

UNIVERSITÀ DEGLI STUDI DI PAVIA
FACOLTÀ DI INGEGNERIA
DIPARTIMENTO DI INGEGNERIA CIVILE E ARCHITETTURA
CORSO DI LAUREA MAGISTRALE IN INGEGNERIA CIVILE

TESI DI LAUREA

TITLE

**Isogeometric collocation methods for 3D
elastic problems with a focus on laminated
structures**

TITOLO

**Metodi di collocazione isogeometrica per
problemi elastici tridimensionali, con
applicazioni a strutture laminate**

Candidato: Alessia Patton

Relatore: Chiar.mo Prof. Alessandro Reali
Correlatori: Prof. Simone Morganti
Dr. John-Eric Dufour

A.A. 2016/2017

Success is not final, failure is not fatal: it is the courage to continue that counts.
(Winston Churchill)

Acknowledgments

I would like to thank Professor A. Reali, simply a real advisor in the truest sense of the word. For giving me the chance to study this method and to apply it to real life. To see me making mistakes and guiding me to use my eyes. I have learnt so much working with you.

I would also like to thank Professor S. Morganti for his precious advices regarding Neumann B.Cs that brought me to optimize my coding style.

I also wish to thank Dr. J. E. Dufour always willing to help me and discuss things of the field and A. Cattenone for teaching me LaTeX basics.

I dedicate this work to my mum and dad for supporting me in all senses and always believing in me, to my little nieces Elin and Lisa, to my brother Ema and to Ane, whom I've missed so much.

To the amazing people I've met here in Pavia, where my studies began. First of all I would like to thank 'The family', without you living far away home wouldn't be the same. And therefore, thank you Serena for being always by my side in every situation and Valeria because I can trust you and confide in you. Thank you Caro for bringing musicality everywhere, Ale for being the motivator you naturally are and Silvia for making me always laugh.

Thank you Anna for being a true friend, your strenght make me stronger, Elisa your energy is compelling for everyone, Valentina for all the experiences we share.

Finally I would like to thank all my far away mates, who are always there for me despite the distance. To Emilio for making me understand the meaning of friendship, to Ele for your craziness, to Fede a column in engineering sense.

Abstract

In this work we focus on the application of collocation in the field of isogeometric analysis and in particular we propose a new approach in the study of orthotropic composite plates based on this technique.

Isogeometric analysis possesses some intrinsic advantages arising directly from the nature of the method in terms of high continuity of the shape functions used (*B-splines* or *NURBS*). This characteristic is emphasized in *Chapter 1*, after giving ample space to the historical motivations which led to the birth of this method, as well as a comparison with the better known Finite Element Analysis (FEA). For the Galerkin-based methods, the use of numerical integration is fundamental. This operation is costly from a computational point of view. Consequently we make use of *collocation*, described in *Chapter 2*, which, once known the constitutive behaviour, leads to the solution using only the shape functions and their derivatives evaluated in special points called *collocation points*, which obey some specific mathematical requirements, here (2) discussed.

Numerical experiments are shown for structural one-, two-, and three-dimensional problems. In every chapter dedicated to numerical tests, we emphasize the principal computational aspects regarding the implementation in Matlab, which has been carried out for every problem shown in this work, and we also discuss convergence issues of the shown results. In *Chapter 3* we focus on scalar problems and particularly on the solution of Poisson's equations. Furthermore we propose, in *Chapter 4*, isotropic linear elasticity problems.

Imposing Neumann boundary conditions, which are widely discussed in *Chapter 5*, requires special attention. We deal with this topic bringing two numerical examples of engineering relevance: the *2D Patch Test* and the *Pressurized thick-walled cylinder test*.

In the field of structural engineering dynamics plays a fundamental role. Consequently we choose to show the normalized discrete spectrum, which can be found in *Chapter 6*. In its simplicity, this problem shows the potential of isogeometric collocation, also in comparison to results achieved with the finite element method.

The main focus of this work, as already mentioned, is represented by the application of this method to orthotropic laminated composites, which are widely used in a variety of fields such as aerospace or automotive. The interest for their use in the engineering field has grown in recent years because of their light weight and very resistant mechanical properties. One composite element is made of several layers, each of which shows a privileged in-plane resistance direction, due to the nature of the orthotropic constitutive behaviour. Consequently, it is natural to assemble the composite element orientating the layers in an alternated way, which allows to exploit the mentioned resistance properties. In this new proposed approach, described theoretically in *Chapter 7*, we average the material properties in order to better exploit collocation properties. This leads to good results in terms of displacements and in-plane stresses, but to unsatisfactory numerical values in terms of out-of-plane stresses. We therefore focus on a post-processing recovery technique of the out-of-plane stresses showing, in *Chapter 8*, IGA collocation results.

Sommario

Il presente lavoro di tesi è incentrato sull'applicazione della collocazione nell'analisi isogeometrica ed in particolare presenta un approccio innovativo allo studio di piastre composite ortotrope basato su tale tecnica.

L'analisi isogeometrica prevede dei vantaggi intrinseci alla natura del metodo in termini di elevata continuità delle funzioni di forma utilizzate (*B-splines* o *NURBS*). Tale caratteristica viene enfatizzata nel *capitolo 1*, dopo aver lasciato ampio spazio alle motivazioni storiche che portano allo sviluppo di tale metodologia, unite ad un confronto con la più nota analisi agli elementi finiti (FEA). I metodi di Galerkin prevedono l'utilizzo dell'operazione di integrazione numerica, costosa da un punto di vista computazionale. Pertanto ci si è avvalsi della *collocazione*, descritta nel *capitolo 2*, che consente di ricavare la soluzione noto il legame costitutivo, a partire dalle sole funzioni di forma e loro derivate calcolate in punti che soddisfano specifici requisiti matematici, denominati *punti di collocazione* e qui (2 discussi).

Si presentano quindi i problemi studiati dal monodimensionale al tridimensionale. In ogni capitolo che riguarda test numerici, si sottolineano i principali aspetti computazionali riguardanti l'implementazione effettuata avvalendosi del software Matlab ed eseguita per ogni problema mostrato in questo lavoro di tesi. Ampio spazio è lasciato alla discussione della convergenza dei risultati mostrati. Il *capitolo 3* è incentrato su problemi di tipo scalare ed in particolare sulla risoluzione delle equazioni di Poisson, mentre nel *capitolo 4* si discutono problemi riguardanti materiali a comportamento lineare elastico isotropo. Particolare attenzione richiede l'imposizione delle condizioni al contorno di Neumann, ampiamente trattate nel *capitolo 5*. Tale argomento è discusso portando due esempi concreti di importanza ingegneristica: il problema bidimensionale di *Patch Test* e di *Cilindro spesso in pressione*.

Nell'ambito dell'ingegneria strutturale la dinamica riveste un ruolo di fondamentale importanza, pertanto si è scelto di fornire un esempio riguardante il calcolo dello spettro discreto normalizzato, riportato nel *capitolo 6*. Pur nella sua semplicità tale problema mostra le potenzialità della collocazione isogeometrica rispetto a quella ottenuta con il metodo agli elementi finiti.

L'applicazione principale del presente elaborato, come già si è sottolineato, riguarda lo studio di laminati compositi ortotropi, utilizzati in vari campi quali l'industria aerospaziale e automotive. Il loro interesse ingegneristico è cresciuto negli ultimi anni a causa del loro peso contenuto ed elevate caratteristiche di resistenza. Un elemento composito è costituito da più strati, ognuno dei quali presenta una direzione di resistenza privilegiata nel piano dovuta alla natura del legame costitutivo ortotropo. Pertanto nell'assemblaggio dell'elemento composito si prevede un naturale orientamento del materiale alternato strato per strato, che consente di sfruttare in termini di resistenza le proprietà della legge costitutiva adottata. Nell'approccio innovativo introdotto, da un punto di vista teorico nel *capitolo 7*, si vanno ad omogeneizzare tali proprietà del materiale per sfruttare al meglio le proprietà del metodo di collocazione. Questo porta a buoni risultati in termini di spostamenti e sforzi nel piano, ma a risultati insoddisfacenti in termini di sforzi fuori dal piano. Ci si concentra quindi sullo studio di una procedura di post-processing di recupero degli sforzi fuori piano, esponendo nel *capitolo 8*, i buoni risultati ottenuti combinando tale post-processing con la collocazione isogeometrica.

Contents

List of Tables	VI
List of Figures	IX
1 Introduction to <i>Isogeometric Analysis</i>	1
1.1 Main reasons that led to isogeometric analysis within historical framework	1
2 Basic principles of Isogeometric Collocation	4
2.1 An introduction to collocation	4
2.2 Basics of NURBS-Based IGA Collocation	6
2.2.1 B-splines	6
2.2.2 B-spline curves	9
2.2.3 B-Spline Surfaces	10
2.2.4 B-Spline solids	10
2.2.5 Refinement	10
2.2.6 Non-Uniform-Rational B-Splines (NURBS)	14
3 B-Splines-Based IGA Collocation for scalar problems	15
3.1 IGA collocation 1D theoretical formulation	15
3.1.1 Collocation points and theoretical results	15
3.2 Benchmark: 1D scalar problem	16
3.3 Benchmark: 2D Poisson's equation	18
3.4 Benchmark: 3D Poisson's equation	21
4 IGA Collocation for Linear Elastostatics	23
4.1 Linear isotropic elasticity	23
4.2 Isotropic linear elasticity tests	24
4.2.1 Square: 2D linear elasticity problem	24
4.2.2 Cube: 3D linear elasticity problem	26
4.2.3 Mapping	32
5 Neumann Boundary Conditions	49
5.1 Neumann boundary conditions imposition	49
5.2 Pressurized thick-walled cylinder test	50
5.3 Patch Test	53
6 Spectral approximation	55
7 A cost-effective IGA-Collocation approach for ortotropic composite plates	58
7.1 A brief introduction to 3D strategies for laminates	59
7.2 Post-processing approach: Reconstruction from Equilibrium	61

8	Test case: The Pagano layered plate	63
8.1	Theoretical background	63
8.1.1	Analytical solution	65
8.2	Implementation aspects and Numerical results	68
8.2.1	Computational aspects	68
8.2.2	Numerical results	71
9	Conclusions and future perspectives	86

List of Tables

1.1	Comparison between FEA and IGA.	3
3.1	1D scalar problem: comparison between imposed degree of approximation and obtained order of convergence.	18
3.2	2D Poisson's problem: comparison between imposed degree of approximation and obtained order of convergence.	20
3.3	3D Poisson's problem: comparison between imposed degree of approximation and obtained order of convergence.	22
4.1	Square 2D linear elasticity test: comparison between imposed degree of approximation and obtained order of convergence considering x direction.	25
4.2	Cube 3D linear elasticity test: comparison between imposed degree of approximation and obtained order of convergence considering x direction.	27
4.3	2D Quarter of annulus: comparison between imposed degree of approximation and obtained order of convergence considering x direction.	46
4.4	3D Quarter of annulus: comparison between imposed degree of approximation and obtained order of convergence considering x direction.	48
5.1	Clamped quarter of an annulus (mixed B.Cs): comparison between imposed degree of approximation and obtained order of convergence.	52

List of Figures

1.1	Increasing complexity in engineering design in terms of manufacturing time.(Courtesy of General Dynamics / Electric Boat Corporation).	1
1.2	Estimation of the relative time costs of each component of the model generation and analysis process at Sandia National Laboratories. Note that the process of building the model completely dominates the time spent performing analysis. (Courtesy of Michael Hardwick and Robert Clay, Sandia National Laboratories).	2
2.1	Basis functions of order 0, 1, and 2 for uniform knot vector $\Xi = \{0, 1, 2, 3, 4, \dots\}$	7
2.2	Quartic ($p = 4$) basis functions for an open, non-uniform knot vector $\{0, 0, 0, 0, 0, 1, 2, 2, 3, 3, 3, 4, 4, 4, 4, 5, 5, 5, 5\}$.The continuity across an interior element boundary is a direct result of the polynomial order and the multiplicity of the corresponding knot value.	8
2.3	B-spline, piecewise quadratic curve in \mathbb{R}^2 . (a) Control point locations are denoted by \bullet . (b) The knots, which define a mesh by partitioning the curve into elements, are denoted by \blacksquare . Basis functions and knot vector as in Figure 2.5.	9
2.4	Refinement: Knot insertion.	11
2.5	Refinement: Order elevation.	12
2.6	Strategy comparison: k-refinement vs p-refinement.	13
3.1	1D scalar problem: exact displacement solution compared with the approximated one for $p=2$, 10 control points.	17
3.2	1D scalar problem convergence test: number of d.o.fs vs displacement error in terms of L^2 -norm.	17
3.3	2D Poisson's problem: Exact solution.	19
3.4	2D Poisson's problem: Approximate solution.	19
3.5	2D Poisson's problem: difference between approximate solution and exact solution.	19
3.6	2D scalar problem convergence test: number of d.o.fs vs displacement error in terms of L^2 -norm.	20
3.7	3D scalar problem convergence test: number of d.o.fs vs error in terms of L^2 -norm.	21
4.1	Square isotropic linear elastic problem convergence test: geometry of the problem.	24
4.2	Square isotropic linear elastic problem convergence test: number of d.o.fs vs error in terms of L^2 -norm in x direction.	25
4.3	Cube: geometry of the problem.	26
4.4	Cube convergence test: number of d.o.fs vs error in terms of L^2 -norm	26
4.5	Clamped quarter of an annulus. Geometry of the problem.	42
4.6	2D Clamped quarter of an annulus: exact displacement in x direction.	43
4.7	2D Clamped quarter of an annulus: approximate displacement in x direction.	43
4.8	2D Clamped quarter of an annulus: displacements difference between exact and approximate solution in x direction.	44

4.9	2D Clamped quarter of an annulus: exact displacement in y direction.	44
4.10	2D Clamped quarter of an annulus: approximate displacement in y direction. . . .	45
4.11	2D Clamped quarter of an annulus: displacements difference between exact and approximate solution in y direction.	45
4.12	2D Quarter of annulus convergence test: number of d.o.fs vs error in terms of L^2 -norm, considering x direction.	46
4.13	3D Quarter of annulus: geometric definition of the problem.	47
4.14	3D quarter or annulus convergence test: number of d.o.fs vs error in terms of L^2 -norm in x direction.	47
5.1	Clamped quarter of an annulus (mixed B.Cs): geometry of the problem.	50
5.2	Clamped quarter of an annulus (mixed B.Cs): obtained approximate solution. . . .	51
5.3	Clamped quarter of an annulus (mixed B.Cs): difference between obtained approximate solution and exact one.	51
5.4	Clamped quarter of an annulus (mixed B.Cs) convergence test in radial direction: number of d.o.fs vs error in terms of L^2 -norm.	52
5.5	Patch traction test: problem geometry and boundary conditions.	53
5.6	Patch traction test: horizontal displacement field.	54
5.7	Patch traction test: vertical displacement field.	54
6.1	Fixed-fixed rod. Normalized discrete spectra using quadratic finite elements and NURBS.	55
6.2	Nodal finite element basis functions for the quadratic p-method. Note the two distinct types of functions corresponding to end-nodes and mid-nodes. These lead to two distinct difference equations corresponding to the end-point nodes at element boundaries and the mid-point nodes in element interiors.	56
6.3	Fixed-fixed rod. Normalized discrete spectra.	57
6.4	1D eigenvalue problem with linear parametrization. Convergence of the first three eigenvalues for p=3.	57
6.5	1D eigenvalue problem with linear parametrization. Convergence of the first three eigenvalues for p=4.	57
7.1	Isogeometric shape functions used in the two considered approaches. Layerwise approach with C_0 lines at material discontinuities on the left and Homogeneized single-element approach on the right.	59
8.1	Plate geometry.	63
8.2	Plate notation and B.Cs	65
8.3	Program Flowchart.	70
8.4	Computed σ_{11} solutions for the 3D Pagano plate problem with 3 layers, at the top, and 11, at the bottom, at the position $X = 0.25L$ and $Y = 0.25L$. The blue solid line represents the analytical solution, the black cross represent the solution obtained with no post-processing treatment.	71
8.5	Computed σ_{22} solutions for the 3D Pagano plate problem with 3 layers, at the top, and 11, at the bottom, at the position $X = 0.25L$ and $Y = 0.25L$. The blue solid line represents the analytical solution, the black cross represent the solution obtained with no post-processing treatment.	72
8.6	Computed σ_{12} solutions for the 3D Pagano plate problem with 3 layers, at the top, and 11, at the bottom, at the position $X = 0.25L$ and $Y = 0.25L$. The blue solid line represents the analytical solution, the black cross represent the solution obtained with no post-processing treatment.	73

8.7	Computed σ_{13} solutions for the 3D Pagano plate problem with 3 layers, at the top, and 11, at the bottom, at the position $X = 0.25L$ and $Y = 0.25L$. The blue solid line represents the analytical solution, the black cross represent the solution obtained with no post-processing treatment, and the red circles represent the solution after post-processing treatment.	74
8.8	Computed σ_{23} solutions for the 3D Pagano plate problem with 3 layers, at the top, and 11, at the bottom, at the position $X = 0.25L$ and $Y = 0.25L$. The blue solid line represents the analytical solution, the black cross represent the solution obtained with no post-processing treatment, and the red circles represent the solution after post-processing treatment.	75
8.9	Computed σ_{33} solutions for the 3D Pagano plate problem with 3 layers, at the top, and 11, at the bottom, at the position $X = 0.25L$ and $Y = 0.25L$. The blue solid line represents the analytical solution, the black cross represent the solution obtained with no post-processing treatment, and the red circles represent the solution after post-processing treatment.	76
8.10	Displacements inside the domain for fix x-y plane: displacement in x directon, at the top; in y direction, in the middle; in z direction, at the bottom.	77
8.11	Recovered (red solid line) σ_{13} compared to the analytical one (blue crosses) for several in plane positions. L is the total length of the plate, that in this case is $L = 110\text{mm}$ (being $L = S t$ with $t = 11\text{mm}$ and $S = 10$), while the number of layers is 11.	78
8.12	Recovered (red solid line) σ_{23} compared to the analytical one (blue crosses) for several in plane positions. L is the total length of the plate, that in this case is $L = 110\text{mm}$ (being $L = S t$ with $t = 11\text{mm}$ and $S = 10$), while the number of layers is 11.	79
8.13	Recovered (red solid line) σ_{33} compared to the analytical one (blue crosses) for several in plane positions. L is the total length of the plate, that in this case is $L = 110\text{mm}$ (being $L = S t$ with $t = 11\text{mm}$ and $S = 10$), while the number of layers is 11.	80
8.14	Maximum relative error between post-processed and analytical σ_{13} , at the top, σ_{23} , at the bottom, obtained considering different length-to-thickness ratios, while the degree of the shape functions is 6 and the number of layers is 3.	81
8.15	Maximum relative error between post-processed and analytical σ_{13} , at the top, σ_{23} , at the bottom, obtained considering different length-to-thickness ratios, while the degree of the shape functions is 6 and the number of layers is 11.	82
8.16	Maximum relative error between post-processed and analytical σ_{13} , at the top, σ_{23} , at the bottom, obtained considering different number of elements and layers, while the degree of the shape functions is 6 and $S=10$	83
8.17	Maximum relative error between post-processed and analytical σ_{13} , at the top, σ_{23} , at the bottom, obtained considering different number of elements and layers, while the degree of the shape functions is 4 and $S=10$	84
8.18	Maximum relative error between post-processed and analytical σ_{13} at the bottom, using different number of elements and layers, while the degree of the shape functions is 6 and $S=20$	85

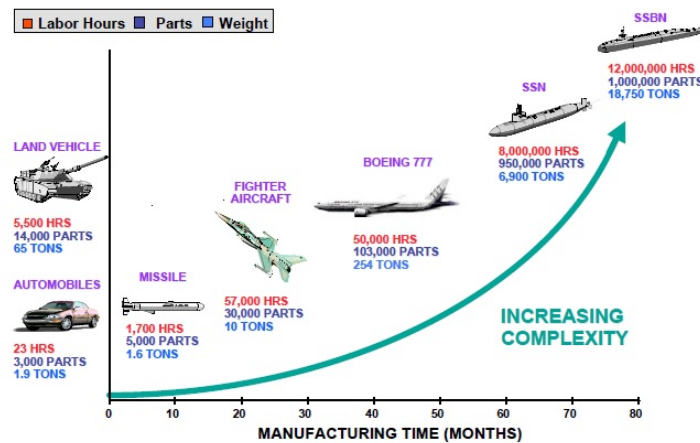
Chapter 1

Introduction to *Isogeometric Analysis*

In this chapter our aim is to introduce Isogeometric Analysis (IGA) from an historical perspective and to focus on the main motivations that lead to this method, supplying a parallel with the well-known Finite Element Analysis (FEA) and stressing some issues that arises in Computer Aided Design (CAD)-FEA interactions.

1.1 Main reasons that led to isogeometric analysis within historical framework

Isogeometric analysis arises from the existing gap between the worlds of Finite Element Analysis (FEA) and Computer-Aided Design (CAD)(see [20]) and seeks to connect these two worlds that are born independently. Historically CAD generated files that needed to be translated into analysis-suitable-geometries, meshed, and input to large-scale finite element analysis. Any mesh refinement operation required interaction with CAD geometry. From a quantitative point of view modelling phase is now estimated to take over the 80% of the overall analysis time.



Courtesy of General Dynamics / Electric Boat Corporation

Figure 1.1: Increasing complexity in engineering design in terms of manufacturing time.(Courtesy of General Dynamics / Electric Boat Corporation).

As a tangible example we report Figure 1.1, which demonstrates that nowadays a typical automobile consists of about 3,000 parts, a fighter jet over 30,000, the Boeing 777 over 100,000, and a modern nuclear submarine over 1,000,000, all of which need to be modelled. Clearly in order to accomplish a full analysis there are many preparatory steps involved, which not only require modelling but other phases aswell as Figure 1.2 shows. The anatomy of the process has been studied by Ted Blacker, Manager of Simulation Sciences, Sandia National Laboratories. At Sandia, mesh generation accounts for about 20% of overall analysis time, whereas creation of the analysis-suitable geometry requires about 60%, and only 20% of overall time is actually devoted to analysis itself, as Figure 1.2 clarifies. Indeed the 80/20 modelling/analysis ratio seems to be a very common industrial experience. Considering that simulation is used in a wide range of

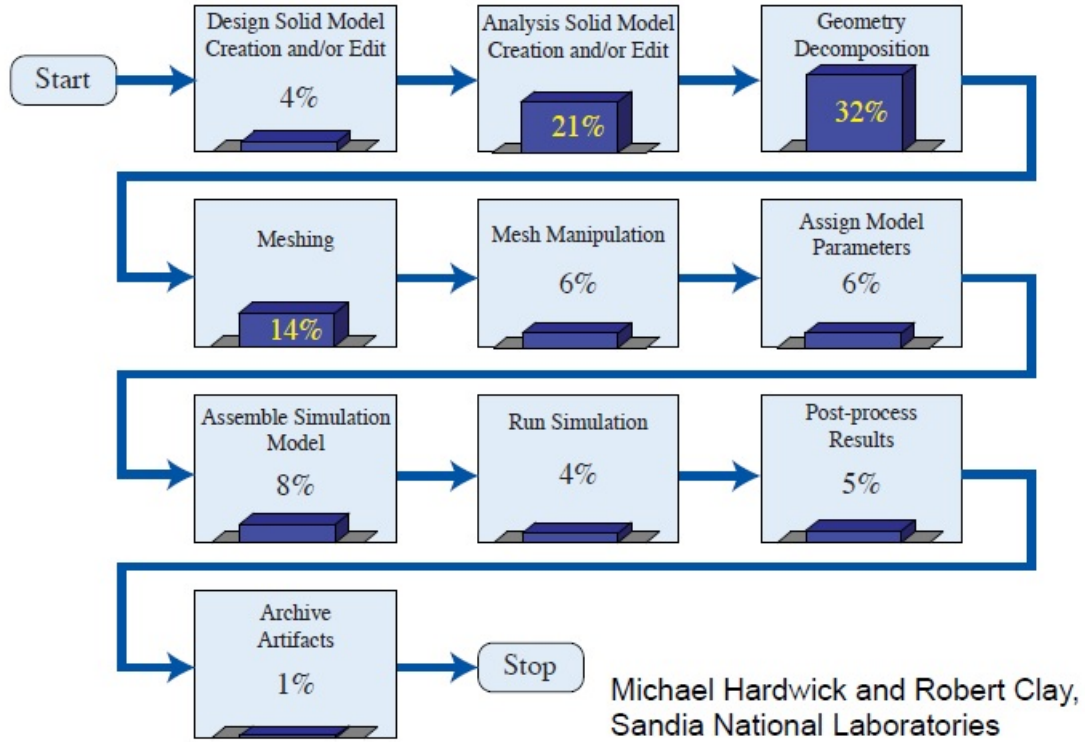


Figure 1.2: Estimation of the relative time costs of each component of the model generation and analysis process at Sandia National Laboratories. Note that the process of building the model completely dominates the time spent performing analysis. (Courtesy of Michael Hardwick and Robert Clay, Sandia National Laboratories).

fields, from the scientific modeling of natural systems to artificial and human ones, remarkable economic implications arise which require a reduction in terms of analysis time. We can therefore conclude, from the remarks we made, that the two main reasons that contributed to IGA birth are represented by the economic dimension of the market and the necessity to improve the time required by the analysis. Now that we stressed the main motivations that led to IGA, we can reasonably ask ourselves why this CAD-FEA interaction didn't happened from the beginning. The answer to this question is contained within the historical framework. In fact FEA developed itself between the 50's and 60's, while CAD advent is to be traced back only between the 70's and 80's. However it's only around 2005 that Hughes et al. ([37]) came up with the idea to make this two worlds interact in the sense that IGA employs typical functions from Computer Aided Design, such as B-splines and NURBS, as shape functions, concept instead related to FEA. In the end the basis functions describe both geometry and field variables in accordance with the isoparametric

paradigm. As a consequence making these two worlds interact means, at a practical level, that the exact CAD geometry is preserved and the geometrical description is unique. Besides a costsaving simplification of the typically expensive mesh generation and refinement processes required by finite element analysis, thanks to the high-regularity properties of its basis functions, IGA showed a better accuracy per degree of freedom and an enhanced robustness with respect to standard finite elements [20]. To sum up IGA prerogatives as an alternative to FEA in a qualitative way we can say that IGA:

- provides a precise and efficient geometry modelling and an integration of design and analysis;
- has smooth basis functions with compact support;
- has superior approximation properties;
- mesh refinement operations are simplified.

Since the transition from one type of analysis to the other is not direct, a parallel between IGA and FEA, which summarizes their main features, is here reported in Table 1.1. The aim for this brief comparison is not to perform a full description of the depicted differences, but just to give an idea of the two kind of analysis. What follows will be clearer in Chapter 2 dedicated to IGA collocation, where an ample mathematical base will be provided in the field of Isogeometric Analysis.

Table 1.1: Comparison between FEA and IGA.

FEA	IGA
nodal coordinates and variables	control points and variables
element	knot span
1 mesh (FE)	2 meshes (physical and control meshes)
interpolatory basis (polynomials)	non-interpolatory basis (NURBS)
approximated geometry	exact geometry
subdomains	patches (parameter space is local to them)
h-refinement	h-p-k refinements possibilities

To conclude this chapter we want to stress some issues that arise from CAD-FEA interactions:

- CAD operations differ from FEA ones and in order to support this statement we bring a practical example. Let's consider a simple lamina, a square with a hole and our goal is to mesh it in the best possible way. FEA approach will probably divide this figure into four equal slices and refine each of them. On the other side CAD sees the 2D object as a set of boolean operations, here reported. The square is parameterized as well as the circle-curve, which is later on eliminated as a surface from the square.
- In the CAD field 3D objects are still bivariate. In fact if we consider a sphere, as example, FEA considers this element as three dimensional, while CAD approach sees only its outer surface. This latter remark can be considered less relevant since shells simulation dominates the market.

Chapter 2

Basic principles of *Isogeometric Collocation*

In this chapter our purpose is to outline the principal features of collocation methods and to present the mathematical definitions useful to understand this work of thesis. We therefore describe B-splines, their geometry and derivatives from which we can construct NURBS (Non Rational B-splines). In the end we discuss refinement operations.

2.1 An introduction to collocation

We recall that IGA aims at integrating design and analysis by employing typical functions from Computer Aided Design for describing both geometry and field variables. As a consequence adopting the isoparametric concept leads to a cost-saving simplification of expensive mesh generation and refinement processes required by standard FEA. Moreover, thanks to the high-regularity properties of its basis functions, IGA has shown a better accuracy per-degree-of-freedom and an enhanced robustness with respect to standard FEA in a number of applications, as we see in [59], ranging from solids and structures (see, e.g., Auricchio et al. [7], 2010b; Borden et al., 2012 [13]; Caseiro et al., 2014 [17]; Cottrell et al., 2006 [19], 2007 [21]; de Falco et al., 2011 [24]; Dhote et al., 2014 [26]; Elguedj et al., 2008 [28]; Hughes et al., 2008 [40], 2014 [39]; Lipton et al., 2010 [47]; Morganti et al., 2014 [52]; Reali, 2006 [57]) to fluids (see, e.g., Akkerman et al., 2007 [1]; Bazilevs et al., 2007 [9]; Bazilevs and Hughes, 2008 [11]; Buffa et al., 2011 [15]; Liu et al., 2014 [48]; Gomez et al., 2010 [31]), opening also the door to geometrically flexible discretizations of higher-order partial differential equations in primal form (see, e.g., Auricchio et al., 2007 [4]; Gomez et al., 2008 [30]; Kiendl et al., 2009 [45]).

However, a well-known important issue of IGA is related to the development of efficient integration rules when higher-order approximations are employed. In fact, element-wise Gauss quadrature, typically used for finite elements and originally adopted for Galerkin-based IGA, does not properly take into account inter-element higher continuity leading to sub-optimal array forming costs, significantly affecting the performance of IGA methods. In an attempt to address this issue taking full advantage of the special possibilities offered by IGA, isogeometric collocation schemes have been proposed in [3]. The aim was to optimize the computational cost still relying on IGA geometrical flexibility and accuracy.

Collocation main idea, in contrast to Galerkin-type formulations, consists of the discretization of the governing partial differential equations in strong form, evaluated at suitable points. Consequently, isogeometric collocation does not require integral computation, resulting in a very fast method providing superior performance with respect to Galerkin formulations in terms of both assembling operations and order of convergence (computational costs), in particular when

higher-order approximation degrees are adopted [62].

Isogeometric collocation has been particularly successful in the context of structural elements [8], where isogeometric collocation has proven to be particularly stable in the context of mixed methods. In particular, Bernoulli–Euler beam and Kirchhoff plate elements have been proposed by [58], and shear-deformable structural elements have been considered in a number of papers. Mixed formulations both for Timoshenko initially-straight planar beams [22] and for curved spatial rods [6] have been proposed and studied, and then successfully extended to the geometrically nonlinear case [66]. Isogeometric collocation has been moreover successfully applied to the solution of Reissner–Mindlin plate problems in [42], and a new single-parameter formulation for shear-deformable beams, recently introduced by Kiendl et al. [43], has been solved also via IGA collocation. Since its introduction, many promising significant works on isogeometric collocation methods have been published also in other fields, including phase-field modeling [33], contact [49],[46], nonlinear elasticity [46]. Moreover, the combination with different spline spaces, like hierarchical splines, generalized B-splines, and T-splines, has been successfully tested in [62],[50],[18], while alternative effective selection strategies for collocation points have been proposed in [2],[33],[51]. In general, IGA collocation methods prove to be a profitable choice in all those situations where evaluation and assembly costs are dominant, as in explicit structural dynamics where the computational cost is dominated by divergence evaluations at quadrature points for the calculation of the residual force vector ([5],[62]).

2.2 Basics of NURBS-Based IGA Collocation

In this section, we aim at introducing the basic mathematical concepts of NURBS-based IGA collocation. After some preliminaries on B-Splines, mainly inspired by [20], we discuss the possible refinement techniques offered by IGA, and conclude describing NURBS from a mathematical point of view, in order to enhance the framework presented in Chapter 1. In Chapter 4 we will instead leave ample space to collocation schemes in a linear elasticity field both from a theoretical and a numerical point of view.

2.2.1 B-splines

B-splines are piece-wise polynomial curves whose components are defined as the linear combination of B-spline basis functions and the components of some points in the space, referred to as *control points*. Fixed the order of the B-spline (i.e. the degree of polynomials), in order to construct the basis functions we have to introduce the so-called knot vector, which is a fundamental ingredient for this operation.

2.2.1.1 Knot vector

A *knot vector* in one dimension is a non-decreasing set of coordinates in the parameter space, written $\Xi = \{\xi_1, \xi_2, \dots, \xi_{n+p+1}\}$, where $\xi_i \in \mathbb{R}$ is the i^{th} *knot*, i is the knot index, $i = 1, 2, \dots, n + p + 1$, p is the polynomial order, and n is the number of basis functions used to construct the B-spline curve or the number of control points which define the *control mesh*. Knots, as we pointed out, live in the parameter space and they partition this space into elements or *knot spans*. They can also be repeated, that is, more than one knot may take on the same value. The multiplicities of knot values have important implications for the properties of the basis as we will later discuss further on. A knot vector is said to be *uniform* if its knots are equally spaced and *non-uniform* otherwise. In our work we will use *open knot vectors* i.e. knot vectors whose first and last control points are repeated $p + 1$ times. As a consequence, in one dimension, basis functions formed from open knot vectors are interpolatory at the ends of the parameter space interval $[\xi_1, \xi_{n+p+1}]$.

2.2.1.2 Basis function

Having defined what a knot vector represents, B-splines basis functions are defined recursively starting with piecewise constants ($p = 0$):

$$N_{i,0}(\xi) = \begin{cases} 1 & \text{if } \xi_i \leq \xi < \xi_{i+1} \\ 0 & \text{otherwise} \end{cases} \quad (2.1)$$

For $p = 1, 2, 3, \dots$, they are defined by

$$N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi) \quad (2.2)$$

which is referred as *Cox-de Boor recursion formula* (Cox, 1971; de Boor, 1972 [12]). We remark that this result is already implemented in the Matlab toolbox used in this work of thesis and excludes indeterminate mathematical forms as $\frac{0}{0}$. The results of applying (2.1) and (2.2) to the uniform knot vector $\Xi = \{0, 1, 2, 3, 4, \dots\}$ are presented Figure 2.1. For B-spline functions with $p = 0$ and $p = 1$, we have the same result as for standard piecewise constant and linear finite element functions, respectively.

