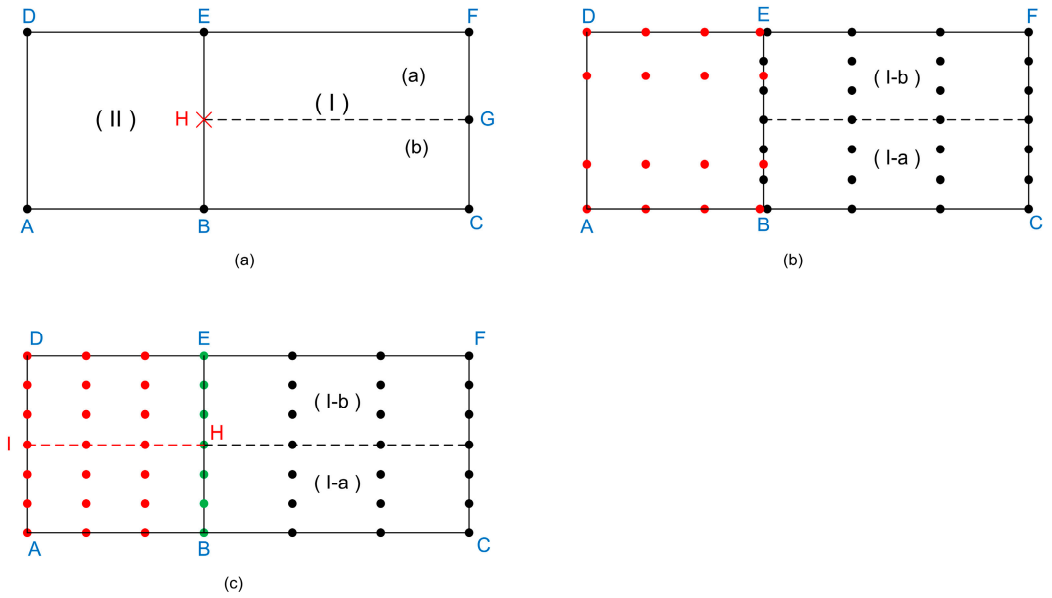


Figure 3. Formation of Bézier elements near to reduced continuity: (**a**) two sub-patches (I) and (II), (**b**) three incompatible elements, and (**c**) four compatible elements.

To be more specific, the discontinuity of the sub-patch (I) in Figure 3a demands its subdivision (using the abovementioned extension line GH) into sub-patches I(a) and I(b). After the introduction of the line GH, the model becomes "analysis-suitable" (AS). This means that each of the produced new elements I(a) = HGFE and I(b) = BCGH is continuous and thus accurate numerical integration is allowed. This task is compulsory even in the original IGA and is performed using only information provided by the index space. However, in the original IGA, the hanging node H is no problem at all, because

it is not associated with an independent degree of freedom (and thus the compatibility makes no sense), but rather—as was previously mentioned—the line GH serves only for domain integration.

On the contrary, in the framework of the present paper it was found that although the abovementioned Bézier elements (I(a) and I(b)) are well formed, their neighboring Bézier element (II), which shares a common edge BE (perpendicular to the end H of the extension line GH) with the sub-patch (I) of reduced continuity, is not compatible (see also Figure 3b). Clearly, this happens although the Bézier element (II) is fully continuous. Although sometimes the accuracy of the numerical solution may be sufficient while ignoring the incompatibility, in general it becomes imperative to perform a further tessellation on the largest element (II), as shown in Figure 3c. The procedure of this tessellation is easy and is based on the common edge previously considered, from the side of the element (I) of reduced continuity. A full description of the applied numerical scheme is given in Appendices A and B.

Therefore, given the index space of a T-spline, the first step in IGA is to introduce the extension lines GH, and thus the number of incompatibilities equals the number of $n_{\text{incompatible}}$ points H at the ends of them. Before the tessellation (MODEL-3), the total number of Bézier elements is n_{nele}, of which a set of $n_{\text{incompatible}}$ (like the abovementioned element (II)) is incompatible. Therefore, $(n_{\text{nele}} - n_{\text{incompatible}})$ out of the initial n_{nele} elements are compatible.

Regarding MODEL-4, we recall that the common edges at which tessellation is further conducted are perpendicular to the ends H of the extension lines GH, whereas the total number of Bézier elements becomes $(n_{\text{nele}} + n_{\text{incompatible}})$. As for the extra computation effort of MODEL-4 with respect to MODEL-3, the former includes one extra rational Bézier element per incompatibility, which means that the extra computer effort for the estimation of the element matrices is $n_{\text{incompatible}}/n_{\text{nele}}$ of the computer effort in MODEL-3. (Note that the matrices of $(n_{\text{nele}} - n_{\text{incompatible}})$ elements remain unaltered.)

4. Matrix Formulation

Numerical results refer to potential problems, particularly with reference to the Laplace equation and wave propagation equation. The latter partial differential equation is written as

$$(1/c^2)\partial^2 u/\partial t^2 - \nabla^2 u = 0. \tag{21}$$

Based on any set of control points (**P** or **Q**) associated with blending functions $\mathbf{R}_{\text{PQ}}$ (column-vector) in general, the matrix formulation of IGA is

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{f}(t), \tag{22}$$

where the stiffness and mass matrices as given by the following domain integrals:

$$\text{Mass matrix:} \qquad \mathbf{M} = (1/c^2)\int_A \mathbf{R}_{PQ}(\mathbf{R}_{PQ})^{\mathrm{T}} dA, \tag{23}$$

and

$$\text{Stiffness matrix:} \qquad \mathbf{K} = \int_A \nabla\mathbf{R}_{PQ}(\nabla\mathbf{R}_{PQ})^{\mathrm{T}} dA., \tag{24}$$

Obviously, Laplace equation ($\nabla^2 u = 0$) is a special, steady-state, case of Equation (22), and thus is described by the matrix equation

$$\mathbf{K}\mathbf{u} = \mathbf{f}, \tag{25}$$

where the force vector is given as

$$\mathbf{f} = \oint_{\Gamma} \mathbf{R}_{PQ} \frac{\partial u}{\partial n} d\Gamma. \tag{26}$$

Regarding the numerical extraction of the eigenvalues ($\lambda = \omega^2$) of an acoustic cavity, we use the following formula:

$$\det(\mathbf{K} - \lambda \mathbf{M}) = 0. \tag{27}$$

5. Results

The theory is supported by five examples. The first two examples are simple rectangles and have been devised to require edge extension in the same direction at two places to ensure analysis-suitable T-splines. The third example refers to an annulus and follows the pattern of [4], which also requires two edge extensions, but in this case, in relatively perpendicular directions. The fourth example utilizes a more complicated index space, as it refers to a fully two-dimensional problem. The fifth example has an index space similar to that of the annulus (third example) but refers to a rectangular domain.

Each example was solved using four models, as follows:

- MODEL-1: B-spline functions **N** were calculated as local tensor-product associated to control points **P**, using de Boor's spcol MATLAB® function; Equation (1) is also applicable as an alternative. Furthermore, weights and normalization were imposed to obtain the final blending functions **R**.
- MODEL-2: B-spline functions **N** were calculated using Bézier extraction operator matrices, $\mathbf{C}_e$, which were associated to initial control points **P**. Furthermore, weights and normalization were imposed to obtain the final blending functions **R**.
- MODEL-3: Basis functions **B** (Bézier–Bernstein polynomials) are known functions and were combined with the new control points **Q** on the Bézier elements of C^0-continuity.
- MODEL-4: When MODEL-3 leads to inter-element incompatibility like that demonstrated in Figure 3, the larger Bézier element is subdivided according to Appendices A and B, thus eventually producing the further-updated set of control points $\mathbf{Q}'$, which is slightly larger than **Q**. Based on the further corrected weights $\mathbf{w}_{Q'}$ and the further updated set of control points $\mathbf{Q}'$, we construct the new rational Bernstein polynomials $\mathbf{B}'$ which constitute the final set of basis functions.

By the theory, MODEL-1 and MODEL-2 are identical, because they only use a different computational procedure to calculate the same B-spline functions **N**, which are associated to the same control points **P**, and at the same Gauss points. The only reason that MODEL-2 is separately mentioned here is because it crosschecks the accuracy of the computer code and thus increases confidence.

In each example, for the first three models (MODEL-1 to MODEL-3), the number of (T-spline or Bézier) elements and the number of Gauss points is the same. Nevertheless, the number of the involved DOFs of MODEL-3 differs from those of (MODEL-1 and MODEL-2), i.e., we always have $n_Q > n_P$, as was explained in Sections 3.1 and 3.2.

Furthermore, in MODEL-4 the number of elements is larger by those subdivided, whereas the number of DOFs ($n_{Q'}$) is slightly larger than n_Q.

In the steady-state problems (Laplace equation), the accuracy of the models was expressed in terms of the L_2-norm in percent (%), using the following formula:

$$L_2 = \left[\frac{\int_\Omega \left(u_{\text{calculated}} - u_{\text{exact}}\right)^2 d\Omega}{\int_\Omega \left(u_{\text{exact}}\right)^2 d\Omega} \right]^{\frac{1}{2}} \times 100(\%) \tag{28}$$

All computations were performed on the same PC, using MATLAB® R2018.

5.1. EXAMPLE 1: Vertical Heat Flow

In the rectangular domain of size $L \times H = 7 \times 4$, of which the index space is shown in Figure 4, the temperature is prescribed on the bottom and top edges as follows: $T_{top} = 1000$ °C (at $y = H$) and $T_{bottom} = 0$ °C (at $y = 0$); the vertical edges at $x = 0$ and $x = L$ are fully insulated ($\partial T/\partial x = 0$).

