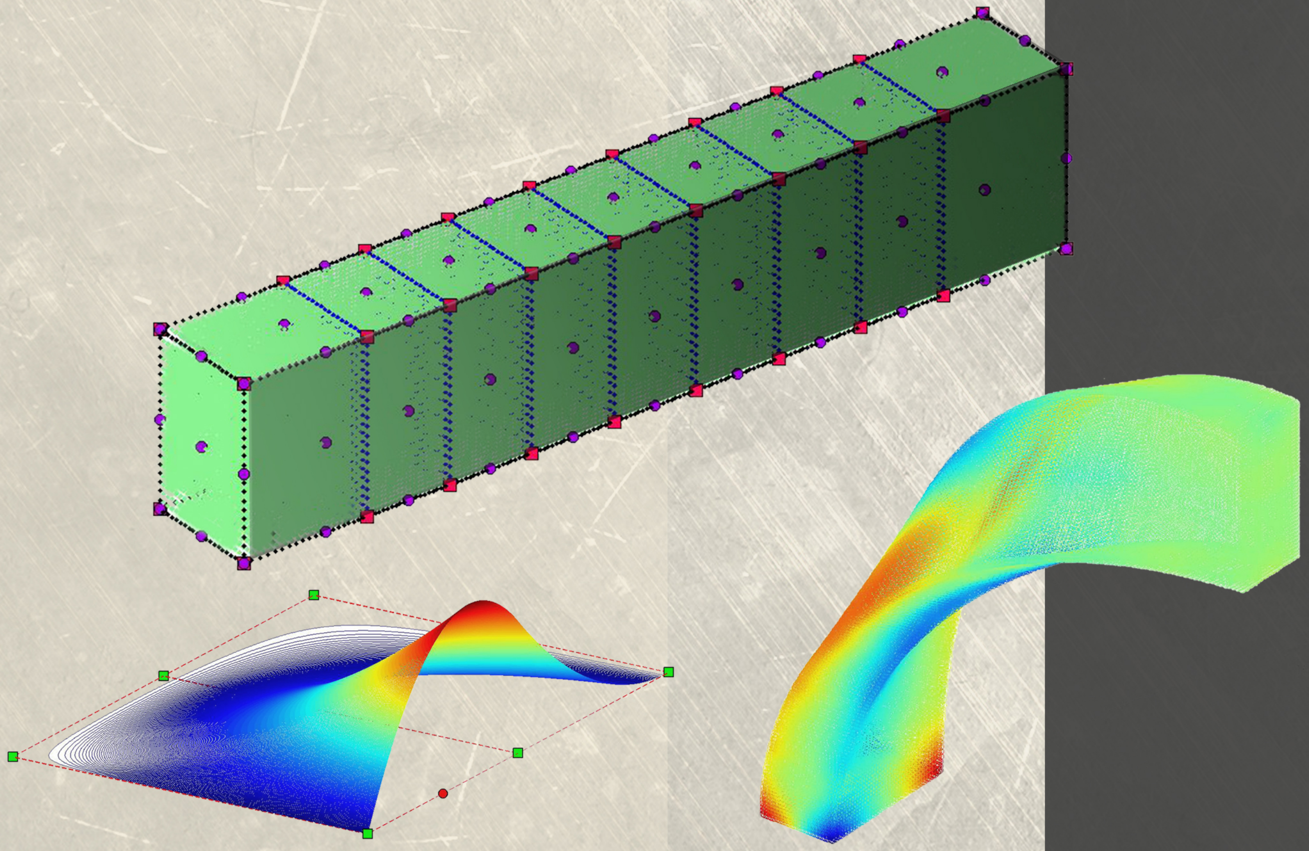
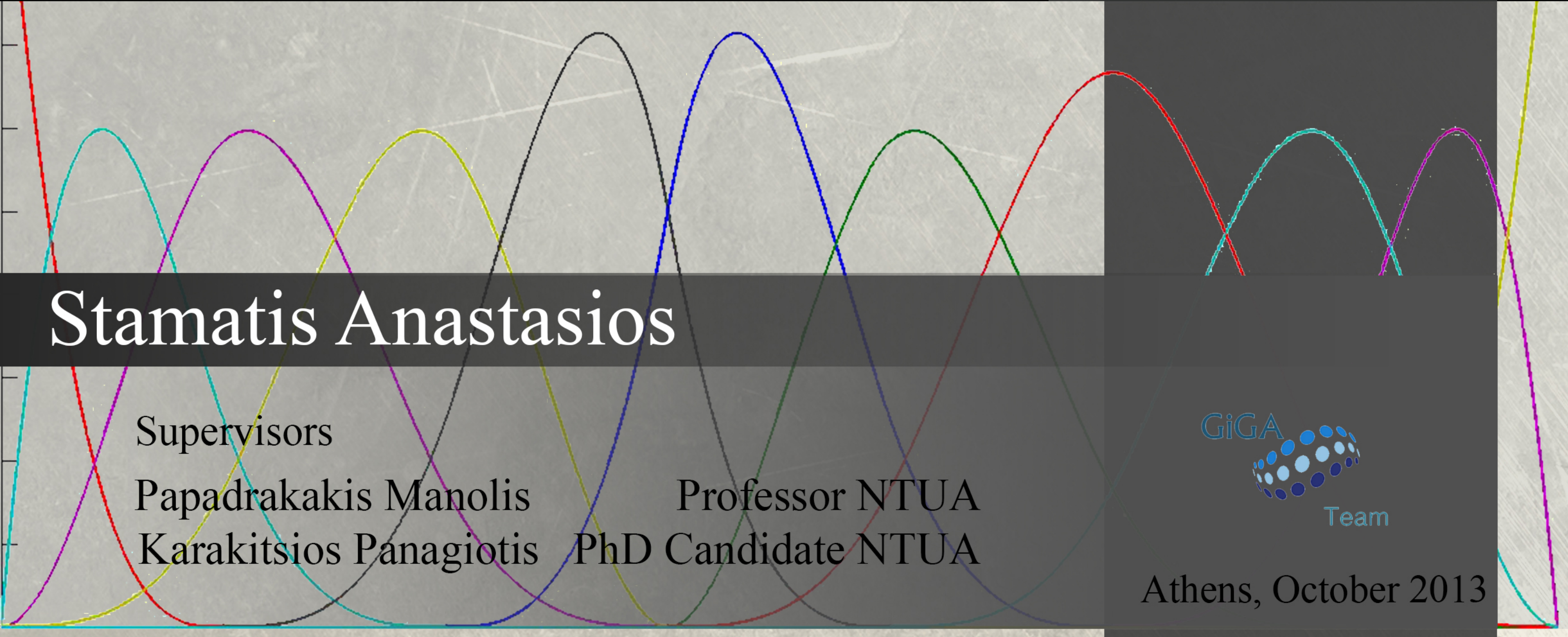
National Technical University of Athens
School of Civil Engineering
Department of Structural Engineering
Institute of Structural Analysis & Antiseismic Research

IsoGeometric Linear Static Analysis with NURBS





Stamatis Anastasios

Supervisors
Papadrakakis Manolis Professor NTUA
Karakitsios Panagiotis PhD Candidate NTUA

Athens, October 2013

Acknowledgements

I would like to express my sincerest gratitude to all the people that contributed to the creation of this thesis. First of all, I would like to thank Professor **Manolis Papadrakakis** for the opportunity to get involved with Isogeometric Analysis and the research experience in NTUA. I would also like to acknowledge PhD Candidate **Panagiotis Karakitsios,** for his collaboration the last two years and his useful guidance and advice all this time. The result presented today would not be the same without him. I would like to show my appreciation to the researchers **Manolis Trypakis**, **Dimitrios Tsapetis**, **Antonios Iakovos**, **Sofia Korlou** and **Athanasios Leontaris** for the many hours they devoted in the careful review of my thesis. Moreover, I would like to thank **Panagiota Stephanou** for the significant contribution to the composition of this thesis and, last but definitely not least, the architecture students **Maria Theodorou** and **Evangelia Stamati** for their artistic point of view in the careful figure revision and creation of the cover.

Ευχαριστίες

Θα ήθελα να εκφράσω την αμέριστη ευγνωμοσύνη μου σε όλους όσους συνέβαλλαν στη δημιουργία αυτής της διπλωματικής. Πρώτο θα ήθελα να ευχαριστήσω τον Καθηγητή **Μανώλη Παπαδρακάκη** για την ευκαιρία που μου προσέφερε να ασχοληθώ με την Ισογεωμετρική Ανάλυση και την εμπειρία της έρευνας στο ΕΜΠ. Θα ήθελα επίσης να ευχαριστήσω τον Υποψήφιο Διδάκτωρ **Παναγιώτη Καρακίτσιο** για την συνεργασία μας αυτά τα δύο χρόνια, την καθοδήγησή του και τις συμβουλές του. Το σημερινό αποτέλεσμα δεν θα ήταν το ίδιο χωρίς τη συνεισφορά του. Θερμές ευχαριστίες και στους ερευνητές **Μανώλη Τρυπάκη**, **Δημήτριο Τσαπέτη**, **Αντώνη Ιάκωβο**, **Σοφία Κορλού** και **Αθανάσιο Λεοντάρη** για τις πολλές ώρες που αφιέρωσαν στην προσεκτική επιμέλεια των κειμένων. Ένα ακόμα «ευχαριστώ» στην **Παναγιώτα Στεφάνου** για την σημαντική βοήθειά της στη συγγραφή αυτής της διπλωματικής και στις φοιτήτριες αρχιτεκτονικής **Μαρία Θεοδώρου** και **Ευαγγελία Σταμάτη** για την καλλιτεχνική τους συμβολή στην προσεκτική αναθεώρηση των εικόνων και τη δημιουργία του εξωφύλλου.

Abstract

The scope of this thesis is the investigation of Linear Static Isogeometric Analysis, an innovative methodology of complete CAD – CAE integration introduced by J. Austin Cottrell, Thomas J.R. Hughes and Yuri Bazilevs. NURBS models have been analyzed assuming linear elasticity. Finite Element Method and Non-Uniform Rational B-SPLines have been examined separately, as the two components of Isogeometric Analysis. The routines, which have been used for the linear static analysis of the presented applications, were developed in the high-level language and interactive environment "MATLAB". Additionally, geometrical representations were acquired through the NURBS-based software "Rhinoceros 3D". The topics considered are B-SPLine and NURBS entity properties, Refinement techniques, Stiffness Matrix Formulation, result post-processing (displacement, stress, strain field) and linear 2D and 3D applications investigating models of various representations.

Σύνοψη

Ο στόχος της παρούσας διπλωματικής είναι η διερεύνηση της Γραμμικής Ισογεωμετρικής Ανάλυσης, που αποτελεί μια καινοτόμο μεθοδολογία πλήρους ενσωμάτωσης των τεχνολογιών CAD – CAE και πρωτοπαρουσιάστηκε από τους J. Austin Cottrell, Thomas J.R. Hughes και Yuri Bazilevs. Η ανάλυση πραγματοποιήθηκε σε προσομοιώματα NURBS με την παραδοχή γραμμικής ελαστικότητας. Η Μέθοδος των Πεπερασμένων Στοιχείων και τα Non-Uniform Rational B-SPLines μελετήθηκαν ξεχωριστά, ως τα δύο συστατικά της Ισογεωμετρικής Ανάλυσης. Οι ρουτίνες που χρησιμοποιήθηκαν για την γραμμική στατική ανάλυση των εφαρμογών αναπτύχθηκαν στην γλώσσα προγραμματισμού υψηλού επιπέδου και το διαδραστικό περιβάλλον "MATLAB".Επιπρόσθετα, οι γεωμετρικές αναπαραστάσεις παράχθηκαν μέσω του NURBS-based λογισμικού "Rhinoceros 3D". Τα θέματα που εξετάσθηκαν αφορούν ιδιότητες των οντοτήτων B-SPLine και NURBS, τεχνικές Διακριτοποίησης, Μόρφωση του Μητρώου Στιβαρότητας, επεξεργασία των αποτελεσμάτων (πεδία μετατοπίσεων, τάσεων, παραμορφώσεων) και γραμμικές εφαρμογές 2D και 3D με τη διερεύνηση διάφορων αναπαραστάσεων.

Prologue

My daily routine as a student in School of Civil Engineering, National Technical University of Athens was very different two years ago. It all began in 2011, a few days after the end of the summer exam period. Professor Manolis Papadrakakis, who taught “Statics III – the Direct Stiffness Method” at this semester, distinguished me as the student with the best project for that year and proposed the beginning of a cooperation that would lead to the composition of my thesis.

After familiarizing with the FEM basics, I began my cooperation with PhD Candidate Panagiotis Karakitsios, who is involved in advanced computational methods, and more specifically Isogeometric Analysis. I had the honor to be the first undergraduate member of GiGA Team, a research team that consists of very talented researchers. Over the past two years, I was introduced into the research community, a different way of thinking and collaborators of remarkable intellect and extraordinary teamwork. I was provided the chance to use my creativity and give something in return to my University.

Institute of Structural Analysis and Antiseismic Research at School of Civil Engineering provides an excellent starting point for young engineers. In the middle of a difficult economical situation, Centers of Excellence are a hopeful prospect for the aspiring researchers. GiGA Team provides the researchers with a chance to work in a modern, innovative environment, discover and create the world of tomorrow. I am very proud to be a member of this Team.

Stamatis Anastasios
Athens, October 2013

Table of Contents

1 The concept of Isogeometric Analysis

1.1 Finite Element Method

1.1.1 Historical Overview

Isogeometric Analysis as a historic evolutionary computational mechanics achievement

Isogeometric Analysis is an innovative method, which integrates design and analysis in the greatest scale ever achieved. It was conceived by Thomas J.R. Hughes in 2003, Professor of Aerospace Engineering and Engineering Mechanics of the University of Texas at Austin.

Thomas J.R. Hughes is one of the greatest experts worldwide in Computational and Applied Mathematics. He began his career as a mechanical design engineer at Grumman Aerospace and then went on to General Dynamics as a research and development engineer. After receiving his Ph.D. from the University of California at Berkley, he joined the faculty and moved on to California Institute of Technology. Afterwards, he was hired by Stanford University before joining The University of Texas at Austin.

Figure 1.1. Thomas Joseph Robert Hughes.
(http://users.ices.utexas.edu/~hughes/)

Isogeometric Analysis is set to bridge the gap that exists between Computer-Aided Design and Computer-Aided Engineering. Thomas J.R. Hughes published a book in 2009, along with J.A. Cottrell and Y. Bazilevs, explaining the fundamentals of this new method. The book, “Isogeometric Analysis: Towards Integration of CAD and FEA” is the first book to be issued on this new field of study and a trustworthy guide to the researchers that want to invest in this method. It contains a vast number of applications of Isogeometric Analysis, potential resources for the new researcher, advantages of the new method and future fields of study. The application of Isogeometric Methods can lead to results and improvements in computational mechanics, structural statics and dynamics and biomechanics.

The need for Isogeometric Analysis

The compelling need for a new method had to be met; greater challenges in Finite Element Analysis were arising every day, demanding faster and more precise results. Even nowadays, although structures worldwide are designed using Finite Element Method instead of the traditional by-hand ways, design errors cannot be avoided. Since 2000, structural collapse cost humanity over 1500 lives; many of them could have been spared, had the engineer a more accurate tool for analysis and design. Unfortunately, present analysis technologies require a lot of man-hours for manual generation of approximated, FEM-suitable geometries and consequently they derive the engineer from his main task and force him to devote less time in result evaluation. The risk is even greater, considering the mistakes the engineer can make during this transformation. Furthermore, the approximation of the geometry sometimes is clearly not enough for the desired convergence. These are the gaps that Isogeometric Analysis is set to fill. In order to fully understand Isogeometric Analysis, one has to acknowledge the evolution of Analysis throughout its history and understand the principles this revolutionary method is based on. This introduction provides the historical review of the technologies and the requirements of structural analysis that led to the creation of this new method.

Before Finite Element Method

Engineers of the past had to meet the demands for structural analysis and representation of accurate results. Construction always needed the cost-efficiency provided by design. In the early years, the only weapon the engineer possessed was his mind. In order to solve a statically indeterminate structure, Equilibrium, Constitutive and Compatibility Laws had to be applied. Several methods existed in order to solve the problem.

The Force Method used the Betti-Maxwell Theorem in combination with virtual works in order to provide support forces and moments. The main idea is the preservation of Equilibrium and calculation of forces, in order to ensure Compatibility.

The Displacement Method, on the other hand, ensures Compatibility is maintained and Equilibrium is achieved via calculation of displacements.

The Moment Distribution Method, also known as the Cross Method, relies upon computational iteration cycles to an initial moment distribution, until the desired approximation is achieved. It only yields results for bending effects and ignores axial and shear tension, but its efficiency made it very popular among the engineers in the 1930s.

These methods had their limits; a vast amount of time was required for the solution and many errors would occur in the manual computational process. More complex problems could only be approximated and sometimes analysts had to solve differential equations by hand in order to obtain the solution.

Finite Element Method

Structural Analysis has been a major part of the engineering field of practice. The knowledge of a structure's reaction to certain loads enhances its safety and makes it cost-effective. It has a wide field of application, including buildings, bridges, airplanes, space shuttles, ships, satellites, nuclear stations and much more. At first, engineers used methods obtained from the solution of differential equations in order to evaluate the stress, strain and displacement conditions of the structure. Structural mechanics theorems were developed and used in order to solve the computational problem. These served their purpose well for relatively simple, everyday linear problems. However, new technologies emerged and the constant demand for faster, more accurate solutions to complicated problems had to be met.

Early computer models had made their appearance and engineering scientists were eager to use them in their problems. The birth of Finite Element Methods can be placed at the late stages of World War II. Structural Engineers working for the Royal Aeronautical Society of London had to design an innovative type of combat jet aircraft whose speed required swept-back wings. Unfortunately, none of the existing theories could fit to solve such a complex problem. The failure of the German ME262 was palpable proof of that matter.

This challenging task was assigned to one of the brightest minds of the Royal Aeronautical Society of London, John Argyris. Argyris was born in Greece, Volos in 1913 and had studied Civil Engineering at National Technical University of Athens.

Figure 1.2. John Argyris.
(http://www.nae.edu/27953.aspx)

Figure 1.3. City of Volos. Central Greece. John Argyris Birthplace.
(http://www.greekscapes.gr/~landscapesatlas)

John Argyris graduated from Technical University of Munich in 1936 and had begun to work in industrial applications of complex structures. He remained in Germany at the beginning of World War II and was accused of giving research info to the Allies, arrested and sent to a concentration camp when the Axis invaded Greece. He was rescued by a German Admiral, Kanaris, who was of Greek ancestry as well. After breaking out from the concentration camp, Argyris escaped to Switzerland by swimming through the Rhine river, in the middle of a raid holding his passport with his teeth. He finished his Doctoral Degree in Aeronautics from ETH, Zurich in 1942. Afterwards he moved to England and was engaged with the Royal Aeronautical Society of London, working as a technical officer. As a researcher, he was really skeptical with Cartesian coordinate systems and the way they were used in engineering. He believed that triangular and tetrahedral elements were far more suitable for engineering applications. John Argyris was not devoted to everyday problems, but would rather get busy with difficult, apparently unsolvable problems. His superiors quickly recognized this trait and he in return welcomed the challenges he was given, including the swept-back wings aircraft problem.

It seems he was the right man for the job. In August 1943, after 3 days and nights of devotion to the problem, he had a breakthrough. He used triangular elements to simulate the swept-back wings and solved the model in the electro-mechanical computing device the Society had recently acquired. The device was able to solve an equation with up to 64 unknowns. Analysis results were very close to the experimental results, with a deviation of approximately 8 %. This was the birth of the Finite Element Method (FEM). All relative papers were at once labeled “secret”. This innovative method included a different measurement of stresses and strains, diverging from the classical Cartesian field and was proving to be useful and easily generalized.

In the following years, the “Matrix Force and Displacement Method”, mostly known as “Finite Element Method” (or, as Ray Clough wrote in 1960, “The Argyris Method”) was developed by many researchers, including Turner, Clough, Zienkiewicz and Cheung. Argyris resumed his academic career with drastic contributions to the research and development of Finite Element Method as well as many other aspects of the Engineering field until he was 88 years old. He invented, among others, the triangular element TRIC and is also well known for the contribution to the solution of the heat protection problem for the NASA space shuttle during the entrance in the atmosphere. He passed away in April 2004 [15].

Due to the swift evolution of computational speed and memory capacity, FEM became very popular within the engineering industry. Millions of dollars were invested in its development. New FEM technologies emerged, such as the isoparametric elements. These allowed for a more general approach and a better adaptation to complex geometries.

NTUA Professor Papardakakis Manolis has devoted his career in the evolution and outspread of FEM technologies. Nowadays, in the middle of the financial crisis in Greece, NTUA's Institute of Structural Analysis and Antiseismic Research has a remarkable research portfolio to show, always being up to date with the latest technological trends. Bright young minds are given a chance to shine, in National Technical University of Athens, continuing and improving the cycle of expanding the boundaries of human knowledge.

In the dawn of 2000, the structural engineering field has changed drastically. Personal computers and a variety of FEM software are now available to engineers. All the hard work done so many years ago by hand is now avoided. Greater speeds and bigger rates of convergence are achieved every other month. Problems once thought to remain unsolved now seem common and relatively easy. Finite Elements are used in a wide range of computational analysis, such as structural and dynamic analysis, fluid mechanics, biomechanics, earthquake engineering and many more. The modern engineer does not need to solve complex mathematical equations by hand, but has to pursue a global and thorough understanding of his field of study as well as knowledge of the innovative computational methods.

The engineering software market consists of many products devoted to the analysis of FEM models. There are generalized and more theoretical software that can solve almost any type of structure. NASTRAN, a widely used FEM platform, was originally developed by NASA in the 1960s in order to cover the Agency's special needs [17]. Simulia Abaqus, originally released in 1978, was developed using an open-source language, Python and was initially intended for non-linear problems. It is particularly popular due to its wide range of Modeling capabilities, both for linear and non-linear problems [16]. ADINA (Automatic Dynamic Incremental Non-Linear Analysis) , first developed in 1974, is used in a wide range of non-linear problems. It has applications in static and dynamic analysis, heat transfer, compressible and incompressible flows and electromagnetic phenomena [19]. FEMAP (Finite Element Modeling and Post-processing) is used as an input creation and output processing tool for the engineers. It cooperates with the solver routines from other platforms (e.g. NASTRAN) and focuses on the easy and accurate communication between software and user [20].

Specialized software is also available in the engineering market. The specific characteristics and complexity of today's structures require a more personal and delicate approach. ATENA, standing for Advanced Tool for Engineering Non-Linear Analysis, is specialized in everything to do with reinforced concrete structures [18]. SOFiSTIK, first used in 1987, is directed towards bridge linear and non-linear analysis [22]. Furthermore, there are platforms dedicated to the special needs of the Greek market. Designing structures in the country with the biggest seismic activity in Europe is not an easy task. StereoSTATIKA is suitable for reinforced concrete analysis in countries with dangerous seismic activity [21]. FESPA is a Greek software dedicated to analysis and design of reinforced concrete and steel buildings [23].

1.1.2 Basic Idea of Finite Element Method

According to [2], the Basic Idea behind Finite Element Method is the approximation of the solution field via piecewise polynomial functions, called the shape functions N. Displacement values U at any internal point of the element can be computed from displacements at nodes d :

$$\underset{(3\times1)}{\{U(X,Y,Z)\}} = \underset{(3\times3n_e)}{[N(X,Y,Z)]}\cdot \underset{(3n_e\times1)}{\{d\}}$$

where n_e is the number of the finite elements.

A generalization for the whole structure leads to

$$\underset{(3\times1)}{\left\{U^S(X,Y,Z)\right\}} = \underset{(3\times3n)}{\left[N^S(X,Y,Z)\right]}\cdot \underset{(3n\times1)}{\left\{d^S\right\}}$$

The problem is directly downsized from infinite unknowns to a finite number of degrees of freedom. The next step is to define stress and strain matrices and their interconnection, which represents Hooke’s Constitutive Law.

$$\{\sigma\} = \begin{bmatrix} \sigma_X \\ \sigma_Y \\ \sigma_Z \\ \sigma_{XY} \\ \sigma_{YZ} \\ \sigma_{ZX} \end{bmatrix} \quad \{\varepsilon\} = \begin{bmatrix} \varepsilon_X \\ \varepsilon_Y \\ \varepsilon_Z \\ \gamma_{XY} \\ \gamma_{YZ} \\ \gamma_{ZX} \end{bmatrix} \quad (\text{3D case})$$

$$\{\sigma\} = [E]\cdot\{\varepsilon\}$$

Deformation Matrix [B] evaluates strains anywhere in the model from nodal displacements.

$$\{\varepsilon(X,Y,Z)\} = [B(X,Y,Z)]\cdot\{d\}$$

Using internal and external virtual work equilibrium, we can evaluate the Stiffness Matrix [k] for each element

$$\underset{(3n_e\times3n_e)}{[k]} = \int_V \underset{(3n_e\times6)}{[B]}^T\cdot \underset{(6\times6)}{[E]}\cdot \underset{(6\times3n_e)}{[B]}\, dV$$

In the same manner, distributed loads can be transformed into equivalent nodal loads.

$$\underset{(3n_e\times1)}{\{r\}} = \int_V \underset{(3n_e\times3)}{[N]}^T \cdot \underset{(3\times1)}{\{f\}} dV$$

The contribution of each element is added to the global matrices, producing the global Stiffness Matrix [K] and the Force Vector {R}. The Displacement Matrix can now be calculated from:

$$\underset{(3n\times1)}{\{R\}} = \underset{(3n\times3n)}{[K]} \cdot \underset{(3n\times1)}{\{D\}}$$

where n is the number of the nodes.

We can observe that (naturally) shape functions [N] and deformation matrix [B] depend on the Cartesian coordinates. This leads to iteration problems when complex geometries (e.g. circles or conic sections in general) are involved. The solution, which contributed to the further generalization of FEM, along with a more efficient iteration algorithm, is isoparametric elements. The basic idea is the existence of a parent element in the parameter space, which can be modeled as a regular shape (e.g. a cube, square or an equilateral triangle). Each element in the physical space (the “real” modeling space) can be described using a linear combination of the parent element’s shape functions. Hence, geometry can be approximated by the same functions used for the solution field. This explains the name “isoparametric”.

1.1.3 Drawbacks

Despite the evolution of FEM all this time, some problems have yet to be solved. For one, even isoparametric elements can only produce an approximation of geometry. The most challenging tasks of the day often require exact geometrical representation in order to achieve the necessary accuracy. After the meshing has been completed, the initial geometry plays no more role in the analysis procedure. This is intuitively worrying, to begin with. Furthermore, it produces a vast number of problems. The inevitable geometrical approximation means there will be convergence errors by definition, regardless of the solution methods and the available computational power. This affects the efficiency of the computational methods used for the solution.

If a finer mesh is required, refinement algorithms will return to the initial geometry and produce a different approximation. The new, fine mesh cannot be directly produced from the coarse mesh. Efficiency is certainly at a low, as procedures already completed have to be repeated in order for the new mesh to be created. Precious analysis time is required and the geometrical differences between the coarse and fine mesh make it difficult to compare the results. Hierarchical structures provide even greater challenges. Hierarchical refinement is considered an efficient refinement technique, as it focuses on the crucial areas of the model, but it is not easily applicable in FEM meshes.

The lack of integration between geometry and mesh generation is crucial. Computational geometry provides simply an input file for the meshing of the finite element model. Exact geometrical representation is not reflected in the new mesh, nor is the smoothness of the initial model, which leads to a slightly different model that analysis solves. Even for a small change in the geometrical model, reintegration and new mesh generation is required. This process is frustrating and pointless for the modern engineer; instead of devoting his time in creativity and design, he has to undergo the mundane task of regenerating a slightly different mesh over and over again. Many design errors are attributed to this task. In structural design, a vast amount of time is spent on these integration cycles between initial, preliminary and final design phases. The risk is even greater in complex and innovative structures, where the engineers cannot know beforehand what structural results to expect and have almost no way of checking the accuracy of the analysis.

Computational geometry has evolved since its birth; new and optimized geometrical structures are being used more often nowadays. Finite Element geometries fail to keep up with that pace and as a result, Computer-Aided Engineering is separated even more from Computer-Aided Design. Finite Elements cannot cooperate with the modern technologies of T-SPLines and Subdivision Surfaces. These problems had always been present throughout the history of Finite Elements. Complicated computational methods and algorithms have been developed in order to overcome them. The problem is that the nature of FEM does not allow for significant steps toward CAD-FEM integration. Improvements to the basic structure of Finite Elements are difficult and quite inefficient.

Figure 1.4 depicts a wine glass. In order to create this object, a designer has to define the following variables:

- degree of shape functions for each parametric axis
- knot value vector for each parametric axis
- control points (Cartesian coordinates and weights).

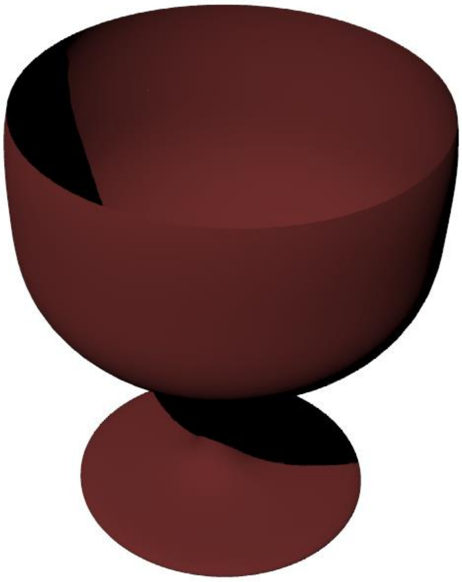


Figure 1.4. Wine glass. Geometry Design.

As a result, a mesh of finite elements is created, the “geometry mesh”, which surprisingly is not used by FEM software, which instead creates a new approximate one. Figure 1.5 represents the initial geometry mesh of the wine glass with the corresponding control net.

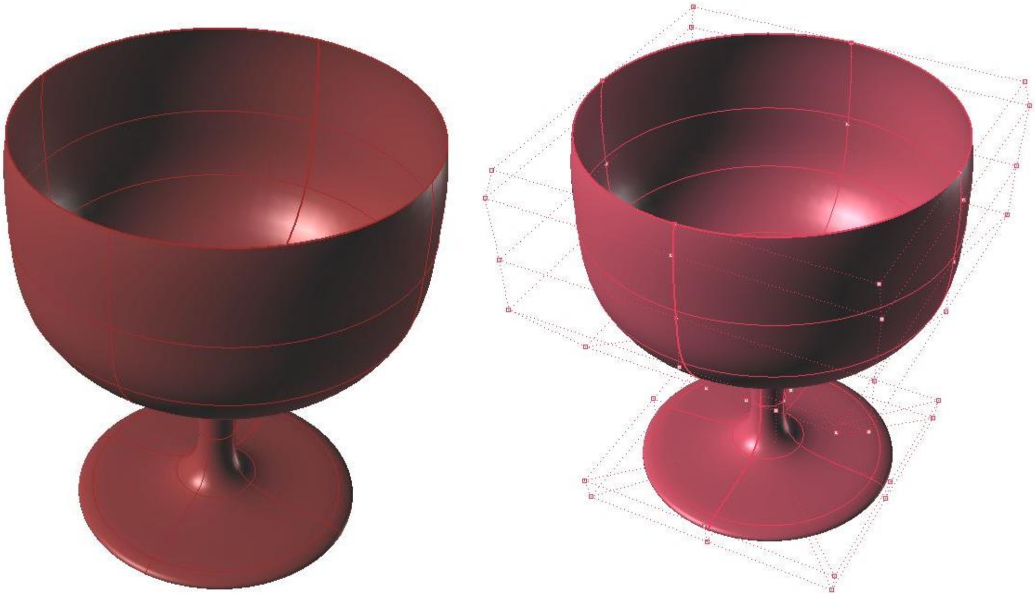


Figure 1.5. Wine glass. Geometry Initial Mesh.

With the previous initial mesh, a designer can represent exactly the wine glass, but an engineer cannot analyze it accurately and has to apply refinement in order to increase the accuracy of the solution field. Figure 1.6 shows a finer mesh of the wine glass for both cases. In FEA, the smooth surface of the cup participates only as input in FEM software and is thereafter replaced by quad finite elements. In IGA, the geometry remains intact and the mesh is the exact geometrical model.

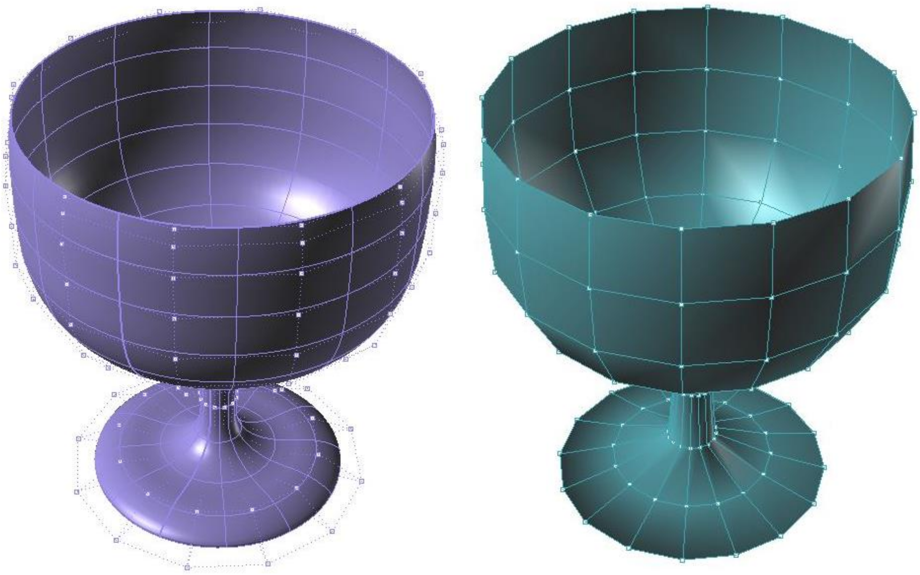


Figure 1.6. Wine glass. Fine FEA Mesh (left one). Fine IGA Mesh.

1.2 Computer Aided Design

1.2.1 Historical Overview

Isogeometric Analysis as a historic evolutionary computational design achievement

The evolution of computing systems made the option to design on a platform very attractive. Drafts could be edited easily and data could be stored and transferred at much higher speeds. Computer Aided Design has many applications in today’s world and a huge industrial support. Computer-generated imagery (CGI) is used in movies even more often; 3D and 2D cartoons are drafted and animated through computer software. Engineers draft complex designs such as cars, space shuttles, long span bridges and so on, one piece at a time in a computer. All the drafts can be edited and the escalation to optimized drafting is easier than ever. Designers’ time is now being devoted to creative thinking and taking ideas to the next level, rather than useless drafting by hand for hours. However, there is still room for improvement. Computational geometry is involved in a vast number of engineering applications and should not be considered independently. Design entities are supposed to cooperate with Finite Element Methods. Reinventing and improving computational geometry structures is the first step in completing this task. In order to understand the modern and future world of Computer-Aided Design, one has to study the history, the creation and the necessities that led to the creation of computational geometry entities.

Computer Aided Design (CAD) emerged in the 1950s from the automotive, shipyard and aircraft industries. In those times, designers were able to produce accurate drafts by hand, but when ship cross sections had to be drafted in real-life size, pencils could not help anymore and things became a bit more complex. The main problem was the definition of a real-size curve, which smoothly interpolated several predetermined points in order to create the shell of the ship. This task was usually carried out in the loft of a building, due to the large amount of space needed. The loftsman, as he was called, used easy-to-bend pieces of steel or wood, the spline, in order to interpolate the points. In order to maintain the spline’s shape, he usually put weights on them on certain points. Anything ring a bell so far [24]?

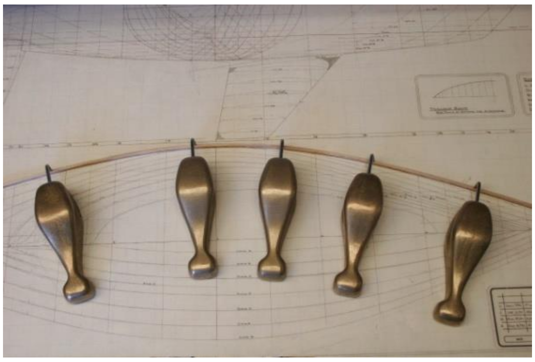


Figure 1.7. Original Splines and Weights (called Ducks).
(http://www.boatdesign.net)

The development of NURBS arose from the need to effectively represent freeform surfaces. Two engineers in France stood at the forefront of this approach, Pierre Bezier from Renault and Paul de Casteljau from Citroen. Bezier's work was the first to reach publication, and soon after the CAD industry started using and enhancing Bezier curves. However, certain disadvantages of Bezier curves led to the search for more convenient forms of representation. Researchers finally introduced B-SPLines, which were similar to Bezier curves, meaning that the curve was defined by a set of points, called the "control points", but their number was independent of the polynomial order of the curve. B-SPLines were a generalization of Bezier curves, but they were also more convenient to edit; changing a point no longer changed the whole curve, but only part of it.

Another problem is that even B-SPLines cannot produce an exact representation of conic sections. This is where NURBS came along. Ken Versprille was the first to work with NURBS on his dissertation in 1975. Later, when he acquired a top position in Computervision, the company began to support NURBS. Boeing, in its ambitious project to create a single curve representation that included Bezier curves and conic sections, became the first to industrialize NURBS [26].

After that, NURBS began to spread across the CAD industry. They possessed a lot of interesting attributes. The parameterization of the whole curve was downsized to a few control points coordinates, numerically stable mathematical procedures and easy modification. Cartoon characters, videogame graphics, ships, cars, airplanes were designed using NUBRS. This led to major investments from research and industrial faculties. Graphic designers became accustomed to them and students were taught about the theory and implementation of NURBS in real-life problems. This cycle led to the increasing popularity of Non-Uniform Rational B-SPLines in the CAD industry [25].

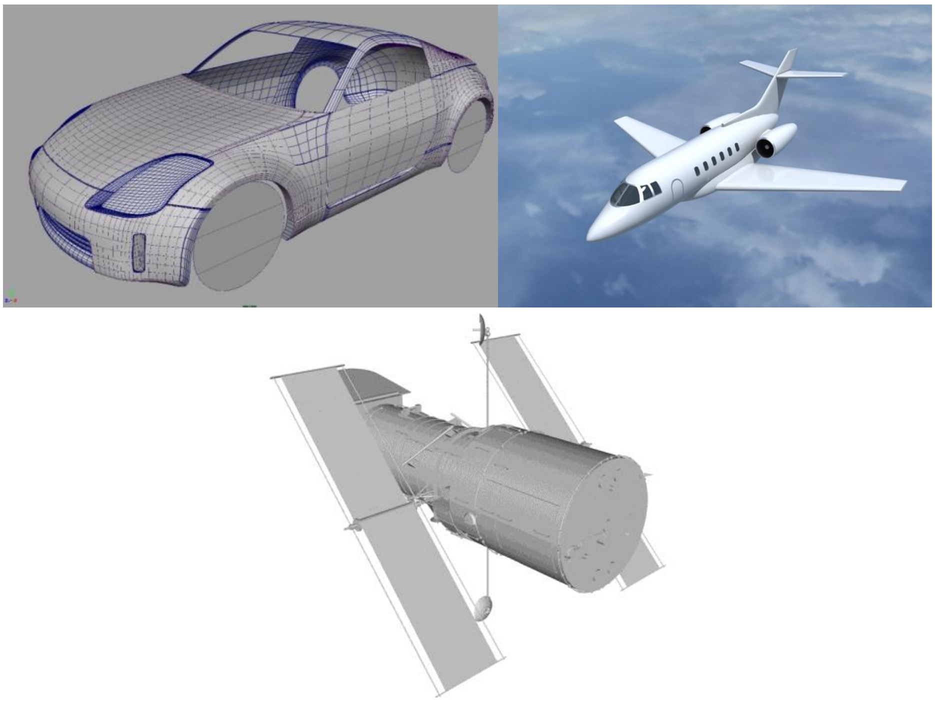
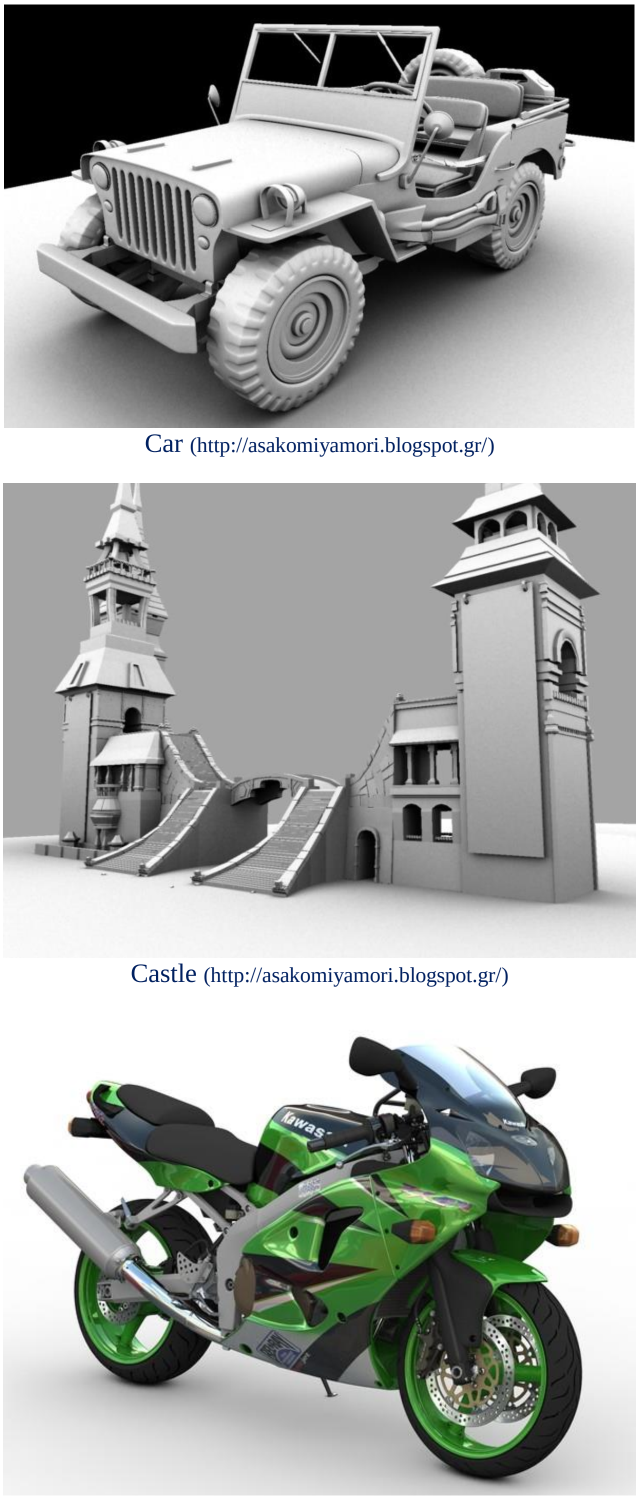


Figure 1.5. Entities created by NURBS.
(http://www.smcars.net/threads)(http://www.turbosquid.com)

Today, even though many other forms of more suitable representations exist, such as T-SPLines, Polycube SPLines and Subdivision surfaces, NURBS still hold a large share of the market. Many platforms exist for NURBS and designers still find it easier to use them. Many handful tools have been developed over the years for them (knot insertion, order elevation, curve fitting, patching). This makes them more attractive and easy-to-use than newer techniques with less industry experience behind them.



Car (http://asakomiyamori.blogspot.gr/)

Castle (http://asakomiyamori.blogspot.gr/)

Motorbike (http://kawasaki.turbosquid.com/3d-Models/3ds/max/xsi/c4d/obj)

Figure 1.6. Entities created by Subdivision Surfaces.

1.3 Isogeometric Analysis

The usual design process requires both exact geometrical representation of the model and accurate engineering results. Unfortunately, computational geometry and Finite Element Method are represented in different file types and are not compatible to each other. The engineer has to create a model for FEM solution and the designer a model for CAD representation. Moreover, the typical design process is not straightforward. The designer produces CAD designs, which are transformed into FEM-compatible forms of representation by the analyst. After generating mesh and obtaining results, the analyst informs the designer of the appropriate changes in geometry. The designer then gives the new CAD model to the engineer, who has to regenerate the FEM model and the new mesh. This cycle of CAD/CAE interaction can go on multiple times. In complex projects each design consists of numerous CAD entities combined together and the integration process is estimated to take up at around 80% of the whole design time. Researchers around the world have been trying to achieve automatic CAD/CAE integration.

The main problem is that CAD and FEM, even though they refer to the same object, evolved differently. This incompatibility drove researchers into separate roads, building a wall between the two methods. Thomas J.R. Hughes, a Professor of Aerospace Engineering and Engineering Mechanics at the University of Texas at Austin, came up with a different point of view. Instead of trying to connect present CAD and CAE formulas, we should reinvent them in ways that enable the integration. This is the scope of Isogeometric Analysis.

The basic idea is to exploit the functions used for the exact geometrical representation in order to describe the solution field. Isogeometric Analysis extends, in essence, isoparametric elements, but the process of altering geometry for the sake of the solution approximation is reversed.

This leads to the creation of a single model, capable both of exact representation and analysis. Designers and engineers will be working on the same platform. Time for meshing and entity translation will be eliminated in an instant. This direct contact between analysis and geometry means that every single change can be integrated as soon as it happens, with no risk of errors or timely tasks involved. Most importantly, the designer has to follow the engineer’s perspective and vice versa; the modern designer has to learn how to help the engineer and the modern engineer has to learn the methods the designer is using.

Isogeometric Analysis brings together two very different technologies, combining their best points to one. This leads to a better adaptation both from engineers and designers. In order to understand and improve Isogeometric Analysis, it simply needs to improve its counterparts. Finite element and computational geometry codes need not change drastically. This makes the new technology even more attractive. Understanding the basics of an innovation and implementation in the daily routine is usually a difficult and time-consuming task.

There are many geometrical forms of representation suitable for analysis, such as NURBS, T-SPLines, Polycube SPLines and Subdivision surfaces. Each entity has its own advantages and drawbacks, but the variety provided ensures a vast number of alternatives to use, depending on the case. This ensures the generalization of Isogeometric Analysis method to even more complex geometries.

FEM's shape functions are defined only in the interior of the element. Each element has C^{-1} continuity in the edges. IGA's shape functions are not contained in one element. Most of the times, they are defined through many elements. This ensures a greater continuity and interconnectivity. This different approximation works better and leads to greater convergence than the classical methods.

Refinement by order elevation or knot insertion has always been important for computational geometry. Hierarchical adaptation has been developed for a vast number of entities. All these technologies can be exploited by IGA. Hierarchical structures can be easily developed, straight from the geometrical model. Meshing and refinement is also immediately accomplished.

2 Basic Ingredients of Isogeometric Analysis

2.1 Introduction

Until recently, the majority of CAD software users had not realized that by designing a model, they simultaneously created its corresponding mesh of finite elements. This information, although redundant for designers devoted to computational geometry, is a revolutionary remark for the engineering community. Before Thomas J.R. Hughes' idea, known as IsoGeometric Analysis (IGA), engineers used to create a new approximate mesh instead of taking advantage of the existing accurate one. The additional geometry error makes the process less accurate, though more time-consuming. This observation seems now very obvious, but it took years of research until 2003, when Thomas J.R. Hughes and his research team succeeded to cut the Gordian Knot of CAD – CAE integration.

The main idea is the elimination of the node mesh in the analysis process. The role and properties of the node mesh are inherited by two separate meshes, obtained directly from the geometrical representation:

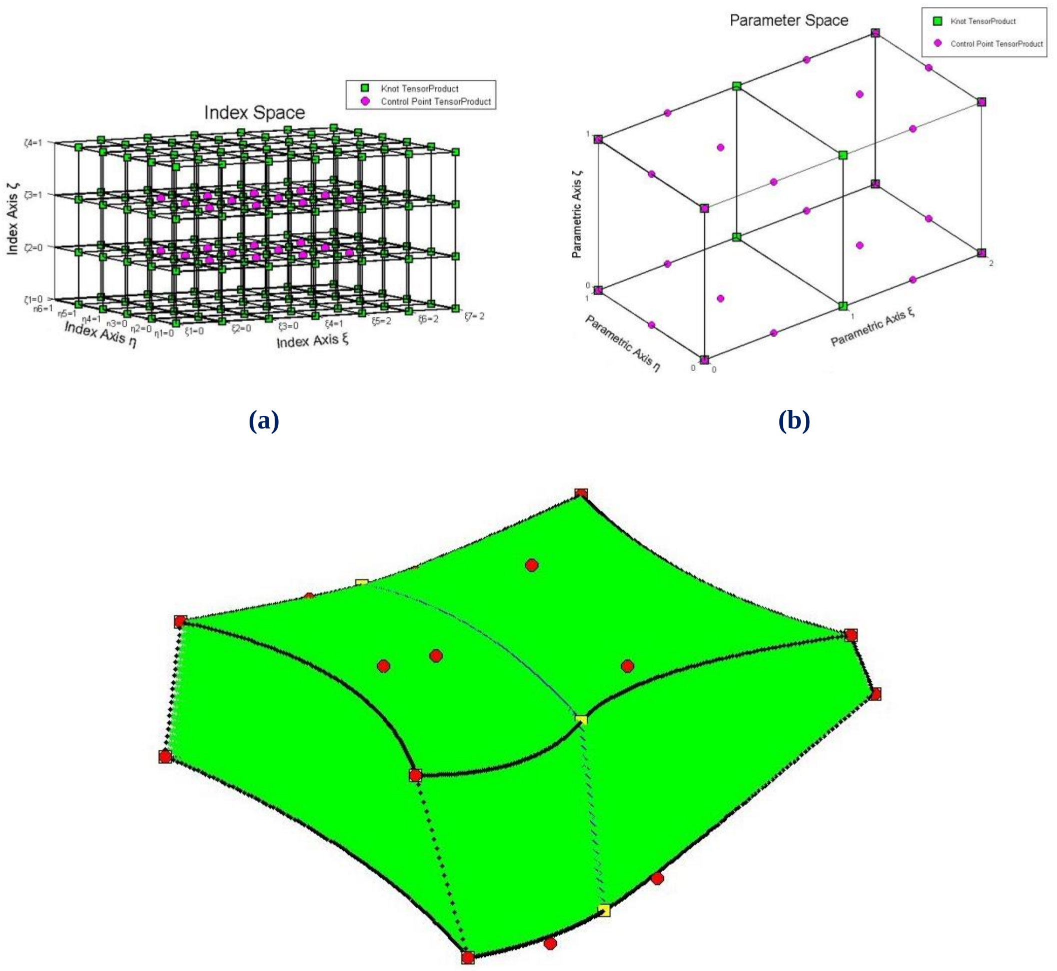
- The Control Point mesh, which defines geometry and the finite number of degrees of freedom that form the problem equation.

- The Knot mesh, which provides appropriate discretization for numerical integration and boundaries for Shape function influence in the model.

For the scope of this thesis, I have worked exclusively with Non-Uniform Rational B-SPLines (NURBS), as they are the most commonly used computational geometry technology. Despite the fact that quite more advanced SPLines have emerged, CAD industry still invests in NURBS. Since 1970, billions of dollars have been directed towards the outspread and evolution of NURBS, establishing them as a common tool for graphic representation around the globe. Both professionals and amateurs still use NURBS despite their disadvantages, such as difficulties in Patch connection and local Refinement. The reason for this is that NURBS are not only much more simple in their definition and use, but also able to represent with accuracy smooth curves and all conic sections.

2.2 Index, Parameter and Physical Space

In most occasions, the exact solution of a natural problem is neither possible nor necessary. The actual objective is to find an accurate solution that satisfies a selected convergence criterion. The ultimate challenge for an engineer is to balance between accuracy and time. Design and analysis of extraordinary geometries is a powerful asset for modern engineers, who are capable of facing surprisingly more complicated problems. Accurate geometrical representations of the natural model are designed in the familiar Cartesian system, called Physical Space. Additionally, it is very helpful to envision a complex structure in an imaginary, basic space, where all geometries can be represented as lines, rectangles and cuboids. This is Parameter Space. This approach is far from new; it is already known from the isoparametric concept in Finite Element Methods. The Parameter Space utilized in Isogeometric Analysis, however, holds some major differences. Furthermore, Isogeometric Analysis also introduces the Index Space. This additional space plays an important role for some kinds of SPLines, but it is only auxiliary for NURBS.



