Comparison between Bézier Extraction and Associated Bézier Elements in Eigenvalue Problems

CHRISTOPHER G. PROVATIDIS
School of Mechanical Engineering,
National Technical University of Athens,
9, Iroon Polytechniou str, 15780 Zografou,
GREECE

Abstract: - This paper shows that the control points which are implicitly encountered in the Bézier extraction during isogeometric analysis can be explicitly used to form Bézier elements of C^0-continuity in several ways, thus eventually leading to a superior accuracy and performance than the C^p-continuity. The study is reduced to the eigenvalue extraction in problems governed by the Helmholtz equation. Analysis is performed in conjunction with piecewise cubic interpolation for three benchmark tests in one and two dimensions. In the latter case a rectangular and a circular acoustical cavity under Neumann boundary conditions are analyzed. Several computational details are also discussed.

Key-Words: - Bézier extraction; isogeometric analysis; NURBS; finite elements; Helmholtz equation

Received: March 28, 2022. Revised: November 24, 2022. Accepted: December 21, 2022. Published: December 31, 2022.

1 Introduction

Within the context of the finite element method (FEM), the idea of using the same set of basis functions for the geometry and the analysis is very old. The practical need of this concept is the treatment of curvilinear domains where the Jacobian matrix has to be found at the integration points aiming at estimating the stiffness (static problems) as well as the mass matrix (time-dependent problems). The first report referring to the so-called 'isoparametric element' is due to Taig, [1]. A later public paper is due to Irons, [2], while a detailed description may be found in classical FEM textbooks (see [3], [4]).

The above idea, i.e., the use of the same functional set for both the geometric model and the analysis, was later extended using the Coons interpolation method (see [5], [6]). The difference of the latter with the previous standard FEM methods is that, instead of using a known functional set such as a power series expansion or a sinusoidal series, global shape functions are extracted from a computer-aided geometric design (CAGD) interpolation formula. These shape functions span an entire patch of the domain which, in general, may be also decomposed into a number of smaller curvilinear patches.

As has been documented in classical CAGD textbooks such as, [7], the abovementioned Coons interpolation formula, due to Steven Coons (1964-1967), is almost the first one which was ever developed in the theory of CAGD. Moreover, in chronological order, the second important formula was due to William Gordon (1969), a third interpolation was due to Pierre Bézier (1970), a fourth interpolation was the 'B-spline' due to Curry-Schoenberg formulation (1966) that was later treated more efficiently thanks to an efficient recursive scheme by Cox-deBoor-Mansfield (1972), the fifth was NURBS due to Versprille (1975) but was popularized twenty years later mainly by the monograph of Piegl and Tiller containing pseudo-codes, [8]. The sixth cornerstone CAGD-interpolation is T-splines, [9], while it is anticipated that novel CAGD-based interpolations will follow to keep busy the next generations of FEM-researchers. The author has not included a lot of other significant interpolations (e.g. alphabetically: Bajaj-, Barnhill-, DeRose-, Gregory-patches, etc.) because he had to restrict himself to those advertised in the most known commercial CAD/CAM/CAE software packages. The research on splines, as a means to solve partial differential equations (PDEs) with given boundary conditions, still continues to this date ([10], [11], [13]).

Based on transfinite Coons-Gordon interpolation formulas a number of papers have appeared for either static or dynamic analysis ([14] and [15]), while [16] is a monograph including the summary of

almost sixty papers (for the period 1989 to 2017) which was published much later in 2019.

Based on B-spline interpolation a number of papers on the numerical solution of PDEs have appeared near 2000 (see [17], [18], [19]), while a relevant monograph is [20].

Regarding the application of non-uniform rational B-splines (NURBS) in engineering analysis, the first relevant studies were reported in 1993 and 1994 (see [21] and [22]). NURBS were revived in 2005 under the title 'NURBS-based isogeometric analysis' (IGA). This was done mainly for product shape design, [23]. In the same year (2005), [24], an extensive paper dealt with a broader spectrum of engineering applications for which all the well-known flexibilities and advantages of previously developed CAGD algorithms such as knot insertion, degree elevation, etc were utilized. A relevant textbook, published in 2009, is [25].

In its original implementation, IGA requires substantial computer time in order to calculate the basis functions at the integration points thus to estimate the stiffness matrix, despite the local support of the basis and the fact that the efficient de Boor-Cox recursive algorithm was used (see [26], [27]). To reduce the computer effort, the Bézier extraction technique has been proposed, [28]. In brief, the main advantage is that Bézier extraction, compared to the original implementation of IGA, [25], is such that apart from the coefficients of the extraction operator (matrix $\mathbf{C}_e$) of the 'e-th' NURBS-element within a patch, the basis functions are identical for all elements in the mesh as it is the case for classical finite elements. Therefore, there is no need to implement B-spline basis function evaluation routines which (as previously said) are costly from a numerical point of view.

Within the context of (tensor-product) NURBS approximation of degree p (in conjunction with control points $\mathbf{P}$), after the computation of the extraction operator $\mathbf{C}_e$ (for definitions, see [28]) the updated set of control points $\mathbf{Q}$ (associated Bernstein-Bézier polynomials) can be easily calculated by the linear formula $\mathbf{Q} = \left(\mathbf{C}_e\right)^{\mathrm{T}} \mathbf{P}$, where the superscript '$^{\mathrm{T}}$' stands for the transpose of the matrix $\mathbf{C}_e$. As a result, if the aforementioned control points $\mathbf{Q}$ are properly grouped, they can build a set of Bézier elements of degree p with C^0 inter-element-continuity.

In this paper it will be shown that the above-mentioned control points $\mathbf{Q}$, which again are *implicitly* encountered in the abovementioned Bézier extraction (MODEL-1), can be explicitly used to form Bézier elements of C^0-continuity (henceforth MODEL-2) with superior accuracy.