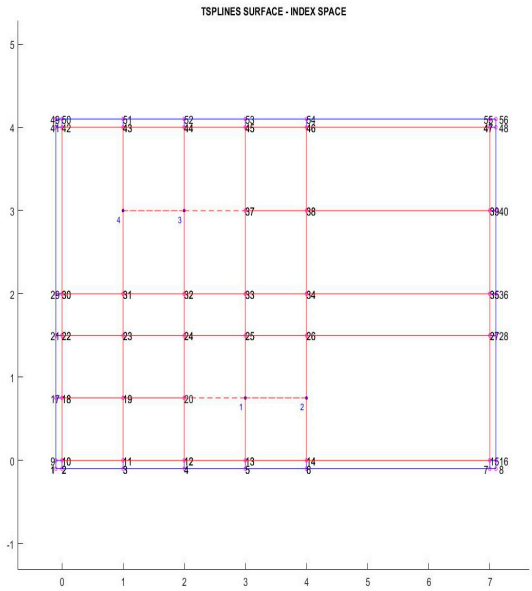


Figure 4. Index space for Example 1 and Example 2.

One may observe in Figure 4 that the index space includes $n_P = 56$ anchors and 19 initial patches. However, to ensure continuity, it becomes necessary to extend the edge 19–20 by two units to the right (thus producing the new points 1 and 2, connected by dashed line), and to extend the edge 38–37 to the left (thus producing the new points 3 and 4, connected again by dashed line). In this way, the number of degrees of freedom (DOF) remains the same at $n_{\mathrm{DOF}} = 56$, but due to the two subdivisions (each by two units), the number of T-spline elements in which Gauss integration will be performed increases, from the initial 19 to, eventually, $n_{ele} = 23$.

The numerical solution using MODEL-1 and MODEL-2, which is associated to the system of $n_{\mathrm{DOF}} = 56$ DOF shown in Figure 4, was found to coincide with the exact solution, $T_{exact}(x, y) = 250y$, with $0 \leq y \leq 4$.

Regarding the formation of Bézier elements produced after multiple knot insertion using BEXT (which constitutes MODEL-3), one may observe in Figure 5 two shortcomings regarding compatibility, as described in the following. The first shortcoming concerns the element at the upper left corner along the edge (31–43 in Figure 4, 108–159 in Figure 5) and the second refers to the element in the lower right corner along the edge labelled (14–26 in Figure 4, 43–64 in Figure 5). Nevertheless, the numerical solution of MODEL-3 ($n_{\mathrm{Q}} = 244$ DOF) was also found to coincide with the exact solution. It is hypothesized that this happens because the heat flow is vertical, and the existing vertical edges do not interrupt the continuity of the interpolation. It is worth mentioning that the implementation of MODEL-4 (not shown but explained in Example 2) did not change the excellent result of MODEL-3 at all.

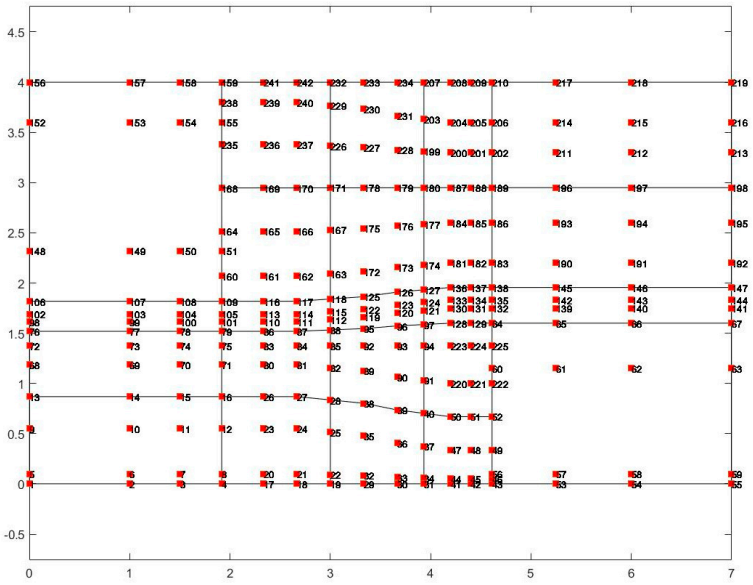


Figure 5. Bézier elements after Bézier extraction in vertical heat flow (Example 1, MODEL-3).

It is worth mentioning that the number of $n_Q = 244$ control points could be predicted by classifying the anchors into three categories according to Section 3.2. Thus, considering (A: 4 corners, B: 14 intermediate anchors on boundary, and C: 9 internal anchors including the extra ones produced after extension), the result, $n_Q = 244$, is shown in Table 1.

Table 1. Estimated number of updated control points (n_Q).

Control Points in MODEL-3			
Type A	**Type B**	**Type C**	**TOTAL (n_Q)**
4×4	14×6	16×9	244

The distribution of the temperature using MODEL-3 is shown in Figure 6, but it is the same in all four models.

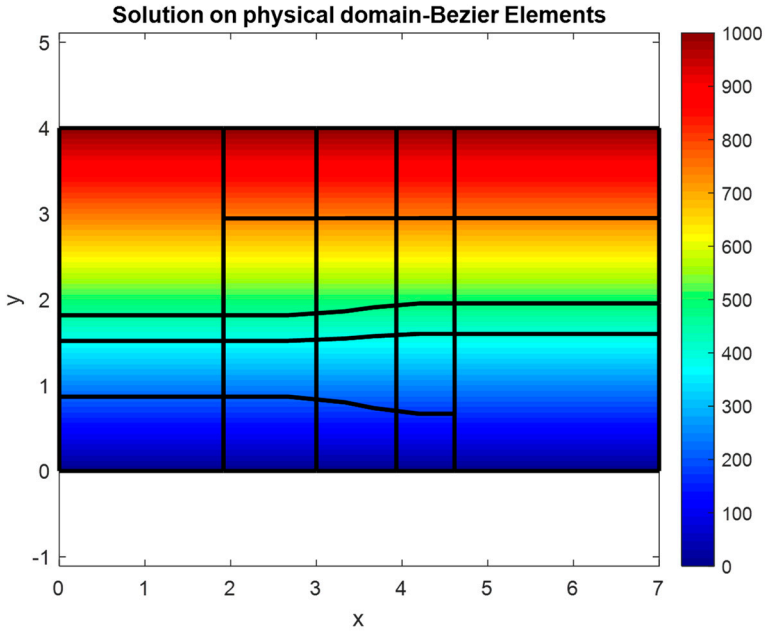


Figure 6. Temperature distribution in vertical heat flow (Example 1).

5.2. EXAMPLE 2: Horizontal Heat Flow

The same geometric model as that of Example 1 is now solved under different Dirichlet boundary conditions: $T_{left} = 0$ °C and $T_{right} = 1000$ °C on the vertical edges, whereas the bottom and top edges are now thermally insulated ($\partial T/\partial y = 0$). Obviously, the heat flow is horizontal, and the exact solution is given by $T_{exact}(x, y) = (x/7)1000$ with $0 \leq x \leq 7$.

Again, the T-mesh is the same as that shown in Figure 4, whereas the obtained set of $n_{ele} = 23$ Bézier elements after Bézier extraction is again that shown in Figure 5 (for MODEL-3).

As previously was the case, again in Example 2, MODEL-1 and MODEL-2 resulted in zero errors across the entire domain, as shown in Figure 7.

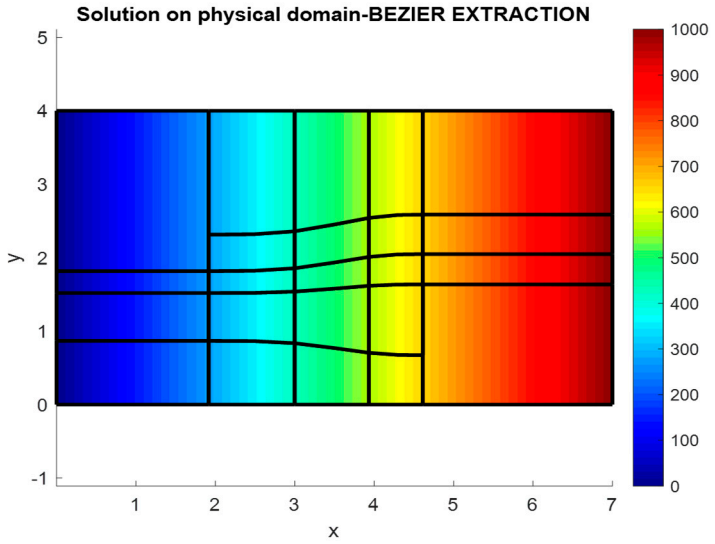


Figure 7. Temperature distribution in MODEL-1 and MODEL-2.

Nevertheless, now, the average numerical error of MODEL-3 (Figure 5, 23 Bézier elements) is substantially larger, equal to about 8.39%, as shown in Figure 8. This problematic result obviously happens because the heat flux $(\partial T/\partial x)$ is discontinuous when it passes through the incompatible vertical edge 108–159 in the large Bézier element at the upper left corner, as well as from the vertical edge 43–64 in the large Bézier element at the lower right corner.