(a) **(b)**

(c)

Figure 2.1. B-SPLine solid.
(a) Index Space **(b)** Parameter Space
(c) Physical Space

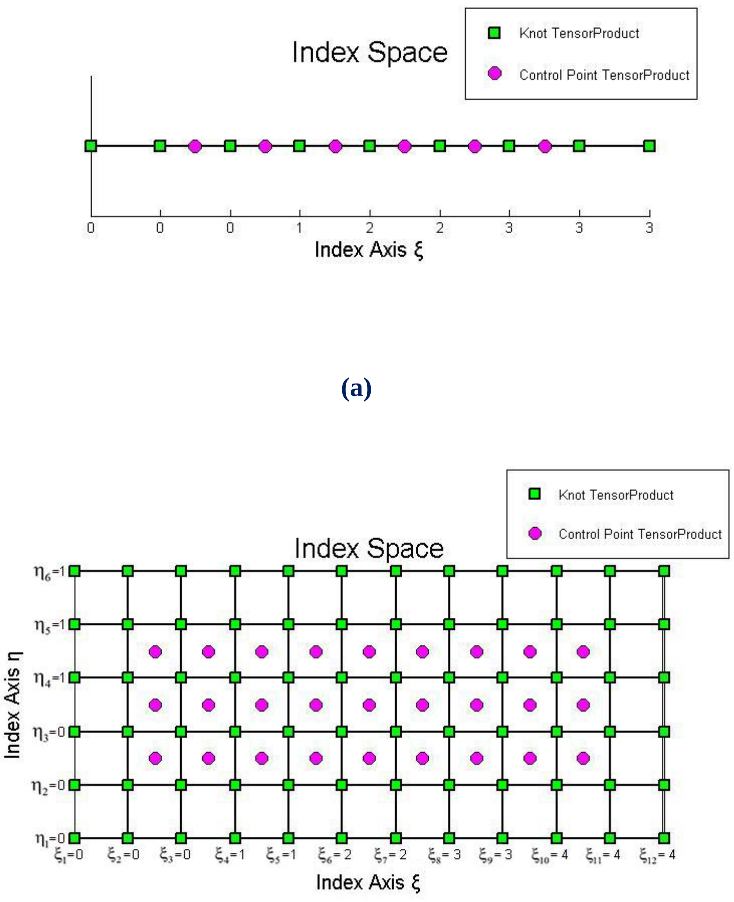
2.2.1 Index Space

Index Space is a representation of the model with respect to Knot Values. It is a line in 1D, containing the corresponding Knot Values in equally spaced positions. This space focuses upon the sequence of Knot Values rather than their actual numerical content.

Index Space describes the contribution of each Knot Value to the creation of a certain B-SPLine basis function. This helps identify the level of interconnection between basis functions and the Knot Value support of each function.

Control Points are also evaluated in the Index Space. In fact, Control Points are defined as the center of the support of Knot Value Spans.

Expansion to 2D or 3D leads to the creation of rectangles or cuboids respectively. Due to tensor product properties, everything mentioned about 1D extends and applies to both 2D and 3D. Thus, Index Space provides information that can contribute to the comprehension of a complex representation.



(a)

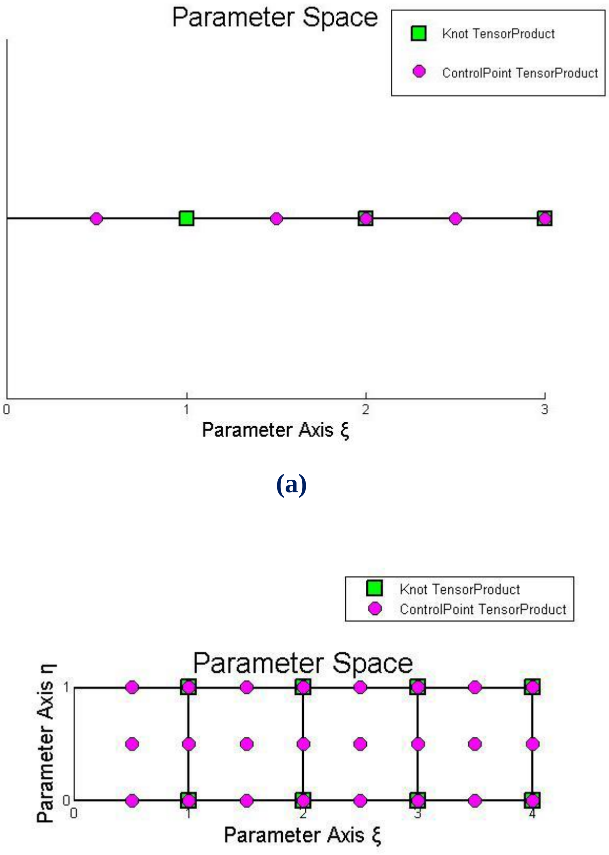
(b)

Figure 2.2. (a) Curve and **(b)** Surface represented in Index Space

2.2.2 Parameter Space

Parameter Space is a representation of the model with respect to Knots. SPLine entities are always represented as orthogonal shapes in Parameter Space. Only lines, rectangles and cuboids exist here. In order to transform those simple patterns to virtually unlimited, complex geometries, the application of a mapping from Parameter to Physical Space is required. Hence, Parameter Space is a primitive, abstract representation of Physical Space. The mapping between Parameter Space and Physical Space is achieved through the Jacobian Matrix and its inverse. This is something widely utilized in FEM as well.

The illustration of basis functions in the Parameter Space allows for a better understanding of concepts such as support, Control Point coordinates and the role of Knots in basis function creation. Each Knot marks the beginning and the end of a basis function domain. By “domain” we mean the area in which the basis function is non-zero, as all basis functions are defined throughout the Parameter Space, but are non-zero only in specific Knot Spans. Basis functions sharing the same domain are overlapping in Parameter Space and controlling a common part of the entity in the Physical Space.



(a)

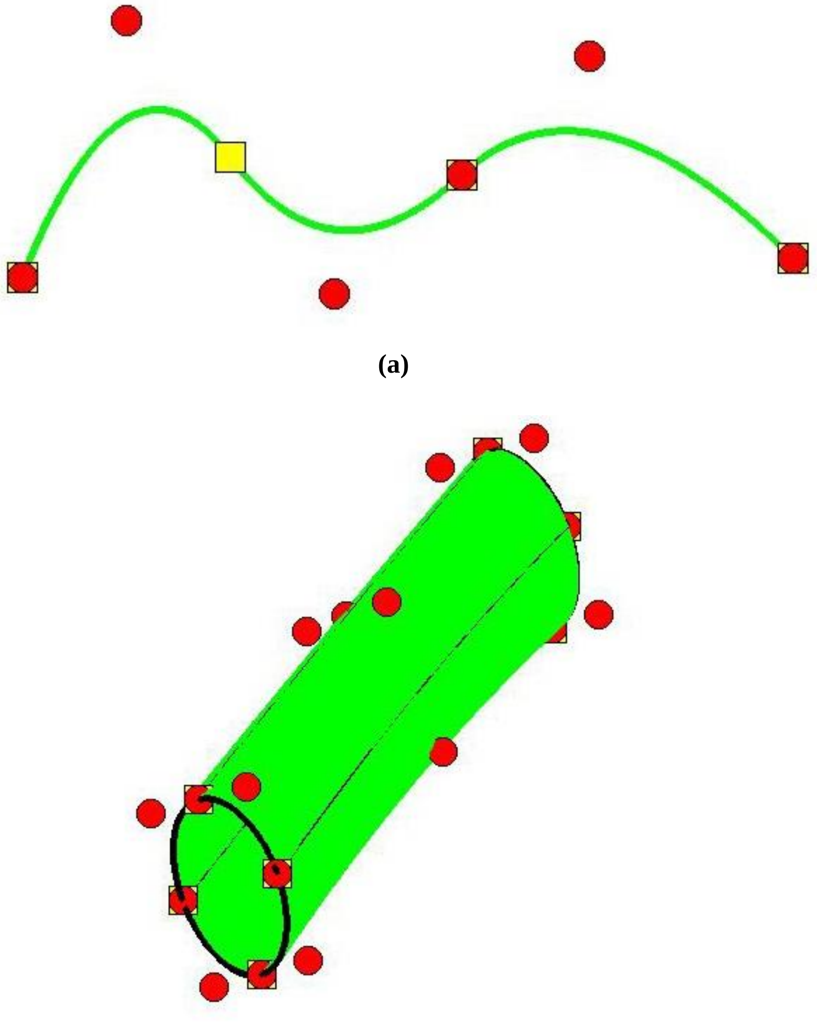
(b)

Figure 2.3. (a) Curve and **(b)** Surface represented in Parameter Space

2.2.3 Physical Space

Physical Space is the already known Cartesian Space, where the real model is represented. Simple orthogonal shapes from Parameter Space are transformed into complex entities in the Physical Space. Physical coordinates of the Control Points play a major role in the aforementioned mapping, but an equally drastic role is set upon basis functions. In fact, for a given set of Control Points, only a single set of basis functions can lead to the same geometry. We will examine this thoroughly later.

Control Points can often be seen outside the model in Physical Space in contrast to FEM's nodes which always belong to the mesh. It is one of the reasons NURBS and SPLine entities in general can accurately represent multiple types of geometries and the understanding of this peculiarity is one of the many challenges of Isogeometric Analysis.



(a)

(b)

Figure 2.4. (a) Curve and **(b)** Surface represented in Physical Space

2.3 B-SPLine Geometries

2.3.1 B-SPLine Basis Functions

Given a sequence of non-decreasing numbers, $\Xi = \left\{ \xi_1 \quad \xi_2 \quad \cdots \quad \xi_{n+p} \quad \xi_{n+p+1} \right\}$, we can evaluate the B-SPLine basis functions at $\xi \in \left[\xi_1, \xi_{n+p+1} \right]$ using the Cox-de Boor recursive formula [1]:

First, for degree $p = 0$ (piecewise constant, Box B-SPLine)

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

The piecewise constant does not include the right edge ξ_{i+1} in order to ensure partition of unity, as the next basis function begins at that edge. The last function, however, includes both left and right edge, in order to be defined for the whole Knot Span.

$$N_{n+p,0}(\xi) = \begin{cases} 1, & \text{if } \xi_{n+p} \le \xi \le \xi_{n+p+1} \\ 0, & \text{otherwise} \end{cases}$$

Afterwards, for degree $p = 1, 2, \ldots$:

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

with the assumption of $\frac{0}{0} \square 0$.

We keep the same symbols with [1], which is quite popular, because we want the reader to browse through this thesis conveniently. In the software implementation of this method, however, we discourage the use of symbolizations and instead represent the variables with full names.

2.3.2 Knot Value Vector as Boundary of Basis Functions Support

Knots define the boundaries of the model's basis functions. They represent "switches" which turn "on" or "off" a certain piece of a B-SPLine basis function. In order to acquire the Knots and the basis functions, a Knot Vector must be defined.

A Knot Vector Ξ is usually defined in bibliography as a set of coordinates ξ_i, with $\xi_i \le \xi_{i+1}$. It can contain the same number multiple times and generates the basis in a unique way.

In order to improve efficiency and communication between members of GiGA Team, we define as

- "Knot Value Vector": the whole set of non-decreasing coordinates ("Knot Values").
- "Knot Vector": the set of unique coordinates ("Knots").

For example, a Knot Value Vector could be $\{0 \quad 1 \quad 1 \quad 1 \quad 2 \quad 2 \quad 3 \quad 3\}$ where $0, 1, 1, 1, 2, 2, 3, 3$ are the separate Knot Values. The corresponding Knot Vector is $\{0 \quad 1 \quad 2 \quad 3\}$ where $0, 1, 2, 3$ are the separate Knots.

Let p be the polynomial degree of the basis function. If the first and the last Knots are repeated $p+1$ times, the Knot Value Vector is considered "Open", because it has C^{-1} Continuity on the edges, creating an open curve that is interpolatory at these points. If Knot Values are equally spaced, the Knot Value Vector is considered "Uniform". In CAD community, non-uniform, open Knot Value Vectors are widely used. An example of such a Knot Value Vector with $p = 2$ is:

$$\{0 \quad 0 \quad 0 \quad 0.5 \quad 1 \quad 1.5 \quad 2 \quad 2.5 \quad 3 \quad 3 \quad 3\}$$

A Knot Value Vector may contain integers or decimals. In fact, the actual numerical content of Knot Values is of no importance. What matters is the relative distance between them. This means a Knot Value Vector can be multiplied by any number, or have a number added to every Knot Value and the resulting basis would still be the same.

In GiGA Team, we generally prefer to use Knot Value Vectors that start from 0 and span by 1, as it is more convenient for the human mind to use an integer system.

2.3.3 Control Points as the Center of the Support

Control Points exist in all three spaces. Their parametric coordinates are defined as the center of the support in the Index Space. Recall that for the i^{th} basis function of order p the support is $\left[\xi_i, \xi_{i+p+1}\right)$. The support contains $p+1$ Knot Value Spans, therefore $p+2$ Knot Values (including the right boundary value ξ_{i+p+1}).

For even degrees, the center of the support in the Index Space lies between two sequential Knot Values, $i+\frac{p}{2}$ and $i+\frac{p}{2}+1$. As a result, the coordinate of the Control Point is defined as the average of these Knot Values:

$$\xi_{CP} = 0.5 \cdot \left(\xi_{i+\frac{p}{2}} + \xi_{i+\frac{p}{2}+1} \right)$$

which means that a Control Point of even degree can either be on a Knot, or in the middle of a Knot Span.

For odd degrees, the center of the support is the Knot Value $i+\frac{p+1}{2}$. Therefore, for odd degrees, Control Points are always coincident with Knots.

2.3.4 Full Tensor Product as a characteristic of B-SPLine Basis Functions

B-SPLine basis functions are of full-tensor product nature. Consequently, it is easy to combine B-SPLines across different directions, in order to evaluate a multi-directional B-SPLine Shape function.

2D B-SPLine Shape functions can be evaluated as tensor product of basis functions $N_{i,p}(\xi)$ and $M_{j,q}(\eta)$:

$$R_{i,j}^{p,q}(\xi,\eta) = N_{i,p}(\xi) \cdot M_{j,q}(\eta)$$

3D B-SPLine Shape functions are a tensor product of basis functions on three directions, $N_{i,p}(\xi)$, $M_{j,q}(\eta)$ and $L_{k,r}(\zeta)$:

$$R_{i,j,k}^{p,q,r}(\xi,\eta,\zeta) = N_{i,p}(\xi) \cdot M_{j,q}(\eta) \cdot L_{k,r}(\zeta)$$

It is understood that all properties of B-SPLine basis functions in one direction are inherited by the multi-directional Shape functions. As a result, thorough adaptation to one-directional B-SPLine basis function properties and techniques is very important for understanding multi-directional complex geometries.

2.3.5 B-SPLine Basis Function Properties

According to “The NURBS Book” [3], B-SPLine basis functions possess the following important properties:

1. Local support:
$$N_{i,p}(\xi) = 0 \ \forall \ \xi \notin \left[\xi_i, \xi_{i+p+1}\right)$$
2. In any given Knot Span, at most $p+1$ functions of order p are non-zero.
3. Nonnegativity:
$$N_{i,p}(\xi) \geq 0 \ \ \forall \ \xi, i, p$$
4. Partition of unity:
$$\sum_{i=1}^{n} N_{i,p}(\xi) = 1 \ \forall \ \xi, p$$
5. C^{p-m} Continuity across Knots with multiplicity m.
6. $N_{i,p}(\xi)$ has exactly one maximum value, except for p=0.
7. A non-periodic Knot Value Vector that produces n functions of order p has $n+p+1$ Knot Values.
8. Every B-SPLine basis function shares support with $2p$ B-SPLines.

2.3.5.1 Local support

Local support means that basis functions are non-zero only in certain Knot Spans in Parameter Space. This can be expressed by

$$N_{i,p}(\xi) = 0 \ \ \forall \xi \notin \left[\xi_i, \xi_{i+p+1}\right)$$

Local support is a result of the recursive character of B-SPLines. For the creation of a B-SPLine function of degree p, two consecutive B-SPLine functions of order $p-1$ are used. For the creation of those consecutive basis functions, three consecutive functions of order $p-2$ are needed. Inductively, $p+1$ consecutive box basis functions are required.

Each box function has a support of one Knot Value Span. As a result, the support of the final basis function is defined by the union of the supports of the box functions, hence $p+1$ consecutive Knot Value Spans.

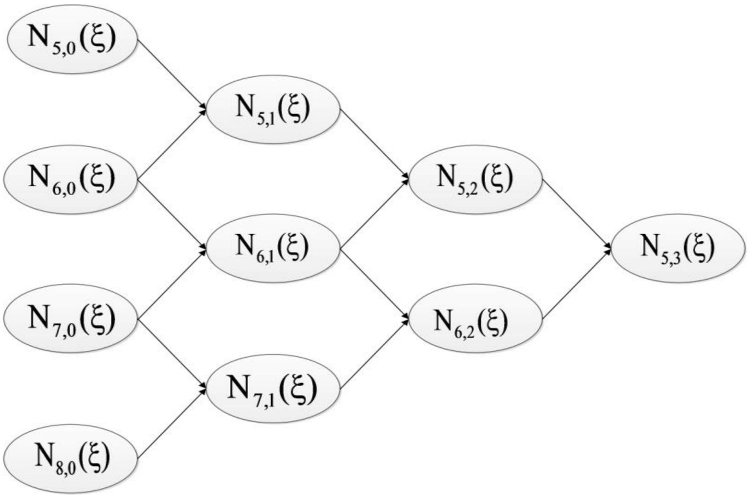


Figure 2.5. Lower-order basis functions required for the creation of $N_{5,3}(\xi)$.

In Figure 2.5, the recursive character of B-SPLines is represented. The box functions, drawn in red, are required in order to build the linear basis functions, drawn in green. Linear functions are combined for the evaluation of quadratic $(p=2)$ basis functions. Bear in mind that some linear and box functions are zero across the entire domain, but still contribute to the evaluation of the next-order B-SPLines.

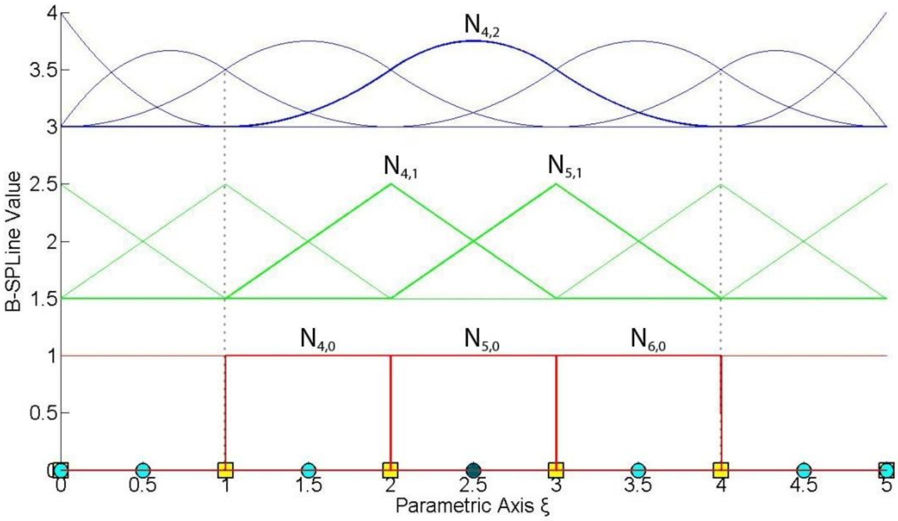


Figure 2.6. B-SPLine recursive character.
Basis functions required for the evaluation of a quadratic B-SPLine function.

For example, $N_{4,2}(\xi)$ in Figure 2.6 is created from $N_{4,1}(\xi)$ and $N_{5,1}(\xi)$. They, in turn, are evaluated from $N_{4,0}(\xi)$, $N_{5,0}(\xi)$ and $N_{5,0}(\xi)$, $N_{6,0}(\xi)$ respectively. The corresponding box functions are non-zero in the Knot Spans $[1,2)$, $[2,3)$ and $[2,3)$, $[3,4)$, thus creating the support $[1,4)$ of $N_{4,2}(\xi)$.

In a similar fashion, a support can be defined with respect to Knot Values contributing to the creation of a B-SPLine basis function. These are the $p+2$ Knot Values that are contained in the Knot Value Span support of the function. Both Knot Value Span Support and Knot Value Support are necessary for the understanding of Isogeometric Methods.

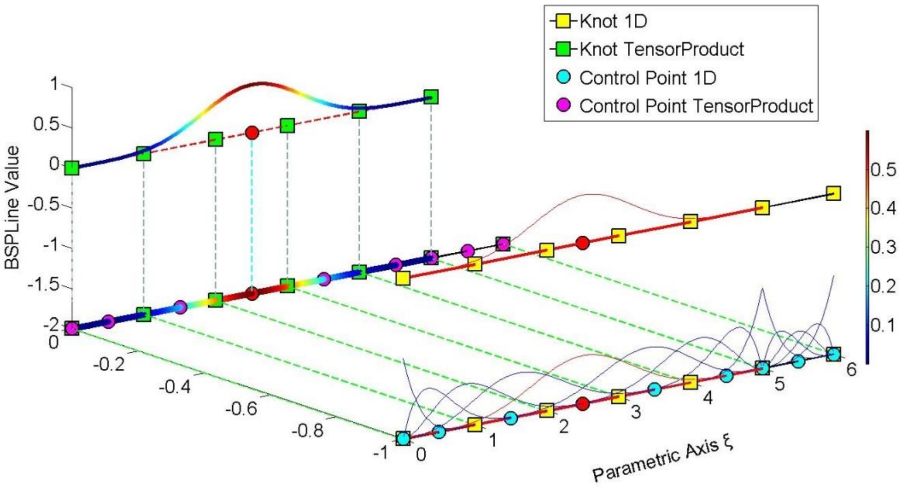


Figure 2.7. B-SPLine Basis functions for Knot Value Vector .
$\Xi = \{0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 5 \quad 5 \quad 5 \quad 6 \quad 6 \quad 6 \quad 6 \quad 6\}$
1D Shape Function $N_{5,4}(\xi)$

In Figure 2.7, the degree is $p=4$, so each basis function has a support of $p+1=4+1=5$ Knot Value Spans. The selected function $N_{5,4}(\xi)$, drawn in red, is non-zero only in the Knot Value Spans $[0,1)$, $[1,2)$, $[2,3)$, $[3,4)$ and $[4,5)$. The $p+2=6$ Knot Values that define this function are $\{0,1,2,3,4,5\}$ shown in green. There are no trivial spans in this B-SPLine function, so it is considered a fully developed basis function.

Bear in mind that Knot Values can be repeated, thus forming trivial spans.

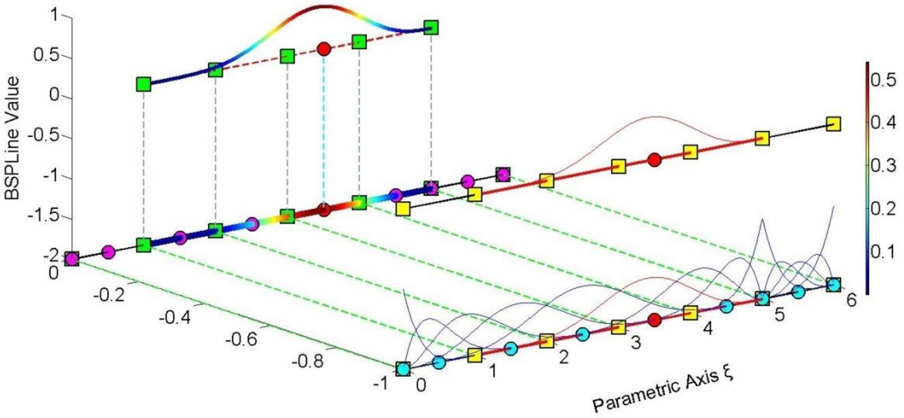


Figure 2.8. B-SPLine Basis functions for Knot Value Vector .
$\Xi = \{0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 5 \quad 5 \quad 5 \quad 6 \quad 6 \quad 6 \quad 6 \quad 6\}$
1D Shape function $N_{6,4}(\xi)$

In Figure 2,8 $N_{6,4}(\xi)$ is highlighted. This function is non-zero in the Knot Value Spans $[1,2)$, $[2,3)$, $[3,4)$, $[4,5)$ and $[5,5)$. The corresponding Knot Values shown in green are $\{1,2,3,4,5,5\}$. The support is five Knot Value Spans, but this leads to only four Knot Spans, as a trivial span is contained.

Observe the next basis functions in this example; as more trivial spans are contained, the Knot Span support of each function is reduced. Still, the Knot Value Span support and the corresponding Knot Value support follow the rules already mentioned.

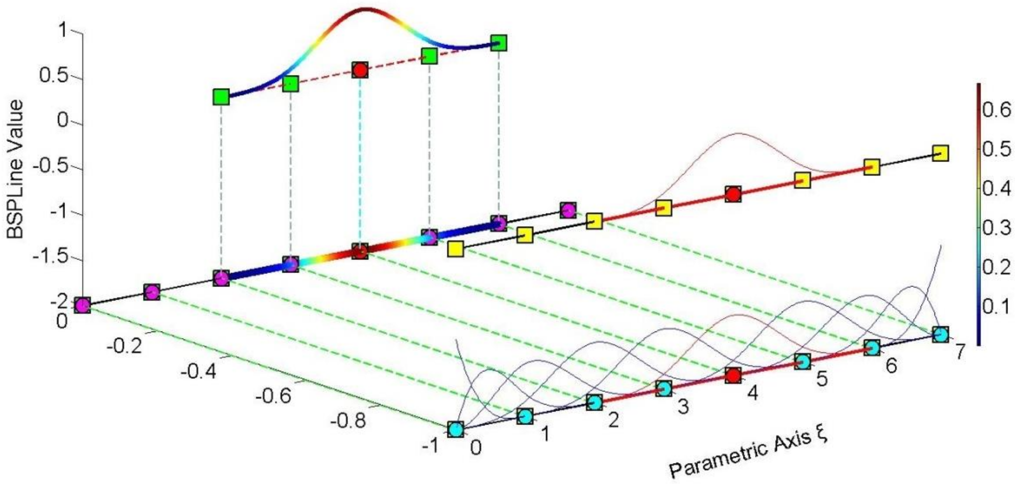


Figure 2.9. B-SPLine Basis functions for Knot Value Vector.
$\Xi = \{0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 7 \quad 7 \quad 7\}$
1D Shape Function $N_{6,3}(\xi)$

In the above figure, the polynomial degree of the basis functions is $p = 3$. Knot Value Span support is $p+1=4$ Knot Value Spans, which requires Knot Value support of $p+2=5$ Knot Values.

$N_{6,3}(\xi)$ is non-zero in the Knot Value Spans $[2,3)$, $[3,4).[4,5)$ and $[5,6)$. No trivial spans are involved, so this is another fully developed B-SPLine. The Knot Value Support consists of the Knot Values $\{2,3,4,5,6\}$.

B-SPLine tensor product properties enable the immediate expansion of 1D properties to 2D and 3D B-SPLine Shape functions.

$N_{3,2}(\xi)$ in Figure 2.10 has a support of 3 Knot Value Spans per ξ, $[0,1)$, $[1,2)$ and $[2,3)$. It is created by Knot Values $\{0,1,2,3\}$.

$M_{4,2}(\eta)$ has a support of 3 Knot Value Spans per η, $[1,2)$, $[2,3)$ and $[3,4)$. It is created by Knot Values $\{1,2,3,4\}$.

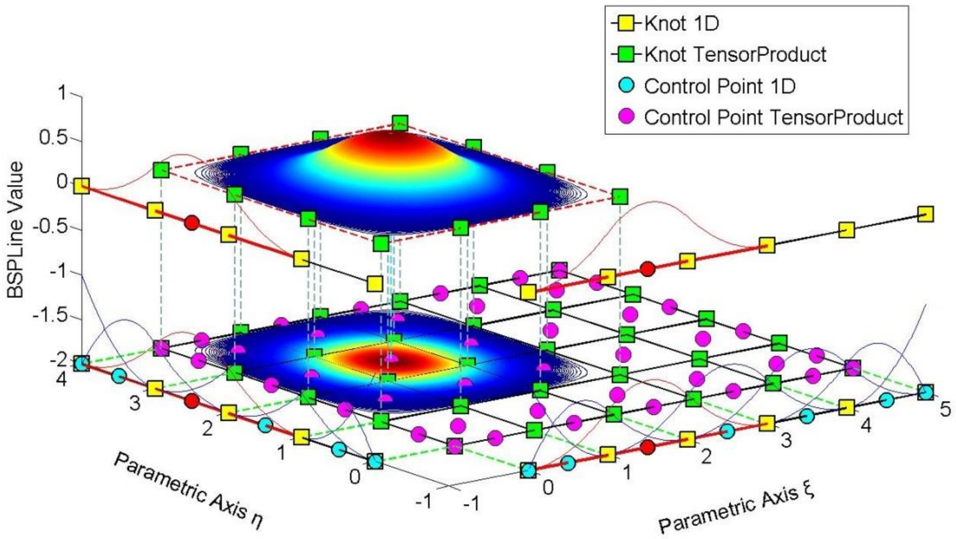


Figure 2.10. Shape Function $R_{i,j}^{p,q} = R_{3,4}^{2,2}(\xi,\eta)$ as a tensor product of $N_{3,2}(\xi)$ and $M_{4,2}(\eta)$

Knot Value Vector $\Xi = \{0 \quad 0 \quad 0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 5 \quad 5\}$

Knot Value Vector $H = \{0 \quad 0 \quad 0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 4 \quad 4\}$

The 2D Shape function $R_{3,4}^{2,2}(\xi,\eta)$ is the full tensor product of $N_{3,2}(\xi)$ and $M_{4,2}(\eta)$. The support of the Shape function is a rectangle created from the supports of the respective basis functions. It has a total area of $3 \cdot 3 = 9$ Knot Value Rectangles.

Observe that the value of the bidirectional B-SPLine is represented both in the third axis of the graph and by projection of the contour in the 2D plane. This is useful for representation of 3D basis functions.

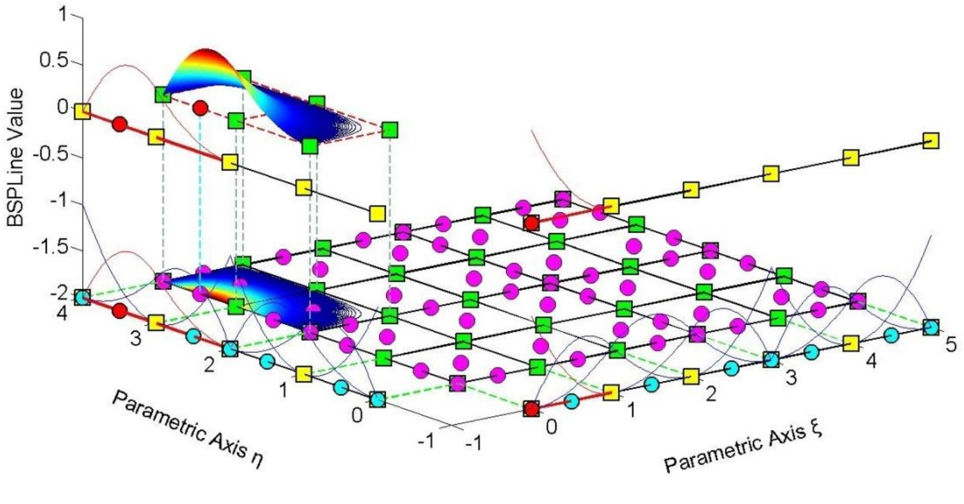


Figure 2.11. Shape Function $R_{i,j}^{p,q} = R_{1,6}^{2,2}(\xi,\eta)$ as a tensor product of $N_{1,2}(\xi)$ and $M_{6,2}(\eta)$

Knot Value Vector: $\Xi = \{0 \quad 0 \quad 0 \quad 1 \quad 2 \quad 3 \quad 3 \quad 4 \quad 5 \quad 5 \quad 5\}$

Knot Value Vector $H = \{0 \quad 0 \quad 0 \quad 1 \quad 2 \quad 2 \quad 3 \quad 4 \quad 4 \quad 4\}$

Both 1D basis functions have reduced Continuity in Figure 2.11.

$N_{1,2}(\xi)$ has a Knot Value Span Support of $[0,0)$, $[0,0)$ and $[0,1)$, leading to a Support of a single Knot Span, $[0,1)$. The corresponding Knot Value Support is $\{0,0,0,1\}$.

$M_{6,2}(\eta)$ has a Knot Value Support of $\{2,3,4,4\}$ which leads to a Knot Value Span Support of $[2,3)$, $[3,4)$, $[4,4)$. One trivial and two non-trivial Spans are contained.

Support for the Shape Function $R_{1,6}^{2,2}(\xi,\eta)$ is defined as the tensor product of the supports of the basis functions, namely $1\cdot 2=2$ Knot Rectangles.

3D B-SPLine Shape functions can be represented as in the Figure 2.12.

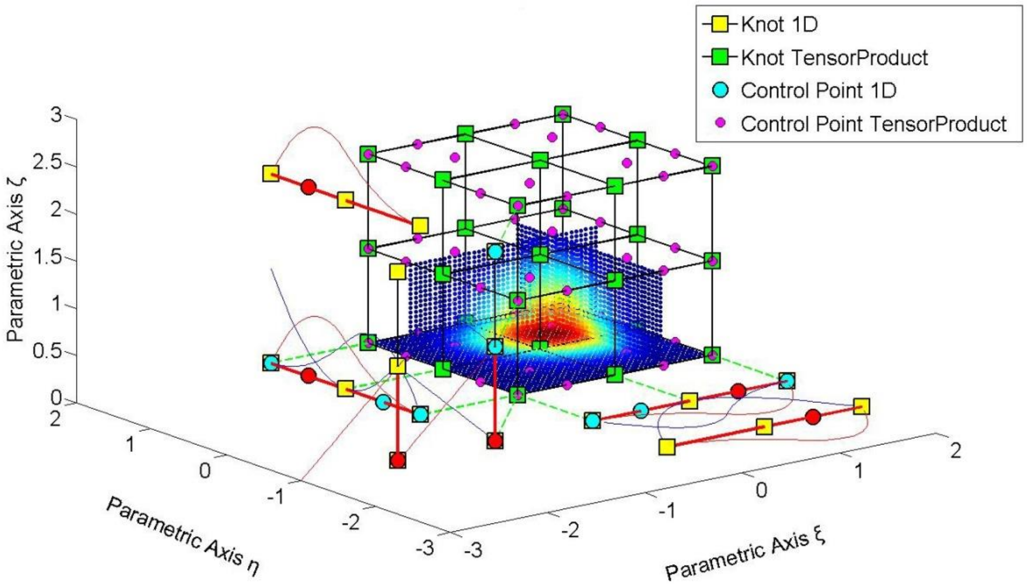


Figure 2.12. Shape Function $R_{i,j,k}^{p,q,r}=R_{3,3,1}^{2,2,1}(\xi,\eta,\zeta)$ as tensor product of $N_{3,2}(\xi)$, $M_{3,2}(\eta)$, $L_{1,1}(\zeta)$.
Knot Value Vector $\Xi=\{0\quad 0\quad 0\quad 1\quad 2\quad 2\quad 2\}$
Knot Value Vector $H=\{0\quad 0\quad 0\quad 1\quad 2\quad 2\quad 2\}$
Knot Value Vector $Z=\{0\quad 0\quad 1\quad 2\quad 2\}$

The three axes represent the three parametric directions ξ,η,ζ. B-SPLine basis functions for ξ are drawn in the $\xi-\eta$ plane, functions for η in the $\eta-\zeta$ plane and functions for ζ in the $\zeta-\xi$ plane.

$N_{3,2}(\xi)$ and $M_{3,2}(\eta)$ both have a support of 3 Knot Value Spans, $[0,1)$, $[1,2)$ and $[2,2)$ in their respective directions. Knot Value Support for those functions is $\{0,1,2,2\}$. Knot Span Support is restrained at 2 Knot Spans for each function.

$L_{1,1}(\zeta)$ has a Knot Value Span Support of $\left[0,0\right)$ and $\left[0,1\right)$, leading to the support of 1 Knot Span. Knot Value Support is defined as $\left\{0,0,1\right\}$.

Their tensor product value has been calculated as a function of two parametric directions at the Control Point coordinate of the remaining direction. The resulting contour is projected on a plane that is parallel to the two directions and intersects with the tensor product Control Point. This process is repeated for all three possible combinations, thus creating contours at $\xi-\eta$, $\eta-\zeta$ and $\zeta-\xi$ planes.

For example, Control Point $\left(3,3,1\right)$ has coordinates $\left(\xi,\eta,\zeta\right)=\left(1.5,1.5,0\right)$.

$R_{3,3,1}^{2,2,1}\left(\xi,\eta,0\right)$ has been calculated and projected in $\zeta=0$ plane. The support of the projection shows the support of the 3D Shape function per ξ,η, namely $2\cdot2=4$ Knot Rectangles.

$R_{3,3,1}^{2,2,1}\left(\xi,1.5,\zeta\right)$ is represented in the plane $\eta=1.5$. Support of the 3D Shape function per ξ, ζ is $2\cdot1=2$ Knot Rectangles.

$R_{3,3,1}^{2,2,1}\left(1.5,\eta,\zeta\right)$ is projected in $\xi=1.5$ plane. The support of the 3D Shape function per η, ζ is $2\cdot1=2$ Knot Rectangles.

The support of the 3D Shape function across the entire domain is the tensor product of the 1D supports. In this particular case, it is $2\cdot2\cdot1=4$ Knot Cuboids. The support is represented by the tensor product of the projections in Figure 2.12.

2.3.5.2 Maximum number of non-zero functions per Knot Span

A box function that is non-zero in one Knot Span contributes to the evaluation of two consecutive B-SPLine basis functions of order $p=1$. These two functions lead to the creation of three consecutive basis functions of order $p=2$. Inductively, a box function contributes to the creation of $p+1$ B-SPLine basis functions of order p.

It applies from Cox de Boor recursive formula that only one box function is non-zero across a selected Knot Span. As a result, only the corresponding $p+1$ B-SPLine basis functions of order p can be non-zero in that specific Knot Span.

Therefore, at a non-trivial Knot Value Span $\left[\xi_i,\xi_{i+1}\right)$ only the basis functions $N_{i-p,p}(\xi)$, $N_{i-p+1,p}(\xi)$, ..., $N_{i,p}(\xi)$ are non-zero. This is used efficiently in Stiffness Matrix Formulation, in order to reduce computational cost.

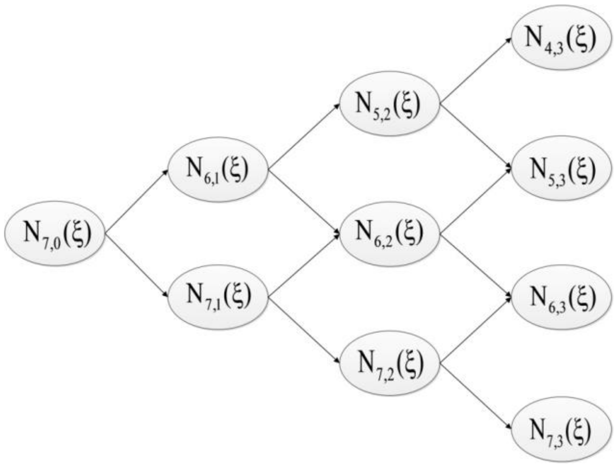


Figure 2.13. Contribution of one box function to the creation of higher-order B-SPLine basis functions.

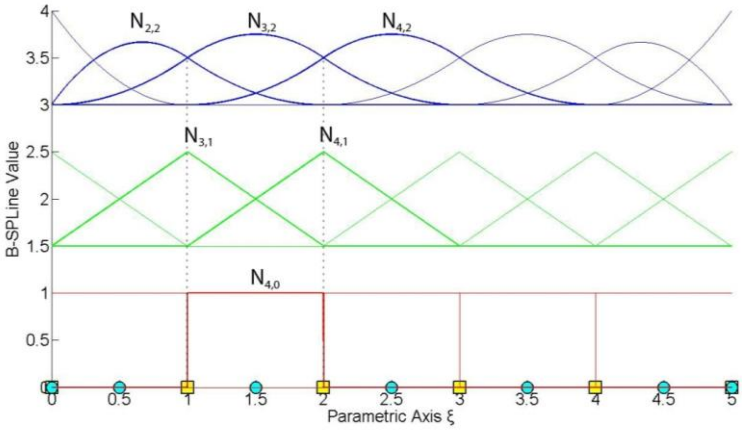


Figure 2.14. Contribution of one box function to non-zero higher-order B-SPLine basis functions across Knot Span $[1,2)$.

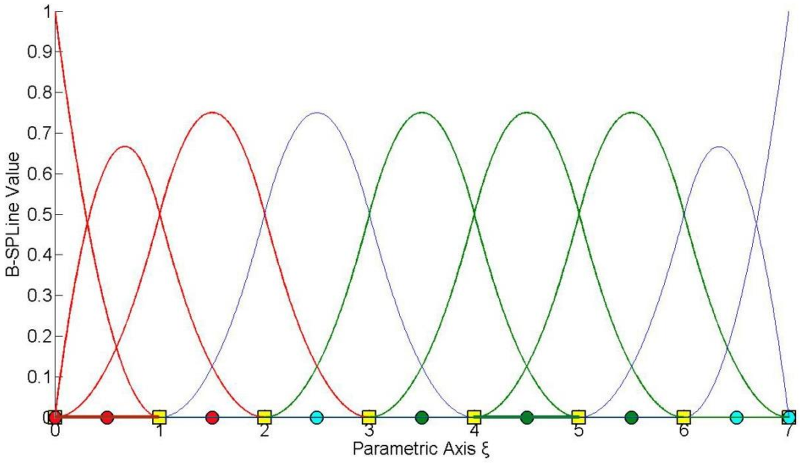


Figure 2.15. B-SPLine Basis functions for Knot Value Vector $\Xi = \{0 \quad 0 \quad 0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 7 \quad 7\}$.

Basis functions that are non-zero in Knot Span $[0,1)$ are drawn in red.

Basis functions that are non-zero in Knot Span $[4,5)$ are drawn in green.

In Figure 2.15, two separate Knot Spans are examined. For every Knot Span, $p+1=3$ basis functions are non-zero.

For the first Knot Span, $[0,1)=[\xi_3,\xi_4)$, basis functions $N_{1,p}(\xi)$, $N_{2,p}(\xi)$, $N_{3,p}(\xi)$ are non-zero.

For the second Knot Span, $[4,5)=[\xi_7,\xi_8)$, basis functions $N_{5,p}(\xi)$, $N_{6,p}(\xi)$, $N_{7,p}(\xi)$ are non-zero.

As a result, in 2D, at a given Knot Rectangle, only the tensor products of the respective non-zero basis functions are non-zero, namely $(p+1)\cdot(q+1)$ Shape functions.

Inductively, in 3D, at a given Knot Cuboid, the tensor products of the corresponding non-zero basis functions create $(p+1)\cdot(q+1)\cdot(r+1)$ non-zero Shape functions.

2.3.5.3 Non-negativity

It has been established that:

$$N_{i,p}(\xi)\geq 0 \ \forall \ \xi, i, p .$$

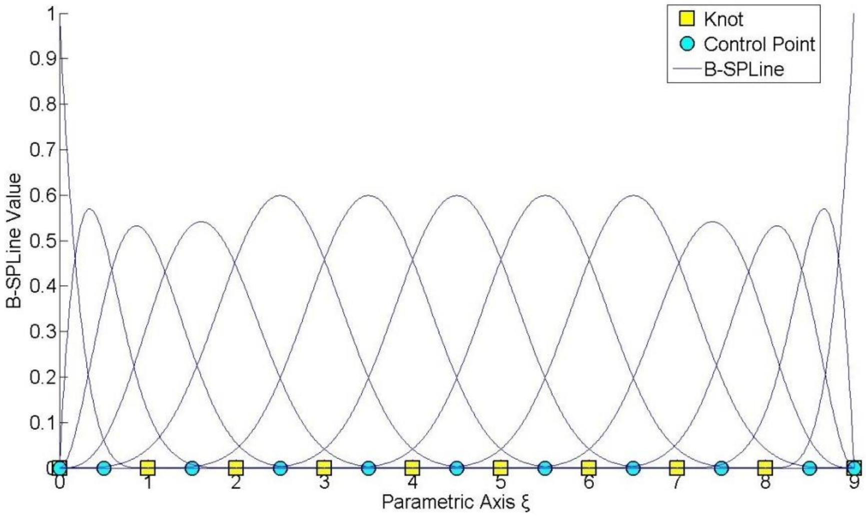


Figure 2.16. B-SPLine Basis functions for Knot Value Vector $\Xi=\{0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \quad 9 \quad 9 \quad 9 \quad 9 \quad 9\}$

The degree in Figure 2.16 is $p=4$. As we see in the picture above, all the basis functions are positive for every ξ, i. This property is very important for Isogeometric Analysis and it does not apply in the classical Shape functions of Finite Element Analysis. Naturally, it is easier for the human mind to work with only positive values, so this attribute simplifies and encourages the understanding of B-SPLines. Non-negativity applies for 2D and 3D Shape functions as well.

2.3.5.4 Partition of unity

Partition of unity is expressed by

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

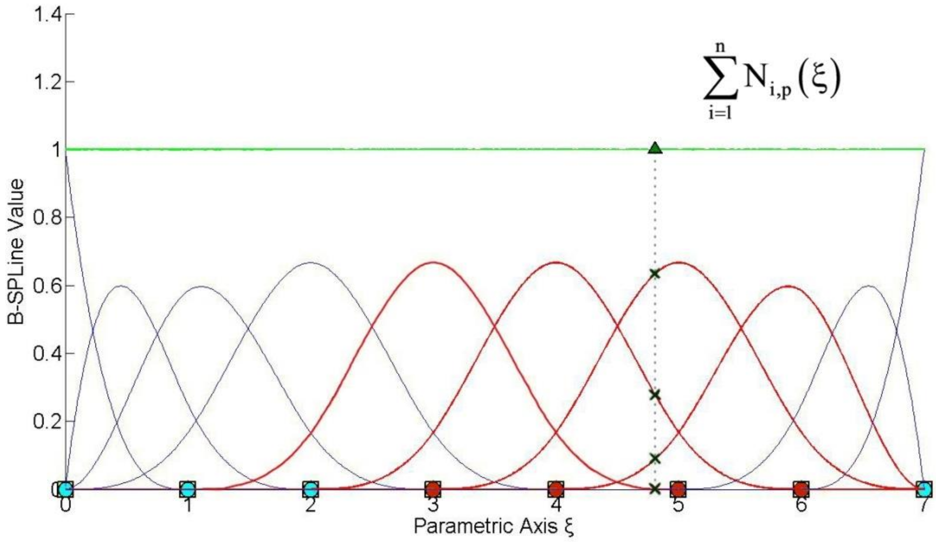


Figure 2.17. B-SPLine Basis functions for Knot Value Vector
$\Xi = \{0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 7 \quad 7 \quad 7\}$
Sum of B-SPLine function value at every ξ . Partition of Unity.

The degree for the B-SPLine basis functions in Figure 2.17 is $p = 3$. The green horizontal line represents the sum of B-SPLine values at the corresponding ξ, which, of course, is equal to 1 for every ξ.

For example, B-SPLine values have been evaluated for $\xi = 4.81$, with the following results for the four non-zero basis functions:

$$N_{5,3}(4.81) = 0.0011$$
$$N_{6,3}(4.81) = 0.2763$$
$$N_{7,3}(4.81) = 0.6340$$
$$N_{8,3}(4.81) = 0.0886$$

The above yields

$$\sum_{i=1}^{10} N_{i,3}(4.81) = 1$$

as expected.

Partition of unity applies for multi-dimensional Shape functions as well.

In 2D, partition of unity is expressed as

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

which is a result of tensor product nature [1]:

$$\sum_{i=1}^{n}\sum_{j=1}^{m} R_{i,j}^{p,q} = \sum_{i=1}^{n}\sum_{j=1}^{m} N_{i,p}(\xi)\cdot M_{j,q}(\eta) = \left(\sum_{i=1}^{n} N_{i,p}(\xi)\right)\cdot\left(\sum_{j=1}^{m} M_{j,q}(\eta)\right) = 1$$

In full analogy, 3D Shape functions also possess partition of unity:

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

Partition of unity is also a property of great importance for the Shape functions that are used in Finite Element Analysis.

2.3.5.5 C^{p-m} Continuity

At a Knot with multiplicity m, $N_{i,p}(\xi)$ produces $p-m$ continuous derivatives or, in other words, it has C^{p-m} Continuity. Continuity less than C^0 is not acceptable for internal Knots, which means they can be repeated up to p times. Bear in mind that as Continuity decreases, B-SPLine basis functions tend to be more “steep”.

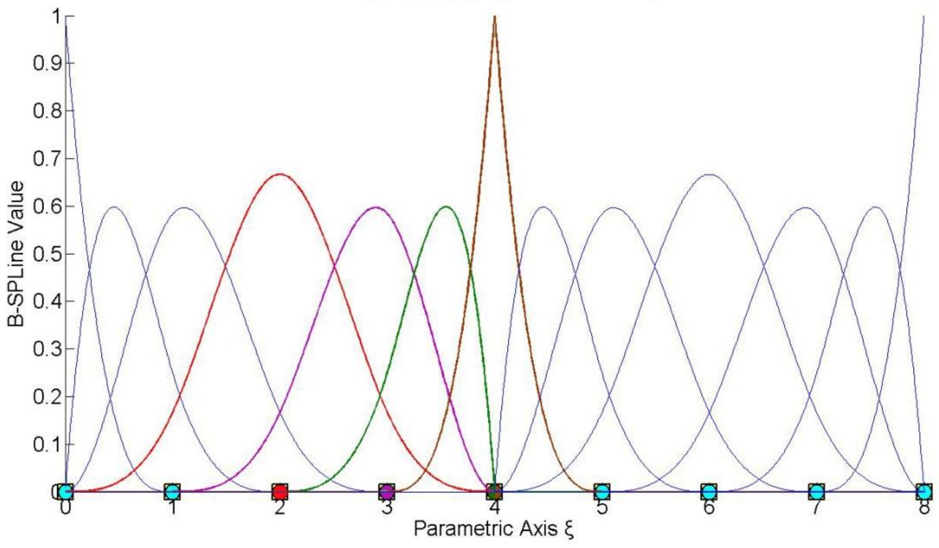


Figure 2.18 B-SPLine Basis functions for Knot Value Vector $\Xi = \{0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 4 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \quad 8 \quad 8 \quad 8\}$

The polynomial order in Figure 2.18 is $p=3$. Continuity for a basis function is affected only by the corresponding Knot Values. Knot $\xi=4$ has multiplicity $m=3$ with respect to the Knot Vector. Its multiplicity with respect to the basis functions depends on each function’s Knot Value Support.

$N_{4,3}(\xi)$, a fully developed basis function, drawn in red, has a Knot Value Support of $\{0,1,2,3,4\}$. The Knot $\xi=4$ has multiplicity $m=1$ with respect to the basis function $N_{4,3}(\xi)$, so this function is $C^{p-m}=C^{3-1}=C^{2}$ Continuous in the entire domain.

The purple basis function, $N_{5,3}(\xi)$ has a Knot Value Support of $\{1,2,3,4,4\}$. The Knot in question has multiplicity $m=2$ with respect to the basis function, which makes $N_{5,3}(\xi)$ $C^{p-m}=C^{3-2}=C^{1}$ Continuous across $\xi=4$.

Knot Value Support for $N_{6,3}(\xi)$ and $N_{7,3}(\xi)$ is $\{2,3,4,4,4\}$ and $\{3,4,4,4,5\}$ respectively. The multiplicity of the Knot is $m=3$ for both basis functions. Therefore, these functions have $C^{p-m}=C^{3-3}=C^{0}$ Continuity across $\xi=4$.

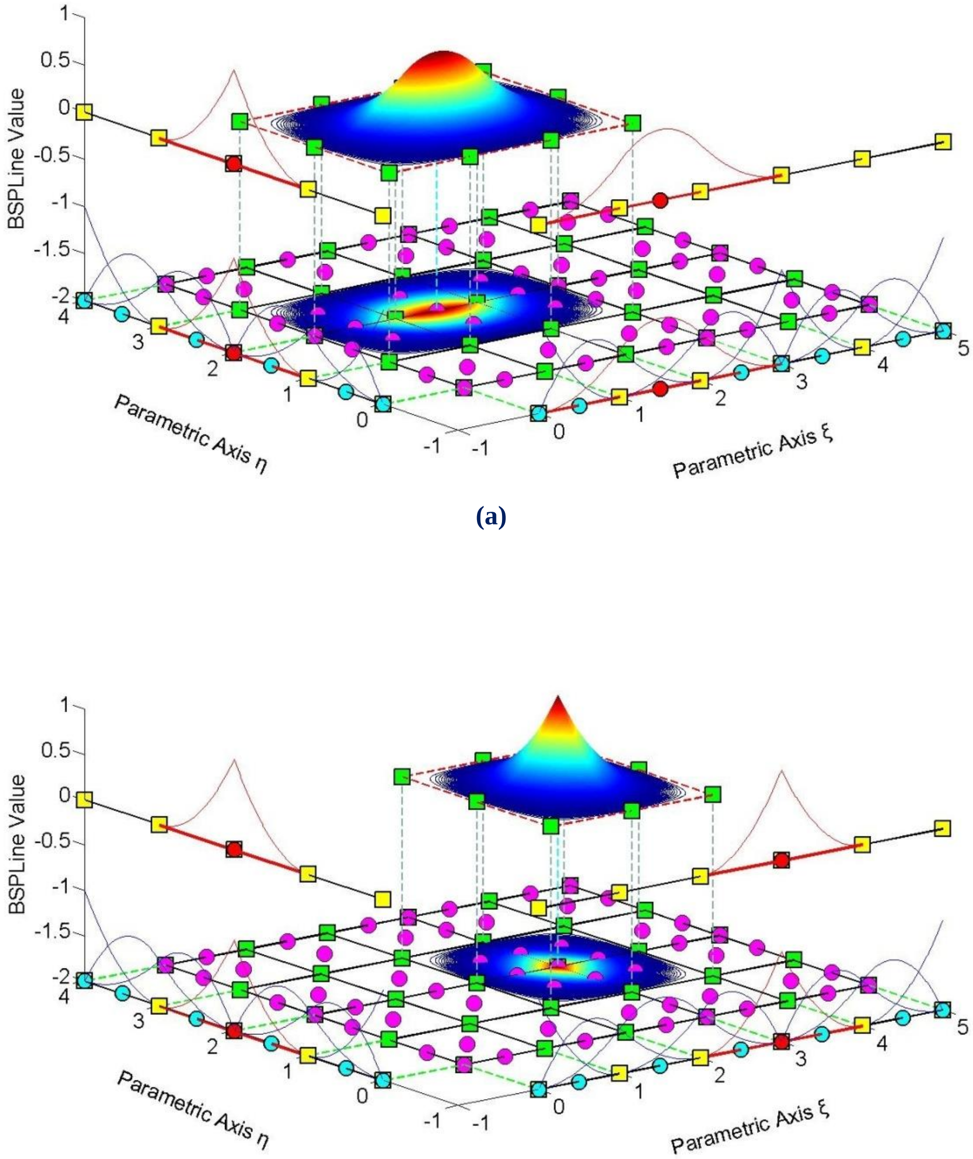
Observe that when a B-SPLine basis function $N_{j,p}(\xi)$ has a Knot ξ_m with multiplicity p in the center of the Knot Value Support, it applies that $N_{j,p}(\xi_m)=1$.