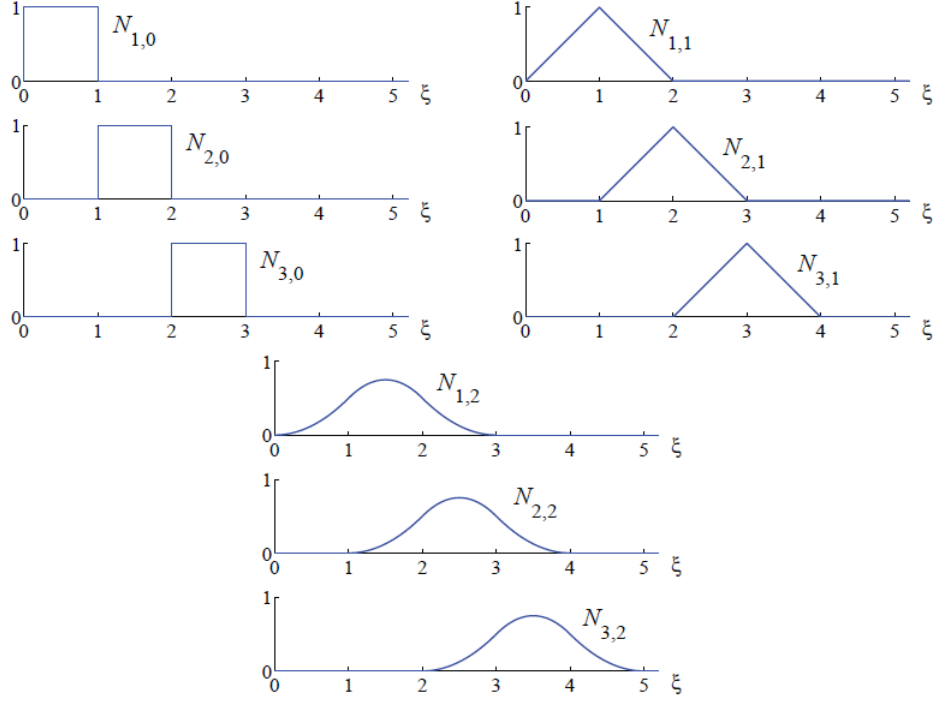


Figure 2.1: Basis functions of order 0, 1, and 2 for uniform knot vector $\Xi = \{0, 1, 2, 3, 4, \dots\}$.

Quadratic B-spline basis functions, however differ from their FEA counterpart as Figure 2.1 shows. They are each identical but shifted relative to each other, whereas the shape of a quadratic finite element function depends on whether it corresponds to an internal node or an end node. This “homogeneous” pattern continues for the B-splines as we continue to higher-orders. We want also to recall Figure 2.1 because it can be used in Chapter 6 to understand the different FEA and IGA behaviour showed in the proposed dynamic 1D test which concerns a free vibrations problem. Among other functions properties we have that:

- the basis constitutes a partition of unity, that is, $\forall \xi$

$$\sum_{i=1}^n N_{i,p}(\xi) = 1 \quad (2.3)$$

- each basis function is pointwise non negative $N_{i,p}(\xi) \geq 0, \forall \xi$ over the entire domain, meaning that all of the entries of a mass matrix will be positive and therefore implications in developing lumped mass schemes arise;
- each p -th order function has $p - 1$ continuous derivatives across the element boundaries (i.e., across the knots), consideration which represents one of the most distinctive features of isogeometric analysis;
- the support of the B-spline functions of order p is always $p + 1$ knot spans, in particular, when a knot has multiplicity p , the basis function is C^0 and interpolatory at that location;
- if a knot has multiplicity of k , the functions are C^{p-k} -continuous in correspondence;
- the functions are C^{p-1} -continuous, if internal knots are not repeated.

This relationship between continuity and the multiplicity of the knots is even more apparent in Figure 2.2, in which we have a fourth order curve with differing levels of continuity at every element boundary.

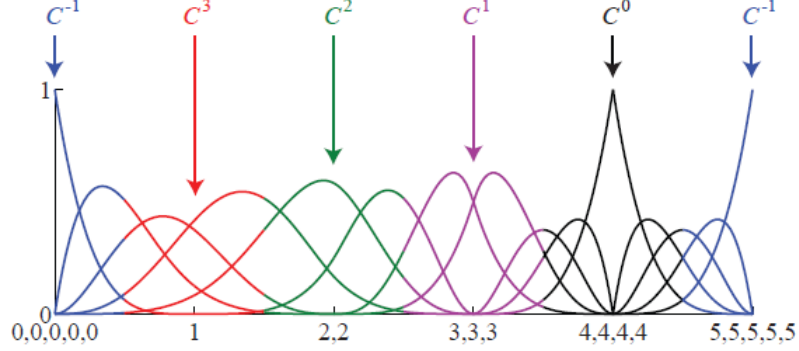


Figure 2.2: Quartic ($p = 4$) basis functions for an open, non-uniform knot vector $\Xi = \{0, 0, 0, 0, 0, 1, 2, 2, 3, 3, 3, 4, 4, 4, 4, 5, 5, 5, 5, 5\}$. The continuity across an interior element boundary is a direct result of the polynomial order and the multiplicity of the corresponding knot value.

At the first internal element boundary, $\xi = 1$, the knot value appears only once in the knot vector, and so we have the maximum level of continuity possible: $C^{p-1} = C^3$. At each subsequent internal knot value, the multiplicity is increased by one, and so the number of continuous derivatives is decreased by one. We underline again that when a knot value is repeated p times, in this case at $\xi = 4$, the C^0 basis is interpolatory. The basis is also interpolatory at the boundary of the domain, where the open knot vector demands that the first and last knot value be repeated $p + 1$ times. The result is C^{-1} -continuity, that is, the basis is fully discontinuous, naturally terminating the domain.

2.2.1.3 Derivatives of B-spline basis function

The derivatives of B-spline basis functions are efficiently represented in terms of B-spline lower order bases. For a given polynomial order and knot vector, the derivative of the i -th basis function is given by

$$\frac{d}{d\xi} N_{i,p}(\xi) = \frac{p}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) - \frac{p}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi) \quad (2.4)$$

The previous result can be generalized as follows

$$\frac{d^k}{d\xi^k} N_{i,p}(\xi) = \frac{p}{\xi_{i+p} - \xi_i} \left(\frac{d^{k-1}}{d\xi^{k-1}} N_{i,p-1}(\xi) \right) - \frac{p}{\xi_{i+p+1} - \xi_{i+1}} \left(\frac{d^{k-1}}{d\xi^{k-1}} N_{i+1,p-1}(\xi) \right) \quad (2.5)$$

and in the end we have after expanding

$$\frac{d^k}{d\xi} N_{i,p}(\xi) = \frac{p!}{(p-k)!} \sum_{j=0}^k \alpha_{k,j} N_{i+j,p-k}(\xi) \quad (2.6)$$

$$\alpha_{0,0} = 1, \quad (2.7)$$

$$\alpha_{k,0} = \frac{\alpha_{k-1,0}}{\xi_{i+p-k+1} - \xi_i}, \quad (2.8)$$

$$\alpha_{k,j} = \frac{\alpha_{k-1,j} - \alpha_{k-1,j-1}}{\xi_{i+p-k+1} - \xi_{i+j}} \quad j = 1, \dots, k-1, \quad (2.9)$$

$$\alpha_{k,k} = \frac{-\alpha_{k-1,k-1}}{\xi_{i+p+1} - \xi_{i+k}} \quad (2.10)$$

The denominator of several of these coefficients can be zero in the presence of repeated knots. Whenever this happens, the coefficient is defined to be zero. Efficient algorithms for these calculations can be found in [55].

2.2.2 B-spline curves

We have seen that, given the order of the B-Spline and knowing the knot vector, it is possible to construct n basis functions $N_{i,p}$. Now, if we consider a set of n *control points* in \mathbb{R}^d , we can obtain the components of the piece-wise polynomial B-Spline curve $\mathbf{C}(\xi)$ of order p by taking the linear combination of the basis functions weighted by the components of control points, as follows

$$\mathbf{C}(\xi) = \sum_{i=1}^n N_{i,p}(\xi) \mathbf{B}_i \quad (2.11)$$

where \mathbf{B}_i is the i -th control point. The piece-wise linear interpolation of the control points is called *control polygon*. In Figure 2.3 we report, as an example, a quadratic 2D B-Spline curve, where the control points can be seen on the left, while on the right side we can observe the knot location which differs from the control points location and the resulting mesh. We remark that a

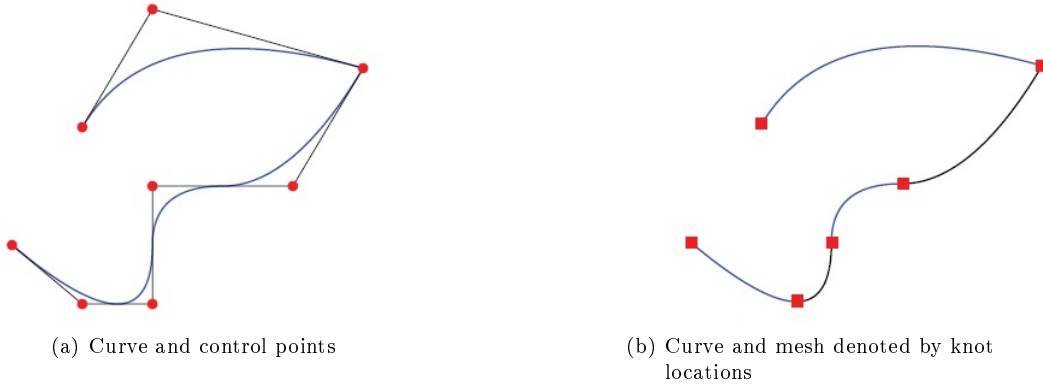


Figure 2.3: B-spline, piecewise quadratic curve in \mathbb{R}^2 . (a) Control point locations are denoted by \bullet . (b) The knots, which define a mesh by partitioning the curve into elements, are denoted by \blacksquare . Basis functions and knot vector as in Figure 2.5.

B-Spline curve has continuous derivatives of order $p-1$, which can be decreased by k if a knot or a control point has multiplicity $k+1$.

2.2.3 B-Spline Surfaces

Performing tensor products operation, B-spline surfaces can be constructed starting from a net of $n \times m$ control points $\mathbf{B}_{i,j}$, $i = 1, 2, \dots, n$, $j = 1, 2, \dots, m$, that form a *control net* and considering knot vectors $\Xi = [\xi_1, \dots, \xi_{n+p+1}]$ and $H = [\eta_1, \dots, \eta_{m+q+1}]$, where p and q are the polynomial orders. Having defined from the two knot vectors the 1D basis functions $N_{i,p}$ and $M_{j,q}$, we can therefore construct the B-Spline surface as:

$$\mathbf{S}(\xi, \eta) = \sum_{i=1}^n \sum_{j=1}^m N_{i,p}(\xi) M_{j,q}(\eta) \mathbf{B}_{i,j} \quad (2.12)$$

2.2.4 B-Spline solids

Tensor product B-spline solids are defined in analogous fashion to B-spline surfaces. Given now a *control lattice* $\mathbf{B}_{i,j,k}$, $i = 1, 2, \dots, n$, $j = 1, 2, \dots, m$, $k = 1, 2, \dots, l$, polynomial orders p, q and r and knot vectors $\Xi = [\xi_1, \dots, \xi_{n+p+1}]$, $H = [\eta_1, \dots, \eta_{m+q+1}]$ and $Z = [\zeta_1, \dots, \zeta_{l+r+1}]$, the B-spline solid can be defined as:

$$\mathbf{S}(\xi, \eta, \zeta) = \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^l N_{i,p}(\xi) M_{j,q}(\eta) L_{k,r}(\zeta) \mathbf{B}_{i,j,k} \quad (2.13)$$

2.2.5 Refinement

In contrast with FEA, IGA as already mentioned possess three different refinement approaches: *knot insertion*, *order elevation*, and *k-refinement*.

2.2.5.1 Knot insertion

The first mechanism by which the basis is enriched is knot insertion and, as the name suggests, is the IGA counterpart of FEA *h-refinement* since new knots are inserted and therefore the number of knot spans is increased. As a matter of fact, knots may be inserted without changing a curve geometrically or parametrically. Given a knot vector $\Xi = [\xi_1, \dots, \xi_{n+p+1}]$ we can "extend it" to $\bar{\Xi} = [\bar{\xi}_1 = \xi_1, \bar{\xi}_2, \dots, \bar{\xi}_{n+p+1} = \xi_{n+p+1}]$ such that the old knot vector is contained in the new one. Therefore the new $n + m$ basis functions derive from Cox De Boor relations (2.1), (2.2), which apply the enriched knot vector. Also $n + m$ new control points are formed from linear combination of the old ones $\bar{\mathcal{B}}$ by

$$\bar{\mathcal{B}} = \mathbf{T}^p \mathcal{B} \quad (2.14)$$

where

$$T_{ij}^0 = \begin{cases} 1 & \bar{\xi}_i \in [\xi_j, \xi_{j+1}) \\ 0 & \text{otherwise} \end{cases} \quad (2.15)$$

and

$$T_{ij}^{q+1} = \frac{\bar{\xi}_{i+q} - \xi_j}{\xi_{j+q} - \xi_j} T_{ij}^q + \frac{\xi_{j+q+1} - \bar{\xi}_{i+q}}{\xi_{j+q+1} - \xi_{j+q}} T_{i+1,j}^q \quad \text{for } q = 0, 1, 2, \dots, p-1 \quad (2.16)$$

A knot insertion example is presented in Figure 2.4.

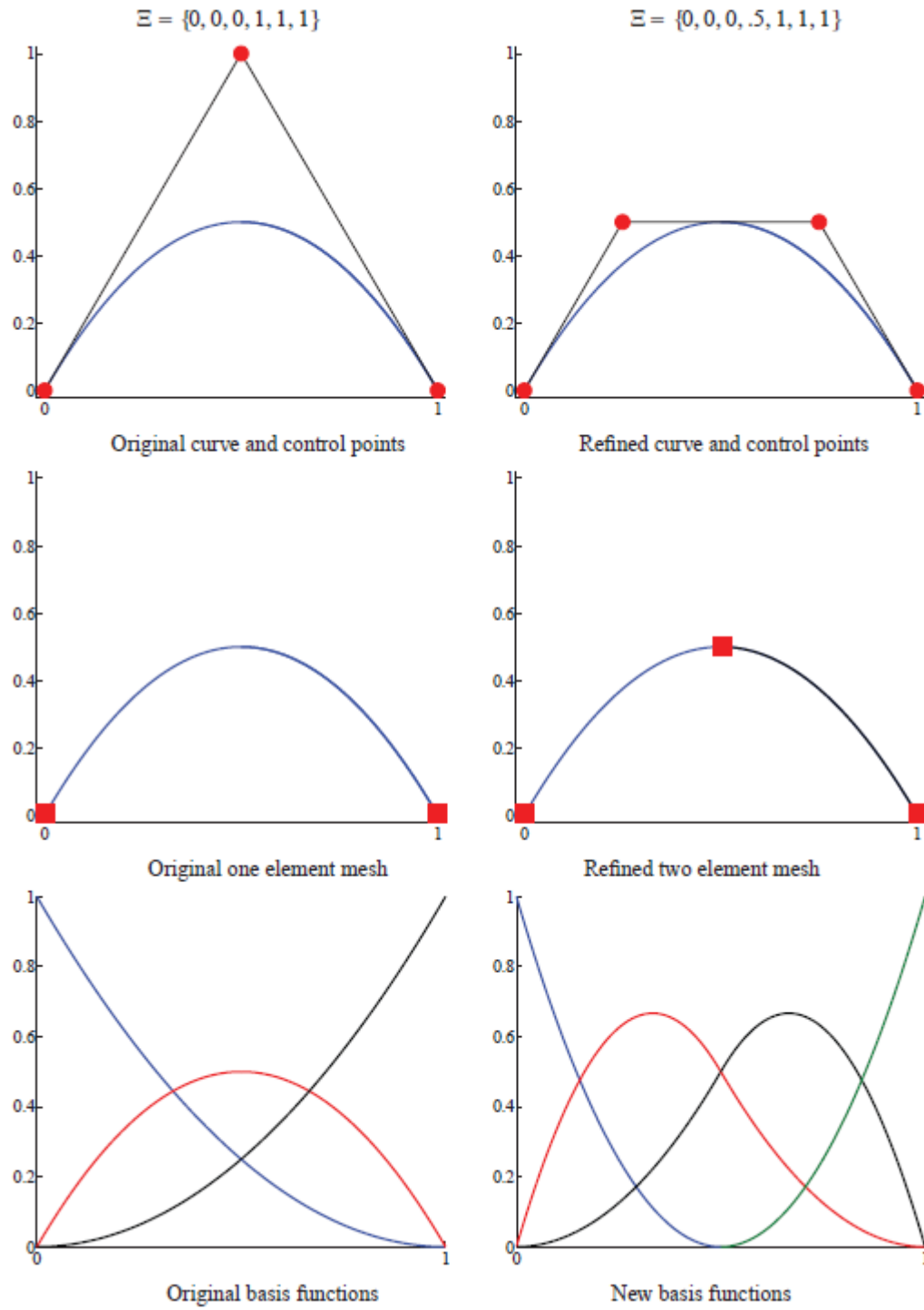


Figure 2.4: Refinement: Knot insertion.

2.2.5.2 Order elevation

As its name implies, the process involves raising the polynomial order of the basis functions used to represent the geometry. Recalling that the basis has $p - m_i$ continuous derivatives across element boundaries, it is clear that when p is increased, m_i must also be increased if we want to preserve the discontinuities in the various derivatives already existing in the original curve. During order elevation, the multiplicity of each knot value is increased by one, but no new knot values are added. As with knot insertion, neither the geometry nor the parameterization are changed. The process for order elevation begins by replicating existing knots until their multiplicity is equal to the polynomial order. The next step is to elevate the order of the polynomial on each of these individual segments. Lastly, excess knots are removed to combine the segments into one, order-elevated, B-spline curve. Several efficient algorithms exist which combine the mentioned steps so as to minimize the computational cost of the process [55]. We report in Figure 2.5 an example of order elevation, which shows the differences between the original setting and the refined one.

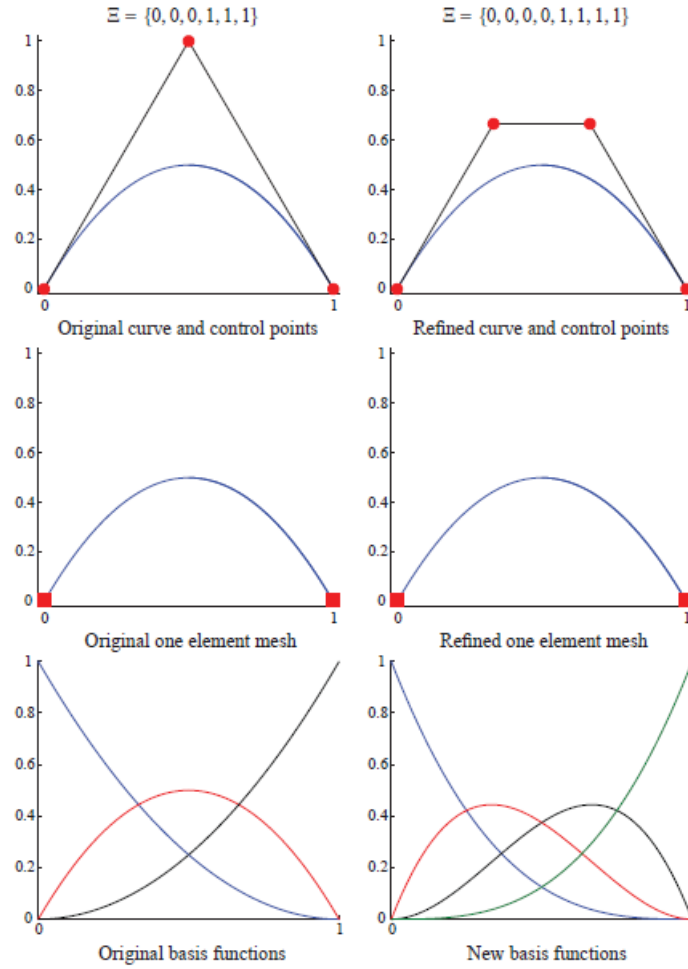


Figure 2.5: Refinement: Order elevation.

2.2.5.3 k-refinement

If a unique knot value, is inserted between two distinct knot values in a curve of order p , the number of continuous derivatives of the basis functions at the original knot value is $p - 1$. If we subsequently elevate the order to q , the multiplicity of every distinct knot value (including the knot just inserted) is increased so that discontinuities in the p^{th} derivative of the basis are preserved. That is, the basis still has $p - 1$ continuous derivatives at the original knot value, although the polynomial order is now q . If, instead, we elevate the order of the original, coarsest curve to q and only then insert the unique knot value, the basis would have $q - 1$ continuous derivatives at the original knot value. We refer to this last strategy as k-refinement. In the end we have stated that inserting knots and degree elevating operations are not commutative. Since the assertion is not trivial, in Figure 2.6 we compare p and k-refinement to stress the different steps who belongs to these two strategies.

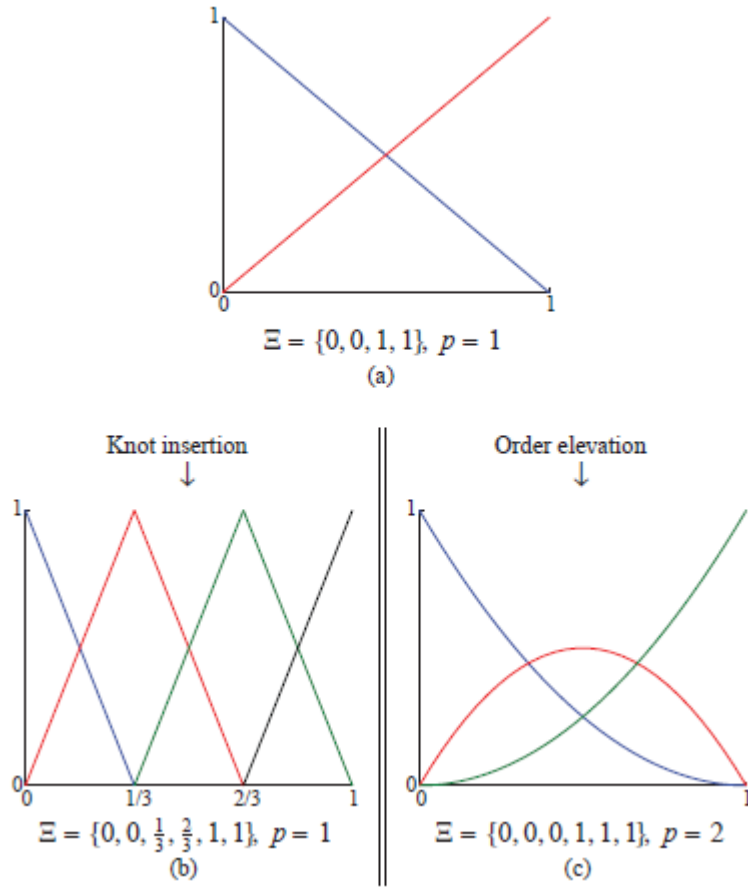


Figure 2.6: Strategy comparison: k-refinement vs p-refinement.

2.2.6 Non-Uniform-Rational B-Splines (NURBS)

A rational B-spline in \mathbb{R}^d is the projection of a non-rational (polynomial) B-Spline defined in $(d+1)$ -dimensional homogeneous coordinate space back into d -dimensional physical space. In this way a great variety of geometric entities can be constructed and in particular conic sections can be obtained exactly. The projective transformation of a B-Spline curve, or *projective curve* leads to a rational polynomial and this is the reason for the name “rational” B-Splines. To obtain a NURBS curve in \mathbb{R}^d , we have to start from a set \mathbf{B}_i^w ($i = 1, \dots, n$) of control points (*projective points*) for a B-spline curve in \mathbb{R}^{d+1} with knot vector Ξ . Then the control points for the NURBS curve are:

$$(\mathbf{B}_i)_j = (\mathbf{B}_i^w)_j / w_i, \quad j = 1, \dots, d \quad (2.17)$$

with

$$w_i = (\mathbf{B}_i^w)_{d+1} \quad (2.18)$$

where \mathbf{B}_i is the j^{th} component of the vector \mathbf{B}_i and w_i is referred to as the i^{th} weight. Therefore dividing the projective control points by the weights is equivalent to applying the projective transformation to them. The NURBS basis functions of order p are then defined as:

$$R_i^p(\xi) = \frac{N_{i,p}(\xi)w_i}{W(\xi)} = \frac{N_{i,p}(\xi)w_i}{\sum_{i=1}^n N_{i,p}(\xi)w_i} \quad (2.19)$$

where we call *weighting function*

$$W(\xi) = \sum_{i=1}^n N_{i,p}(\xi)w_i \quad (2.20)$$

and their first and second derivatives are referred as

$$\frac{d}{d\xi} R_i^p(\xi) = w_i \frac{W(\xi)N'_{i,p}(\xi) - W'(\xi)N_{i,p}(\xi)}{(W(\xi))^2} \quad (2.21)$$

where $N'_{i,p}(\xi) = \frac{d}{d\xi} N_{i,p}(\xi)$ and $W'(\xi) = \sum_{i=1}^n N'_{i,p}(\xi)w_i$ and

$$\begin{aligned} (R_i^p)''(\xi) = & \frac{N''_{i,p}(\xi)w_i}{\sum_{i=1}^n N_{i,p}(\xi)w_i} + \frac{2N_{i,p}(\xi)w_i(\sum_{i=1}^n N'_{i,p}(\xi)w_i)^2}{(\sum_{i=1}^n N_{i,p}(\xi)w_i)^3} + \\ & - \frac{2N'_{i,p}(\xi)w_i \sum_{i=1}^n N'_{i,p}(\xi)w_i + N_{i,p}(\xi)w_i \sum_{i=1}^n N''_{i,p}(\xi)w_i}{(\sum_{i=1}^n N_{i,p}(\xi)w_i)^2} \end{aligned} \quad (2.22)$$

The NURBS curve components are the linear combination of the basis functions weighted by the components of control points:

$$(\mathbf{C}(\xi))_j = \sum_{i=1}^n R_i^p(\xi)\mathbf{B}_i \quad (2.23)$$

Rational surfaces and solids are defined in an analogous way in terms of the basis functions, respectively:

$$R_{i,j}^{p,q}(\xi, \eta) = \frac{N_{i,p}(\xi)M_{j,q}(\eta)w_{i,j}}{\sum_{i=1}^n \sum_{j=1}^m N_{i,p}(\xi)M_{j,q}(\eta)w_{i,j}} \quad (2.24)$$

and

$$R_{i,j,k}^{p,q,l}(\xi, \eta, \zeta) = \frac{N_{i,p}(\xi)M_{j,q}(\eta)L_{k,r}(\zeta)w_{i,j,k}}{\sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^l N_{i,p}(\xi)M_{j,q}(\eta)L_{k,r}(\zeta)w_{i,j,k}} \quad (2.25)$$

We remark that if all weights are equal and therefore we can simplify, obtaining that NURBS become B-splines.

Chapter 3

B-Splines-Based IGA Collocation for scalar problems

In the following chapter we introduce the theoretical background that leads to the development of a collocation scheme in 1D and we provide several tests in the field of scalar problems from one to three dimensions.

3.1 IGA collocation 1D theoretical formulation

In this section we formulate the collocation scheme in a 1D setting.

Formulation. Let f, a_0, a_1 , be real functions in $C^0[a, b]$, with $a < b$ given real numbers. Let $g_0, g_1 \in \mathbb{R}$ be scalars and $BC_1, BC_1: C^1[a, b] \rightarrow \mathbb{R}$ be linear operators. We are interested in the following simple one-dimensional model differential problem. Find a real function $u \in C^2[a, b]$ such that

$$\begin{cases} u''(x) + a_1(x)u'(x) + a_0(x)u(x) = f(x) & \forall x \in (a, b) \\ BC_i(u) = g_i & i = 0, 1 \end{cases} \quad (3.1)$$

where u', u'' represent the first and second derivatives of u , respectively (we note that in the following we will indicate the derivative operator of order i also as $D^i, i \in \mathbb{N}$). We assume that the presented equation has one and only one solution u , and that the boundary condition operators BC_i are linearly independent on $Ker(D^2)$, that is, on the space of linear functions.

To discretize the presented problem via IGA collocation, we proceed as follows.

Given $n \in \mathbb{N}$, let $\mathbb{V}_{n+2} \subset C^2[a, b]$ be a NURBS space of dimension $n + 2$ on the interval $[a, b]$, associated with a spline space $\hat{\mathbb{S}}_{n+2} \subset C^2[0, 1]$ on the parametric interval $[0, 1]$. With standard assumptions on the one-dimensional geometrical map F , we consider $DF > 0$ on the parametric domain $[0, 1]$. Given, for all $n \in \mathbb{N}, \tau_1 < \tau_2 < \dots < \tau_n$ assigned collocation points in $[a, b]$, we obtain the following discrete problem: Find $u_n \in \mathbb{V}_{n+2}$ such that

$$\begin{cases} u''_n(\tau_j) + a_1(\tau_j)u'_n(\tau_j) + a_0(\tau_j)u_n(\tau_j) = f(\tau_j) & \forall x \in (a, b) \\ BC_i(u_n) = g_i & i = 0, 1 \end{cases} \quad (3.2)$$

3.1.1 Collocation points and theoretical results

The discrete problem is defined once a strategy for the selection of the n collocation points is set. Such a selection is of paramount importance, because it directly influences the stability and convergence properties of the collocation scheme. In the IGA collocation literature, the images of so-called Greville abscissae (see de Boor, 2001 [23]) have been widely adopted as the default

choice for collocation points. *Greville abscissae* are n points easily defined from the knot vector as

$$\bar{\xi}_i = \frac{\xi_{i+1} + \xi_{i+2} + \dots + \xi_{i+p}}{p} \quad (3.3)$$

and are well known in the CAD literature for a number of properties, among which the fact that they typically give a stable interpolation (except in some cases when high degrees are combined with particular non-uniform meshes). The selection of points guaranteeing a stable interpolation is a fundamental issue for a collocation scheme, since it is proven in Auricchio et al. (2010a) [3] that this implies optimal convergence (i.e., of order $p - 1$) in the $W^{2,\infty}$ -norm (or, equivalently, in the H^2 -norm). Such a proof is valid only in 1D and cannot be extended to higher dimensions. However extensive numerical testing has shown that the convergence rates obtained in 1D are attained also in higher dimensions. Moreover, optimal convergence rates are not recovered in the L^∞ - and $W^{1,\infty}$ -norms (or, equivalently, in the L^2 - and H^1 -norms), where it has been numerically shown that orders of convergence p and $p - 1$ for even and odd degrees, respectively, are attained. It is important to note that, despite not being optimal in the L^2 - and H^1 -norms as it happens instead for Galerkin methods, the obtained orders of convergence are increasing with p , whereas the cost of collocation is much lower than that of Galerkin approaches of the same order, especially as p increases. This makes IGA collocation very competitive with respect to Galerkin on the basis of an accuracy-to-computational-cost ratio, in particular when higher degrees (e.g., $p > 3$) are adopted (see, e.g., [62]).

3.2 Benchmark: 1D scalar problem

First of all we want recall the L^2 -norm definition, which will be used to verify if our collocation scheme resembles the known exact displacement solution

$$err_{L_2} = \sqrt{\frac{\int |u - u_h|^2}{\int |u|^2}} \quad (3.4)$$

Another L^2 -norm approximate definition is possible accordingly to the *Mean Value Theorem for Integrals*, if the considered points are evenly spaced.

$$err_{L_2} = \sqrt{\frac{\sum |u - u_h|^2}{\sum |u|^2}} \quad (3.5)$$

In order to prove the previously discussed convergence rates, we propose the following 1D problem in $[0, 1]$

$$\begin{cases} au(x)'' + bu(x)' + cu(x) = f(x) \\ u(0) = u(1) = 0 \end{cases} \quad (3.6)$$

which admits the exact solution

$$u(x) = \sin(2\pi x) \quad (3.7)$$

The analytical strong form is consequently approximated by B-splines shape functions $\mathbf{u}_h(x) = \mathbf{N}_p(\xi)\hat{\mathbf{u}}$, while the shape of the loading f is obtained inserting the exact solution, (3.6), in (3.7). We report in Figure 3.1 the approximate displacement compared with the analytical one, obtained using $p = 2$ and 10 control points. We therefore observe that both of them respect the prescribed boundary conditions.

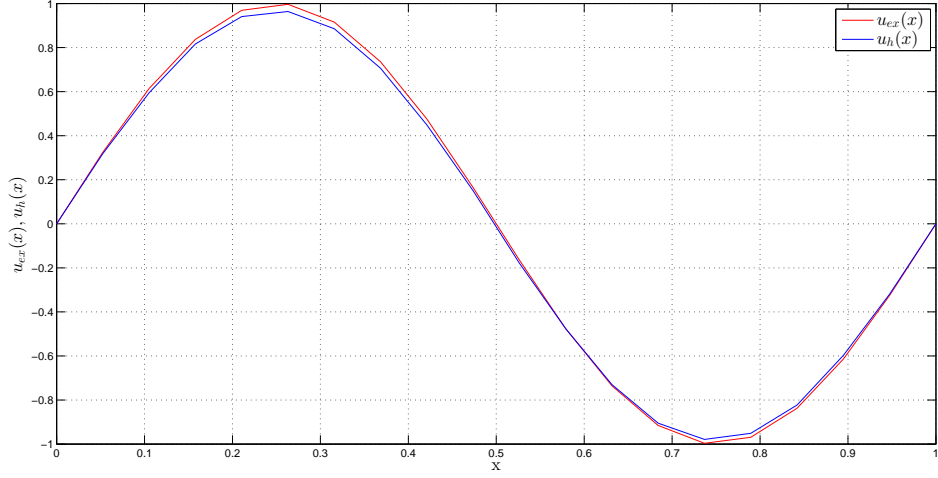


Figure 3.1: 1D scalar problem: exact displacement solution compared with the approximated one for $p=2$, 10 control points.

We also display Figure 3.2, that reports the number of degrees of freedom compared to the displacement error in terms of L^2 -norm, which confirms the convergence rates discussed in Chapter 2. They are computed calculating the slope starting from the last two numerical values, shown in the graphic. In order to obtain Figure 3.2 we considered degrees of approximation from 2 to 7 and 10 up to 130 control points for each analysis.