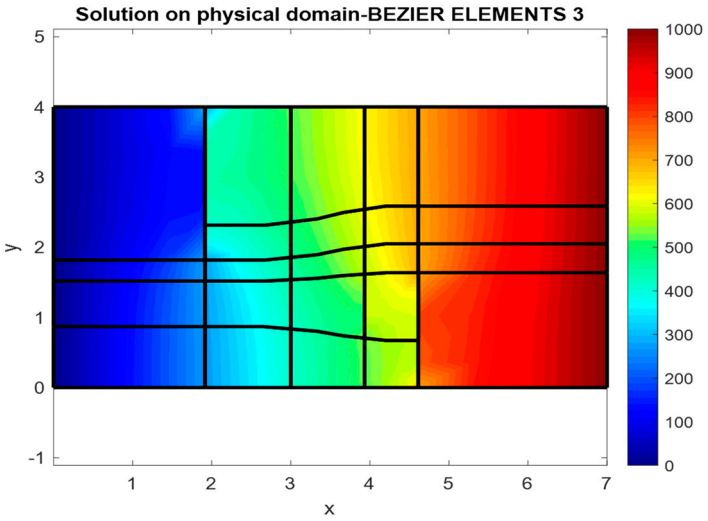


Figure 8. Temperature distribution in MODEL-3.

To fix this error, we must ensure continuity along the two abovementioned interfaces 108–159 and 43–64 by splitting the neighboring elements and thus obtaining $(n_{ele})_4 = 25$ Bézier elements (an extra 2, in addition to previously $(n_{ele})_3 = 23$ elements) associated to $n_{Q''} = 256$ nodes (previously $n_Q = 244$). This configuration defines MODEL-4, shown in Figure 9, which eventually leads to zero error (precisely, $L_2 = 4.91 \times 10^{-10}\%$). The distribution of the temperature for MODEL-4 is perfect, as shown in Figure 10.

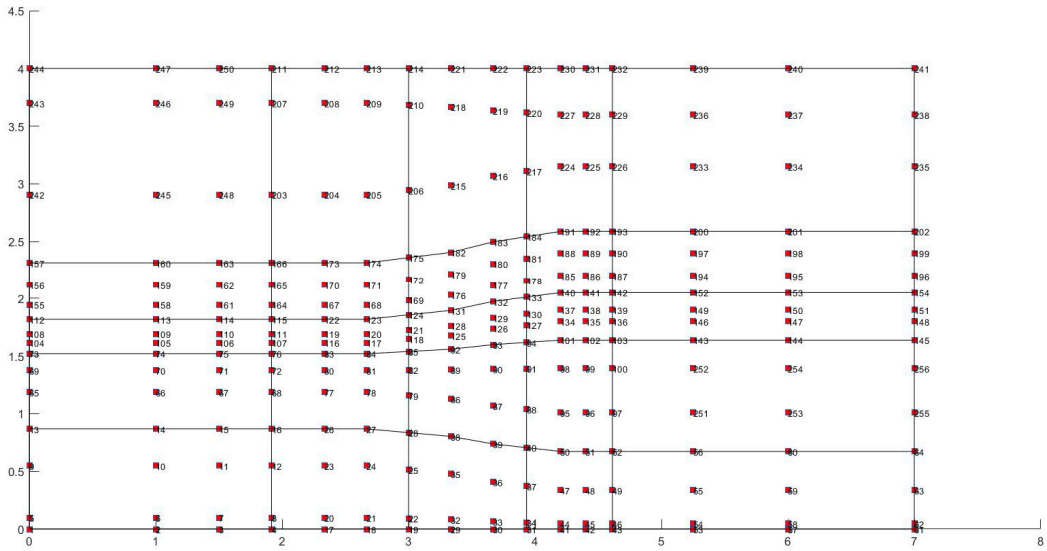


Figure 9. Final set of Bézier elements after two subdivisions (Example 2, MODEL-4).

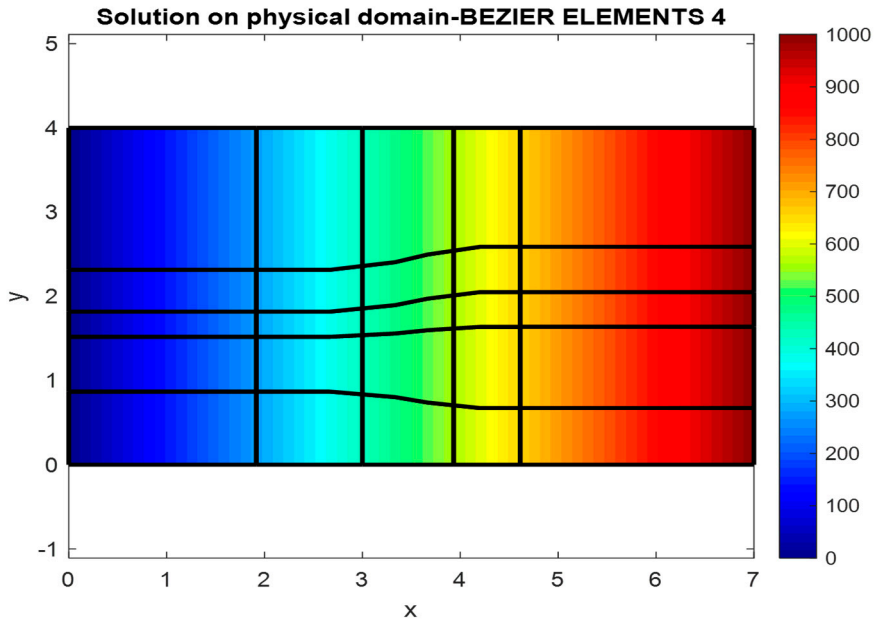


Figure 10. Example 2: Temperature distribution using MODEL-4.

5.3. EXAMPLE 3: Annulus

5.3.1. T-Spline Model

This example is a standard benchmark test [4], originally from Scott's Ph.D. thesis [11], for which all technical data (coordinates, weights) have been provided. It refers to an annulus of a central angle of 90 degrees, with internal radius $R_1 = 1.5$ and external radius $R_2 = 3$. The index space, for polynomial degree $p = 3$, is shown in Figure 11.

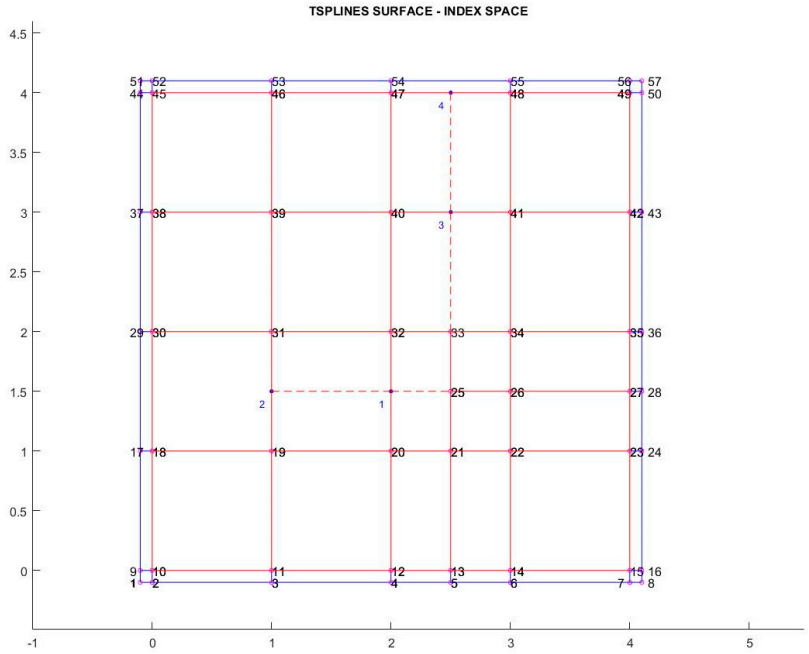


Figure 11. Example 3 and Example 4: Index space.

One may observe in Figure 11 that the index space consists of $n_P = 57$ anchors, and those numbered '25' and '33' are of valence 3. Therefore, as previously, we must extend the associated central edge ('26-25' and '25-33', respectively) to recover the reduced continuity along the pseudo-lines '25-1-2' and '33-3-4', respectively. In this way, the initial 20 sub-patches increase to $n_{ele} = 24$ T-spline elements (note that those between double boundaries are of zero area and thus ignored). In more detail, the sub-patch bounded by the anchors '19-20-32-31' splits into '19-20-1-2' and '2-1-32-31'. Similarly, the sub-patch '20-21-33-32' splits into '20-21-25-1' and '1-25-33-32'. Moreover, the sub-patch '32-33-34-41-40' splits into '32-33-3-40' and '33-34-41-3'. Finally, the sub-patch '40-41-48-47' splits into '40-3-4-47' and '3-41-48-4'. It is worth mentioning that, even after the final decomposition in $n_{ele} = 24$ C^2-continuous elements, the number of DOFs remains at the initial $n_P = 57$ anchors (the double anchors along the boundary are also included).

For the above model of $n_P = 57$ DOF (MODEL-1 and MODEL2), the result is shown in Figure 12.

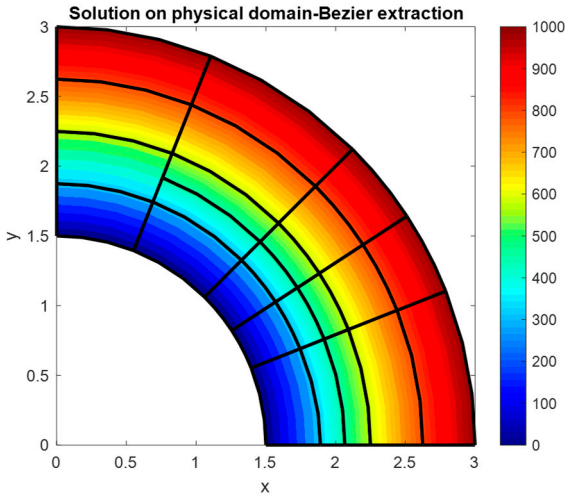


Figure 12. Example 3: Temperature distribution (MODEL-1 and MODEL-2).