In 2D and 3D cases, Continuity per direction is obtained straight from the Continuity of the corresponding one-directional B-SPLine basis functions.

In Figure 2.19.a, $R_{3,4}^{2,2}(\xi,\eta)$ is tensor product of $N_{3,2}(\xi)$ and $M_{4,2}(\eta)$. $N_{3,2}(\xi)$ has $C^{p-m}=C^{2-1}=C^{1}$ Continuity across $\xi=3$ and $M_{4,2}(\eta)$ has $C^{p-m}=C^{2-2}=C^{0}$ Continuity across $\eta=2$. Therefore, $R_{3,4}^{2,2}(\xi,\eta)$ has C^{1} Continuity with respect to ξ and C^{0} Continuity with respect to η across $(\xi,\eta)=(3,2)$.

In Figure 2.19.b, $R_{5,4}^{2,2}(\xi,\eta)$ is tensor product of $N_{5,2}(\xi)$ and $M_{4,2}(\eta)$. Both B-SPLine basis functions have C^{0} Continuity across $\xi=3$, $\eta=2$ respectively. As a result, $R_{5,4}^{2,2}(\xi,\eta)$ has C^{0} Continuity across $(\xi,\eta)=(3,2)$. This particular case of C^{0} Continuity for both directions will be utilized in future applications.

Note that C^{0} Continuity per both directions leads to $R_{5,4}^{2,2}(3,2)=N_{5.2}(3)\cdot M_{4.2}(2)=1$. Non-negativity and Partition of unity properties require that $R_{i,j}^{2,2}(3,2)=0$, $\forall(i,j)\neq(5,4)$.



(a)

(b)

Figure 2.19. B-SPLine Basis functions for directions ξ, η

Knot Vector $\Xi = \{0 \quad 0 \quad 0 \quad 1 \quad 2 \quad 3 \quad 3 \quad 4 \quad 5 \quad 5 \quad 5\}$

Knot Vector $H = \{0 \quad 0 \quad 0 \quad 1 \quad 2 \quad 2 \quad 3 \quad 4 \quad 4 \quad 4\}$

(a): Shape Function $R_{i,j}^{p,q} = R_{3,4}^{2,2}(\xi, \eta)$ as tensor product of $N_{3,2}(\xi)$ and $M_{4,2}(\eta)$

(b): Shape Function $R_{i,j}^{p,q} = R_{5,4}^{2,2}(\xi, \eta)$ as tensor product of $N_{5,2}(\xi)$ and $M_{4,2}(\eta)$

Figure 2.20 represents $R_{4,3,3}^{2,2,2}(\xi,\eta,\zeta)$, a Shape function formed as a tensor product of three C^0 Continuous functions across $(2,1,1)$. The corresponding basis functions are $N_{4,2}(\xi)$, $M_{3,2}(\eta)$ and $L_{3,2}(\zeta)$. Naturally, at the point of C^0 Continuity for all three directions, it applies that $R_{4,3,3}^{2,2,2}(2,1,1)=N_{4,2}(2)\cdot M_{3,2}(1)\cdot L_{3,2}(1)=1$.

Observe that all functions of reduced Continuity tend to be more "steep". This happens because they contain multiple Knot Values of the same Knot, and therefore develop across trivial spans. This "steepness" is the first indication that a basis function has reduced Continuity.

Moreover, internal C^0 Continuity produced exactly the same B-SPLine basis functions as C^{-1} Continuity on the edge of the Knot Vector. This means that at every Knot with multiplicity $m=p$, only one function remains non-zero. Due to partition of unity, that function's value at that Knot is equal to 1.

In multi-dimensional functions, C^0 Continuity across a Knot requires the basis functions per all directions to be C^0 Continuous at that point. In this case, the value of the multi-dimensional Shape function across this Knot is equal to 1.

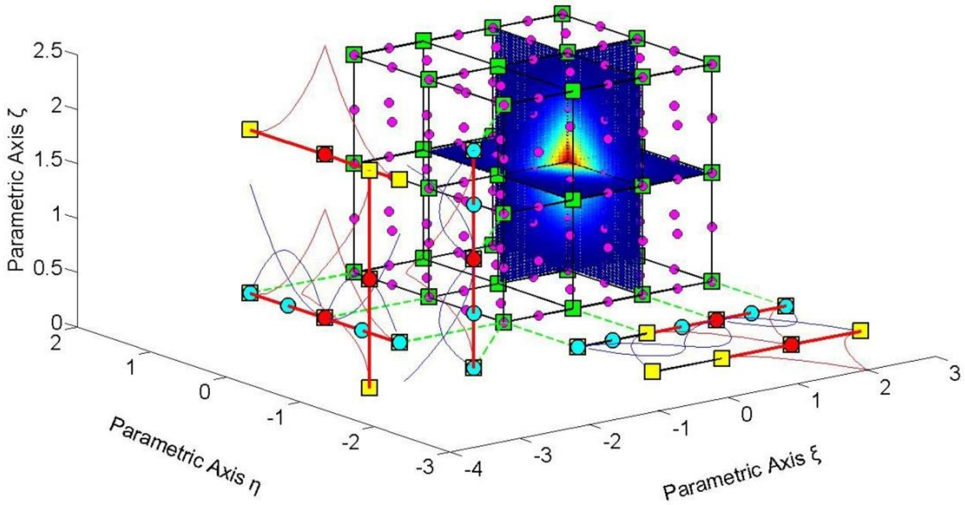


Figure 2.20. Shape Function $R_{i,j,k}^{p,q,r}(\xi,\eta,\zeta)=R_{4,3,3}^{2,2,2}(\xi,\eta,\zeta)$

Knot Value Vector $\Xi=\{0 \quad 0 \quad 0 \quad 1 \quad 2 \quad 2 \quad 3 \quad 3 \quad 3\}$

Knot Value Vector $H=\{0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 2 \quad 2 \quad 2\}$

Knot Value Vector $Z=\{0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 2 \quad 2 \quad 2\}$

2.3.5.6 Linear Independence

Given a finite number of distinct vectors $\{u_1 \quad u_2 \quad ... \quad u_n\}$ and scalars $\{a_1 \quad a_2 \quad ... \quad a_n\}$, the subset S of a vector space V is called linearly independent if the equation $a_1u_1 + a_2u_2 + ... + a_nu_n = 0$ leads to the unique solution $a_1 = a_2 = ... = a_n = 0$. Thus the subset is linearly independent if a linear combination of the vectors is the zero vector, only for $a_1 = a_2 = ... = a_n = 0$. The linear independence in Isogeometric Analysis applies for the basis functions $\{N_{1,p}(\xi) \quad N_{2,p}(\xi) \quad ... \quad N_{n,p}(\xi)\}$. The equation

$$a_1 \cdot N_{1,p}(\xi) + a_2 \cdot N_{2,p}(\xi) + ... + a_n \cdot N_{n,p}(\xi) = 0$$

leads to $a_1 = a_2 = ... = a_n = 0$. We can reach the conclusion that no B-SPLine basis function can be expressed as a linear combination of the other B-SPLine basis functions.

These linearly independent vectors form a basis for the vector space V. Some interesting attributes of such vectors include:

- The basis of the vector space V can be formed by different sets of linearly independent vectors. Any set can be used, provided that the vectors are linearly independent and all the properties above apply for every vector. Thus, in Isogeometric Analysis, we can choose different sets of B-SPLine basis functions in order to represent the vector space.

- The number of vectors of any basis chosen is equal to the dimension of V, often represented as $\dim(V)$. In Isogeometric Analysis $\dim(V)$ is equal to the number of Control Points and, consequently, the number of basis functions used.

- Let there be a mass of vectors $\{u_1 \quad u_2 \quad ... \quad u_n\}$ which form the basis of a vector space with $\dim(V) = n$ and the numbers $\{a_1 \quad a_2 \quad ... \quad a_n\}$, called the coordinates of u. In Isogeometric Analysis, the basis of the vector space is the set of B-SPLine basis functions $\{N_{1,p}(\xi) \quad N_{2,p}(\xi) \quad ... \quad N_{n,p}(\xi)\}$ and the numbers are the coordinates $\{X_1 \quad X_2 \quad ... \quad X_n\}$ of the Control Points of the curve. A random vector u of the vector space V can be represented by the equation:

$$u = a_1u_1 + a_2u_2 + ... + a_nu_n$$

which in Isogeometric Analysis applies as

$$C(\xi) = X_1 \cdot N_{1,p}(\xi) + X_2 \cdot N_{2,p}(\xi) + ... + X_n \cdot N_{n,p}(\xi)$$

Any vector in V can be described as a linear combination of the basis. The numbers $\{a_1 \quad a_2 \quad ... \quad a_n\}$ are the coordinates of u and they are unique for this specific u and this specific basis.

Vector space V in $\mathbb{R}$ is a mass of vectors with the properties below:

1. Commutativity of addition:

$$u+v=v+u,\ \forall u,v\in V$$

In Isogeometric Analysis this property applies as

$$C_1(\xi)+C_2(\xi)=C_2(\xi)+C_1(\xi)$$

2. Associativity of addition:

$$u+(v+w)=(u+v)+w,\ \forall u,v,w\in V$$

Respectively in Isogeometric Analysis,

$$C_1(\xi)+(C_2(\xi)+C_3(\xi))=(C_1(\xi)+C_2(\xi))+C_3(\xi)$$

3. Identity element of addition: There is an element $0\in V$, the zero vector, so that

$$u+0=u,\ \forall u\in V$$

In Isogeometric Analysis the equation applies as

$$C(\xi)+0=C(\xi)$$

4. Inverse elements of addition: There is an element $-u\in V$ such that

$$u+(-u)=0,\ \forall u\in V$$

The $-u$ is called the additive inverse of u. The equivalent in Isogeometric Analysis is

$$C(\xi)+(-C(\xi))=0$$

5. Identity element of scalar multiplication: For the real number 1 , it applies:

$$1\cdot u=u,\ \forall u\in V$$

Respectively, in Isogeometric Analysis

$$1\cdot C(\xi)=C(\xi)$$

6. Distributivity of scalar multiplication with respect to vector addition: $\forall a \in \square$ and $\forall u, v \in V$ applies the relation

$$a \cdot (u + v) = a \cdot u + a \cdot v$$

In Isogeometric Analysis this equation applies as

$$a \cdot (C_1(\xi) + C_2(\xi)) = a \cdot C_1(\xi) + a \cdot C_2(\xi)$$

7. Distributivity of scalar multiplication with respect to field addition: $\forall a, b \in \square$ and $\forall u \in V$ applies the relation

$$(a + b) \cdot u = a \cdot u + b \cdot u$$

which is implemented in Isogeometric Analysis as

$$(a + b) \cdot C(\xi) = a \cdot C(\xi) + b \cdot C(\xi)$$

8. Compatibility of scalar multiplication with field multiplication: $\forall a, b \in \square$ and $\forall u \in V$ applies the equation

$$a \cdot (b \cdot u) = (a \cdot b) \cdot u$$

Its equivalent in Isogeometric Analysis is

$$a \cdot (b \cdot C(\xi)) = (a \cdot b) \cdot C(\xi)$$

B-SPLine basis functions are indeed the basis for a vector space with $\dim(V) = n$, where n is the number of Control Points. Control Points are the coordinates that transform the basis functions at any point in the given Physical Space. Conclusively,

$$\{N_{1,p}(\xi) \quad N_{2,p}(\xi) \quad \ldots \quad N_{n,p}(\xi)\}$$

is the basis of the Vector space, while

$$\{X_1 \quad X_2 \quad \ldots \quad X_n\}$$

are the coordinates for a vector $C(\xi)$. The familiar linear combination applies:

$$C(\xi) = X_1 \cdot N_{1,p}(\xi) + X_2 \cdot N_{2,p}(\xi) + \ldots + X_n \cdot N_{n,p}(\xi)$$

Due to linear independence and vector space properties, it is understood that for a specific set of basis functions, only one set of Control Points can yield the appropriate geometry. If one wants to change the basis functions, the Control Points have to be shifted accordingly.

2.3.5.7 Fixed Number of Knot Values

A non-periodic Knot Value Vector that produces n functions of order p has $n+p+1$ Knot Values.

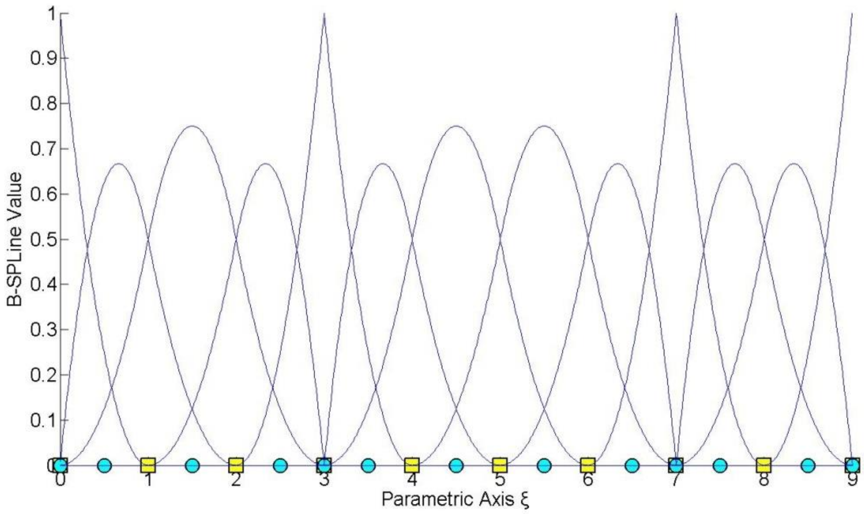


Figure 2.21. B-SPLine Basis functions for Knot Value Vector $\Xi=\{0\quad 0\quad 0\quad 1\quad 2\quad 3\quad 3\quad 4\quad 5\quad 6\quad 7\quad 7\quad 8\quad 9\quad 9\quad 9\}$

In this example the degree is $p=2$ and we have thirteen basis functions produced, so $n=13$.

We observe that the Knot Value Number is $k=n+p+1=13+2+1=16$. This occurs because for the creation of n basis functions of degree p, needed to construct the curve, $n+p$ basis functions of order $p=0$ are used, hence $n+p+1$ Knot Values are needed.

In another approach, each Control Point has a Knot Value Support of $p+2$ Knot Values. For n Control Points, $n\cdot(p+2)$ Knot Values are needed. There are $(p+1)$ Knot Values repeated in $(n-1)$ Control Point interconnections. The total number of Knot Values is:

$$n\cdot(p+2)-(n-1)\cdot(p+1)=n\cdot p+2n-n\cdot p-n+p+1=n+p+1$$

Therefore, $n+p+1$ Knot Values are required.

2.3.5.8 Shared Support

Each B-Spline shares support with at most $2p$ B-Splines. More specifically, each basis function shares support with at most p basis functions on each side. This results in higher interconnection, compared to equivalent Finite Element Method Shape functions. Basis function overlapping leads to interconnectivity between Control Points as well.

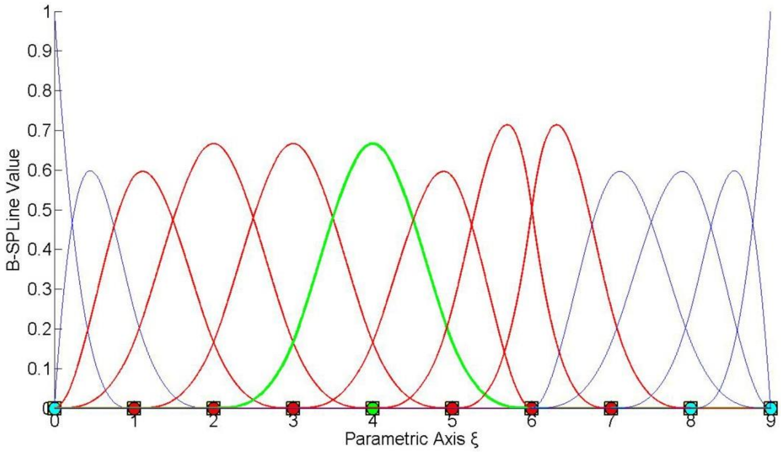


Figure 2.22. Shared support for $N_{6,3}(\xi)$.
$\Xi = \{0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 6 \quad 7 \quad 8 \quad 9 \quad 9 \quad 9 \quad 9\}$

$N_{6,3}(\xi)$ interacts with $p = 3$ basis functions on each side, $N_{3,2}(\xi)$, $N_{4,2}(\xi)$ and $N_{5,2}(\xi)$ on the left, $N_{7,2}(\xi)$, $N_{8,2}(\xi)$, $N_{9,2}(\xi)$ on the right. As a result, the respective positions in the Stiffness Matrix will be non-zero.

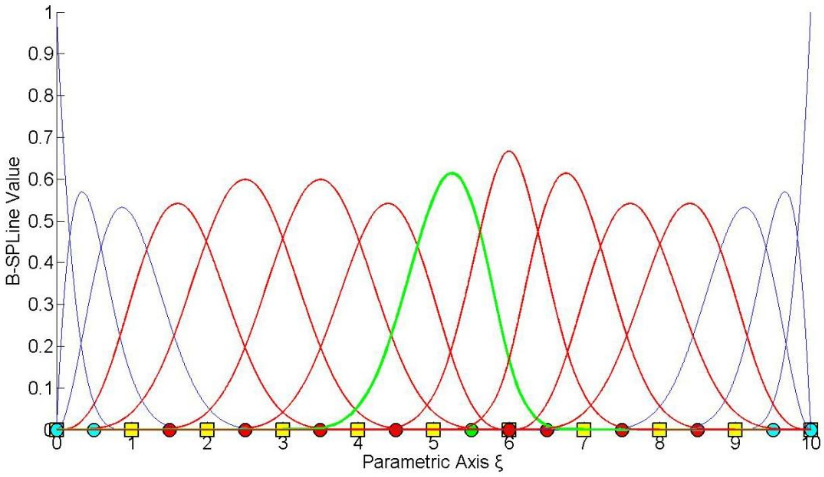


Figure 2.23. Shared support for $N_{8,4}(\xi)$.
$\Xi = \{0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 6 \quad 7 \quad 8 \quad 9 \quad 10 \quad 10 \quad 10 \quad 10 \quad 10\}$

$N_{8,4}(\xi)$ shares support with $2p = 8$ basis functions, 4 on each side. Greater-order basis functions tend to create more dense Stiffness Matrices and therefore demand more computational power.

2.3.6 B-SPLine Basis Function Derivatives

Basis function derivatives are widely used in Isogeometric Analysis. Deformation and Stiffness Matrices are built upon the derivatives of Shape functions. As a result, the distribution of stresses and strains across the model is based on those derivatives. The derivatives of B-SPLines, as obtained from the recursive formula, are represented as a linear combination of previous polynomial order basis functions [1]:

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

This leads to a generalized equation for the k^th derivative:

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

$$a_{0,0}=1$$

$$a_{k,0}=\frac{a_{k-1,0}}{\xi_{i+p-k+1}-\xi_i}$$

$$a_{k,j}=\frac{a_{k-1,j}-a_{k-1,j-1}}{\xi_{i+p+j-k+1}-\xi_{i+j}},\quad j=1,\dots,k\text{-}1,$$

$$a_{k,k}=\frac{-a_{k-1,k-1}}{\xi_{i+p+1}-\xi_{i+k}}$$

The partial derivatives of two-directional B-SPLine Shape functions can be easily obtained by application of the quotient rule:

$$\frac{\partial}{\partial\xi}R_{i,j}^{p,q}(\xi,\eta)=\left(\frac{d}{d\xi}N_{i,p}(\xi)\right)\cdot M_{j,q}(\eta)$$

$$\frac{\partial}{\partial\eta}R_{i,j}^{p,q}(\xi,\eta)=N_{i,p}(\xi)\cdot\left(\frac{d}{d\eta}M_{j,q}(\eta)\right)$$

3D Shape function derivatives per direction can be obtained in the same manner

$$\frac{\partial}{\partial\xi}R_{i,j,k}^{p,q,r}(\xi,\eta,\zeta)=\left(\frac{d}{d\xi}N_{i,p}(\xi)\right)\cdot M_{j,q}(\eta)\cdot L_{k,r}(\zeta)$$

$$\frac{\partial}{\partial\eta}R_{i,j,k}^{p,q,r}(\xi,\eta,\zeta)=N_{i,p}(\xi)\cdot\left(\frac{d}{d\eta}M_{j,q}(\eta)\right)\cdot L_{k,r}(\zeta)$$

$$\frac{\partial}{\partial\zeta}R_{i,j,k}^{p,q,r}(\xi,\eta,\zeta)=N_{i,p}(\xi)\cdot M_{j,q}(\eta)\cdot\left(\frac{d}{d\zeta}L_{k,r}(\zeta)\right)$$

2.3.7 B-SPLine Curves, Surfaces and Solids

Given a Knot Value Vector Ξ and a polynomial order p, we can evaluate the B-SPLine functions at every ξ. In order to create the B-SPLine curve, we also need a vector of coordinates for each basis function, the Control Points $X_i = \{X_i \quad Y_i \quad Z_i\}$. The curve is evaluated for every ξ as a linear combination of basis functions:

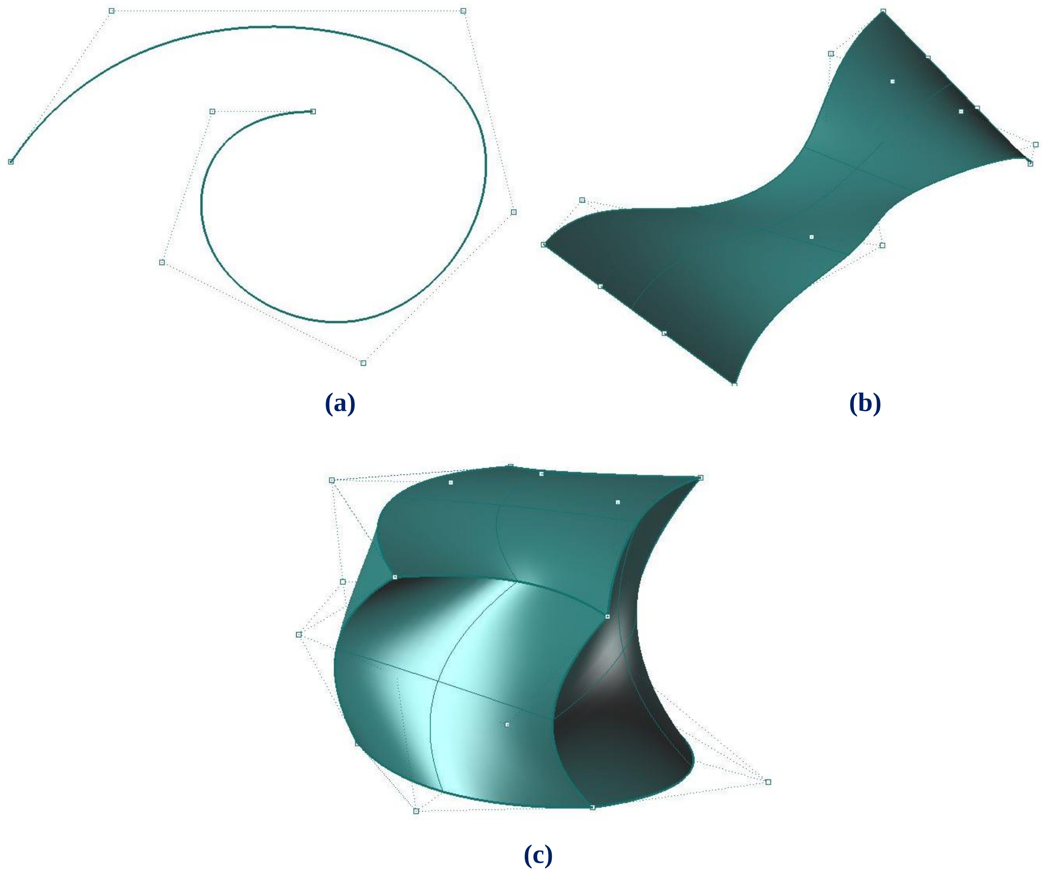
$$C(\xi) = \sum_{i=1}^{n} \left\{ N_{i,p}(\xi) \cdot X_i \right\}$$

After evaluating the Shape functions, the B-SPLine surface is defined in analogy to the B-SPLine curve:

$$S(\xi,\eta) = \sum_{i=1}^{n}\sum_{j=1}^{m} \left\{ N_{i,p}(\xi) \cdot M_{j,q}(\eta) \cdot X_{i,j} \right\} = \sum_{i=1}^{n}\sum_{j=1}^{m} \left\{ R_{i,j}^{p,q} \cdot X_{i,j} \right\}$$

Using the tensor product properties, we can also evaluate the solid function:

$$S(\xi,\eta,\zeta) = \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{l} \left\{ N_{i,p}(\xi) \cdot M_{j,q}(\eta) \cdot L_{k,q}(\zeta) \cdot X_{i,j,k} \right\} = \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{l} \left\{ R_{i,j,k}^{p,q,r}\left(\xi,\eta,\zeta\right) \cdot X_{i,j,k} \right\}$$


(a)

(b)

(c)

Figure 2.24. B-SPLine Entities. **(a)** Curve, **(b)** Surface and **(c)** Solid.

2.3.8 B-SPLine Curve Properties

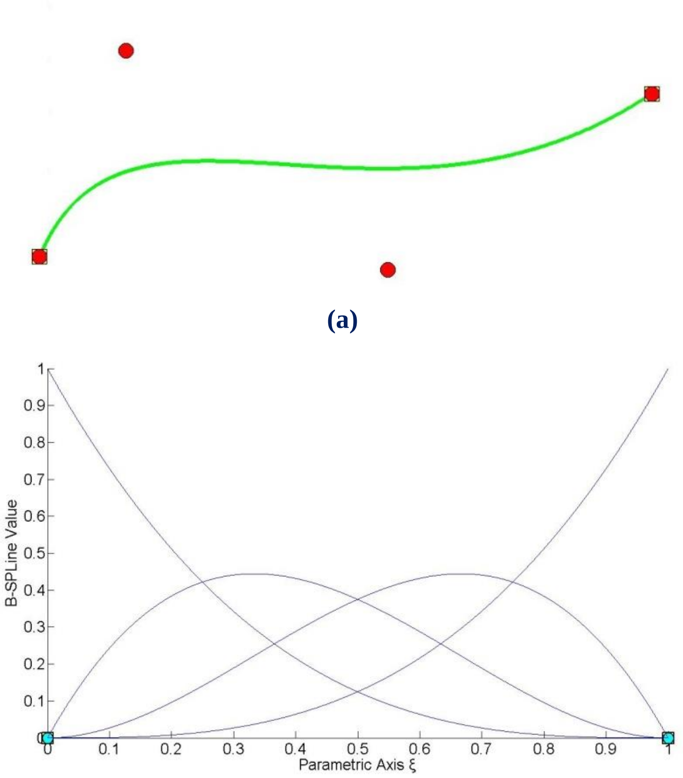
B-SPLine curve entities have the following properties, as obtained from Piegl, Tiller [3]:

1. B-SPLine curves are a generalization of Bezier curves.
2. $C(\xi)$ is a piecewise polynomial curve.
3. Each basis function corresponds to a certain Control Point.
4. The first and last Control Point as well as internal Control Points corresponding to C^0 Continuous basis functions are interpolatory to the curve.
5. B-SPLine curves possess strong convex hull property.
6. Moving a Control Point X_i only changes part of the curve.
7. The Control Polygon represents a piecewise linear approximation to the curve.
8. It is possible to use multiple Control Points with the same coordinates.
9. Any transformation applied to the curve can be applied directly at the Control Points. This property is known as “affine invariance” or “affine covariance”.
10. In every Knot Span, at most p+1 Control Points contribute to the definition of the curve, corresponding to the $p+1$ non-zero basis functions.
11. Since $C(\xi)$ is a linear combination of $N_{i,p}(\xi)$, curve Continuity and differentiability are obtained straight from the basis functions.
12. No line has more intersections with the curve, than with the Control Polygon.

2.3.8.1 Generalization of Bezier curves

B-SPLine curves are a generalization of Bezier curves. Given an open Knot Vector and $n=p+1$ Control Points, a Bezier curve is produced.

Bezier curves were utilized in CAD methods before B-SPLines. Figure 2.25 presents a Bezier curve with $p=3$, an open Knot Value Vector $\{0\quad 0\quad 0\quad 0\quad 1\quad 1\quad 1\quad 1\}$ and $n=p+1=4$ Control Points. Bezier curves are B-SPLine curves defined in only one Knot Span. As a result, every basis function is non-zero across the entire parameter space and Control Points affect the shape of the entire curve. Bezier surfaces are a special case of one-rectangle B-SPLine surfaces and Bezier solids a special case of one-cuboid B-SPLine solids as well.



(a)

(b)

Figure 2.25. B-SPLine as generalization of Bezier curves. **(a)** Physical Space and **(b)** basis functions

2.3.8.2 Piecewise polynomial curve

$C(\xi)$ is formed from piecewise polynomials $N_{i,p}(\xi)$, and therefore is a piecewise polynomial curve.

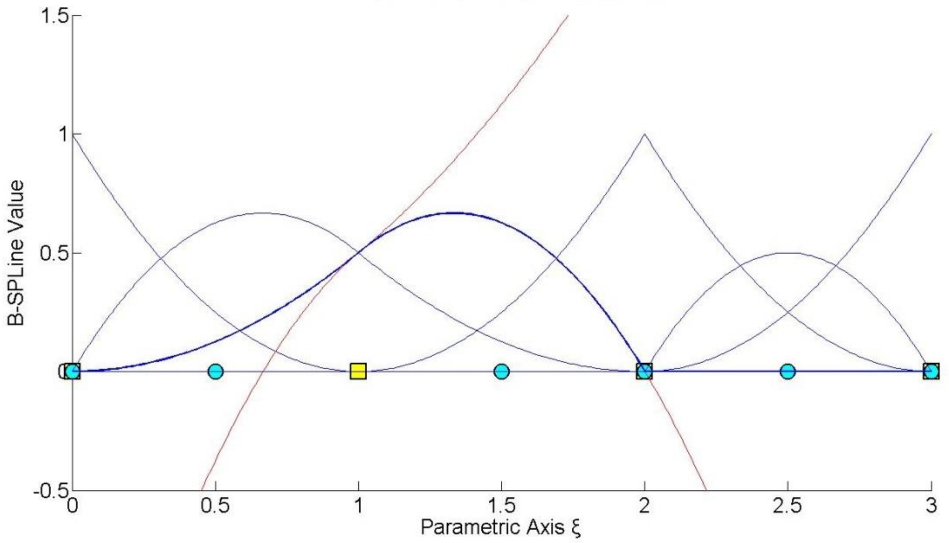


Figure 2.26 Piecewise polynomials that form a B-SPLine basis function. Knots represent the boundaries of the pieces.

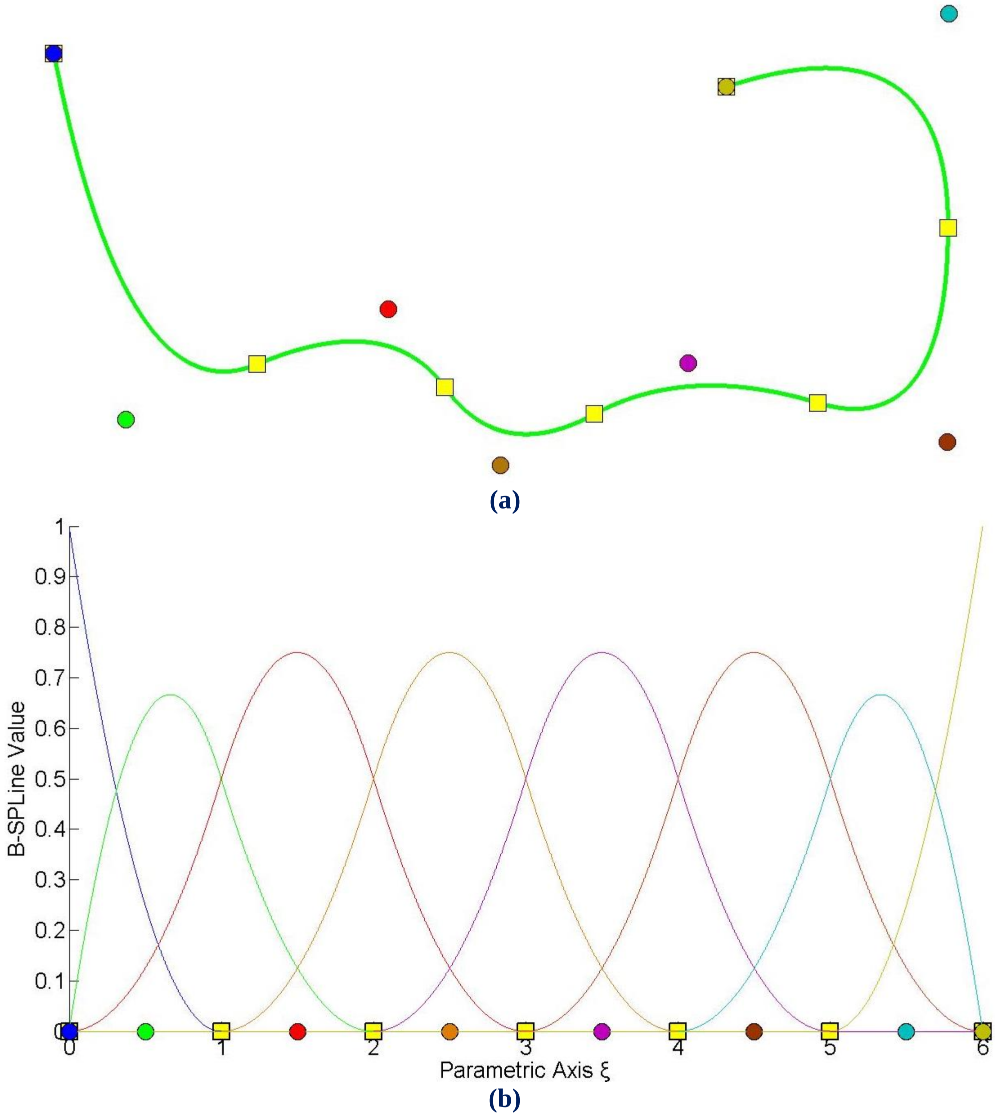
A B-SPLine curve is obtained through the curve function:

$$C(\xi)=\sum_{i=1}^{n} N_{i,p}(\xi)\cdot X_i$$

which is a linear combination of piecewise polynomial basis functions. This applies in multi-directional entities as well. Tensor product Shape functions are piecewise polynomials with respect to the corresponding directions, thus B-SPLine Surfaces and Solids are also piecewise polynomials. The term B-SPLine, after all, stands for Basis - Smooth Polynomial Line.

2.3.8.3 Control Point – Basis Function Correspondence

Each basis function corresponds to a certain Control Point. There are n basis functions and n Control Points in a B-SPLine curve.



(a)

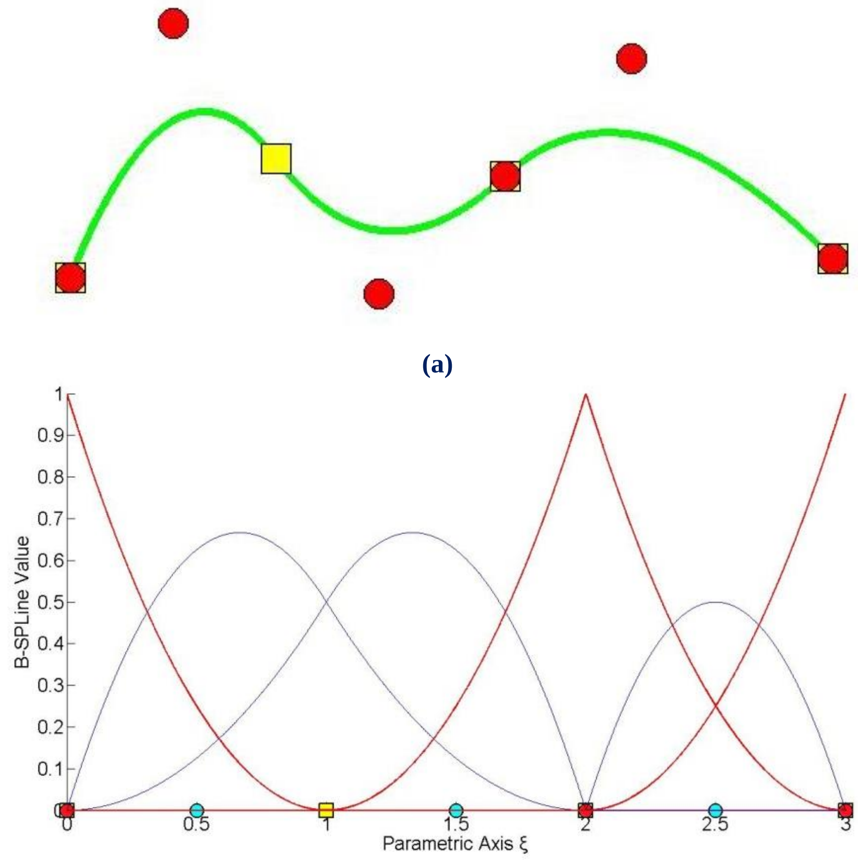
(b)

Figure 2.27. (a) Physical Space and **(b)** basis functions with the corresponding Control Points.

In Figure 2.27, Control Points are represented both in Parameter and Physical Space. Each Point controls a specific basis function. This property also applies for multiple directions. Every Control Point of the surface or the solid is tensor product of a Control Point in directions ξ, η and ζ. By extension, the corresponding B-SPLine is tensor product of the basis functions. In the most general case, there are $n \cdot m \cdot l$ Control Points and basis functions.

2.3.8.4 Interpolation to the Curve

The first and last Control Points are interpolatory to the curve. Any internal Control Point corresponding to C^0 Continuous basis function is also interpolatory to the curve.



(a)

(b)

Figure 2.28. Control Point interpolation. **(a)** B-SPLine curve and **(b)** the reciprocal basis functions

In Figure 2.28, the first and the last Control Point, which have C^{-1} Continuity, are interpolatory to the curve. This can be explained with the help of the equation of the curve:

$$C(\xi) = \sum_{i=1}^{n} \left\{ N_{i,2}(\xi) \cdot X_i \right\}$$

For $\xi = 0$, it applies that:

$$C(0) = \sum_{i=1}^{n} N_{i,2}(0) \cdot X_i$$

where,

$$N_{1,2}(0) = 1$$

$$N_{i,2}(0) = 0, \ i = 2,...,6$$

so,

$$C(0) = N_{1,2}(0) \cdot X_1 = X_1$$

And for $\xi = 3$:

$$C(3) = \sum_{i=1}^{n} N_{i,2}(3) \cdot X_i$$

$$N_{6,2}(3) = 1$$

$$N_{i,2}(3) = 0, \ i = 1,...,5$$

so,

$$C(3) = N_{6,2}(3) \cdot X_6 = X_6$$

Likewise, the internal Control Point, with C^0 Continuity across $\xi = 2$ is interpolatory to the curve because:

$$C(2) = \sum_{i=1}^{n} \left\{ N_{i,2}(2) \cdot X_i \right\}$$

$N_{4,2}(2) = 1$, as this is the only non-zero basis function across $\xi = 2$

$$N_{i,2}(2) = 0, \ i \neq 4$$

so,

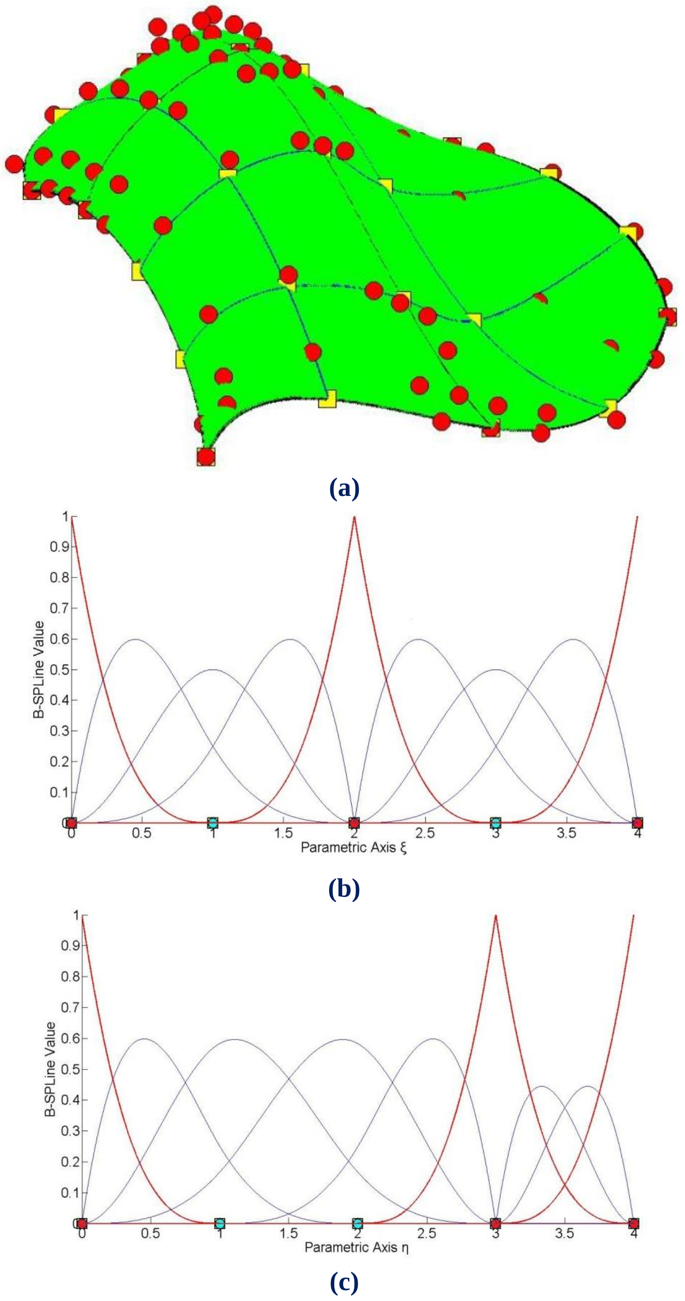
$$C(2) = N_{4,2}(2) \cdot X_4 = X_4$$

Observe that both the form of the curve and the form of the basis functions indicate that this geometry could be represented by two different sets of Knot Vectors and Control Points, with absolutely no deflections from the current representation. This will be examined thoroughly later.

Interpolation also applies for Surfaces and Solids, when appropriately reduced Continuity is used for all directions at a Knot. C^{-1} Continuity is required for external Knots and C^0 for internal.

In Figure 2.29, $N_{5,2}(\xi)$ at axis ξ and $M_{6,2}(\eta)$ at axis η have C^0 Continuity. Therefore the Control Point corresponding to $R^{2,2}_{5,6}(\xi,\eta)$ is interpolatory to the surface. The external Control Points with C^{-1} Continuity at both directions are also interpolatory to the surface.

In Figure 2.30, $N_{4,2}(\xi)$ at parametric axis ξ and $M_{4,2}(\eta)$ at parametric axis η have C^0 Continuity, thus Control Points corresponding to the Shape functions $R^{2,2,1}_{4,4,1}(\xi,\eta,\zeta)$ and $R^{2,2,1}_{4,4,2}(\xi,\eta,\zeta)$, are interpolatory to the solid. As we can see, the external Control Points are also interpolatory to the solid, because Continuity is reduced to C^{-1}.

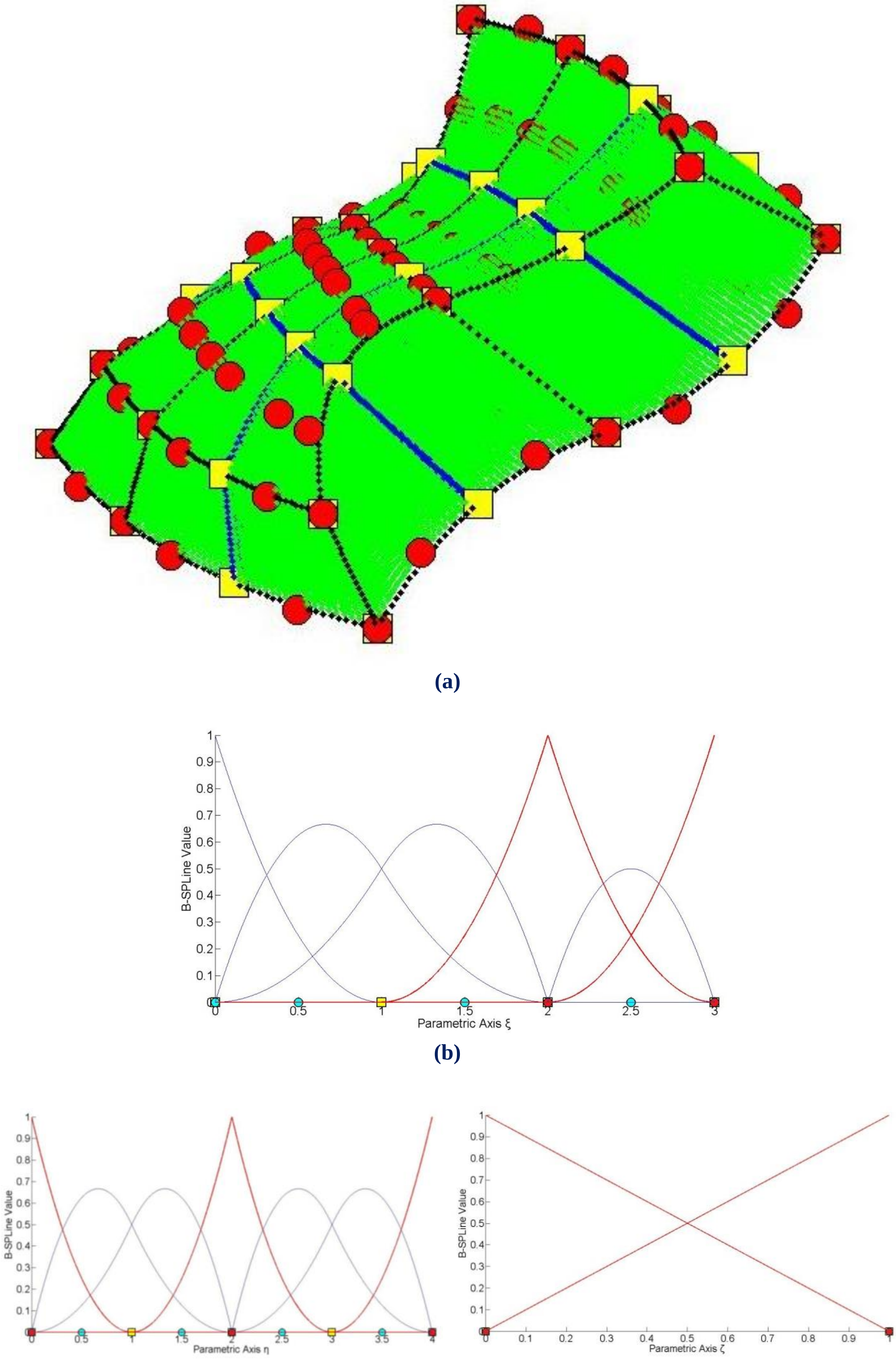


(a)

(b)

(c)

Figure 2.29. (a) B-SPLine Surface and **(b), (c)** the corresponding basis functions in axes ξ, η .



(a)

(b)

(c)

(d)

Figure 2.30. (a) Solid in the Physical Space
with **(b), (c), (d)** the associated basis functions in axes ξ, η, ζ .

2.3.8.5 Convex Hull

B-SPLine curves possess strong convex hull property. The convex hull of the curve is defined as the sum of the convex hulls of p+1 consecutive Control Points. The curve is always contained in the convex hull.

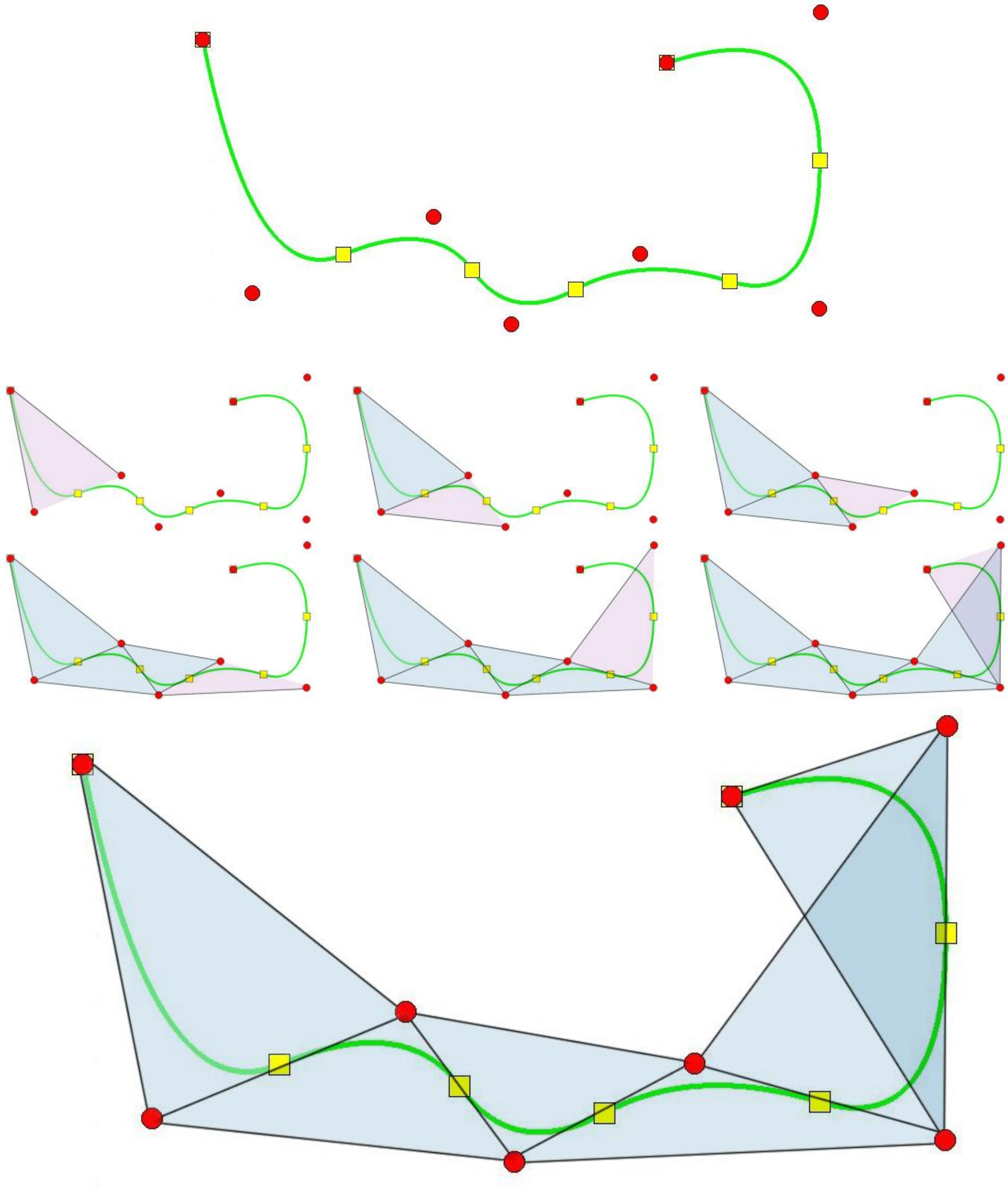


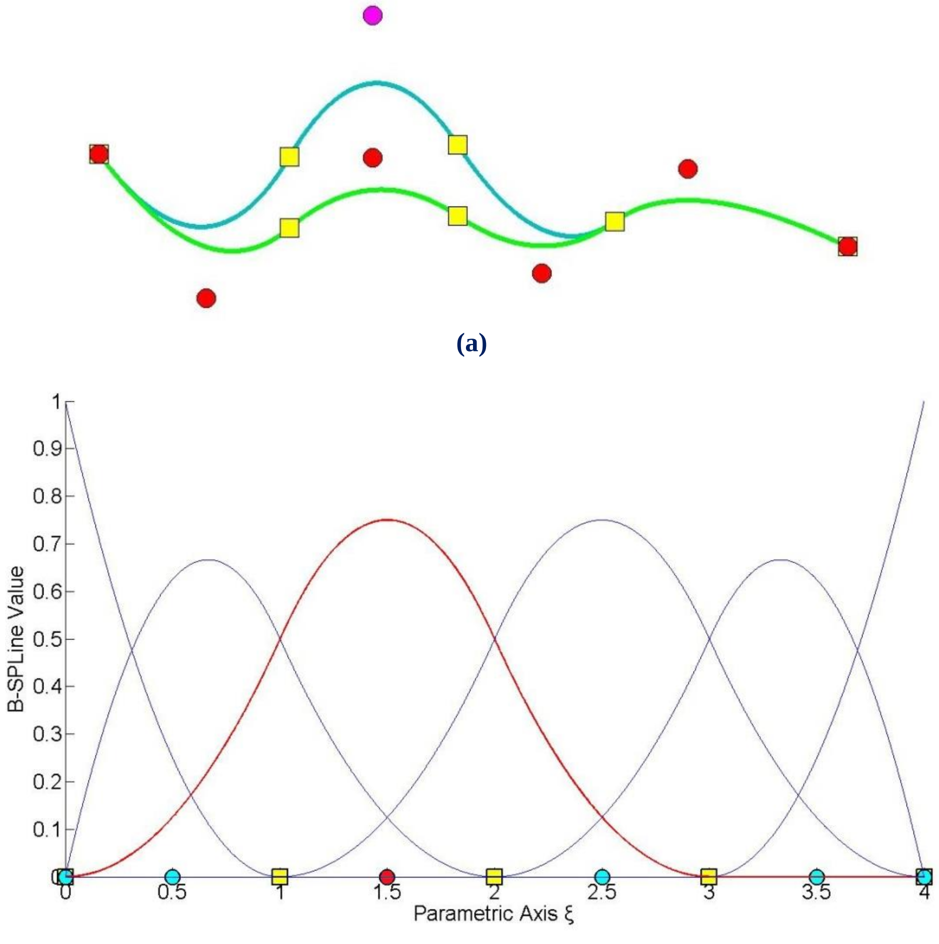
Figure 2.31. Step-by-step convex hull creation for a B-SPLine curve.