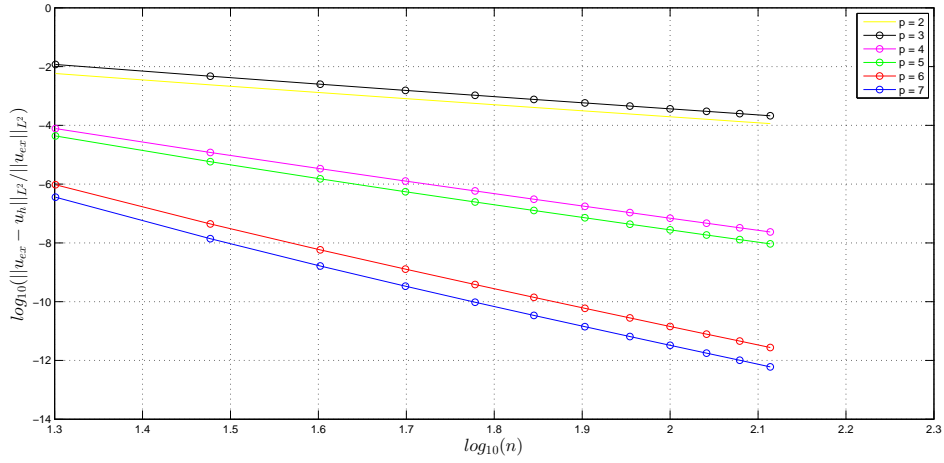


Figure 3.2: 1D scalar problem convergence test: number of d.o.fs vs displacement error in terms of L^2 -norm.

The obtained convergence rates for this case are reported in Table 3.1 and prove to be p for even degrees and $p - 1$ for even ones.

Table 3.1: 1D scalar problem: comparison between imposed degree of approximation and obtained order of convergence.

Degree of approximation	Order of convergence
2	2.03
3	2.05
4	4.11
5	4.17
6	6.25
7	6.40

3.3 Benchmark: 2D Poisson's equation

Accordingly with what we claimed in the first section we extend now our analysis to a 2D case and in particular we consider bidimensional Poisson's equation in $[0, 1]^2$ to see whether the prescribed order of convergence are maintained.

$$\begin{cases} u_{,xx} + u_{,yy} = -f(x, y) \\ u = 0 \end{cases} \quad \forall (x, y) \in \partial\Omega_D \quad (3.8)$$

and we consider as exact solution

$$u(x, y) = \sin(2\pi x)\sin(2\pi y) \quad (3.9)$$

To be specific this means that the boundary conditions prescribes zero displacement and the solution is now a double sinusoidal load. The problem we are dealing with is scalar and therefore an extension of the given error definition is not needed.

We therefore substitute our approximation

$$\mathbf{u}_h(x, y) = \mathbf{N}_p(\xi)\mathbf{M}_q(\eta) \quad (3.10)$$

into (3.8) and obtain our collocation scheme. We stress, with a little abuse of notation which we keep for all the chapter, that (3.10), as well as (3.2) and (3.11), implies that we are considering tensor product, spacing all the collocation points.

We consider 1D test results as a model we would like to obtained, accordingly to the considered case. We therefore plot in Figure 3.3 the exact solution and we compare it with the approximate one in Figure 3.4. Both the solutions respect the prescribed boundary conditions, in fact at the limit of the domain we obtain zero displacement and also the shape, showed in 3.3 and 3.4, resembles the double sinus solution. We also plot the difference of the two in Figure 3.5, showing that by means of $p = q = 2$ and 10 collocation points per direction a good level of approximation is obtained.

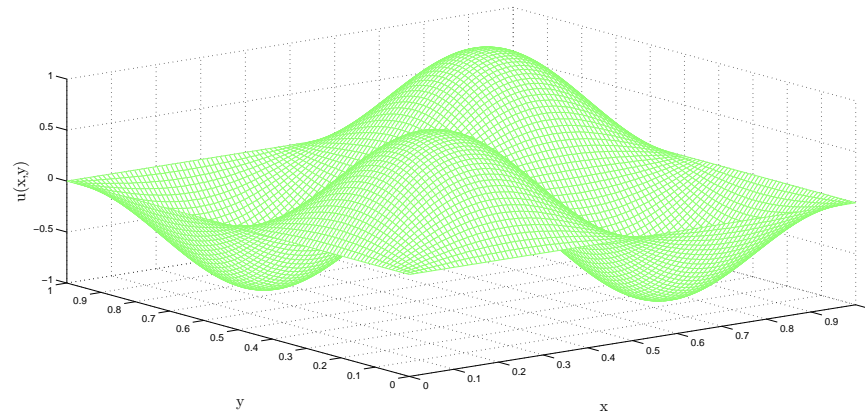


Figure 3.3: 2D Poisson's problem: Exact solution.

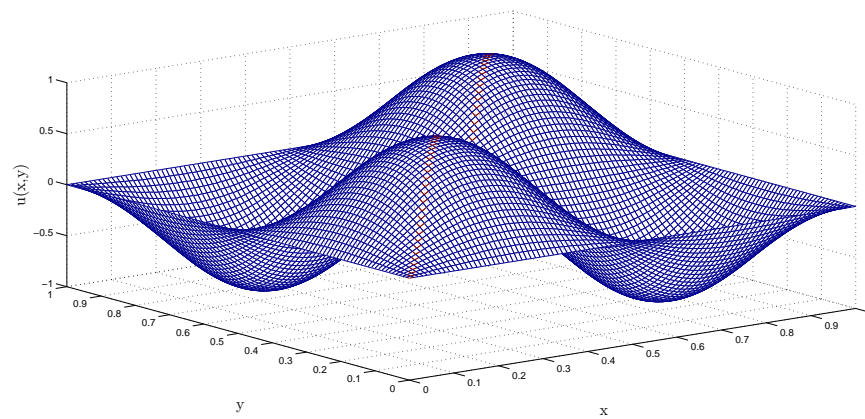


Figure 3.4: 2D Poisson's problem: Approximate solution.

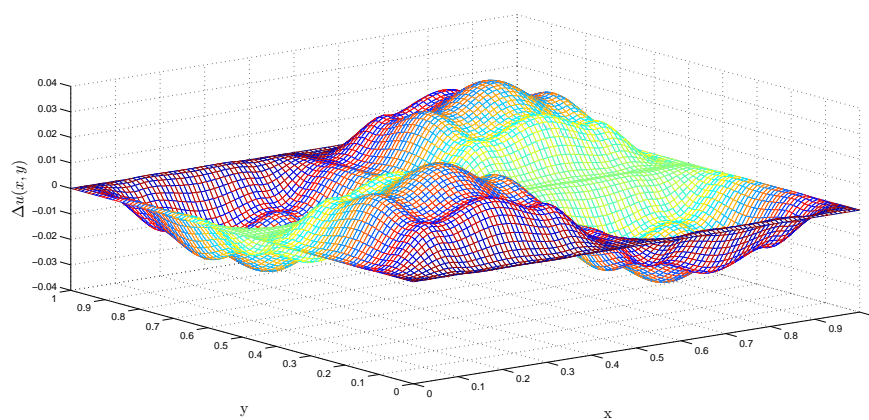


Figure 3.5: 2D Poisson's problem: difference between approximate solution and exact solution.

In Figure 3.6 we report the number of degrees of freedom compared to the displacement error in terms of L^2 -norm, which again confirms the convergence rates. This graphic is obtained testing for each direction degrees of approximation from 2 to 7, using 10 to 130 degrees of freedom and considering a step of 10 d.o.fs for each convergence analysis. The obtained convergence rates for

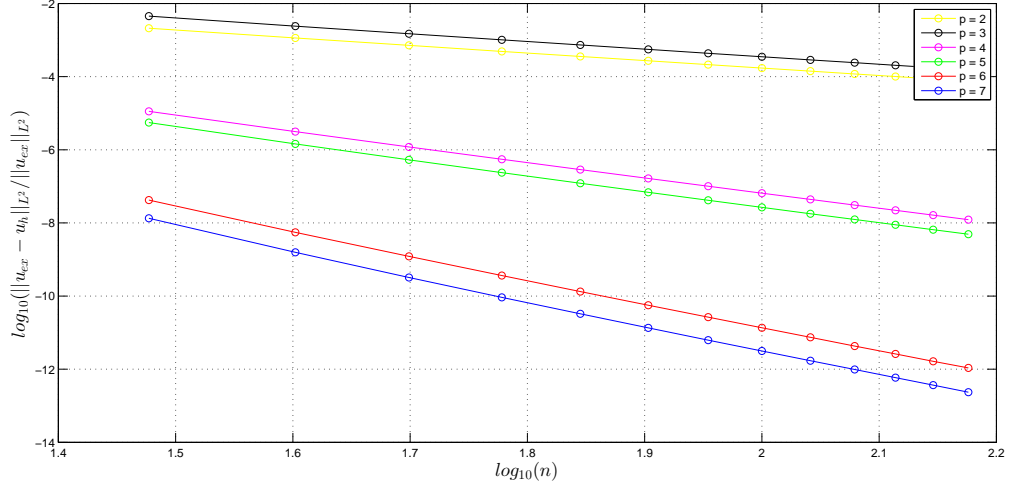


Figure 3.6: 2D scalar problem convergence test: number of d.o.fs vs displacement error in terms of L^2 -norm.

this case are reported in Table 3.2 and confirm the prescribed convergence rates.

Table 3.2: 2D Poisson's problem: comparison between imposed degree of approximation and obtained order of convergence.

Degree of approximation	Order of convergence
2	2.03
3	2.05
4	4.11
5	4.17
6	6.25
7	6.40

3.4 Benchmark: 3D Poisson's equation

In the end we report our dimensional extension of the order of convergence for a 3D cases. We therefore consider now Poisson's 3D equation in $[0, 1]^3$ to test if the prescribed order of convergence are maintained

$$\begin{cases} u_{,xx} + u_{,yy} + u_{,zz} = -f(x, y, z) \\ u = 0 \end{cases} \quad \forall (x, y, z) \in \partial\Omega_D \quad (3.11)$$

and we consider as exact solution

$$u(x, y, z) = \sin(2\pi x)\sin(2\pi y)\sin(2\pi z) \quad (3.12)$$

The solution is now a triple sinus and is accordingly zero at the boundary. We keep the partial differential equation as it is, in strong form and substitute our approximation

$$\mathbf{u}_h(x, y, z) = \mathbf{N}_p(\xi)\mathbf{M}_q(\eta)\mathbf{M}_l(\zeta) \quad (3.13)$$

into (3.11) and obtain our collocation scheme to be implemented. In the present chapter we will not underline implementation aspects since we are widely exposing them in Chapter 4. From Figure 3.7 we can point out that also, as far as the presented 3D case is concerned, we get convergence and we underline the presence of a little of superconvergence, which is expected for the highest degrees. Moreover we used a less number of d.o.fs to perform this 3D test due to unavoidably computational limits. In order to produce Figure 3.7 we used 10 to 30 collocation points per direction, testing the approximation degrees from 2 to 7.

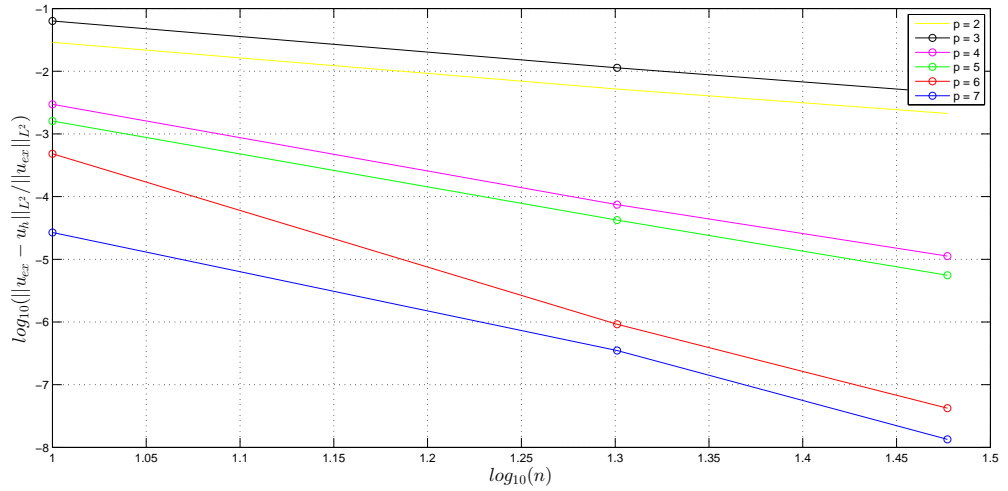


Figure 3.7: 3D scalar problem convergence test: number of d.o.fs vs error in terms of L^2 -norm.

As we expect, the obtained order of convergence showed in Table 3.3 confirms the prescribed convergence rates.

Table 3.3: 3D Poisson's problem: comparison between imposed degree of approximation and obtained order of convergence.

Degree of approximation	Order of convergence
2	2.21
3	2.27
4	4.66
5	5
6	7.6
7	8

Chapter 4

IGA Collocation for Linear Elastostatics

In the following chapter we aim at introducing linear elasticity in collocation schemes from a theoretical point of view. After that we will move to collocation solution schemes showing several benchmark tests. The proposed problems space from very trivial geometries towards examples which require the introduction of 'mapping' and the necessity to use NURBS function. This chapter is fundamental, as far as implementation is concerned and accordingly we propose and comment two of the several 3D codes implemented in this work of thesis.

4.1 Linear isotropic elasticity

As introduced in Chapter 2 and better exposed in the Chapter 3 collocation methods requires a solution scheme starting from a strong form, in our case in terms of displacements. In order to achieve our goal we recall equilibrium equations

$$\sigma_{ij,j} + b_i = 0 \quad \text{in } \Omega \quad (4.2)$$

$$\sigma_{ij} n_j = t_i \quad \text{on } \partial\Omega_N \quad (4.3)$$

$$\bar{u}_i = u_i \quad \text{on } \partial\Omega_D \quad (4.4)$$

as well as compatibility relations as follows

$$\epsilon_{ij} = \frac{u_{i,j} + u_{j,i}}{2} \quad (4.5)$$

To have a well-balanced problem (15 unknown versus 15 equations) we need to also recall constitutive relations

$$\sigma_{ij} = \mathbb{C}_{ijkl} \epsilon_{kl} \quad (4.6)$$

We insert compatibility equations in the constitutive ones and finally we apply them to equilibrium relations, in order to obtain a solution in terms of displacements in strong form (with $i, j = 2, 3$, according to the considered case):

$$\mu u_{i,jj} + (\lambda + \mu) u_{j,ji} + b_i = 0 \quad \text{in } \Omega \quad (4.8)$$

$$\overline{u_i} = u_i \quad \text{on } \partial\Omega_D \quad (4.9)$$

These two set of equations will prove enough for this chapter, but to complete our scheme we also report consequentially Neumann B.Cs, whose application will be later discussed in Chapter 6.

$$\mu(u_{i,j} + u_{j,i})n_j + \lambda u_{j,j} = t_i \quad \text{on } \partial\Omega_N \quad (4.10)$$

The next step consist of applying our approximation of the state variables (B-splines for the first class of proposed problems and NURBS for the second one). We remark that in the proposed tests the exact solution is known and from that, we can recover b_i expression using for example "Matlab Symbolic Toolbox".

4.2 Isotropic linear elasticity tests

For what concerns all the presented results we recall from Chapter 3 that the item that allows to show whether we are converging or not is the introduced L^2 - *norm* approximation

$$err_{L_2} = \sqrt{\frac{\sum |u - u_h|^2}{\sum |u|^2}} \quad (4.11)$$

The error here considered for the 1D case is extended to the 2D case and 3D case simply considering, for each case, the appropriate components. We want to underline that, given the simmetry of the presented problems we will report the results only for the displacement solution in x direction.

4.2.1 Square: 2D linear elasticity problem

As a first test we consider a square having side equal to one so that no mapping is required. The geometry of the problem can be derived from Figure 4.1. The square is clamped at the boundary and the exact solution is a double sine which we observe respects boundary conditions

$$u(x, y) = v(x, y) = \sin(2\pi x)\sin(2\pi y) \quad (4.12)$$

We therefore procede to substitute our displacements approximation in (4.1) in order to obtain

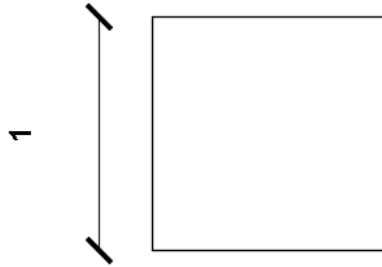


Figure 4.1: Square isotropic linear elastic problem convergence test: geometry of the problem.

a collocation scheme, as the method states, directly into the partial differential equation strong form, as we widely discussed in Chapter 3.

We also display Figure 4.2,4.1 that reports the number of degrees of freedom compared to the displacement error in terms of L^2 -norm in x direction, and also Table 4.1, which confirms the convergence rates, discussed in Chapter 2. We underline that in order to obtain 4.2 we used 2 to 7 degrees of approximation and 20 to 130 collocations points per direction.

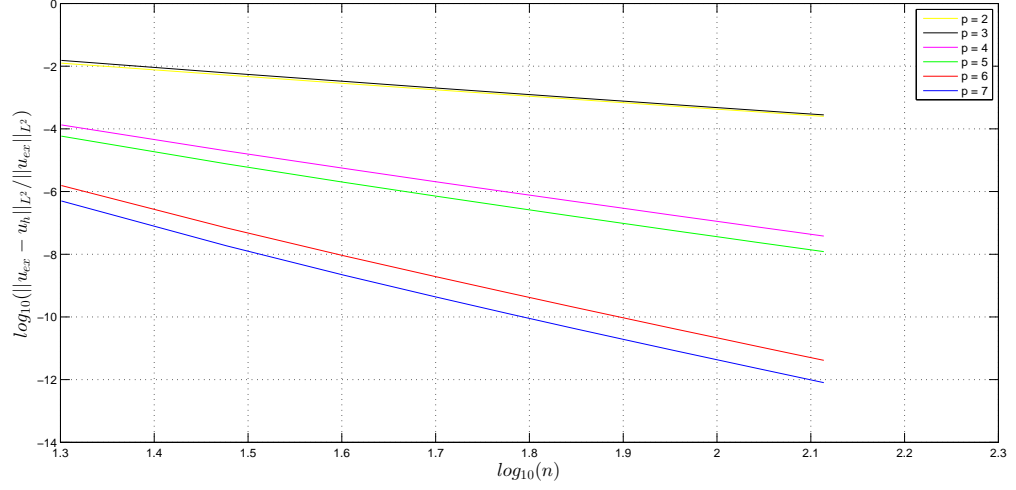


Figure 4.2: Square isotropic linear elastic problem convergence test: number of d.o.fs vs error in terms of L^2 -norm in x direction.

Table 4.1: Square 2D linear elasticity test: comparison between imposed degree of approximation and obtained order of convergence considering x direction.

Degree of approximation	Order of convergence
2	1.9
3	2.05
4	4.07
5	4.17
6	6.21
7	6.40

4.2.2 Cube: 3D linear elasticity problem

We consider now a cube, whose side is equal to one, as we recall in Figure 4.3, and therefore no mapping is required. The exact solution consist of a triple sinus loading, as states the reported equation

$$u(x, y, z) = v(x, y, x) = \sin(2\pi x)\sin(2\pi y)\sin(2\pi z) \quad (4.13)$$

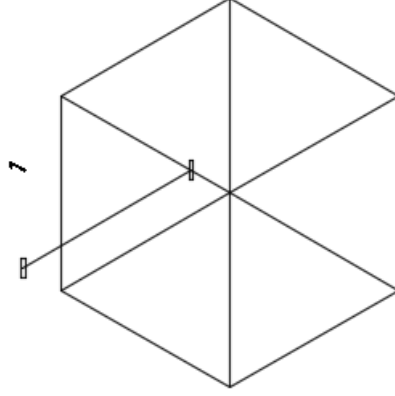


Figure 4.3: Cube: geometry of the problem.

From Figure 3.7, obtained using $p=q=r$ from 2 to 7 and 5 to 15 control points per direction, we point out that also, as far as the 3D cube isotropic linear-elastic case is concerned, we get convergence and from Table 4.2, we observe that the prescribed order of converge are maintained.

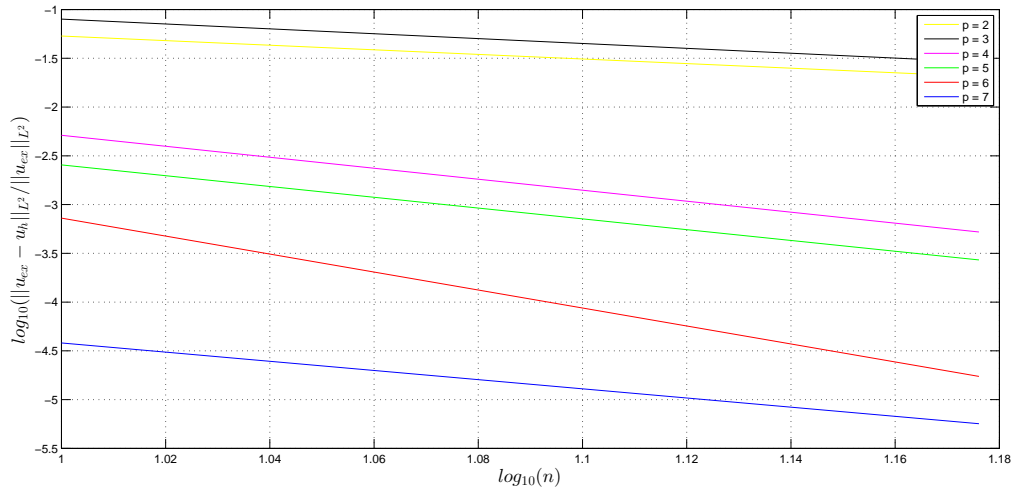


Figure 4.4: Cube convergence test: number of d.o.fs vs error in terms of L^2 -norm

We report below the Matlab code that provided the basis for the previous results, in order to logically explain the computational steps that brings to the displacement error evaluation.

- The script only requires the degree of approximation as input as well as the number of control

Table 4.2: Cube 3D linear elasticity test: comparison between imposed degree of approximation and obtained order of convergence considering x direction.

Degree of approximation	Order of convergence
2	2.36
3	2.50
4	5.63
5	5.53
6	9.22
7	4.71

points used along the three directions, as well as material parameters (Young modulus, E , and Poisson's coefficient, ν).

- The first operation consists of creating open knot vectors. After that h-refinement or knot insertion operation is performed.
- Greville abscissae are computed and we proceed to initialize the needed quantities.
- Only the active degrees of freedom are selected and therefore we exclude the boundary since everything is clamped.
- B-splines and their derivatives are computed.
- The following triple "for" loop involves spatially all the collocation points, computes volume forces and performs tensor product.
- All the necessary quantities are stored in an optimized way taking into account the constitutive relations. In fact in three dimensions three rows of the global sparse stiffness matrix and the residual force are considered point by point.
- The storage procedure seeks the non-zero entries vertically using the "find" command and is performed using an inner counter which is updated every time the loop is activated.
- The problem is then solved and the displacements are obtained.
- The displacement control variables need to be re-interpolated as a post-process operation into a new created spline.
- The error in terms of L^2 -norm is performed.

```

1 function error = IGAcoll_elLin3D(p,q,r,mcp,ncp,ocp,E,nu)
2 % 3D IGA collocation code
3 %
4 % Solve 3D linear elasticity problem with homogeneous Dirichlet b.c.'s
   and
5 % with a manufactured body load s.t. the analytical solution is
   available
6 %
7 % Input:  p    = approximation degree(xi)
8 %         q    = approximation degree(eta)
9 %         r    = approximation degree(zeta)
10 %        mcp   = number of collocation points(xi)
11 %        ncp   = number of collocation points(eta)
12 %        ocp   = number of collocation points(zeta)
13 %        E     = Young modulus
14 %        nu    = Poisson's ratio
15 %% Output: error = (approximation of the) L2-norm error
16 % set material parameters
17 mi = E/(2*(1 + nu));
18 lambda = E*nu/((1 + nu)*(1 - 2*nu));
19
20 % set mesh [linear parameterization, continuity C^(p-1)]
21 csi      = augknt([0 1],p+1); % knot csi direction
22 eta      = augknt([0,1],q+1); % knot eta direction
23 zeta     = augknt([0,1],r+1); % knot zeta direction
24 hxi      = 1/(mcp - p);
25 heta     = 1/(ncp - q);
26 hzeta    = 1/(ocp - r);
27 insxi    = hxi:hxi:1-hxi;
28 inseta   = heta:heta:1-heta;
29 inszeta  = hzeta:hzeta:1-hzeta;
30
31 [~,csi]   = bspkntins(p,0:1/p:1,csi,insxi); % [-,ik] = bspkntins(spline
   degree,c.p.,knot seq.,new knot)
32
33 [~,eta]   = bspkntins(q,0:1/q:1,eta,inseta);
34
35 [~,zeta]  = bspkntins(r,0:1/r:1,zeta,inszeta);
36
37 % compute Greville abscissae (collocation points)
38 collptxi = aveknt(csi,p+1);
39 collpteta = aveknt(eta,q+1);
40 collptzeta = aveknt(zeta,r+1);
41
42 % intializations
43 ndof = 3*mcp*ncp;
44 nbc = 3*(mcp-2)*(ncp-2)*(ocp-2);
45 nbci = nbc/3;
46 f_gl = zeros(nbc,1);
47 row = zeros(1,ndof^2);

```

```

48 col = zeros(1,ndof^2);
49 val = zeros(1,ndof^2);
50 k_nk = zeros(3,nbc);
51
52 % "loop" over active collocation points
53 ii = 2:mcp-1;
54 jj = 2:ncp-1;
55 kk = 2:ocp-1;
56
57 % set active collocation points
58 collptxi = collptxi(ii);
59 collpteta = collpteta(jj);
60 collptzeta = collptzeta(kk);
61
62 % compute basis functions and their derivatives at collocation pts
63 Nx = spcol(csi,p+1,sort([collptxi,collptxi,collptxi])); % N1(csi1) N2(
    csi1) N3(csi1);% N1'(csi1) N2'(csi1) N3'(csi1);% N1''(csi1) N2''(
    csi1) N3''(csi1)...
64 % N1(csi2) N2(csi2) N3(csi2);
65 Ny = spcol(eta,q+1,sort([collpteta,collpteta,collpteta]));
66 Nz = spcol(zeta,r+1,sort([collptzeta,collptzeta,collptzeta]));
67
68 % construct the stiffness matrix
69 kk = 0;
70 icount = 1;
71
72 for k = 1:ocp-2
73     for j = 1:ncp-2
74         for i = 1:mcp-2
75
76             a = 3*icount-2;
77             c = 3*icount;
78             f1 = sin(2*pi*collptxi(i))*sin(2*pi*collpteta(j))*sin(2*pi*collptzeta(
                k));
79             f2 = cos(2*pi*collptxi(i))*cos(2*pi*collpteta(j))*sin(2*pi*collptzeta(
                k));
80             f3 = cos(2*pi*collptxi(i))*cos(2*pi*collptzeta(k))*sin(2*pi*collpteta(
                j));
81             f4 = cos(2*pi*collpteta(j))*cos(2*pi*collptzeta(k))*sin(2*pi*collptxi(
                i));
82
83             b1 = 12*pi^2*mi*f1 - (lambda + mi)*(4*pi^2*f2 + 4*pi^2*f3 - 4*pi^2*f1)
                ;
84             b2 = 12*pi^2*mi*f1 - (lambda + mi)*(4*pi^2*f2 + 4*pi^2*f4 - 4*pi^2*f1)
                ;
85             b3 = 12*pi^2*mi*f1 - (lambda + mi)*(4*pi^2*f3 + 4*pi^2*f4 - 4*pi^2*f1)
                ;
86
87             f_gl(a:c) = -[b1;b2;b3];
88
89             d2Nxi = reshape(kron(Nx(3*i,2:mcp-1)'*Ny(3*j-2,2:ncp-1),Nz(3*k-2,2:ocp

```



```

-1)),1,nbci);
90 d2Neta = reshape(kron(Nx(3*i-2,2:mcp-1)'*Ny(3*j,2:ncp-1),Nz(3*k-2,2:
    ocp-1)),1,nbci);
91 d2Nzeta = reshape(kron(Nx(3*i-2,2:mcp-1)'*Ny(3*j-2,2:ncp-1),Nz(3*k,2:
    ocp-1)),1,nbci);
92 d2Nxieta = reshape(kron(Nx(3*i-1,2:mcp-1)'*Ny(3*j-1,2:ncp-1),Nz(3*k
    -2,2:ocp-1)),1,nbci);
93 d2Netazeta = reshape(kron(Nx(3*i-2,2:mcp-1)'*Ny(3*j-1,2:ncp-1),Nz(3*k
    -1,2:ocp-1)),1,nbci);
94 d2Nxizeta = reshape(kron(Nx(3*i-1,2:mcp-1)'*Ny(3*j-2,2:ncp-1),Nz(3*k
    -1,2:ocp-1)),1,nbci);
95 midivN = mi*(d2Nxi + d2Neta + d2Nzeta);
96
97 k_nk(1,1:3:end-2) = midivN + (lambda+mi)*d2Nxi;
98 k_nk(1,2:3:end-1) = (lambda+mi)*d2Nxieta;
99 k_nk(1,3:3:end) = (lambda+mi)*d2Nxizeta;
100
101 k_nk(2,1:3:end-2) = (lambda+mi)*d2Nxieta;
102 k_nk(2,2:3:end-1) = midivN + (lambda+mi)*d2Neta;
103 k_nk(2,3:3:end) = (lambda+mi)*d2Netazeta;
104
105 k_nk(3,1:3:end-2) = (lambda+mi)*d2Nxizeta;
106 k_nk(3,2:3:end-1) = (lambda+mi)*d2Netazeta;
107 k_nk(3,3:3:end) = midivN + (lambda+mi)*d2Nzeta;
108
109 [rowk,colk,walk] = find(k_nk);
110
111 l = length(walk);
112
113 row(kk+1:kk+l) = 3*icount-3+rowk;
114
115 col(kk+1:kk+l) = colk;
116 val(kk+1:kk+l) = walk;
117
118 kk = kk + l;
119 icount = icount + l;
120 end
121 end
122 end
123
124 row = row(1:kk);
125 col = col(1:kk);
126 val = val(1:kk);
127
128 % assemble stiffness
129 k_gl = sparse(row,col,val,nbc,nbc);
130 clear row col val
131
132 % solve linear system
133 sol = k_gl\f_gl;
134 clear k_gl f_gl

```

```

135
136 % approx relative L^2-norm error vs the manufactured analytical
    solution
137 coeffs1 = zeros(mcp,ncp,ocp);
138 coeffs1(2:(mcp-1),2:(ncp-1),2:(ocp-1)) = reshape(sol(1:3:end-2),(mcp
    -2),(ncp-2),(ocp-2));
139 bspl1 = spmak({csi,eta,zeta},coeffs1);
140
141 coeffs2 = zeros(mcp,ncp,ocp);
142 coeffs2(2:(mcp-1),2:(ncp-1),2:(ocp-1)) = reshape(sol(2:3:end-1),(mcp
    -2),(ncp-2),(ocp-2));
143 bspl2 = spmak({csi,eta,zeta},coeffs2);
144
145 coeffs3 = zeros(mcp,ncp,ocp);
146 coeffs3(2:(mcp-1),2:(ncp-1),2:(ocp-1)) = reshape(sol(3:3:end),(mcp-2)
    ,(ncp-2),(ocp-2));
147 bspl3 = spmak({csi,eta,zeta},coeffs3);
148
149 ptx = linspace(0,1,10*mcp);
150 pty = linspace(0,1,10*ncp);
151 ptz = linspace(0,1,10*ocp);
152
153 approx1 = fnval(bspl1,{ptx,pty,ptz});
154 approx2 = fnval(bspl2,{ptx,pty,ptz});
155 approx3 = fnval(bspl3,{ptx,pty,ptz});
156
157 [xx,yy,zz] = meshgrid(ptx,pty,ptz);
158 exact = sin(2*pi*xx).*sin(2*pi*yy).*sin(2*pi*zz);
159
160 error1 = sqrt(sum(sum(sum((exact-approx1).^2)))/sum(sum(sum(exact.^2))
    ));
161 error2 = sqrt(sum(sum(sum((exact-approx2).^2)))/sum(sum(sum(exact.^2))
    ));
162 error3 = sqrt(sum(sum(sum((exact-approx3).^2)))/sum(sum(sum(exact.^2))
    ));
163
164 error = [error1;error2;error3];
165 return

```

4.2.3 Mapping

Up to now we proposed examples which do not require mapping operation. To fully understand this non trivial change we move to 1D again to build the required map. We recall that the Jacobian in 1D is

$$J = x'(x) = \sum_i N'(\xi) B_i = \frac{dN}{d\xi} B_i \quad (4.14)$$

after the Jacobian is computed the real mapping requires its inversion since we know the derivatives of the shape functions with respect to ξ .

$$N'(\xi) = \frac{dN}{dx} = \frac{dN}{d\xi} \frac{d\xi}{dx} = \frac{dN}{d\xi} J^{-1} \quad (4.15)$$

This can be trivially extended to 3D. In the following, we report our Matlab implementation of IGA collocation to solve 3D elastic problems on mapped geometries. The main differences with respect to the previous code are:

- The code requires the target number of control points and degree of approximation per direction and provides already the minimum value for the cited quantities.
- From the basic information it constructs the geometry in terms of NURBS to have an idea of the problem and refinement is performed in order to reach the target degree of approximation.
- Since the boundary is clamped the actual degrees of freedom are found but put aside for postprocessing since all the B-Splines informations are required inside the triple "for" loop for the mapping. The latter prescribes several steps: NURBS are obtained from B-splines tensor product and put aside; local derivation and normalization is performed; the mapping goes from local to global as an end.

```

1 function error = nrb_IGAcoll_elLin3D(p,q,r,mcp,ncp,ocp,E,nu)
2 % 3D IGA collocation code
3 %
4 % Solve 3D linear elasticity problem with homogeneous Dirichlet b.c.'s
   and
5 % with a manufactured body load s.t. the analytical solution is
   available
6 %
7 % Input:  p    = approximation degree(csi)
8 %         q    = approximation degree(eta)
9 %         r    = approximation degree(zeta)
10 %        mcp   = number of collocation points(csi)
11 %        ncp   = number of collocation points(eta)
12 %        ocp   = number of collocation points(zeta)
13 %        E     = Young modulus
14 %        nu    = Poisson's ratio
15 %
16 % quarter annulus
17 mcp0 = 3;
18 ncp0 = 2;
19 ocp0 = 2;
20 p0 = 2;
21 q0 = 1;
22 r0 = 1;
23 w = ones(mcp0,ncp0,ocp0);
24
25 XX0 = zeros(mcp0,ncp0,ocp0);
26 X0 = [1      4;
27       1      4;
28       0      0];
29 XX0(:, :, 1) = X0;
30 XX0(:, :, 2) = X0;
31
32 YY0 = zeros(mcp0,ncp0,ocp0);
33 Y0 = [0      0;
34       1      4;
35       1      4];
36 YY0(:, :, 1) = Y0;
37 YY0(:, :, 2) = Y0;
38
39 ZZ0 = zeros(mcp0,ncp0,ocp0);
40 ZZ0(:, :, 2) = ones(mcp0,ncp0);
41
42 w(2, :, :) = sqrt(2)/2;
43
44 csi = [0 0 0 1 1 1];
45 eta = [0 0 1 1];
46 zeta = [0 0 1 1];
47
48 BB = zeros(4,mcp0,ncp0,ocp0);