Moreover, in Figure 13 we present the $n_{ele} = 24$ Bézier elements associated to $n_Q = 247$ unique nodes (MODEL-3), exactly as were produced using the Bézier extraction operator matrix. One may observe that the second element from the bottom in the first column (also shown as '18-19-31-30' in Figure 11) is not compatible with its neighboring element (the edge '19-2-31' of Figure 11).

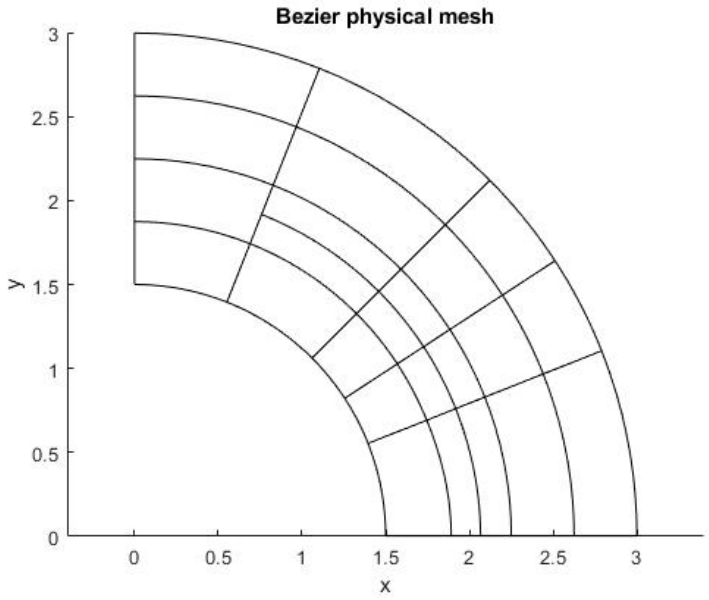


Figure 13. Example 3 and Example 4: Incompatibility near the element of reduced continuity (MODEL-3).

After the tessellation of the abovementioned element '18-19-31-30' (of Figure 11) in the vertical direction, the number of nodes in the assembly of the final $n'_{ele} = 25$ Bézier elements further increases, from $n_Q = 247$ to $n_{Q''} = 256$, as shown in Figure 14 (MODEL-4).

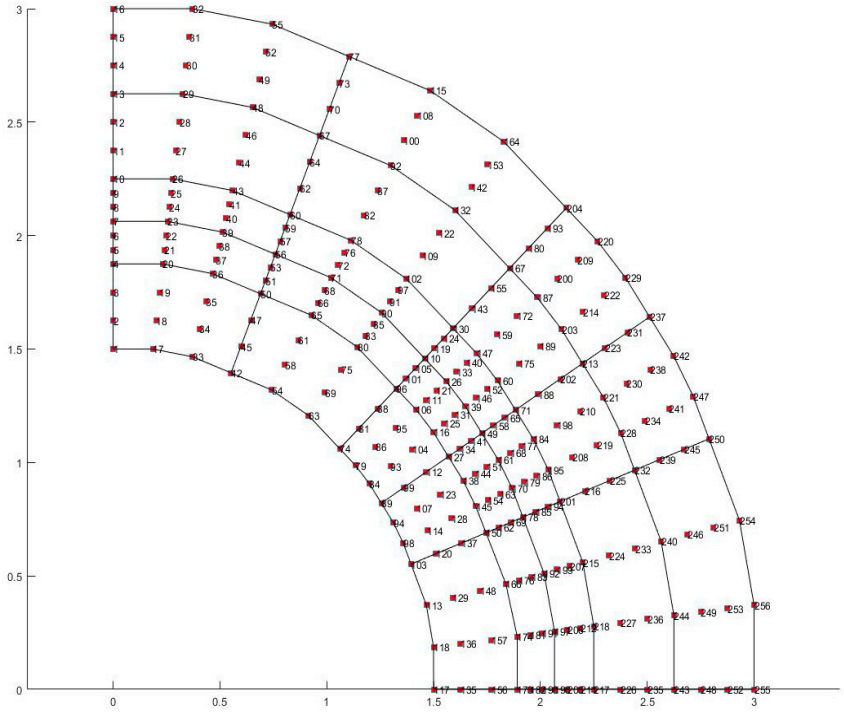


Figure 14. Final configuration of 25 compatible Bézier elements (MODEL-4).

For all of the four models, the calculated average errors, according to Equation (28), are given in Table 2. One may observe that MODEL-3 is worse than MODEL-1 and -2, the

value for which was previously presented in Figure 12 (due to incompatibility), whereas MODEL-4 is the most accurate.

Table 2. Calculated errors for T-mesh (L_2-norm in percent).

L_2-Norm in Percent (%)			
MODEL-1 (57 DOF)	**MODEL-2** (57 DOF)	**MODEL-3** (247 DOF)	**MODEL-4** (256 DOF)
0.0058	0.0058	0.0098	0.0015

5.3.2. Tensor-Product Model

For the sake of completeness and to sustain the tendency of the accuracy in the four models tested, we also present results obtained using the tensor-product using uniform knot distribution. More precisely, for both directions we used the uniform knot vector:

$$U = [0, 0, 0, 0,\ 1/4, 2/4, 3/4,\ 1, 1, 1, 1]. \tag{29}$$

This means four uniform subdivisions per direction (both circumferential and radial), and thus $n_P = 7^2 = 49$ DOFs, for MODEL-1 and MODEL-2.

Regarding MODEL-3 and MODEL-4, obviously they are identical, consisting of 4×4 = 16 Bézier elements, and associated to $n_Q = 13^2 = 169$ DOFs.

For all of the four models, the results are shown in Table 3. One may observe here that, in contrast to the T-spline model (reported in Table 2), the MODEL-3 is superior to MODEL-1 and MODEL-2 (has half the error), because it also represents MODEL-4, which in all the previous cases was the best overall. However, we must consider that MODEL-4 used 3.4 times more DOFs than MODEL-1 and MODEL-2.

Table 3. Calculated errors for the tensor-product (L_2-norm, in percent).

L_2-Norm in Percent (%)			
MODEL 1 (49 DOF)	**MODEL 2** (49 DOF)	**MODEL 3** (169 DOF)	**MODEL 4** (169 DOF)
1.0785935×10^{-3}	1.0785935×10^{-3}	5.2938191×10^{-4}	5.2938191×10^{-4}

5.4. EXAMPLE 4: Thermal Analysis of Rectangular Domain

This example is a fully two-dimensional problem in which the Laplace equation dominates ($\nabla^2 u = 0$), under partial Dirichlet and Neumann boundary conditions. It refers to a rectangular domain of size $a \times b = 7 \times 4$ (Figure 15), of which the temperature along the right vertical edge is given as

$$u(x = a, y) = u_m \cos\left(\frac{\pi y}{2b}\right),\ 0 \le y \le b. \tag{30}$$

The exact solution is given as

$$u(x, y) = u_m \frac{\sinh\left(\frac{\pi x}{2b}\right)}{\sinh\left(\frac{\pi a}{2b}\right)} \cos\left(\frac{\pi y}{2b}\right),\ 0 \le x \le a,\quad 0 \le y \le b. \tag{31}$$

The index space consists of 54 anchors (Figure 16) and requires extension in all of the four directions. Therefore, using an in-house computer code we have inserted four new knots (55, 56, 57, and 58), dividing an element into three parts (i.e., 22-55-57-56, 56-57-58-31, and 55-23-32-58-57), in combination with their updated weights.

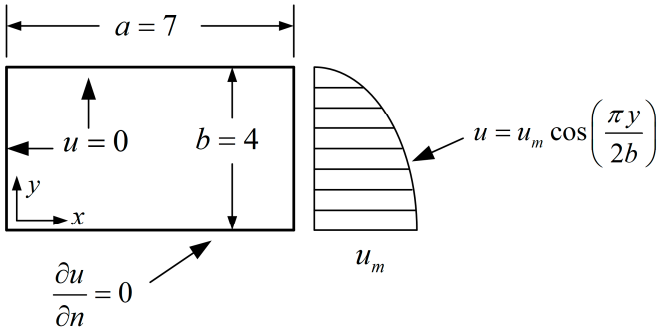


Figure 15. Dimensions and boundary conditions for the rectangular domain.

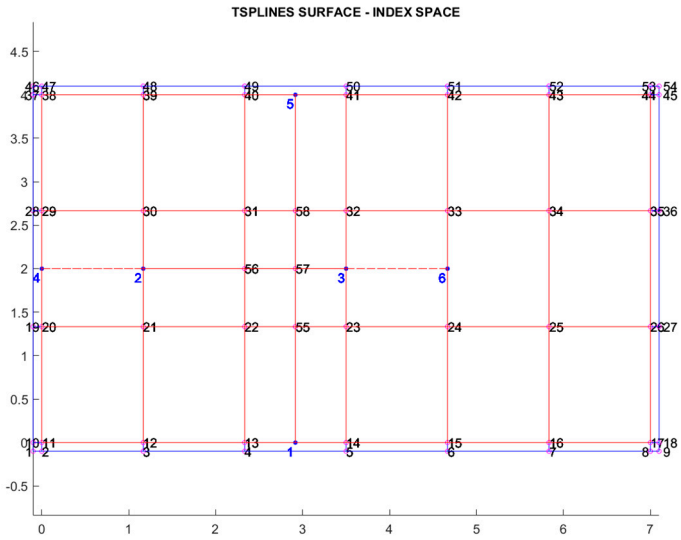


Figure 16. Index space for the rectangular domain.

The initial ANS-model consists of the 26 elements which are used in the first three computational models (MODEL-1, MODEL-2, and MODEL-3). Due to the incompatibility along the double face 24-33 shown in the index space (Figure 16), the patch (24-25-34-33) is subdivided into two parts and thus in MODEL-4 the number of Bézier elements increases (from 26) to 28, as shown in Figure 17. The deterioration of the numerical solution in MODEL-3 and its spectacular improvement in MODEL-4 is shown in Table 4. The distributions of the exact and calculated temperatures are shown in Figures 18 and 19.