The curve in Figure 2.31 has a degree of $p = 2$. The convex hull is formed by connecting each Control Point with the $p = 2$ successive ones. As we can easily see in the figure, the union of the convex hulls contains the curve. The convex hull is a way to assume the general form of a B-SPLine curve.

2.3.8.6 Control Point Local support

Moving a Control Point X_i only changes part of the curve, more specifically the part corresponding to the $[\xi_i, \xi_{i+p+1})$ Knot Value Spans. This is a result of the local support of the corresponding B-SPLine function.

Moving along the curve, basis function $N_{i,p}(\xi)$ is switched "on" at the Knot Value ξ_i and then again switched "off" at the Knot Value ξ_{i+p+1}, where the function $N_{i+p+1,p}(\xi)$ is switched "on".



(a)

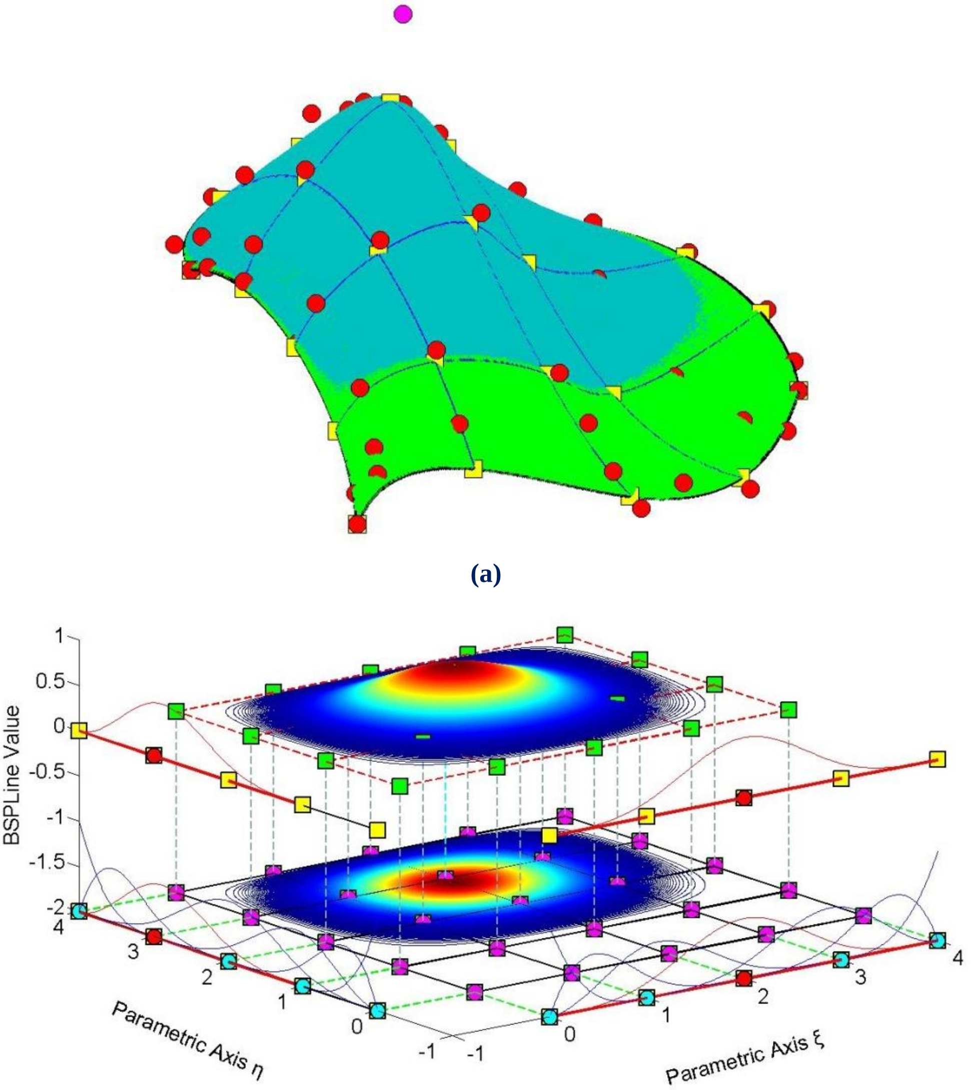
(b)

Figure 2.32. Control Point Local support **(a)** in Physical and **(b)** in Parameter Space.

Every Control Point is associated with a basis function. Support of the Control Point is defined by the support of the corresponding basis function. Therefore, Control Points affect only part of the curve. In Figure 2.32, a curve with $p = 2$ and Knot Value Vector $\{0 \quad 0 \quad 0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 4 \quad 4\}$, is presented. The Control Point for $i = 3$ is moved and this affects partially the entity.

In this example, $N_{3,2}(\xi)$ spans from $\xi = 0$ to $\xi = 3$. The Knots act as boundaries of the support. For $\xi = 3$, $N_{3,2}(\xi)$ is switched "off" and $N_{6,2}(\xi)$ is switched "on".

Local support of Control Points is also expanded by tensor product properties. The local support of a Multi-directional Control Point is the tensor product of the respective supports.



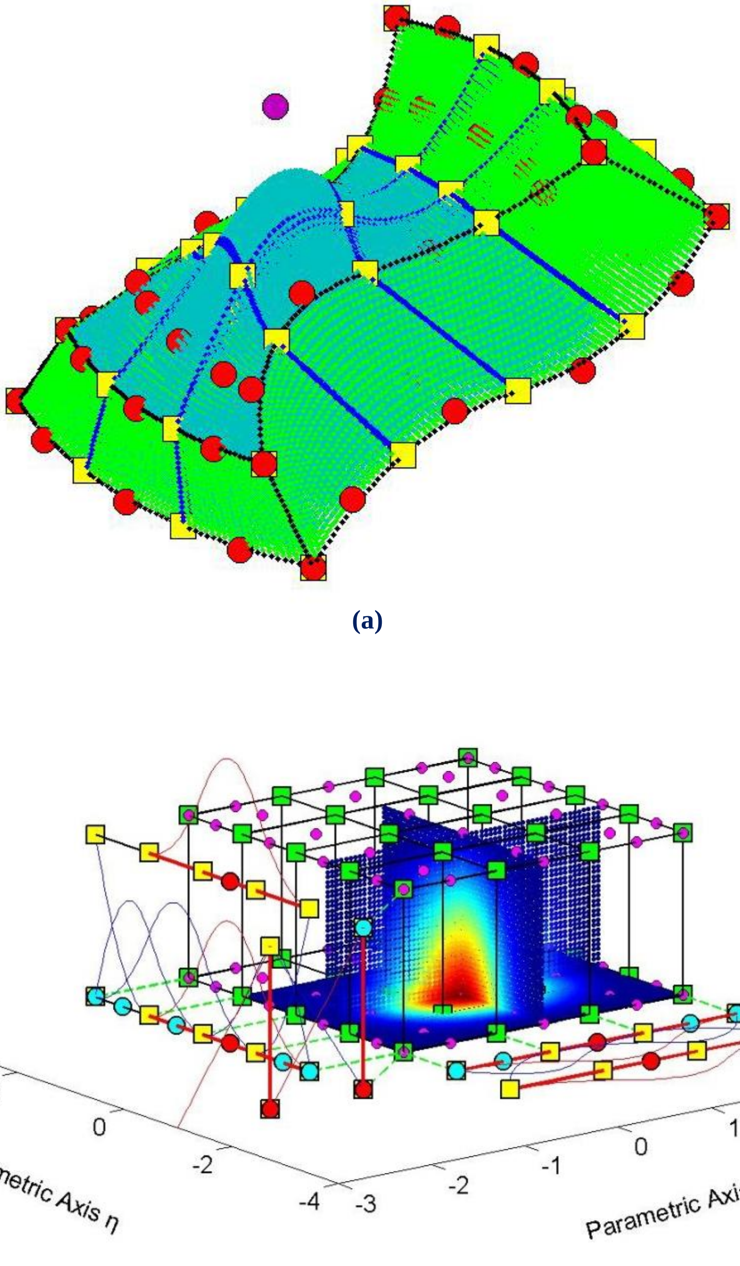
(a)

(b)

Figure 2.33. Local support of a 2D Control Point.
Surface **(a)** in Physical and **(b)** in Parameter Space.

Figure 2.33 represents the local support in the parametric axis ξ, η of $N_{4,2}(\xi)$ and $M_{3,2}(\eta)$ B-SPLine curves respectively. In parametric axis ξ the local support expands throughout the axis, whereas in parametric axis η the basis function is switched "on" at the second Knot Span. In Figure 2.33.a, the tensor product of the respective supports for ξ, η is represented in cyan. It spans across $4x3 = 12$ Knot Rectangles in total.

The same properties apply for B-SPLine solids as well.


(a)

(b)

Figure 2.34. Local support of a 3D Control Point.
Solid **(a)** in Physical and **(b)** in Parameter Space.

Figure 2.34.b presents the local support of $N_{3,2}(\xi)$, $M_{4,2}(\eta)$ and $L_{1,1}(\zeta)$. In parametric axes ξ, ζ, the local support expands at all Knot Spans, whereas in the parametric axis η the local support spans between Knots 1 and 4. In Figure 2.34.a, the local support is displayed in cyan and it does not reach all the Knot Spans in parametric axis η. A total of $3 \cdot 3 \cdot 1 = 9$ Knot Cuboids represent the Support of the Control Point.

2.3.8.7 Control Polygon approximation

The Control Polygon represents a piecewise linear approximation to the curve. Due to convex hull properties, Refinement by knot insertion or order elevation brings the Control Polygon closer to the curve.

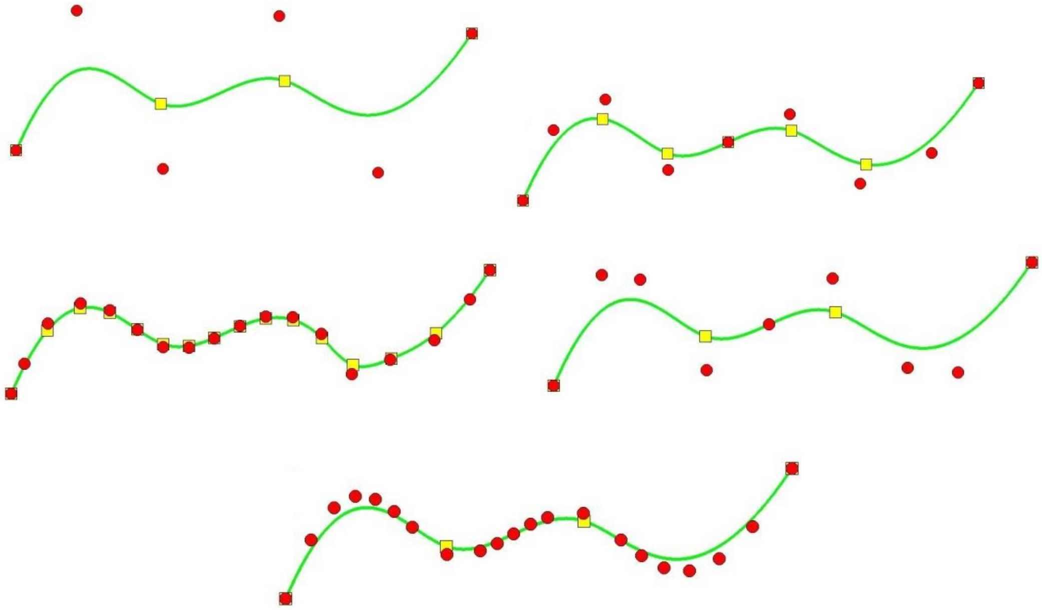


Figure 2.35. Control Polygon approximation through Refinement.

In Figure 2.35 a curve of degree $p = 3$ is designed. The Control Polygon already represents a linear approximation to the curve. When consecutive h- or p-Refinements are applied, the Control Polygon is brought even closer to the curve. Refined Control Polygons provide a general idea of the form of the curve. This property also applies for multiple directions.

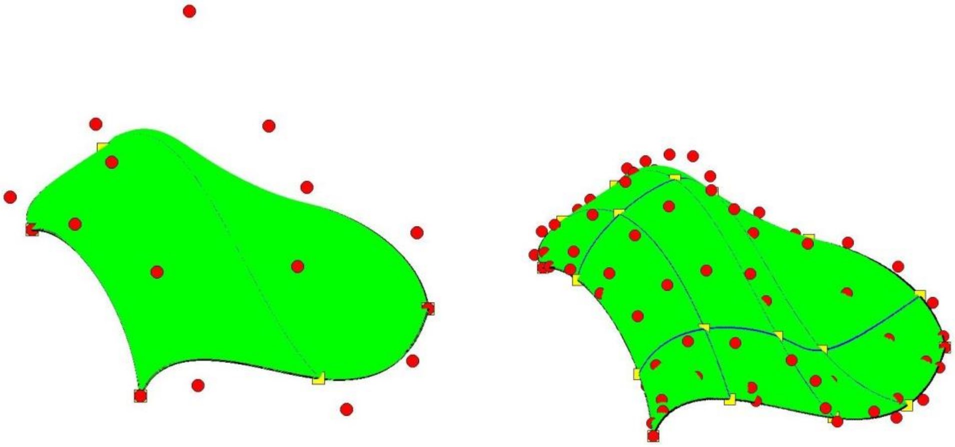
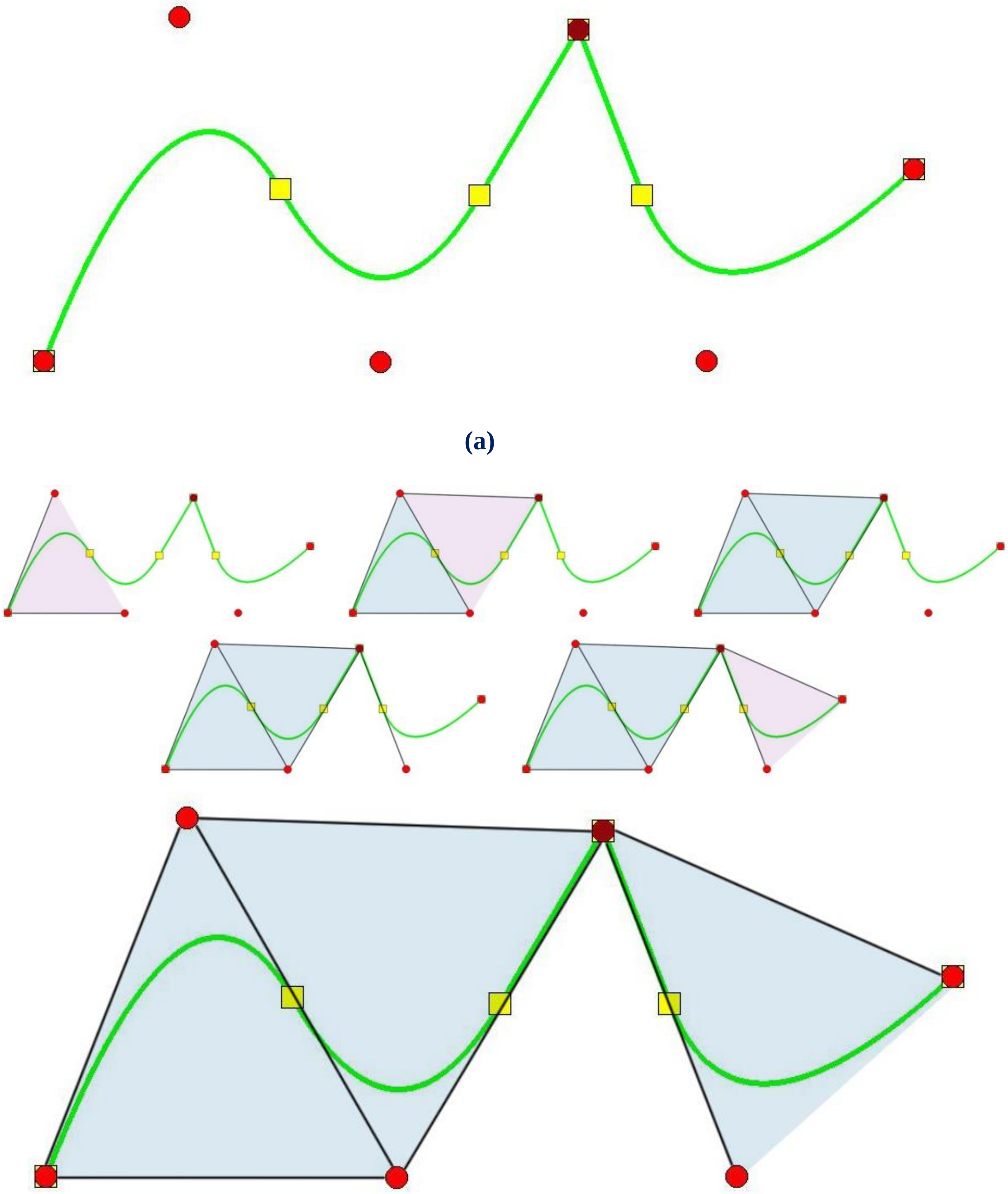


Figure 2.36. Control Net approximation through Surface h-Refinement.

For example, in Figure 2.36, the Refinements, that were made, brought the Control Net closer to the surface.

2.3.8.8 Multiple Control Points

It is possible to use multiple Control Points with the same coordinates. This can prove to be very useful.



(a)

(b)

Figure 2.37. (a) Physical Space and **(b)** Convex Hull creation for a curve with two coincident Control Points (drawn in deep red).

In Figure 2.37.a, a quadratic curve with a double Control Point is designed. The curve is interpolatory at these Points and a sharp edge is formed. This is explained in Figure 2.37.b, where the convex hull of the curve is designed. The curve is always contained in the convex hull, therefore a sharp edge has to be formed exactly at the double Point coordinates. Inductively, this applies when p coincident Control Points are used in a curve of polynomial degree p.

2.4 Non-Uniform Rational B-SPLines

B-SPLine geometries may have many promising attributes, but they also have several weaknesses in geometrical representation. Some basic geometrical forms cannot be presented as B-SPLine entities, such as circles or conic sections in general. In order to solve this problem, the CAD industry introduced Non-Uniform Rational B-Splines (NURBS). The basic idea is simple. A NURBS entity is produced from the actual section of a cone with a plane.

2.4.1 Basic Idea

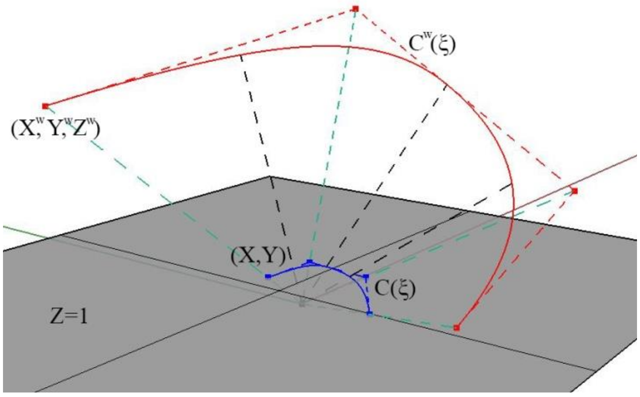


Figure 2.38. B-SPLine curve and projective transformation to NURBS Curve.

As shown in Figure 2.38, the projective B-SPLine curve $C^w(\xi)$ is created from the projective 3D Control Points $X^w = \{X^w \quad Y^w \quad Z^w\}$.

Projection of the curve and Control Points on the plane $z = 1$ produces the NURBS curve $C(\xi)$ and the 2D Control Points:

$$X = \{X \quad Y\}$$

where,

$$\{X_i \quad Y_i\} = \left\{ \frac{X_i^w}{Z_i^w} \quad \frac{Y_i^w}{Z_i^w} \right\}$$

The Weights of the NURBS curve are defined as:

$$w = \{Z^w\}$$

In generalization, n-dimensional Rational B-SPLines are projections of $(n+1)$-dimensional non-rational B-SPLines.

2.4.2 NURBS Shape Functions

In order to evaluate NURBS Shape functions, the Weighting function is defined:

$$W(\xi)=\sum_{i=1}^{n}\left\{N_{i,p}(\xi)\cdot w_i\right\}$$

In most engineering applications, Weights have positive values. Unless otherwise stated, they will be considered positive for the scope of this thesis. $W(\xi)$ is in fact the Z-coordinate of the projective B-SPLine curve. Projective transformation is applied by dividing the other two coordinates of the B-SPLine curve with the Z-coordinate. NURBS Shape functions are calculated from

$$R_i^p(\xi)=\frac{N_{i,p}(\xi)\cdot w_i}{W(\xi)}=\frac{N_{i,p}(\xi)\cdot w_i}{\sum\limits_{i'=1}^{n}\left\{N_{i',p}(\xi)\cdot w_{i'}\right\}}$$

$R_i^p(\xi)$ are piecewise rational functions. The expression "the order of NURBS" refers to the order of the projective B-SPLine curve.

NURBS Shape Functions in multiple directions can be obtained as tensor products of one-directional basis functions:

Shape functions for two directions:

$$R_{i,j}^{p,q}(\xi,\eta)=\frac{N_{i,p}(\xi)\cdot M_{j,q}(\eta)\cdot w_{ij}}{\sum\limits_{i'=1}^{n}\sum\limits_{j'=1}^{m}\left\{N_{i',p}(\xi)\cdot M_{j',q}(\eta)\cdot w_{i'j'}\right\}}$$

$$W(\xi,\eta)=\sum_{i'=1}^{n}\sum_{j'=1}^{m}\left\{N_{i',p}(\xi)\cdot M_{j',q}(\eta)\cdot w_{i'j'}\right\}$$

Similarly, we expand the equations in order to obtain tensor product 3D Shape functions:

$$R_{i,j,k}^{p,q,r}(\xi,\eta,\zeta)=\frac{N_{i,p}(\xi)\cdot M_{j,q}(\eta)\cdot L_{k,r}(\zeta)\cdot w_{ijk}}{\sum\limits_{i'=1}^{n}\sum\limits_{j'=1}^{m}\sum\limits_{k'=1}^{l}\left\{N_{i',p}(\xi)\cdot M_{j',q}(\eta)\cdot L_{k',r}(\zeta)\cdot w_{i'j'k'}\right\}}$$

The Weighting function is now defined as:

$$W(\xi,\eta,\zeta)=\sum_{i'=1}^{n}\sum_{j'=1}^{m}\sum_{k'=1}^{l}\left\{N_{i',p}(\xi)\cdot M_{j',q}(\eta)\cdot L_{k',r}(\zeta)\cdot w_{i'j'k'}\right\}$$

Observe that for $w_{ijk}=1,\ \forall\, i,j,k$, it applies that NURBS Shape functions downgrade to B-SPLine basis functions. Actually, NURBS entities are a generalization of B-SPLine entities. All the B-SPLine properties examined in this thesis apply for NURBS as well.

2.4.3 NURBS Shape Function Derivatives

Simple application of the quotient rule yields the derivatives of NURBS Shape functions for one and multiple directions.

$$\frac{d}{d\xi}R_i^p(\xi)=w_i\cdot\frac{\left(\frac{d}{d\xi}N_{i,p}(\xi)\right)\cdot W(\xi)-\left(\frac{d}{d\xi}W(\xi)\right)\cdot N_{i,p}(\xi)}{\left(W(\xi)\right)^2}$$

where

$$\frac{d}{d\xi}W(\xi)=\sum_{i=1}^{n}\left(\frac{d}{d\xi}N_{i,p}(\xi)\right)\cdot w_i$$

For bidirectional Shape functions:

$$\frac{\partial}{\partial\xi}R_{i,j}^{p,q}(\xi,\eta)=w_{ij}\cdot\frac{\left(\frac{d}{d\xi}N_{i,p}(\xi)\right)\cdot M_{j,q}(\eta)\cdot W(\xi,\eta)-\left(\frac{\partial}{\partial\xi}W(\xi,\eta)\right)\cdot N_{i,p}(\xi)\cdot M_{j,q}(\eta)}{\left(W(\xi,\eta)\right)^2}$$

$$\frac{\partial}{\partial\eta}R_{i,j}^{p,q}(\xi,\eta)=w_{ij}\cdot\frac{N_{i,p}(\xi)\cdot\left(\frac{d}{d\eta}M_{j,q}(\eta)\right)\cdot W(\xi,\eta)-\left(\frac{\partial}{\partial\eta}W(\xi,\eta)\right)\cdot N_{i,p}(\xi)\cdot M_{j,q}(\eta)}{\left(W(\xi,\eta)\right)^2}$$

Derivatives of 3D Shape functions per direction are evaluated as shown

$$\frac{\partial}{\partial\xi}R_{i,j,k}^{p,q,r}(\xi,\eta,\zeta)=$$

$$=w_{ijk}\cdot\frac{\left(\frac{d}{d\xi}N_{i,p}(\xi)\right)\cdot M_{j,q}(\eta)\cdot L_{k,r}(\zeta)\cdot W(\xi,\eta,\zeta)-\left(\frac{\partial}{\partial\xi}W(\xi,\eta,\zeta)\right)\cdot N_{i,p}(\xi)\cdot M_{j,q}(\eta)\cdot L_{k,r}(\zeta)}{\left(W(\xi,\eta,\zeta)\right)^2}$$

$$\frac{\partial}{\partial\eta}R_{i,j,k}^{p,q,r}(\xi,\eta,\zeta)=$$

$$=w_{ijk}\cdot\frac{N_{i,p}(\xi)\cdot\left(\frac{d}{d\eta}M_{j,q}(\eta)\right)\cdot L_{k,r}(\zeta)\cdot W(\xi,\eta,\zeta)-\left(\frac{\partial}{\partial\eta}W(\xi,\eta,\zeta)\right)\cdot N_{i,p}(\xi)\cdot M_{j,q}(\eta)\cdot L_{k,r}(\zeta)}{\left(W(\xi,\eta,\zeta)\right)^2}$$

$$\frac{\partial}{\partial\zeta}R_{i,j,k}^{p,q,r}(\xi,\eta,\zeta)=$$

$$=w_{ijk}\cdot\frac{N_{i,p}(\xi)\cdot M_{j,q}(\eta)\cdot\left(\frac{d}{d\zeta}L_{k,r}(\zeta)\right)\cdot W(\xi,\eta,\zeta)-\left(\frac{\partial}{\partial\zeta}W(\xi,\eta,\zeta)\right)\cdot N_{i,p}(\xi)\cdot M_{j,q}(\eta)\cdot L_{k,r}(\zeta)}{\left(W(\xi,\eta,\zeta)\right)^2}$$

2.4.4 NURBS Entities

NURBS entities are created as a linear combination of NURBS Shape functions, exactly the same way as B-SPLine entities. The following is the equation for the creation of NURBS Curves:

$$C(\xi) = \sum_{i=1}^{n} \left\{ R_i^p(\xi) \cdot X_i \right\}$$

Surfaces:

$$S(\xi, \eta) = \sum_{i=1}^{n} \sum_{j=1}^{m} \left\{ R_{i,j}^{p,q}(\xi, \eta) \cdot X_{i,j} \right\}$$

and Solids:

$$S(\xi, \eta, \zeta) = \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{l} \left\{ R_{i,j,k}^{p,q,r}(\xi, \eta, \zeta) \cdot X_{i,j,k} \right\}$$

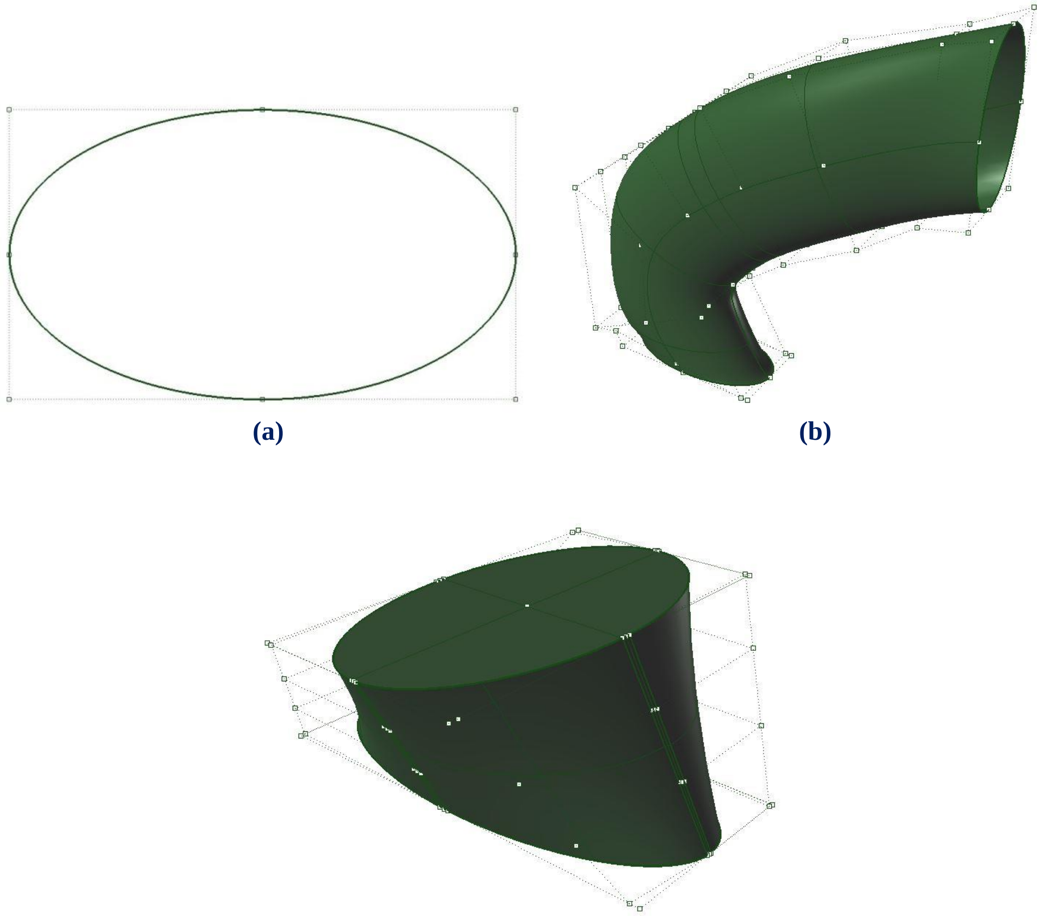
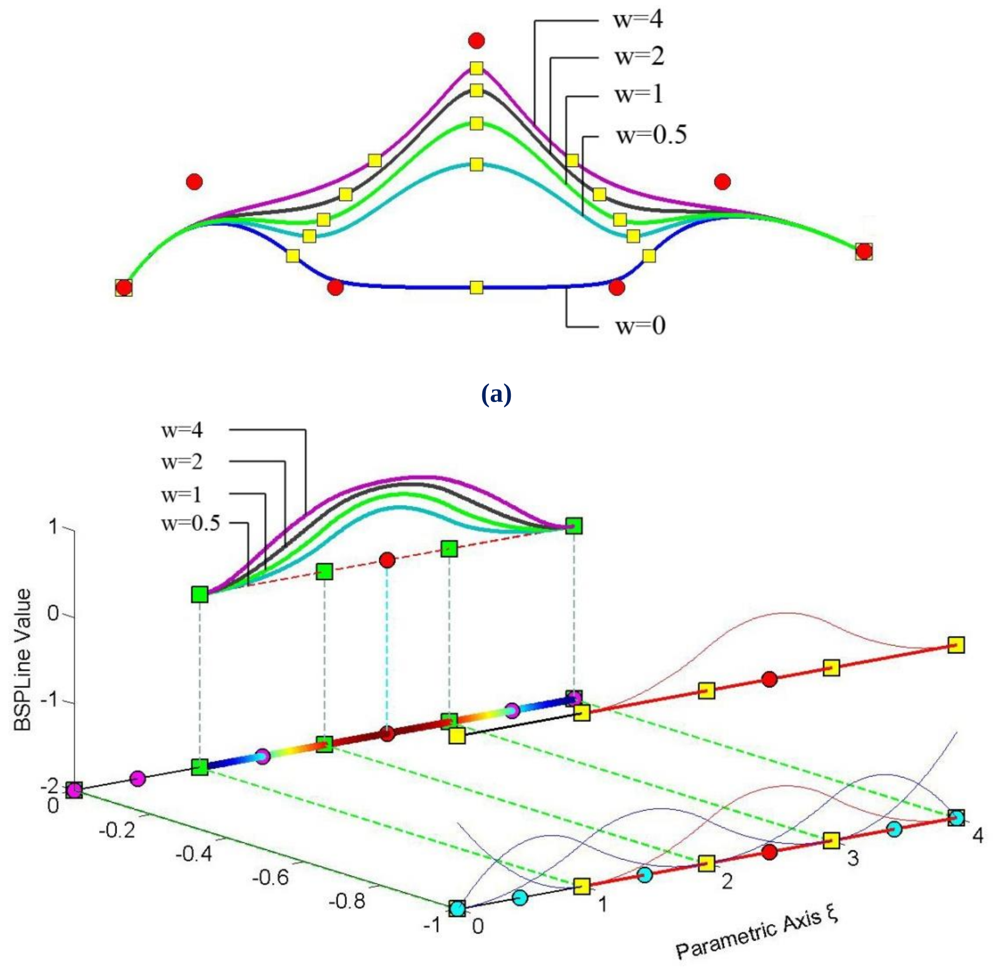


Figure 2.39. NURBS elliptical Entities.
(a) Curve, **(b)** Surface and **(c)** Solid.

2.4.5 NURBS Examples



(a)

(b)

Figure 2.40. (a) NURBS curves and **(b)** Shape Functions for different Weight values.

In Figure 2.40, five NURBS Curves with the same set of Control Point Coordinates are represented. The corresponding Weights are $w_i = 1$, for $i \neq 3$. The third Control Point has a different Weight for each curve. Observe that, as the Weight value increases, the Shape function and, as a result, the corresponding Control Point tends to dominate the $p+1$ Knot Spans of the support. Thus, the Knots are gravitated closer to the corresponding Control Point.

According to [1], in order to accurately represent an arc of $\theta < 180$ degrees, three Control Points are required. Weights for the first and last Points are $w_1 = w_3 = 1$, whereas the middle one has a weight of $w_2 = \cos\left(\frac{\theta}{2}\right)$.

In Figure 2.41, the same circle is represented by NURBS Shape functions of different order. This is a closed curve, so the first and last Control Points coincide. Weights are shown for each Control Point. A NURBS circle is usually represented by four consecutive Patches, bound together by a common Knot Value Vector.

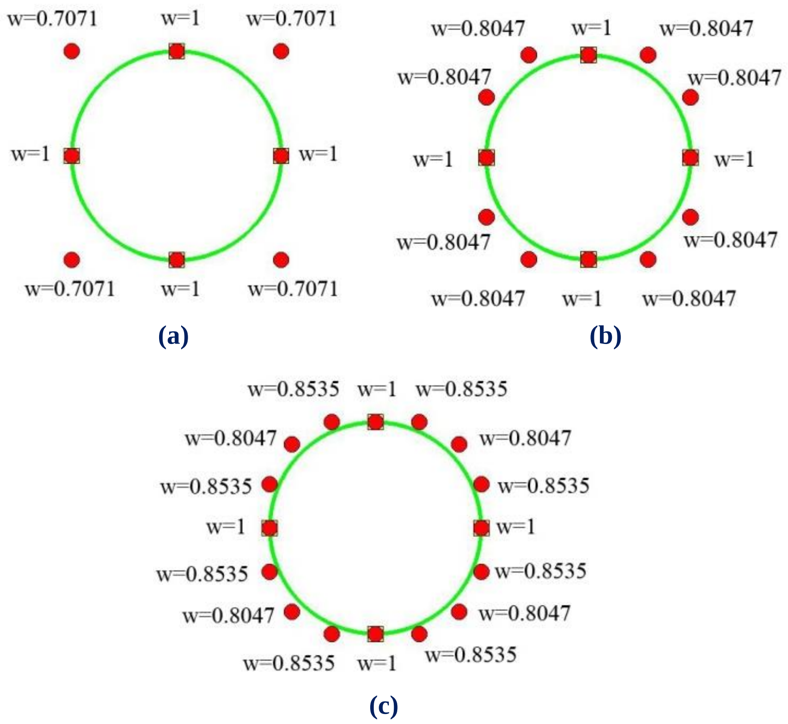


Figure 2.41. NURBS Circle of different polynomial degree. Weight values for each Control Point.

(a) Quadratic basis functions. Knot Value Vector

$$\Xi=\{0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 2 \quad 2 \quad 3 \quad 3 \quad 4 \quad 4 \quad 4\}$$

(b) Cubic basis functions. Knot Value Vector

$$\Xi=\{0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1 \quad 2 \quad 2 \quad 2 \quad 3 \quad 3 \quad 3 \quad 4 \quad 4 \quad 4 \quad 4\}$$

(c) Quadric basis functions. Knot Value Vector

$$\Xi=\{0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1 \quad 1 \quad 2 \quad 2 \quad 2 \quad 2 \quad 3 \quad 3 \quad 3 \quad 3 \quad 4 \quad 4 \quad 4 \quad 4 \quad 4\}$$

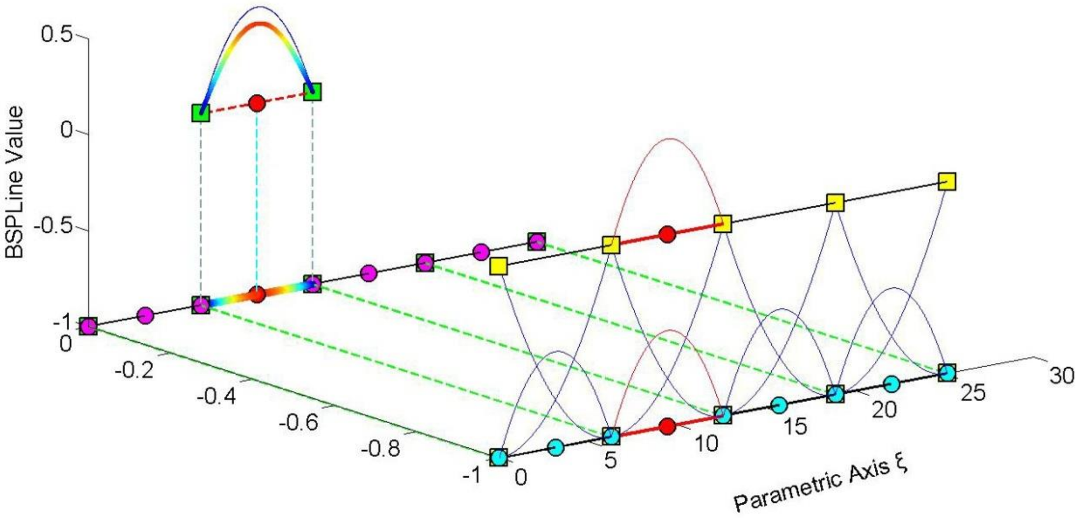


Figure 2.42. Basis functions for circle represented in Figure 2.41.

Contour distribution for Shape Function $R_5^2(\xi)$, with corresponding Weight $\frac{\sqrt{2}}{2}=0.7071$.

Comparison with basis function $N_{5,2}(\xi)$ shown in blue.

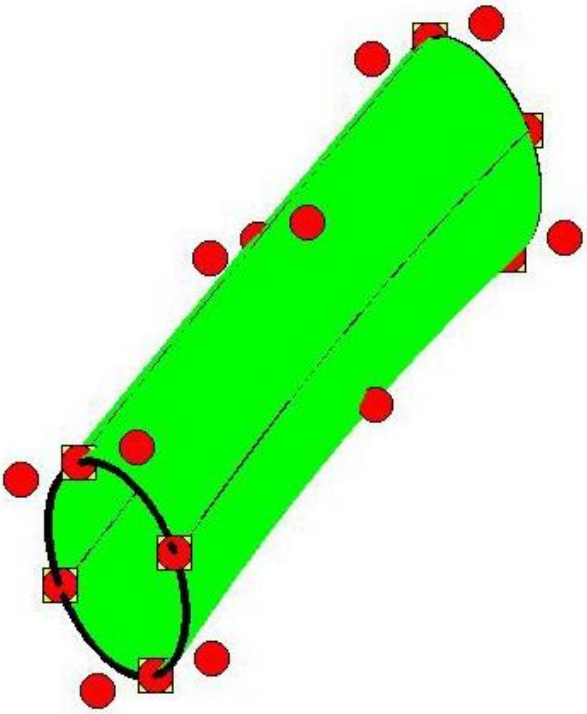


Figure 2.43. NURBS Surface created from consecutive circle cross-sections.

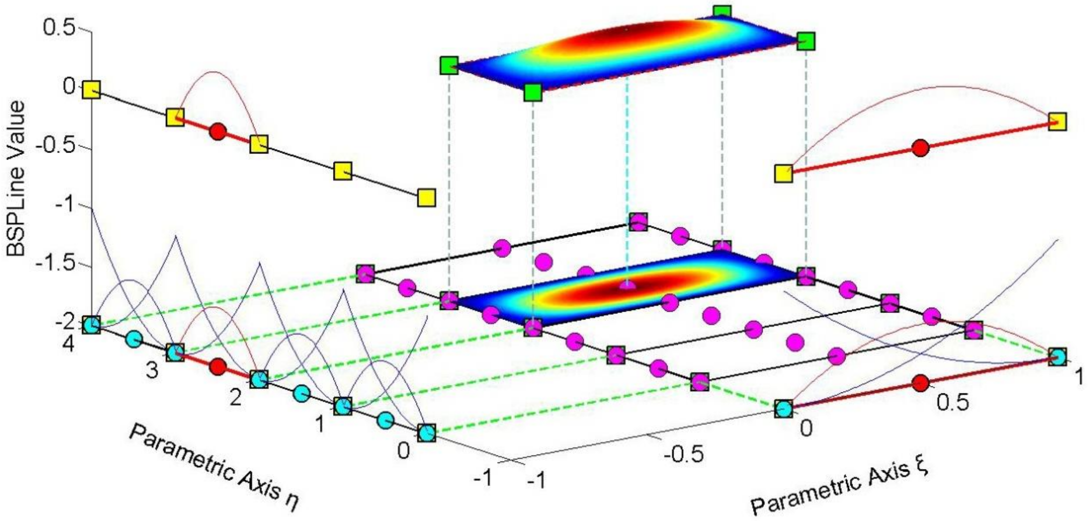
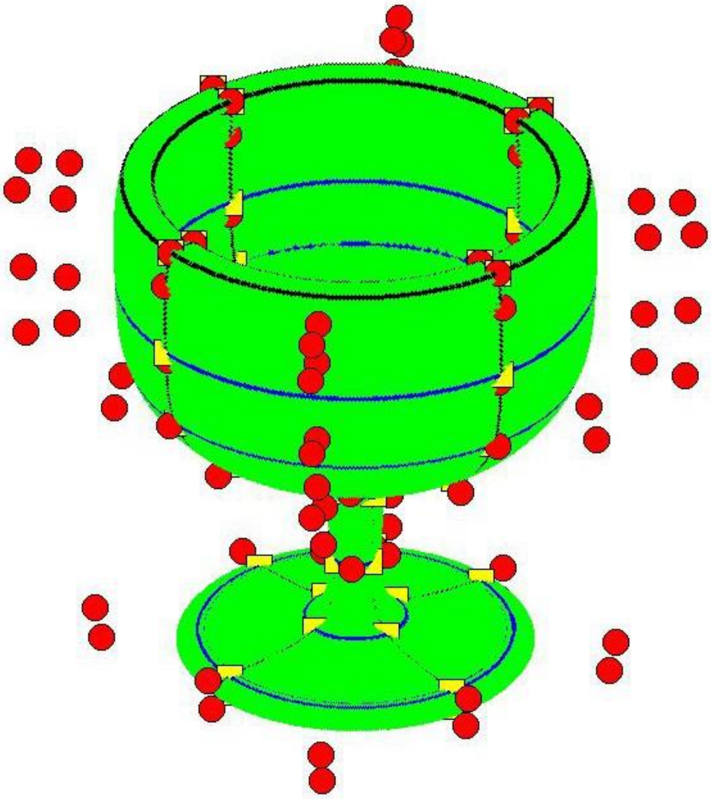


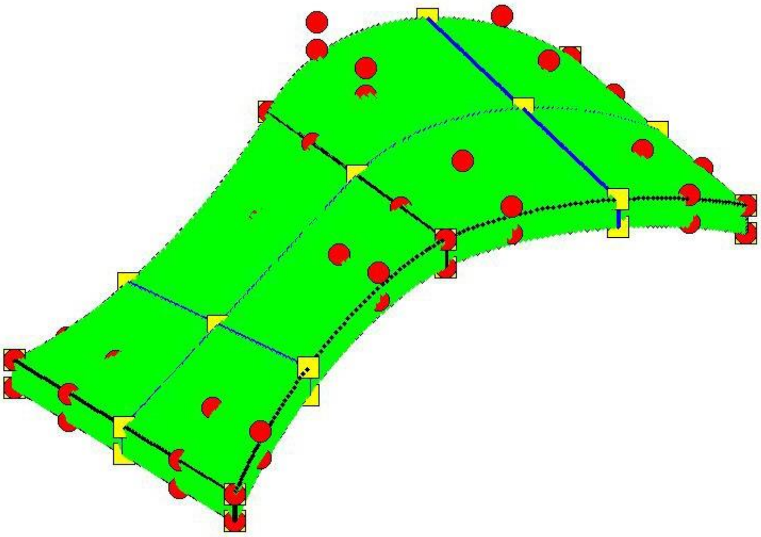
Figure 2.44. Shape Function $R_{2,6}^{2,2}(\xi,\eta)$ for the surface of Figure 2.43.

The corresponding Weight is $\frac{\sqrt{2}}{2} = 0.7071$.

The representation and main attributes are exactly the same as those of B-SPLine Shape Functions.



(a)



(b)

Figure 2.45. NURBS Solids.
(a) Wine Glass and **(b)** Abstract NURBS Solid

The glass of wine displayed in Figure 2.45.a is a 3D NURBS solid. The potential of Isogeometric Analysis is clearly represented in this model. Observe the exact representation of conic sections and smooth surfaces, in combination with immediate mesh generation. The mesh, that is depicted in the picture, can be instantly used for analysis. An abstract form of another NURBS solid is represented in Figure 2.45.b.

2.5 Patches

NURBS entities are created by transforming a simple Parametric shape (line, rectangle, cube) to a model in Physical Space (curve, surface, solid). They are used for the exact and efficient representation of complex geometrical structures. Sometimes, the mapping of a single Parametric shape is not the optimum solution. A designer might need two or three parametric cubes in order to efficiently represent solids with major changes in geometrical attributes. As displayed in Figure 2.46, each of these cubes, mapped to a portion of the solid in Physical Space, is a NURBS Patch. As expected, each Patch has continuity C^{p-m} on interior Knots and C^{-1} on the edge.

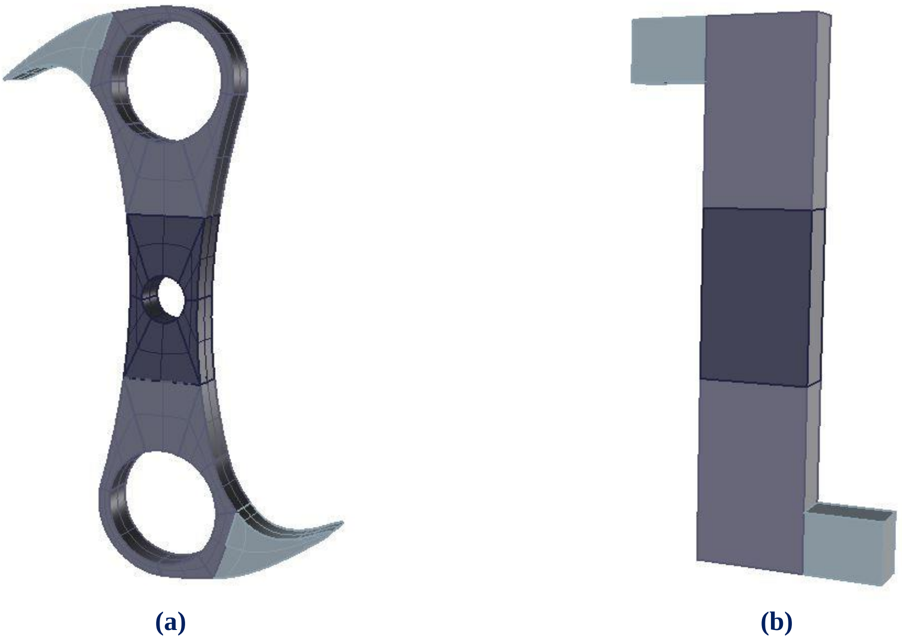


Figure 2.46. (a) Geometrical Representation of the famous Falkirk Wheel "abutment". Five separate patches are used.
(b) Each Patch portrays a cube in Parameter Space mapped as a complex shape in Physical Space.

Interconnection between Patches can be roughly achieved by choosing coincident Control Points on the edges. Still, Patch connection rarely is leak-proof. This is one of the major disadvantages of NURBS, downsized and eliminated in the next version of SPLines (T-SPLines etc.).

Sometimes, Patches exist for other reasons. For example, a major change in material properties, as displayed in Figure 2.47, requires a Patch boundary. Interpolation through a certain Control Point calls for Patch boundary to be established there. Even application of C^0 Continuity, for analysis purposes, is enabled by introduction of a Patch. If the same polynomial order is used, the mapping can be unified. In these special occasions, the separate Parameter Spaces of the Patches can be united in one Parameter Space, using one set of basis functions and one Knot Value Vector. The distinction between Patches can be applied by enforcing C^0 Continuity across the boundary. In the examples used in this book, the latter option is preferred, when possible.

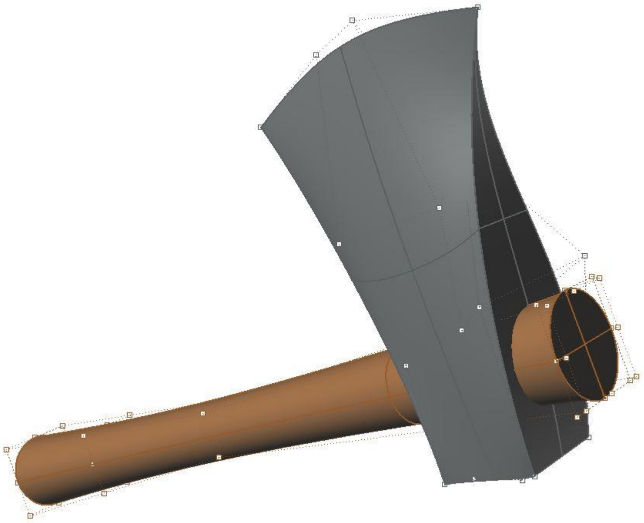


Figure 2.47. NURBS Patches enforced in order to distinct timber from steel. Each material requires separate Patch Stiffness Matrix evaluation, before Global Stiffness Matrix creation.