```

```

49 BB(1, :, :, :) = XX0.*w;
50 BB(2, :, :, :) = YY0.*w;
51 BB(3, :, :, :) = ZZ0.*w;
52 BB(4, :, :, :) = w;
53 nurbs = nrbmak(BB,{csi eta zeta});
54
55 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
56 % ptu = linspace(0,1,10*mcp0);
57 % ptv = linspace(0,1,10*ncp0);
58 % ptw = linspace(0,1,10*ocp0);
59 % pnts = nrbeval(nurbs,{ptu,ptv,ptw});
60 % xx = squeeze(pnts(1, :, :));
61 % yy = squeeze(pnts(2, :, :));
62 % zz = squeeze(pnts(3, :, :));
63 % figure, surf(xx,yy,zz), view(3)
64 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
65
66 % refinement
67 inc_p = p - p0;
68 inc_q = q - q0;
69 inc_r = r - r0;
70 nurbs = nrbdegelev(nurbs,[inc_p inc_q inc_r]);
71 h1 = 1/(ncp - p);
72 h2 = 1/(mcp - q);
73 h3 = 1/(ocp - r);
74 nurbs = nrbkntins(nurbs,{h1:h1:1-h1, h2:h2:1-h2, h3:h3:1-h3});
75 mcp = nurbs.number(1);
76 ncp = nurbs.number(2);
77 ocp = nurbs.number(3);
78 % p = nurbs.order(1)-1;
79 % q = nurbs.order(2)-1;
80 % r = nurbs.order(3)-1;
81 nnod = mcp*ncp*ocp;
82 csi = nurbs.knots{1};
83 eta = nurbs.knots{2};
84 zeta = nurbs.knots{3};
85 X = squeeze(nurbs.coefs(1, :, :, :)./nurbs.coefs(4, :, :, :));
86 Y = squeeze(nurbs.coefs(2, :, :, :)./nurbs.coefs(4, :, :, :));
87 Z = squeeze(nurbs.coefs(3, :, :, :)./nurbs.coefs(4, :, :, :));
88 w = squeeze(nurbs.coefs(4, :, :, :));
89
90 % compute Greville abscissae (collocation points)
91 collpt_csi = aveknt(csi,p+1);
92 collpt_eta = aveknt(eta,q+1);
93 collpt_zeta = aveknt(zeta,r+1);
94
95 % compute basis functions and their derivatives at collocation pts
96 NN = spcol(csi,p+1,sort([collpt_csi,collpt_csi,collpt_csi]));
97 MM = spcol(eta,q+1,sort([collpt_eta,collpt_eta,collpt_eta]));
98 RR = spcol(zeta,r+1,sort([collpt_zeta,collpt_zeta,collpt_zeta]));
99

```

```

100 % set material parameters
101 mi = E/(2*(1 + nu));
102 lambda = E*nu/((1 + nu)*(1 - 2*nu));
103
104 % initializations
105 ndof = 3*nnod;
106 f_gl = zeros(ndof,1);
107 est = p^2*ncp^5;
108 row = zeros(1,est); % zeros(1,ndof^2);
109 col = zeros(1,est); % zeros(1,ndof^2);
110 val = zeros(1,est); % zeros(1,ndof^2);
111 k_nk = zeros(3,ndof);
112 IJK = zeros(3,nnod);
113 NNN = zeros(nnod,nnod,3);
114
115 % set active d.o.f.s
116 A = zeros(mcp,ncp,ocp);
117 A(2:end-1,2:end-1,2:end-1) = 1;
118 A = reshape(A,1,nnod);
119 act_nodes = find(A);
120 act_dofs = sort([3*act_nodes-2,3*act_nodes-1,3*act_nodes]);
121
122 % construct the stiffness matrix
123 kk = 0;
124 icount = 1;
125 x = reshape(X,nnod,1);
126 y = reshape(Y,nnod,1);
127 z = reshape(Z,nnod,1);
128
129 for k = 1:ocp
130     for j = 1:ncp
131         for i = 1:mcp
132             %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
133             % mapping
134             %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
135             Nx = [NN(3*i-2,:);NN(3*i-1,:);NN(3*i,:)]; % N,N',N''
136             Ny = [MM(3*j-2,:);MM(3*j-1,:);MM(3*j,:)];
137             Nz = [RR(3*k-2,:);RR(3*k-1,:);RR(3*k,:)];
138
139             N = tens_prod(Nx(1,:),Ny(1,:),Nz(1,:),w);
140             den_sum = sum(N);
141
142             [dN,ddN] = local_derive(Nx,Ny,Nz,w);
143             [dN,ddN] = normalize_shapefun(N,den_sum,dN,ddN);
144             N = N/den_sum;
145             XYZ = [y,x,z]';
146             [~,ddN] = local_to_global(XYZ,dN,ddN);
147             %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
148             N = N';
149             ddN = ddN';
150             ddNxi2 = ddN(1,:);

```

```

151 ddNeta2 = ddN(2,:);
152 ddNzeta2 = ddN(3,:);
153 ddNxieta = ddN(4,:);
154 ddNetazeta = ddN(5,:);
155 ddNxizeta = ddN(6,:);
156
157 x
158 y
159 pti = N*x
160 ptj = N*y
161 ptk = N*z;
162
163 b1 = - mi*(8*pti^2*sin(2*pi*ptk)*sin(pti)*sin(ptj) + 8*ptj^2*sin(2*pi*
    ptk)*sin(pti)*sin(ptj) + 4*sin(2*pi*ptk)*sin(pti)*sin(ptj)*(pti^2 +
    ptj^2 - 1) + 4*sin(2*pi*ptk)*sin(pti)*sin(ptj)*(pti^2 + ptj^2 -
    16) + 4*pti*sin(2*pi*ptk)*cos(pti)*sin(ptj)*(pti^2 + ptj^2 - 1) +
    4*pti*sin(2*pi*ptk)*cos(pti)*sin(ptj)*(pti^2 + ptj^2 - 16) + 4*ptj*
    sin(2*pi*ptk)*cos(ptj)*sin(pti)*(pti^2 + ptj^2 - 1) + 4*ptj*sin(2*
    pi*ptk)*cos(ptj)*sin(pti)*(pti^2 + ptj^2 - 16) - 2*sin(2*pi*ptk)*
    sin(pti)*sin(ptj)*(pti^2 + ptj^2 - 1)*(pti^2 + ptj^2 - 16) - 4*pi
    ^2*sin(2*pi*ptk)*sin(pti)*sin(ptj)*(pti^2 + ptj^2 - 1)*(pti^2 + ptj
    ^2 - 16)) - (lambda + mi)*(8*pti^2*sin(2*pi*ptk)*sin(pti)*sin(ptj)
    + 2*sin(2*pi*ptk)*sin(pti)*sin(ptj)*(pti^2 + ptj^2 - 1) + 2*sin(2*
    pi*ptk)*sin(pti)*sin(ptj)*(pti^2 + ptj^2 - 16) + 4*pti*sin(2*pi*ptk)
    *cos(pti)*sin(ptj)*(pti^2 + ptj^2 - 1) + 2*pti*sin(2*pi*ptk)*cos(
    ptj)*sin(pti)*(pti^2 + ptj^2 - 1) + 4*pti*sin(2*pi*ptk)*cos(pti)*
    sin(ptj)*(pti^2 + ptj^2 - 16) + 2*pti*sin(2*pi*ptk)*cos(ptj)*sin(
    pti)*(pti^2 + ptj^2 - 16) + 2*ptj*sin(2*pi*ptk)*cos(pti)*sin(ptj)*(
    pti^2 + ptj^2 - 1) + 2*ptj*sin(2*pi*ptk)*cos(pti)*sin(ptj)*(pti^2 +
    ptj^2 - 16) + sin(2*pi*ptk)*cos(pti)*cos(ptj)*(pti^2 + ptj^2 - 1)
    *(pti^2 + ptj^2 - 16) + 8*pti*ptj*sin(2*pi*ptk)*sin(pti)*sin(ptj) -
    sin(2*pi*ptk)*sin(pti)*sin(ptj)*(pti^2 + ptj^2 - 1)*(pti^2 + ptj^2
    - 16) + 4*pi*pti*cos(2*pi*ptk)*sin(pti)*sin(ptj)*(pti^2 + ptj^2 -
    1) + 4*pi*pti*cos(2*pi*ptk)*sin(pti)*sin(ptj)*(pti^2 + ptj^2 - 16)
    + 2*pi*cos(2*pi*ptk)*cos(pti)*sin(ptj)*(pti^2 + ptj^2 - 1)*(pti^2 +
    ptj^2 - 16));
164
165
166 b2 = - mi*(8*pti^2*sin(2*pi*ptk)*sin(pti)*sin(ptj) + 8*ptj^2*sin(2*pi*
    ptk)*sin(pti)*sin(ptj) + 4*sin(2*pi*ptk)*sin(pti)*sin(ptj)*(pti^2 +
    ptj^2 - 1) + 4*sin(2*pi*ptk)*sin(pti)*sin(ptj)*(pti^2 + ptj^2 -
    16) + 4*pti*sin(2*pi*ptk)*cos(pti)*sin(ptj)*(pti^2 + ptj^2 - 1) +
    4*pti*sin(2*pi*ptk)*cos(pti)*sin(ptj)*(pti^2 + ptj^2 - 16) + 4*ptj*
    sin(2*pi*ptk)*cos(ptj)*sin(pti)*(pti^2 + ptj^2 - 1) + 4*ptj*sin(2*
    pi*ptk)*cos(ptj)*sin(pti)*(pti^2 + ptj^2 - 16) - 2*sin(2*pi*ptk)*
    sin(pti)*sin(ptj)*(pti^2 + ptj^2 - 1)*(pti^2 + ptj^2 - 16) - 4*pi
    ^2*sin(2*pi*ptk)*sin(pti)*sin(ptj)*(pti^2 + ptj^2 - 1)*(pti^2 + ptj
    ^2 - 16)) - (lambda + mi)*(8*ptj^2*sin(2*pi*ptk)*sin(pti)*sin(ptj)
    + 2*sin(2*pi*ptk)*sin(pti)*sin(ptj)*(pti^2 + ptj^2 - 1) + 2*sin(2*
    pi*ptk)*sin(pti)*sin(ptj)*(pti^2 + ptj^2 - 16) + 2*pti*sin(2*pi*ptk)
    *cos(ptj)*sin(pti)*(pti^2 + ptj^2 - 1) + 2*pti*sin(2*pi*ptk)*cos(

```

```

    ptj)*sin(pti)*(pti^2 + ptj^2 - 16) + 2*ptj*sin(2*pi*ptk)*cos(pti)*
    sin(ptj)*(pti^2 + ptj^2 - 1) + 4*ptj*sin(2*pi*ptk)*cos(ptj)*sin(pti)
    *(pti^2 + ptj^2 - 1) + 2*ptj*sin(2*pi*ptk)*cos(pti)*sin(ptj)*(pti
    ^2 + ptj^2 - 16) + 4*ptj*sin(2*pi*ptk)*cos(ptj)*sin(pti)*(pti^2 +
    ptj^2 - 16) + sin(2*pi*ptk)*cos(pti)*cos(ptj)*(pti^2 + ptj^2 - 1)*
    (pti^2 + ptj^2 - 16) + 8*pti*ptj*sin(2*pi*ptk)*sin(pti)*sin(ptj) -
    sin(2*pi*ptk)*sin(pti)*sin(ptj)*(pti^2 + ptj^2 - 1)*(pti^2 + ptj^2
    - 16) + 4*pi*ptj*cos(2*pi*ptk)*sin(pti)*sin(ptj)*(pti^2 + ptj^2 -
    1) + 4*pi*ptj*cos(2*pi*ptk)*sin(pti)*sin(ptj)*(pti^2 + ptj^2 - 16)
    + 2*pi*cos(2*pi*ptk)*cos(ptj)*sin(pti)*(pti^2 + ptj^2 - 1)*(pti^2 +
    ptj^2 - 16));
167
168
169 b3 = - mi*(8*pti^2*sin(2*pi*ptk)*sin(pti)*sin(ptj) + 8*ptj^2*sin(2*pi*
    ptk)*sin(pti)*sin(ptj) + 4*sin(2*pi*ptk)*sin(pti)*sin(ptj)*(pti^2 +
    ptj^2 - 1) + 4*sin(2*pi*ptk)*sin(pti)*sin(ptj)*(pti^2 + ptj^2 -
    16) + 4*pti*sin(2*pi*ptk)*cos(pti)*sin(ptj)*(pti^2 + ptj^2 - 1) +
    4*pti*sin(2*pi*ptk)*cos(ptj)*sin(pti)*(pti^2 + ptj^2 - 16) + 4*ptj*
    sin(2*pi*ptk)*cos(ptj)*sin(pti)*(pti^2 + ptj^2 - 1) + 4*ptj*sin(2*
    pi*ptk)*cos(pti)*sin(ptj)*(pti^2 + ptj^2 - 16) - 2*sin(2*pi*ptk)*
    sin(pti)*sin(ptj)*(pti^2 + ptj^2 - 1)*(pti^2 + ptj^2 - 16) - 4*pi
    ^2*sin(2*pi*ptk)*sin(pti)*sin(ptj)*(pti^2 + ptj^2 - 1)*(pti^2 + ptj
    ^2 - 16)) - (lambda + mi)*(4*pi*pti*cos(2*pi*ptk)*sin(pti)*sin(ptj)
    *(pti^2 + ptj^2 - 1) - 4*pi^2*sin(2*pi*ptk)*sin(pti)*sin(ptj)*(pti
    ^2 + ptj^2 - 1)*(pti^2 + ptj^2 - 16) + 4*pi*pti*cos(2*pi*ptk)*sin(
    pti)*sin(ptj)*(pti^2 + ptj^2 - 16) + 4*pi*ptj*cos(2*pi*ptk)*sin(pti)
    *sin(ptj)*(pti^2 + ptj^2 - 1) + 4*pi*ptj*cos(2*pi*ptk)*sin(pti)*
    sin(ptj)*(pti^2 + ptj^2 - 16) + 2*pi*cos(2*pi*ptk)*cos(pti)*sin(ptj)
    *(pti^2 + ptj^2 - 1)*(pti^2 + ptj^2 - 16) + 2*pi*cos(2*pi*ptk)*cos
    (ptj)*sin(pti)*(pti^2 + ptj^2 - 1)*(pti^2 + ptj^2 - 16));
170
171 c = 3*icount;
172 a = c-2;
173
174 f_gl(a:c) = -[b1;b2;b3];
175 IJK(:,icount) = [pti;ptj;ptk];
176 NNN(icount,:,1) = N;
177 NNN(icount,:,2) = N;
178 NNN(icount,:,3) = N;
179
180 midivN = mi*(ddNxi2 + ddNeta2 + ddNzeta2);
181 k_nk(1,1:3:end-2) = midivN + (lambda+mi)*ddNxi2;
182 k_nk(1,2:3:end-1) = (lambda+mi)*ddNxieta;
183 k_nk(1,3:3:end) = (lambda+mi)*ddNxizeta;
184
185 k_nk(2,1:3:end-2) = (lambda+mi)*ddNxieta;
186 k_nk(2,2:3:end-1) = midivN + (lambda+mi)*ddNeta2;
187 k_nk(2,3:3:end) = (lambda+mi)*ddNetazeta;
188
189 k_nk(3,1:3:end-2) = (lambda+mi)*ddNxizeta;
190 k_nk(3,2:3:end-1) = (lambda+mi)*ddNetazeta;

```



```

191 k_nk(3,3:3:end) = midivN + (lambda+mi)*ddNzeta2;
192
193 [rowk, colk, valk] = find(k_nk);
194
195 l = length(valk);
196
197 row(kk+1:kk+l) = 3*icount-3+rowk;
198 col(kk+1:kk+l) = colk;
199 val(kk+1:kk+l) = valk;
200
201 kk = kk + l;
202 icount = icount + 1;
203 end
204 end
205 end
206
207 row = row(1:kk);
208 col = col(1:kk);
209 val = val(1:kk);
210
211 % assemble stiffness
212 k_gl = sparse(row, col, val, ndof, ndof);
213 clear row col val
214
215 % solve linear system
216 sol = k_gl(act_dofs, act_dofs)\f_gl(act_dofs);
217 clear k_gl f_gl
218
219 % valutazione + raffinata
220 ptu = linspace(0,1,2*mcp);
221 ptv = linspace(0,1,2*ncp);
222 ptw = linspace(0,1,2*ocp);
223 pnts = nrbeval(nurbs, {ptu, ptv, ptw});
224 x = squeeze(pnts(1, :, :, :));
225 y = squeeze(pnts(2, :, :, :));
226 z = squeeze(pnts(3, :, :, :));
227
228 uu = zeros(mcp, ncp, ocp);
229 uu(2:mcp-1, 2:ncp-1, 2:ocp-1) = reshape(sol(1:3:end-2), mcp-2, ncp-2, ocp-2);
230 vv = zeros(mcp, ncp, ocp);
231 vv(2:mcp-1, 2:ncp-1, 2:ocp-1) = reshape(sol(2:3:end-1), mcp-2, ncp-2, ocp-2);
232 ww = zeros(mcp, ncp, ocp);
233 ww(2:mcp-1, 2:ncp-1, 2:ocp-1) = reshape(sol(3:3:end), mcp-2, ncp-2, ocp-2);
234 BB = zeros(4, mcp, ncp, ocp);
235 BB(1, :, :, :) = uu.*w;
236 BB(2, :, :, :) = vv.*w;
237 BB(3, :, :, :) = ww.*w;
238 BB(4, :, :, :) = w;
239 nurbs_u = nrbmak(BB, {csi eta zeta});

```

```

240 pnts_u = nrbeval(nurbs_u,{ptu,ptv,ptw});
241 u_val= squeeze(pnts_u(1,:,:,:));
242 v_val = squeeze(pnts_u(2,:,:,:));
243 w_val = squeeze(pnts_u(3,:,:,:));
244
245 exact_u = (x.^2+y.^2-1).*(x.^2+y.^2-16).*sin(x).*sin(y).*sin(2*pi*z);
246 exact_v = exact_u;
247 exact_w = exact_u;
248
249 error1 = sqrt(sum(sum(sum((exact_u - u_val).^2)))/sum(sum(sum(exact_u
    .^2))));
250 error2 = sqrt(sum(sum(sum((exact_v - v_val).^2)))/sum(sum(sum(exact_v
    .^2))));
251 error3 = sqrt(sum(sum(sum((exact_w - w_val).^2)))/sum(sum(sum(exact_w
    .^2))));
252
253 error = [error1;error2;error3];
254
255
256 %% subroutines
257 function M=tens_prod(A,B,C,w)
258
259 AB = A'*B;
260 M = AB(:)*C;
261 M = M(:).*w(:);
262
263 end
264 function [dN,ddN] = local_derive(Nx,Ny,Nz,w)
265
266 dN(:,1) = tens_prod(Nx(2,:),Ny(1,:),Nz(1,:),w); % dNMP/d_xi
267 %
268 dN(:,2) = tens_prod(Nx(1,:),Ny(2,:),Nz(1,:),w); % dNMP/d_eta
269 %
270 dN(:,3) = tens_prod(Nx(1,:),Ny(1,:),Nz(2,:),w); % dNMP/d_zeta
271 %
272 ddN(:,1)= tens_prod(Nx(3,:),Ny(1,:),Nz(1,:),w); % d2NMP/d_xi2
273 %
274 ddN(:,2)= tens_prod(Nx(1,:),Ny(3,:),Nz(1,:),w); % d2NMP/d_eta2
275 %
276 ddN(:,3)= tens_prod(Nx(1,:),Ny(1,:),Nz(3,:),w); % d2NMP/d_zeta2
277 %
278 ddN(:,4)= tens_prod(Nx(2,:),Ny(2,:),Nz(1,:),w); % d2NMP/d_xi_d_eta
279 %
280 ddN(:,5)= tens_prod(Nx(1,:),Ny(2,:),Nz(2,:),w); % d2NMP/d_eta_d_zeta
281 %
282 ddN(:,6)= tens_prod(Nx(2,:),Ny(1,:),Nz(2,:),w); % d2NMP/d_xi_d_zeta
283
284 end
285 function [dN,ddN] = normalize_shapefun(N,den_sum,dN,ddN)
286 der_sumx = sum(dN(:,1)); % dNMP/d_xi
287 der_sumy = sum(dN(:,2)); % dNMP/d_eta

```

```

288 der_sumz = sum(dN(:,3)); % dNMP/d_zeta
289
290 der2_sumx = sum(ddN(:,1)); % d2NMP/d_xi2
291 der2_sumy = sum(ddN(:,2)); % d2NMP/d_eta2
292 der2_sumz = sum(ddN(:,3)); % d2NMP/d_zeta2
293 der2_sumxy = sum(ddN(:,4)); % d2NMP/d_xi_d_eta
294 der2_sumyz = sum(ddN(:,5)); % d2NMP/d_eta_d_zeta
295 der2_sumxz = sum(ddN(:,6)); % d2NMP/d_xi_d_zeta
296
297 % this is actually R, the NURBS basis function: OK
298 ddN(:,1) = ddN(:,1)/den_sum - ... %d2R/d_xi2
299 (2*dN(:,1)*der_sumx + N*der2_sumx)/den_sum^2 + 2*N*der_sumx^2/den_sum
300 ^3;
301 ddN(:,2) = ddN(:,2)/den_sum - ... %d2R/d_eta2
302 (2*dN(:,2)*der_sumy + N*der2_sumy)/den_sum^2 + 2*N*der_sumy^2/den_sum
303 ^3;
304 ddN(:,3) = ddN(:,3)/den_sum - ... %d2R/d_zeta2
305 (2*dN(:,3)*der_sumz + N*der2_sumz)/den_sum^2 + 2*N*der_sumz^2/den_sum
306 ^3;
307 ddN(:,4) = ddN(:,4)/den_sum - (dN(:,1)*der_sumy + ... %d2R/d_xi_d_eta
308 dN(:,2)*der_sumx + N*der2_sumxy)/den_sum^2 + 2*N*der_sumx*der_sumy/
309 den_sum^3;
310 ddN(:,5) = ddN(:,5)/den_sum - (dN(:,2)*der_sumz + ... %d2R/
311 d_eta_d_zeta
312 dN(:,3)*der_sumy + N*der2_sumyz)/den_sum^2 + 2*N*der_sumy*der_sumz/
313 den_sum^3;
314 ddN(:,6) = ddN(:,6)/den_sum - (dN(:,3)*der_sumx + ... d2R/d_xi_d_zeta
315 dN(:,1)*der_sumz + N*der2_sumxz)/den_sum^2 + 2*N*der_sumz*der_sumx/
316 den_sum^3;
317
318 dN(:,1) = dN(:,1)/den_sum - N*der_sumx/den_sum^2; %dR/d_xi
319 dN(:,2) = dN(:,2)/den_sum - N*der_sumy/den_sum^2; %dR/d_eta
320 dN(:,3) = dN(:,3)/den_sum - N*der_sumz/den_sum^2; %dR/d_zeta
321
322 end
323 %%
324 function [dN,ddN] = local_to_global(XYZ,dN,ddN)
325
326 D = (XYZ*dN)';
327
328 Dinv = inv(D);
329 dN = dN*Dinv'; %first order global
330 % derivatives
331
332 D2 = (XYZ*ddN)';
333
334 RHS = ddN - dN*D2';
335
336 dxddxi2 = [D(1,1)^2 D(1,2)^2 D(1,3)^2 2*D(1,1)*D(1,2)

```

```

331      2*D(1,2)*D(1,3)      2*D(1,1)*D(1,3);
D(2,1)^2      D(2,2)^2      D(2,3)^2      2*D(2,1)*D(2,2)
      2*D(2,2)*D(2,3)      2*D(2,1)*D(2,3);
332 D(3,1)^2      D(3,2)^2      D(3,3)^2      2*D(3,1)*D(3,2)
      2*D(3,2)*D(3,3)      2*D(3,1)*D(3,3);
333 D(1,1)*D(2,1) D(1,2)*D(2,2) D(1,3)*D(2,3) D(1,1)*D(2,2)+D(2,1)*D(1,2)
      D(1,2)*D(2,3)+D(2,2)*D(1,3) D(1,1)*D(2,3)+D(2,1)*D(1,3);
334 D(2,1)*D(3,1) D(2,2)*D(3,2) D(2,3)*D(3,3) D(2,1)*D(3,2)+D(3,1)*D(2,2)
      D(2,2)*D(3,3)+D(3,2)*D(2,3) D(2,1)*D(3,3)+D(3,1)*D(2,3);
335 D(1,1)*D(3,1) D(1,2)*D(3,2) D(1,3)*D(3,3) D(1,1)*D(3,2)+D(3,1)*D(1,2)
      D(1,2)*D(3,3)+D(3,2)*D(1,3) D(1,1)*D(3,3)+D(3,1)*D(1,3) |;
336 %
337
338 dx dxi2 inv = inv(dx dxi2);
339 ddN          = RHS*dx dxi2 inv';
340
341 end
342 end

```

4.2.3.1 2D Clamped quarter of annulus

The proposed problem is a bidimensional quarter of annulus clamped on the boundary, where $R_1 = 1$ and $R_2 = 4$. The geometry of the problem can be better recovered in the Figure 4.5.

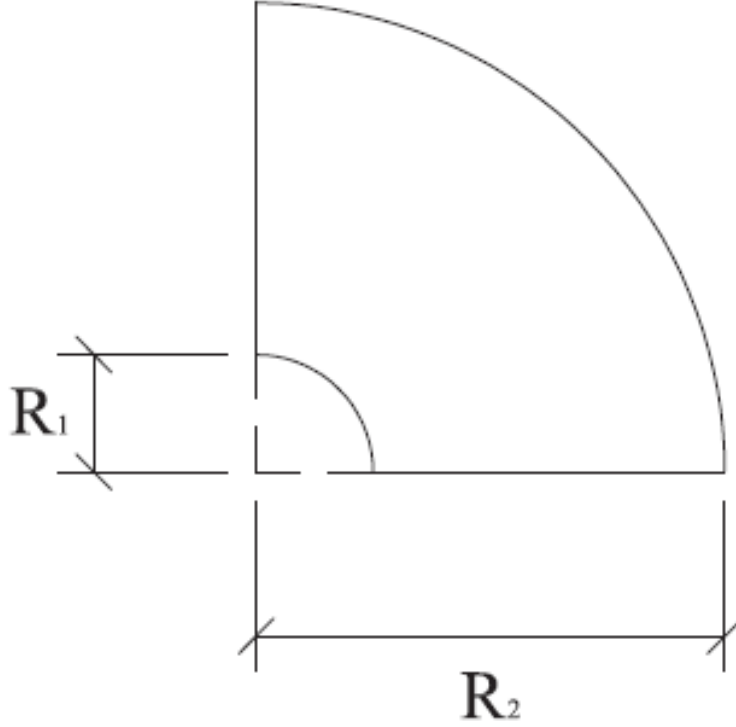


Figure 4.5: Clamped quarter of an annulus. Geometry of the problem.

The exact solution to this problem is here reported:

$$u(x, y) = v(x, y) = (x^2 + y^2 - 1)(x^2 + y^2 - 16)\sin(x)\sin(y) \quad (4.16)$$

Consequently we show in Figures 4.6 and 4.9 the exact solution, respectively in x and y directions and we compare it with the approximate one in Figures 4.7 and 4.10. We stress that both solutions respect the prescribed boundary conditions. We also plot the difference of the two considered solution in Figures 4.8 and 4.11 for each direction, showing that with $p = q = 2$ and 15 collocation points per direction a good level of approximation is obtained.

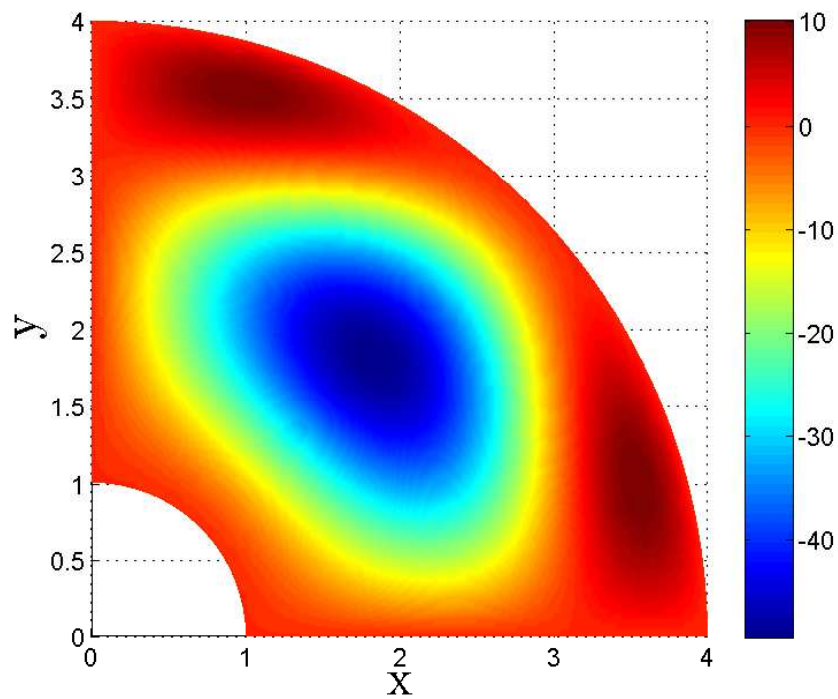


Figure 4.6: 2D Clamped quarter of an annulus: exact displacement in x direction.

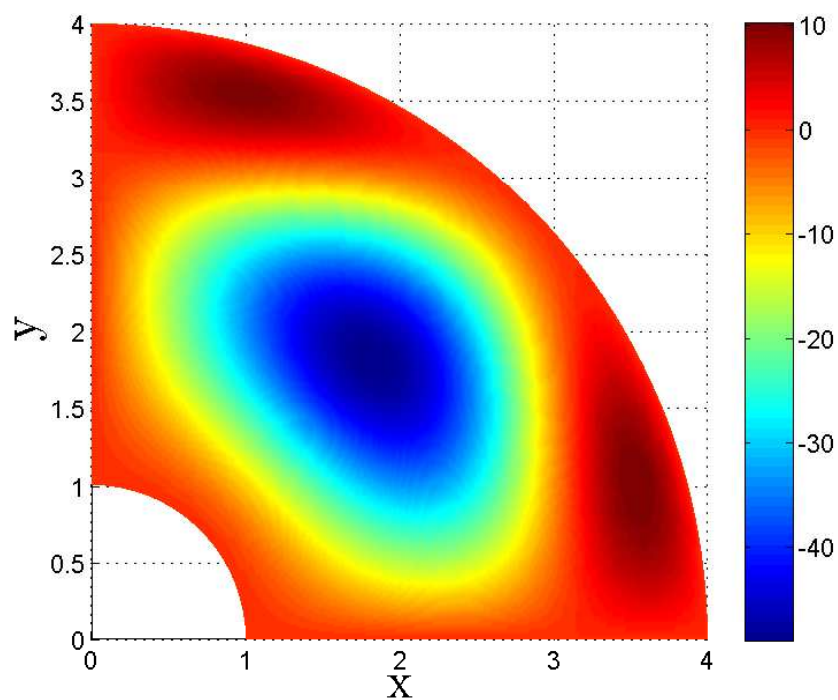


Figure 4.7: 2D Clamped quarter of an annulus: approximate displacement in x direction.

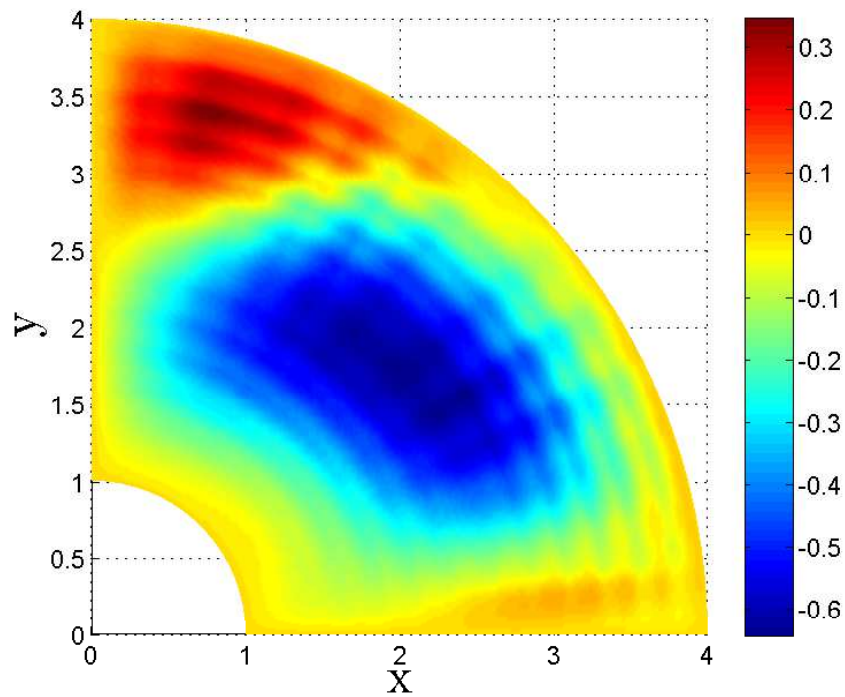


Figure 4.8: 2D Clamped quarter of an annulus: displacements difference between exact and approximate solution in x direction.