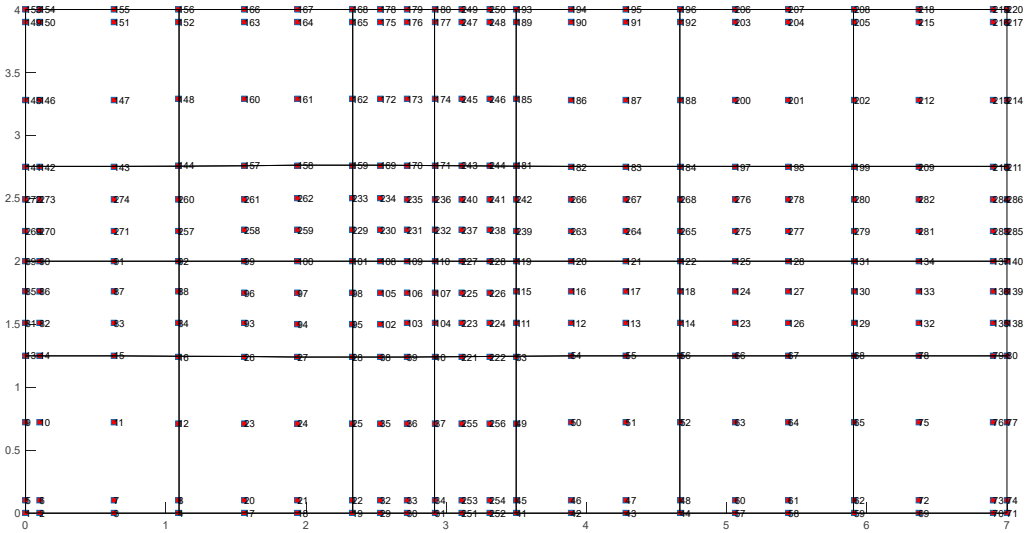
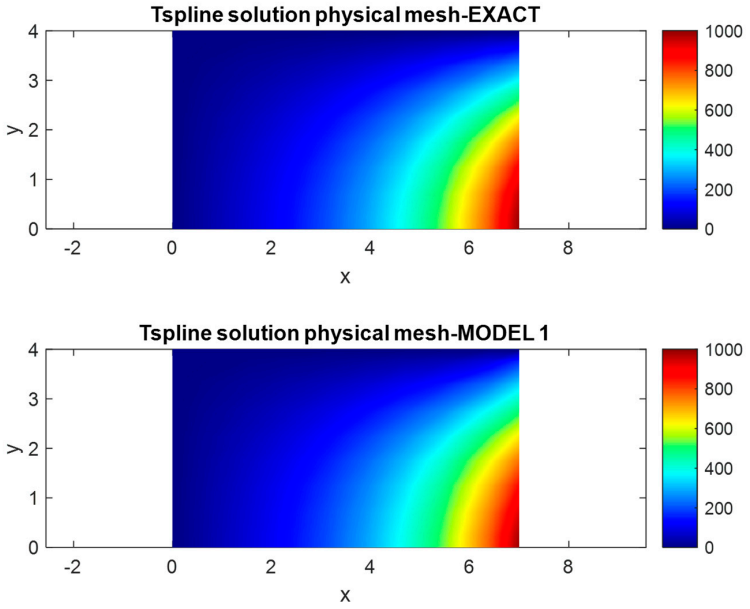


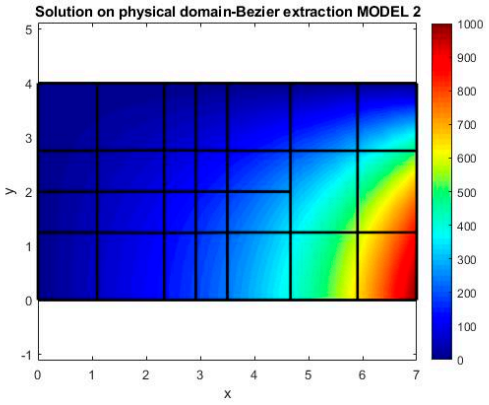
Figure 17. Final configuration of 28 Bézier elements in MODEL-4.

Table 4. Calculated errors for rectangular domain (L_2-norm, in percent).

L_2-Norm in Percent (%)			
MODEL 1 (54 DOF)	**MODEL 2** (54 DOF)	**MODEL 3** (270 DOF)	**MODEL 4** (286 DOF)
2.6992	2.6992	4.9125	0.0051

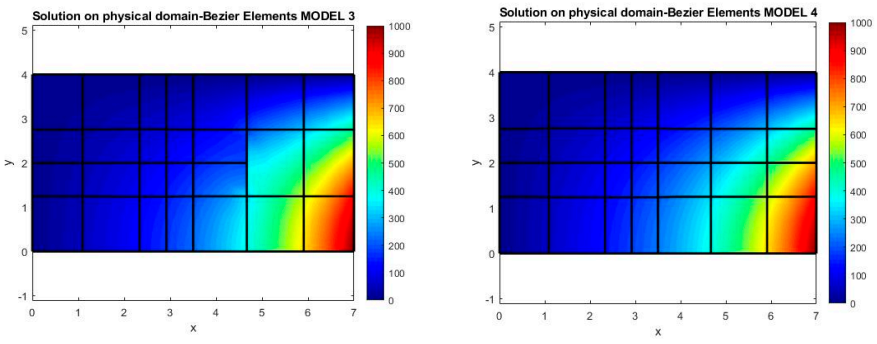


(**a**)



(**b**)

Figure 18. Temperature distribution (top: exact solution; bottom: (**a**) MODEL-1 and (**b**) MODEL-2).



(**a**) (**b**)

Figure 19. Temperature distribution ((**a**) MODEL-3 and (**b**) MODEL-4).

5.5. EXAMPLE 5: Rectangular Acoustic Cavity

We consider a rectangular cavity of size $a \times b = 3.5\ \text{m} \times 1.5\ \text{m}$, and wave velocity $c = 1\ \text{m/s}$, under Neumann boundary conditions ($\partial u/\partial n = 0$). The closed-form analytical eigenvalues may be found in [12].

The index space is shown in the previous figure, Figure 11, and thus consists of $n_P = 57$ anchors. All of the four models were taken to be the same as those in Example 3.

For each separate acoustic mode, the error (in percent) was calculated according to the following formula:

$$\text{Error(in \%)} = \frac{\omega^2_{\text{calculated}} - \omega^2_{\text{exact}}}{\omega^2_{\text{exact}}} \times 100(\%). \tag{32}$$

In Figure 20, we present the error of the first fifty calculated eigenvalues using MODEL-1 (57 DOFs, 24 elements), MODEL-3 ($n_Q = 247$ DOFs, $n_{ele} = 24$ elements), and MODEL-4 ($n'_Q = 256$ DOFs, $n'_{ele} = 25$ elements). One may observe the superiority of MODEL-4 in the entire spectrum of frequencies, mostly in the lower ones, in which MODEL-3 (although better than MODEL-1 and MODEL-2) slightly underestimates the exact eigenvalues.

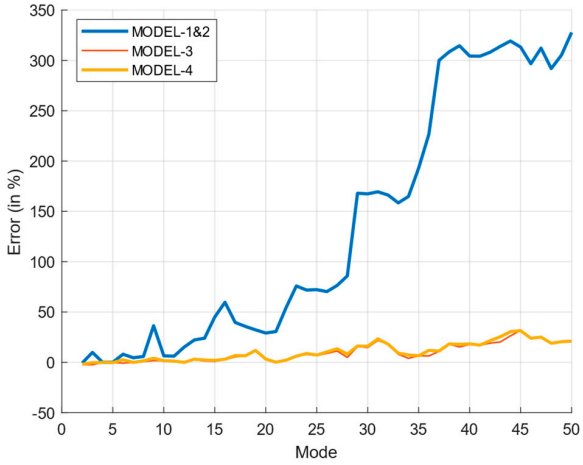


Figure 20. Calculated eigenvalues for the rectangular acoustic cavity.

Moreover, we also present the second normalized calculated eigenvector in Figure 21, as calculated in MATLAB®, which is consistent with the properly scaled exact eigenvector $u(x, y) = \cos(\pi x/a)$.

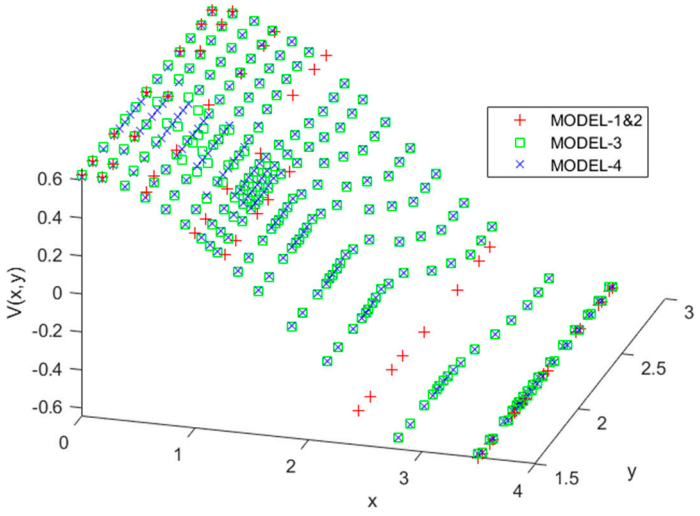


Figure 21. The second calculated eigenvector of the acoustic cavity.