2 The Main Idea

The numerical method of 'Bézier extraction' is based on the insertion of additional knots at all the inner knots of the patch until their multiplicity becomes equal to the polynomial degree p. Then a linear relationship is derived between the NURBS basis functions $\mathbf{N}$ and the Bernstein polynomials $\mathbf{B}$ according to the expression:

$$\mathbf{N} = \mathbf{C}_e \mathbf{B} \tag{1}$$

where $\mathbf{C}_e$ is the well known *Bézier operator*, [28]. Since after knot insertion the shape of the patch remains unaltered, in form and parametrically, the patch $\mathbf{x}(\xi,\eta)$ is described by the condition:

$$\mathbf{x}(\xi,\eta) = \mathbf{N}^{\mathrm{T}}\mathbf{P} = \mathbf{B}^{\mathrm{T}}\mathbf{Q}\,, \tag{2}$$

where $\mathbf{Q}$ denotes the control points after the abovementioned multiple knot insertion (those involved in Bézier extraction). Substitution of Eq. (1) into Eq. (2) and further drop out of the common factor $\mathbf{B}^{\mathrm{T}}$ on the left of the second equality, leads to

$$\left(\mathbf{C}_e\right)^{\mathrm{T}} \mathbf{P} = \mathbf{Q}\,, \tag{3}$$

or equivalently:

$$\mathbf{P} = \left(\mathbf{C}_e^{-1}\right)^{\mathrm{T}} \mathbf{Q} \tag{4}$$

In one-dimensional (1D)-problems, and for every NURBS-element 'e', $\mathbf{P}$ is of a column-vector of size $(p+1)$, $\mathbf{Q}$ is of a column-vector of size $(p+1)$, whereas the matrix $\mathbf{C}_e$ is a matrix of size $(p+1)\times(p+1)$.

Implementing the concept of isogeometric analysis (i.e., same basis functions for the analysis as well), a similar relationship will hold in both systems for the degrees of freedom (DOFs), $\mathbf{a}$, as well, thus the analogue of Eq. (4)will be:

$$\mathbf{a}_P = \left(\mathbf{C}_e^{-1}\right)^{\mathrm{T}} \mathbf{a}_Q \tag{5}$$

Regarding MODEL-1 (original IGA), for any NURBS element 'e' let the corresponding stiffness and mass matrices be $\mathbf{K}_e$ and $\mathbf{M}_e$, respectively.

Below we shall derive the relevant matrices for MODEL-2.

Actually, Eq. (5)depicts a linear relationship between the 'local' degrees of freedom (DOFs), $\mathbf{a}_P$, of MODEL-1 (B-spline or NURBS) and the 'global' DOFs, $\mathbf{a}_Q$, of MODEL-2, through the transformation matrix $\mathbf{T}=\left(\mathbf{C}_e^{-1}\right)^{\mathrm{T}}$. This *change of basis* leads to the well-known quadratic form ($\mathbf{K}_{global} = \mathbf{T}^{\mathrm{T}}\ \mathbf{K}_{local}\ \mathbf{T}$) thus the matrices of MODEL-2 can be written in terms of the local matrices as follows:

$$\left(\mathbf{K}_e\right)_Q=\left(\mathbf{C}_e^{-1}\right)\left(\mathbf{K}_e\right)\left(\mathbf{C}_e^{-1}\right)^{\mathrm{T}} \tag{6}$$

and

$$\left(\mathbf{M}_e\right)_Q=\left(\mathbf{C}_e^{-1}\right)\left(\mathbf{M}_e\right)\left(\mathbf{C}_e^{-1}\right)^{\mathrm{T}}. \tag{7}$$

In addition to the general quadratic form of Eqs. (6) and (7) which requires only matrix operations, *three* more equivalent alternatives for MODEL-2 will be discussed below.

To make the procedure as clear as possible, we shall start with a typical 1D-problem and then continue with a couple of 2D-problems.

2.1 One-dimensional problem

Problem: Consider the Helmholtz equation $(\partial^2 u/\partial x^2)+k^2u=0$ with wave-number $k=\omega/c$, in the interval $[0,L]$ with $L=3$. The boundary condition at the left end is of Dirichlet type ($u=0$ at $x=0$) while at the right end is of Neumann type ($\partial u/\partial x=0$ at $x=L$). Find the eigenvalues implementing IGA for the polynomial degree $p=3$ and three uniform B-spline elements.

Solution: The exact eigenvalues $\lambda_i=\omega_i^2$ are given by:

$$\lambda_i=(2i-1)\pi^2c^2/(4L^2),\quad i=1,2,\ldots. \tag{8}$$

2.1.1 Original IGA (MODEL-1)

First, B-splines analysis (MODEL-1) is performed for the knot vector $\Xi=\{0,0,0,0,1,2,3,3,3,3\}$ (i.e, for $n=3$ uniform breakpoint spans), for polynomial degree $p=3$, which involves three B-spline elements and six control points as shown in Fig. 1(a). In general, the position of the control points may be arbitrary. However, if we allow a linear isoparametric mapping, i.e. that $x(\xi)=\xi$ with $(0\le\xi\le 3)$, then the aforementioned six involved control points are found to be located at the particular positions $x_{ctrl}=\{0,\frac{1}{3},1,2,\frac{8}{3},3\}$ as shown in Fig. 1(a). The aforementioned linear mapping is not obligatory but permits the accurate representation of ideally linear functions. Based on these six control points, for each element $'e'$ $(e=1,2,3)$ the Jacobian will be a constant, because the domain is a straight line. Wherever the control points are, the six basis functions are those illustrated in Fig. 2(a). Two of them span one-third, also two of them span two-thirds and the final two basis functions span the entire domain.

Elementary B-spline theory suggests that the connectivity vector which allocates the global DOFs to each of the abovementioned three B-spline elements, will be ***IEN-1*** = [(1,2,3,4), (2,3,4,5), (3,4,5,6)], so as only $p+1$ basis functions affect each element. For each of these three elements the local matrices are given by:

$$[\mathbf{K}]_{ij}^{e}=\int_0^{l_e}\frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x}dx,\ i=1,2,3,4 \tag{9}$$

and

$$[\mathbf{M}]_{ij}^{e}=\frac{1}{c^2}\int_0^{l_e}N_iN_j\,dx,\ i=1,2,3,4\,. \tag{10}$$

After matrix assembly, the boundary conditions are imposed (deletion of the first raw and column in both the mass and stiffness matrices due to the Dirichlet boundary condition), and the eigenvalues are calculated in terms of the system matrices, **K** and **M** (after deletion) requiring that the determinant vanishes:

$$\det\left\|\mathbf{K}-\lambda_i\mathbf{M}\right\|=0\,, \tag{11}$$

with $\lambda_i=\omega_i^2$ denoting the i-th eigenvalue of the problem.