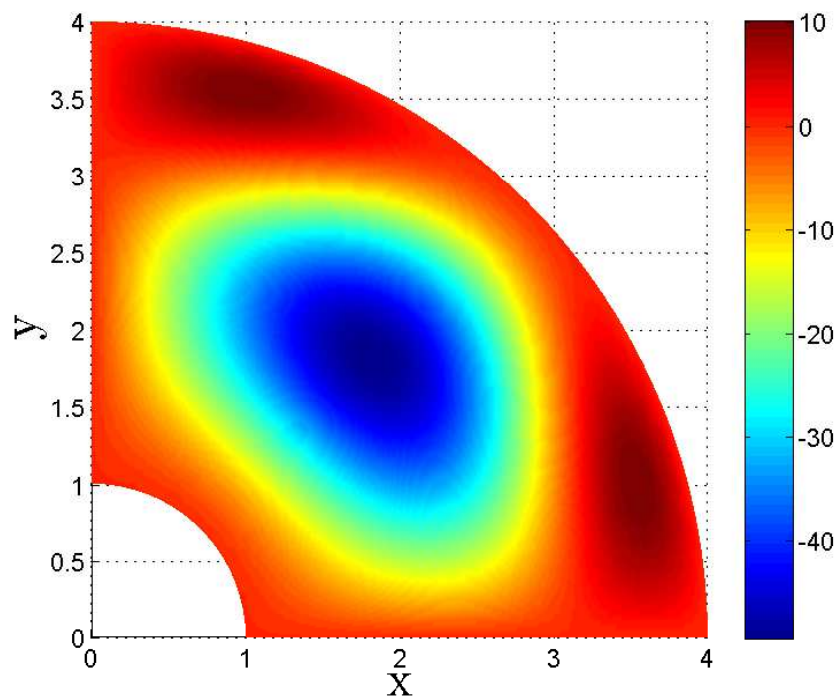


Figure 4.9: 2D Clamped quarter of an annulus: exact displacement in y direction.

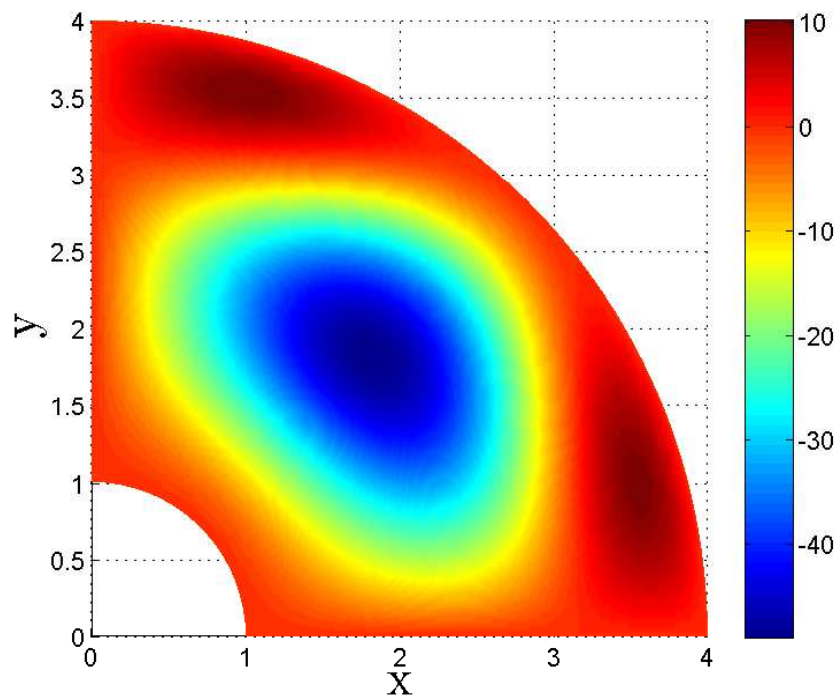


Figure 4.10: 2D Clamped quarter of an annulus: approximate displacement in y direction.

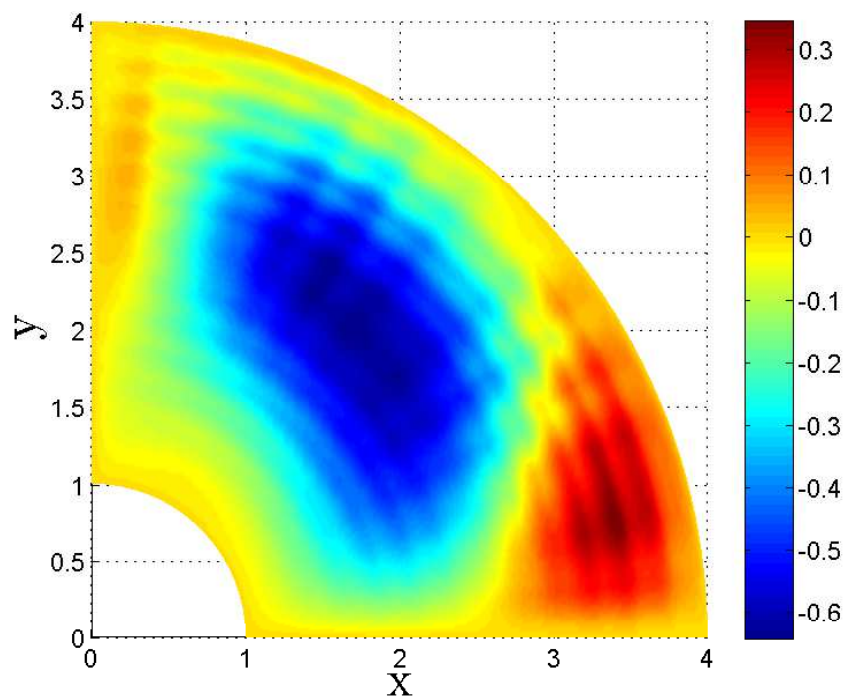


Figure 4.11: 2D Clamped quarter of an annulus: displacements difference between exact and approximate solution in y direction.

Here we report the results in terms of convergence, which correspond to what expected also for the 2D mapped benchmark. We also display Figure 4.12, that reports the number of degrees of freedom compared to the displacement error in terms of L^2 -norm in x direction, which proves the convergence rates, previously discussed, computed calculating the slope starting from last two numerical values, shown in the graphic. In order to present Figure 4.12 we considered degrees of approximation from 2 to 7 and 20 to 130 control points for each direction.

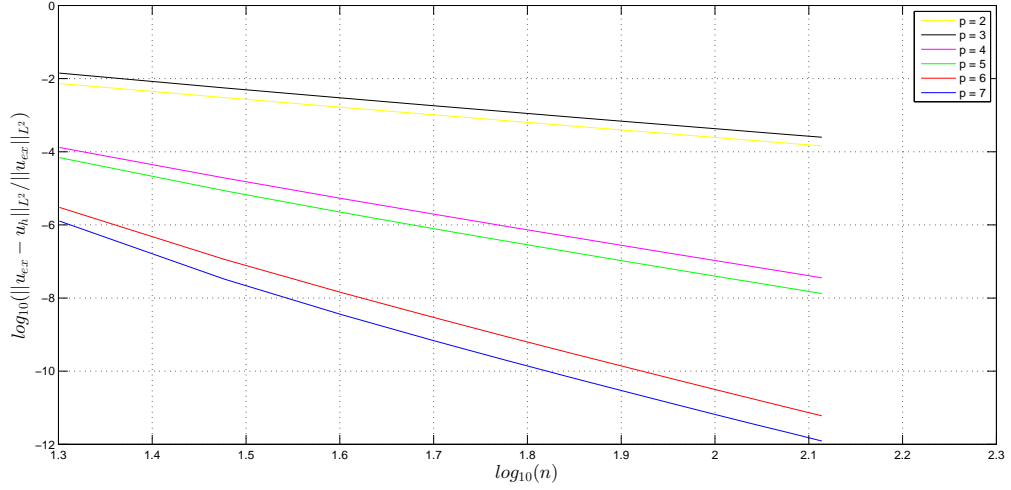


Figure 4.12: 2D Quarter of annulus convergence test: number of d.o.fs vs error in terms of L^2 -norm, considering x direction.

The obtained convergence rates for this case are reported in Table 4.3.

Table 4.3: 2D Quarter of annulus: comparison between imposed degree of approximation and obtained order of convergence considering x direction.

Degree of approximation	Order of convergence
2	2.03
3	2.05
4	4.12
5	4.17
6	6.27
7	6.41

4.2.3.2 3D Clamped quarter of annulus

We also consider a 3D quarter of annulus case. The geometry of this problem can be detected in Figure 4.13. In particular $R_i = 1$ and $R_e = 4$.

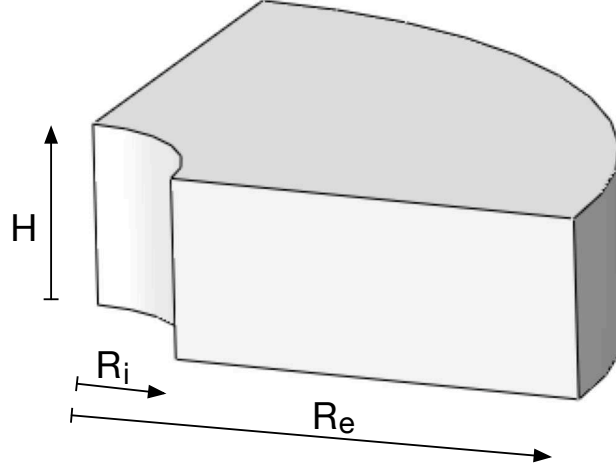


Figure 4.13: 3D Quarter of annulus: geometric definition of the problem.

The exact solution for this problem is here reported:

$$u(x, y) = v(x, y) = w(x, y, z) = (x^2 + y^2 - 1)(x^2 + y^2 - 16)\sin(x)\sin(y)\sin(2\pi z) \quad (4.17)$$

We can easily verify, analytically, that the presented exact solution respects the boundary conditions, since it's zero on the domain limits. As a conclusion we report the convergency test run for this 3D case (see Figure 4.14), obtained testing all approximation degrees from 2 to 7 and considering 5 to 15 control points per directions. The presented results prove to respect the prescribed rates of convergence as Table 4.4 demonstrates.

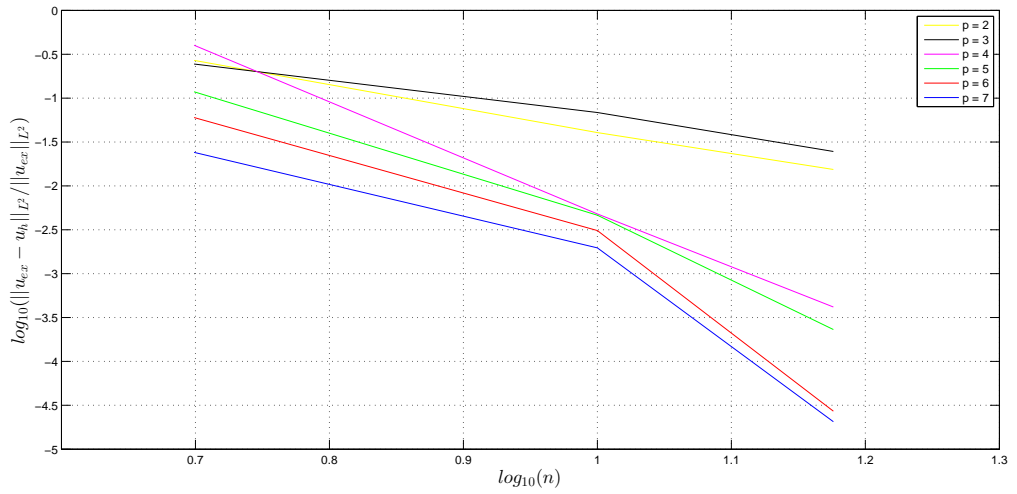


Figure 4.14: 3D quarter or annulus convergence test: number of d.o.fs vs error in terms of L^2 -norm in x direction.

Table 4.4: 3D Quarter of annulus: comparison between imposed degree of approximation and obtained order of convergence considering x direction.

Degree of approximation	Order of convergence
2	2.31
3	2.47
4	5.61
5	6.3
6	11
7	11

Chapter 5

Neumann Boundary Conditions

In this chapter we want be more specific regarding the imposition of Neumann boundary conditions. For this purpose in the first section we stress their analytical form in engineering notation and discuss their implementation. After that we show two tests relevant from an engineering point of view.

5.1 Neumann boundary conditions imposition

Accordingly with Chapter 4 we recall Neumann boundary conditions in terms of displacements:

$$\mu(u_{i,j} + u_{j,i})n_j + \lambda u_{j,j} = t_i \quad \text{on } \partial\Omega_N \quad (5.1)$$

We therefore proceed to approximate the displacements with the following NURBS shape functions

$$\mathbf{u}(\xi, \eta, \zeta) = \sum_{i,j,k} R_{i,j,k}^{p,q}(\xi, \eta, \zeta) \hat{\mathbf{u}}_{i,j,k} \quad (5.2)$$

or in compact form

$$\mathbf{u} = \mathbf{R}\hat{\mathbf{u}} \quad (5.3)$$

inserting (5.3) into (5.1) we obtain

$$\begin{bmatrix} \lambda \mathbf{R}_{1,1}n_1 + \mu(2\mathbf{R}_{1,1}n_1 + \mathbf{R}_{1,2}n_2 + \mathbf{R}_{1,3}n_3) & \lambda \mathbf{R}_{2,2}n_1 + \mu \mathbf{R}_{2,1}n_2 \\ \lambda \mathbf{R}_{1,1}n_2 + \mu \mathbf{R}_{1,2}n_1 & \lambda \mathbf{R}_{2,2}n_2 + \mu(2\mathbf{R}_{2,2}n_2 + \mathbf{R}_{2,1}n_1 + \mathbf{R}_{2,3}n_3) \\ \lambda \mathbf{R}_{1,1}n_3 + \mu \mathbf{R}_{1,3}n_1 & \lambda \mathbf{R}_{2,2}n_3 + \mu \mathbf{R}_{2,3}n_2 \\ \lambda \mathbf{R}_{3,3}n_1 + \mu \mathbf{R}_{3,1}n_3 & \\ \lambda \mathbf{R}_{3,3}n_2 + \mu \mathbf{R}_{3,2}n_3 & \\ \lambda \mathbf{R}_{3,3}n_3 + \mu(2\mathbf{R}_{3,3}n_3 + \mathbf{R}_{3,1}n_1 + \mathbf{R}_{3,2}n_2) & \end{bmatrix} \cdot \begin{pmatrix} \hat{\mathbf{u}} \\ \hat{\mathbf{v}} \\ \hat{\mathbf{w}} \end{pmatrix} = \begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix} \quad (5.4)$$

Thanks to (5.4) we can now move on towards implementation aspects of a 3D code that can solve isotropic linear elasticity problems, which comprehend Neumann boundary conditions applications. As we pointed out in Chapter 4, the proposed 3D code provides a triple "for" loop over the collocation points.

This cycle

- allows to perform B-Splines tensor product in order to construct NURBS basis functions;
- performs mapping operation for all collocation points;
- assembles the stiffness matrix and the residual force vector as sparse quantities.

The code is thought to work three rows of the stiffness matrix at a time for every considered collocation point. Since we want to impose Neumann B.Cs inside the loop only on special geometry occasions, we decide to use 'if cases' within the cycle. As far as if cycles is concerned, inside of which we impose the prescribed conditions, we distinguish between:

- corners
- edges
- faces

As a final comment we remark that in order to enforce equation (5.4) we should project the traction vector. This projection operation differs if we are considering corners, edges or faces, positively or negative oriented accordingly to the direction identified by the outer normal, if we are imposing normal traction, or by the tangential versors, if we are enforcing shear conditions.

5.2 Pressurized thick-walled cylinder test

We now consider an infinitely long and internally pressurized thick-walled cylinder. We take advantage of the symmetry, considering only a quarter of the cylinder, reducing to the geometry of $u = 0$ for $x = 0$ and $v = 0$ for $y = 0$ and assume a radial pressure load P , uniformly distributed at the inner radius (see [5]). The geometry of the problem can be recovered in Figure 5.1.

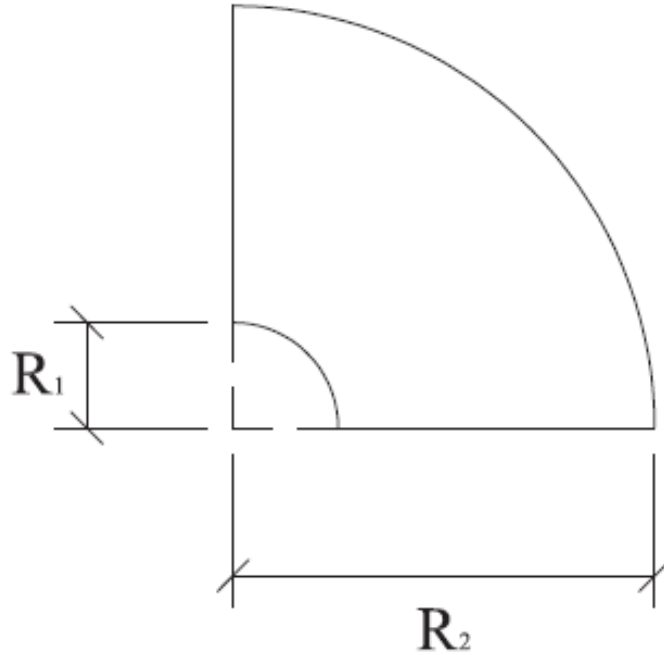


Figure 5.1: Clamped quarter of an annulus (mixed B.Cs): geometry of the problem.

For the problem under investigation the analytical exact solution in terms of radial displacement is

$$u_r(r) = \frac{PR_i^2}{E(R_o^2 - R_i^2)} \left[(1 - \nu)r + (1 + \nu)\frac{R_o^2}{r} \right] \quad (5.5)$$

where r is the radial coordinate, R_i and R_o are the internal and the outer radii, and E and ν are the Young's modulus and the Poisson's ratio. Setting $R_i = 1$ and $R_o = 4$, $E = 1$ and $\nu = 0$, the

solution becomes

$$u_r(r) = \frac{P}{15} \left(r + \frac{16}{r} \right) \quad (5.6)$$

Imposing $P = 15/8$, we obtain $u_r(1) = 2.125$ and $u_r(4) = 1$, so that we can check our approximated solution. In Figure 5.2 we plot the obtained approximated solution, while in picture 5.3 we report the difference between the exact solution and the approximate one. The presented results are obtained using 40×40 control points and $p = q = 4$, as suggested in [5]. Figure 5.2 shows that we can exactly capture the analytical solution at the boundary, while in picture 5.3 we stress that, using this mesh and degree of approximation, the difference between the exact and approximate solution is precise up to 10^{-5} .

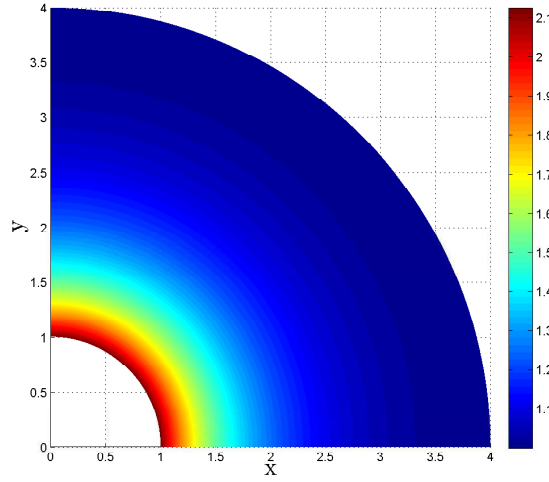


Figure 5.2: Clamped quarter of an annulus (mixed B.Cs): obtained approximate solution.

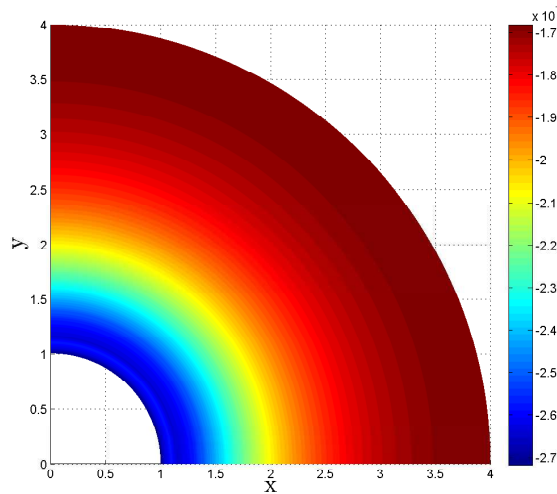


Figure 5.3: Clamped quarter of an annulus (mixed B.Cs): difference between obtained approximate solution and exact one.

In Figure 5.4 we plot the relative solution error in the L^2 -norm versus the total number of control points in a log-log plane the radial direction and we prove that convergence is attained. This result is obtained using 20x20 up to 130x130 control points, with a step of 10 control points. We therefore calculate the slope considering the last two points of the curves showed in 5.4 and we

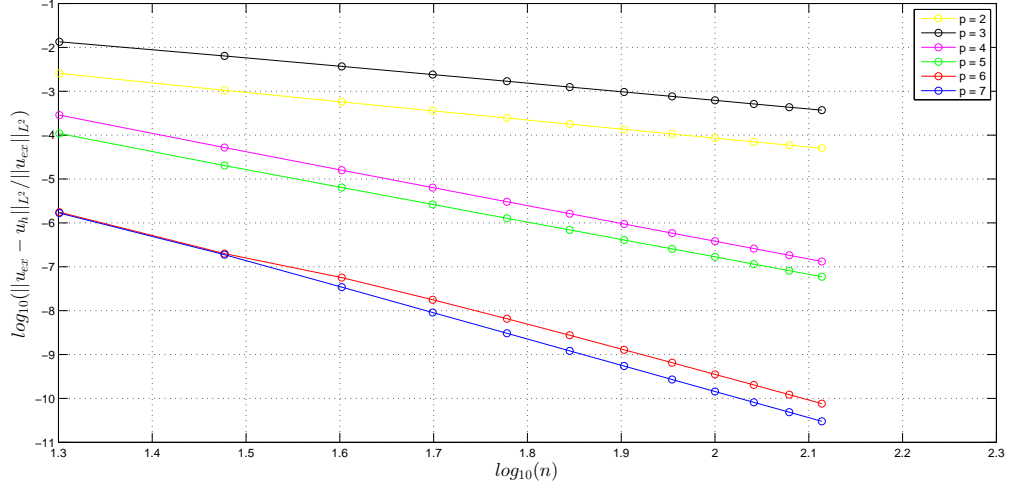


Figure 5.4: Clamped quarter of an annulus (mixed B.Cs) convergence test in radial direction: number of d.o.fs vs error in terms of L^2 -norm.

obtain the order of convergence of the method for each imposed degree of approximation, which proves to be in line with what we discussed in section 2.1, i.e. we get p order of convergence for even degrees, while we obtain $p - 1$ order for odd degrees, as Table 5.1 states.

Table 5.1: Clamped quarter of an annulus (mixed B.Cs): comparison between imposed degree of approximation and obtained order of convergence.

Degree of approximation	Order of convergence
2	2.03
3	2.05
4	4.11
5	4.17
6	6.29
7	6.43

5.3 Patch Test

We now consider a square domain Ω of side $L = 1$, subjected to uniform traction, as shown in Figure 5.5. Accordingly, we specify the following boundary conditions $u = 0$ for $x = 0$ and $v = 0$ for $y = 0$, while we assume a uniform traction q for $x = L$ and traction-free conditions for $y = L$ (see [5]). The domain consists of a single material and it is represented by a single NURBS patch. It is worth emphasizing that for this kind of problem it is necessary to explicitly impose not only traction boundary conditions (as in classical Galerkin methods) but also traction-free ones (which are instead naturally satisfied in typical Galerkin methods). Moreover, the problem under investigation is also characterized by a corner (point A in Figure 5.5) with a combination of traction boundary condition in one direction and traction-free in the other direction. The

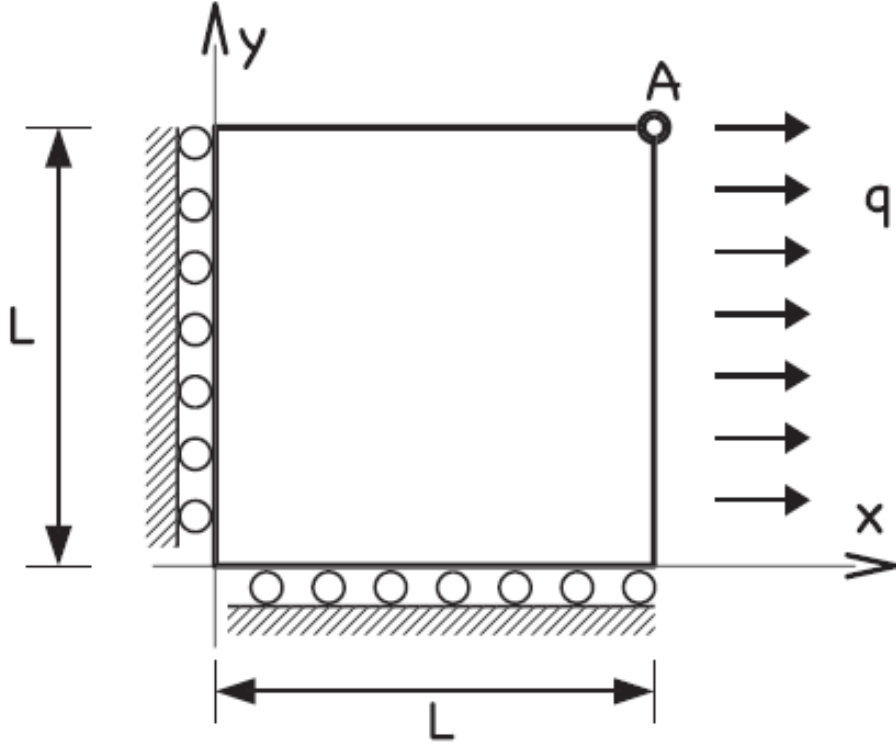


Figure 5.5: Patch traction test: problem geometry and boundary conditions.

analytical problem solution is homogenous and governed by the following strain-stress flexibility relations:

$$\begin{pmatrix} \epsilon_{11} \\ \epsilon_{22} \\ \epsilon_{33} \\ \epsilon_{12} \end{pmatrix} = \frac{1}{E} \begin{bmatrix} 1 & \nu & \nu & 0 \\ \nu & 1 & \nu & 0 \\ \nu & \nu & 1 & 0 \\ 0 & 0 & 0 & 1 + \nu \end{bmatrix} \cdot \begin{pmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{12} \end{pmatrix} \quad (5.7)$$

Being in a plane-strain situations, enforcing $\epsilon_{33} = 0$, it is possible to express ϵ_{33} in terms of σ_{11} and σ_{22} and then, requiring $\sigma_{12} = \sigma_{22} = 0$, it is possible to compute the solution as

$$\begin{cases} \epsilon_{11} = \frac{1-\nu^2}{E} \sigma_{11} \\ \epsilon_{22} = \frac{-\nu}{E} (1 + \nu) \sigma_{11} \end{cases} \quad (5.8)$$

Assuming a distributed load per unit length $q = 10$ and material constants $E = 1000$ and $\nu = 0.25$, the displacement components of point A are then $u_A = 9.375 \cdot 10^{-3}$ and $v_A = 3.125 \cdot 10^{-3}$. Such an

analytical solution is reproduced up to machine precision (10^{-15}) by the numerical one computed using a single element, illustrating the good behavior of the proposed numerical scheme for the case under investigation. Figures 5.6 and 5.7 show the horizontal and vertical displacement fields (obtained using $p = q = 2$ and 3×3 control points, i.e., one element), which are linear in the two coordinate variables, as expected.

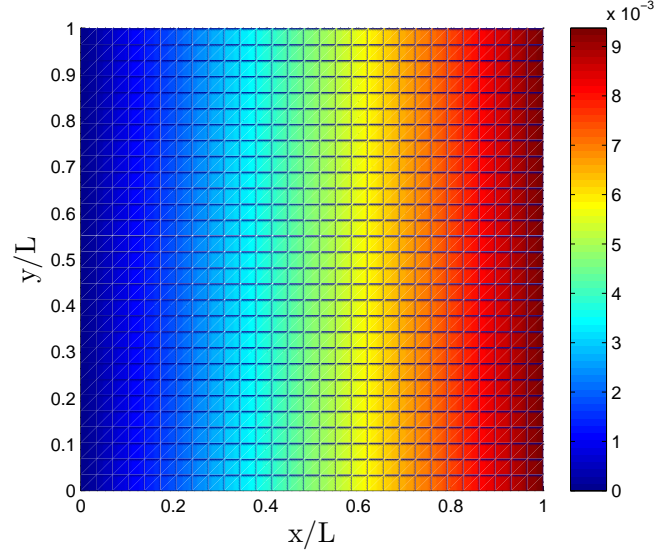


Figure 5.6: Patch traction test: horizontal displacement field.

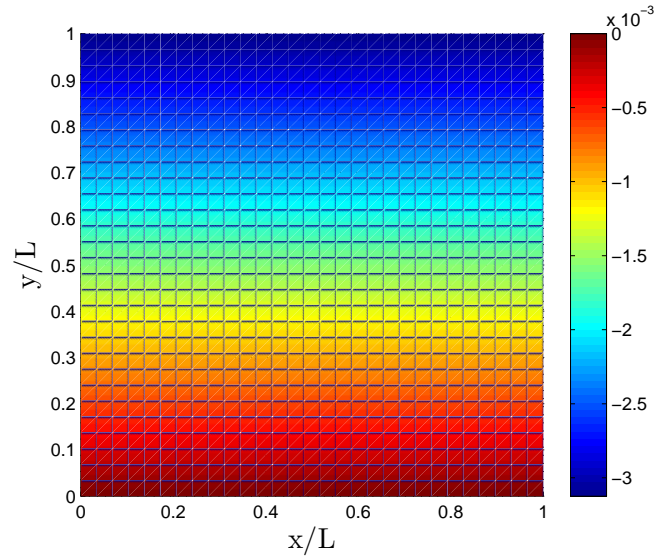


Figure 5.7: Patch traction test: vertical displacement field.

Chapter 6

Spectral approximation

In the field of structural engineering dynamics plays a fundamental role. In fact the analysis of a structure under free motion provides the most important dynamic properties of the structure itself which are the natural frequencies and the corresponding modal shapes. This structural information is obtained considering a free vibration problem.

One of the most important results of Galerkin IGA [20], with a fundamental impact on the solution of structural dynamics problems, is its capability of approximating higher modes, without introducing spurious “optical branches” in the numerical spectrum. In order to understand this concept we compare C^1 -continuous quadratic NURBS functions with the classical C^0 -continuous quadratic finite elements. The results are shown in Figure 6.1, where we report the normalized frequency results, ω_n^h/ω_n , versus the mode number, n , normalized by the total number of degrees-of-freedom, $N = n_{eq} = 999$. This Figure 6.1 illustrates the power of IGA NURBS basis functions

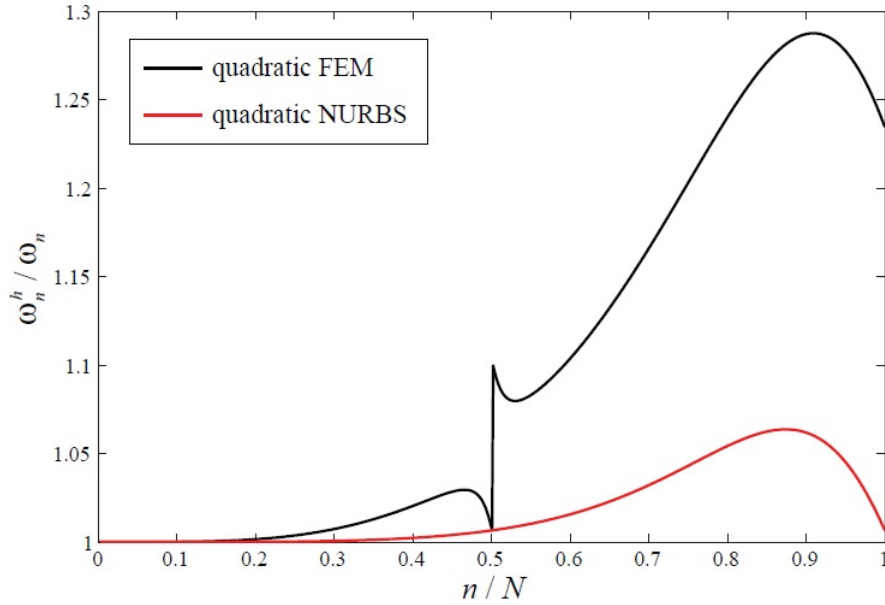


Figure 6.1: Fixed-fixed rod. Normalized discrete spectra using quadratic finite elements and NURBS.

compared with FEA ones. In this case, the finite element results shows a so-called *acoustical branch* for $n/N < 0.5$ and an *optical branch* for $n/N > 0.5$ (see [14]). This branching is due to the fact that there are two distinct types of difference equations for the finite elements: those

corresponding to the end-point nodes at element boundaries, and those corresponding to mid-point nodes on element interiors; as shows Figure 6.2. The acoustical branch corresponds to modes in which the neighbouring end- and mid-point nodes oscillate in phase with each other, and the optical branch modes are the modes in which they are out of phase. Alternatively, the quadratic NURBS difference equations are all identical (see Figure 2.1), and no such branching takes place. The same cited characteristic are present also in IGA collocation, as it has been shown in [5].

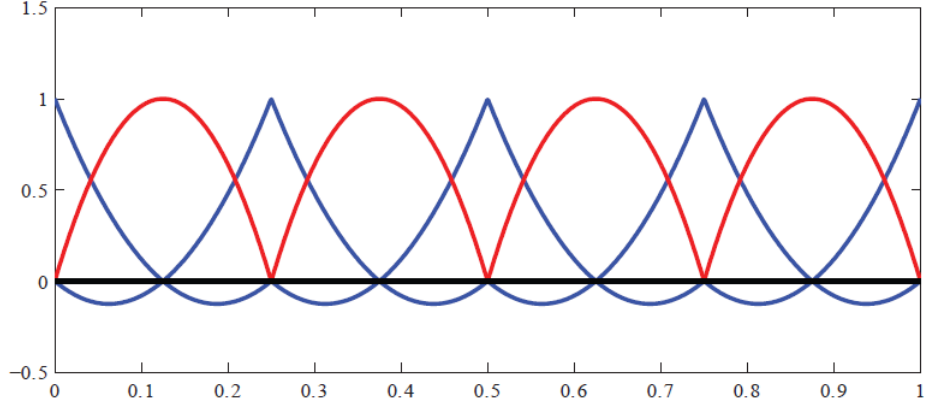


Figure 6.2: Nodal finite element basis functions for the quadratic p-method. Note the two distinct types of functions corresponding to end-nodes and mid-nodes. These lead to two distinct difference equations corresponding to the end-point nodes at element boundaries and the mid-point nodes in element interiors.