Overall, one may observe that in this example, the tessellation of the one neighboring Bézier element did not seriously affect the quality of the numerical solution.

6. Computational Issues

In Section 5 we have demonstrated that once the solution of MODEL-4 is completed, we have a sufficiently accurate numerical solution across the entire domain, and one which can be used for a posteriori error estimation. Afterwards, the initial T-mesh may be properly refined following well-known (or even possible novel) procedures, and thus a second C^{p-1}-continuous IGA may follow, and so on.

In this pilot study, the approach we followed was as follows. We completed the IGA solution (MODEL-2) and then continued with the computation of the Bézier elements (MODEL-3). At the end, we performed MODEL-4. In other words, in this study each model was solved *sequentially* and separately.

However, it is possible for MODEL-2 [with element matrices $(\mathbf{K}_{P,e}, \mathbf{M}_{P,e})$ associated to n_P DOFs] and MODEL-3 [with element matrices $(\mathbf{K}_{Q,e}, \mathbf{M}_{Q,e})$ associated to n_Q DOFs] to be integrated into the same loop, at least in two alternative ways, as follows.

- The first way is to take into consideration that spline elements and Bézier elements share the *same* Gauss points and have the *same* Jacobian matrix at them (because the shape of the patch does not change after knot insertion), and thus the computer effort for the Bézier elements may be substantially reduced, if properly programmed.
- The second way is to consider Equation (5), in which the Bézier extraction operator $\mathbf{C}_e$ is the *transformation* matrix between the two sets of basis functions (spline elements denoted by the subscript 'P', and Bézier elements denoted by 'Q'). Thus, the element matrices of each Bézier element are directly calculated in terms of the associated NURBS element using the following algebraic quadratic form:

$$\mathbf{K}_{Q,e} = (\mathbf{C}_e^{-1})\mathbf{K}_{P,e}(\mathbf{C}_e^{-1})^{\mathrm{T}} \text{and } \mathbf{M}_{Q,e} = (\mathbf{C}_e^{-1})\mathbf{M}_{P,e}(\mathbf{C}_e^{-1})^{\mathrm{T}}, \tag{33}$$

without performing any numerical integration at all (see, [13]). At the end of each element loop, we will receive two sets of matrices, the former $(\mathbf{K}_{P,e}, \mathbf{M}_{P,e})$ and the latter $(\mathbf{K}_{Q,e}, \mathbf{M}_{Q,e})$, both of standard size, which will eventually occupy different positions in their corresponding total matrices $(\mathbf{K}_{\text{total}}, \mathbf{M}_{\text{total}})_P$ and $(\mathbf{K}_{\text{total}}, \mathbf{M}_{\text{total}})_Q$, according to the associated connectivity (topology) arrays, $(\text{IEN})_\text{P}$ and $(\text{IEN})_\text{Q}$, respectively.

Again, although the spline elements in a NURBS patch and the assembly of the associated Bézier elements are produced by the *same* number of *non-zero* matrix elements (k_{ij}, m_{ij}) according to Equation (33), it is their final location in the system matrices $(\mathbf{K}_{\text{total}}, \mathbf{M}_{\text{total}})_Q$ which determines a larger model for the Bézier elements ($n_Q > n_P$). In this sense, the utilization of the assembly of associated Bézier elements should not be confused with any arbitrary higher-order FEM, because this assembly is within the IGA context, having a particular quality due to the similarities of the Jacobian matrix with IGA across the entire domain.

To remove any doubt, it is instructive to discuss the NURBS model (36 control points) and its associated Bézier model (100 control points), which are shown in Figure 2a,b, respectively. Since the problem under study is potential (Laplace equation, wave equation, etc.), we have one degree of freedom per control point. The NURBS model consists of nine spline elements, each with a size of 16×16, and thus in principle it utilizes $9 \times (16 \times 16) = 2304$ non-zero matrix elements, which eventually build a total stiffness matrix $(\mathbf{K}_{\text{total}})_P$ of size 36×36 including 900 non-zero matrix elements $(k_{ij}, m_{ij})_P$ with bandwidth 21 (Figure 22a). On the other hand, the associated assembly includes nine Bézier elements, which again utilize in principle 2304 non-zero matrix elements according

to Equation (33), and eventually build a total stiffness matrix $(\mathbf{K}_{\text{total}})_Q$ of size 100 × 100 with 2116 non-zero matrix elements $(k_{ij}, m_{ij})_Q$ and bandwidth 33 (Figure 22b).

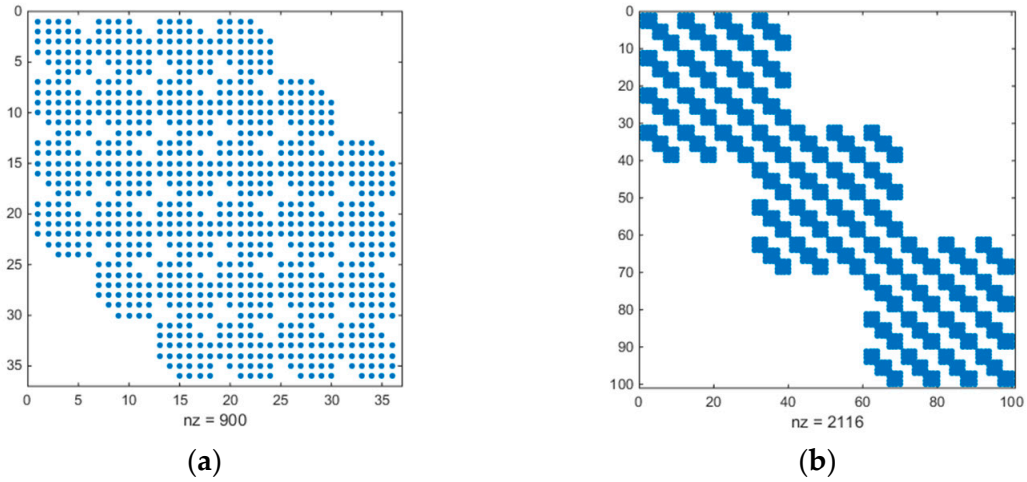


Figure 22. Sparsity patterns of total matrices: (**a**) NURBS (MODEL-2), (**b**) Bézier elements (MODEL-3).

In other words, the IGA element matrices (MODEL-2) are transformed (according to Equation (33)) and are eventually located in a matrix larger than NURBS, but less populated. This is the central point of the present paper.

Moving from the abovementioned NURBS to *T-splines*, the two alternative computational ways are again valid, but the second one may conditionally need some modifications. Therefore, since the BEXT $\mathbf{C}_e$ is *not* always a square matrix (e.g., in the present paper we noticed a few elements of size 17 × 17, whereas most of them were of the standard size 16 × 16), we cannot always construct the inverse matrix $(\mathbf{C}_e^{-1})$ as was the case in Equation (33). For these elements (larger than 16 × 16), after trivial matrix manipulation, one may easily validate that the matrices of the Bézier elements are slightly different, given by:

$$\mathbf{K}_{Q,e} = (\mathbf{C}_e^{\mathrm{T}}\mathbf{C}_e)^{-1}\mathbf{C}_e^{\mathrm{T}} \cdot \mathbf{K}_{P,e} \cdot \mathbf{C}_e(\mathbf{C}_e^{\mathrm{T}}\mathbf{C}_e)^{-1} \text{ and } \mathbf{M}_{Q,e} = (\mathbf{C}_e^{\mathrm{T}}\mathbf{C}_e)^{-1}\mathbf{C}_e^{\mathrm{T}} \cdot \mathbf{M}_{P,e} \cdot \mathbf{C}_e(\mathbf{C}_e^{\mathrm{T}}\mathbf{C}_e)^{-1}. \quad (34)$$

Equations (33) and (34) have been carefully tested, and only minor truncation errors were noticed between the several alternatives.

7. Discussion

It is well known to people working on IGA (isogeometric analysis) that knot insertion preserves the number of elements as well as the shape and the parameterization of the patch. To achieve Bézier elements of NURBS with C^0-continuity, it is adequate to insert repeated interior knots to the global knot vectors in both directions until the multiplicity equals the polynomial degree, p. This will increase the degrees of freedom, even though the element number is not changed. Since the solution space is expanded, the accuracy will be definitely improved, because the resulting solution space is a superset of the original solution space. The advantage of using the assembly of the associated Bézier elements, compared with the classical higher-order FEM, is that the shape and the parameterization of the patch, and thus the *Jacobian* matrix, are preserved, which means a mesh *quality* in the aforementioned assembly which is very consistent with IGA [13,14].

Beyond the state of the art, the present paper showed that each member in the assembly of the associated Bézier elements is merely a quadratic form (or similar, according to Equation (33) or (34)) of the same stiffness (and mass) element matrices used in continuous IGA, and thus it is the other side of the same coin. It was made quite clear that, although both formulations (MODEL-2 and MODEL-3) start with the same raw material

[i.e., both with the element matrices $(\mathbf{K}_{P,e}, \mathbf{M}_{P,e})$], after the implementation of Equation (33) or Equation (34) to element matrices of sizes 16 × 16 or 17 × 17, respectively, the produced element matrices $(\mathbf{K}_{Q,e}, \mathbf{M}_{Q,e})$ of MODEL-3 (always of size 16 × 16) are merely stored in total (system) matrices larger than MODEL-2, according to the corresponding connectivity arrays, $(\text{IEN})_{\text{P}}$ or $(\text{IEN})_{\text{Q}}$.