Either the basis functions N_i involved in Eqs. (9) and (10) are computed in terms of usual B-splines (e.g., using a standard function such as the `spcol` in MATLAB) or in terms of Bernstein polynomials using the Bézier operator $\mathbf{C}_e$ [see, Eq. (1)], the obtained matrices are the same thus leading to the numerical results shown in Fig. 3 (blue line). One may observe that the computational error increases with the serial number of the eigenmode.

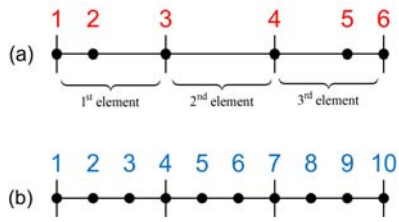

Fig. 1: Arrangement of control points in (a) initial B-spline and (b) after the multiple knot insertion.

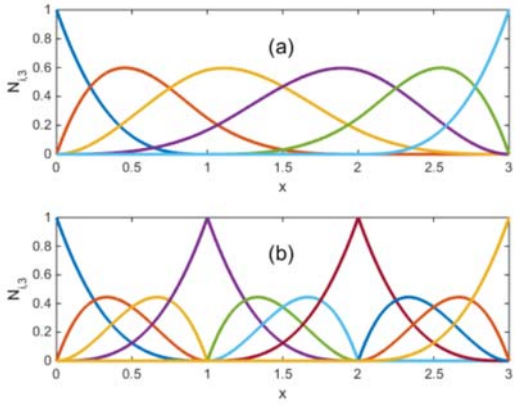

Fig. 2: Basis functions for (a) initial B-spline and (b) after the multiple knot insertion.

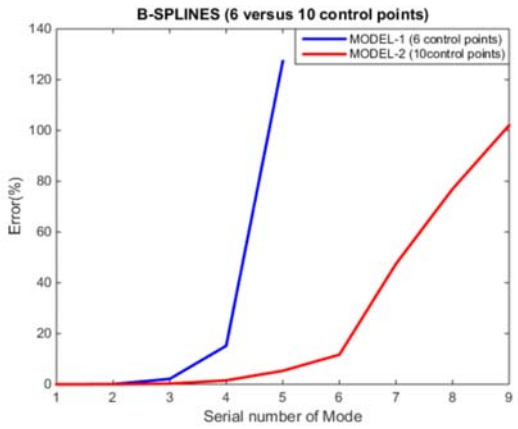

Fig. 3: Errors (in percent) of the calculated eigenvalues (Model-1: 6 and Model-2: 10 control points).

2.1.2 Bézier elements and similar (MODEL-2)

Increasing the multiplicity of the two inner knots (at $\xi = 1$ and $\xi = 2$) from '1' to '3' (note that $p = 3$), the new control points increase from 6 (**P**) to 10 (**Q**) and are uniformly arranged within the interval [0,3] as shown in Fig. 1(b) while the corresponding basis functions are shown in Fig. 2(b). In the latter figure, one may observe that the new basis functions are *interpolatory* at the ends of each element (which are also ends of repeated knots) and span two elements (one on the left and the other on the right of each multiple inner knot). The new connectivity vector will be: ***IEN-2*** = [(1,2,3,4), (4,5,6,7), (7,8,9,10)]. Again, only $p + 1$ basis functions affect each element.

At this point, four alternative procedures (all of them labeled as '*MODEL-2*') are tested as follows:

2.1.2.1 Procedure (2a): de Boor B-splines

The first procedure for Model-2 consists of practically using the same code as that of MODEL-1 but now setting the multiplicity of the (two) inner knots equal to $p = 3$. For the produced updated basis functions $N_i, i = 1, \ldots, 10$, we can use the standard recursive (de Boor) formula, such as `spcol` existing in MATLAB®. For each of the three B-spline elements we use four (i.e., $p + 1$) Gauss points and significant computer savings are achieved when the local support is encountered through the above-mentioned connectivity vector ***IEN-2***.

2.1.2.2 Procedure (2b): Bézier elements

The second procedure for Model-2 consists of using the three cubic (4-node) *Bézier* elements of equal size shown in Fig. 1(b) and Fig. 2(b). Each Bézier element is made of four DOFs associated to four Bernstein polynomials as basis functions. If one does not wish to perform numerical Gauss integration in conjunction with the aforementioned Bernstein polynomials, the analytical formulas of the matrices in each element are given in Appendix A. The connectivity vector will be again the aforementioned ***IEN-2***.

2.1.2.3 Procedure (2c): Lagrange elements

The third procedure for Model-2 consists of using the three classical cubic *Lagrange* elements of equal size which are shown in Fig. 1(b). The analytical formulas of the matrices in each element are given in Appendix B. The same connectivity vector, i.e., ***IEN-2*** will be applied.

2.1.2.4 Procedure (2d): Matrix transformation

The fourth and last alternative procedure for Model-2 consists of blindly implementing Eqs. (6) and (7) at each of the three B-spline elements shown in Fig. 1(a) that have been found in Model-1, and then rearranging the produced quadratic form in the form

of enlarged matrices according to the connectivity vector ***IEN-2***. From the computational point of view, we can use two loops as follows. The outer loop refers to the three elements while the inner loop refers to the $(p+1)$ Gauss points in each element. Between these two loops, the local matrices ($\mathbf{K}_e$, $\mathbf{M}_e$) are calculated and then $\left(\mathbf{K}_e\right)_Q$ as well as $\left(\mathbf{M}_e\right)_Q$ are found according to Eqs. (6) and (7).