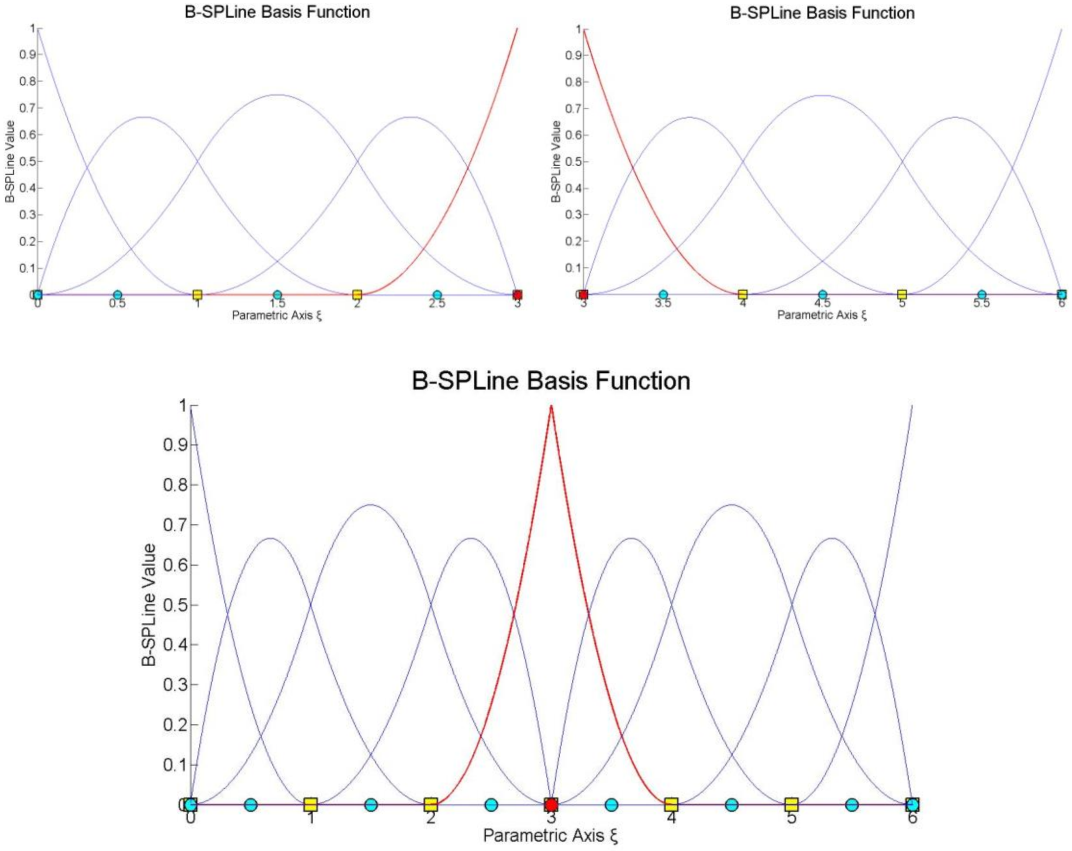


Figure 2.48. Separate Knot Vectors united into one. The Control Points at the boundary are merged. C^0 Continuity is applied.

In the geometrical models presented in this thesis, Knot boundaries are drawn in blue and Patch boundaries in black. This separates C^{-1} and C^0 Continuity from C^1 and greater Continuity. The importance of Continuity in Analysis is examined in the following Chapters.

3 Refinement

3.1 Introduction

NURBS geometries lie at the forefront of designing technology as they provide numerous possibilities for representation, while supplying a reliable basis which can be easily adjusted for the purpose of analysis. Thus, it is commonplace to use a more complex mesh to obtain better analysis results. The transition from a coarse mesh to a fine mesh preserves the geometrical features of the model, while providing the designer with the ability to modify specific parts of the entity. In order to use Refinement efficiently in analysis, the engineer has to understand the principles involved and the ways in which the basis is altered. Refinement is an automated process that requires minimal manual effort, but complicated nonetheless.

The combination of a limited number of Control Points with a small polynomial order is the optimum solution for a designer. This coarse mesh provides exact geometrical representation with reduced computational cost. It provides a certain level of understanding and control for the user, who can comprehend the simplified patterns of the basis and Control Net. The creation of Stiffness Matrix is also less time-consuming, due to the limited number of degrees of freedom and interconnections involved.

The coarse mesh, however, has flexibility issues. This means that each Control Point affects a rather large part of the model, so that small changes to Control Point variables reflect to significant changes in geometry. Moreover, basis function overlapping is reduced in coarse meshes. This makes a coarse mesh unsuitable for analysis. A feasible approach is the introduction of a coarse mesh for design, and afterwards Refinement of this mesh for analysis purposes.

With the creation of a finer mesh, a detailed, flexible Control Net is introduced. Each Control Point affects a smaller portion of the model and the number of interconnections is usually increased. Mapping from Parameter to Physical Space remains unchanged; this is very important in terms of analysis. Stiffness Matrix calculation is more time-consuming, but accuracy per degree of freedom is also improved.

There are three ways in which a B-SPLine basis can be enriched. A different Knot Value Vector may be selected, the polynomial order may be increased or a combination of both may occur; the raise of the polynomial order followed by the introduction of a richer Knot Vector. These techniques are referred to as h-, p- and k-Refinement respectively.

3.2 Knot Value Insertion

Knot Value insertion is the introduction of a finer mesh by enrichment of the existing Knot Value Vector Ξ. A new Knot Value Vector $\bar{\Xi}$ is created, such that $\Xi \subset \bar{\Xi}$. Only internal Knot Values can be added; the boundaries have to remain intact. A new set of B-SPLine basis functions is created. The coarse mesh representation is evaluated, as usual, by:

$$C(\xi) = \sum_{j=1}^{n} \left\{ N_{j,p}(\xi) \cdot X_j \right\} = \underset{1\times n}{\left\{ N(\xi) \right\}^T} \cdot \underset{n\times 1}{\left\{ X \right\}}$$

whereas the new representation

$$C(\xi) = \sum_{i=1}^{m} \left\{ \bar{N}_{i,p}(\xi) \cdot \bar{X}_i \right\} = \underset{(1\times m)}{\left\{ \bar{N}(\xi) \right\}^T} \cdot \underset{(m\times 1)}{\left\{ \bar{X} \right\}}$$

Therefore, the new set of Control Points $\left\{ \bar{X} \right\}$ has to be defined. According to Hughes [1], this can be achieved with the evaluation of a Transformation matrix, so that:

$$\underset{(mx1)}{\left\{ \bar{X} \right\}} = \underset{(m\times n)}{\left[T^p \right]} \cdot \underset{(n\times 1)}{\left\{ X \right\}}$$

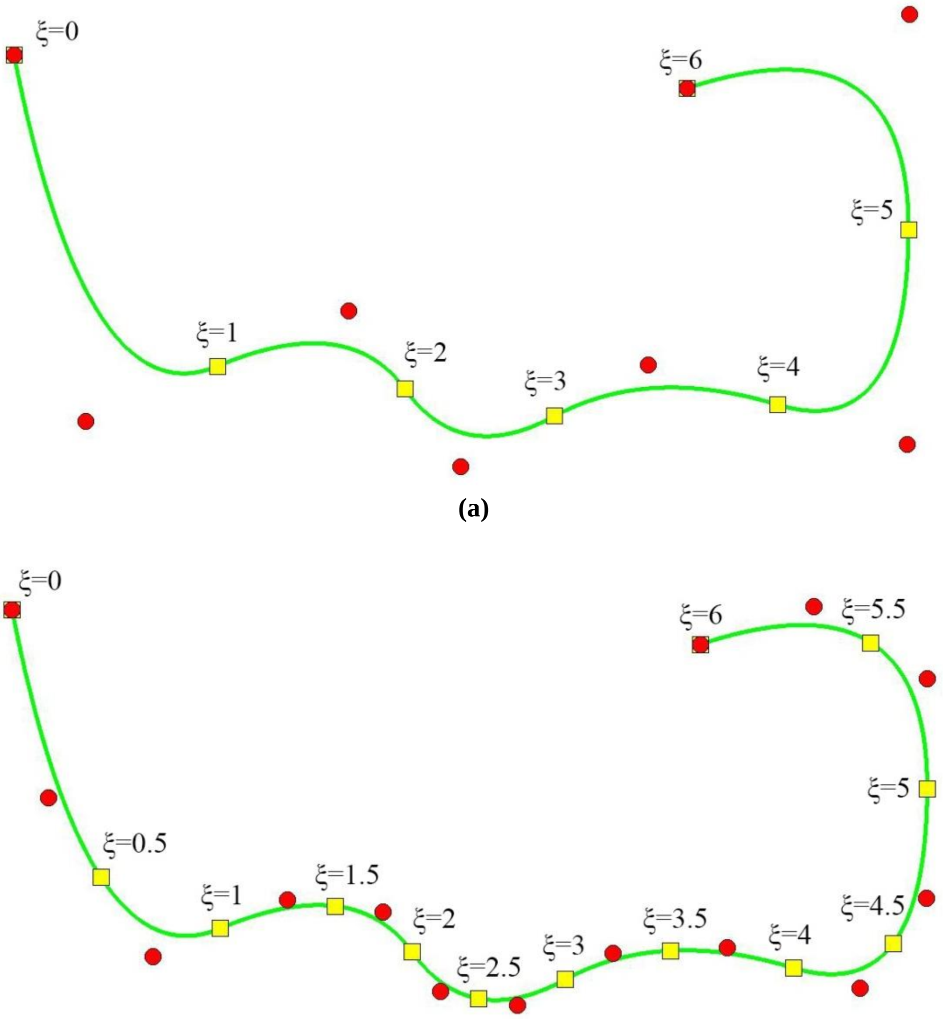
This matrix is formed recursively:

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

$$T_{i,j}^{q} = \frac{\bar{\xi}_{i+q} - \xi_j}{\xi_{j+q} - \xi} \cdot T_{i,j}^{q-1} + \frac{\xi_{j+q+1} - \bar{\xi}_{i+q}}{\xi_{j+q+1} - \xi_{j+1}} \cdot T_{i,j+1}^{q-1}, \text{ for } q = 1, 2, \ldots, p$$

This technique is called h-Refinement.

The B-SPLine curve represented in Figure 3.1.a is refined by Knot Value Insertion in Figure 3.1.b. Both geometry and parametric mapping remain intact. This can be confirmed by the fact that already existing Knots have not been moved. For each new Knot Value, a basis function and a corresponding Control Point have been created. The support is still $p+1$ Knot Value Spans, but their size has been reduced by the introduction of the new Knot Value Vector. Therefore, each Control Point now affects a much smaller part of the curve. This can be seen in Figure 2. The limited area of effect for each Control Point leads to a more accurate approximation to the curve.



(a)

(b)

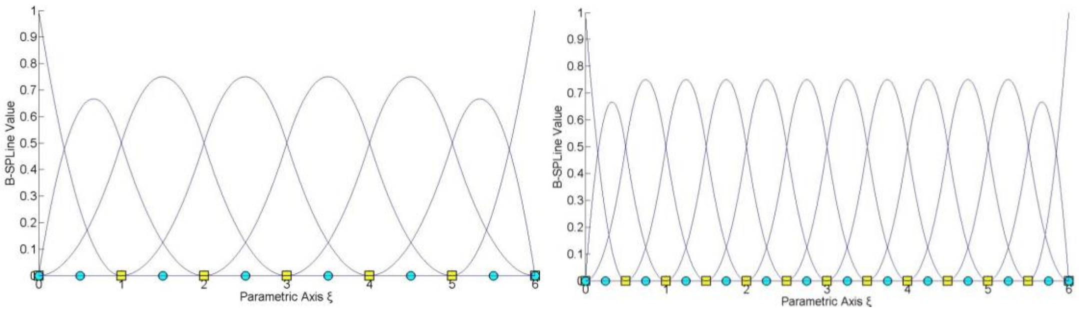
Figure 3.1. h-Refinement applied on a B-SPLine curve.

(a) Coarse Mesh. Knot Value Vector:

$$\Xi = \{0 \quad 0 \quad 0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 6 \quad 6\}$$

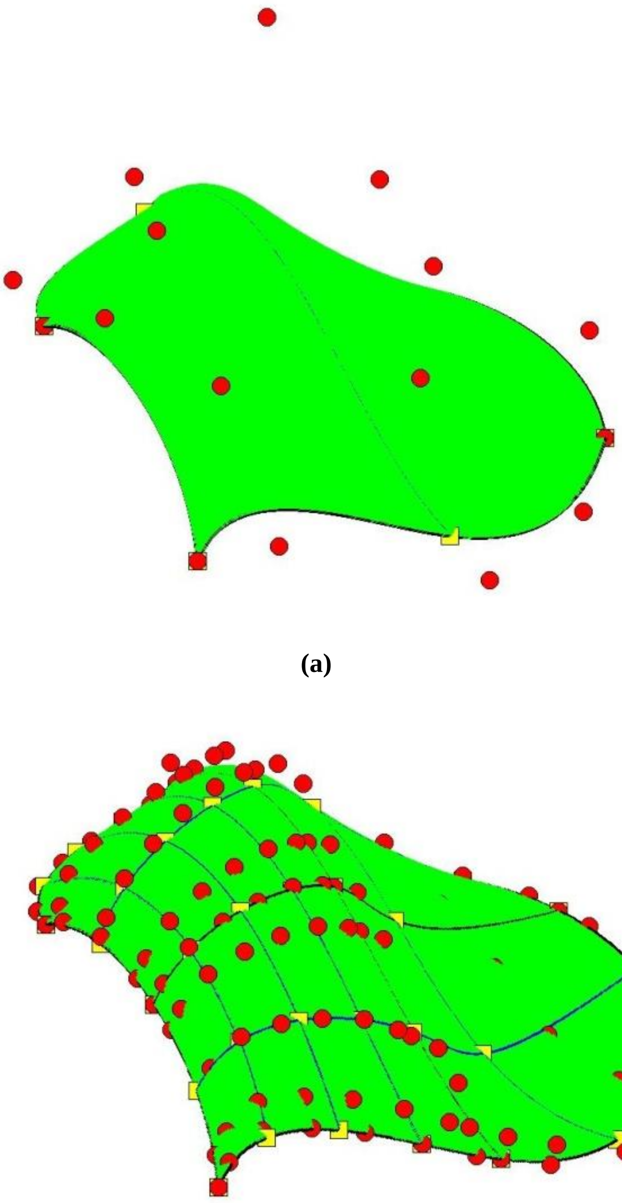
(b) Fine Mesh. Knot Value Vector:

$$\bar{\Xi} = \{0 \quad 0 \quad 0 \quad 0.5 \quad 1 \quad 1.5 \quad 2 \quad 2.5 \quad 3 \quad 3.5 \quad 4 \quad 4.5 \quad 5 \quad 5.5 \quad 6 \quad 6 \quad 6\}$$



(a) **(b)**

Figure 3.2: Basis Functions (a) before and (b) after h-Refinement.



(a)

(b)

Figure 3.3. Knot Insertion in multiple directions.

(a) Coarse Mesh. Knot Value Vectors:

$$\Xi = \{0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 2 \quad 2 \quad 2 \quad 2\}$$

$$H = \{0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1 \quad 1\}$$

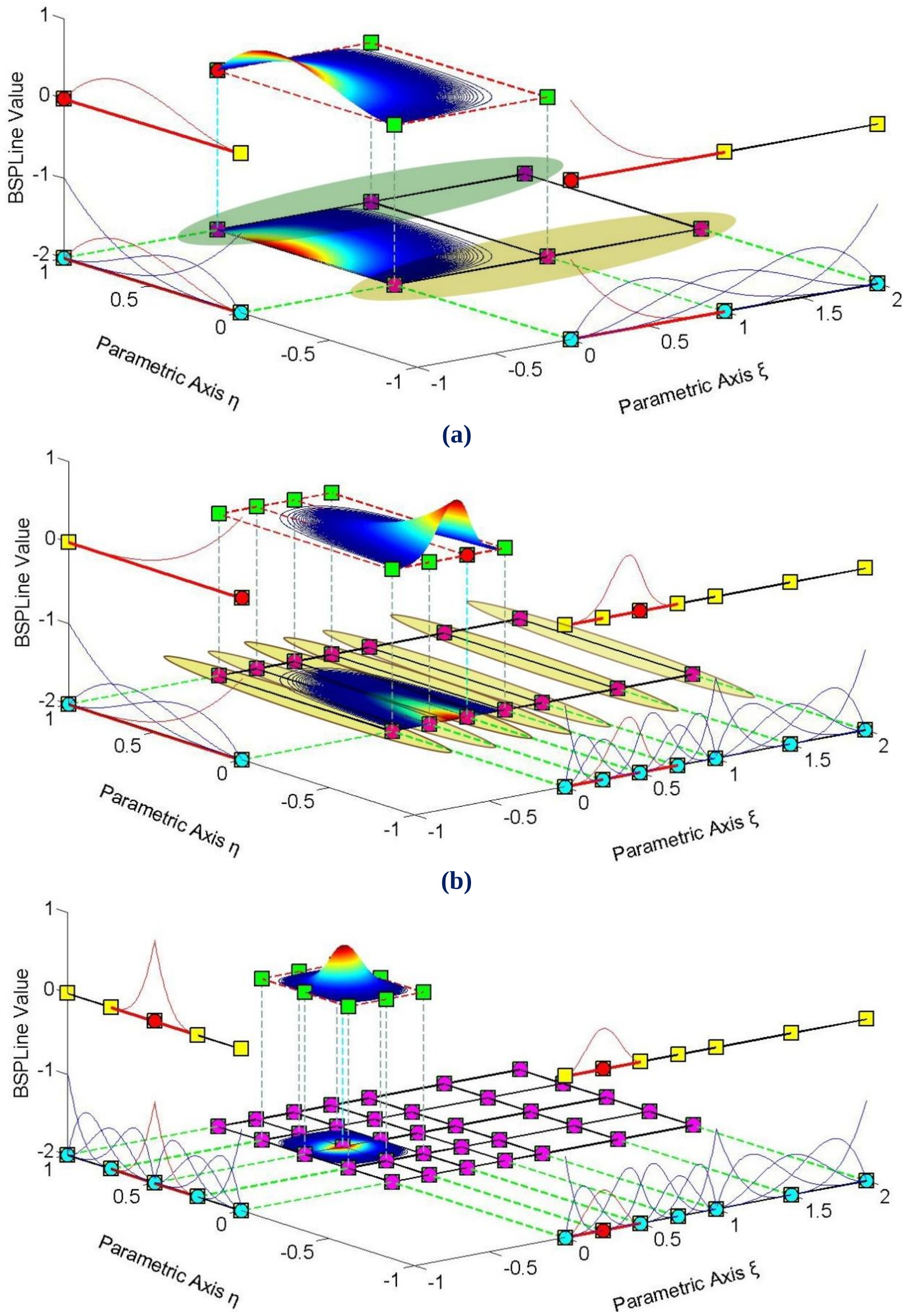
(b) Fine Mesh. Knot Value Vectors:

$$\Xi = \{0 \quad 0 \quad 0 \quad 0 \quad 0.25 \quad 0.5 \quad 0.5 \quad 0.75 \quad 1 \quad 1 \quad 1 \quad 1.5 \quad 2 \quad 2 \quad 2 \quad 2\}$$

$$H = \{0 \quad 0 \quad 0 \quad 0 \quad 0.25 \quad 0.5 \quad 0.5 \quad 0.5 \quad 0.75 \quad 1 \quad 1 \quad 1 \quad 1\}$$

Refinement is applicable in multi-directional problems as well. The surface in Figure 3.3.a is refined both per ξ and η. Figure 3.4 represents basis functions before and after Refinement.

Note that for every univariate (1D) Control Point per η, a whole set of Control Points per ξ are defined. Each set is refined individually, as shown in Figures 3.4.a and 3.4.b. After the new Control Points per ξ are introduced, the same process is followed for Refinement per η. The order in which Refinement is applied is of no importance; due to tensor product properties, Control Points per η could have been refined first, but the result would be the same.



(a)

(b)

(c)

Figure 3.4. Basis and Shape Functions for Surface Coarse and Fine Mesh

(a) Coarse Mesh. **(b)** Refinement per ξ direction.
(c) Refinement per η direction. Fine Mesh.

3.3 Degree Elevation

Instead of adding Knot Values to an existing Knot Value Vector, the increment of the polynomial order can also enrich the basis. Apart from geometry, the mapping from parameter to physical space must also remain unchanged. This is achieved by keeping the same continuity per Knot both for coarse and fine mesh.

In order to increase the degree of a B-SPLine curve from p to $\overline{p}$, the new Knot Value Vector has to be defined first. No new Knots are added, but every existing Knot's multiplicity is increased by $\overline{p}-p$ times. The coarse mesh representation is defined as:

$$C(\xi)=\sum_{j=1}^{n}\left\{N_{j,p}(\xi)\cdot X_j\right\}=\underset{(1\times n)}{\left\{N(\xi)\right\}^T}\cdot\underset{(n\times 1)}{\left\{X\right\}}$$

whereas the fine mesh representation as:

$$C(\xi)=\sum_{i=1}^{m}\left\{\overline{N}_{i,p}(\xi)\cdot\overline{X}_i\right\}=\underset{(1\times m)}{\left\{\overline{N}(\xi)\right\}^T}\cdot\underset{(m\times 1)}{\left\{\overline{X}\right\}}$$

This leads to:

$$\underset{(1\times m)}{\left\{\overline{N}(\xi)\right\}^T}\cdot\underset{(m\times 1)}{\left\{\overline{X}\right\}}=\underset{(1\times n)}{\left\{N(\xi)\right\}^T}\cdot\underset{(n\times 1)}{\left\{X\right\}}$$

The new set of Control Points $\left\{\overline{X}\right\}$, necessary for the representation, will be defined through a Transformation matrix for p-Refinement. There are many efficient algorithms for degree elevation. The following is a quick, reliable approach we have been efficiently using:

At first, coarse mesh B-SPLine basis functions are evaluated at m points throughout the patch. Careful selection (such as no points of C^0 Continuity or other irregularities are involved) is preferred. This way, a B-SPLine function matrix is created, $\underset{(n\times m)}{\left[N\right]}$, which contains the values of the n basis functions for the m selected points.

The same points are used for the evaluation of new basis functions, with the refined degree and the new Knot Value Vector. Thus, $\underset{(m\times m)}{\left[\overline{N}\right]}$ is created. Since the same points are evaluated and the parametric mapping remains the same, it applies that:

$$\underset{(m\times m)}{\left[\overline{N}\right]^T}\cdot\underset{(n\times 1)}{\left\{\overline{X}\right\}}=\underset{(m\times n)}{\left[N\right]^T}\cdot\underset{(n\times 1)}{\left\{X\right\}}\Rightarrow\underset{(n\times 1)}{\left\{\overline{X}\right\}}=\left(\underset{(m\times m)}{\left[\overline{N}\right]^T}\right)^{-1}\cdot\underset{(m\times n)}{\left[N\right]^T}\cdot\underset{(n\times 1)}{\left\{X\right\}}$$

Therefore,

$$\underset{(m\times n)}{[T]} = \left(\underset{(m\times m)}{\left[\overline{N}\right]}^{T}\right)^{-1} \cdot \underset{(m\times n)}{[N]}^{T}$$

If $\underset{(m\times m)}{\left[\overline{N}\right]}^{T}$ cannot be reversed, a different set of points have to be selected. This procedure can also be used for h-Refinement applications.

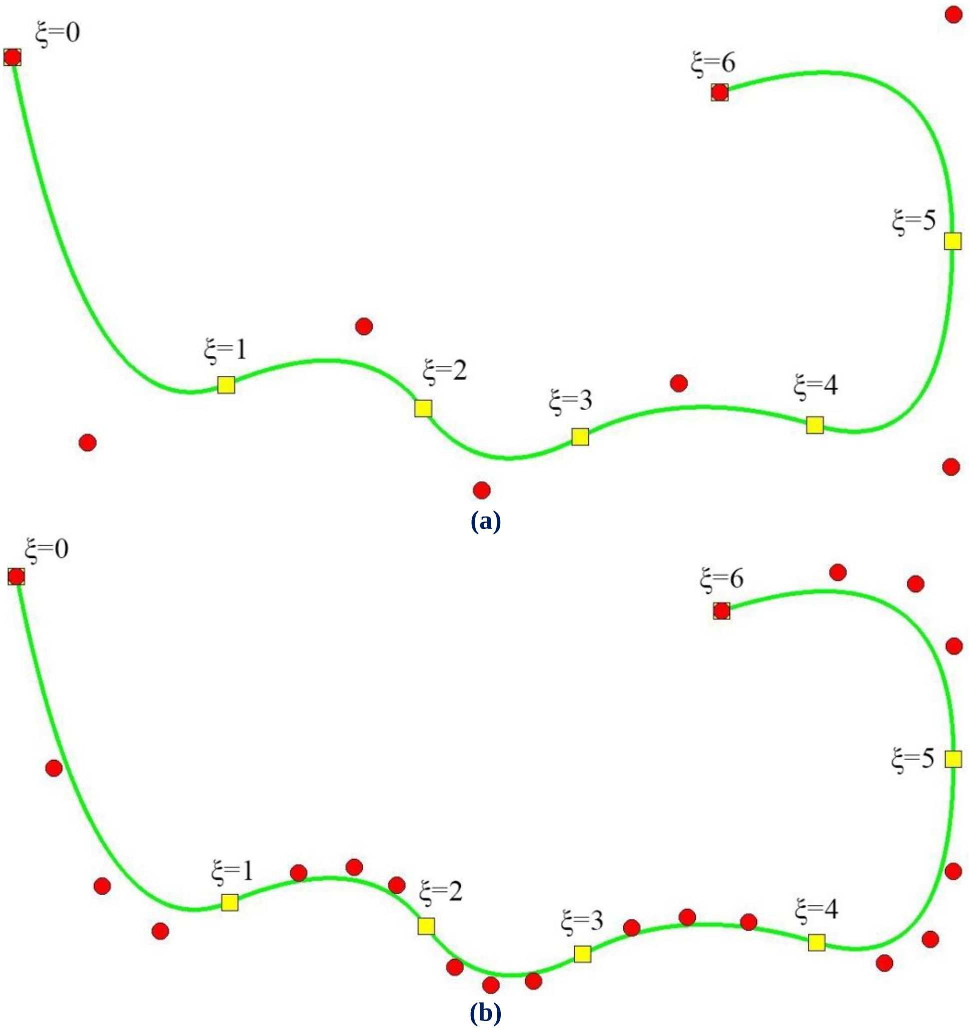


Figure 3.5. p-Refinement on a B-SPLine curve, from $p = 2$ to $\bar{p} = 4$.

(a) Coarse Mesh. Knot Value Vector:
$\Xi = \{0 \quad 0 \quad 0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 6 \quad 6\}$

(b) Fine Mesh. Knot Value Vector:
$\Xi = \{0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1 \quad 2 \quad 2 \quad 2 \quad . \quad . \quad . \quad 6 \quad 6 \quad 6 \quad 6 \quad 6\}$

The curve in Figure 3.5 is subjected to p-Refinement. Note that Knot Spans are unchanged. Geometry and mapping from Parameter Space remain intact in p-Refinement as well. Continuity remains C^1 across every internal Knot. Curve approximation by Control Points is also improved with p-Refinement.

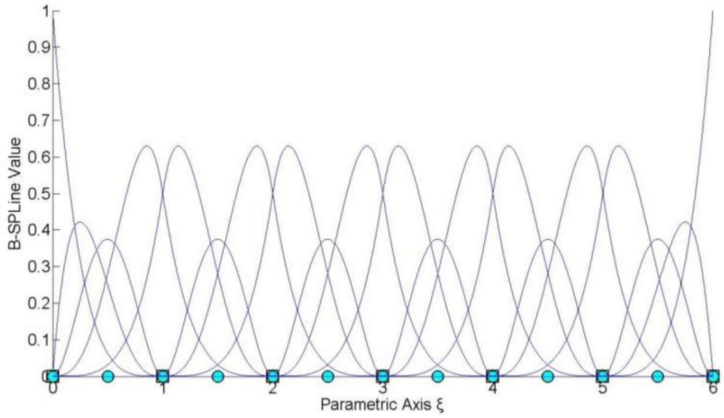
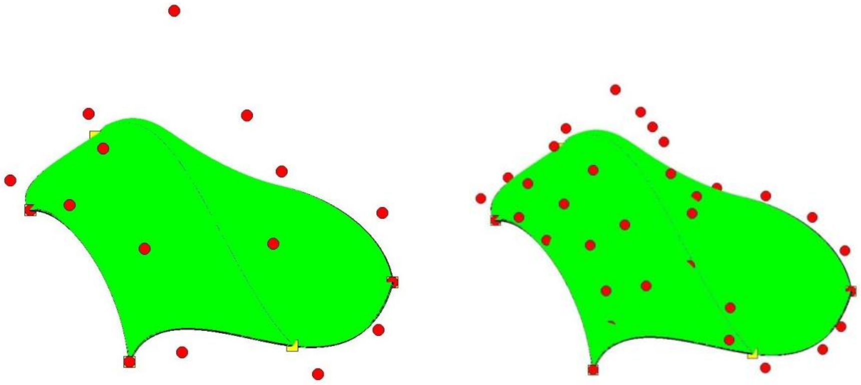


Figure 3.6. Basis Functions for p-Refinement of Figure 3.5. C^1 continuity across internal Knots.

Tensor product properties prove that p-Refinement can be utilized in multiple directions.



(a) **(b)**

Figure 3.7. Order Elevation in multiple directions.

(a) Coarse Mesh. Polynomial Degree: $(p,q)=(3,3)$.

(b) Fine Mesh. Polynomial Degree: $(p,q)=(4,5)$.

Observe that Knot Rectangles have not changed in this example as well. The mesh is finer, but the continuity of the basis and the whole Parameter Space remain the same. Basis function interconnectivity, however, is improved.

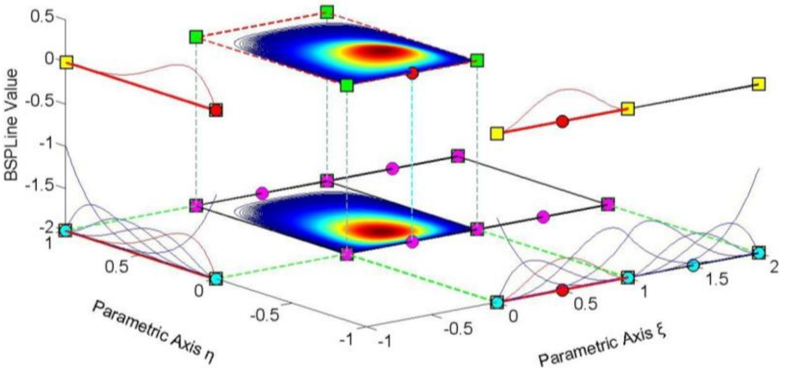
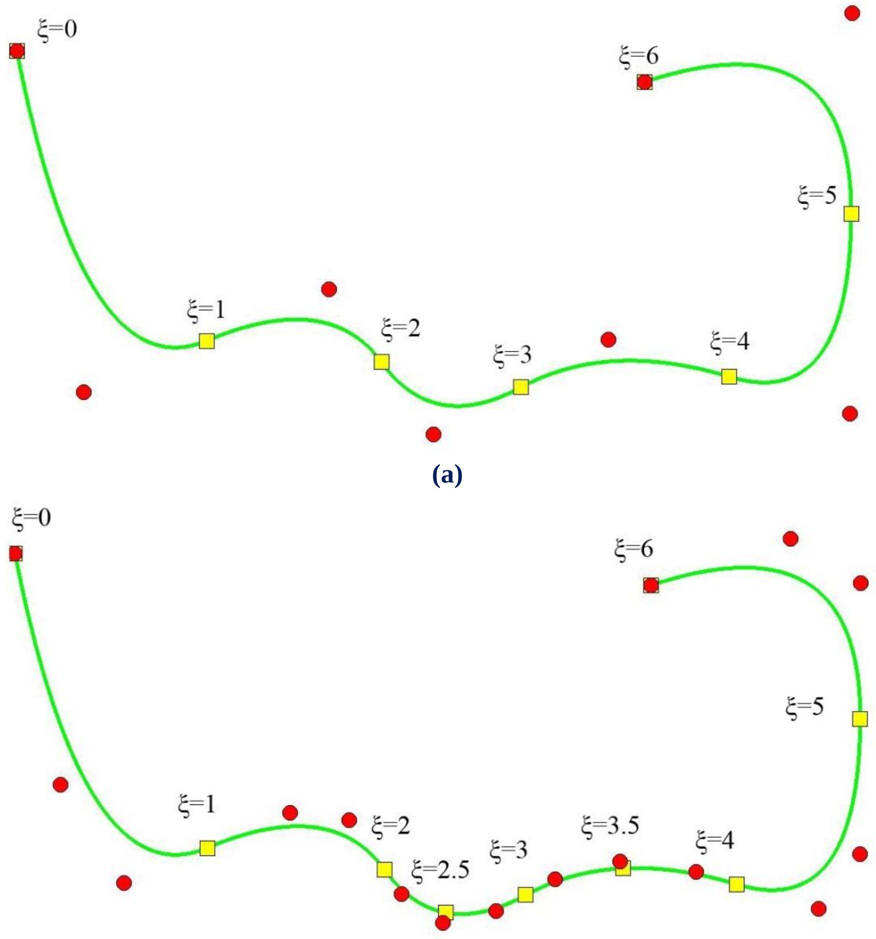


Figure 3.8. Basis and Shape Functions for p-Refinement. Reduced continuity across internal Knots.

3.4 Degree Elevation and Knot Insertion

Increasing polynomial degree by p-Refinement is an improvement to the basis, but continuity remains the same as in the coarse mesh. In order to improve this aspect, k-Refinement was introduced by Hughes. The basic idea is that, after p-Refinement, h-Refinement can be applied in order to create basis functions of C^{p-1} Continuity. This is a powerful tool that can lead to greater convergence rates for our models.



(a)

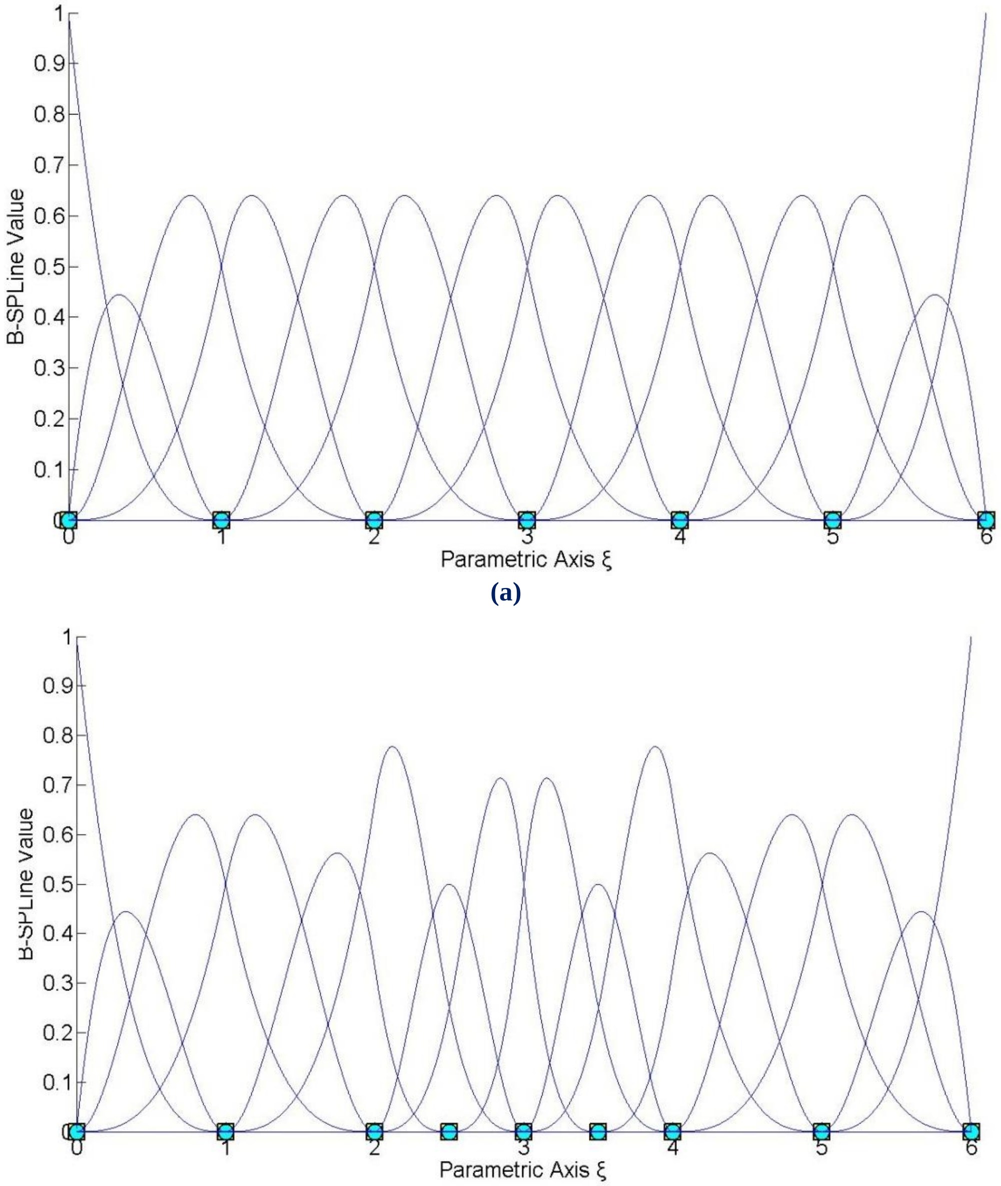
(b)

Figure 3.9. k-Refinement on a B-SPLine curve

(a) Coarse mesh.
(b) Fine mesh. Polynomial order $p = 3$. Knot Value Vector:

$$\Xi = \{0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 2 \quad 2 \quad 2.5 \quad 3 \quad 3 \quad 3.5 \quad 4 \quad 4 \quad 5 \quad 5 \quad 6 \quad 6 \quad 6 \quad 6\}$$

In Figure 3.9, p-Refinement is performed first, so the order is elevated to $\bar{p} = 3$. The C^1 continuity that existed in the coarse mesh is maintained. Afterwards, h-Refinement is applied, with the insertion of two extra Knot Values. Continuity across this Knot Values is C^2, thus taking greater advantage of the new polynomial order. The combination of p- and h- Refinement brings their best features together. This is why k-Refinement is considered so effective.



(a)

(b)

Figure 3.10. B-SPLine basis functions for the two stages of k-Refinement.

(a) p-Refinement to $p = 3$. Knot Value Vector:

$$\Xi = \{0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 2 \quad 2 \quad 3 \quad 3 \quad 4 \quad 4 \quad 5 \quad 5 \quad 6 \quad 6 \quad 6 \quad 6\}$$

(b) h-Refinement. Knot Value Vector:

$$\Xi = \{0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 2 \quad 2 \quad 2.5 \quad 3 \quad 3 \quad 3.5 \quad 4 \quad 4 \quad 5 \quad 5 \quad 6 \quad 6 \quad 6 \quad 6\}$$

After p-Refinement, internal Knots still possess C^1 continuity. Note that only two basis functions are non-zero across every Knot. After h-Refinement, two new C^2 continuity Knots are introduced. Three B-SPLine basis functions are non-zero across these Knots, $\xi = 2.5$ and $\xi = 3.5$. This way, greater levels of continuity for the new polynomial order are utilized.

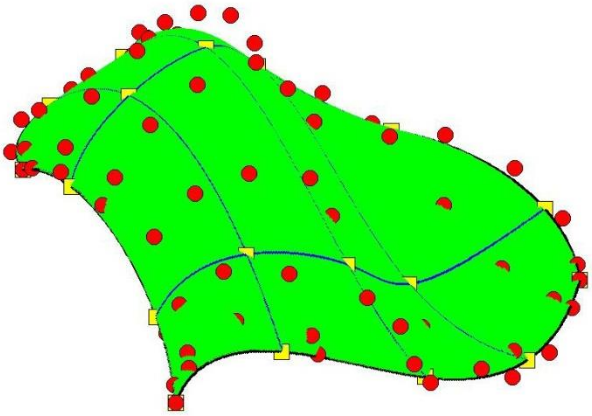


Figure 3.11. k-Refinement for a NURBS Surface.

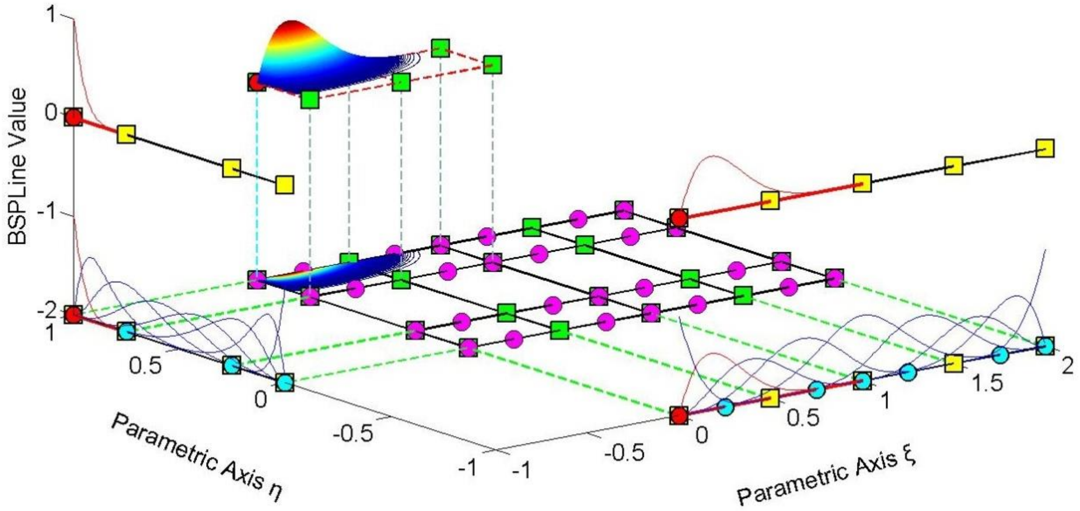


Figure 3.12. Basis functions for Surface k-Refinement.

Knot Value Vector (axis ξ)

$$\Xi = \{0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0.5 \quad 1 \quad 1 \quad 1.5 \quad 2 \quad 2 \quad 2 \quad 2 \quad 2\}$$

Knot Value Vector (axis η)

$$H = \{0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0.25 \quad 0.75 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1\}$$

In Figure3.12, the application of k-Refinement for a Surface is represented. In fact, h-Refinement is performed after the p-Refinement of Figure 3.7. When p-Refinement is applied, continuity of the internal Knots is still C^2. After h-Refinement, continuity of the additional Control Points in axes ξ, η is increased to C^3 and C^4 respectively. This augmentation of the continuity has proven to be very helpful for the analysis.

3.5 Reverse Refinement

Transformation from a fine to a coarse mesh is also required for some applications. The ability to go back-and-forth between coarse and fine mesh is useful in certain aspects of the analysis and design. This process is Reverse Refinement.

Reverse Refinement does not always provide an accurate solution. Removal of certain Knots or downgrading of polynomial order may affect the shape of the entity. It is obvious, for example, that reverse p-Refinement is not possible if there are Knots of C^{p-1} continuity involved. In general, reverse Refinement is an over determined problem and can only be solved in a certain number of cases. It is generally applicable when Refinement has already been implemented and the engineer wants, for some reason, to return to the initial state of the model. The procedure we generally follow is similar to Refinement.

In order to calculate the coarse mesh Control Points, a Transformation Matrix from Coarse to Fine mesh has to be introduced, so that:

$$\underset{(n\times 1)}{\{X\}} = \underset{(n\times m)}{[T_{CF}]} \cdot \underset{(mx1)}{\{\overline{X}\}}$$

This can be created from the Transformation Matrix from Fine to Coarse Mesh, $\underset{(m\times n)}{[T_{FC}]}$.

$$\underset{(mx1)}{\{\overline{X}\}} = \underset{(m\times n)}{[T_{FC}]} \cdot \underset{(n\times 1)}{\{X\}} \Rightarrow \underset{(n\times m)}{[T_{FC}]^T} \cdot \underset{(mx1)}{\{\overline{X}\}} = \underset{(n\times m)}{[T_{FC}]^T} \cdot \underset{(m\times n)}{[T_{FC}]} \cdot \underset{(n\times 1)}{\{X\}}$$

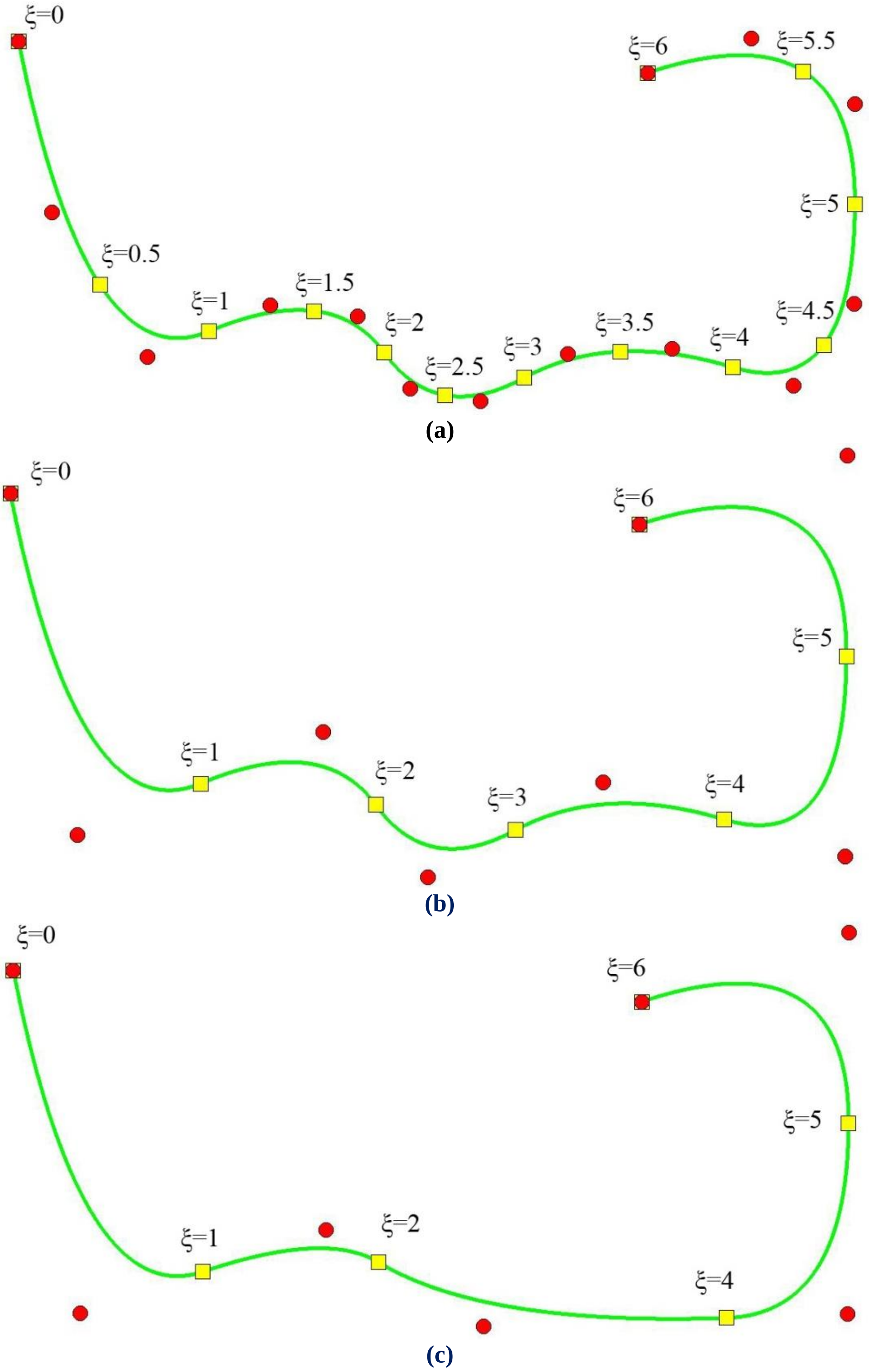
Therefore,

$$\underset{(n\times 1)}{\{X\}} = \left(\underset{(n\times m)}{[T_{FC}]^T} \cdot \underset{(m\times n)}{[T_{FC}]} \right)^{-1} \cdot \underset{(n\times m)}{[T_{FC}]^T} \cdot \underset{(mx1)}{\{\overline{X}\}}$$

and

$$\underset{(n\times m)}{[T_{CF}]} = \left(\underset{(n\times m)}{[T_{FC}]^T} \cdot \underset{(m\times n)}{[T_{FC}]} \right)^{-1} \cdot \underset{(n\times m)}{[T_{FC}]^T}$$

As in Figure 3.13, Reverse Refinement works when going back-and-forth between coarse and fine mesh, but it is not always applicable. The curve in Figure 3.13.c has failed to accurately represent the required geometry. Refinement reverse should be carefully used as a tool of intercommunication between initial and refined model.



(a)

(b)

(c)

Figure 3.13. Reverse h-Refinement. Success and Failure of Reverse Refinement.

(a) Fine Mesh. Knot Value Vector:

$$\Xi = \{0 \quad 0 \quad 0 \quad 0.5 \quad 1 \quad 1.5 \quad 2 \quad 2.5 \quad 3 \quad 3.5 \quad 4 \quad 4.5 \quad 5 \quad 5.5 \quad 6 \quad 6 \quad 6\}$$

(b) Reverse Refined Coarse Mesh. Knot Value Vector:

$$\Xi = \{0 \quad 0 \quad 0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 6 \quad 6\}$$

(c) Reverse-Refined Failed Mesh. Knot Value Vector:

$$\Xi = \{0 \quad 0 \quad 0 \quad 1 \quad 2 \quad 4 \quad 5 \quad 6 \quad 6 \quad 6\}$$

3.6 Refinement with NURBS

Everything mentioned about Refinement is applicable to B-SPLine entities. The question is what happens with NURBS. Different weight values and the complexity of NURBS shape functions prevent the Refinement straight at the NURBS entity. However, every NURBS is created from the projection of a B-SPLine; therefore, NURBS Refinement can be achieved by refining the corresponding projective B-SPLine curve.

The first step is to evaluate the Projective Control Points $\left\{B^{W}\right\}$, by multiplying every coordinate with the corresponding weight, as shown below:

$$\left\{X_i^w \quad Y_i^w \quad Z_i^w\right\} = \left\{X_i \cdot W_i \quad Y_i \cdot W_i \quad Z_i \cdot W_i\right\}$$

As mentioned in 2.4.1 (Chapter 2), Weights are the fourth-coordinate of the projective curve.

Refinement is applied for each set of the four coordinates, $\left\{X^w \quad Y^w \quad Z^w \quad W\right\}$. Afterwards, updated NURBS Control Point Coordinates and Weights are obtained by dividing the Cartesian coordinates with the weights:

$$\left\{X_i \quad Y_i \quad Z_i\right\} = \left\{\frac{X_i^w}{W_i} \quad \frac{Y_i^w}{W_i} \quad \frac{Z_i^w}{W_i}\right\}$$

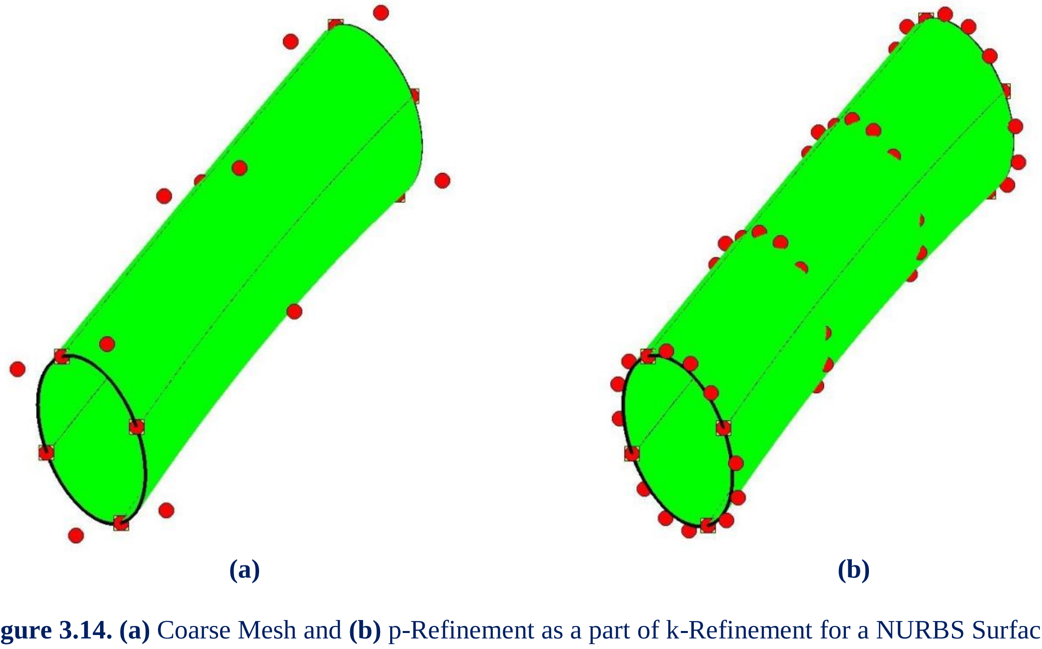
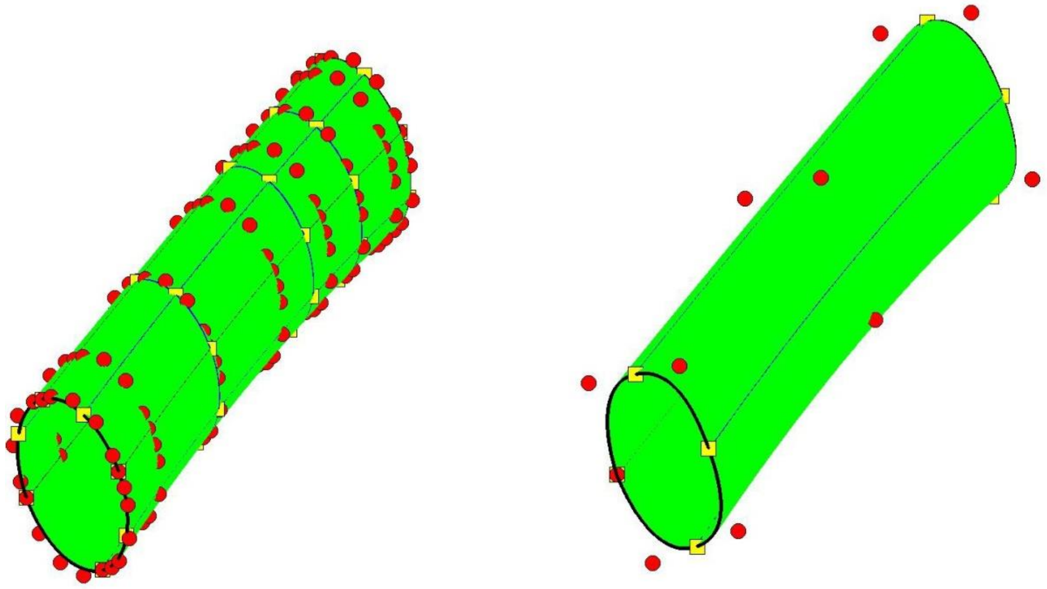


Figure 3.14. (a) Coarse Mesh and **(b)** p-Refinement as a part of k-Refinement for a NURBS Surface.