At this point, it should become quite clear that the utilization of the abovementioned Bézier elements (MODEL-3 and MODEL-4) is not competitive with respect to the IGA (MODEL-2) but should be considered only as a *supplementary* step in the determination of a reasonable accurate reference value for a posteriori error estimation. Therefore, in no way do we imply the replacement of the IGA (C^{p-1}-continuity) with Bézier elements (C^0-continuity), because then we lose the CAD-based information; it is only as a step to determine the refined NURBS or T-mesh for the next continuous IGA model in an adaptive sequence. Interestingly, although the assembly of the Bézier elements (MODEL-4, with n_Q DOFs) is generally more accurate than the IGA solution (MODEL-2, with n_P DOFs), if we seek to determine the size of continuous IGA model which has the same accuracy as MODEL-4, it will be found to have more spline elements but fewer control points than MODEL-4 (i.e., $n'_P < n_Q$). In other words, ongoing research shows that it is preferrable to update the C^{p-1}-continuous spline model, rather than increase the number of Bézier elements.

The background of the present paper is as follows. In the context of T-splines IGA, basis functions are usually calculated through the Bézier extraction operator $\mathbf{C}_e$, which is derived for each T-spline element. The merits of using Bézier extraction in IGA over the standard IGA have been reported in [15] (p. 75). To derive this operator, knot insertion is implicitly applied to all the knots of the model until the multiplicity equals the polynomial degree; the outcome of this procedure was documented many years ago in the form of a pseudocode [4]. Since the usual IGA is implemented using the Bézier operator anyway, $\mathbf{C}_e$ can be further used to easily calculate the new control points that form the rational Bézier elements of C^0-continuity; a similar idea was recently implemented for the solution of topology optimization problems [16]. In other words, the control points which are implicitly involved in the already-computed Bézier extraction operator matrix can serve as a vehicle to build rational Bézier elements of C^0-continuity. This is implemented with just a multiplication of the extraction matrix $\mathbf{C}_e$ by the vector of control points of the associated e-th element, but special care is required to calculate the unique control points. Interestingly, Evans et al. [17] extended Bézier extraction to hierarchical analysis-suitable T-splines (HASTS), which are utilized as a basis for adaptive IGA.

As has been previously reported by others, "there is no one-to-one correspondence between the T-spline elements and the Bézier elements if there is T-junction" [15] (p. 78), and this issue has been handled in MODEL-4 throughout the present paper.

To make the whole procedure clear, let us describe the workflow. Designers often use commercial software such as Fusion360$^{\text{TM}}$ (by Autodesk), Rhinoceros 3D$^{\text{TM}}$ (developed by TLM, Inc., Seattle, DC, USA), and similar to develop mechanical parts. Using such programs, freely producing new geometries and finishing the procedure, we reach the point where we must perform the Analysis (IGA). Due to the design procedure, T-junctions and extraordinary points very commonly appear in the model. As a result, there is a need to check and convert (if necessary) the T-spline design from the form of "None Analysis Suitable" T-spline (N-AST) to the preferable "Analysis Suitable" T-spline (AST). For detailed explanations, the reader may refer to [18]. Clearly, AST means that there are no sub-patches of reduced continuity, and this determines a slightly higher total number of T-spline elements, compared to those existing in the initial N-AST model; this happens

if either the usual IGA (MODEL-1 and MODEL-2) or the procedure relating to Bézier elements (MODEL-3 and MODEL-4) is implemented.

To overcome the above drawback, engineers use subroutines to convert any (N-AST) to (AST). Even then, geometry may include T-junctions, but then the procedure may integrate to the next phase, which is the Analysis module, without obstacles. In the proposed methodology, an AST scheme is always confirmed to present the new approach.

Although the generation of Bézier elements is a straightforward procedure in NURBS approximation [13], T-splines suffer from minor shortcomings, close to the (known in advance) borders of sub-patches with reduced continuity, i.e., those involved in the transformation from N-AST to AST. Therefore, it becomes conditionally necessary also to subdivide the closest Bézier element into the previously subdivided elements of reduced continuity. For example, one exception to this rule is when the extension reaches the boundary layer (and thus no additional Bézier element can be formed within the domain in the direction of the extension), as shown in Figure 13, in which only one element needs to be tessellated (as eventually shown in Figure 14), despite there being two elements of reduced continuity.

It was found that the abovementioned tessellation is not always necessary, but strongly depends on the boundary conditions. In Example 1, which was concerned with vertical uniaxial heat flow, no tessellation was necessary for either of the two discontinuous Bézier elements, because the line of discontinuity did not influence the flow (it was parallel to it).

In contrast, in Example 2, with the same T-mesh, but with BCs leading to horizontal flux, the lines of discontinuity were perpendicular to the flow and thus influenced it to a large degree (8.4%). It is worth mentioning that when only the Bézier element to the top-left part of the domain was tessellated, the average error decreased (from 8.4%) to only 7%, and this can be attributed to the low average temperature within it. Eventually, when both discontinuous Bézier elements were subdivided, the overall error completely vanished.

The tessellation of the neighboring Bézier elements close to them (which constitutes MODEL-4), is a standard technique which was developed in the context of the present paper. After the tessellation, the neighboring rational Bézier elements are not only C^0- but also G^0-continuous, which means that they share the same control points. Although the results are found to be very encouraging, a shortcoming of this approach is the need for tessellation in the entire row of elements until the boundary of the entire patch is reached. To resolve this drawback, instead of the tessellation of the discontinuous Bézier element and those in the same row, a single rational transient element (i.e., the one closest to the line of discontinuity) may be constructed [19,20].

In Example 3 regarding the annulus, the average error was small because, due to the small ratio of the radii, specifically, $R_1/R_2 = 1/2$, the logarithmic distribution of the temperature was almost linear in terms of the radius r. In this example, Table 2 clearly shows that MODEL-3 has an error 1.7 times larger than that in MODEL-1 and MODEL-2, whereas MODEL-4 is the most accurate among the four models (with an average error being 3.6 times smaller than that in MODEL-1 and MODEL-2). Note that all of them refer to the same number of $n_{ele} = 24$ T-spline elements; however, MODEL-3 deals with 24 Bézier elements (the same as the T-spline elements) whereas MODEL-4 deals with 25. For the avoidance of any doubt regarding the relative accuracy of the models due to round-off errors of coordinates and weights, tensor-product (NURBS) analysis was added to elucidate this example (see, Table 3), because standard coordinates and standard weights may be applied in its reproduction by any researcher.

In Example 4, concerned with the thermal analysis of a rectangular domain under fully two-dimensional boundary conditions, the quality of the results was like those of Example 3, but the value of the initial error was larger than what it was in Example 3 (in MODEL-1 and MODEL-2: about 2.7%). In more detail, MODEL-3 showed an error 1.8 times greater than that of MODEL-1 and MODEL-2, whereas MODEL-4 outperformed with an error about 500 times smaller than that of MODEL-1 and MODEL-2. Obviously, this finding is in favor of our claim that MODEL-4 can be used for a posteriori error estimation.

In Example 5, concerned with the eigenvalue–eigenvector analysis of a rectangular acoustic cavity, impressive results had been previously received for NURBS-based IGA [14] for different dimensions of the cavity (aspect ratio 2.5:1.1 $\cong$ 2.27 in [14], compared to 3.5:1.5 $\cong$ 2.33 of the present paper). Quite similarly, equally impressive results were obtained in T-spline-based IGA as well.

Overall, the superior accuracy of MODEL-4 dictates that it can take the role of a reference value for a posteriori error estimation.

In principle, the proposed methodology is applicable to the entire spectrum of computational mechanics, including elasticity, where, for the most part, the research is focused [21,22].

8. Conclusions

Bézier extraction matrices, the use of which is the golden standard in isogeometric analysis (IGA), may be used to construct updated unique control points which further determine a set of rational Bézier elements with C^0-continuity characterized by more degrees of freedom than found in the standard IGA model. This in turn leads to a second numerical solution of higher accuracy, practically without affecting the number of T-spline elements, and thus useful for a posteriori error estimation. It was found that in most cases, the blind formation of the Bézier elements using the Bézier extraction operator matrix of the patch is problematic, and thus it is necessary to further subdivide a small number of Bézier elements which share the same interface with T-spline elements of reduced continuity. Although in the present paper a standard tessellation of Bézier elements was applied, transient rational elements are also applicable. The applied methodology is generally applicable to the whole spectrum of computational mechanics as well as to three-dimensional problems.

Author Contributions: Conceptualization, C.P.; methodology, C.P. and I.D.; software, C.P. and I.D.; validation, C.P. and I.D.; writing—original draft preparation, C.P.; writing—review and editing, C.P.; visualization, I.D.; supervision, C.P.; project administration, C.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Determination of Common Points in Neighboring Patches

In Appendix A we describe the procedure of the subdivision of a Bézier patch. Since the three patches shown in Figure 3c have been produced by the procedure of Bézier extraction, which preserves the shape of the patch, it is obvious that they share a unique common point H. It is apparent that the point H belongs to the position $(\xi = 0, \eta = 1)$ of the element (I-a), and to the position $(\xi = 0, \eta = 0)$ of the element (I-b); as well, as it lies along the edge AD (with $\xi_H = 1$) of the larger element ABED. Since the Cartesian coordinates (x_H, y_H) of the point H are known because it coincides with a corner control

point in the two small sub-patches, the determination of the parameter η_H along the edge AD is merely to fulfil the following condition:

$$\sum_{i=1}^{n} B_{i,3}(\eta) y_i = y_H, \tag{A1}$$

where $B_{i,3}(\eta)$ are the Bernstein basis polynomials of degree $p = 3$:

$$B_{0,3}(\eta) = (1-\eta)^3,\ B_{1,3}(\eta) = 3(1-\eta)^2\eta,\ B_{2,3}(\eta) = 3(1-\eta)\eta^2,\ B_{3,3}(\eta) = \eta^3, \tag{A2}$$

while y_i is the corresponding coordinate of the i-th control point along the edge AD.