It was found that three out of the four abovementioned procedures (i.e., the Procedures **2a**, **2b** and **2d**) lead to *identical* mass and stiffness matrices. Not only that, but although the Procedure **2c** gives different matrices than the other three procedures, it eventually leads to the same numerical results (calculated eigenvalues) with the Procedures (**2a**, **2b** and **2d**). Therefore, the error of the calculated eigenvalues for all these four procedures, represented under the umbrella of MODEL-2, is shown in Fig. 3 (red line).

2.1.3 Comparison of MODEL-1 with MODEL-2

One may observe that MODEL-1 [i.e., three B-spline elements (6DOFs, C^2-continuity)] is *less* accurate than MODEL-2 [i.e., a totality of three Bézier elements (10 DOFs, C^0-continuity)]. One reason for the superiority of the C^0-continuity (MODEL-2) is probably the larger number of the participating DOFs (10 versus 6 before the imposition of the BCs, and 9 versus 5 after the deletion of the restrained DOFs).

It is noted that in the above one-dimensional test case it is very difficult to compare the CPU-times because all of them are very small and the whole comparison is very sensitive.

2.2 Two-dimensional problems

2.2.1 Rectangular cavity

Consider an acoustic cavity of size $a \times b = 2.5m \times 1.1m$, with normalized wave speed $c = 1\,m/s$, and rigid walls ($\partial u/\partial n = 0$). For the governing Helmholtz equation $\nabla^2 u + k^2 u = 0$ (with $k = \omega/c$), calculate the lowest 15 eigenvalues $\lambda_i = \omega_i^2, i = 1,2,\cdots,15$. We recall that the exact eigenvalues are given by the formula:

$$\omega_{mn}^2 = \lambda_{mn} = \pi^2 c^2 \left[\left(\frac{m}{a}\right)^2 + \left(\frac{n}{b}\right)^2\right],\ m,n = 0,1,\ldots,\infty \quad \#(12)$$

Solution: The initial isogeometric model (*MODEL-1* of C^2-continuity) comprises 10 cubic B-spline elements in a 5×2 setup, which includes $8 \times 5 = 40$ control points (see, Fig. 4(a)). Therefore, each matrix of (**K**, **M**) comprises 40 rows and 40 columns, of which none is deleted when imposing the Neumann-type BCs. The results of this model are shown in Fig. 5: (blue line).

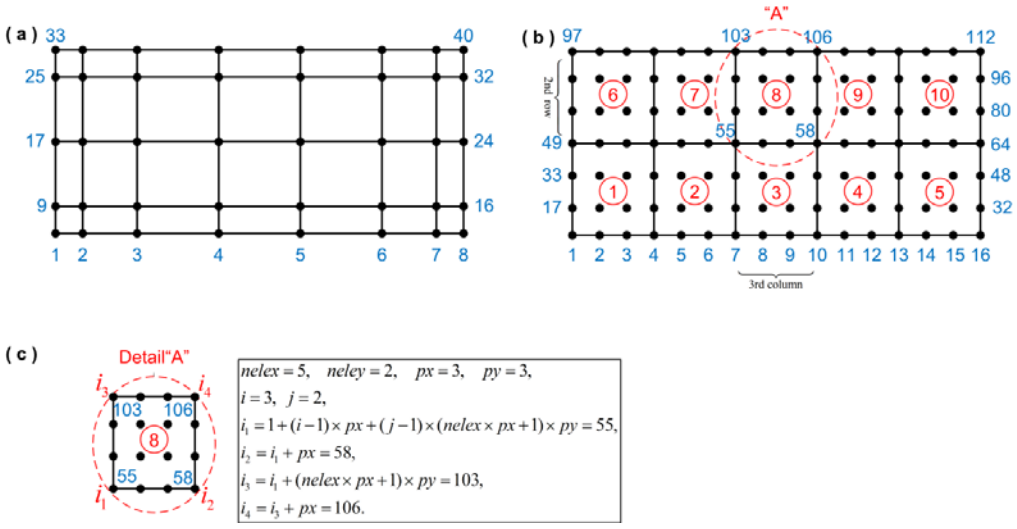

Fig. 4: Control points for (a) C^2- and (b) C^0-continuity, (c) detail of Bézier element No. 8.

After the Bézier decomposition, the totality of the new control points are rearranged according to Fig. 4(b), in which their number has increased (from 40) to $16 \times 7 = 112$. Now, each of the ten cubic Bézier elements consists of 16 control points arranged in a 4×4 setup. Here it happens that all the extreme control points belong to the true boundary of each element and they split each edge in three equal pieces, as if we had to deal with conventional rectangular finite elements. As previously happened with the 1D-problem, four equivalent procedures have been developed; three of them are coincident (i.e., the procedures **2a**, **2b**, and **2d**), one is equivalent (procedure **2c**), and all of them constitute '*MODEL-2*'. The ***IEN-2*** matrix can be automatically computed in advance, based on the formulas which are illustrated in Fig. 4(c), where the indices (i_1, i_2, i_3 and i_4) denote the serial numbers of the control points at the corners of each NURBS element. In this convention, the Bézier elements are numbered sequentially starting from the lower left and increasing until the bottom edge is completed; then we continue with the above-the-bottom layer from left to right, and so on (see, the numbers inside the small red circles of Fig. 3b). Again, Procedure-2*a* involves the MATLAB's function **`spcol`** in conjunction with multiplicity 3 (or its equivalent Bézier extraction), Procedure-2b includes the ten tensor product cubic Bézier elements, Procedure-2c includes ten conventional tensor product cubic Lagrange elements, whereas the algebraic Procedure-2d includes change of basis using Eqs. (6) and (7). In all these four alternative procedures (2a÷2d) for *Model-2* the results were found to be identical (see Fig. 5:, red line) and one may observe that they *outperform* the *Model-1*.

Although MODEL-2 (based on C^0-continuity) is superior to MODEL-1 (C^2-continuity), we have to admit that the former deals with 112 DOFs compared to 40 DOFs existing in the later model.