(a)

(b)

Figure 3.15. (a) k-Refinement and **(b)** failed Reverse Refinement applied to the NURBS Surface.

4 Stiffness Matrix Formulation

4.1 Preliminary Steps for Analysis

The main difference between Isogeometric Analysis and Finite Element Analysis is the type of the shape functions used for the approximation of the solution field. The substitution of Lagrange polynomials with NURBS (widely utilized in the CAD community) leads to several special characteristics of Isogeometric Analysis.

4.1.1 Shape Functions

Finite Element Method Shape Functions are usually polynomials (e.g. Lagrange polynomials). However, they hold certain disadvantages. FEM Shape functions are interpolatory at all nodes, internal and external, and have C^{-1} Continuity at the edge. This results in the incompetence of FEM to define stresses and strains at any boundary of any element, leading in the introduction of other corrective methods such as extrapolation, in order to achieve that.

NURBS as Shape functions hold an overlapping that is useful, as nearby elements are strongly connected, thus the simulation provides an improved approach to the natural problem. Continuous Shape function derivatives lead to a continuous stress and strain field, minimizing the need for application of corrective methods.

4.1.2 Control Points

Classical FEM downsizes the natural problem of infinite unknowns to a finite number of unknowns, which are the degrees of freedom of the nodes. The position of the nodes depends on the element type. As a general rule, the nodes can be usually found in the corners and middle of the sides of the elements. They are part of the element and therefore part of the physical model. Displacements in other areas of the model can be approximated by a linear combination of displacements on the degrees of freedom. Distribution in the model is evaluated via the corresponding Shape functions. In isoparametric elements, Shape functions and their respective nodes are also used to approximate the geometry, thus enabling relatively complex shapes to be approximated with Lagrange polynomials.

In Isogeometric Analysis, NURBS are chosen as Shape functions. The isoparametric concept is reversed, as geometrical mapping now defines the solution approximation. The geometrical representation is achieved through a combination of Control Points and their corresponding Shape functions. Degrees of freedom at the Control Points are now the unknowns.

4.1.3 Elements

Two ingredients of the Isogeometric universe correspond to the essence of FEM “element”. The Isogeometric Element could be either the Patch or the Knot Span of the Patch. In order to perform exact numerical integration [4], a certain number of Gauss points are chosen for the domain of every piece of the polynomial basis functions. The domain of this piece is the knot span, which resembles the finite element of FEM. This is the reason why in this thesis knot spans and not patches are considered isogeometric elements. In IGA, shape functions are not restricted to the interior of each element (Knot Span). Instead, they are non-zero across $p+1$ Knot Spans and overlap with more Shape functions.

This overlapping results in a denser Stiffness Matrix than the classical Finite Element Matrix with the same degrees of freedom. Apart from that, the fact that B-SPLine functions are defined in the whole domain allows for integration throughout the Patch without building local element matrices separately. Of course, this would be time-consuming and it is not advised for advanced software technologies, but serves well for research purposes, where a flexible quadrature code is needed in order to test and discover new methods and ideas.

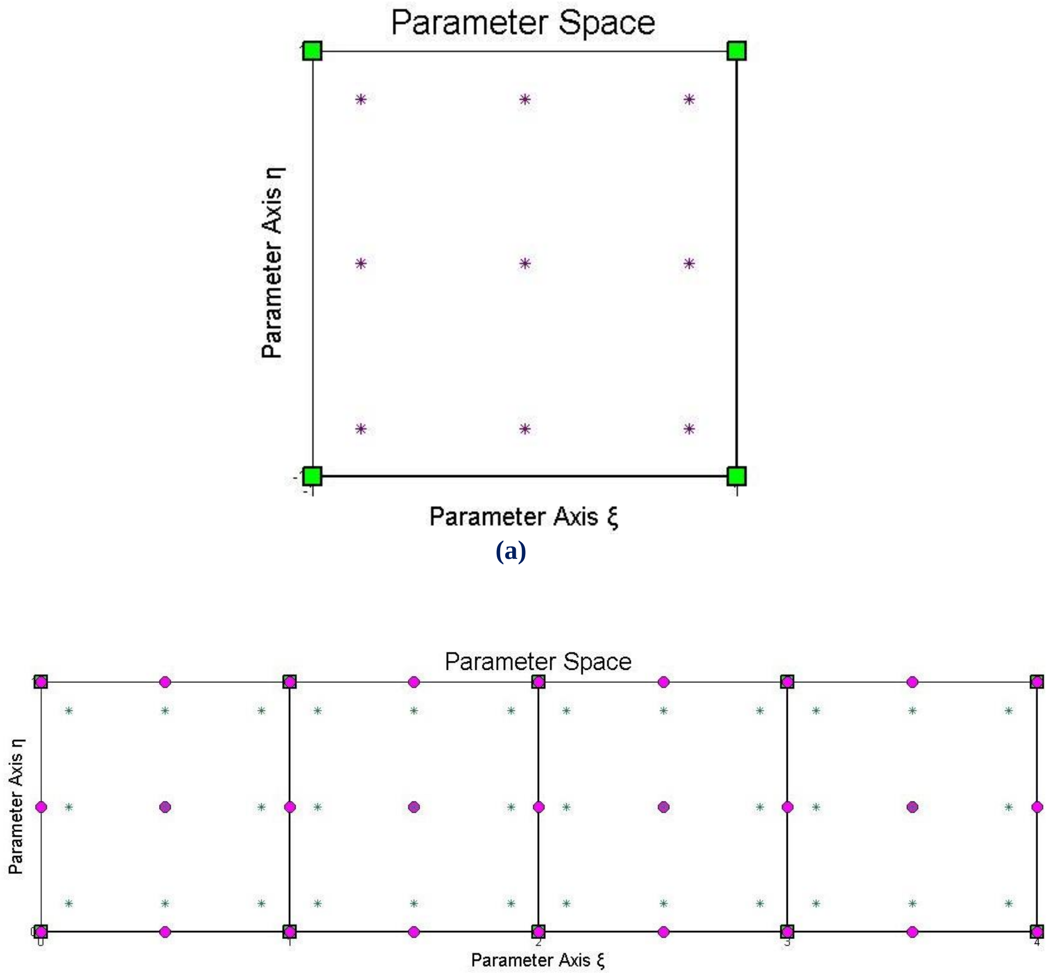
4.1.4 Gauss Points

4.1.4.1 Parametric Coordinates

As mentioned above, Gauss Points are chosen for each Knot Span. Their coordinates are obtained on a reference element spanning [-1,1] as the roots of the Legendre Polynomial. The next step is to transform the coordinates and weights from the reference Knot Span ξ^R to the desired Knot Span $[\xi_i, \xi_{i+1})$.

$$\xi = \frac{(\xi_{i+1} - \xi_i) \cdot \xi^R + (\xi_{i+1} + \xi_i)}{2}$$

$$w^{GP\xi} = \frac{(\xi_{i+1} - \xi_i)}{2} \cdot w_\xi^R$$



(a)

(b)

Figure 4.1. Gauss Points
(a) on Reference Knot Span and **(b)** transferred to Parameter Space.

Full tensor product properties apply here as well, leading in similar equations for the other two parametric directions:

$$\eta = \frac{\left(\eta_{j+1} - \eta_j\right) \cdot \eta^R + \left(\eta_{j+1} + \eta_j\right)}{2}$$

$$\zeta = \frac{\left(\zeta_{k+1} - \zeta_k\right) \cdot \zeta^R + \left(\zeta_{k+1} + \zeta_k\right)}{2}$$

$$w^{GP\eta} = \frac{\left(\eta_{j+1} - \eta_j\right)}{2} \cdot w_\eta^R$$

$$w^{GP\zeta} = \frac{\left(\zeta_{k+1} - \zeta_k\right)}{2} \cdot w_\zeta^R$$

4.1.4.2 Gauss Point Number

Gauss Point Coordinates and Weights are evaluated for every Patch. According to [4], for the exact integration of a polynomial of degree q, $\frac{q+1}{2}$ or $\frac{q+2}{2}$ Gauss Points are required per Knot Span for q odd and even respectively.

For 1D problems, the maximum degree of the Deformation Matrix is defined from the derivation of piecewise polynomial Shape functions, therefore $p-1$.

$\left[B(\xi)\right]^T \cdot [E] \cdot \left[B(\xi)\right]$ yields to the product of polynomials of maximum order $p-1$ resulting in a polynomial of maximum order $(p-1)+(p-1)=2p-2$. Thus, the minimum number of Gauss Points per Knot Span required for exact integration is:

$$\frac{(2p-2)+2}{2}=p$$

For 2D and 3D problems, the maximum order of the Deformation Matrix is determined by partial derivation of piecewise polynomial Shape functions. Therefore the maximum degree is p for derivation in the remaining directions.

The order of the product $\left[B(\xi)\right]^T \cdot [E] \cdot \left[B(\xi)\right]$ is $p+p=2p$. In order to achieve exact integration, the minimum number of Gauss Points per Knot Span is:

$$\frac{2p+2}{2}=p+1$$

Conclusively per Knot Span:

- For 1D problems, $\mathbf{p}$ Gauss Points per knot span are required.
- For 2D and 3D problems, $\mathbf{p+1}$ Gauss Points per knot span are required.

4.1.5 Patches

As mentioned before, Patches are used when a change in geometry type or material occurs. Patches can also be used in any case C^{-1} or C^0 Continuity is required. If separate Knot Vectors are used for every Patch, Stiffness Matrices will be produced for every Patch. The separate Matrices are combined into one via connectivity arrays. If the connection is watertight, the procedure is the same as the one used in classical FEM to combine local element matrices to a single, global Stiffness Matrix.

4.1.6 Elasticity Matrix

Elasticity Matrix types vary depending on the stress and strain field for each case. Data is obtained straight from classical FEM applications. Elasticity matrices for 1D elasticity, plane strain, plane stress and 3D elasticity are presented as follows, where E is the Young's modulus and ν is the Poisson's ratio.

1D Elasticity:

$$\underset{(1x1)}{[E]} = E$$

2D Elasticity, Plane Stress:

$$\underset{(3x3)}{[E]} = \frac{E}{1-\nu^2} \cdot \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix}$$

2D Elasticity, Plane Strain:

$$\underset{(3x3)}{[E]} = \frac{E}{(1-\nu)\cdot(1-2\nu)} \cdot \begin{bmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix}$$

3D Elasticity:

$$\underset{(6x6)}{[E]} = \frac{E}{(1-\nu)\cdot(1-2\nu)} \cdot \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix}$$

The corresponding stress and strain vectors are:

1D Elasticity:

$$\underset{(1\times1)}{\{\sigma\}} = \{\sigma_X\}$$

$$\underset{(1\times1)}{\{\varepsilon\}} = \{\varepsilon_X\}$$

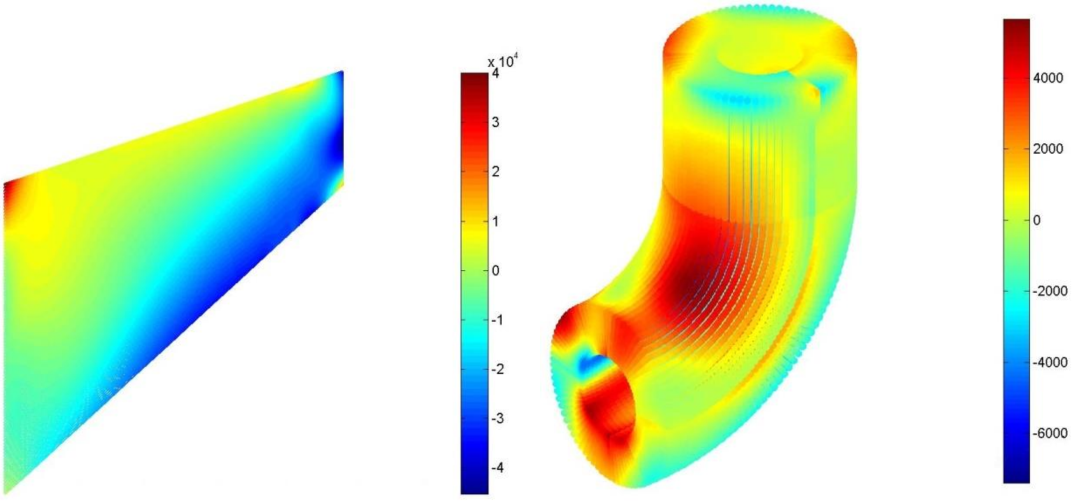
2D Elasticity, Plane Stress and Plane Strain:

$$\underset{(3\times1)}{\{\sigma\}} = \begin{Bmatrix} \sigma_X \\ \sigma_Y \\ \tau_{XY} \end{Bmatrix}$$

$$\underset{(3\times1)}{\{\varepsilon\}} = \begin{Bmatrix} \varepsilon_X \\ \varepsilon_Y \\ \gamma_{XY} \end{Bmatrix}$$

3D Elasticity:

$$\underset{(6\times1)}{\{\sigma\}} = \begin{bmatrix} \sigma_X \\ \sigma_Y \\ \sigma_Z \\ \sigma_{XY} \\ \sigma_{YZ} \\ \sigma_{ZX} \end{bmatrix}, \quad \underset{(6\times1)}{\{\varepsilon\}} = \begin{bmatrix} \varepsilon_X \\ \varepsilon_Y \\ \varepsilon_Z \\ \gamma_{XY} \\ \gamma_{YZ} \\ \gamma_{ZX} \end{bmatrix}$$



(a) **(b)**

Figure 4.2. Stress contour distribution.
(a) σ_{xx} for Plane Stress, **(b)** τ_{yz} 3D Elasticity

4.2 Stiffness Matrix Assembly.

4.2.1 General Procedure

The general process for the Global Stiffness Matrix assembly, as obtained from Finite Element Method, is shown in the following flow chart:

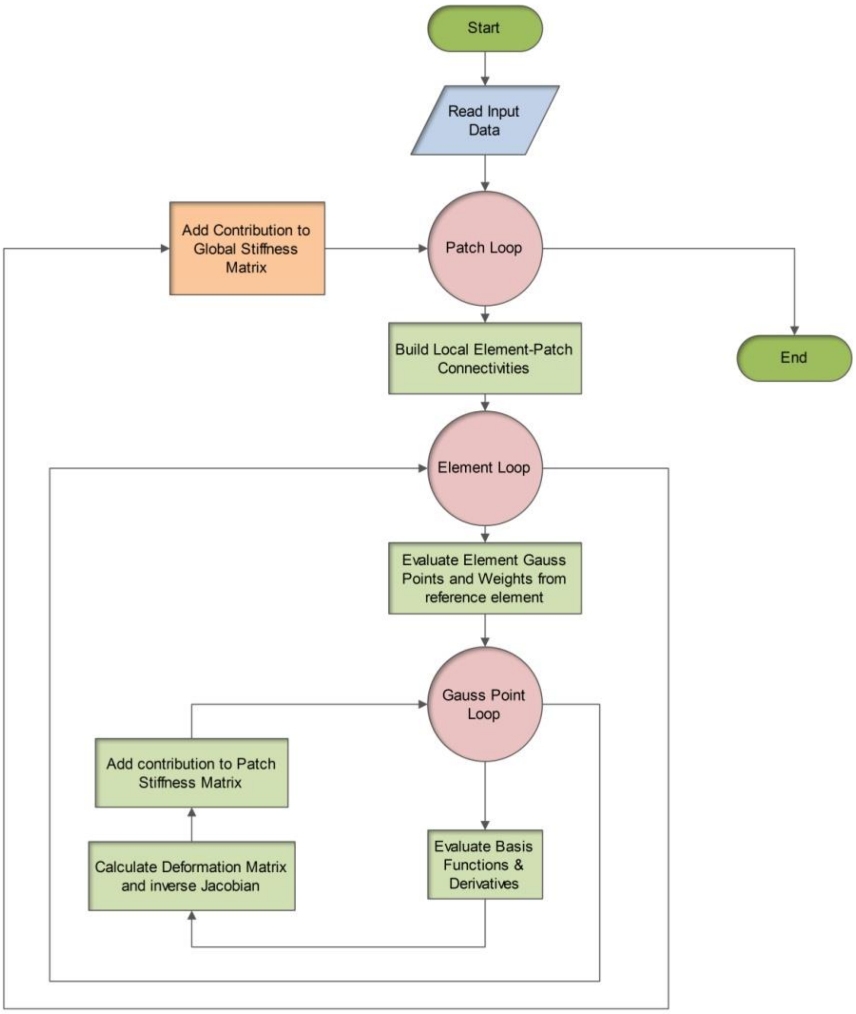

Start
Read Input Data
Add Contribution to Global Stiffness Matrix
Patch Loop
End
Build Local Element-Patch Connectivities
Element Loop
Evaluate Element Gauss Points and Weights from reference element
Gauss Point Loop
Add contribution to Patch Stiffness Matrix
Calculate Deformation Matrix and inverse Jacobian
Evaluate Basis Functions & Derivatives

Figure 4.3. Stiffness Matrix Assembly in Finite Element Method.

There are three loops:

- Patch loop
- Element loop
- Gauss Point loop

It is worth mentioning that the element loop in IGA can be avoided. In this case, the stiffness contribution of each Control Point pair is added directly to the Stiffness Matrix of the Patch. The reason for this is that Parameter Space is local to Patch rather than elements.

Therefore, an engineer accustomed to the methods of Isogeometric Analysis knows that the element loop can be completely avoided, leading to the following flow chart:

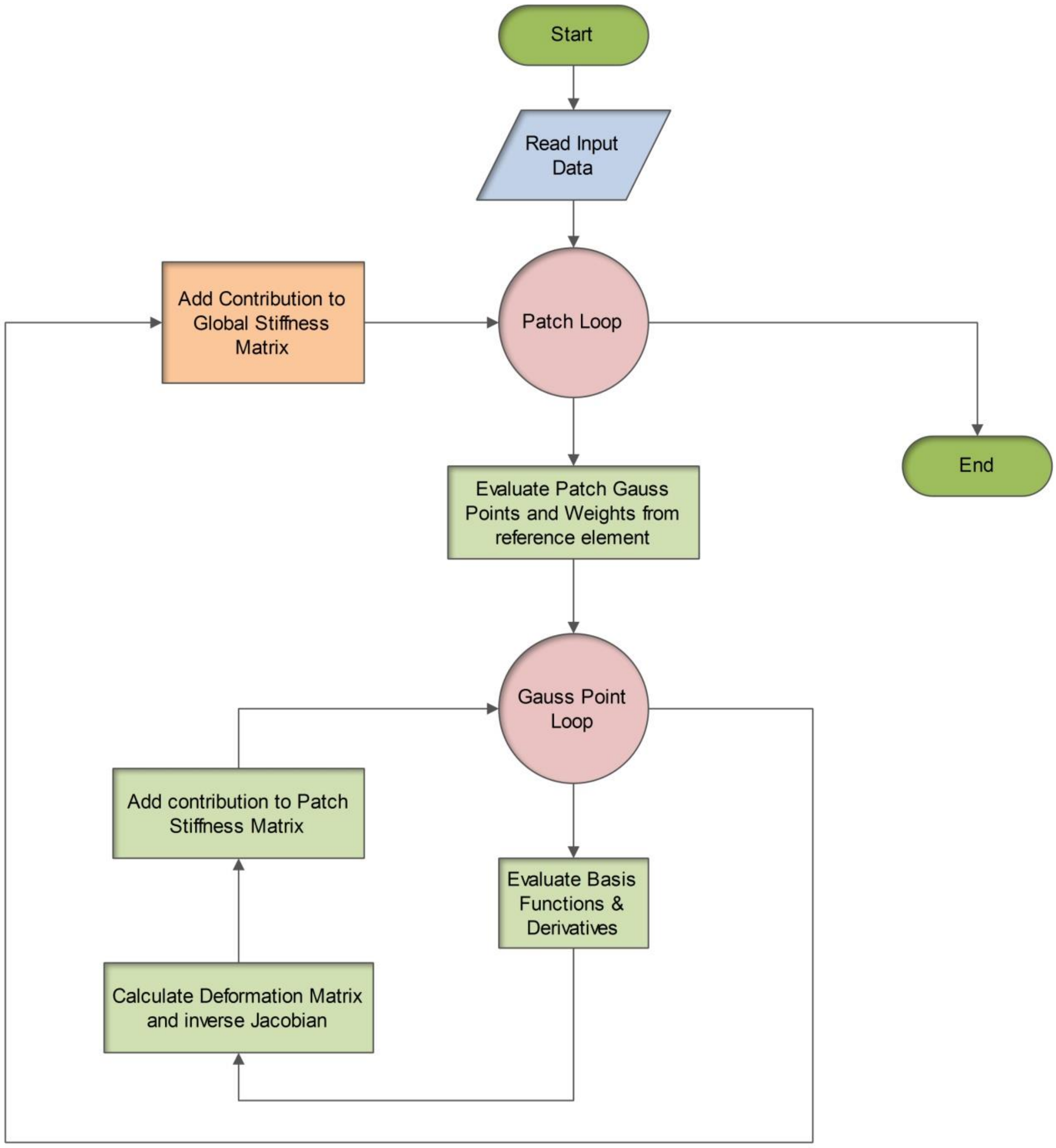

Start
Read Input Data
Add Contribution to Global Stiffness Matrix
Patch Loop
End
Evaluate Patch Gauss Points and Weights from reference element
Gauss Point Loop
Add contribution to Patch Stiffness Matrix
Evaluate Basis Functions & Derivatives
Calculate Deformation Matrix and inverse Jacobian

Figure 4.4. Stiffness Matrix Assembly in Isogeometric Analysis

4.2.1.1 Input Data

Information necessary for the whole process of analysis is given as input at this point. The information essential for the formulation of the Stiffness Matrix is divided into two categories:

- Structural Analysis and Material
- Computational Geometry

4.3 Stiffness Matrix

4.3.1 Stiffness Matrix 1D

Let us consider a simple 1D application as an example. These applications are utilized only in truss systems with axial tension, but their significance is mostly academic. Simplifying the problem, without loss of generality, can lead to a better understanding of the principles involved. In our research career, we often address 1D analogies for the solution of complex problems and it has always been extremely helpful.

In such a case, only axial deformation for each point of the truss exists. This deformation is $u(x)=u(C(\xi))=u(\xi)$. The respective strain matrix consists of one value:

$$\underset{(1\times1)}{\{\varepsilon\}}=[\varepsilon_x]=\left[\frac{\partial u}{\partial x}\right]$$

In order to calculate the derivative of u, we must first establish a transition from Physical Space to Parameter Space:

$$\frac{\partial\varphi}{\partial x}=\frac{\partial\varphi}{\partial\xi}\frac{\partial\xi}{\partial x}$$
$$\frac{\partial\varphi}{\partial\xi}=\frac{\partial\varphi}{\partial x}\frac{\partial x}{\partial\xi}$$

$$\left[\frac{\partial\varphi}{\partial\xi}\right]=[J]\cdot\left[\frac{\partial\varphi}{\partial x}\right]$$

where $[J]$ is the Jacobian Matrix enabling transition from Physical to Parameter Space and vice-versa. It can be evaluated with the help of basis functions $R_i(\xi)$ and the Control Points' Cartesian coordinates X_i as shown:

$$\underset{(1\times1)}{\left[J(\xi)\right]}=\underset{(1\times n)}{\left[R_{1,\xi}(\xi)\quad R_{2,\xi}(\xi)\quad \dots\quad \dots\quad \dots\quad R_{n,\xi}(\xi)\right]}\cdot\underset{(n\times1)}{\begin{bmatrix}X_1\\X_2\\.\\.\\.\\X_n\end{bmatrix}}$$

where $R_{i,\xi}(\xi)=\frac{\partial}{\partial\xi}R_i(\xi)$.

In Finite Element Analysis, the reverse transformation is utilized. This is why the inverse matrix $[J]^{-1}$ is needed.

$$\underset{(1x1)}{[J]^{-1}} = \frac{1}{\underset{(1\times1)}{[J]}}$$

Special care has to be taken in order for the Jacobian to be correct. The positive direction of the axes in Parameter and Physical Space must coincide, or the determinant of the Jacobian will be negative and the matrix $[J]$ irreversible. Numerical integration on points of singularity, such as two points on Parameter Space mapped into the same point on Physical Space, has to be avoided as well.

The next step is to calculate the matrices $[B_1]$ and $[B_2]$. The matrix $[B_1]$ transfers the strains of the element from Parameter to Physical Space and the matrix $[B_2]$ transfers the nodal displacements of the elements to the strains at the Parameter Space. Therefore, the matrices $[B_1]$and $[B_2]$ can be calculated from the equations below:

$$\underset{(1\times1)}{\{\varepsilon\}} = \left[\frac{\partial u}{\partial x}\right] = [J]^{-1} \cdot \left[\frac{\partial u}{\partial \xi}\right]$$

$$\underset{(1\times1)}{[B_1(\xi)]} = \left[\frac{1}{J_{11}}\right]$$

$$\underset{(1\times1)}{\{\varepsilon\}} = \underset{(1\times1)}{[B_1]} \cdot \underset{(1\times1)}{\{u_\xi\}}$$

$$\left[\frac{\partial u}{\partial \xi}\right] = \left[R_{1,\xi}(\xi) \quad R_{2,\xi}(\xi) \quad \ldots \quad \ldots \quad \ldots \quad R_{n,\xi}(\xi)\right] \cdot \begin{bmatrix} u_1 \\ u_2 \\ . \\ . \\ . \\ u_n \end{bmatrix}$$

$$\underset{(1\times n)}{[B_2(\xi)]} = \left[R_{1,\xi}(\xi) \quad R_{2,\xi}(\xi) \quad \ldots \quad \ldots \quad \ldots \quad R_{n,\xi}(\xi)\right]$$

$$\underset{(1\times1)}{\{u\}} = \underset{(1\times n)}{[B_2]} \cdot \underset{(n\times1)}{\{d\}}$$

After that, the Deformation Matrix is evaluated.

$$\underset{(1\times n)}{[B(\xi)]} = \underset{(1\times1)}{[B_1(\xi)]} \cdot \underset{(1\times n)}{[B_2(\xi)]}$$

Deformation Matrix produces strain values anywhere in the model, by utilizing nodal displacements.

$$\underset{(1\times1)}{\left\{\varepsilon(\xi)\right\}}=\underset{(1\times n)}{\left[B(\xi)\right]}\cdot\underset{(n\times1)}{\left\{d\right\}}$$

The Stiffness Matrix for the Patch is evaluated as shown:

$$\underset{(n\times n)}{\left[K\right]}=\int_{\xi_0}^{\xi_{n+p+1}}\underset{(n\times1)}{\left[B(\xi)\right]^T}\cdot\underset{(1x1)}{\left[E\right]}\cdot\underset{(1\times n)}{\left[B(\xi)\right]}\cdot A\cdot\det\left[J\right]\,d\xi$$

Direct integration is almost never applicable. Numerical integration is used instead, looping through all the Gauss Points of a Patch and their respective weights:

$$\underset{(n\times n)}{\left[K\right]}=\sum_{i=1}^{GP\xi}\left\{\underset{(n\times1)}{\left[B(\xi_i)\right]^T}\cdot\underset{(1x1)}{\left[E\right]}\cdot\underset{(1\times n)}{\left[B(\xi_i)\right]}\cdot A\cdot\det\left[J\right]\cdot w_i^{GP\xi}\right\}$$

where

- A : the area of the cross-section
- GP_ξ : the total number of Gauss Points for the specific Patch
- ξ_i : the coordinates of the Gauss Points
- $w_i^{GP\xi}$: the corresponding weights.

4.3.2 Stiffness Matrix 2D

2D elasticity problems have many applications in modern analysis. The logic is exactly the same as in 1D problems. The main difference is, obviously, the utilization of one more dimension. Parameter Space is defined on (ξ,η) and Physical Space on (x,y). Displacements per x, y at any point in the entire domain are defined as $u(x,y)=u(S(\xi,\eta))=u(\xi,\eta)$ and $v(x,y)=v(\xi,\eta)$ respectively.

The strain vector is defined as:

$$\underset{(3\times1)}{\left\{\varepsilon\right\}}=\begin{bmatrix}\varepsilon_x\\ \varepsilon_y\\ \gamma_{xy}\end{bmatrix}=\begin{bmatrix}\dfrac{\partial u}{\partial x}\\ \dfrac{\partial v}{\partial y}\\ \dfrac{\partial u}{\partial y}+\dfrac{\partial v}{\partial x}\end{bmatrix}\Rightarrow\underset{(3\times1)}{\left\{\varepsilon\right\}}=\begin{bmatrix}\dfrac{\partial}{\partial x} & 0\\ 0 & \dfrac{\partial}{\partial y}\\ \dfrac{\partial}{\partial y} & \dfrac{\partial}{\partial x}\end{bmatrix}\cdot\begin{bmatrix}u\\ v\end{bmatrix}$$

The transformation of a function φ between Parameter and Physical Space yields:

$$\frac{\partial\varphi}{\partial x}=\frac{\partial\varphi}{\partial\xi}\frac{\partial\xi}{\partial x}+\frac{\partial\varphi}{\partial\eta}\frac{\partial\eta}{\partial x}$$

$$\frac{\partial\varphi}{\partial y}=\frac{\partial\varphi}{\partial\xi}\frac{\partial\xi}{\partial y}+\frac{\partial\varphi}{\partial\eta}\frac{\partial\eta}{\partial y}$$

$$\frac{\partial\varphi}{\partial\xi}=\frac{\partial\varphi}{\partial x}\frac{\partial x}{\partial\xi}+\frac{\partial\varphi}{\partial y}\frac{\partial y}{\partial\xi}$$

$$\frac{\partial\varphi}{\partial\eta}=\frac{\partial\varphi}{\partial x}\frac{\partial x}{\partial\eta}+\frac{\partial\varphi}{\partial y}\frac{\partial y}{\partial\eta}$$

Thus, the 2D Jacobian Matrix can be defined as:

$$\begin{bmatrix}\frac{\partial\varphi}{\partial\xi}\\ \frac{\partial\varphi}{\partial\eta}\end{bmatrix}=\begin{bmatrix}\frac{\partial x}{\partial\xi} & \frac{\partial y}{\partial\xi}\\ \frac{\partial x}{\partial\eta} & \frac{\partial y}{\partial\eta}\end{bmatrix}\cdot\begin{bmatrix}\frac{\partial\varphi}{\partial x}\\ \frac{\partial\varphi}{\partial y}\end{bmatrix}\Rightarrow\begin{bmatrix}\frac{\partial\varphi}{\partial\xi}\\ \frac{\partial\varphi}{\partial\eta}\end{bmatrix}=\underset{(2\times2)}{[J]}\cdot\begin{bmatrix}\frac{\partial\varphi}{\partial x}\\ \frac{\partial\varphi}{\partial y}\end{bmatrix}$$

and the inverse mapping:

$$\begin{bmatrix}\frac{\partial\varphi}{\partial x}\\ \frac{\partial\varphi}{\partial y}\end{bmatrix}=\underset{(2\times2)}{[J]}^{-1}\cdot\begin{bmatrix}\frac{\partial\varphi}{\partial\xi}\\ \frac{\partial\varphi}{\partial\eta}\end{bmatrix}$$

The Jacobian matrix can be evaluated as shown:

$$\underset{(2\times2)}{[J]}=\begin{bmatrix}R_{1,\xi}(\xi,\eta) & R_{2,\xi}(\xi,\eta) & \ldots & \ldots & \ldots & R_{N,\xi}(\xi,\eta)\\ R_{1,\eta}(\xi,\eta) & R_{2.\eta}(\xi,\eta) & \ldots & \ldots & \ldots & R_{N,\eta}(\xi,\eta)\end{bmatrix}\cdot\begin{bmatrix}X_1 & Y_1\\ X_2 & Y_2\\ . & .\\ . & .\\ . & .\\ X_N & Y_N\end{bmatrix}$$

where $N=n\cdot m$ is the total number of Control Points.

The inverse Jacobian matrix is used in Stiffness Matrix calculation:

$$\underset{(2\times2)}{[J]}^{-1}=\begin{bmatrix}J^{*}_{11} & J^{*}_{12}\\ J^{*}_{21} & J^{*}_{22}\end{bmatrix}=\frac{1}{\det[J]}\cdot\begin{bmatrix}J_{22} & -J_{12}\\ -J_{21} & J_{11}\end{bmatrix}$$

The determinant of the Jacobian matrix is also required and is equal to:

$$\det\left[J\right]=J_{11}\cdot J_{22}-J_{21}\cdot J_{12}$$

In order to calculate the Deformation Matrix for 2D problems, $\left[B_1\right]$ and $\left[B_2\right]$ have to be evaluated as usual.

To obtain matrix $\left[B_1\right]$:

$$\underset{(3\times1)}{\{\varepsilon\}}=\begin{bmatrix}\dfrac{\partial u}{\partial x}\\ \dfrac{\partial v}{\partial y}\\ \dfrac{\partial u}{\partial y}+\dfrac{\partial v}{\partial x}\end{bmatrix}=\frac{1}{\det\left[J\right]}\cdot\begin{bmatrix}J_{22} & -J_{12} & 0 & 0\\ 0 & 0 & -J_{21} & J_{11}\\ -J_{21} & J_{11} & J_{22} & -J_{12}\end{bmatrix}\cdot\begin{bmatrix}\dfrac{\partial u}{\partial \xi}\\ \dfrac{\partial u}{\partial \eta}\\ \dfrac{\partial v}{\partial \xi}\\ \dfrac{\partial v}{\partial \eta}\end{bmatrix}$$

Hence,

$$\underset{(3\times4)}{\left[B_1(\xi,\eta)\right]}=\frac{1}{\det\left[J\right]}\cdot\begin{bmatrix}J_{22} & -J_{12} & 0 & 0\\ 0 & 0 & -J_{21} & J_{11}\\ -J_{21} & J_{11} & J_{22} & -J_{12}\end{bmatrix}$$

To calculate matrix $\left[B_2\right]$:

$$\begin{bmatrix}\dfrac{\partial u}{\partial \xi}\\ \dfrac{\partial u}{\partial \eta}\\ \dfrac{\partial v}{\partial \xi}\\ \dfrac{\partial v}{\partial \eta}\end{bmatrix}=\begin{bmatrix}R_{1,\xi} & 0 & R_{2,\xi} & 0 & \dots & \dots & \dots & R_{N,\xi} & 0\\ R_{1,\eta} & 0 & R_{2,\eta} & 0 & \dots & \dots & \dots & R_{N,\eta} & 0\\ 0 & R_{1,\xi} & 0 & R_{2,\xi} & \dots & \dots & \dots & 0 & R_{N,\xi}\\ 0 & R_{1,\eta} & 0 & R_{2,\eta} & \dots & \dots & \dots & 0 & R_{N,\eta}\end{bmatrix}\cdot\begin{bmatrix}u_1\\ v_1\\ u_2\\ v_2\\ \cdot\\ \cdot\\ \cdot\\ u_N\\ v_N\end{bmatrix}$$

Hence,

$$\underset{(4\times2N)}{\left[B_2(\xi,\eta)\right]}=\begin{bmatrix}R_{1,\xi} & 0 & R_{2,\xi} & 0 & \dots & \dots & \dots & R_{N,\xi} & 0\\ R_{1,\eta} & 0 & R_{2,\eta} & 0 & \dots & \dots & \dots & R_{N,\eta} & 0\\ 0 & R_{1,\xi} & 0 & R_{2,\xi} & \dots & \dots & \dots & 0 & R_{N,\xi}\\ 0 & R_{1,\eta} & 0 & R_{2,\eta} & \dots & \dots & \dots & 0 & R_{N,\eta}\end{bmatrix}$$

Having determined $\left[B_1\right]$ and $\left[B_2\right]$, the Deformation Matrix is calculated as:

$$\underset{(3\times 2N)}{\left[B(\xi,\eta)\right]} = \underset{(3\times 4)}{\left[B_1(\xi,\eta)\right]} \cdot \underset{(4\times 2N)}{\left[B_2(\xi,\eta)\right]}$$

In order to evaluate the Stiffness Matrix, integration is required.

$$\underset{(2N\times 2N)}{\left[K\right]} = \int_{\xi_0}^{\xi_{n+p+1}} \int_{\eta_0}^{\eta_{m+q+1}} \underset{(2N\times 3)}{\left[B(\xi,\eta)\right]^T} \cdot \underset{(3x3)}{\left[E\right]} \cdot \underset{(3\times 2N)}{\left[B(\xi,\eta)\right]} \cdot t \cdot \det\left[J\right] d\eta d\xi$$

Numerical integration procedures for ξ, η lead to integration for tensor product Gauss Points.

$$\underset{(2N\times 2N)}{\left[K\right]} = \sum_{i=1}^{GP\xi} \sum_{j=1}^{GP\eta} \underset{(2N\times 3)}{\left[B\left(\xi_i,\eta_j\right)\right]^T} \cdot \underset{(3x3)}{\left[E\right]} \cdot \underset{(3\times 2N)}{\left[B\left(\xi_i,\eta_j\right)\right]} \cdot t \cdot \det\left[J\right] \cdot w_i^{GP\xi} \cdot w_j^{GP\eta}$$

where:

- t: the thickness of the cross-section
- GP_ξ: the total number of Gauss Points per ξ for the specific Patch
- GP_η: the total number of Gauss Points per η for the specific Patch
- ξ_i, η_j: the coordinates of the tensor product Gauss Point i, j
- $w_i^{GP\xi}$, $w_j^{GP\eta}$: the corresponding weights

The only difference, at this point, between plane stress and plane strain is the Elasticity Matrix which is the result of the utilized Constitutive Law.

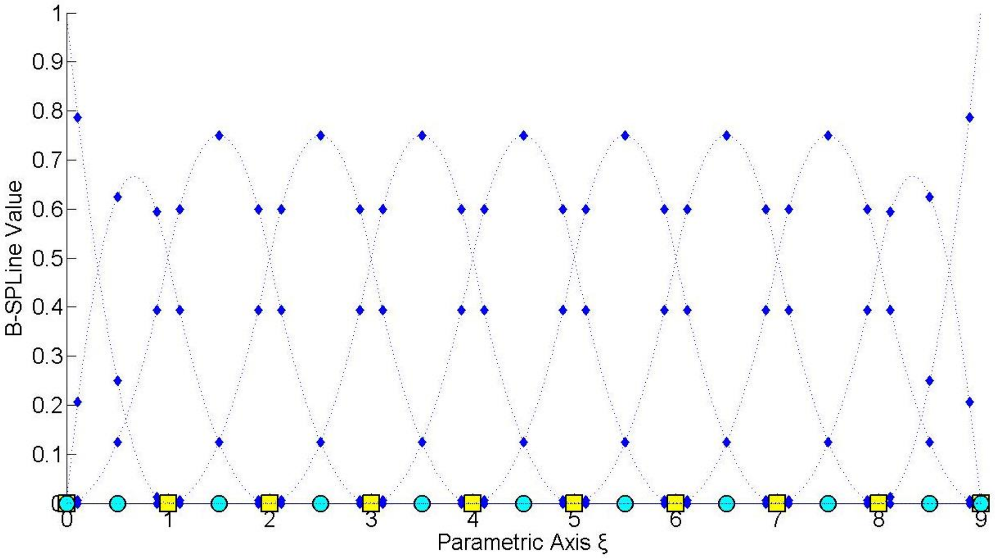


Figure 4.5. Basis functions evaluated as Gauss Points.
$\Xi = \{0 \quad 0 \quad 0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \quad 9 \quad 9 \quad 9\}$

4.3.3 Stiffness Matrix 3D

3D Elasticity is merely the extension of 2D Elasticity in all directions, with a complete stress field. Every other problem can be created by downgrading 3D problems into 2D and 1D problems.

The displacement field for each point in Physical Space is now defined for x, y, z by $u(x,y,z)=u\left(S(\xi,\eta,\zeta)\right)=u(\xi,\eta,\zeta)$, $v(\xi,\eta,\zeta)$, $w(\xi,\eta,\zeta)$ respectively. The strain field can now be defined as:

$$\underset{(6\times1)}{\{\varepsilon\}}=\begin{bmatrix}\varepsilon_x\\ \varepsilon_y\\ \varepsilon_z\\ \gamma_{xy}\\ \gamma_{yz}\\ \gamma_{zx}\end{bmatrix}=\begin{bmatrix}\dfrac{\partial u}{\partial x}\\ \dfrac{\partial v}{\partial y}\\ \dfrac{\partial w}{\partial z}\\ \dfrac{\partial u}{\partial y}+\dfrac{\partial v}{\partial x}\\ \dfrac{\partial v}{\partial z}+\dfrac{\partial w}{\partial y}\\ \dfrac{\partial w}{\partial x}+\dfrac{\partial u}{\partial z}\end{bmatrix}=\begin{bmatrix}\dfrac{\partial}{\partial x} & 0 & 0\\ 0 & \dfrac{\partial}{\partial y} & 0\\ 0 & 0 & \dfrac{\partial}{\partial z}\\ \dfrac{\partial}{\partial y} & \dfrac{\partial}{\partial x} & 0\\ 0 & \dfrac{\partial}{\partial z} & \dfrac{\partial}{\partial y}\\ \dfrac{\partial}{\partial z} & 0 & \dfrac{\partial}{\partial x}\end{bmatrix}\cdot\begin{bmatrix}u\\ v\\ w\end{bmatrix}$$

, which leads to the definition of the Jacobian Matrix for 3D:

$$\begin{bmatrix}\dfrac{\partial\varphi}{\partial\xi}\\ \dfrac{\partial\varphi}{\partial\eta}\\ \dfrac{\partial\varphi}{\partial\zeta}\end{bmatrix}=\begin{bmatrix}\dfrac{\partial x}{\partial\xi} & \dfrac{\partial y}{\partial\xi} & \dfrac{\partial z}{\partial\xi}\\ \dfrac{\partial x}{\partial\eta} & \dfrac{\partial y}{\partial\eta} & \dfrac{\partial z}{\partial\eta}\\ \dfrac{\partial x}{\partial\zeta} & \dfrac{\partial y}{\partial\zeta} & \dfrac{\partial z}{\partial\zeta}\end{bmatrix}\cdot\begin{bmatrix}\dfrac{\partial\varphi}{\partial x}\\ \dfrac{\partial\varphi}{\partial y}\\ \dfrac{\partial\varphi}{\partial z}\end{bmatrix}\Rightarrow\begin{bmatrix}\dfrac{\partial\varphi}{\partial\xi}\\ \dfrac{\partial\varphi}{\partial\eta}\\ \dfrac{\partial\varphi}{\partial\zeta}\end{bmatrix}=\underset{3\times3}{[J]}\cdot\begin{bmatrix}\dfrac{\partial\varphi}{\partial x}\\ \dfrac{\partial\varphi}{\partial y}\\ \dfrac{\partial\varphi}{\partial z}\end{bmatrix}$$

and the inverse Jacobian Matrix as well:

$$\begin{bmatrix}\dfrac{\partial\varphi}{\partial x}\\ \dfrac{\partial\varphi}{\partial y}\\ \dfrac{\partial\varphi}{\partial z}\end{bmatrix}=\underset{(3\times3)}{[J]^{-1}}\cdot\begin{bmatrix}\dfrac{\partial\varphi}{\partial\xi}\\ \dfrac{\partial\varphi}{\partial\eta}\\ \dfrac{\partial\varphi}{\partial\zeta}\end{bmatrix}$$

Jacobian Matrix can be calculated from the derivatives of the Shape functions.

$$\underset{(3\times3)}{[J]} = \begin{bmatrix} R_{1,\xi}(\xi,\eta,\zeta) & R_{2,\xi}(\xi,\eta,\zeta) & \dots & \dots & \dots & R_{N,\xi}(\xi,\eta,\zeta) \\ R_{1,\eta}(\xi,\eta,\zeta) & R_{2.\eta}(\xi,\eta,\zeta) & \dots & \dots & \dots & R_{N,\eta}(\xi,\eta,\zeta) \\ R_{1,\zeta}(\xi,\eta,\zeta) & R_{2,\zeta}(\xi,\eta,\zeta) & \dots & \dots & \dots & R_{N,\zeta}(\xi,\eta,\zeta) \end{bmatrix} \begin{bmatrix} X_1 & Y_1 & Z_1 \\ X_2 & Y_2 & Z_2 \\ . & . & . \\ . & . & . \\ . & . & . \\ X_N & Y_N & Z_N \end{bmatrix}$$

The inverse of the Jacobian Matrix is:

$$\underset{(3\times3)}{[J]^{-1}} = \begin{bmatrix} J^*_{11} & J^*_{12} & J^*_{13} \\ J^*_{21} & J^*_{22} & J^*_{23} \\ J^*_{31} & J^*_{32} & J^*_{33} \end{bmatrix}$$

$[B_1]$ is evaluated as:

$$\underset{(6\times9)}{\left[B_1(\xi,\eta,\zeta)\right]} = \begin{bmatrix} J^*_{11} & J^*_{12} & J^*_{13} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & J^*_{21} & J^*_{22} & J^*_{23} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & J^*_{31} & J^*_{32} & J^*_{33} \\ J^*_{21} & J^*_{22} & J^*_{23} & J^*_{11} & J^*_{12} & J^*_{13} & 0 & 0 & 0 \\ 0 & 0 & 0 & J^*_{31} & J^*_{32} & J^*_{33} & J^*_{21} & J^*_{22} & J^*_{23} \\ J^*_{31} & J^*_{32} & J^*_{33} & 0 & 0 & 0 & J^*_{11} & J^*_{12} & J^*_{13} \end{bmatrix}$$

As for $[B_2]$:

$$\underset{(9\times3N)}{\left[B_2(\xi,\eta,\zeta)\right]} = \begin{bmatrix} R_{1,\xi} & 0 & 0 & R_{2,\xi} & 0 & 0 & \dots & \dots & \dots & R_{N,\xi} & 0 & 0 \\ R_{1,\eta} & 0 & 0 & R_{2,\eta} & 0 & 0 & \dots & \dots & \dots & R_{N,\eta} & 0 & 0 \\ R_{1,\zeta} & 0 & 0 & R_{2,\zeta} & 0 & 0 & \dots & \dots & \dots & R_{N,\zeta} & 0 & 0 \\ 0 & R_{1,\xi} & 0 & 0 & R_{2,\xi} & 0 & \dots & \dots & \dots & 0 & R_{N,\xi} & 0 \\ 0 & R_{1,\eta} & 0 & 0 & R_{2,\eta} & 0 & \dots & \dots & \dots & 0 & R_{N,\eta} & 0 \\ 0 & R_{1,\zeta} & 0 & 0 & R_{2,\zeta} & 0 & \dots & \dots & \dots & 0 & R_{N,\zeta} & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & R_{1,\xi} & 0 & 0 & R_{2,\xi} & \dots & \dots & \dots & 0 & 0 & R_{N,\xi} \\ 0 & 0 & R_{1,\eta} & 0 & 0 & R_{2,\eta} & \dots & \dots & \dots & 0 & 0 & R_{N,\eta} \\ 0 & 0 & R_{1,\zeta} & 0 & 0 & R_{2,\zeta} & \dots & \dots & \dots & 0 & 0 & R_{N,\zeta} \end{bmatrix}$$

As a result, the Deformation Matrix for 3D Elasticity is calculated as:

$$\underset{(6\times 3N)}{\left[B(\xi,\eta,\zeta)\right]} = \underset{(6\times 9)}{\left[B_1(\xi,\eta,\zeta)\right]} \cdot \underset{(9\times 3N)}{\left[B_2(\xi,\eta,\zeta)\right]}$$

The corresponding Stiffness Matrix is produced by integration

$$\underset{(3N\times 3N)}{[K]} = \int_{\xi_0}^{\xi_{n+p+1}} \int_{\eta_0}^{\eta_{m+q+1}} \int_{\zeta_0}^{\zeta_{l+r+1}} \underset{(3N\times 6)}{\left[B(\xi,\eta,\zeta)\right]^T} \cdot \underset{(6x6)}{[E]} \cdot \underset{(6\times 3N)}{\left[B(\xi,\eta,\zeta)\right]} \cdot \det[J]\, d\zeta d\eta d\xi$$

Numerical integration is used in 3D as well

$$\underset{(3N\times 3N)}{[K]} = \sum_{i=1}^{GP\xi} \sum_{j=1}^{GP\eta} \sum_{k=1}^{GP\zeta} \underset{(3N\times 6)}{\left[B(\xi_i,\eta_j,\zeta_k)\right]^T} \cdot \underset{(6x6)}{[E]} \cdot \underset{(6\times 3N)}{\left[B(\xi_i,\eta_j,\zeta_k)\right]} \cdot \det[J] \cdot w_i^{GP\xi} \cdot w_j^{GP\eta} \cdot w_k^{GP\zeta}$$

where:

- GP_ξ : the total number of Gauss Points per ξ for the specific Patch
- GP_η : the total number of Gauss Points per η for the specific Patch
- GP_ζ : the total number of Gauss Points per ζ for the specific Patch
- ξ_i, η_j, ζ_k : the coordinates of the tensor product Gauss Point ijk
- $w_i^{GP\xi}$, $w_j^{GP\eta}$, $w_k^{GP\zeta}$: the corresponding weights of Gauss Points

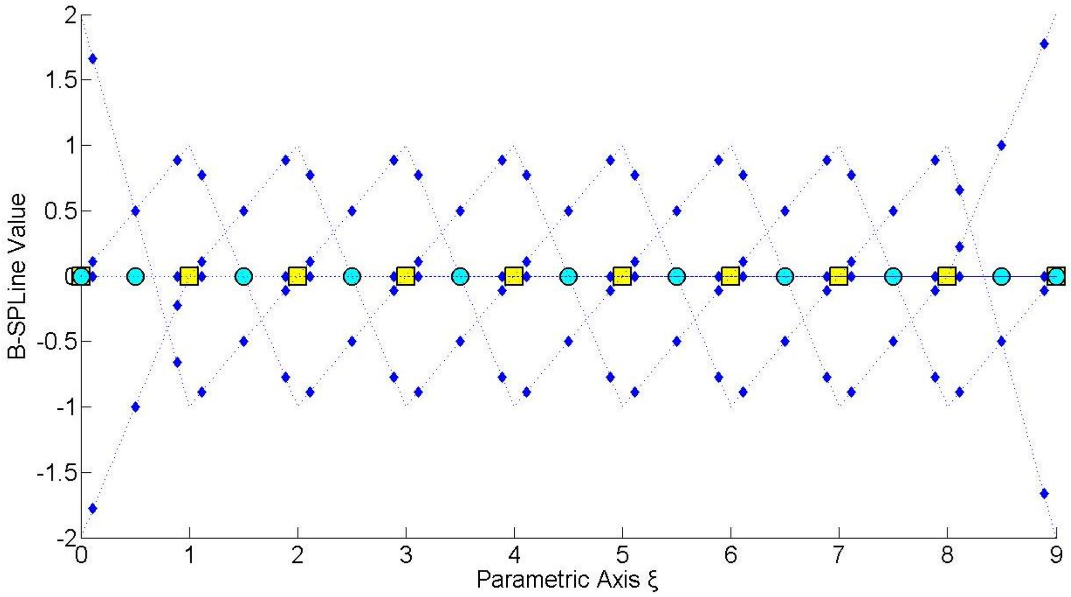
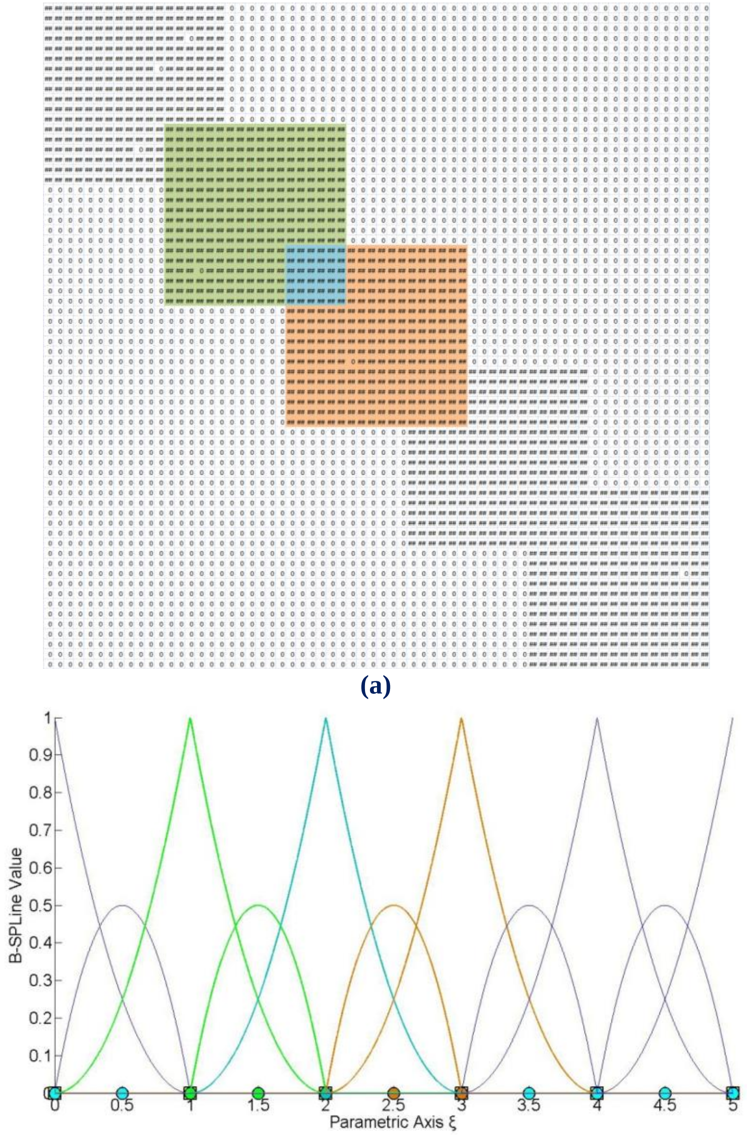


Figure 4.6. Basis function derivatives evaluated as Gauss Points.
Knot Value Vector
$\Xi = \{0 \quad 0 \quad 0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \quad 9 \quad 9 \quad 9\}$

4.3.4 Stiffness Matrix Examples



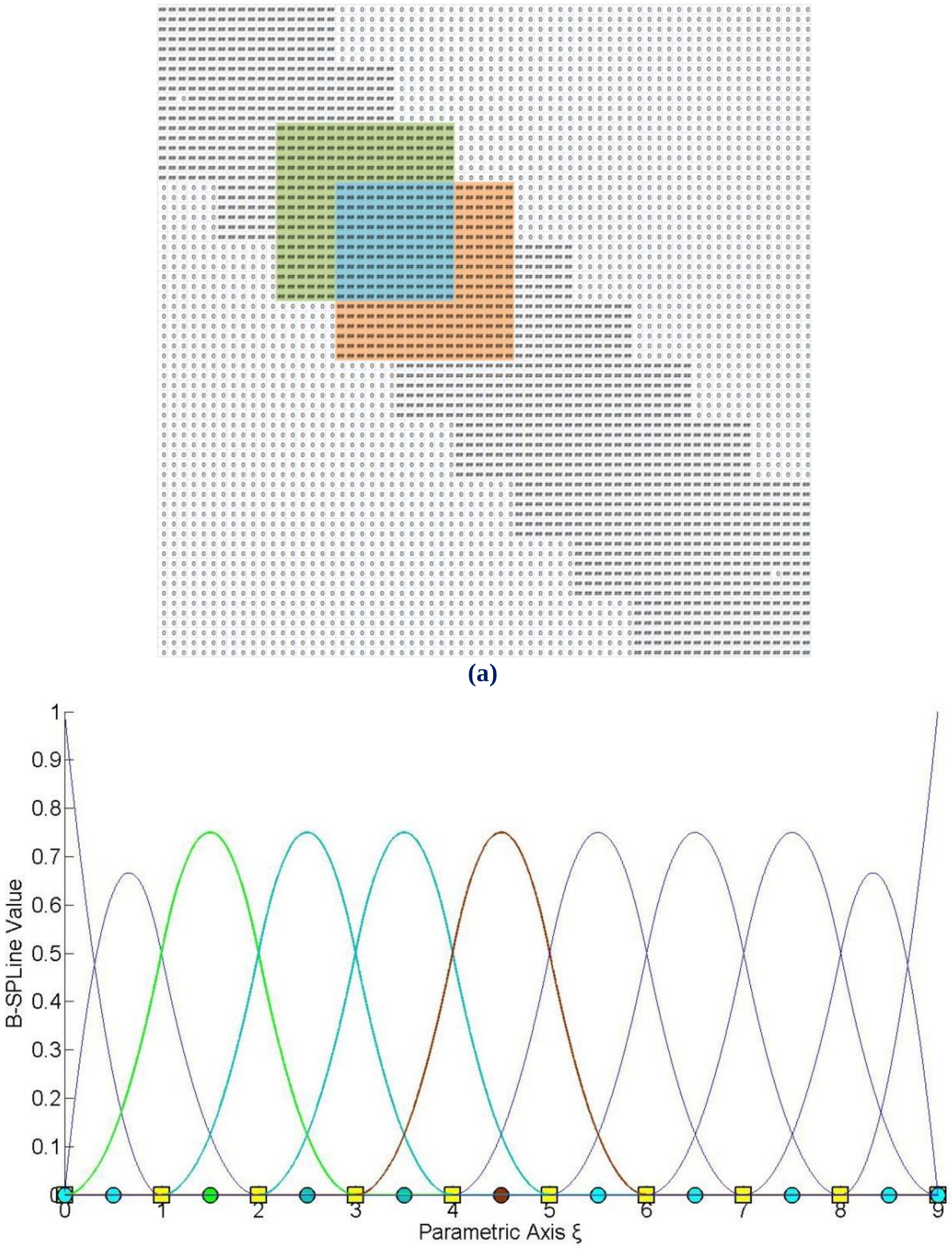
(a)

(b)

Figure 4.7. (a) Stiffness Matrix of the model where 1470 non-zero elements are produced and **(b)** its corresponding basis functions.