In order to show these prerogatives, the following 1D eigenvalue problem is considered:

$$\begin{cases} u'' + \omega^2 u = 0 & \forall x \in (0, 1) \\ u(0) = u(1) = 0 \end{cases} \quad (6.1)$$

for which the exact frequencies ω_n are given by

$$\omega_n = \pi n \quad \text{with } n = 1, 2, 3, \dots \quad (6.2)$$

Problem (6.1) is solved using the collocation method with Greville abscissae and, in Figure 6.3, we report the results in terms of normalized discrete spectra, obtained considering a linear parameterization and using different degrees of approximation (1000 d.o.fs have been used to produce each spectrum). It is possible to observe the good behavior of all spectra, which converge for an increasing degree p as it happens with Galerkin IGA (for more details on eigenvalue problems solved via IGA collocation, see [3]).

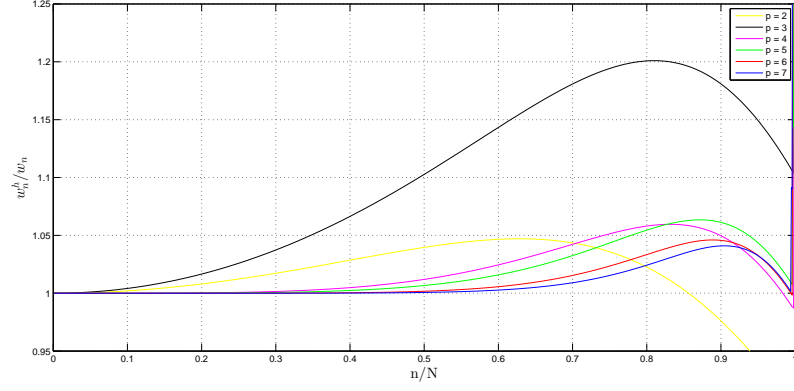


Figure 6.3: Fixed-fixed rod. Normalized discrete spectra.

We also report two relevant graphics (6.4 and 6.5) which show the convergence of the first three eigenvalues which prove to be satisfactory and in agreement with the reference line tendencies.

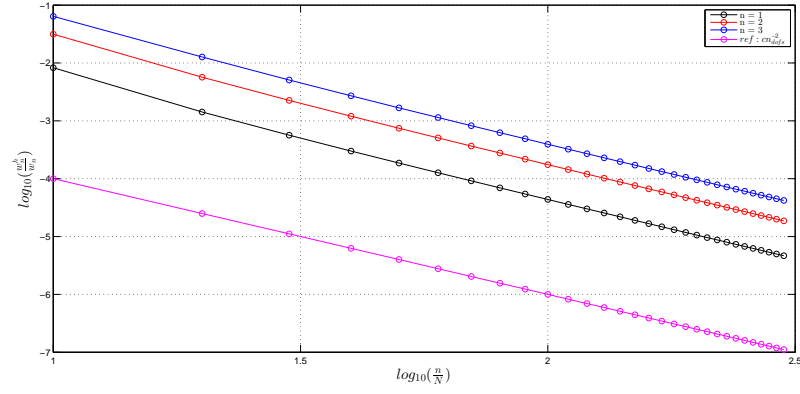


Figure 6.4: 1D eigenvalue problem with linear parametrization. Convergence of the first three eigenvalues for $p=3$.

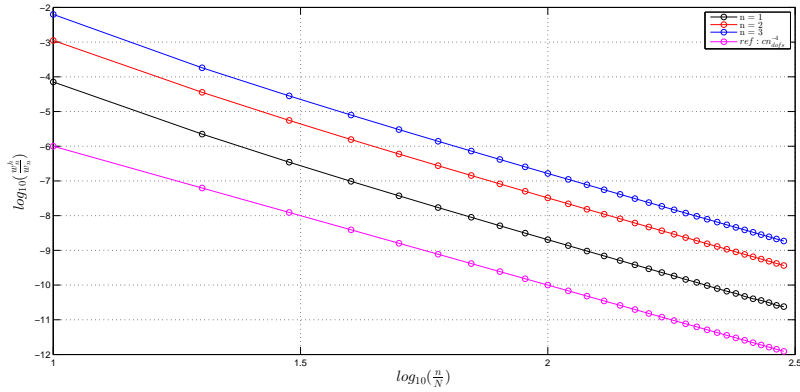


Figure 6.5: 1D eigenvalue problem with linear parametrization. Convergence of the first three eigenvalues for $p=4$.

Chapter 7

A cost-effective IGA-Collocation approach for ortotropic composite plates

In this chapter we present the main motivations, as well as the theoretical background, which lead to the proposed approach for an accurate simulation of composite laminates. For instance composite materials are used in a wide variety of fields such as aerospace or automotive. Their study has become more and more important along the past years especially because of their light weight and very resistant mechanical properties [27]. From a physical point of view composite laminates are usually made of several layers of highly resistant fibers embedded in a soft matrix. Laminate structures are subjected to delamination, which means that they tend to damage themselves at the interfaces between layers as a mode of failure. The prediction and evaluation of damage in composite laminates requires an accurate esteem of the three-dimensional stress state through the thickness, although most of the studies available in the literature consider the laminate as a two-dimensional element. In order to predict this physical phenomena several approaches are possible and here summarized:

- two-dimensional theories such as *shell approaches* prove to be not accurate enough to reliably predict interlaminar damage and delamination;
- *layerwise theories*, which rely on heavy computations and hybrid approaches, have been developed in the field of FEA ([16] and [60]) to compute more accurately the mechanical state inside the laminate;
- over the last decade new methods have been proposed such as Isogeometric Analysis (IGA) (Hughes et al. [38]) by means of which the shape functions connects CAD field, i.e. the geometry, and what is obtained from a FEA analysis i.e., the state variables, since they are the same.

As already widely discussed in Chapter 2 this method relies on B-Splines or NURBS (Non Uniform Rational B-Splines) which possess peculiar mathematical properties, such as high order approximation due to continuity and lead to new possibilities for refinement operations, that can be used to improve the simulation. Moreover, their smoothness guarantees higher accuracy and opens the door to the discretization of high-order PDEs in primal form. IGA has been successfully used to solve a wide range of problems from structural field (see e.g. [13],[19],[28],[44] and [52]) to fluids (see e.g. [1] and [9]) and fluid-structure interaction (see e.g. [10]) and [36]). This kind of methods have already been used to solve composite laminate problems, especially relying on high-order theories ([64], [41], [61]) or recovering the full 3D stress state taking advantage of 3D isogeometric

analysis such as in ([61], [34], [35]). In such approaches, each ply of the laminate is delimited by a C^0 -continuous interface. Such an approach (referred as "layerwise") is proven to be costly, though accurate, since it involves a large number of degrees of freedom when a lot of layers are composing the laminate.

7.1 A brief introduction to 3D strategies for laminates

In this section we briefly introduce the standard strategies taken into account in [27] as a starting point. In [27] two types of approach to compute displacements are considered:

- Layerwise approach ([61],[34]), which is similar to higher order FE methods and considers each layer as one patch through the thickness and C^0 -continuity at the material discontinuities as can be seen in Figure 7.1 on the left. Since this approach is completely layerwise and, as a consequence, the main drawback is represented by the fact that the number of degrees of freedom is proportional to the number of layers. This method uses a standard integration rule: $p + 1$ points in each direction (p being the degree of the shape function).
- Single-element approach, which uses a single element through the thickness (see Figure 7.1 on the right). Its nature strongly reduces the number of degrees of freedom with respect to the previously mentioned method. To account for the presence of the layers, the material matrix is homogenized. Such a method in order to get satisfying results is coupled with a post-processor to improve the solution, as explained in the next section. It should be noted that the in-plane continuity of the shape functions in both approaches is the same.

In this work we will consider an IGA-collocation approach (see Chapter 2 where NURBS are used for approximating both geometry and displacements, starting from the strong formulation of the problem and evaluating what we need at the collocation points in order to post-process the out-of-plane stresses. We therefore need to average the material through the thickness, due to the nature

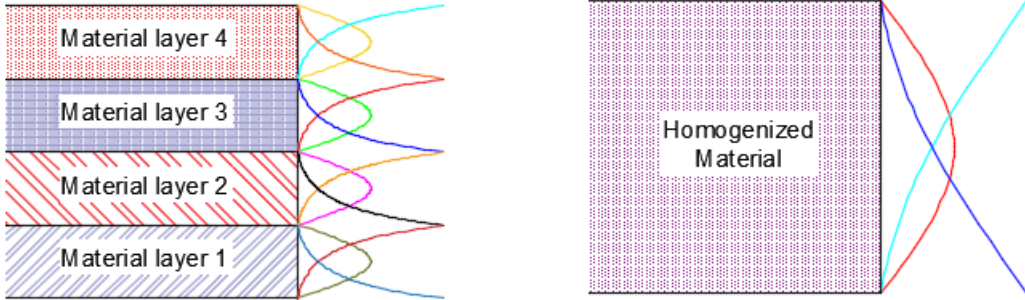


Figure 7.1: Isogeometric shape functions used in the two considered approaches. Layerwise approach with C_0 lines at material discontinuities on the left and Homogenized single-element approach on the right.

of the selected method. To do so we refer to [63] in order to homogenize the material properties to create an equivalent laminate. Explicit expressions for effective elastic constants for general thick

laminates are presented as follow

$$\begin{aligned}
\overline{C_{11}} &= \sum_{k=1}^N \nu_k C_{11}^{(k)} + \sum_{k=2}^N (C_{13}^{(k)} - \lambda_{13}) \nu_k \frac{(C_{13}^{(1)} - C_{13}^{(k)})}{C_{33}^{(k)}} \\
\overline{C_{12}} &= \sum_{k=1}^N \nu_k C_{12}^{(k)} + \sum_{k=2}^N (C_{13}^{(k)} - \lambda_{13}) \nu_k \frac{(C_{23}^{(1)} - C_{23}^{(k)})}{C_{33}^{(k)}} \\
\overline{C_{13}} &= \sum_{k=1}^N \nu_k C_{13}^{(k)} + \sum_{k=2}^N (C_{33}^{(k)} - \lambda_{33}) \nu_k \frac{(C_{13}^{(1)} - C_{13}^{(k)})}{C_{33}^{(k)}} \\
\overline{C_{22}} &= \sum_{k=1}^N \nu_k C_{22}^{(k)} + \sum_{k=2}^N (C_{23}^{(k)} - \lambda_{23}) \nu_k \frac{(C_{23}^{(1)} - C_{23}^{(k)})}{C_{33}^{(k)}} \\
\overline{C_{23}} &= \sum_{k=1}^N \nu_k C_{23}^{(k)} + \sum_{k=2}^N (C_{33}^{(k)} - \lambda_{33}) \nu_k \frac{(C_{23}^{(1)} - C_{23}^{(k)})}{C_{33}^{(k)}} \\
\overline{C_{33}} &= \frac{1}{\left(\sum_{k=1}^N \frac{\nu_k}{C_{33}^{(k)}} \right)} \\
\overline{C_{66}} &= \sum_{k=1}^N \nu_k C_{66}^{(k)} \\
\overline{C_{44}} &= \frac{\left(\sum_{k=1}^N \frac{\nu_k C_{44}^{(k)}}{\Delta_k} \right)}{\Delta} \\
\overline{C_{55}} &= \frac{\left(\sum_{k=1}^N \frac{\nu_k C_{55}^{(k)}}{\Delta_k} \right)}{\Delta} \\
\lambda_{13} &= \overline{C_{13}} \\
\lambda_{23} &= \overline{C_{23}} \\
\lambda_{33} &= \overline{C_{33}} \\
\Delta &= \left(\sum_{k=1}^N \frac{\nu_k C_{44}^{(k)}}{\Delta_k} \right) \left(\sum_{k=1}^N \frac{\nu_k C_{55}^{(k)}}{\Delta_k} \right) \\
\Delta_k &= C_{44}^k C_{55}^k
\end{aligned} \tag{7.1}$$

We remark that $\nu_k = \frac{t_k}{h}$ stands for the volume fraction of the k -th lamina, where h is the total thickness and t_k is the k -th thickness. From the consideration of stress and displacement continuity conditions at the interfaces of the laminas, the authors assume that stresses and strains of the k -th layer can be approximated with the average of the laminate. Starting from this assumption the averaged constants present the following structure:

- the first part simply represents the weighted average of the constants through the thickness;
- the second part stands for a correction of the prior weighted average through the use of specific coefficient based on the constant itself.

7.2 Post-processing approach: Reconstruction from Equilibrium

We remark that the in-plane stress components are almost correctly captured using the single-element approach, while the out-of-plane components prove not to be satisfying (as it can be seen in Figures 8.7, 8.8, 8.9 of Chapter 8 dedicated to test results). This aspect can be explained making the following remarks:

- displacement solution is continuous in all cases;
- the in-plane stresses are discontinuous along the thickness, due to the constitutive law. In fact they read information only from layer to layer and are therefore well captured;
- Equilibrium instead prescribes continuity of out-of-plane stresses and in fact using a *layerwise* approach the C^0 lines (continuous displacements) allow to obtain discontinuous strains, that along with the effective properties layer by layer of the material, bring to continuous stresses;
- choosing single-element approach the C^0 concept at interfaces is unavoidably lost and consequently we get continuous strains that, together with the material properties, bring necessarily to discontinuous stresses, violating what derives from equilibrium.

As interlaminar delamination and other fracture processes rely mostly on such out-of-plane components, a proper through-the-thickness stress description is required. In order to compute a more accurate stress state, we choose to use the following post-processing approach based on the equilibrium equations (as proposed in ([56],[29],[65],[25]), relying on the higher regularity granted by IGA shape functions. This procedure proves to be successful in [27] in the field of IGA-FE and therefore we apply it to IGA collocation. In equilibrium state, the stresses inside the material should satisfy the equilibrium equation

$$\nabla \cdot \sigma + \mathbf{b} = \mathbf{0} \quad (7.2)$$

where $\nabla \cdot$ is the divergence operator in compact notation. In engineering notation this is translated as follows

$$\begin{aligned} \sigma_{11,1} + \sigma_{12,2} + \sigma_{13,3} &= -b_1 \\ \sigma_{21,1} + \sigma_{22,2} + \sigma_{23,3} &= -b_2 \\ \sigma_{31,1} + \sigma_{32,2} + \sigma_{33,3} &= -b_3 \end{aligned} \quad (7.3)$$

with $\sigma_{ij,k} = \frac{\partial \sigma_{ij}}{\partial x_k}$

By integrating along the thickness, we recover the out-of-plane stresses as

$$\sigma_{13}(z) = - \int_{z_0}^z (\sigma_{11,1} + \sigma_{12,2} + b_1) dz + \sigma_{13}(z_0) \quad (7.4)$$

$$\sigma_{23}(z) = - \int_{z_0}^z (\sigma_{21,1} + \sigma_{22,2} + b_2) dz + \sigma_{23}(z_0) \quad (7.5)$$

$$\sigma_{33}(z) = - \int_{z_0}^z (\sigma_{31,1} + \sigma_{32,2} + b_3) dz + \sigma_{33}(z_0) \quad (7.6)$$

We stress that, by inserting equations (7.4) and (7.5) into (7.6), the component σ_{33} can then be computed as

$$\sigma_{33}(z) = \int_{z_0}^z (\sigma_{11,11} + \sigma_{22,22} + 2\sigma_{12,12} - b_3 + b_{1,1} + b_{2,2}) dz + \sigma_{33}(z_0) \quad (7.7)$$

The integral constants should be chosen to fulfil the boundary conditions at the top or bottom surfaces.

$$\begin{aligned}\sigma_{ij,k} &= \mathbb{C}_{ijmn}\varepsilon_{mn,k}, \\ \sigma_{ij,kl} &= \mathbb{C}_{ijmn}\varepsilon_{mn,kl},\end{aligned}\tag{7.8}$$

Remark 1. *It should be noted that the integration could be performed from both surfaces and then averaged in order to divide the resulting error by two. Although it is easy to compute the boundary conditions in this case, we chose to avoid this as it relies on the perfect knowledge of the stress boundary conditions on both surfaces (including the loaded one) which are not always available.*

Assuming that the elasticity tensor \mathbb{C} of equation $\boldsymbol{\sigma} = \mathbb{C}\boldsymbol{\varepsilon}$ is constant, the derivatives of the in-plane components of the stress are computed from displacements as

$$\begin{aligned}\varepsilon_{mn,k} &= \frac{1}{2}(u_{m,nk} + u_{n,mk}), \\ \varepsilon_{mn,kl} &= \frac{1}{2}(u_{m,nkl} + u_{n,mkl})\end{aligned}\tag{7.9}$$

Equations (7.9) clearly demonstrates the necessity of a highly regular displacement solution in order to recover a proper stress state. Such condition can be easily achieved using IGA, due to the possibility to benefit from higher order shape functions (B-Splines & NURBS with $p > 2$). One could thus reconstruct a good approximation of the outer-plane stress state once an accurate description of the in-plane stress state is available. The displacement solution computed in this work is obtained using Non Rational B-Splines in the context of collocation method. This kind of shape functions can be easily constructed and differentiated since many tools are available, as the one used in this context (Matlab B-Splines and NURBS toolbox, which benefits from the results from [55]).

Chapter 8

Test case: The Pagano layered plate

In this chapter we will calibrate the new proposed approach to simulate composite structure behaviour. We will therefore describe the considered test case from a theoretical point of view in the first section, and then we will provide several numerical results in the second one.

8.1 Theoretical background

The test case used in this work of thesis is the classical one proposed in the literature by Pagano [53], since it enables us to simulate the mechanical response of composite laminates. It consists of a simply supported multilayered 3D plate with a sinusoidal loading on top and a loading-free bottom face as shown in Figure 8.1. This problem can be easily parameterized which allows to analyze

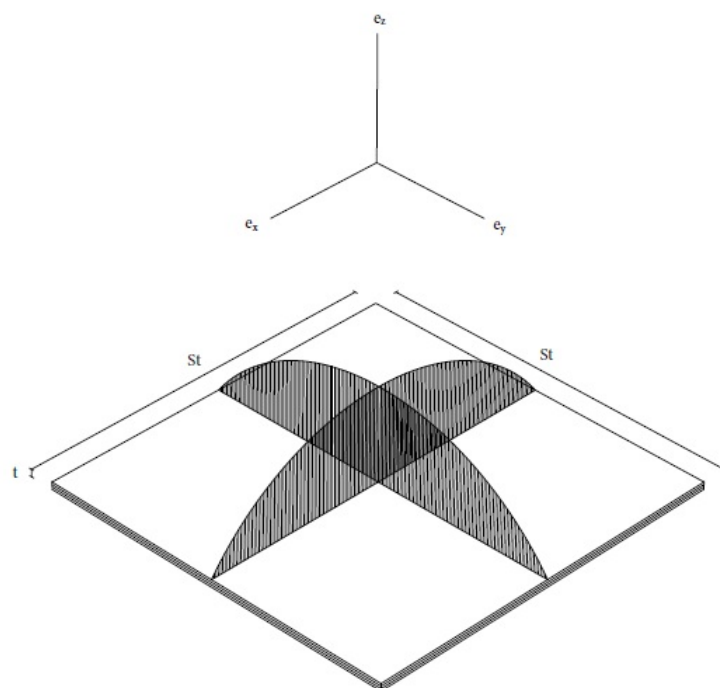


Figure 8.1: Plate geometry.

many cases in terms of layer number and distribution. In the following study, a few examples are considered using different numbers of layers (i.e., 3, 11, and 33). In all these cases, the loading conditions are the same (namely, a two dimensional sinus with a period equal to twice the length of the plate), while the thickness of every single layer is set to 1 mm, and the length of the plate is chosen to be S times larger than the total thickness t of the laminate. The laminate is composed by orthotropic layers placed orthogonally on top of each other (thus creating a 90/0/90/... laminate). From the constitutive point of view we have

$$\boldsymbol{\sigma} = \mathbb{C}\boldsymbol{\varepsilon} \quad (8.1)$$

where the elasticity tensor \mathbb{C} can be expressed, using Voigt notation, as:

$$\begin{bmatrix} \frac{1}{E_x} & \frac{-\nu_{yx}}{E_y} & \frac{-\nu_{zx}}{E_z} & 0 & 0 & 0 \\ \frac{-\nu_{xy}}{E_x} & \frac{1}{E_y} & \frac{-\nu_{zy}}{E_z} & 0 & 0 & 0 \\ \frac{-\nu_{xz}}{E_x} & \frac{-\nu_{yz}}{E_y} & \frac{1}{E_z} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G_{yz}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G_{xz}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{G_{xy}} \end{bmatrix} \quad (8.2)$$

and the material parameters for these layers are the following:

$$\begin{aligned} E_1 &= 25GPa \\ G_{23} &= 0.5GPa \\ E_y = E_z &= \frac{E_x}{25} \\ G_{xy} = G_{xz} &= \frac{G_{yz}}{2.5} \\ \nu_{xy} = \nu_{xz} = \nu_{yz} &= 0.25 \end{aligned} \quad (8.3)$$

The boundary conditions used in this test are explained graphically in Figure 8.2 and from a mathematical point of view divided in

- Dirichlet boundary condition

$$\begin{aligned} At \ x = 0, \ a : \ \sigma_x = v = w &= 0 \\ At \ y = 0, \ b : \ \sigma_y = u = w &= 0 \end{aligned} \quad (8.4)$$

- Neuman boundary condition

$$\begin{aligned} \sigma_z(x, y, \frac{h}{2}) &= q_0(x, y), \ q_0(x, y) = \sigma \sin(px) \sin(qy) \\ \sigma_z(x, y, -\frac{h}{2}) &= \tau_{xz}(x, y, \pm \frac{h}{2}) = \tau_{yz}(x, y, \pm \frac{h}{2}) = 0 \end{aligned} \quad (8.5)$$

We remark that for our case study

$$q_0(x, y) = p(x, y) = \sigma_0 \sin(\frac{\pi x}{St}) \sin(\frac{\pi y}{St}), \text{ with } \sigma_0 = 1 \quad (8.6)$$

From a computational point of view, the application of the boundary condition has been provided using an external function wich is based on if cases that we recongnize as follows:

- the 8 corners and the vertical edges have Dirichlet B.Cs: clamped displacements in all

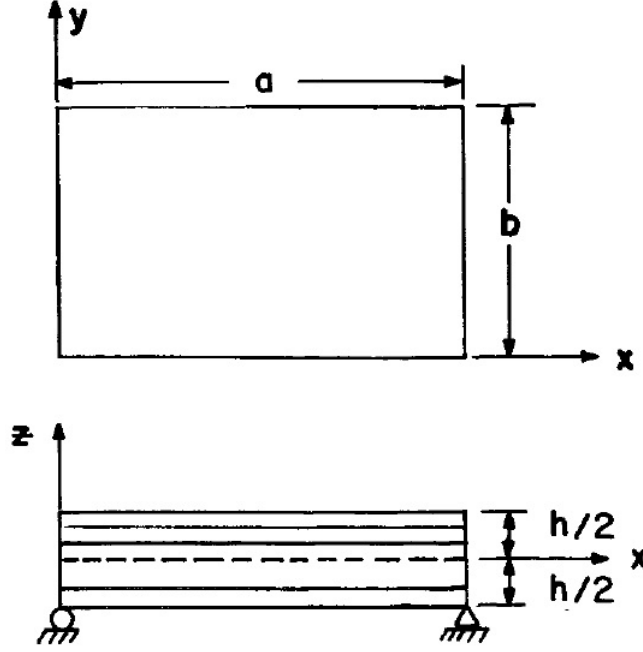


Figure 8.2: Plate notation and B.Cs

directions;

- faces having outer normal $\pm e_x$ and relative edges have displacements in x and z directions clamped and zero stress parallel to the norm component in x direction (mixed conditions);
- faces having outer normal $\pm e_y$ and relative edges have displacements in y and z directions clamped and zero stress parallel to the norm component in y direction (mixed conditions)
- the top faces feel the double-sinus loading and has zero shear components (Neuman BCs);
- the bottom face has zero shear components as well as zero normal stress component (Neuman BCs).

8.1.1 Analytical solution

The analytical solution is derived considering a generalized laminate composed by N orthotropic layers. The constitutive equations for any layer are expressed by

$$\begin{pmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \end{pmatrix} = \begin{pmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{pmatrix} \begin{pmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_z \end{pmatrix} \quad (8.7)$$

and by

$$\begin{aligned} \tau_{yz} &= C_{44} \tau_{yz} \\ \tau_{xz} &= C_{55} \tau_{xz} \\ \tau_{xy} &= C_{66} \tau_{xy} \end{aligned} \quad (8.8)$$

while the governing field equations can be written in terms of the displacement components as

$$\begin{aligned} C_{11}u_{,xx} + C_{66}u_{,yy} + C_{55}u_{,zz} + (C_{12} + C_{66})v_{,xy} + (C_{13} + C_{55})w_{,xz} &= 0 \\ (C_{12} + C_{66})u_{,xy} + C_{66}v_{,xx} + C_{22}v_{,yy} + C_{44}v_{,zz} + (C_{23} + C_{44})w_{,yz} &= 0 \\ (C_{13} + C_{55})u_{,xz} + (C_{23} + C_{44})v_{,yz} + C_{55}w_{,xx} + C_{44}w_{,yy} + C_{33}w_{,zz} &= 0 \end{aligned} \quad (8.9)$$

What follows consist of the solution of the problem for one layer. A trial displacement form is given

$$\begin{aligned} u &= U(z)\cos(px)\sin(py) \\ v &= V(z)\sin(px)\cos(py) \\ w &= W(z)\sin(px)\sin(py) \end{aligned} \quad (8.10)$$

where U, V, and W are functions of z only, and

$$\begin{aligned} p &= p(n) = \frac{n\pi}{a} = \frac{\pi}{St} \\ q &= q(m) = \frac{m\pi}{b} = \frac{\pi}{St} \end{aligned} \quad (n = m = 1, 2, 3, \dots) \quad (8.12)$$

and is inserted, in exponential fashion, in the internal equilibrium equations (8.9) obtaining an algebraic system. Non-trivial solutions of this system only exist if the determinant of the coefficient vanishes, which leads to the equation

$$\begin{aligned} \gamma^3 + d\gamma + f &= 0 \\ d &= \frac{(3CA + B^2)}{(-3A^2)} \\ f &= \frac{(2B^3 + 9ABC + 27DA^2)}{(-27A^3)} \end{aligned} \quad (8.13)$$

where

$$\begin{aligned} A &= C_{33}C_{44}C_{55} \\ B &= p^2[C_{44}(C_{11}C_{33} - C_{13}^2) + C_{55}(C_{33}C_{66} - 2C_{13}C_{44})] + \\ &\quad + q^2[C_{55}(C_{22}C_{33} - C_{23}^2) + C_{44}(C_{33}C_{66} - 2C_{23}C_{55})] \\ C &= -p^4[C_{66}(C_{11}C_{33} - C_{13}^2) + C_{55}(C_{11}C_{44} - 2C_{13}C_{66})] + \\ &\quad + p^2q^2[-C_{11}(C_{22}C_{33} - C_{23}^2) - 2(C_{12} + C_{66})(C_{13} + C_{55})(C_{23} + C_{44}) - \\ &\quad + 2C_{44}C_{55}C_{66} + 2C_{11}C_{23}C_{44} + C_{12}C_{33}(C_{12} + 2C_{66}) + C_{13}C_{22}(C_{13} + 2C_{55})] + \\ &\quad - q^4[C_{66}(C_{22}C_{33} - C_{23}^2) + C_{44}(C_{22}C_{55} - 2C_{23}C_{66})] \\ D &= p^6C_{11}C_{55}C_{66} + p^4q^2[C_{55}(C_{11}C_{22} - C_{12}^2) + C_{66}(C_{11}C_{44} - 2C_{12}C_{55})] + \\ &\quad + p^2q^4[C_{44}(C_{11}C_{22} - C_{12}^2) + C_{66}(C_{22}C_{55} - 2C_{12}C_{44})] + q^6C_{22}C_{44}C_{66} \end{aligned} \quad (8.14)$$

Once the roots of the equation in terms of γ variable are defined the general solution can be written for U, V, W

$$\begin{aligned} U(z) &= \sum_{j=1}^3 U_j(z) \\ V(z) &= \sum_{j=1}^3 L_j U_j \\ W(z) &= \sum_{j=1}^3 R_j W_j \end{aligned} \quad (8.15)$$

where

$$\begin{aligned} U_j(z) &= F_j C_j(z) + G_j S_j(z), \\ W_j(z) &= G_j C_j(z) + \alpha_j F_j S_j(z) \end{aligned} \quad (j = 1, 2, 3) \quad (8.17)$$

we stress that F_j and G_j are constant to be determined imposing the boundary conditions. Moreover we have

$$C_j(z) = \cosh(m_j z), \quad S_j(z) = \sinh(m_j z), \quad \alpha_j = 1, \quad \text{if } \left(\gamma_j + \frac{B}{3A} \right) > 0 \quad (8.19)$$

$$C_j(z) = \cos(m_j z), \quad S_j(z) = \sin(m_j z), \quad \alpha_j = -1, \quad \text{if } \left(\gamma_j + \frac{B}{3A} \right) < 0 \quad (8.20)$$

with

$$m_j = \sqrt{\left| \gamma_j + \frac{B}{3A} \right|} \quad (8.21)$$

and

$$\begin{aligned} J_j &= C_{33} C_{44} m_j^4 + \alpha_j m_j^2 [-p^2 (C_{44} C_{55} + C_{33} C_{66}) + q^2 (C_{23}^2 - C_{22} C_{33} + 2C_{23} C_{44})] + \\ &\quad + (C_{66} p^2 + C_{22} q^2) (C_{55} p^2 + C_{44} q^2) \\ L_j &= \frac{pq}{J_j} \{ \alpha_j m_j^2 [C_{33} (C_{12} + C_{66}) - (C_{23} + C_{44}) (C_{13} + C_{55})] + \\ &\quad - (C_{12} + C_{66}) (C_{55} p^2 + C_{44} q^2) \} \\ R_j &= \frac{pm_j}{J_j} [\alpha_j m_j^2 C_{44} (C_{13} + C_{55}) - (C_{13} + C_{55}) (C_{66} p^2 + C_{22} q^2) + \\ &\quad + q^2 (C_{23} + C_{44}) (C_{12} + C_{66})] \end{aligned} \quad (8.23)$$

Finally, using the strain-displacement relations of linear elasticity in conjunction with the expressions of $U(z)$, $V(z)$ and $W(z)$, we find that the stress components are given by

$$\sigma_i = \sin(px)\sin(qy) \sum_{j=1}^3 M_{ij}U_j(z), \quad (i = 1, 2, 3) \quad (8.25)$$

$$\tau_{yz} = C_{44}\sin(px)\cos(qy) \sum_{j=1}^3 (m_jLj + qRj)W_j(z) \quad (8.26)$$

$$\tau_{xz} = C_{55}\cos(px)\sin(qy) \sum_{j=1}^3 (m_j + pRj)W_j(z) \quad (8.27)$$

$$\tau_{xy} = C_{66}\cos(px)\cos(qy) \sum_{j=1}^3 (q + pLj)U_j(z) \quad (8.28)$$

where

$$M_{ij} = -pC_{1i} - qC_{2i}Lj + \alpha_j m_j R_j C_{3i}, \quad (i, j = 1, 2, 3) \quad (8.29)$$

We remark that the results showed in the next section provide normalized stress components according to [53], using the following formulas:

$$\overline{\sigma_{ij}} = \frac{\sigma_{ij}}{\sigma_0 S^2}, \quad i, j = 1, 2 \quad (8.31)$$

$$\overline{\sigma_{i3}} = \frac{\sigma_{i3}}{\sigma_0 S}, \quad i = 1, 2 \quad (8.32)$$

$$\overline{\sigma_{33}} = \frac{\sigma_{33}}{\sigma_0} \quad (8.33)$$

8.2 Implementation aspects and Numerical results

8.2.1 Computational aspects

All the results shown in this work have been obtained using an in-house code for what concerns the displacement solution and the geometry output. As far as the implemented resolutive scheme, in the spirit of collocation we rewrite the equilibrium equations in terms of displacements second derivatives combining compatibility equation and constitutive law, keeping the coefficients as they present themselves in the constitutive matrix. The displacements second derivatives are therefore approximated by the derivatives of the NURBS shape functions. On the boundary we apply consistently Neumann or Dirichlet B.Cs.

EQUILIBRIUM :

$$\sigma_{ij,j} = 0 \quad \text{in } \Omega \quad (8.35)$$

$$\sigma_{ij}n_j = t_i \quad \text{on } \partial\Omega_N \quad (8.36)$$

$$\bar{u}_i = u_i \quad \text{on } \partial\Omega_D \quad (8.37)$$

COMPATIBILITY :

$$\epsilon_{ij} = \frac{u_{i,j} + u_{j,i}}{2} \quad (8.38)$$

CONSTITUTIVE LAW :

$$\sigma_{ij} = \mathbb{C}_{ijkl}\epsilon_{kl} \quad (8.39)$$

Once that a good displacement solution is obtained we adapt the post processing code ([27]) based on the igatools library (see [54] for further details) to communicate with the IGA-collocation previously cited solutor. The implementation part can be summarized in Figure 8.3 and summed up as follows:

- The user need to set the material properties, which will be averaged by the program, as well as geometry informations (length-to thickness ratio, total thickness, number of points per layer and in-plane coordinates of the points to compute the solution with a 'vertical approach') and the minimum information for the IGA-collocation solutor i.e. the number of control points per direction and the shape functions degree of approximation.
- The program itself call a stress evaluation function, which communicate with the IGA-collocation solutor in order to obtain a rough displacements solution and the geometry given in terms of Non Rational B-splines. As far as the collocation solutor, after averaging the material with an external function, it provides a basic geometry of the problem from the length to thickness ratio and the total thickness itself, which will be later enriched by refinement operations as described in the first chapter. The B-splines are later on constructed starting from Greville abscissae (see Chapter 2). Because of the geometry, a mapping operation is required and performed in the triple "for" cycle loop on the collocation points and the boundary conditions are applied according to the expected "if" case, (see Chapter 5). Out from the "for" cycle, the problem is solved in an optimized way, because sparse matrices and vectors are used. Then stresses are calculated through strains, obtained from the given displacements, and considering the constitutive averaged matrix. We also calculate the derivatives of the stresses that will be fundamental for the post-processing recovery technique.
- The program perform the outer-plane shear stresses post-processing and consequently the normal outer-stress post-processing.
- The analytical solution is computed.
- To check if the solution obtained is satisfactory we use the following error rule

$$error = \frac{\max(\sigma_{analytic} - \sigma_{postprocessed})}{\max(\sigma_{analytic})} \quad (8.40)$$

excluding the boundary, since there the solution is known and brings to an indeterminated form $\begin{pmatrix} 0 \\ 0 \end{pmatrix}$ of the proposed error esteem.