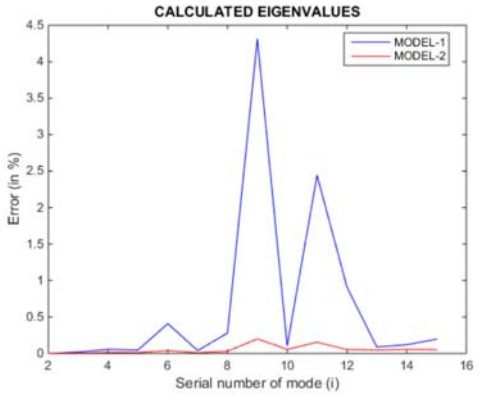

Fig. 5: Errors of the calculated eigenvalues for the rectangular cavity.

2.2.2 Circular cavity

A circular acoustic cavity of unit radius ($a=1$) under Neumann boundary conditions is studied. The analytical solution is given by

$$J'_m(ka) = 0, \quad m = 0,1,2, \qquad \#(13)$$

where $J'_m(ka)$ is the first derivative of the Bessel function $J_m(ka)$ of the first kind and order m and $k = \omega/c$ is the wavenumber. We wish to find the lowest seventeen eigenvalues applying IGA.

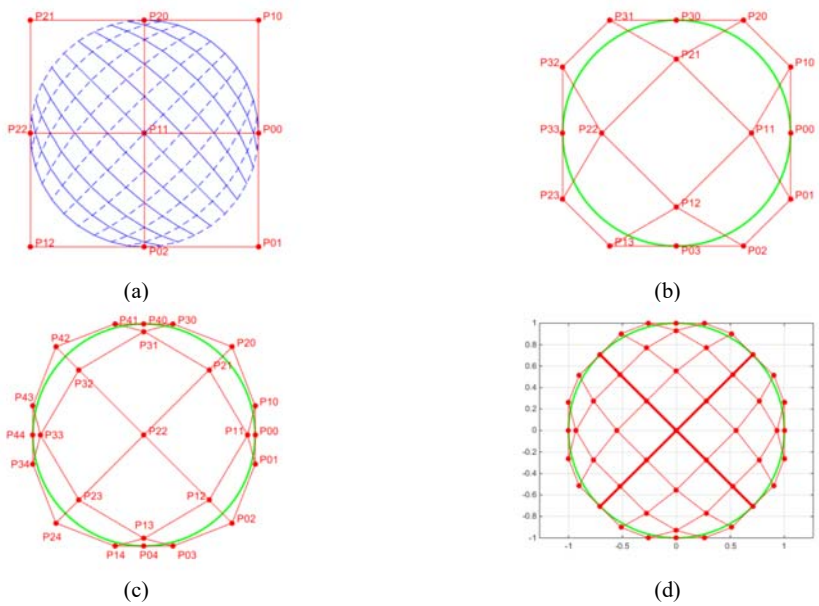

Fig. 6: Progressive knot insertion in a circular cavity (a) quadratic Bézier with isolines, (b) cubic Bézier, (c) cubic NURBS with C^2-continuity, (d) cubic NURBS with C^0-continuity.

Solution: The starting point is the well-known 9-point rational Bézier tensor product (a 3×3 setup, see Fig. 6a) for degree $p = 2$, knot vector $\Xi = \{0,0,0,1,1,1\} \times \{0,0,0,1,1,1\}$ and weights $\boldsymbol{w} = \{1, \frac{1}{\sqrt{2}}, 1\} \times \{1, \frac{1}{\sqrt{2}}, 1\}$.

Then the degree is elevated to $p = 3$ and, as a result, the new set of the sixteen control points are arranged in a 4×4 setup, as shown in Fig. 6(b), where the new knot vector per direction becomes $\Xi' = \{0,0,0,0,1,1,1,1\} \times \{0,0,0,0,1,1,1,1\}$.

Eventually, to produce a true NURBS patch which will continue to accurately represent the circumference of the circle, an inner knot is inserted at the middle of each knot vector in either direction. Therefore, the twenty-five control points are arranged in a 5×5 setup, as shown in Fig. 6(c), and therefore the final knot vector becomes $\Xi'' = \{0,0,0,0,\frac{1}{2},1,1,1,1\} \times \{0,0,0,0,\frac{1}{2},1,1,1,1\}$ while the weights are $\{1, 0.9024, 0.8047, 0.9024, 1\}$.

It is noted that the circle in green colour shown in Fig. 6(b,c,d) is the accurate one which is ensured in all the rational formulations of Model-1 and Model-2.

The above-mentioned 25-DOF model (*MODEL-1*, in a 5×5 setup, as shown in Fig. 6(c), is solved first in conjunction with standard NURBS-based IGA. This is accomplished once using the usual de Boor functions (`spcol` in MATLAB) and another using the equivalent Bézier extraction. Obviously, the results were found to be identical, and the associated errors are shown in the third column of Table 1. It is noted that the aforementioned Bézier extraction was based on increasing the multiplicity of the nine inner knots (in a 3×3 setup) to $p = 3$, thus the involved Bézier operator $\mathbf{C}_e$ *implicitly* utilizes 49 control points.

Then, the 49-DOF model (*MODEL-2*, in a 7×7 setup), of which the control points had been implicitly used in the aforementioned Bézier extraction, is solved using four rational cubic Bézier elements (separated by the two perpendicular thick lines in red colour, as shown in Fig. 6(d)) and the relevant results are shown in the fourth column of Table 1.

Once again, one may observe that MODEL-2 outperforms in accuracy. Since the difference between the degrees of freedom in the two Models is less than 1:2, this problem is not the worst case.

Due to the above fact, in Section 3 the comparison of CPU-times will be reported for only the rectangular cavity as is explained therein.