In Figure 4.7, the degree is equal to $p = 2$, Continuity is C^0, and the model has 66 degrees of freedom. The non-zero elements of the matrix are 1470. The interconnectivity between the elements applies in a small part of the matrix, as it is shown in Figure 4.7.a. This occurs because the overlapping between the B-SPLines exists only for $N_{5,2}(\xi)$.

The non-zero elements at the matrix are a result of the shared support between the B-SPLines, which exists for $2p$ B-SPLines. $N_{5,2}(\xi)$ shares support with the B-SPLines $N_{3,2}(\xi)$, $N_{4,2}(\xi)$, $N_{6,2}(\xi)$ and $N_{7,2}(\xi)$. For these B-SPLines, the elements of the matrix are non-zero. This process requires less composition time, because of the limited number of interconnections.



(a)

(b)

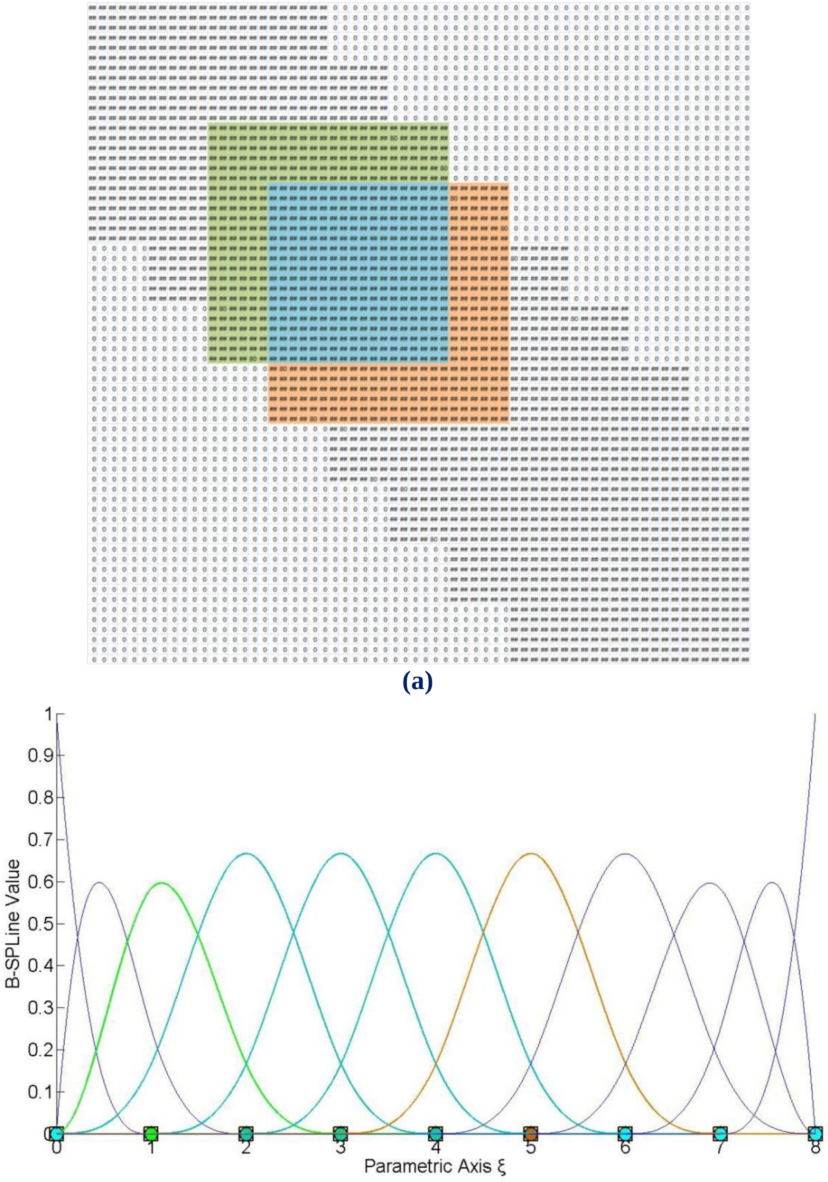
Figure 4.8. (a) Stiffness Matrix of the model where 1762 non-zero elements are produced and **(b)** its corresponding basis functions.

In Figure 4.8, the model is analyzed for a degree of $p = 2$ and C^1 Continuity with the same 66 degrees of freedom. The interconnectivity between the elements has increased and therefore provides a better approximation of the model than the one in Figure 4.7. In this case, the configuration of the matrix requires more time, but the overlapping between the B-SPLines leads to better results.

A Stiffness Matrix coefficient is non-zero when shared support applies for the corresponding B-SPLine and extensively between Control Points.

In particular, every B-SPLine shares support with $2p$ B-SPLines. In Knot Span $[2,3)$ of Figure 4.8, the B-SPLines $N_{3,2}(\xi)$, $N_{4,2}(\xi)$ and $N_{5,2}(\xi)$ are non-zero and in Knot Span $[3,4)$, $N_{4,2}(\xi)$, $N_{5,2}(\xi)$ and $N_{6,2}(\xi)$. These two Knot Spans are connected to each other with the B-SPLines $N_{4,2}(\xi)$ and $N_{5,2}(\xi)$.

The B-SPLine $N_{4,2}(\xi)$ shares support with p B-SPLines on each side, in particular $N_{2,2}(\xi)$, $N_{3,2}(\xi)$, $N_{5,2}(\xi)$ and $N_{6,2}(\xi)$. The same applies for the B-SPLine $N_{5,2}(\xi)$, as it shares support with $N_{3,2}(\xi)$, $N_{4,2}(\xi)$, $N_{6,2}(\xi)$ and $N_{7,2}(\xi)$.



(a)

(b)

Figure 4.9. (a) Stiffness Matrix of the model where 2340 non-zero elements are produced and **(b)** its corresponding basis functions.

In Figure 4.9, the model is analyzed for $p = 3$ and C^2 Continuity and for the same 66 degrees of freedom.

In Knot Span $[2,3)$ $N_{3,3}(\xi)$, $N_{4,3}(\xi)$, $N_{5,3}(\xi)$ and $N_{6,3}(\xi)$ are non-zero and in Knot Span $[3,4)$ the B-SPLines $N_{4,3}(\xi)$, $N_{5,3}(\xi)$, $N_{6,3}(\xi)$ and $N_{7,3}(\xi)$ respectively. In this case, the interconnectivity between the elements expands in almost all the degrees of freedom, because the overlapping between basis functions exists for $N_{4,3}(\xi)$, $N_{5,3}(\xi)$ and $N_{6,3}(\xi)$.

The B-SPLine $N_{4,3}(\xi)$ shares support with $2p$ B-SPLines as it is previously mentioned, three B-SPLines on the left and three B-SPLines on the right, as it is shown in Figure 4.9. In particular, $N_{4,3}(\xi)$ shares support with $N_{1,3}(\xi)$, $N_{2,3}(\xi)$, $N_{3,3}(\xi)$, $N_{5,3}(\xi)$, $N_{6,3}(\xi)$, $N_{7,3}(\xi)$.

For these B-SPLines, the corresponding coefficients at the Stiffness Matrix are non-zero. The same applies for $N_{5,3}(\xi)$ and $N_{6,3}(\xi)$. Therefore, the number of the non-zero elements has increased even more, thus the approximation in this solution is more close to the real problem.

Figure 4.11 represents the influence of the continuity in the analysis of the model in Figure 4.10. As the continuity increases, the interconnectivity of the elements affects more degrees of freedom. This leads to a better approximation of the physical problem, but also the creation of the Stiffness Matrix becomes more time-consuming. For the same polynomial order, more non-zero elements are created at the Stiffness Matrix.

This is a result of the shared support of the B-SPLine basis functions, that exists for $2 \cdot p$ B-SPLines, which leads to overlapping between the B-SPLines and interconnectivity of the degrees of freedom in the Stiffness Matrix. Therefore, as Continuity increases, the number of non-zero elements increases as well.

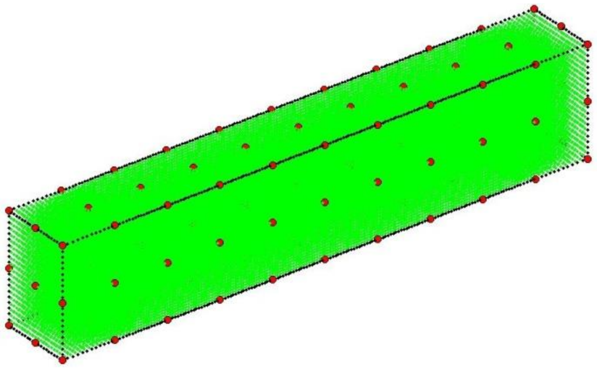
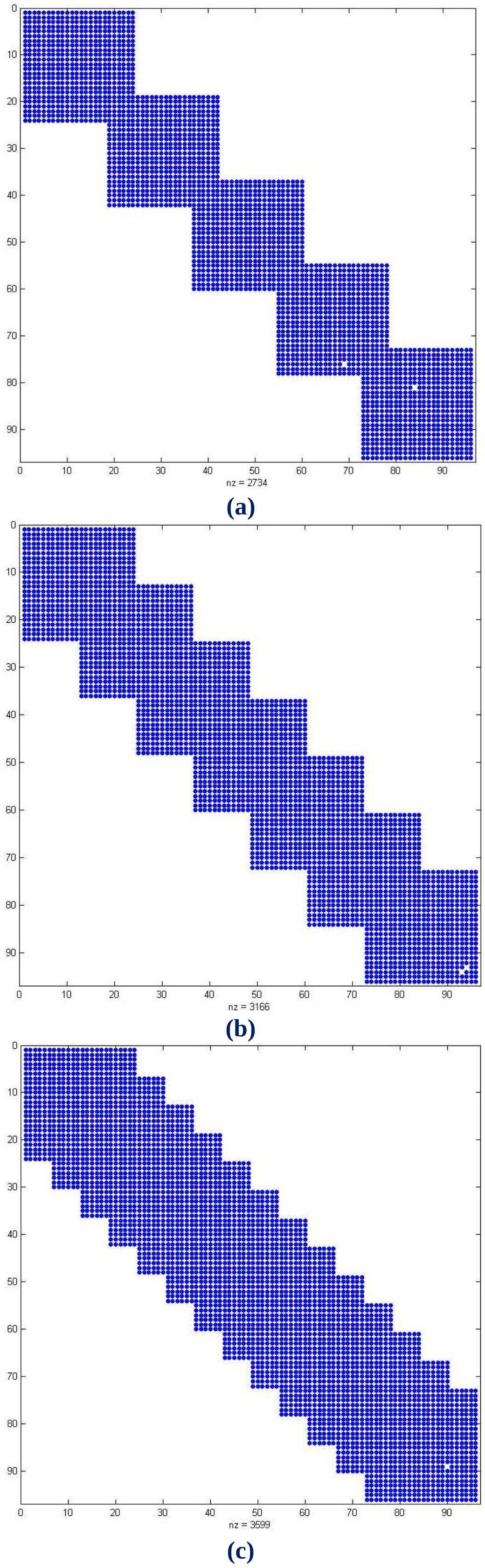


Figure 4.10. 3D Structure for Continuity investigation.



(a)

(b)

(c)

Figure 4.11. Stiffness Matrix for $p = 3$ and 99 degrees of freedom.

(a) C^0 Continuity

(b) C^1 Continuity

(c) C^2 Continuity

5 External Loads, Boundary Conditions and Stress Field

5.1 Analysis

5.1.1 External Load

After the Stiffness Matrix Assembly, the external Load Vector has to be calculated. Concentrated Loads can be assigned directly at the Degrees of Freedom. Distributed loads $f(\xi,\eta,\zeta)$ have to be transformed into equivalent concentrated loads by integration [2]:

$$\underset{(N\times 1)}{\{F\}} = \int_{\xi_0}^{\xi_{n+p+1}} \int_{\eta_0}^{\eta_{m+q+1}} \int_{\zeta_0}^{\zeta_{l+r+1}} \underset{(N\times 1)}{\{R(\xi,\eta,\zeta)\}} \cdot \underset{(1\times 1)}{f(\xi,\eta,\zeta)} \cdot \det[J]\, d\zeta d\eta d\xi$$

More specifically, for each case:

1D:

$$\underset{(N\times 1)}{\{F\}} = \int_{\xi_0}^{\xi_{n+p+1}} \underset{(N\times 1)}{\{R(\xi)\}} \cdot \underset{(1\times 1)}{f(\xi)} \cdot \det[J] d\xi$$

2D:

$$\underset{(N\times 1)}{\{F\}} = \int_{\xi_0}^{\xi_{n+p+1}} \int_{\eta_0}^{\eta_{m+q+1}} \underset{(N\times 1)}{\{R(\xi,\eta)\}} \cdot \underset{(1\times 1)}{f(\xi,\eta)} \cdot \det[J] d\eta d\xi$$

3D:

$$\underset{(N\times 1)}{\{F\}} = \int_{\xi_0}^{\xi_{n+p+1}} \int_{\eta_0}^{\eta_{m+q+1}} \int_{\zeta_0}^{\zeta_{l+r+1}} \underset{(N\times 1)}{\{R(\xi,\eta,\zeta)\}} \cdot \underset{(1\times 1)}{f(\xi,\eta,\zeta)} \cdot \det[J] d\zeta d\eta d\xi$$

This way, the Load Vector $\{F\}$ is assembled.

5.1.2 Refined Load

If a load vector has already been evaluated for a coarse mesh, there is no need to apply numerical integration for the new load vector of the fine mesh. New load values can be evaluated directly from the coarse mesh.

Let $\left\{L^C\right\}$, $\left\{L^F\right\}$ be the load vectors, $\left\{N^C\left(\xi\right)\right\}$, $\left\{N^F\left(\xi\right)\right\}$ the basis functions and $\left\{X^C\right\}$, $\left\{X^F\right\}$ the Control Point Cartesian Coordinates for the coarse and fine mesh respectively. Let $f\left(\xi\right)$ be the load distribution in the physical space. It applies that:

$$\underset{(n\times1)}{\left\{L^C\right\}} = \int_{\xi_0}^{\xi_{max}} \underset{(n\times1)}{\left\{N^C\left(\xi\right)\right\}} \cdot \underset{(1\times1)}{f\left(\xi\right)} d\xi$$

Both meshes provide the same geometrical representation, resulting in:

$$C\left(\xi\right) = \underset{(1\times n)}{\left\{N^C\left(\xi\right)\right\}^T} \cdot \underset{(n\times1)}{\left\{X^C\right\}}$$

$$C\left(\xi\right) = \underset{(1\times m)}{\left\{N^F\left(\xi\right)\right\}^T} \cdot \underset{(m\times1)}{\left\{X^F\right\}}$$

Thus,

$$\underset{(1\times m)}{\left\{N^F\left(\xi\right)\right\}^T} \cdot \underset{(m\times1)}{\left\{X^F\right\}} = \underset{(1\times n)}{\left\{N^C\left(\xi\right)\right\}^T} \cdot \underset{(n\times1)}{\left\{X^C\right\}}$$

It has been established, in Chapter 3, that:

$$\underset{(n\times1)}{\left\{X^C\right\}} = \underset{(n\times m)}{\left[T^{CF}\right]} \cdot \underset{(m\times1)}{\left\{X^F\right\}}$$

Therefore,

$$\underset{(1\times m)}{\left\{N^F\left(\xi\right)\right\}^T} \cdot \underset{(m\times1)}{\left\{X^F\right\}} = \underset{(1\times n)}{\left\{N^C\left(\xi\right)\right\}^T} \cdot \underset{(n\times1)}{\left\{X^C\right\}} = \underset{(1\times n)}{\left\{N^C\left(\xi\right)\right\}^T} \cdot \underset{(n\times m)}{\left[T^{CF}\right]} \cdot \underset{(m\times1)}{\left\{X^F\right\}} \Rightarrow$$

$$\underset{(1\times m)}{\left\{N^F\left(\xi\right)\right\}^T} = \underset{(1\times n)}{\left\{N^C\left(\xi\right)\right\}^T} \cdot \underset{(n\times m)}{\left[T^{CF}\right]}$$

This leads to

$$\underset{m\times1}{\left\{N^F(\xi)\right\}} = \underset{m\times n}{\left[T^{CF}\right]^T} \cdot \underset{n\times1}{\left\{N^C\left(\xi\right)\right\}}$$

The fine mesh load vector is evaluated by:

$$\underset{(m\times1)}{\left\{L^{F}\right\}}=\int_{\xi_0}^{\xi_{max}}\underset{(m\times1)}{\left\{N^{F}\left(\xi\right)\right\}}\cdot\underset{(1\times1)}{f\left(\xi\right)}d\xi\Rightarrow$$

$$\underset{(m\times1)}{\left\{L^{F}\right\}}=\int_{\xi_0}^{\xi_{max}}\underset{(m\times n)}{\left[T^{CF}\right]^{T}}\cdot\underset{(n\times1)}{\left\{N^{C}\left(\xi\right)\right\}}\cdot\underset{(1\times1)}{f\left(\xi\right)}d\xi\Rightarrow$$

$$\underset{(m\times1)}{\left\{L^{F}\right\}}=\underset{(m\times n)}{\left[T^{CF}\right]^{T}}\int_{\xi_0}^{\xi_{max}}\underset{(n\times1)}{\left\{N^{C}\left(\xi\right)\right\}}\cdot\underset{(1\times1)}{f\left(\xi\right)}d\xi$$

In conclusion,

$$\underset{(m\times1)}{\left\{L^{F}\right\}}=\underset{(m\times n)}{\left[T^{CF}\right]^{T}}\cdot\underset{(n\times1)}{\left\{L^{C}\right\}}$$

In the same manner, equivalent loads for Reverse-Refinement can be defined as:

$$\underset{(n\times1)}{\left\{L^{C}\right\}}=\underset{(n\times m)}{\left[T^{FC}\right]^{T}}\cdot\underset{(m\times1)}{\left\{L^{F}\right\}}$$

An engineer must always bear in mind the restrictions set for Reverse Refinement. Both in geometrical representation and in equivalent load evaluation, this procedure only works under certain circumstances. More specifically, loads refined from a coarse mesh can be transformed back to that coarse mesh. On the other hand, loads created initially from a fine mesh will probably be transformed incorrectly to the coarse mesh.

5.1.3 Boundary Conditions

Certain Degrees of Freedom are fixed, in that their displacements are zero. These are called stationary and the corresponding rows and columns are deleted from the Stiffness Matrix and the Load Vector. This leaves a Stiffness Matrix and a Load Vector having only free Degrees of Freedom, $\left[K_{ff}\right]$ and $\left\{F_{f}\right\}$ respectively. The solution of the equation is the final step in analysis:

$$\left\{F_{f}\right\}=\left[K_{ff}\right]\cdot\left\{D_{f}\right\}\Rightarrow\left\{D_{f}\right\}=\left[K_{ff}\right]^{-1}\cdot\left\{F_{f}\right\}$$

The (zero) displacements for the stationary Degrees of Freedom are added back to the result, thus creating the Displacement Vector $\left\{D\right\}$.

5.2 Displacement, Strain and Stress Field

5.2.1 Displacement

After the solution of the equation, Control Point Displacements are obtained. Unlike classical FEM, Control Points are usually placed outside the area of the model. In general, displacements of the model differ from the displacements of the corresponding Control Points. Conclusively, analysis results are considered “Pseudo-Displacements” and play an auxiliary role in calculating the physical model ones. As mentioned before, the distribution of the displacement field is achieved via Shape Functions [2]:

1D:

$$d(\xi)=\sum_{i=1}^{N}\left\{R_i(\xi)\cdot D_i\right\}=\underset{(1\times N)}{\left\{R_i(\xi)\right\}^T}\cdot\underset{(N\times 1)}{\left\{D\right\}}$$

2D:

$$d(\xi,\eta)=\sum_{i=1}^{N}\left\{R_i(\xi,\eta)\cdot D_i\right\}=\underset{(1\times N)}{\left\{R_i(\xi,\eta)\right\}^T}\cdot\underset{(N\times 1)}{\left\{D\right\}}$$

3D:

$$d(\xi,\eta,\zeta)=\sum_{i=1}^{N}\left\{R_i(\xi,\eta,\zeta)\cdot D_i\right\}=\underset{(1\times N)}{\left\{R_i(\xi,\eta,\zeta)\right\}^T}\cdot\underset{(N\times 1)}{\left\{D\right\}}$$

where:

- R_i is the Shape Function i
- D_i is the Displacement of the corresponding Control Point
- N is the total number of Control Points

If a Control Point c is interpolatory to the curve at (ξ_c,η_c,ζ_c), it follows that:

$$d(\xi_c,\eta_c,\zeta_c)=\sum_{i=1}^{N}\left\{R_i(\xi_c,\eta_c,\zeta_c)\cdot D_i\right\}=1\cdot D_c=D_c\,.$$

Displacements of interpolatory Control Points are physical model’s displacements as well.

5.2.2 Stress and Strain

The Strain Vector can be evaluated at any point in the field with the help of Control Point Displacements and the Deformation Matrix $\left[\mathrm{B}\right]$ [2]:

1D

$$\underset{(1\times1)}{\left\{\varepsilon(\xi)\right\}}=\underset{(1\times\mathrm{N})}{\left[\mathrm{B}(\xi)\right]}\cdot\underset{(\mathrm{N}\times1)}{\left\{\mathrm{D}\right\}}$$

2D

$$\underset{(3\times1)}{\left\{\varepsilon(\xi,\eta)\right\}}=\underset{(3\times2\mathrm{N})}{\left[\mathrm{B}(\xi,\eta)\right]}\cdot\underset{(2\mathrm{N}\times1)}{\left\{\mathrm{D}\right\}}$$

3D

$$\underset{(6\times1)}{\left\{\varepsilon(\xi,\eta,\zeta)\right\}}=\underset{(6\times3\mathrm{N})}{\left[\mathrm{B}(\xi,\eta,\zeta)\right]}\cdot\underset{(3\mathrm{N}\times1)}{\left\{\mathrm{D}\right\}}$$

Applying Hooke’s Constitutive Law leads to:

1D

$$\underset{(1\times1)}{\left\{\sigma(\xi)\right\}}=\underset{(1\times1)}{\left[\mathrm{E}\right]}\cdot\underset{(1\times1)}{\left\{\varepsilon(\xi)\right\}}=\underset{(1\times1)}{\left[\mathrm{E}\right]}\cdot\underset{(1\times\mathrm{N})}{\left[\mathrm{B}(\xi)\right]}\cdot\underset{(\mathrm{N}\times1)}{\left\{\mathrm{D}\right\}}$$

2D

$$\underset{(3\times1)}{\left\{\sigma(\xi,\eta)\right\}}=\underset{(3\times3)}{\left[\mathrm{E}\right]}\cdot\underset{(3\times1)}{\left\{\varepsilon(\xi,\eta)\right\}}=\underset{(3\times3)}{\left[\mathrm{E}\right]}\cdot\underset{(3\times2\mathrm{N})}{\left[\mathrm{B}(\xi,\eta)\right]}\cdot\underset{(2\mathrm{N}\times1)}{\left\{\mathrm{D}\right\}}$$

3D

$$\underset{(6\times1)}{\left\{\sigma(\xi,\eta,\zeta)\right\}}=\underset{(6\times6)}{\left[\mathrm{E}\right]}\cdot\underset{(6\times1)}{\left\{\varepsilon(\xi,\eta,\zeta)\right\}}=\underset{(6\times6)}{\left[\mathrm{E}\right]}\cdot\underset{(6\times3\mathrm{N})}{\left[\mathrm{B}(\xi,\eta,\zeta)\right]}\cdot\underset{(3\mathrm{N}\times1)}{\left\{\mathrm{D}\right\}}$$

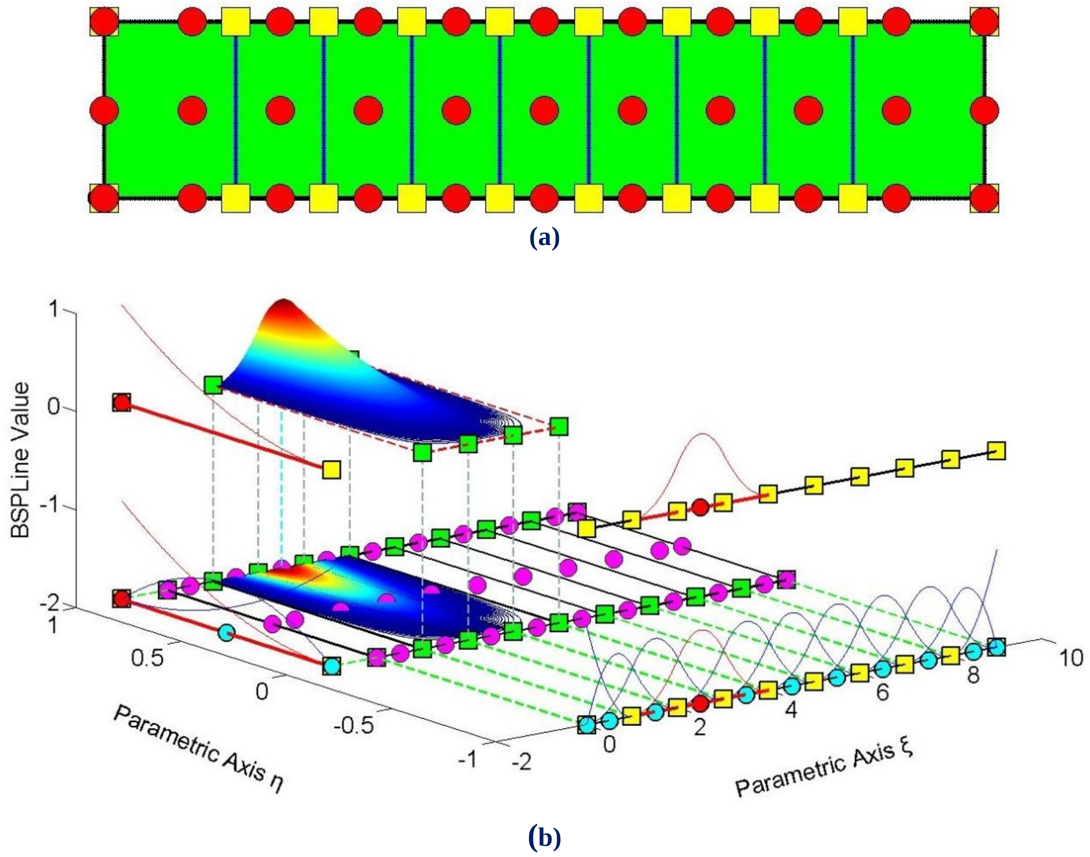
Note that stress and strain vectors are evaluated via the derivatives of the shape functions. This means that their distribution is going to be one order less than the displacement distribution. This is why stress and strain continuity cannot be achieved in FEM models, where shape functions are always C^{-1} continuous. This problem is solved when the derivatives of the shape functions are also continuous, which means using shape functions with C^{1} continuity or higher.

6 Applications

6.1 Cantilever 2D

The cantilever presented here is subjected to plane stress. The third dimension (thickness) is significantly smaller than the other two. The degrees of freedom on the left side are fixed. The load is applied on the right side of the cantilever.

The first representation consists of quadratic basis functions on axes ξ,η. There are formed 9 Knot Rectangles (Isogeometric elements). Knot boundaries are displayed in blue.



(a)

(b)

Figure 6.1. (a) Physical Space and **(b)** Basis Functions for Cantilever 15x3x0.01 m

Knot Value Vector: $\Xi = \{0 \quad 0 \quad 0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \quad 9 \quad 9 \quad 9\}$

Knot Value Vector $\mathrm{H} = \{0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1\}$

Gauss point coordinates are evaluated for every Knot Span. In this 2D problem, $p+1=3$ Gauss points per Knot Span are required. Thus, $3\cdot 3=9$ Gauss points per Isogeometric Element.

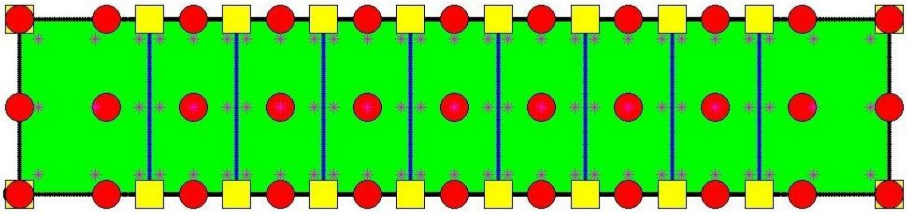


Figure 6.2. Gauss Point coordinates in Physical Space. 9 Gauss Points per Knot Rectangle.

There are $n=11$ Control points on ξ and $m=3$ Control points on η, for a total of 33 points and 66 degrees of freedom. The overlapping of the basis functions can also be seen in the non-zero elements of the Stiffness Matrix.

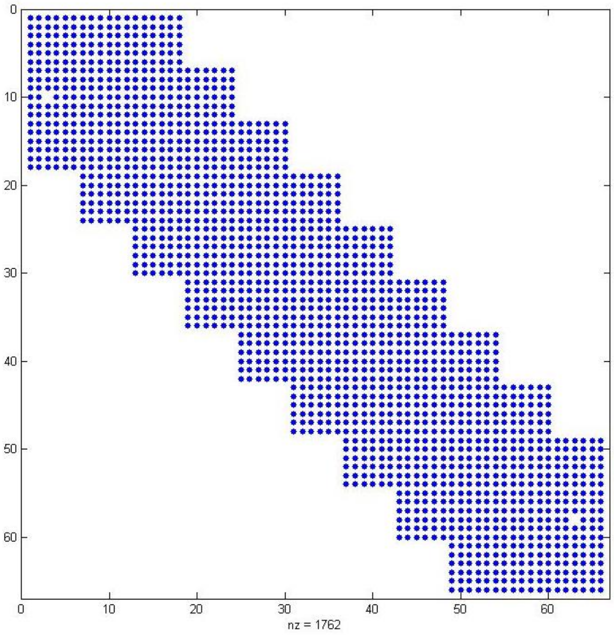


Figure 6.3. Stiffness Matrix. 66 degrees of freedom. Non-zero elements are drawn in blue.

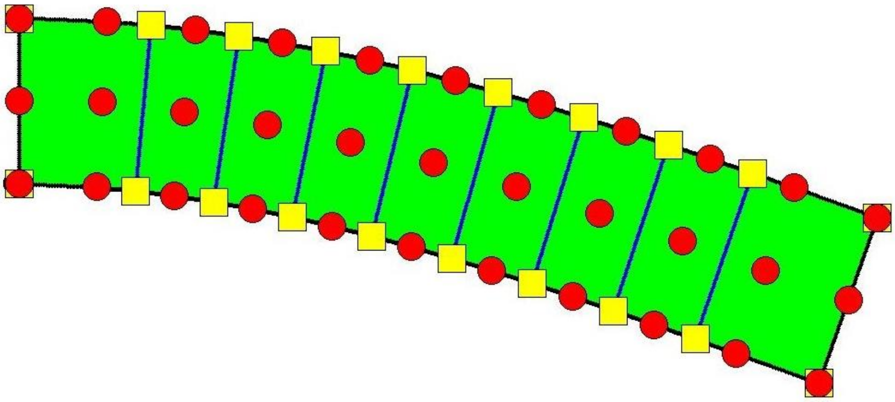


Figure 6.4. Real Displacements on Physical Space

In Figure 6.5, contour for σ_{xx} is represented. The cantilever is subjected to transverse bending. Note that stresses are continuous everywhere in the model, as a result of C^1 Continuity basis functions.

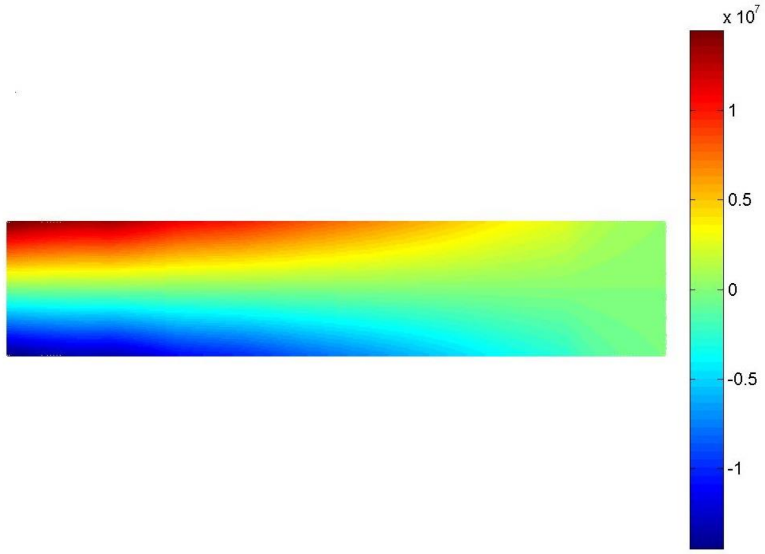
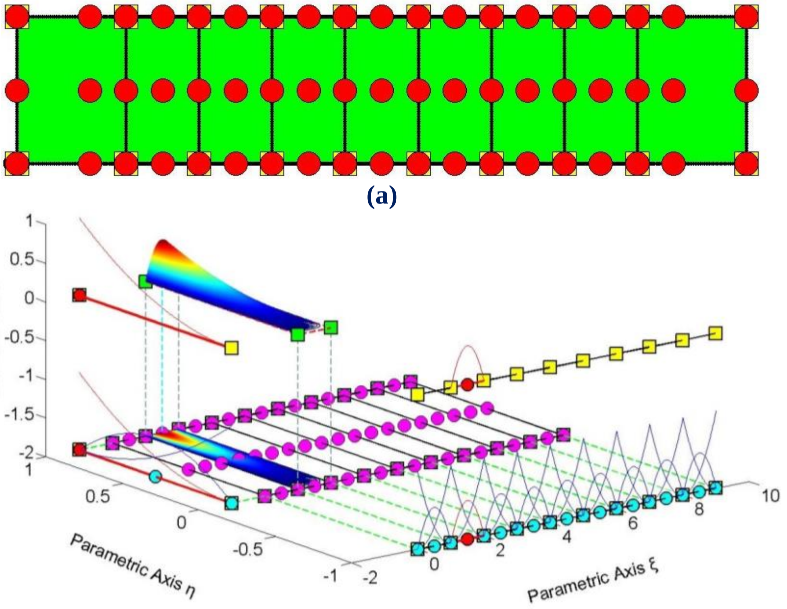


Figure 6.5. Contour for σ_{xx}

The same model is analyzed with C^0 Continuity across all Knots. This is achieved by h-Refinement. No new Knots are added, but multiplicity of internal Knots is increased. The model and basis functions are represented in Figure 6.6, where the patch boundaries are drawn in black.



(b)

Figure 6.6. (a) Physical Space
(b) Basis Functions for Cantilever 15x3x0.01 m

$$\Xi = \{0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 2 \quad 2 \quad 3 \quad 3 \quad 4 \quad 4 \quad 5 \quad 5 \quad 6 \quad 6 \quad 7 \quad 7 \quad 8 \quad 8 \quad 9 \quad 9 \quad 9\}$$

$$H = \{0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1\}$$

In order to maintain the same elements, more control points have to be added. The refined, C^0 Continuity model consists of $19 \cdot 3 = 57$ Control Points and 114 degrees of freedom. Limited Continuity between the elements is also illustrated in the form of the Stiffness Matrix. The limited interconnectivity can be seen by the vast number of zero elements in the Matrix.

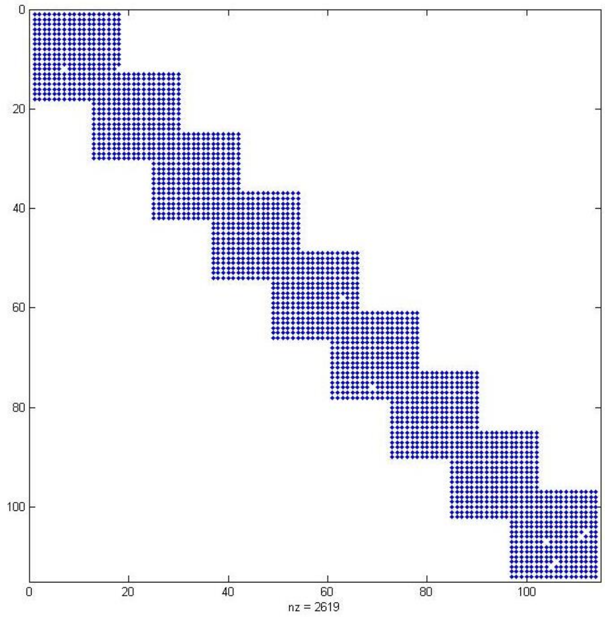


Figure 6.7. Stiffness Matrix for C^0 Continuity. 114 degrees of freedom. Non-zero elements are drawn in blue.

Contour for σ_{xx} is represented in Figure 6.8. Discontinuity of stresses across Patch boundary is clearly visible. This is a great advantage of Isogeometric Analysis over FEM.

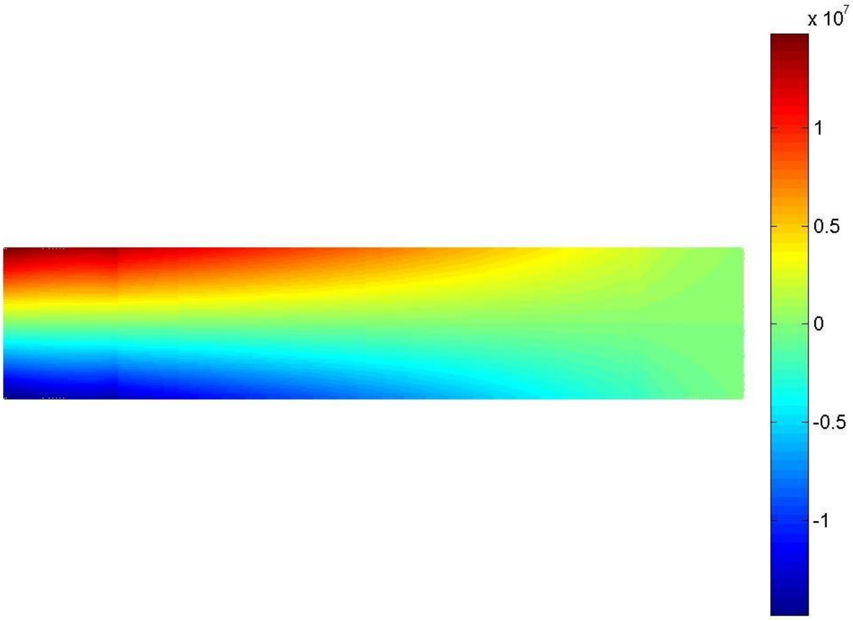
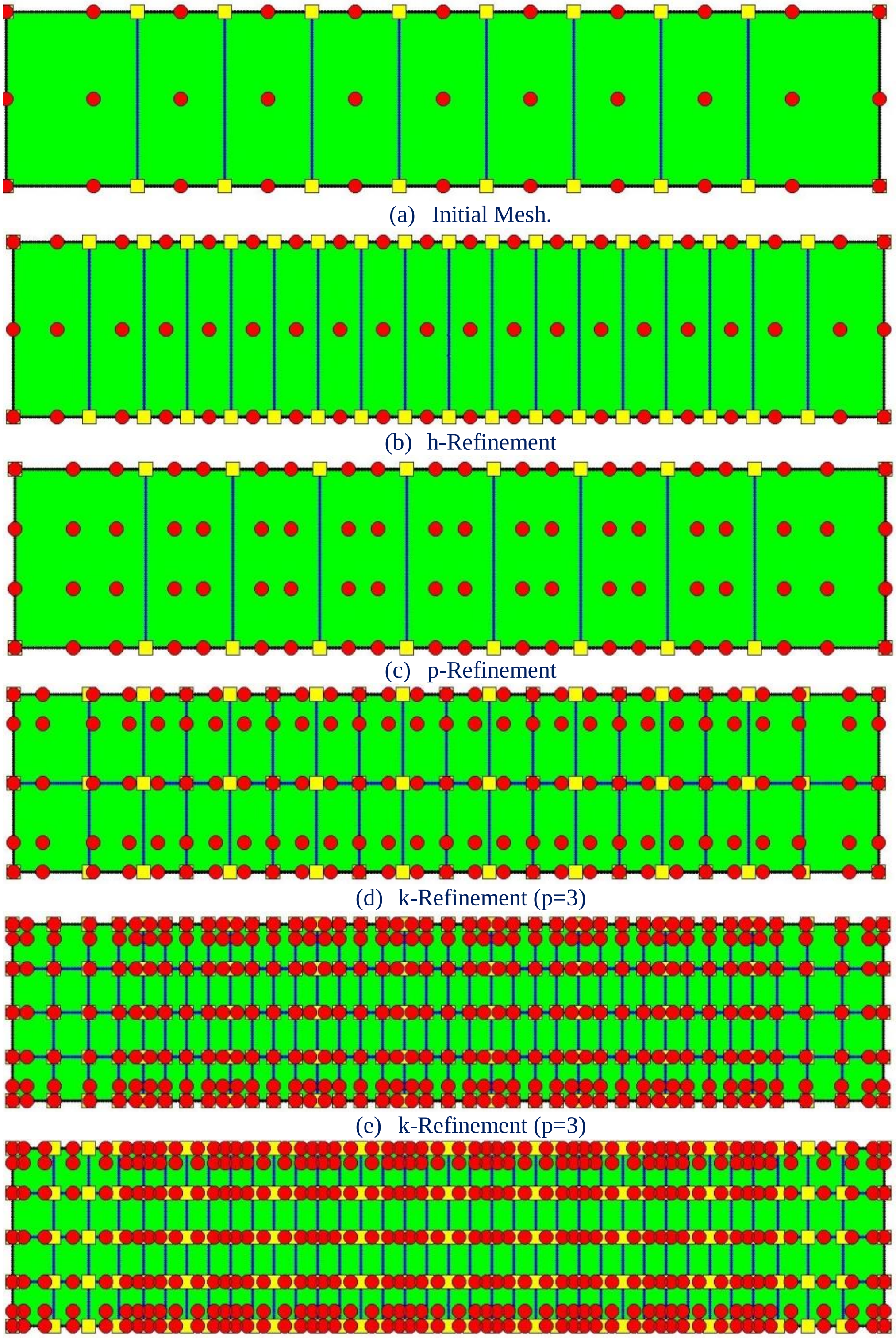


Figure 6.8. Contour for σ_{xx}

A parametric investigation was conducted to the Cantilever Model. One h-Refinement, one p-Refinement and three k-Refinements were applied. Physical space, Control Net and Knot Mesh are presented in Figure 6.9 below.



(a) Initial Mesh.

(b) h-Refinement

(c) p-Refinement

(d) k-Refinement (p=3)

(e) k-Refinement (p=3)

(f) k-Refinement (p=4)

Figure 6.9. Physical Space for Refined Cantilever.

After the first k-Refinement, a deviation of 0.07 % is achieved. The combination of different order with increased continuity yields very accurate results. Moreover, cubic basis functions tend to be very efficient for analysis; this is yet to be explained.

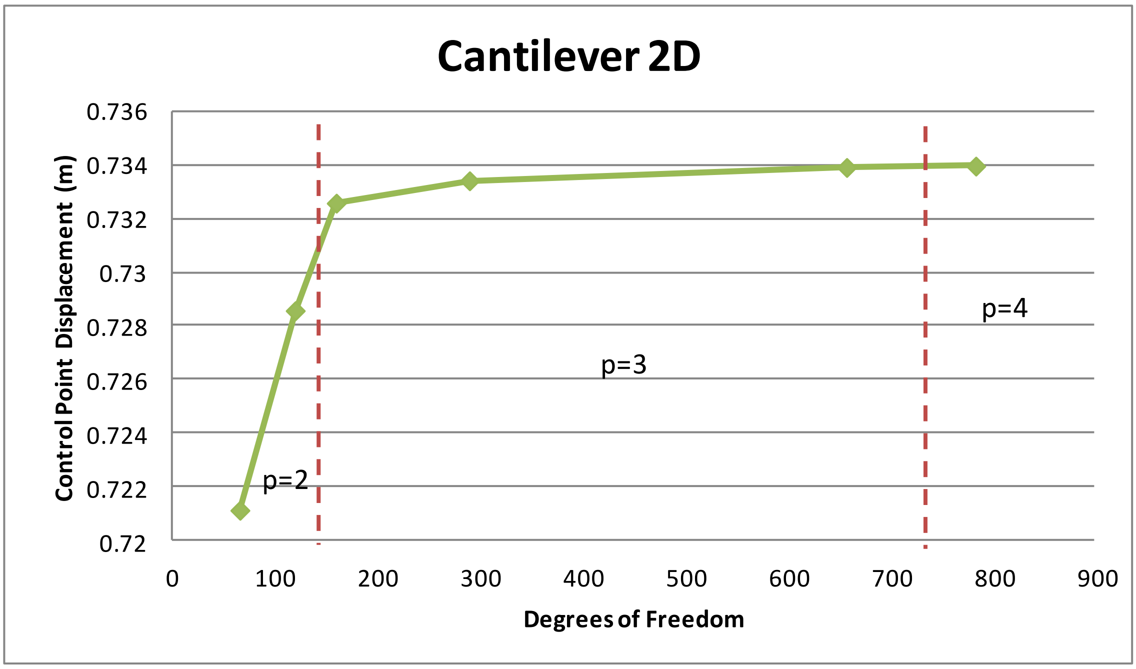


Figure 6.10. Control Point Displacement in comparison with total Degrees of Freedom. As soon as cubic B-SPLines are used, results are surprisingly good.