Substituting Equation (A2) into Equation (A1), we receive the following cubic equation:

$$a\eta^3 + b\eta^2 + c\eta + d = 0, \tag{A3}$$

with

$$a = (3y_1 - y_0 - 3y_2 + y_3),\ \ b = (3y_0 - 6y_1 + 3y_2),\ \ c = (3y_1 - 3y_0),\ \ d = y_0. \tag{A4}$$

The finding of the unique real root $\eta_H \in (0, 1)$ of the cubic polynomial in Equation (A3) is a trivial task of numerical analysis, which may be facilitated using a mathematical package such as MATLAB®, and using the following commands: coefficients = [a, b, c, d]; root_values = roots (coefficients); in conjunction with the condition find (root_values < 1 and root_values > 0).

Appendix B. Control Points After the Subdivision

Suppressing the first subscript '3', which indicates the isoline $\xi = 1$, let the control points along the common edge BE (see Figure 3c) be P_0, P_1, P_2, P_3, as shown in Figure A1. Having determined the parameter η_H of the common point H, of known Cartesian coordinates (x_H, y_H) implementing Appendix A, we set it as a new control point, and continue with the subdivision of the Bézier patch ABED by the line HH'. Then, we must determine the three control points $(Q_0, Q_1, Q_2, Q_3 = H)$ on the left of H, and the other three control points $(H = Q_3, Q_4, Q_5, Q_6)$ on the right of it. Based on the same subdivision ratio η_H, it is easy to show that these seven updated projected control points, useful for the two subdivided Bézier elements, are given by the following:

$$\begin{aligned}
Q_0^w &= P_0^w,\\
Q_1^w &= (1-\eta_H)P_0^w + \eta_H P_1^w,\\
Q_2^w &= (1-\eta_H)^2 P_0^w + 2(1-\eta_H)\eta_H P_1^w + \eta_H^2 P_2^w,\\
Q_3^w &= (1-\eta_H)^3 P_0^w + 3(1-\eta_H)^2\eta_H P_1^w + 3(1-\eta_H)\eta_H^2 P_2^w + \eta_H^2 P_3^w,\\
Q_4^w &= (1-\eta_H)^2 P_1^w + 2(1-\eta_H)\eta_H P_2^w + \eta_H^2 P_3^w,\\
Q_5^w &= (1-\eta_H)P_2^w + \eta_H P_3^w,\\
Q_6^w &= P_3^w.
\end{aligned} \tag{A5}$$

The above procedure is repeated with respect to the next three polygon lines (P0, i, P1, i, and P2, i, with i = 0, 1, 2) shown in Figure A1.

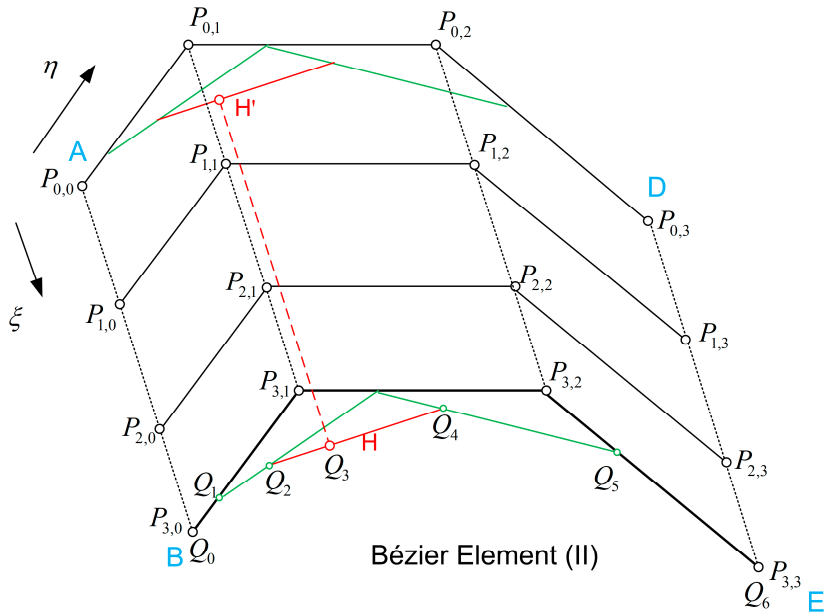


Figure A1. Element to be tessellated.

References

1. Hughes, T.J.R.; Cottrell, J.A.; Bazilevs, Y. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Comput. Methods Appl. Mech. Eng.* **2005**, *194*, 4135–4195. [CrossRef]
2. Bazilevs, Y.; Calo, V.; Cottrell, J.; Evans, J.; Hughes, T.J.R.; Lipton, S.; Scott, M.; Sederberg, T.W. Isogeometric analysis using T-splines. *Comput. Methods Appl. Mech. Eng.* **2010**, *199*, 229–263. [CrossRef]
3. Borden, M.J.; Scott, M.A.; Evans, J.A.; Hughes, T.J.R. Isogeometric finite element data structures based on Bézier extraction of NURBS. *Int. J. Numer. Methods Eng.* **2011**, *87*, 15–47. [CrossRef]
4. Scott, M.A.; Borden, M.J.; Verhoosel, C.V.; Sederberg, T.W.; Hughes, T.J.R. Isogeometric finite element data structures based on Bézier extraction of T-splines. *Int. J. Numer. Methods Eng.* **2011**, *88*, 126–156. [CrossRef]
5. Cox, M.G. The numerical evaluation of B-splines. *J. Inst. Math. Its Appl.* **1972**, *10*, 134–149. [CrossRef]
6. De Boor, C. On calculating with B-splines. *J. Approx. Theory* **1972**, *6*, 50–62. [CrossRef]
7. Piegl, L.; Tiller, W. *The NURBS Book*; Springer: Berlin, Germany, 1997.
8. Wang, Y.; Zhang, J. *Control Point Removal Algorithm for T-Spline Surfaces*; Kim, M.-S., Shimada, K., Eds.; GMP 2006, LNCS 4077; Springer: Berlin/Heidelberg, Germany, 2006; pp. 385–396. [CrossRef]
9. Wang, A.; Li, L.; Wang, W.; Du, X.; Xiao, F.; Cai, Z.; Zhao, G. Linear Independence of T-Spline Blending Functions of Degree One for Isogeometric Analysis. *Mathematics* **2021**, *9*, 1346. [CrossRef]
10. Lyche, T.; Mørken, K. Making the Oslo Algorithm More Efficient. *SIAM J. Numer. Anal.* **1986**, *23*, 663–675. [CrossRef]
11. Scott, M.A. T-Splines as a Design-Through-Analysis Technology, Dissertation. Ph.D. Thesis, The University of Texas at Austin, Austin, TX, USA, 2011.
12. Courant, R.; Hilbert, D. *Methods of Mathematical Physics, 1st English ed.*; InterScience: New York, NY, USA, 1966; Volume I, pp. 300–304, (translated and revised from the German original).
13. Provatidis, C.G. Comparison between Bézier extraction and associated Bézier elements in eigenvalue problems. *WSEAS Trans. Syst. Control* **2022**, *17*, 605. [CrossRef]
14. Lai, W.; Yu, T.; Bui, T.Q.; Wang, Z.; Curiel-Sosa, J.L.; Das, R.; Hirose, S. 3-D elasto-plastic large deformations: IGA simulation by Bézier extraction of NURBS. *Adv. Eng. Softw.* **2017**, *108*, 68–82. [CrossRef]
15. Singh, S.K.; Singh, I.V.; Mishra, B.K.; Bhardwaj, G.; Bui, T.Q. A simple, efficient and accurate Bézier extraction based T-spline XIGA for crack simulations. *Theor. Appl. Fract. Mech.* **2017**, *88*, 74–96. [CrossRef]
16. Zhang, X.; Xiao, M.; Gao, L.; Gao, J. A T-splines-oriented isogeometric topology optimization for plate and shell structures with arbitrary geometries using Bézier extraction. *Comput. Methods Appl. Mech. Eng.* **2024**, *425*, 116929. [CrossRef]
17. Evans, E.J.; Scott, M.A.; Li, X.; Thomas, D.C. Hierarchical T-splines: Analysis suitability, Bézier extraction, and application as an adaptive basis for isogeometric analysis. *Comput. Methods Appl. Mech. Eng.* **2015**, *284*, 1–20. [CrossRef]
18. Habib, S.H.; Kezrane, C.; Hachi, B.E. Moving local mesh based on analysis-suitable T-splines and Bézier extraction for extended isogeometric finite element analysis—Application to two-dimensional crack propagation. *Finite Elem. Anal. Des.* **2023**, *213*, 103854. [CrossRef]

19. Eisenträger, S.; Eisenträger, J.; Gravenkamp, H.; Provatidis, C. High order transition elements: The xNy-element concept, Part II: Dynamics. *Comput. Methods Appl. Mech. Eng.* **2021**, *387*, 114145. [CrossRef]
20. Provatidis, C.G. Non-Rational and rational transfinite interpolation using Bernstein polynomials. *Int. J. Comput. Geom. Appl.* **2022**, *32*, 55–89. [CrossRef]
21. Guo, M.; Wang, W.; Zhao, G.; Du, X.; Zang, R.; Yang, J. T-Splines for Isogeometric Analysis of the Large Deformation of Elastoplastic Kirchhoff–Love Shells. *Appl. Sci.* **2023**, *13*, 1709. [CrossRef]
22. Fakkoussi, S.E.L.; Koubaiti, Q.; Elkhalfi, A.; Vlase, S.; Marin, M. Numerical analysis of the cylindrical shell pipe with preformed holes subjected to a compressive load using non-uniform rational B-splines and T-splines for an isogeometric analysis approach. *Axioms* **2024**, *13*, 529. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.