Table 1. Eigenvalues of the circular cavity with hard walls

Mode	EXACT	Errors (in %) of Calculated Eigenvalues	
		MODEL-1 (25 DOFs)	MODEL-2 (49 DOFs)
1	0	-	-
2	3.389957716671888	0.43	0.07
3	3.389957716671888	0.43	0.07
4	9.328363213746355	0.34	0.32
5	9.328363213746355	3.75	0.47
6	14.681970642123899	1.13	0.96
7	17.649988519749648	11.85	1.99
8	17.649988519749648	11.85	1.99
9	28.276371248725660	13.68	3.25
10	28.276371248725660	98.64	3.25
11	28.424282047372301	97.60	3.01
12	28.424282047372301	157.67	8.97
13	41.160133480153071	140.69	12.39
14	41.160133480153071	159.07	15.31
15	44.972222417793944	179.09	5.53
16	44.972222417793944	179.09	46.28
17	49.218456321694596	219.70	35.55

3 Discussion

If we restrict our discussion to the polynomial degree $p = 3$, then MODEL-1 is of C^2-continuity and leads to a pair of matrices, say ($\mathbf{K}_1$,$\mathbf{M}_1$), whereas MODEL-2 is of C^0-continuity and leads to another pair of matrices, say ($\mathbf{K}_2$,$\mathbf{M}_2$). Each pair of matrices leads to a separate set of eigenvalues and our findings suggest that the set ($\mathbf{K}_2$,$\mathbf{M}_2$) will lead to lower (thus most accurate) values.

Actually, from the three numerical examples of this study, it becomes clear that MODEL-1 (IGA of C^2-continuity) is less accurate than MODEL-2 (Bézier elements of C^0-continuity), when the control points which are implicitly encountered in the Bézier extraction (MODEL-1) are explicitly used to form Bézier elements (MODEL-2).

One may advocate that the aforementioned fact is due to the higher number of DOFs involved in MODEL-2. Actually, in the 1D-problem the 6 DOF of MODEL-1 increased (by 67 percent) to 10 in MODEL-2. In the 2D-rectangular cavity the 40 DOF of MODEL-1 increased (by 180 percent) to 112, while for the circular cavity the 25 DOF of MODEL-1 increased (by 96 percent) to 49 in MODEL-2.

Of course, a fair comparison would require extensive experimentation regarding the computer effort and a relevant cost-benefit analysis, not only for the Helmholtz equation but also for other partial differential equations (PDEs) as well (e.g., Laplace-Poisson, elasticity problems, electro-magnetics, etc.). At the moment, for the examples of this study which restricts to the Helmholtz equation, we have found that MODEL-2 outperforms in accuracy but its superiority in computer effort is sensitive and depends also on the particular procedure followed (see Procedures 2a÷2c, which were defined in sub-section 2.1.2).

To become clear, we focus on the second example (the rectangular cavity of subsection 2.2.1) because it is characterized by the abovementioned largest increase (180 percent) of the control points between MODEL-1 and MODEL-2 (Procedure 2b) thus it is the hardest case. To allow our work to be reproduced by anyone interested, MODEL-1 (Bézier extraction) was programmed in the MATLAB environment cited in [29]. In the latter software only slight modifications have been made in subroutine `FormK` as follows: the Laplace operator (∇^2) of our study has substituted the Navier–Cauchy equations of two-dimensional elasticity therein and thus has led to a different stiffness matrix [K]. Moreover, a mass matrix [M] has been added below the stiffness matrix. The eigenvalues have been calculated using the command `eig(K,M)`. Furthermore, MODEL-2 has been programmed as an independent MATLAB code in which Bernstein polynomials and their derivatives were found using three alternatives given in Table 2 under the header MODEL-2, as follows:

1) Label **spcol**: The `spcol` function, which is inherent in the Spline Toolbox of MATLAB, was used to calculate the Bernstein polynomials in each Bézier element, independently.
2) Label **Analytical**: The four Bernstein polynomials and their derivatives were analytically calculated using the formulas: $B_{0,3} = (1-\xi)^3$, $B_{1,3} = 3\xi(1-\xi)^2, B_{2,3} = 3\xi^2(1-\xi), B_{3,3} = \xi^3$, etc.
3) Label **Data**: After the analytical computation of Bernstein polynomials and their derivatives, the numerical results were tabulated in matrices of size 4×4 and were put in the beginning of the subroutine which estimates the element stiffness and mass matrices.

In both models, the CPU-time has been calculated as the mean average of 30 trials, on the same computer under the same conditions, and the results are given in Table 2. One may observe that CPU-time required for the computer execution of MODEL-2 is smaller than that required in MODEL-1, even for the hardest (worst) case of using the standard `spcol` function while it requires halve computer time when the data at Gauss points are

given in advance. In any case, considering the more accurate results of MODEL-2 shown in Fig. 5:, it is concluded that MODEL-2 outperforms with respect to MODEL-1 from *all* the points of view.

Table 2. CPU-time for a rectangular cavity

MODEL-1	**MODEL-2** (Procedure 2a)		
	Bernstein polynomials using		
	spcol	**Analytical**	**Data**
0.088757	0.080333	0.055735	0.041925

The comparison in Table 2 is quite fair. In both models the Jacobians have been calculated at the 4×4 Gauss points of all the ten B-spline or Bézier elements shown in Fig. 4, accordingly. To make this point still clearer, if –for instance– the element matrices $(\mathbf{K}_e, \mathbf{M}_e)$ are analytically calculated in MODEL-2 once for the first 16-node Bézier element only, then the CPU-time dramatically reduces to only 0.01328 seconds. Note that in MODEL-1 we have to calculate 40 eigenvalues while in MODEL-2 we deal with 112 ones; so the CPU-time in MODEL-2 includes extra computer effort for the computation of a higher number of eigenvalues (performed by the `eig` function).

While the abovementioned Procedure 2b (Bézier element of C^0-continuity) clearly requires less computer effort than the usual Bézier extraction, this is not the case for Procedure 2**d** (Eqs. (6) and (7)). Obviously, this is because Procedure 2d is a *continuation* of MODEL-1. On this issue there is still space for further contribution, with the hope that the extra computer effort will be compensated by the smaller numerical error. For example, regarding the efficient computation of the two involved quadratic forms $\left(\mathbf{K}_e\right)_Q = \left(\mathbf{C}_e^{-1}\right)\left(\mathbf{K}_e\right)\left(\mathbf{C}_e^{-1}\right)^{\mathrm{T}}$ etc., one could apply ideas found in a recent study on symmetric matrices, [30].