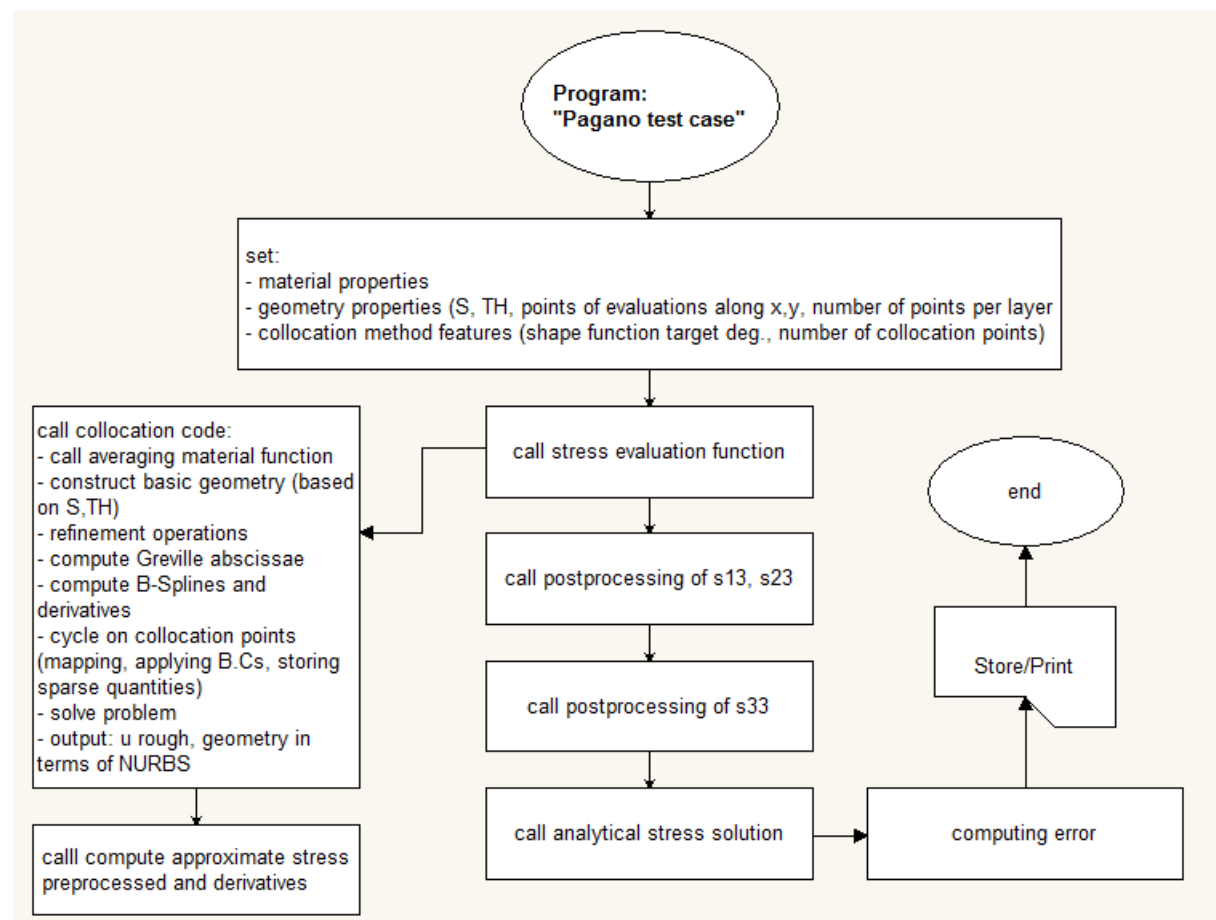


Figure 8.3: Program Flowchart.

8.2.2 Numerical results

In this section, we comment the results obtained using the proposed IGA-collocation approach, which is compared with the analytical solution. In Figures 8.4, 8.5, 8.6, 8.7, 8.8, 8.9 we give a comparison, for the cases of 3 and 11 layers (length to thickness ratio equal to 10, degree of approximation 6 and 10 collocation points per direction), between analytical stresses and approximate stresses for the in-plane quantities, while we add also the preprocessed entities for the out-of-plane stresses. For both cases the in-plane stresses show a good behaviour, as expected, while the out-of-plane stresses, without post-processing treatment, are discontinuous. This aspect is due to the fact that high order continuity of the shape functions leads to the continuity of strains, while the material properties are discontinuous.

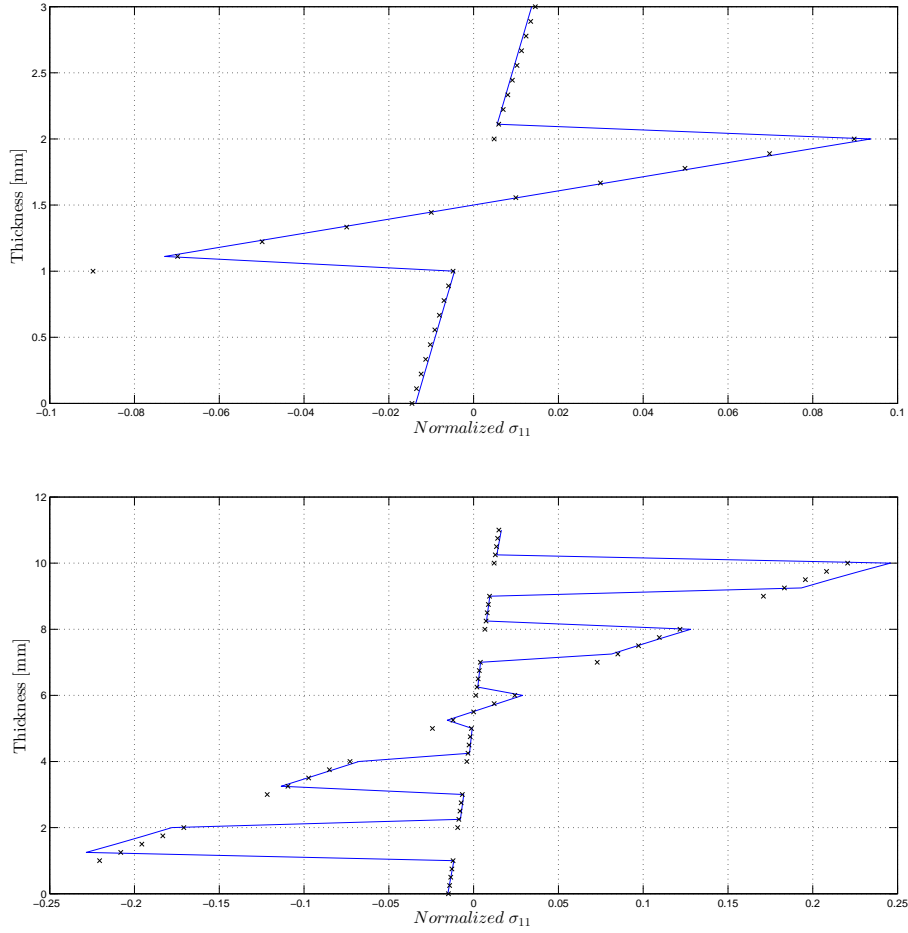


Figure 8.4: Computed σ_{11} solutions for the 3D Pagano plate problem with 3 layers, at the top, and 11, at the bottom, at the position $X = 0.25L$ and $Y = 0.25L$. The blue solid line represents the analytical solution, the black cross represent the solution obtained with no post-processing treatment.

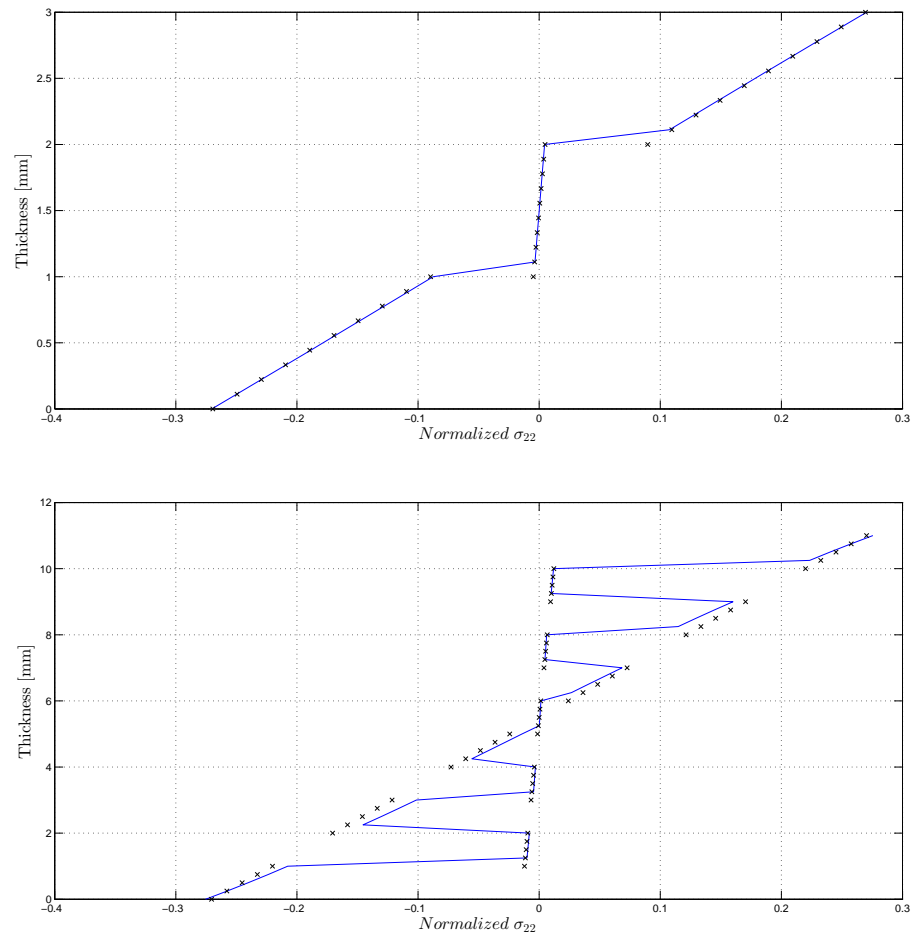


Figure 8.5: Computed σ_{22} solutions for the 3D Pagano plate problem with 3 layers, at the top, and 11, at the bottom, at the position $X = 0.25L$ and $Y = 0.25L$. The blue solid line represents the analytical solution, the black cross represent the solution obtained with no post-processing treatment.

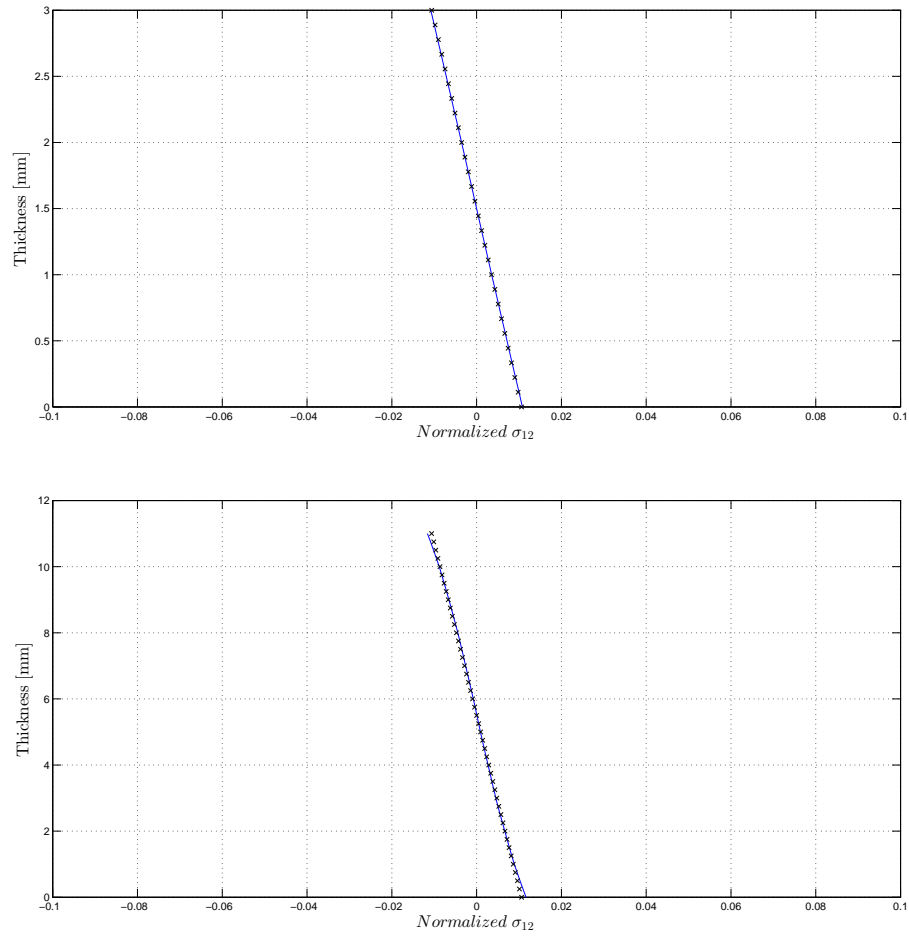


Figure 8.6: Computed σ_{12} solutions for the 3D Pagano plate problem with 3 layers, at the top, and 11, at the bottom, at the position $X = 0.25L$ and $Y = 0.25L$. The blue solid line represents the analytical solution, the black cross represent the solution obtained with no post-processing treatment.

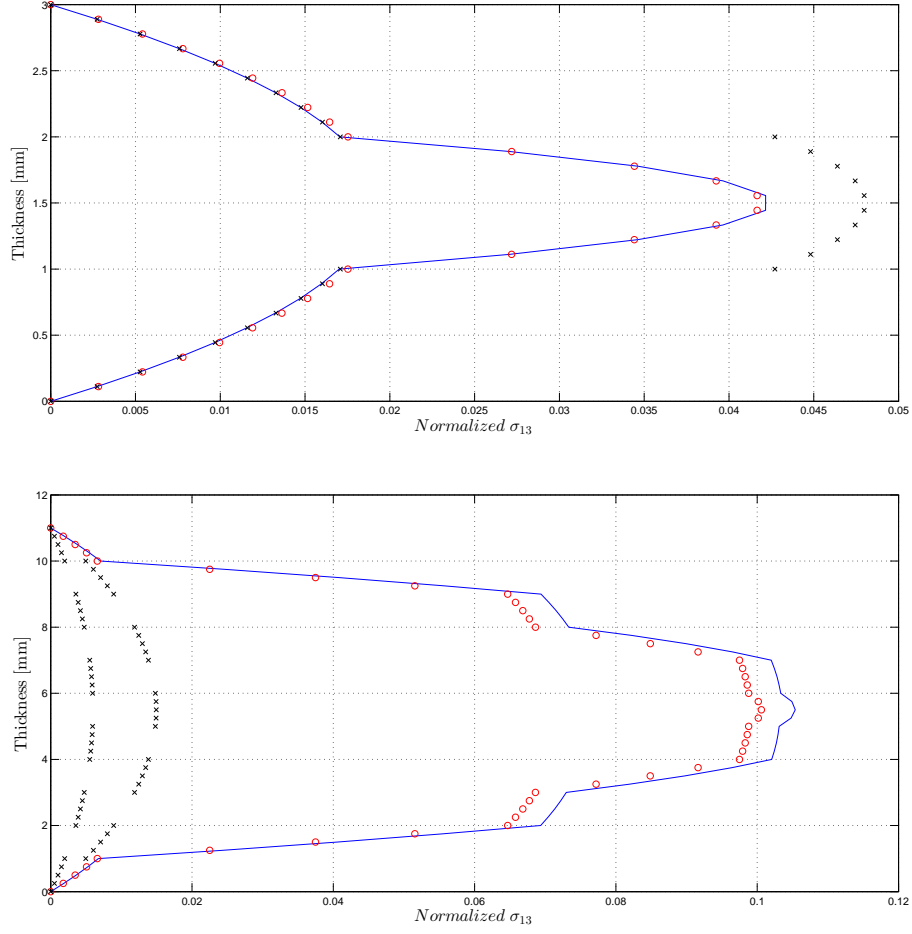


Figure 8.7: Computed σ_{13} solutions for the 3D Pagano plate problem with 3 layers, at the top, and 11, at the bottom, at the position $X = 0.25L$ and $Y = 0.25L$. The blue solid line represents the analytical solution, the black cross represent the solution obtained with no post-processing treatment, and the red circles represent the solution after post-processing treatment.

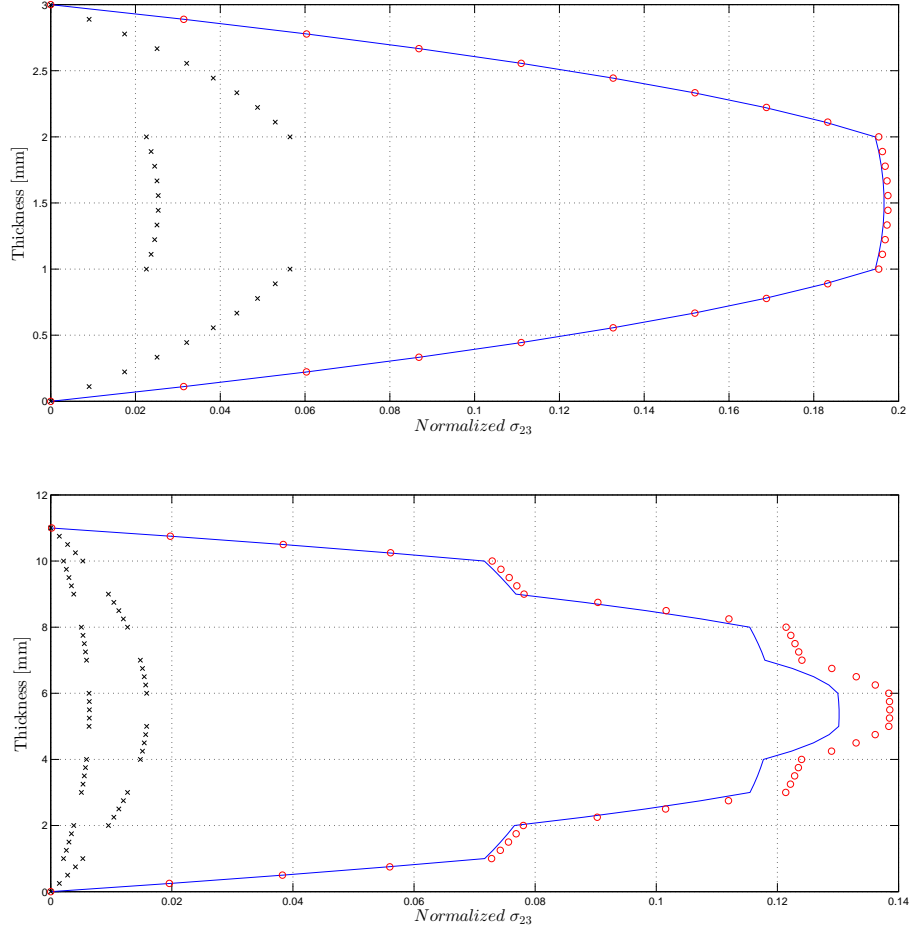


Figure 8.8: Computed σ_{23} solutions for the 3D Pagano plate problem with 3 layers, at the top, and 11, at the bottom, at the position $X = 0.25L$ and $Y = 0.25L$. The blue solid line represents the analytical solution, the black cross represent the solution obtained with no post-processing treatment, and the red circles represent the solution after post-processing treatment.

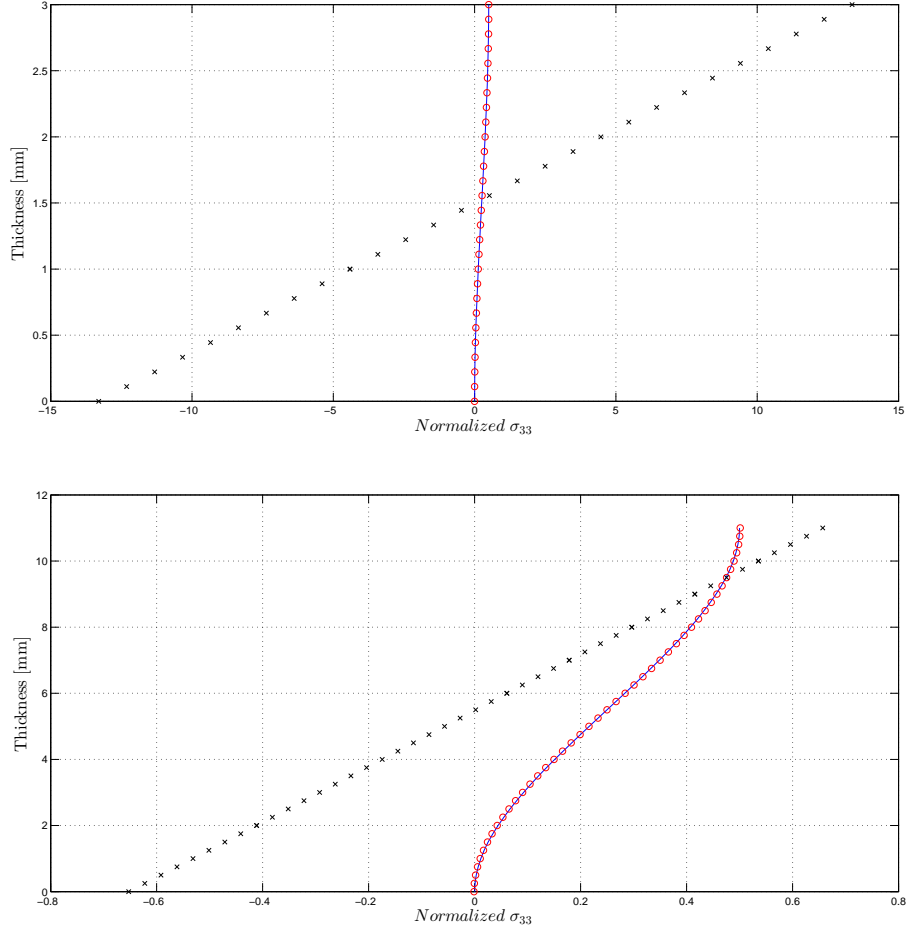


Figure 8.9: Computed σ_{33} solutions for the 3D Pagano plate problem with 3 layers, at the top, and 11, at the bottom, at the position $X = 0.25L$ and $Y = 0.25L$. The blue solid line represents the analytical solution, the black cross represent the solution obtained with no post-processing treatment, and the red circles represent the solution after post-processing treatment.

In the following results (Figures 8.11,8.12,8.2.2), we observe that the postprocessed stress state is very close to the analytical one. Our assertion demonstrates to be accurate since we are sampling every quarter of length of the plate in both directions. We conclude that the correct stress state is recovered on the whole plate, and a great improvement is observed everywhere. The results were obtained using 10 collocations points per direction with shape functions of grade 6 for the case of 11 layers and length to thickness ratio equal to 10. The obtained solution respects the boundary conditions as far as the out-of-plane stresses, in fact:

- where $x = 0$ or $x = L$ displacements in x direction exist and so do σ_{13} as a consequence;
- where $y = 0$ or $y = L$ displacements in y direction exist and therefore σ_{23} is different from zero apart from the corners;
- the sign of the stresses is compatible with the obtained displacements.

We derive our considerations on the displacements from Figure 8.10.

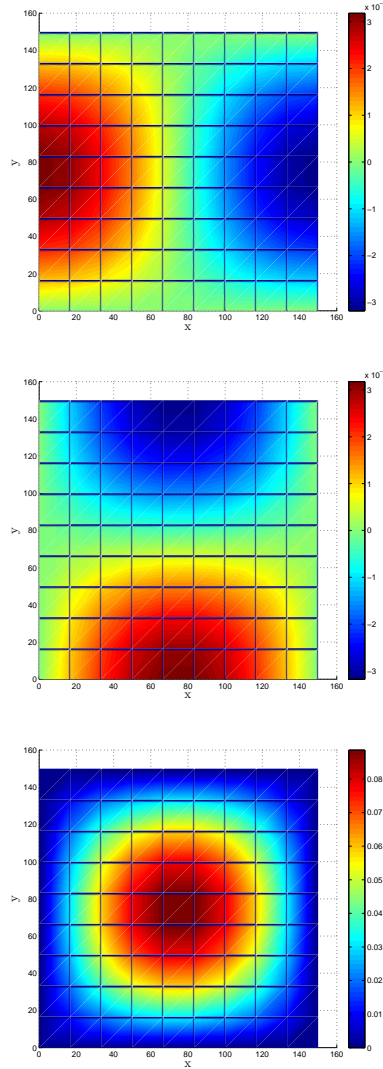


Figure 8.10: Displacements inside the domain for fix x-y plane: displacement in x direction, at the top; in y direction, in the middle; in z direction, at the bottom.

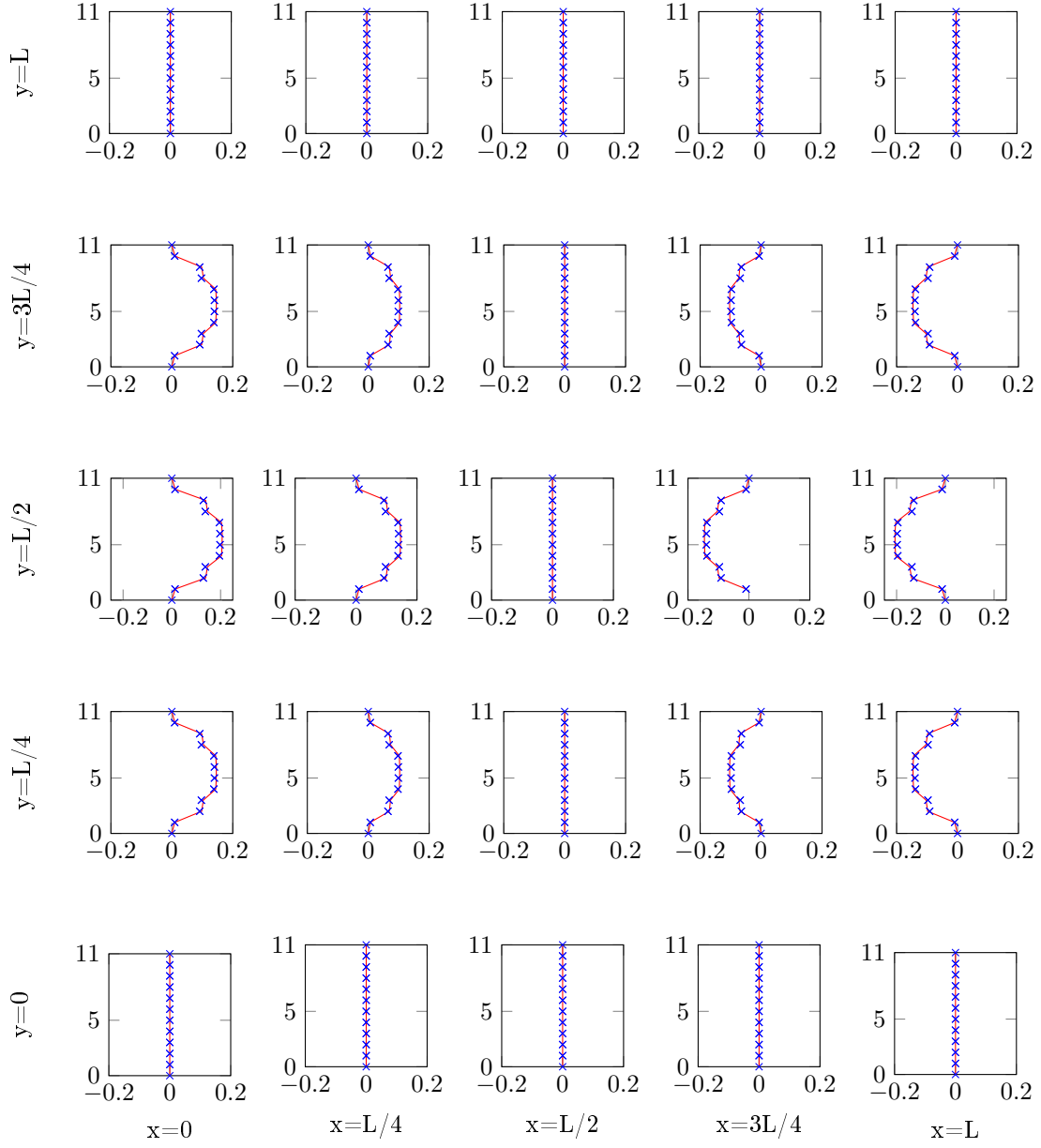


Figure 8.11: Recovered (red solid line) σ_{13} compared to the analytical one (blue crosses) for several in plane positions. L is the total length of the plate, that in this case is $L = 110\text{mm}$ (being $L = S t$ with $t = 11\text{mm}$ and $S = 10$), while the number of layers is 11.

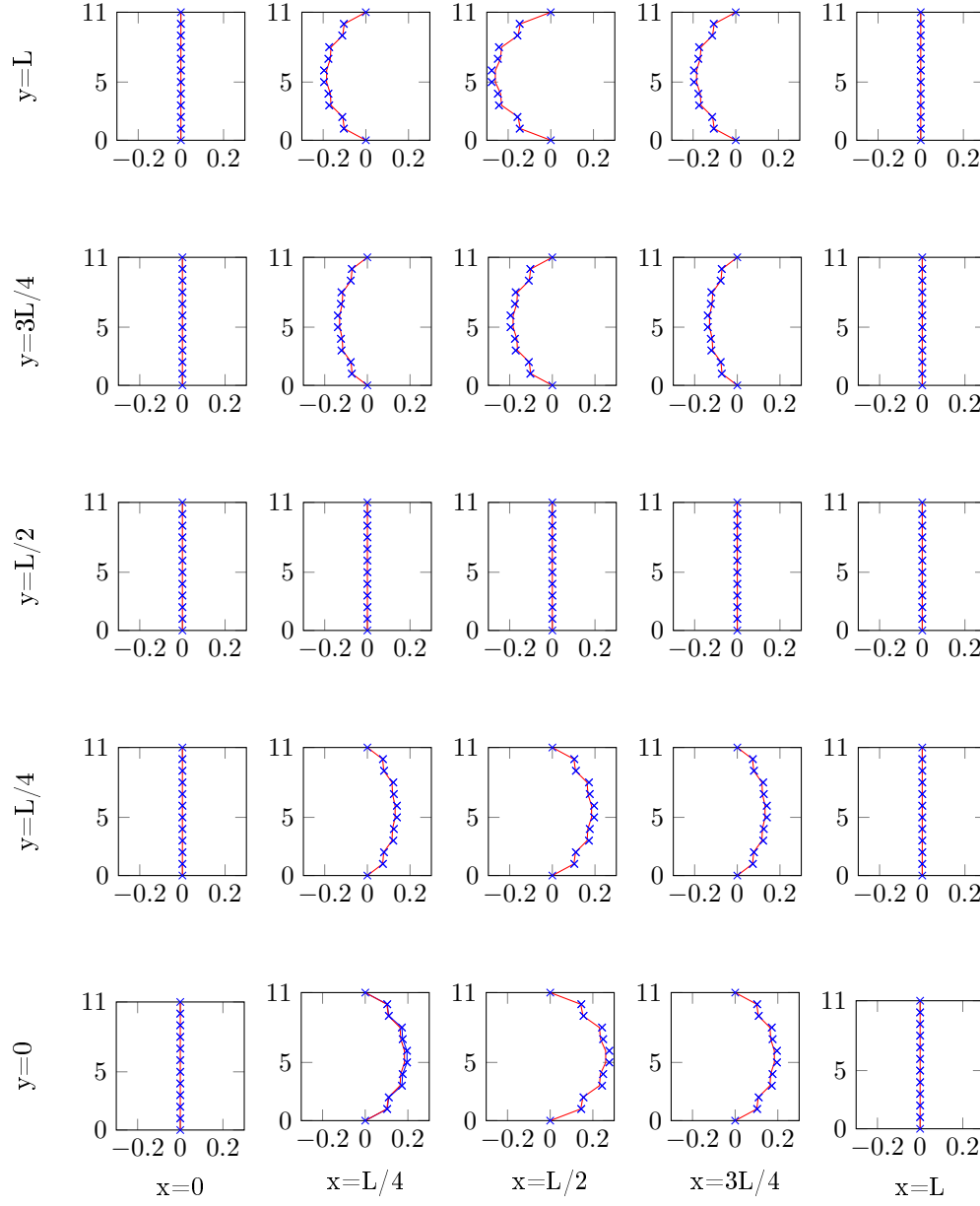


Figure 8.12: Recovered (red solid line) σ_{23} compared to the analytical one (blue crosses) for several in plane positions. L is the total length of the plate, that in this case is $L = 110\text{mm}$ (being $L = S t$ with $t = 11\text{mm}$ and $S = 10$), while the number of layers is 11.

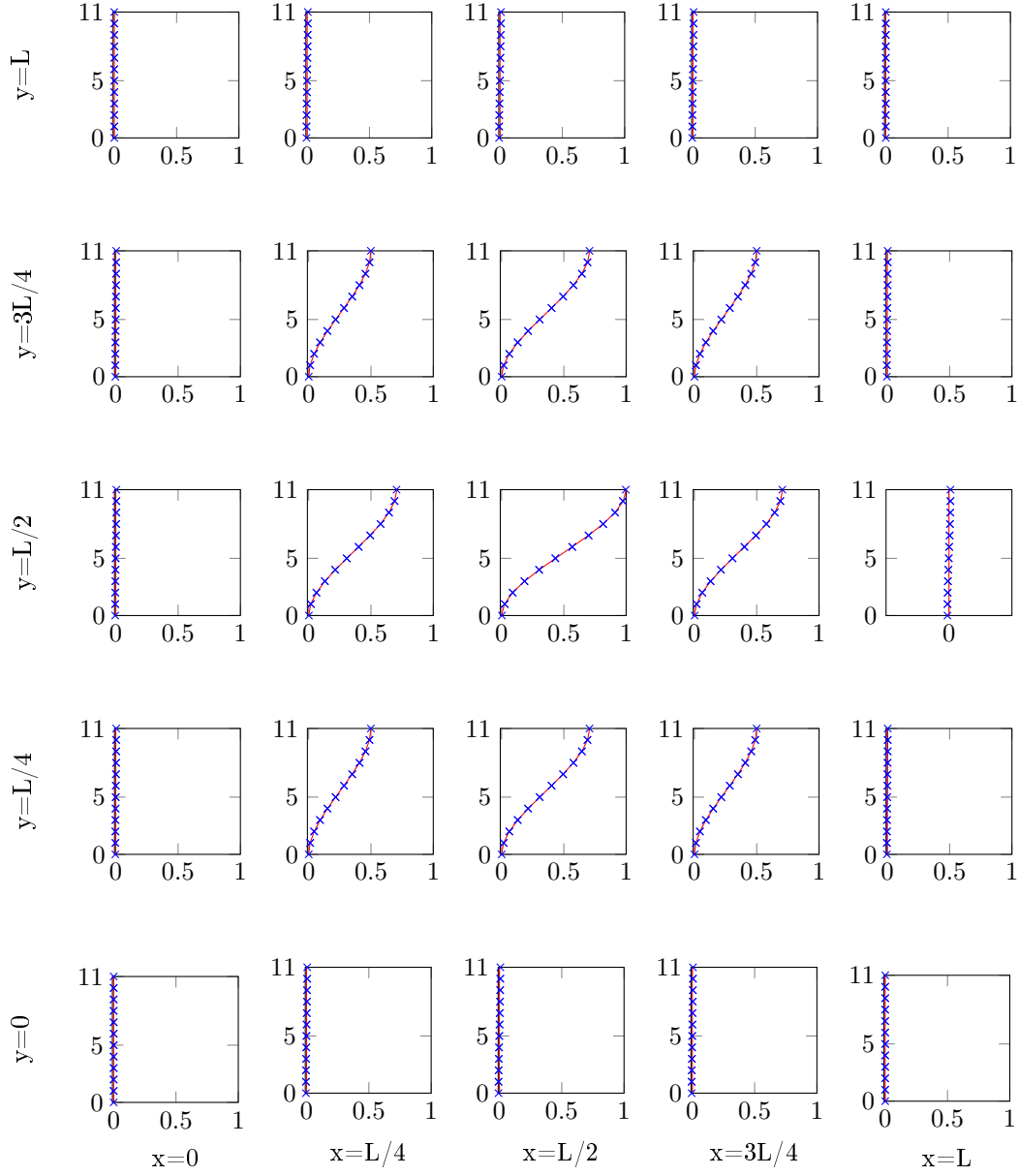


Figure 8.13: Recovered (red solid line) σ_{33} compared to the analytical one (blue crosses) for several in plane positions. L is the total length of the plate, that in this case is $L = 110\text{mm}$ (being $L = S t$ with $t = 11\text{mm}$ and $S = 10$), while the number of layers is 11.

In order to validate the proposed approach in a wider variety of cases, computations with a different ratio between the thickness of the plate and its length, namely 10, 20, 50, are performed for 3 and 11 layers and prove in figures 8.14, 8.15, that the method converges.

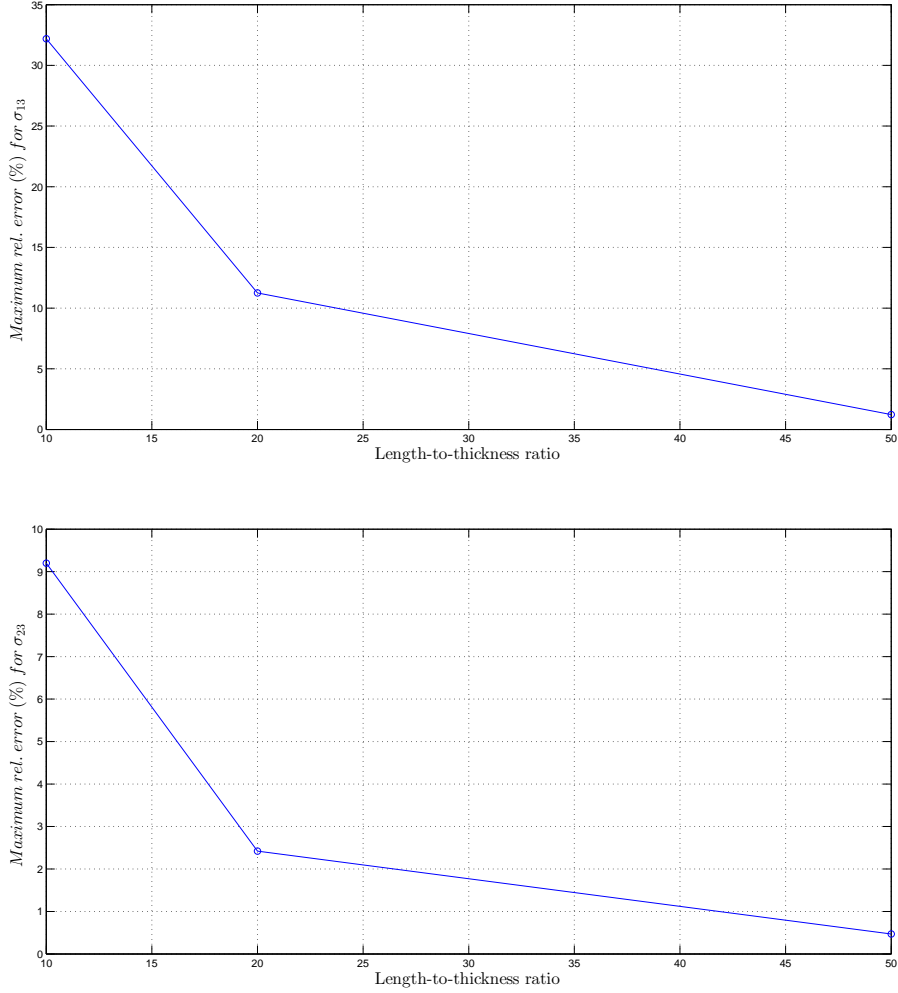


Figure 8.14: Maximum relative error between post-processed and analytical σ_{13} , at the top, σ_{23} , at the bottom, obtained considering different length-to-thickness ratios, while the degree of the shape functions is 6 and the number of layers is 3.

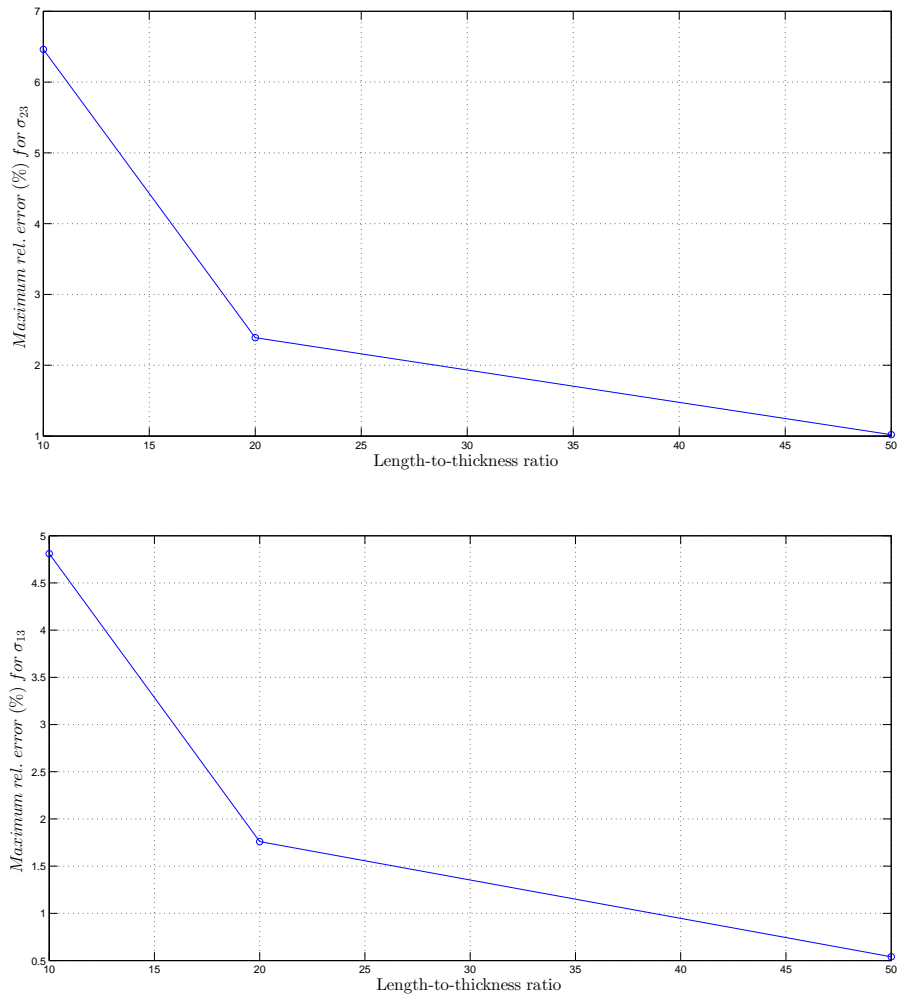


Figure 8.15: Maximum relative error between post-processed and analytical σ_{13} , at the top, σ_{23} , at the bottom, obtained considering different length-to-thickness ratios, while the degree of the shape functions is 6 and the number of layers is 11.

A simple mesh sensitivity analysis, using 1,2,4,8 elements has been carried out testing shape function degree of approximation equal to 4 and 6, for $S=10$. Also in these cases the method prove (see figures 8.16, 8.17) to be robust and to converge. We also check the influence of the number of element used in the analysis reported in Figure 8.18 using $p=6$ and $S=20$.

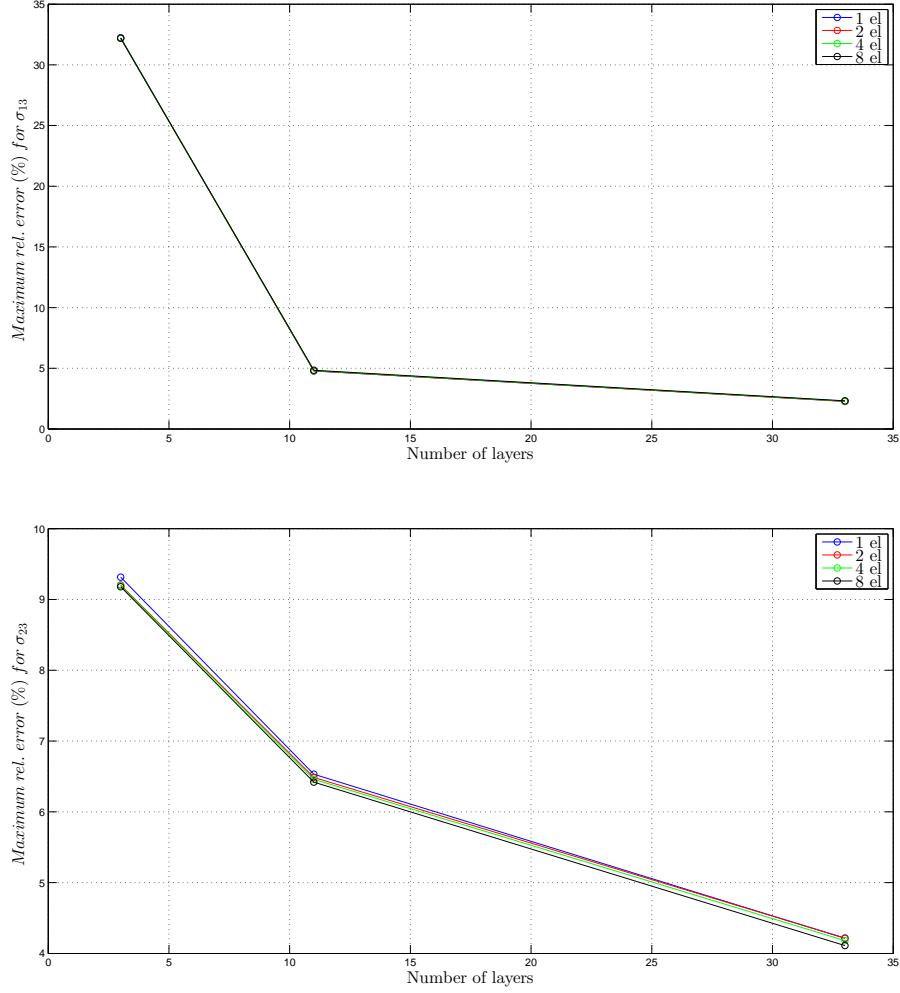


Figure 8.16: Maximum relative error between post-processed and analytical σ_{13} , at the top, σ_{23} , at the bottom, obtained considering different number of elements and layers, while the degree of the shape functions is 6 and $S=10$.

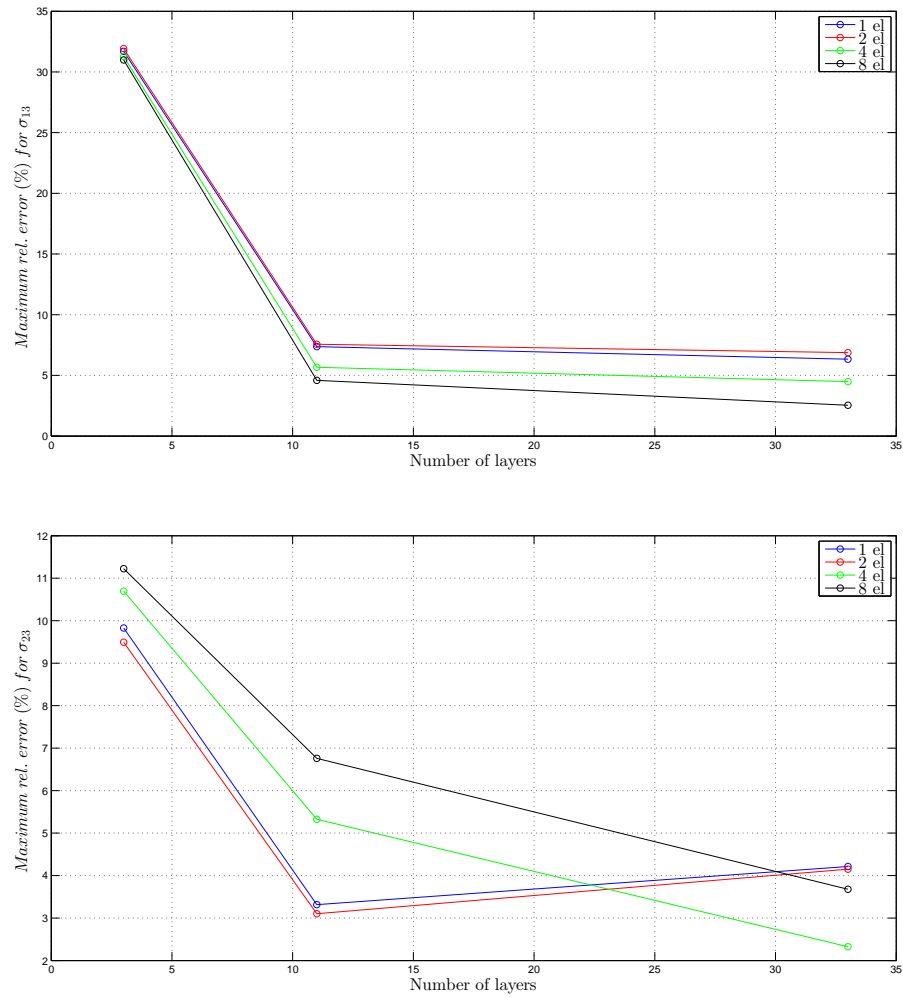


Figure 8.17: Maximum relative error between post-processed and analytical σ_{13} , at the top, σ_{23} , at the bottom, obtained considering different number of elements and layers, while the degree of the shape functions is 4 and $S=10$.

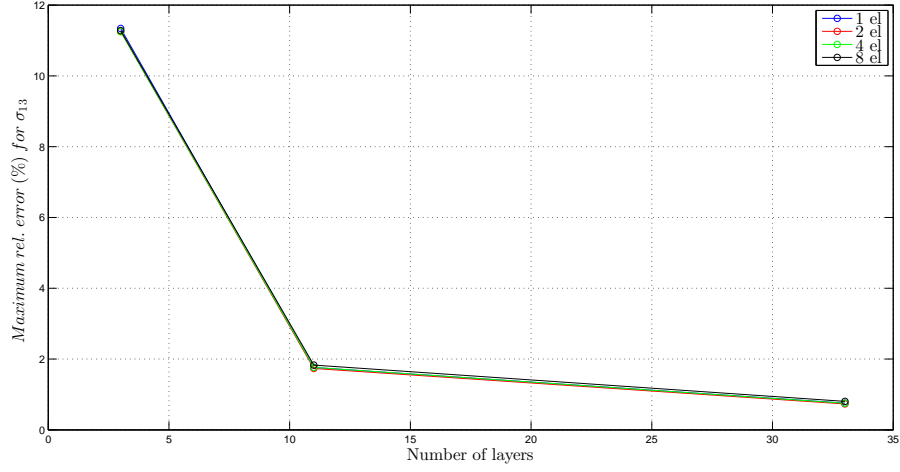


Figure 8.18: Maximum relative error between post-processed and analytical σ_{13} at the bottom, using different number of elements and layers, while the degree of the shape functions is 6 and $S=20$.

From figures err8.1,8.15,8.16,8.17,8.18 we can conclude that:

- $p=4$ is not enough to correctly describe the out-of planes stresses σ_{13} and σ_{23} .
- Using $p=6$ and 8 elements we can capture the post-processed stresses σ_{13} and σ_{23} .
- Results shown in Figures 8.14,8.15 agree with the one reported in figures 8.16,8.18.

For what concerns layers disposition, a good behaviour has been proven for odd plies, since their natural grouping is symmetric. The proposed method do not apply to even plies disposition because of the lack of symmetry with respect to the thickness, which affects the homogenization, even though we can foretell that this behaviour tend to soften increasing the length to thickness ratio as a sort of "scale effect".

Chapter 9

Conclusions and future perspectives

In this work of thesis we introduce the principles of IGA collocation, applying this method to several linear elasticity examples. The results obtained from an in-house Matlab implementations prove to be satisfactory for one-, two-, and three-dimensional problems according to L^2 -norm convergence plots.

Furthermore a new cost-effective approach for an accurate simulation of composite laminates has been presented. This technique combines a standard 3D coarse collocation isogeometric analysis with a post-processing based on equilibrium equations.

Since our collocation method is based on the solution of the governing equations in strong form without the need of integration, to properly describe the variation of the material properties through the plate thickness, we choose to average them considering an homogenized response. In this way, we can obtain satisfactory results, but only in terms of displacements and in-plane stresses. By means of our inexpensive equilibrium-based post-processing technique (allowed by the higher-continuity granted by isogeometric analysis), good results are instead recovered also in terms of out-of-plane stresses, even for coarse meshes.

The post-processing stress-recovery technique is only based on the integration of equilibrium equations, and all the required components can be easily computed knowing the displacement solution and the geometry of the problem, taking advantage of the shape function high continuity. As the IGA-collocation code provides NURBS fields, its derivation and handling are straightforward and the whole process (coarse simulation plus post-processing) is far less time consuming than a full layerwise FEA 3D approach.

The proposed technique has been tested sampling the solution obtained from post-processing treatment application every quarter of length of the considered plate in both directions, in order to check the overall quality of the numerical results. In addition to this, other analyses are carried out in order to test the sensitivity of the method to different length-to-thickness ratios and number of layers. Moreover, a basic mesh sensitivity analysis is presented and, in light of the results, we conclude that the newly presented approach works well also for very coarse meshes.

Further research topics currently under investigation involve the extension of this approach to more complex problems involving curved geometries.

Bibliography

- [1] I. Akkerman, Y. Bazilevs, V.M. Calo, T.J.R. Hughes, and S. Hulshoff. The role of continuity in residual-based variational multiscale modeling of turbulence. *Computational Mechanics*, 41:371–378, 2008.
- [2] C. Anitescu, Y. Jia, Y. J. Zhang, and T. Rabczuk. An isogeometric collocation method using superconvergent points. *Computer Methods in Applied Mechanics and Engineering*, 284:1073–1097, 2015.
- [3] F. Auricchio, L. Beirão da Veiga, T.J.R. Hughes, A. Reali, and G. Sangalli. Isogeometric collocation methods. *Mathematical Models and Methods in Applied Sciences*, 20(11):2075–2107, 2010.
- [4] F. Auricchio, L. Beirão da Veiga, A. Buffa, C. Lovadina, A. Reali, and G. Sangalli. A fully “locking-free” isogeometric approach for plane linear elasticity problems: A stream function formulation. *Computer Methods in Applied Mechanics and Engineering*, 197:160–172, 2007.
- [5] F. Auricchio, L. Beirão da Veiga, T.J.R. Hughes, A. Reali, and G. Sangalli. Isogeometric collocation for elastostatics and explicit dynamics. *Comput. Methods Appl. Mech. Engrg.*, 249-252:2–14, 2012.
- [6] F. Auricchio, L. Beirão da Veiga, C. Lovadina J. Kiendl, and A. Reali. Locking-free isogeometric collocation methods for spatial timoshenko rods. *Computer Methods in Applied Mechanics and Engineering*, 263:113–126, 2013.
- [7] F. Auricchio, L. Beirão da Veiga, C. Lovadina, and A. Reali. The importance of the exact satisfaction of the incompressibility constraint in nonlinear elasticity: mixed fems versus nurbs-based approximations. *Computer Methods in Applied Mechanics and Engineering*, 199:314–323, 2010.
- [8] G. Balduzzi, S. Morganti, F. Auricchio, and A. Reali. Non-prismatic timoshenko-like beam model: Numerical solution via isogeometric collocation. *Computers & Mathematics with Applications*, <http://dx.doi.org/10.1016/j.camwa.2017.04.025>.
- [9] Y. Bazilevs, V.M. Calo, J.A. Cottrell, T.J.R. Hughes, A. Reali, and G. Scovazzi. Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 197:173–201, 2007.
- [10] Y. Bazilevs, M.-C. Hsu, J. Kiendl, R. Wuchner, and K. U. Bletzinger. 3d simulation of wind turbine rotors at full scale. part ii: Fluid-structure interaction modeling with composite blades. *International Journal for Numerical Methods in Fluids*, 65:236–253, 2011.
- [11] Y. Bazilevs and T.J.R. Hughes. Nurbs-based isogeometric analysis for the computation of flows about rotating components. *Computational Mechanics*, 43:143–150, 2008.
- [12] C. De Boor. On calculation with b-splines. *Journal of Approximation Theory*, 6:50–62, 1972.

- [13] M.J. Borden, C.V. Verhoosel, M.A. Scott, T.J.R. Hughes, and C.M. Landis. A phase-field description of dynamic brittle fracture. *Methods in Applied Mechanics and Engineering*, 217:77–95, 2012.
- [14] L. Brillouin. *Wave Propagation in Periodic Structures*. Dover Publications, Inc, 1953.
- [15] A. Buffa, C. de Falco, and G. Sangalli. Isogeometric analysis: Stable elements for the 2d stokes equation. *International Journal for Numerical Methods in Fluids*, 65:140–1422, 2011.
- [16] E. Carrera. Theories and finite elements for multilayered, anisotropic, composite plates and shells. *Archives of Computational Methods in Engineering*, 9(2):87–140, 2002.
- [17] J.F. Caseiro, R.A.F. Valente, A. Reali, J. Kiendl, F. Auricchio, and R.J. Alves de Sousa. On the assumed natural strain method to alleviate locking in solid-shell nurbs-based finite elements. *Computational Mechanics*, 53:1341–1353, 2014.
- [18] H. Casquero, C. Bona-Casas, and H. Gomez. A nurbs-based immersed methodology for fluid-structure interaction. *Computer Methods in Applied Mechanics and Engineering*, 284:943–970, 2015.
- [19] J. A. Cottrell, A. Reali, Y. Bazilevs, and T. J. R. Hughes. Isogeometric analysis of structural vibrations. *Computer Methods in Applied Mechanics and Engineering*, 195:5257–5296, 2006.
- [20] J. Austin Cottrell, Thomas J. R. Hughes, and Yuri Bazilevs. *Isogeometric analysis toward integration of CAD and FEA*. John Wiley and Sons, 2009.
- [21] J.A. Cottrell, T.J.R. Hughes, and A. Reali. Studies of refinement and continuity in isogeometric structural analysis. *Computer Methods in Applied Mechanics and Engineering*, 196:4160–4183, 2007.
- [22] L. Beirão da Veiga, C. Lovadina, and A. Reali. Avoiding shear locking for the timoshenko beam problem via isogeometric collocation methods. *Comput. Methods Appl. Mech. Engrg.*, 241:38–51, 2012.
- [23] C. de Boor. *A practical guide to Splines (revised edition)*. Springer, 2001.
- [24] C. de Falco, A. Reali, and R. Vázquez. Geopdes: A research tool for isogeometric analysis of pdes. *Advances in Engineering Software*, 42:1020–1034, 2011.
- [25] S. de Miranda and F. Ubertini. Recovery of consistent stresses for compatible finite elements. *Computer Methods in Applied Mechanics and Engineering*, 191(15):1595–1609, 2002.
- [26] R.P. Dhote, H. Gomez, R N.V. Melnik, and J. Zu. Isogeometric analysis of a dynamic thermo-mechanical phase-field model applied to shape memory alloys. *Computational Mechanics*, 53:1235–1250, 2014.
- [27] John-Eric Dufour, Pablo Antolin, Giancarlo Sangalli, Ferdinando Auricchio, and Alessandro Reali. A cost-effective isogeometric approach for composite plates based on a stress recovery procedure. submitted, 2017.
- [28] T. Elguedj, Y. Bazilevs, V.M. Calo, and T.J.R. Hughes. B and f projection methods for nearly incompressible linear and non-linear elasticity and plasticity using higher-order nurbs elements. *Computer methods in applied mechanics and engineering*, 197:2732–2762, 2008.
- [29] J.J. Engblom and O.O. Ochoa. Through-the-thickness stress predictions for laminated plates of advanced composite materials. *International Journal for Numerical Methods in Engineering*, 21(10):1759–1776, 1985.

- [30] H. Gomez, V. M. Calo, Y. Bazilevs, and T. J. R. Hughes. Isogeometric analysis of the cahn–hilliard phase-field model. *Computer Methods in Applied Mechanics and Engineering*, 197:4333–4352, 2008.
- [31] H. Gomez, T.J.R. Hughes, X. Nogueira, and V.M. Calo. Isogeometric analysis of the isothermal navier-stokes-korteweg equations. *Computer Methods in Applied Mechanics and Engineering*, 199:1828–1840, 2010.
- [32] H. Gomez and L. De Lorenzis. The variational collocation method. *Computer Methods in Applied Mechanics and Engineering*, 309:152–181, 2016.
- [33] H. Gomez, A. Reali, and G. Sangalli. Accurate, efficient, and (iso)geometrically flexible collocation methods for phase-field models. *Journal of Computational Physics*, pages 153–171, 2014.
- [34] Y. Guo, A. P. Nagy, and Z. Gürdal. A layerwise theory for laminated composites in the framework of isogeometric analysis. *Composite Structures*, 107:447–457, 2014.
- [35] Y. Guo and M. Ruess. A layerwise isogeometric approach for nurbs derived laminate composite shells. *Composite Structures*, 124:912–917, 2015.
- [36] M.-C. Hsu, D. Kamensky, F. Xu, J. Kiendl, C. Wang, M.C.H. Wu, J. Minero, A. Reali, Y. Bazilevs, and M.S. Sacks. Dynamic and fluid-structure interaction simulations of bio-prosthetic heart valves using parametric design with t-splines and fung-type material models. *Computational Mechanics*, 55:1211–1222, 2015.
- [37] T.J.R. Hughes, J.A. Cottrell, and Y. Bazilevs. Isogeometric analysis: Cad, finite elements, nurbs, exact geometry, and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194:4135–4195, 2005.
- [38] T.J.R. Hughes, J.A. Cottrell, and Y. Bazilevs. Isogeometric analysis: Cad, finite elements, nurbs, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194(39-41):4135–4195, 2005.
- [39] T.J.R. Hughes, J.A. Evans, and A. Reali. Finite element and nurbs approximations of eigenvalue, boundary-value, and initial-value problems. *Computer Methods in Applied Mechanics and Engineering*, 272:290–320, 2014.
- [40] T.J.R. Hughes, A. Reali, and G. Sangalli. Duality and unified analysis of discrete approximations in structural dynamics and wave propagation: Comparison of p-method finite elements with k-method nurbs. *Computer Methods in Applied Mechanics and Engineering*, 197:4104–4124, 2008.
- [41] H. Kapoor, R.K. Kapania, and S.R. Soni. Interlaminar stress calculation in composite and sandwich plates in nurbs isogeometric finite element analysis. *Composite Structures*, 106:537–548, 2013.
- [42] J. Kiend, F. Auricchio, L. Beirão da Veiga, C. Lovadina, and A. Reali. Isogeometric collocation methods for the reissner-mindlin plate problem. *Computer Methods in Applied Mechanics and Engineering*, 284:489–507, 2015.
- [43] J. Kiendl, F. Auricchio, Thomas J. R. Hughes, and A. Reali. Single-variable formulations and isogeometric discretizations for shear deformable beams. *Computer Methods in Applied Mechanics and Engineering*, 284:988–1004, 2015.
- [44] J. Kiendl, K. U. Bletzinger, J. Linhard, and R. Wüchner. Isogeometric shell analysis with

- kirchhoff-love elements aortic valve closure. *Computer Methods in Applied Mechanics and Engineering*, 198:3902–3914, 2009.
- [45] J. Kiendl, K.-U. Bletzinger, J. Linhard, and R. Wüchner. Isogeometric shell analysis with kirchhoff-love elements. *Computer Methods in Applied Mechanics and Engineering*, 198:3902–3914, 2009.
- [46] R. Kruse, N. Nguyen-Thanh, L. De Lorenzis, and Thomas J. R. Hughes. Isogeometric collocation for large deformation elasticity and frictional contact problems. *Computer Methods in Applied Mechanics and Engineering*, 296:73–112, 2015.
- [47] S. Lipton, J.A. Evans, Y. Bazilevs, T. Elguedj, and T.J.R. Hughes. Robustness of isogeometric structural discretizations under severe mesh distortion. *Computer Methods in Applied Mechanics and Engineering*, 199:357–373, 2010.
- [48] J. Liu, H. Gomez, J.A. Evans, T.J.R. Hughes, and C.M. Landis. Functional entropy variables: A new methodology for deriving thermodynamically consistent algorithms for complex fluids, with particular reference to the isothermal navier-stokes-korteweg equations. *Journal of Computational Physics*, 248:47–86, 2014.
- [49] L. De Lorenzis, J.A. Evans, Thomas J. R. Hughes, and A. Reali. Isogeometric collocation: Neumann boundary conditions and contact. *Computer Methods in Applied Mechanics and Engineering*, 284:21–54, 2015.
- [50] C. Manni, A. Reali, and H. Speleers. Isogeometric collocation methods with generalized bsplines. *Computer and Mathematics with Applications*, 70:1659–1675, 2015.
- [51] M. Montardini, G. Sangalli, and L. Tamellini. Optimal-order isogeometric collocation at galerkin superconvergent points. *Computer Methods in Applied Mechanics and Engineering*, 316:741–757, 2017.
- [52] S. Morganti, F. Auricchio, D. J. Benson, F. I. Gambarin, S. Hartmann, T. J. R. Hughes, and A. Reali. Patient-specific isogeometric structural analysis of aortic valve closure. *Computer Methods in Applied Mechanics and Engineering*, 284:508–520, 2015.
- [53] N. J. Pagano. Exact solutions for rectangular bidirectional composites and sandwich plates. *Journal of Composite Materials*, 4:20–34, 1970.
- [54] S. Pauletti, M. Martinelli, N. Cavallini, and P. Antolin. Igatools: an isogeometric analysis library. *SIAM J. Sci. Comput.*, 37(4), 2015.
- [55] Les Piegl and Wayne Tiller. *The NURBS book*. Springer, 1995.
- [56] Jr. Pryor, C.W., and R.M. Barker. A finite-element analysis including transverse shear effects for applications to laminated plates. *AIAA Journal*, 9(5):300–309, 1971.
- [57] A. Reali. An isogeometric analysis approach for the study of structural vibrations. *Computer Methods in Applied Mechanics and Engineering*, 1–30:15–27, 2006.
- [58] A. Reali and H. Gomez. An isogeometric collocation approach for bernoulli-euler beams and kirchhoff plates. *Computer Methods in Applied Mechanics and Engineering*, 284:623–636, 2015.
- [59] A. Reali and T.J.R. Hughes. An introduction to isogeometric collocation methods. In S. Bordas (Eds.) G. Beer, editor, *Isogeometric Methods for Numerical Simulation*, volume 561 of *CISM International Centre for Mechanical Sciences*. Springer, Vienna, 2015.

- [60] J. N. Reddy. *Mechanics of Laminated Composite Plates and Shells: Theory and Analysis*, volume 2. 2004.
- [61] J.J.C. Remmers, C.V. Verhoosel, and R. de Borst. *Isogeometric analysis for modelling of failure in advanced composite materials*. Numerical Modelling of Failure in Advanced Composite Materials, Woodhead Publishing Series in Composites Science and Engineering. Woodhead Publishing, 2015.
- [62] D. Schillinger, J.A. Evans, A. Reali, M.A. Scott, and T.J.R. Hughes. Isogeometric collocation: Cost comparison with galerkin methods and extension to adaptive hierarchical nurbs discretizations. *Computer Methods in Applied Mechanics and Engineering*, 170–232, 2013.
- [63] C.T. Sun and Sijian Li. Three-dimensional effective elastic constants for thick laminates. *Journal of Composite Materials*, 22:629–639, 1988.
- [64] C.H. Thai, H. Nguyen-Xuan, S.P.A. Bordas, N. Nguyen-Thanh, and T. Rabczuk. Isogeometric analysis of laminated composite plates using the higher-order shear deformation theory. *Mechanics of Advanced Materials and Structures*, 55:451–469, 2015.
- [65] F. Ubertini. Patch recovery based on complementary energy. *International Journal for Numerical Methods in Engineering*, 59(11):1501–1538, 2004.
- [66] O. Weeger, S.K. Yeung, and M.L. Dunn. Isogeometric collocation methods for cosserat rods and rod structures. *Comput. Methods Appl. Mech. Engrg.*, 2016.