Ongoing research on programming details shows that if we wish to calculate both models (i.e. MODEL-1 and MODEL-2: Procedure 2**a**) *simultaneously*, then at each Gauss point we have to perform the Bézier extraction (MODEL-1) and then calculate the Jacobian matrix in terms of the first set of control points (see, [28]). But since after knot insertion the shape of the domain remains parametrically the *same*, the numerical values of the aforementioned Jacobian matrix may be *preserved* for MODEL-2 as well. The latter consists of using the value of the Bernstein polynomials and their local derivatives at the same Gauss points, a procedure that can be done *once* outside the subroutine which performs the numerical integration of the matrices. Unlike the eigenvalue problem of this study, regarding other types of PDEs in the general form $D(u) - f = 0$, the *simultaneous* computation of the variable u and its gradient ∇u at the same points in both models (MODEL-1 and MODEL-2: Procedure 2**a**) is promising to build a new '*error-estimator*' for further adaptation of the control points for a still more accurate numerical solution.

Having already discussed the three Procedures 2(a, b and d), let us now turn to the remaining Procedure 2c, which is related to tensor product cubic Lagrange polynomials (here of uniform 4×4 nodes). This can be programmed as usual, following standard FEM programming rules. The most critical issue is that Lagrange polynomials and their local partial derivatives should preferably be calculated *in advance* at the Gauss points and stored, so the computer effort substantially decreases. Interestingly, Lagrange elements lead to identical eigenvalues with Bézier elements of the same degree. Here, we ought to explain the reason that although Procedure 2b (Bézier elements) *differs* from Procedure 2c (Lagrange elements) in their matrices they both lead to the *same* numerical result (not only in the eigenvalues of this paper but in all other boundary-value problems). The explanation is the fact that both functional sets of these polynomials share the *same* monomials in the form $x^i y^j$, so there is a *linear relationship* between Bernstein-Bézier and Lagrange polynomials. In other words, we only have a change of basis thus the final numerical results become identical. On this issue the interested reader may consult a detailed discussion in [16, pp. 258-268].

In general, MODEL-1 should be carefully programmed otherwise sometimes Bézier extraction may even be slower than the usual IGA. But even our programming is efficient the gain is not always that big. For example, regarding a particular 2D boundary value problem governed by *Laplace* equation, and for control points which are tensor product of Fig. 1 (i.e., 6×6 = 36 control points in total), working with an older hardware we have found that the required CPU-time for the usual IGA is 0.971 sec, for the usual Bézier extraction is 0.928 sec, while for a smart implementation of Bézier extraction is 0.900 seconds (7.3 percent reduction with respect to the usual IGA). It is worthy to mention that in the latter case the univariate Bernstein-Bézier polynomials as well as their local derivatives have been pre-calculated at the four Gauss points in each direction *once* and then stored for future demand (see, [31, p. 50]).

4 Conclusion

In the implementation of NURBS-based isogeometric analysis (IGA), at this date Bézier extraction is the standard procedure. The latter requires that the multiplicity of inner knots increases up to the polynomial degree, thus the Bézier operator is calculated and then the basis functions (of C^2-continuity) are efficiently estimated. In this paper it was shown that implementing trivial matrix operations such as matrix inversion and multiplications on the standard IGA stiffness and mass matrices, it is possible to estimate larger matrices which correspond to the involved Bézier elements characterized by C^0-continuity. Interestingly, the same matrices may be derived treating the Bézier elements independently, i.e., in a similar way with the standard finite element method, and this model showed to outperform.

APPENDIX A

Stiffness and mass matrices for the cubic *Bézier* element of length l_e:

Using the well known formulas

$$k_{ij}=\int_0^{l_e} B_i' B_j' dx \text{ and } m_{ij}=\rho A\int_0^{l_e} B_i B_j dx, \quad \text{(A-1)}$$

in conjunction with the standard Bernstein polynomials of degree three, i.e.,

$$\begin{aligned} &B_1=(1-\xi)^3, B_2=3(1-\xi)^2\xi, \\ &B_3=3(1-\xi)\xi^2, B_4=\xi^3, \end{aligned} \quad \text{(A-2)}$$

it can be found that:

$$\mathbf{K}_{\text{Bezier}}=\frac{3}{10l_e}\begin{bmatrix} 6 & -3 & -2 & -1 \\ -3 & 4 & 1 & -2 \\ -2 & 1 & 4 & -3 \\ -1 & -2 & -3 & 6 \end{bmatrix}, \quad \mathbf{M}_{\text{Bezier}}=\frac{l_e}{420}\begin{bmatrix} 60 & 30 & 12 & 3 \\ 30 & 36 & 27 & 12 \\ 12 & 27 & 36 & 30 \\ 3 & 12 & 30 & 60 \end{bmatrix}. \quad \text{(A-3)}$$

APPENDIX B

Stiffness and mass matrices for the cubic *Lagrange* element of length l_e:

Using the well known formulas

$$k_{ij}=\int_0^{l_e} L_i' L_j' dx \text{ and } m_{ij}=\rho A\int_0^{l_e} L_i L_j dx, \quad \text{(B-1)}$$

in conjunction with the standard Lagrange polynomials of degree three, i.e.,

$$\begin{aligned} &L_1=\tfrac{1}{2}(3\xi-1)(3\xi-2)(1-\xi), L_2=\tfrac{9}{2}\xi(3\xi-2)(\xi-1), \\ &L_3=\tfrac{9}{2}\xi(3\xi-1)(1-\xi), L_4=\tfrac{1}{2}\xi(3\xi-1)(3\xi-2) \end{aligned}, \quad \text{(B-2)}$$

it can be found that:

$$\mathbf{K}_{\text{Lagrange}}=\frac{1}{40l_e}\begin{bmatrix} 148 & -189 & 54 & -13 \\ -189 & 432 & -297 & 54 \\ 54 & -297 & 432 & -189 \\ -13 & 54 & -189 & 148 \end{bmatrix}, \quad \mathbf{M}_{\text{Lagrange}}=\frac{l_e}{1680}\begin{bmatrix} 128 & 99 & -36 & 19 \\ 99 & 648 & -81 & -36 \\ -36 & -81 & 648 & 99 \\ 19 & -36 & 99 & 128 \end{bmatrix}. \quad \text{(B-3)}$$

References:

[1] Taig I.C., *Structural Analysis by the matrix displacement method*, Engl. Electric Aviation Report No. S017, 1961.

[2] Irons B.M., Engineering applications of numerical integration in stiffness matrix, *J.A.I.A.A.*, Vol. 4, No. 11, 1966, pp. 2035-2037.

[3] Zienkiewicz O.C., *The Finite Element Method*, McGraw-Hill, 1977.

[4] Bathe K.-J., *Finite Element Procedures*, Prentice-Hall, 1996.

[5] Gordon W.J., and Hall C.A., Transfinite element methods: Blending-function interpolation over arbitrary curved element domains, *Numerische Mathematik*, Vol. 21, 1973, pp. 109–112.

[6] Cavendish J.C., Gordon W.J., and Hall C.A., Ritz-Galerkin approximations in blending function spaces, *Numerische Mathematik*, Vol. 26, 1976, pp. 155–178.

[7] Farin G., *Curves and surfaces for computer aided geometric design: a practical guide*, Academic Press, 1990.

[8] Piegl L., and Tiller W., *The NURBS Book* (1st ed.), Springer, 1995.

[9] Sederberg T.W., Zheng J., Bakenov A., and Nasri A., T-splines and T-NURCCSs. *ACM Transactions on Graphics*, Vol. 22, No. 3, 2003, pp. 477–484.
[10] Dem'yanovich Y.K., Belyakova O.V., and Le B.T.V., General smoothness of the Hermite type splines, *WSEAS Transactions on Mathematics*, Vol, 17, 2018, pp. 359-368.
[11] Dem'yanovich Y.K., Miroshnichenko I.D., and Musafarova E.F., On splines' smoothness, *WSEAS Transactions on Mathematics*, Vol, 18, 2019, pp. 129-136.
[12] Burova I.G., Application of non-polynomial splines to solving differential equations, *WSEAS Transactions on Mathematics*, Vol, 19, 2020, pp. 531-548.
[13] Garnier L., Druoton L., Bécar J.-P., Fuchs L, and Morin G., Subdivisions of horned or spindle Dupin cyclides using Bézier curves with mass points, *WSEAS Transactions on Mathematics*, Vol, 20, 2021, pp. 756-776.
[14] Kanarachos A., and Deriziotis D., On the solution of Laplace and wave propagation problems using 'C-elements', *Finite Elemenets in Analysis and Design*, Vol. 5, 1989, pp. 97–109.
[15] Provatidis C., and Kanarachos A., Performance of a macro-FEM approach using global interpolation (Coons') functions in axisymmetric potential problems, *Computers & Structures*, Vol. 79, No. 19, 2001, pp. 1769–1779.
[16] Provatidis C.G., *Precursors of Isogeometric Analysis: Finite elements, Boundary elements and Collocation methods*, Springer, 2019.
[17] Kagan P., Fischer A., and Bar-Yoseph P.Z., New B-spline finite element approach for geometrical design and mechanical analysis, *International Journal for Numerical Methods in Engineering*; Vol. 41, No. 3, 1998, pp. 435–458.
[18] Kagan P., and Fischer A., Integrated mechanically based CAE system using B-Spline finite elements, Computer-Aided Design 2000; 32:539–552.
[19] Kagan P., Fischer A., Bar-Yoseph PZ. Mechanically based models: Adaptive refinement for B-spline finite element. International Journal for Numerical Methods in Engineering 2003; 57:1145–1175 (DOI: 10.1002/nme.717)
[20] Höllig K., *Finite element methods with B-splines*, SIAM, 2003.
[21] Schramm U., and Pilkey W.D., The coupling of geometric descriptions and finite element using NURBS—a study in shape optimization, Finite Elem Anal Des, Vol. 15, 1993, pp. 11–34.
[22] Schramm U., and Pilkey W.W., Higher order boundary elements for shape optimization using rational B-splines, *Eng Anal Boundary Elem*, Vol. 14, No. 3, 1994, pp. 255–266.
[23] Inoue K., Kikuchi Y., and Masuyama. T., A NURBS finite element method for product shape design, *J Eng Des*, Vol. 16, No. 2, 2005, pp. 157–174.
[24] Hughes T.J.R., Cottrell J.A., and Bazilevs Y., Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement, *Comput Methods Appl Mech Eng*, Vol. 194, 2005, pp. 4135–4195.
[25] Cottrell J.A., Hughes T.J.R., and Bazilevs Y., *Isogeometric analysis: towards integration of CAD and FEA*, Wiley, 2009.
[26] De Boor C., On calculating with B-splines, *J Approx Theory*, Vol. 6, 1972, pp. 50–62.
[27] Cox M.G., The numerical evaluation of B-splines, *J Inst Math Its Appl*, Vol. 10, 1972, pp. 134–149.
[28] Borden M.J., Scott M.A., Evans J.A., and Hughes T.J.R., Isogeometric finite element data structures based on Bézier extraction of NURBS, *International Journal for Numerical Methods in Engineering*, Vol. 87, 2011, pp. 15–47.
[29] Nguyen T.N., *Isogeometric Finite Element Analysis based on Bézier Extraction of NURBS and T-Splines*, Master's Thesis, Department of Structural Engineering, Faculty of Engineering Science and Technology, NTNU- Norwegian University of Science and Technology, June 14, 2011. https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/236860
[30] Mitrouli M., Polychronou A., Roupa P., and Turek O., Estimating the Quadratic Form $\mathbf{x}^T\mathbf{A}^{-m}\mathbf{x}$ for Symmetric Matrices: Further Progress and Numerical Computations, *Mathematics*, Vol. 9, No. 12, 2021, 1432.
[31] Raditsas J., *Performance of Bézier extraction in isogeometric analysis*, Master's Thesis, National Technical University of Athens, Post-graduate course in "Applied Mathematical Scieces", February 2020 (Advisor: Prof. C. Provatidis, in Greek). https://dspace.lib.ntua.gr/xmlui/handle/123456789/51215

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself
None.