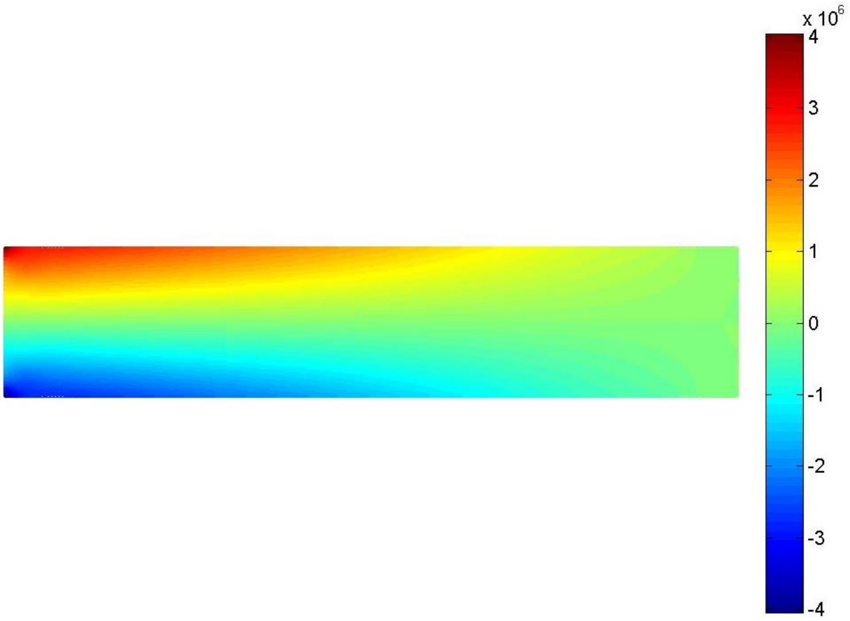
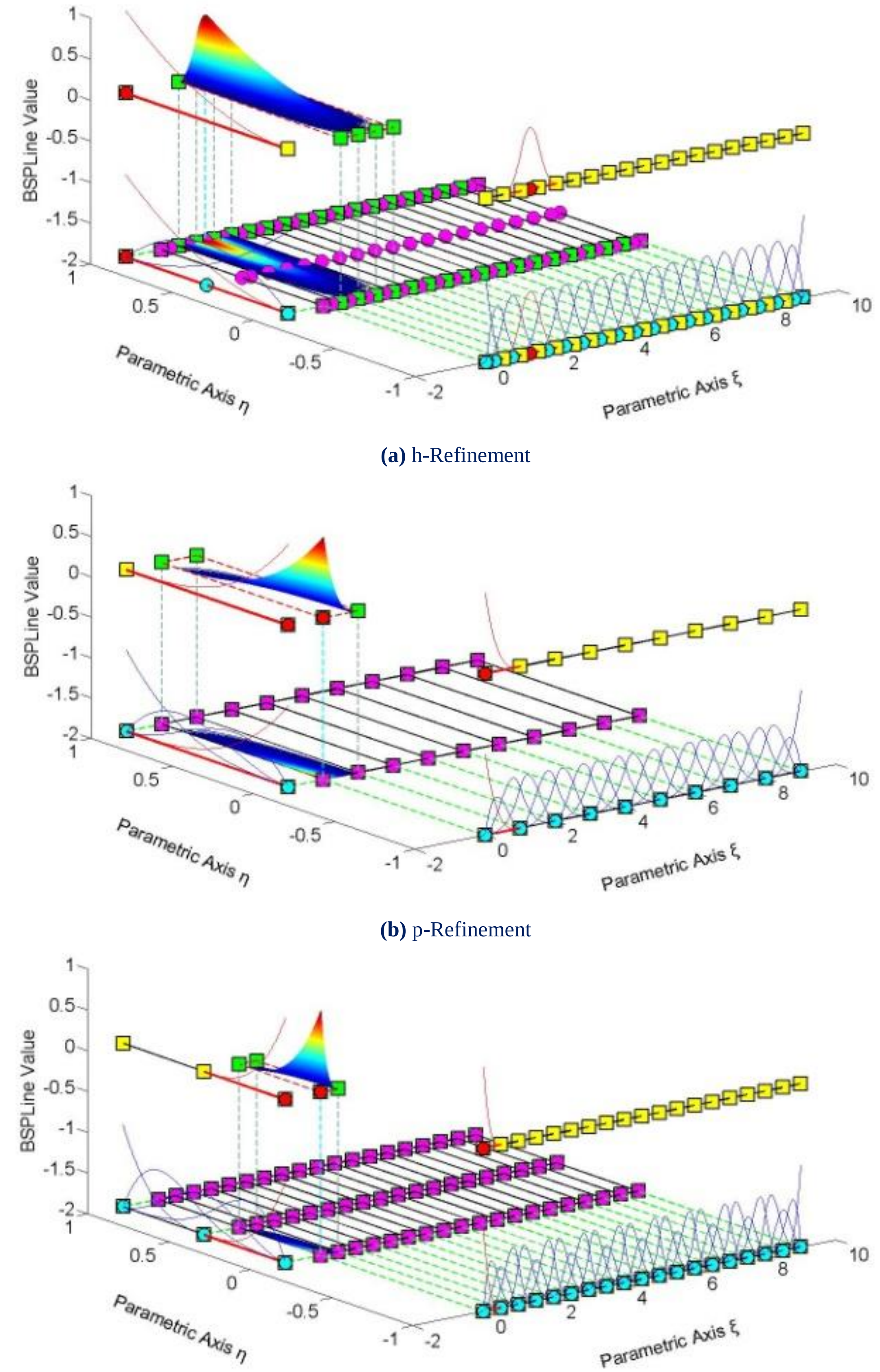


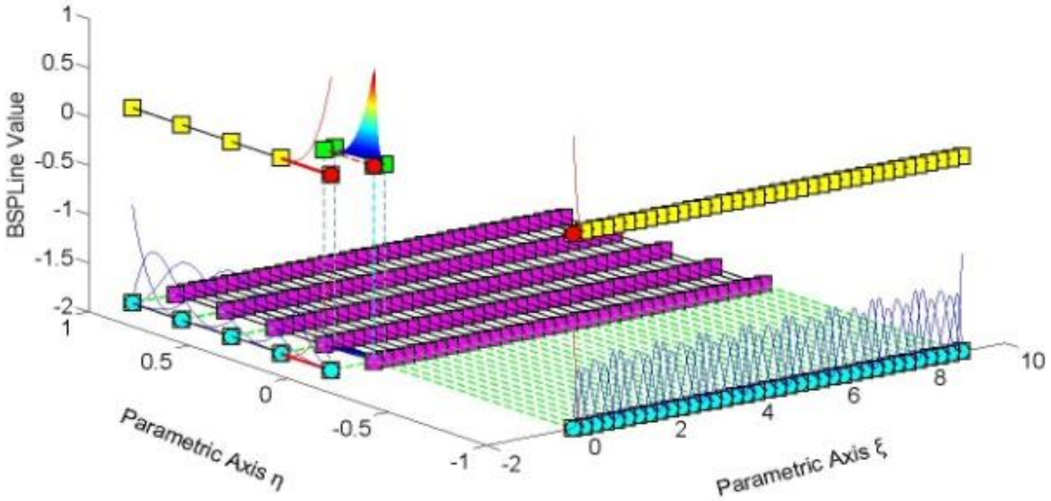
Figure 6.11. Contour for σ_{xx} after the final k-Refinement.



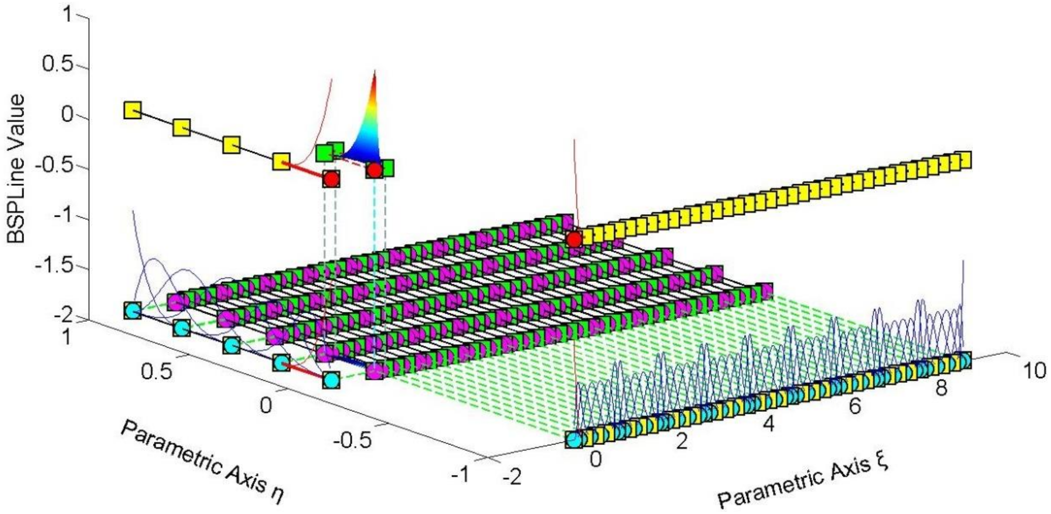
(a) h-Refinement

(b) p-Refinement

(c) k-Refinement (p=3)



(d) k-Refinement (p=3)



(e) k-Refinement (p=3)

Figure 6.12. B-SPLine Shape functions for the Refinements.

6.2 Cook’s Cantilever

Cook’s Cantilever is a characteristic plane stress problem. Both Cook’s Cantilever and the Simple Cantilever were represented using the exact same parameterization. Only the Cartesian coordinates of the Control Points differentiate.

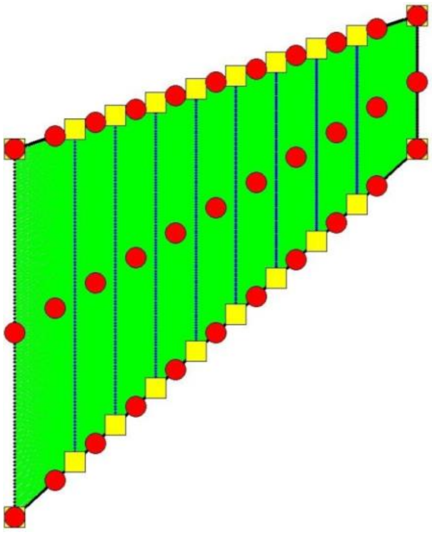
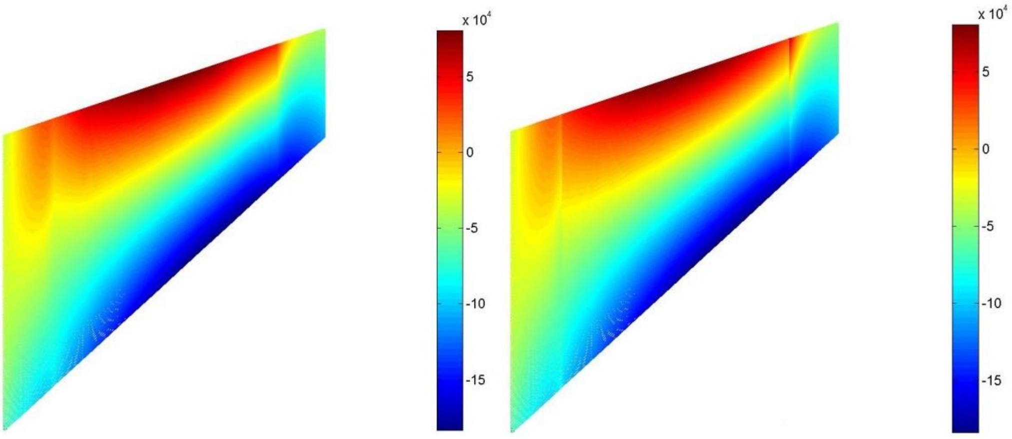


Figure 6.13. Cook’s Cantilever. Coarse mesh representation.

Analysis has been conducted with Shape Functions of C^1 and C^0 continuity. The results are displayed in Figure 6.14, where the discontinuity of the stress field is remarkably obvious for the C^0 representation.



(a) C^1 Continuity.

(b) C^0 Continuity

Figure 6.14. Contour for σ_{xx} .

The same parametric investigation is conducted for Cook’s Cantilever as well. It consists of an h-Refinement, a p-Refinement and three k-Refinements. The displacement results for a characteristic Control Point are displayed in Figure 6.15.

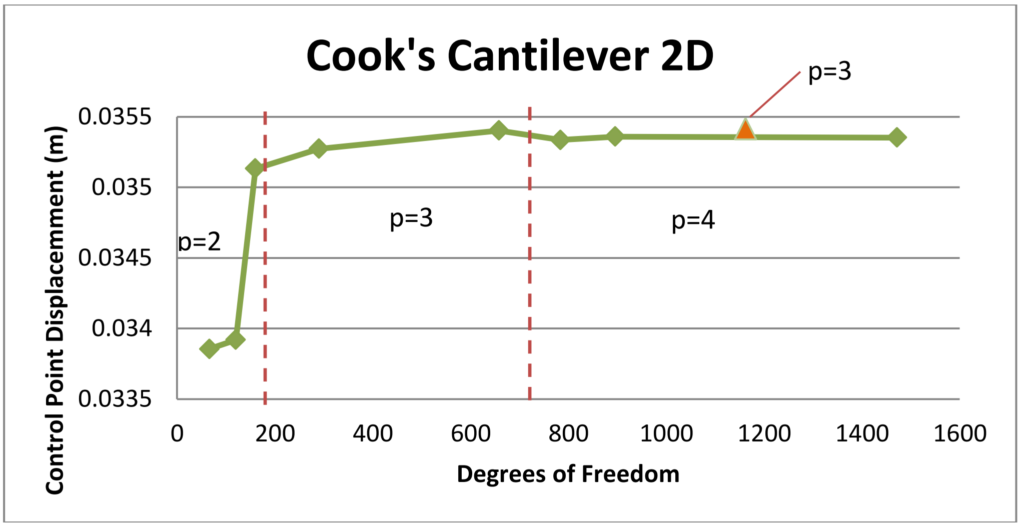
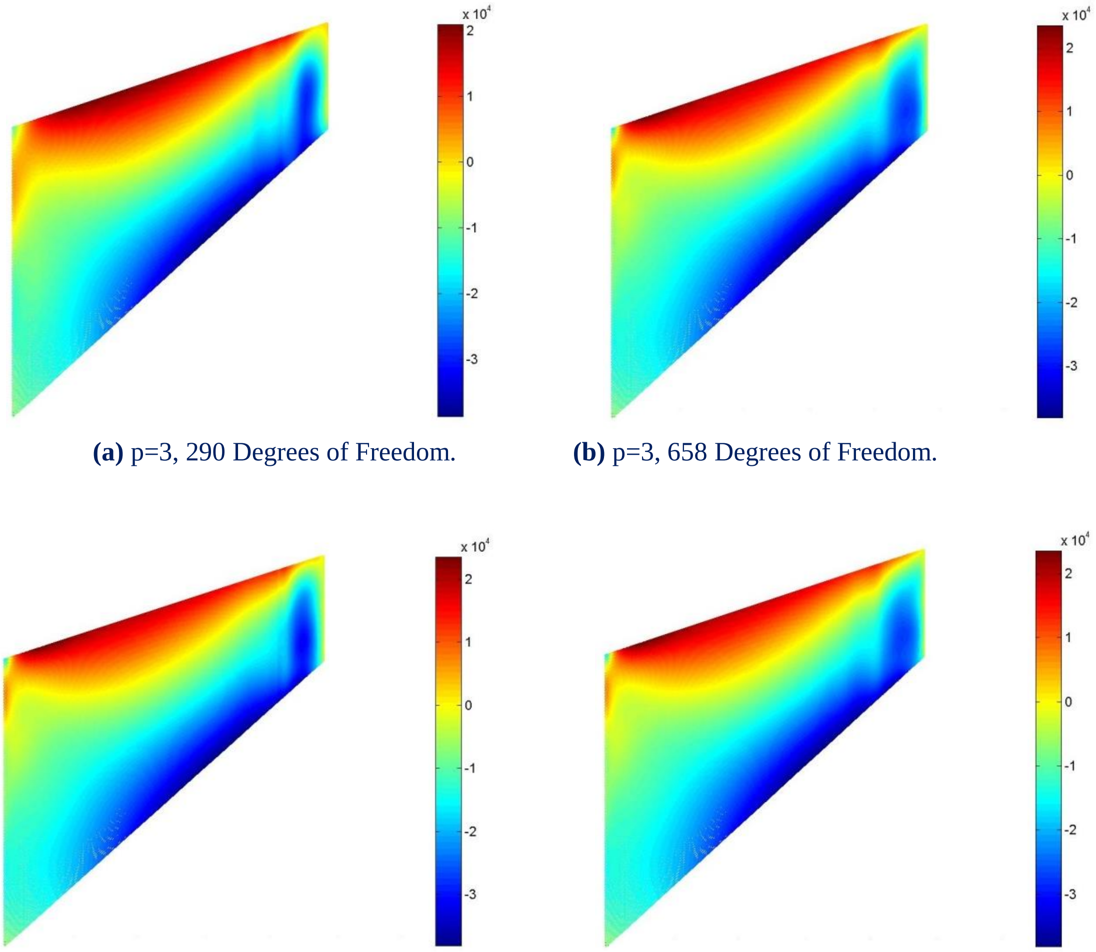


Figure 6.15. Control Point Displacement in comparison with total Degrees of Freedom.

Note that cubic basis functions produce better results than quartic basis functions. A divergence of 0.04% is achieved with 658 degrees of freedom, whereas quartic basis functions reach the exact solution at a much slower rate, as 1472 degrees of freedom are required in order for displacement norm of 0,18 % to be achieved.



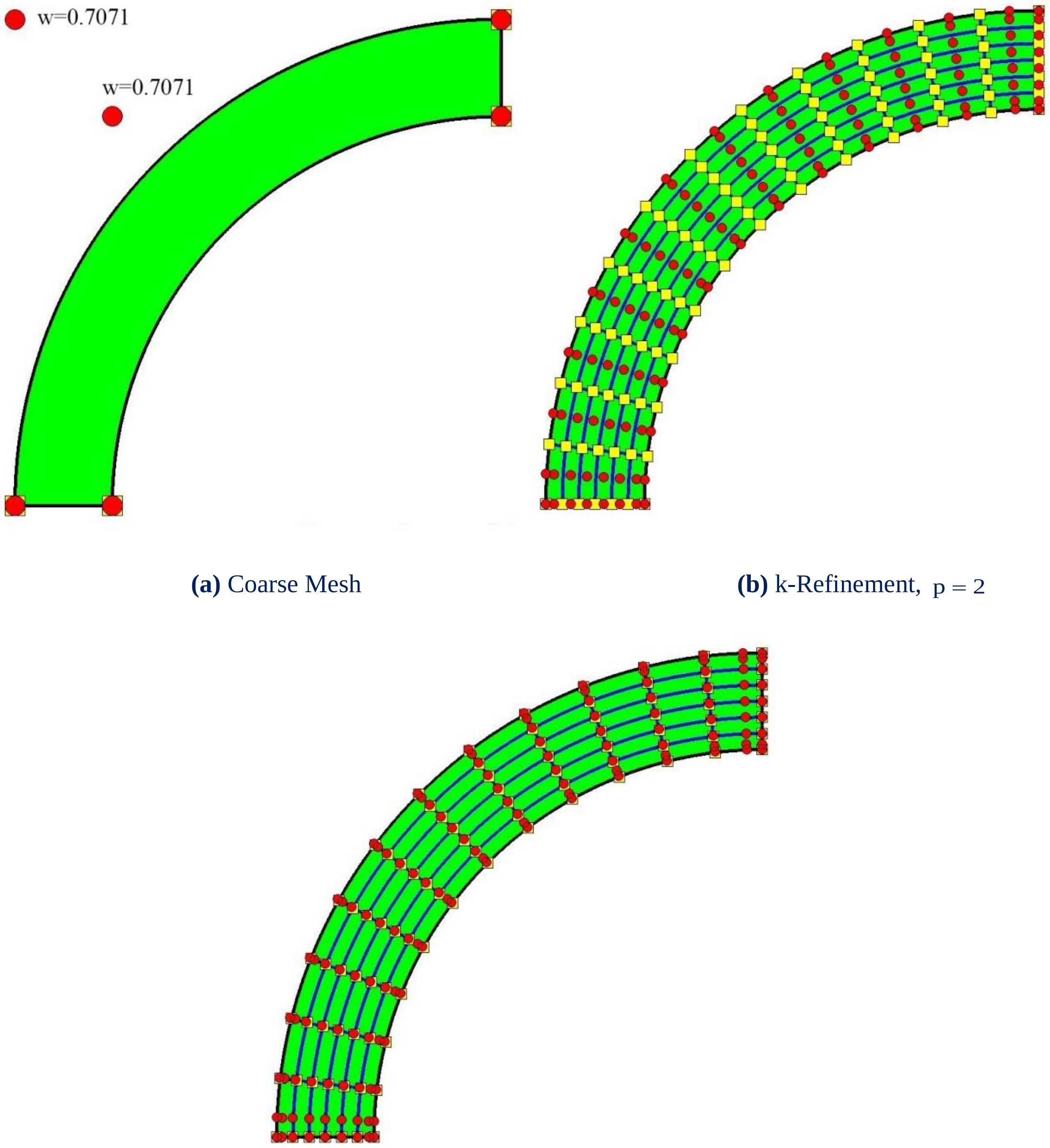
(a) p=3, 290 Degrees of Freedom. **(b)** p=3, 658 Degrees of Freedom.

(c) p=4, 784 Degrees of Freedom. **(d)** p=3, 1162 Degrees of Freedom.

Figure 6.16. τ_{xy} Contour for consecutive k-Refinements.

6.3 NURBS Arc

The next model is a cantilever of quadrant shape. The circular shape is represented by NURBS entities of different weights. Coarse mesh representation and the corresponding weight values are displayed in Figure 6.17.



(a) Coarse Mesh

(b) k-Refinement, $p = 2$

(c) k-Refinement, $p = 3$

Figure 6.17. Geometrical representations for NURBS Arc

Parametric investigation with different polynomial orders has been conducted. For each order, 3 k-Refinements were applied. The results are displayed in Figure 6.18.

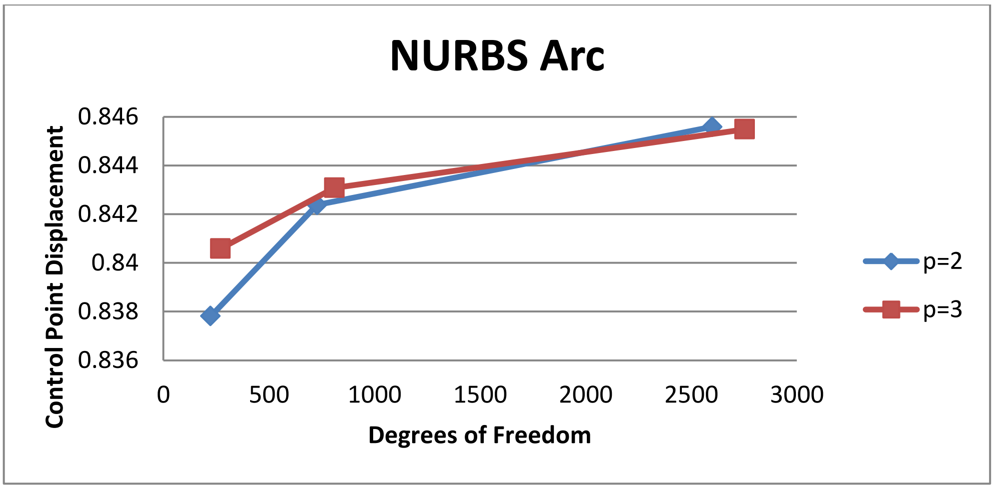
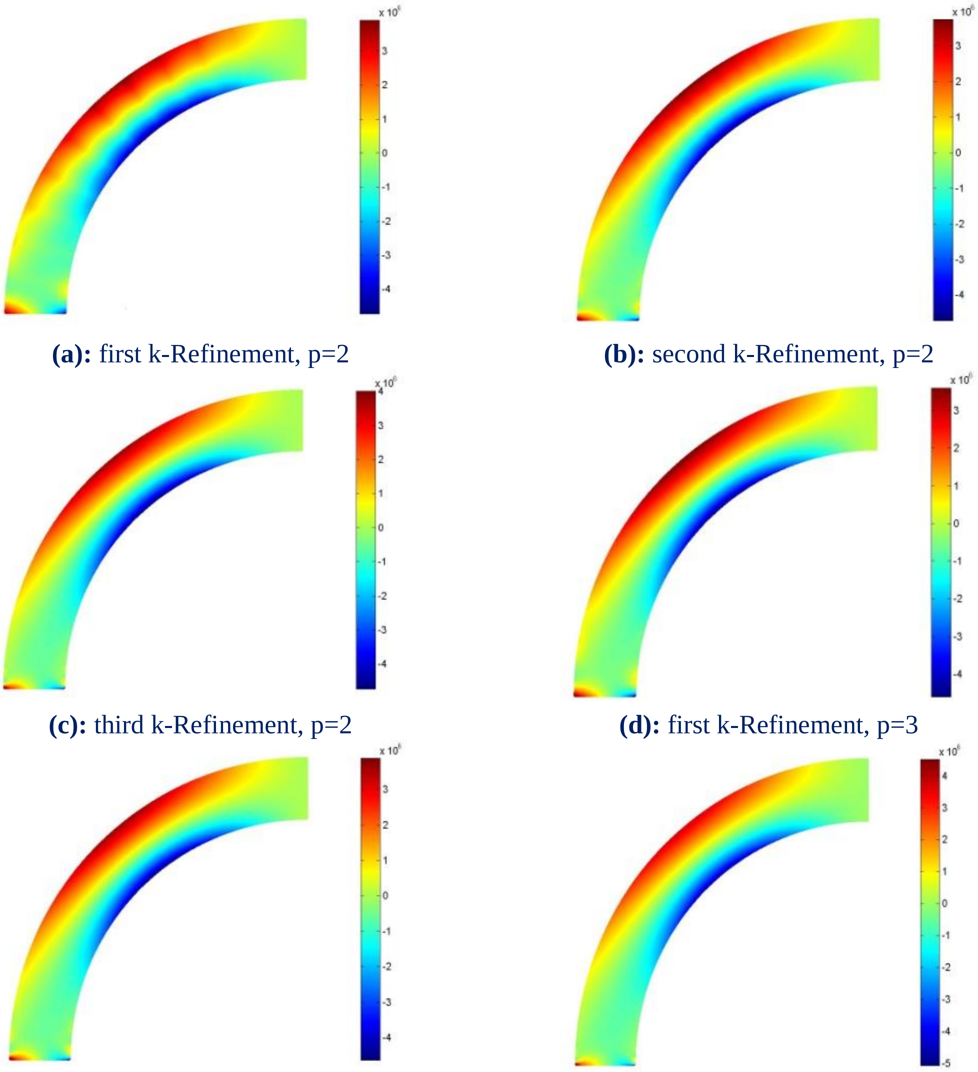


Figure 6.18. Results for NURBS Arc.



(a): first k-Refinement, p=2

(b): second k-Refinement, p=2

(c): third k-Refinement, p=2

(d): first k-Refinement, p=3

(e): second k-Refinement, p=3

(f): third k-Refinement, p=3

Figure 6.19. σ_{xx} Contour for NURBS Arc.

6.4 Plate with a hole

This example was first presented in terms of Isogeometric Analysis [1]. It has become a characteristic example of the potential of this innovative method.

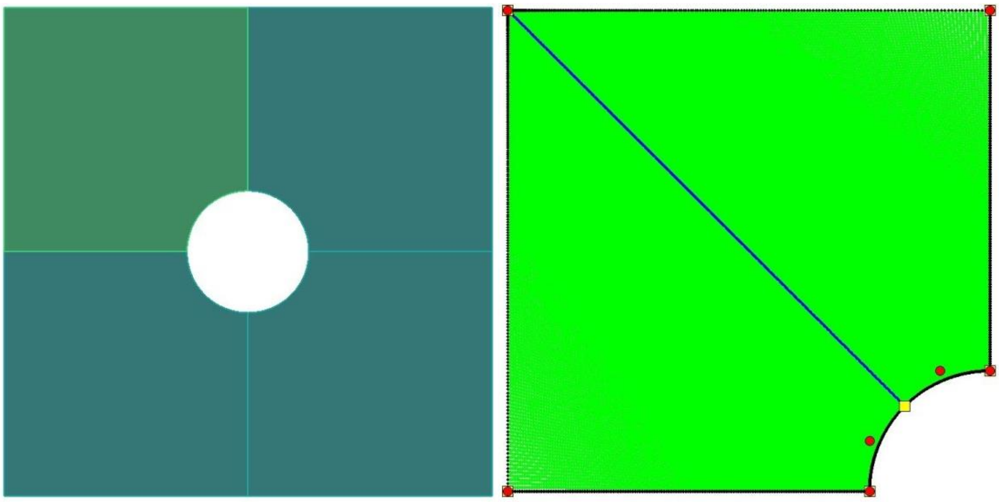
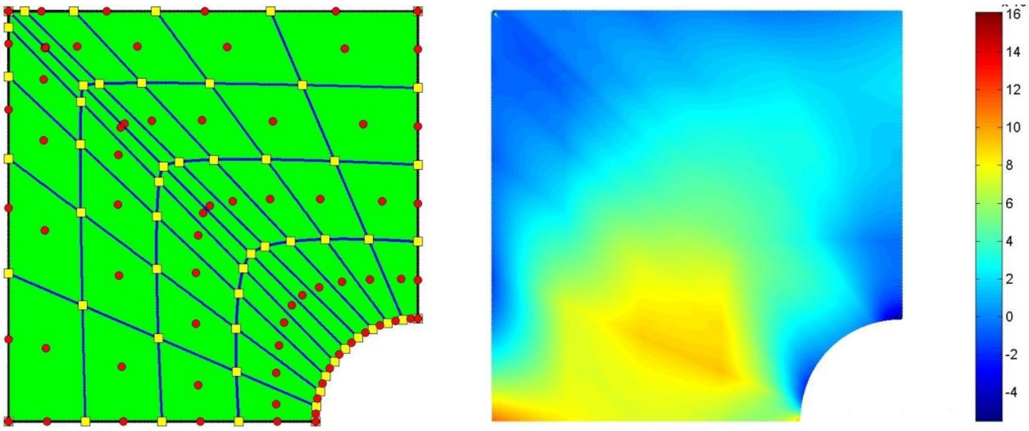


Figure 6.20. Plate with a hole and geometrical representation of a quadrant.

Due to double symmetry, only a quarter of the plate needs to be analyzed. The upper left corner is generated by a double control point at that position. The minimum degree for the circular quadrant and the outer side (parametric direction ξ) is $p = 2$.

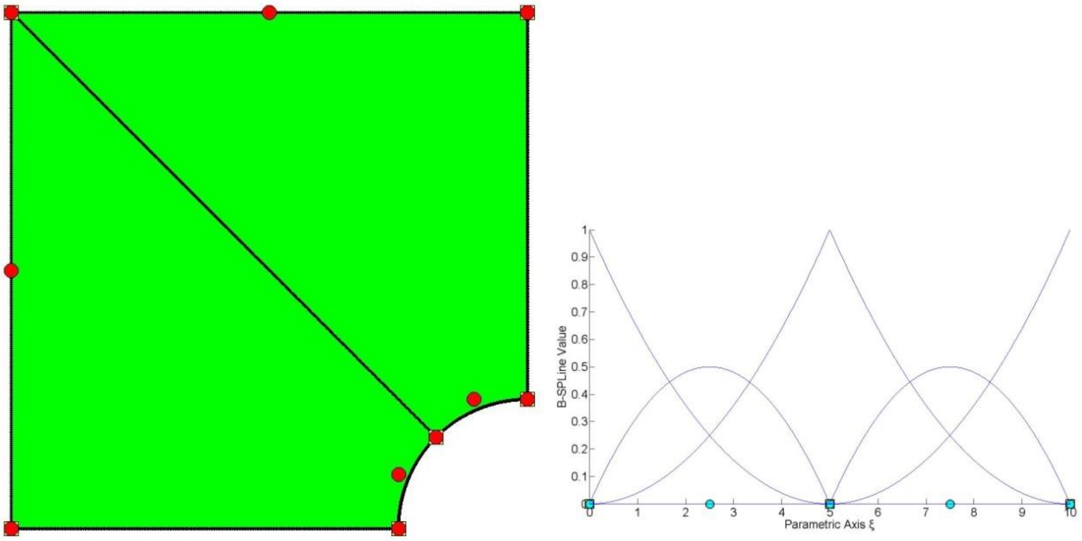
The double control point is a point of singularity, as two points from parameter space coincide into one in the physical space. This does not affect the formulation of the Stiffness Matrix, because numerical integration does not include data from this specific point. It does affect, however, the stress results. Figure 6.21 represents a fine mesh and the corresponding stress contour. The point of singularity has been removed, as τ_{xy} was infinite there. Still, in the small area close to the point, stress values are unusually high.



(a) **(b)**

Figure 6.21. **(a)** Fine Mesh representation and **(b)** τ_{xy} contour

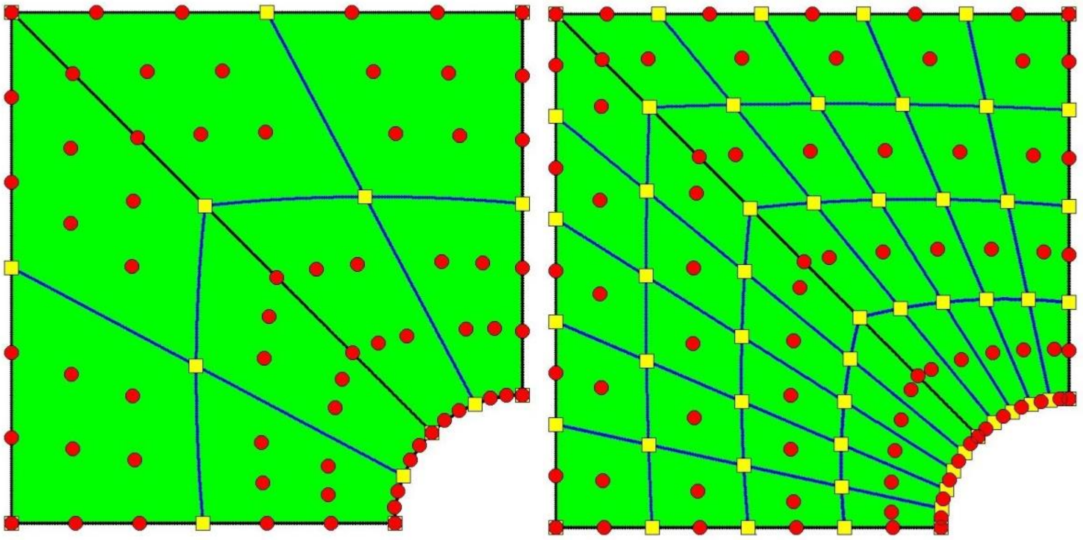
A different approach for the representation of the corner is to consider a linear B-SPLine curve and then perform p-Refinement in order to reach the desired quadratic basis functions. This method contains no points of singularity, but a patch boundary in the middle of parametric direction ξ is necessary.



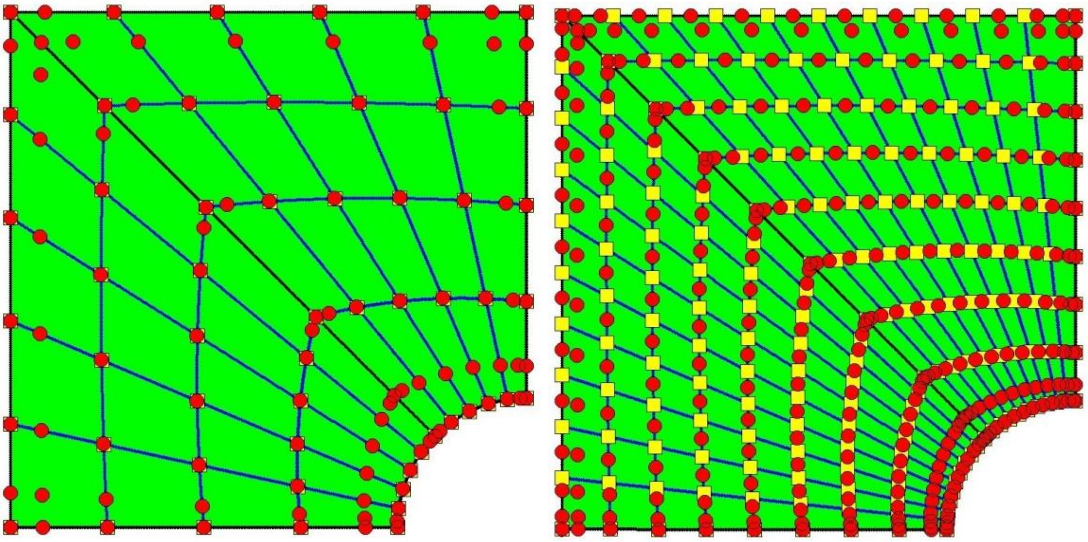
(a) **(b)**

Figure 6.22. (a) New geometrical representation and **(b)** basis functions.

A number of Refinements with the corresponding solutions were applied. The fine meshes and results are displayed below:



(a) p-Refinement **(b)** h-Refinement



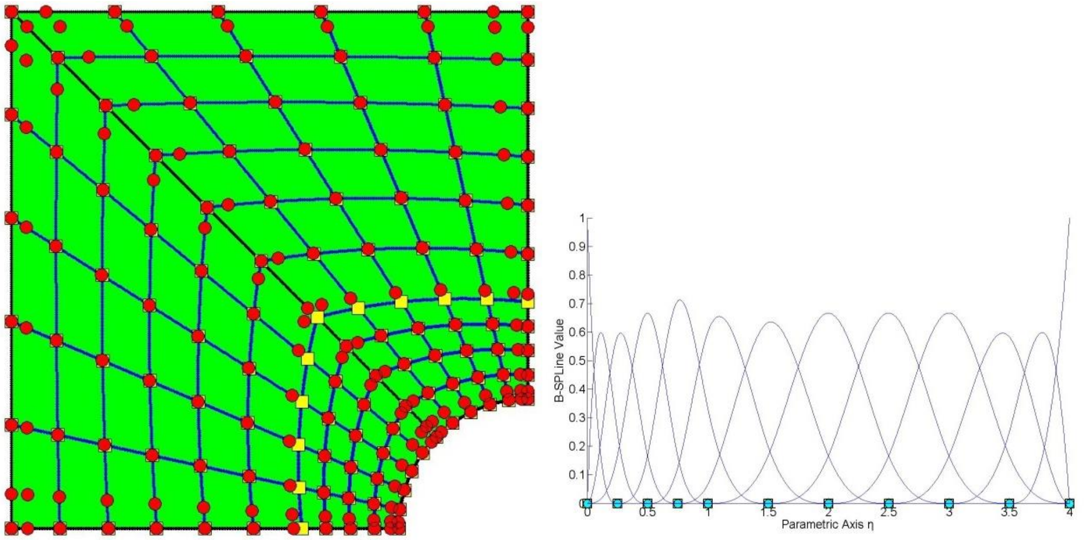
(c) k-Refinement with p=3

(d) k-Refinement with p=4

Figure 6.23. Fine meshes for Refinement.

All these meshes provide exact geometrical representation. However, certain problems may arise in analysis. The meshing leads to quadrilateral elements with sides of different size. As it is known from FEM, this leads to a poor approximation of the solution field.

The engineer plays a decisive role in Refinement. It is up to him to choose the correct Refinement techniques in order to obtain accurate results. The mesh in Figure 6.24 provides a much better approximation than all the meshes of Figure 6.23.



(a)

(b)

Figure 6.24. (a) Mesh for suitable k-Refinement and
(b) basis functions on η

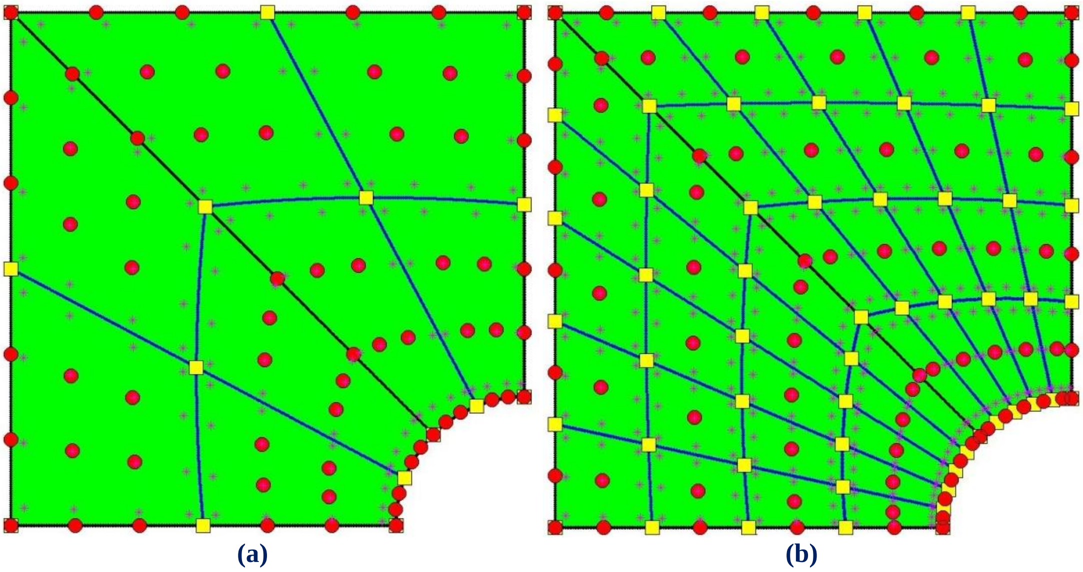


Figure 6.25. Gauss points. Physical coordinates for each mesh.

(a) p-Refinement ($4 \cdot 4 = 16$ Gauss points per Knot Rectangle)

(b) h-Refinement ($3 \cdot 3 = 9$ Gauss points per Knot Rectangle)

Results are displayed in Figure 6.26. When distorted elements are used, the results are less accurate. The final, suitable k-Refinement approximates at most the exact solution.

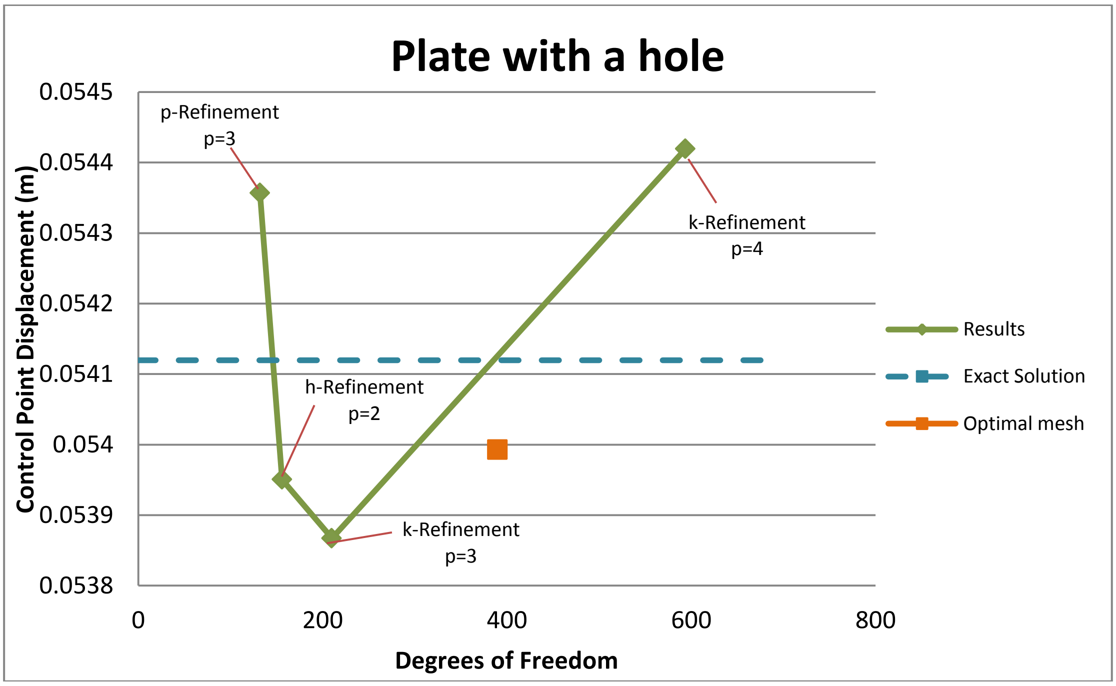
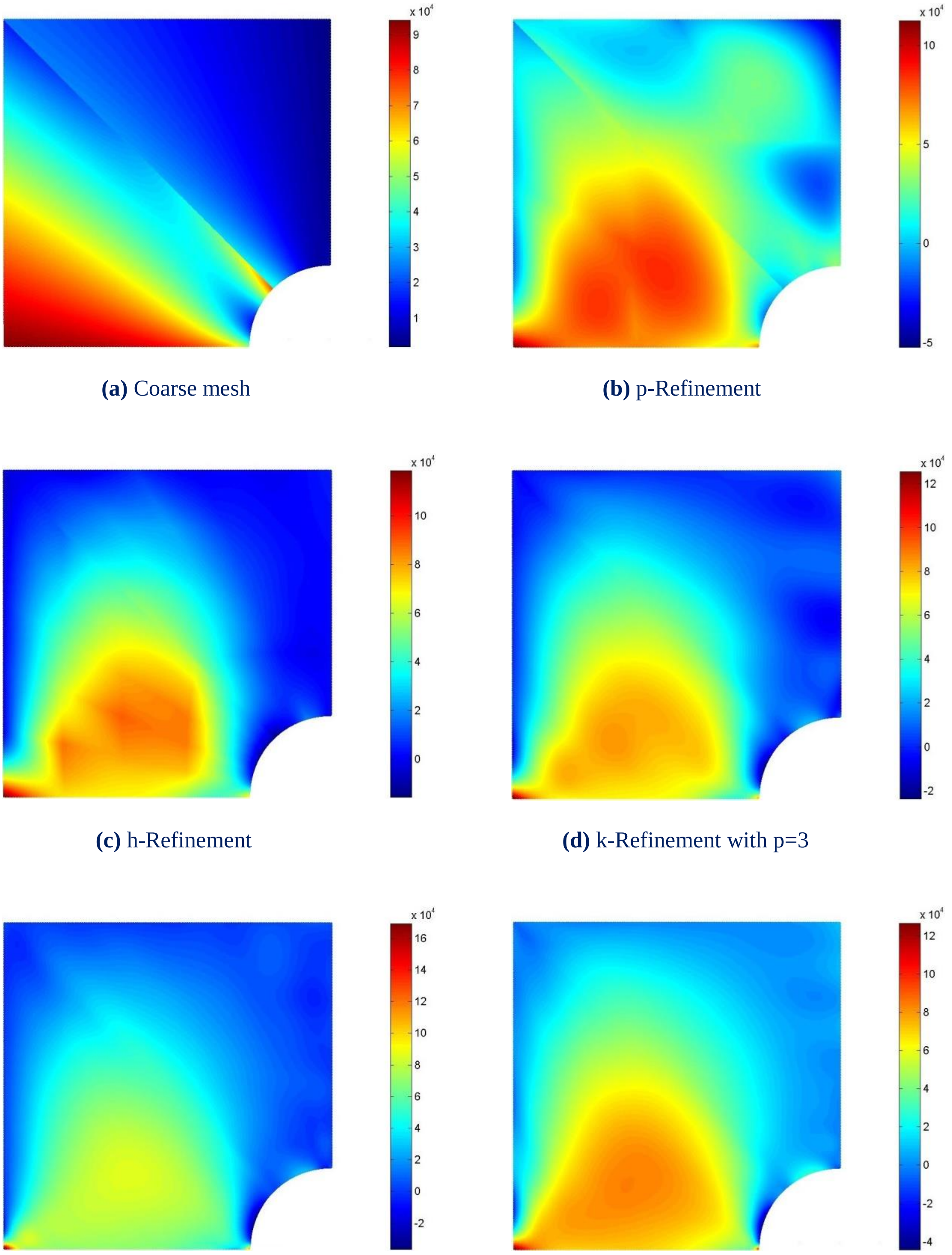


Figure 6.26. Results for plate with a hole.
Representations with elements of irregular shape
diverge from the exact solution when polynomial order increases.

Contour results are displayed for all the representations.



(a) Coarse mesh

(b) p-Refinement

(c) h-Refinement

(d) k-Refinement with p=3

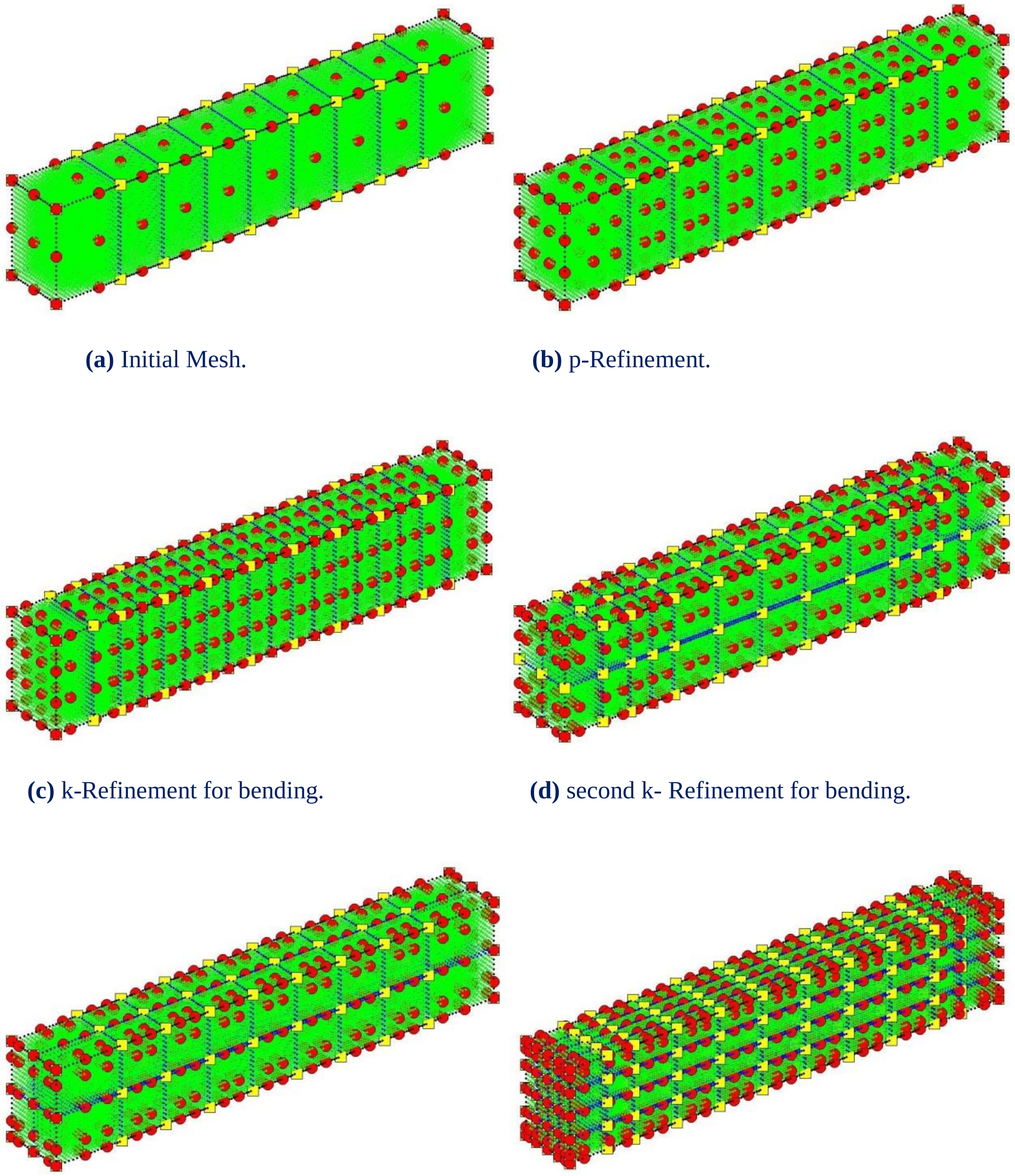
(e) k-Refinement with p=4

(f) Suitable k-Refinement with p=3

Figure 6.27. τ_{xy} Contour.

6.5 Cantilever 3D

The first 3D representation examined is a simple cantilever, with a rectangle cross-section. The initial mesh is similar to the 2D Cantilever. A parametric direction ζ with one Knot Span is added. The Cantilever is subjected to bending and torsion loads. Suitable Refinements are applied for each case.



(a) Initial Mesh.

(b) p-Refinement.

(c) k-Refinement for bending.

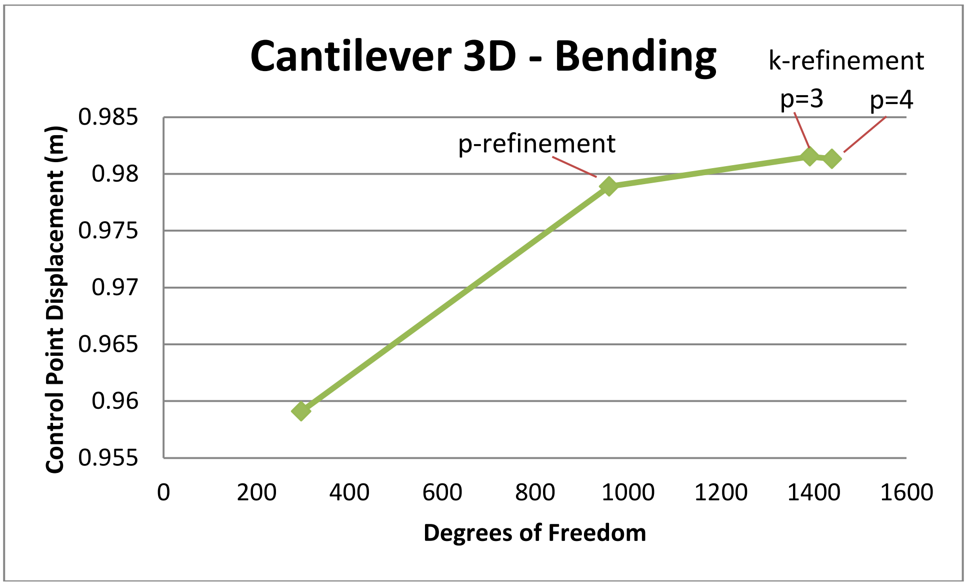
(d) second k- Refinement for bending.

(e) k-Refinement for torsion.

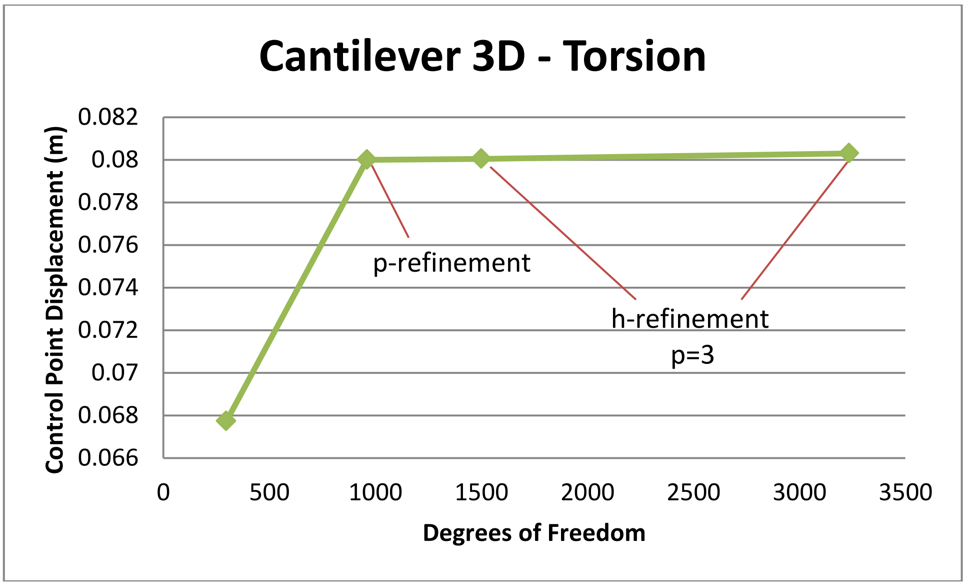
(f) second k-Refinement for torsion.

Figure 6.28. Initial Mesh and Refinements.

Results for bending and torsion are presented below. As parameterization becomes more detailed, the structure becomes more flexible and is able to represent the natural problem more accurately. The final k-Refinement for Bending diverges from the path to exact solution, because, as mentioned before, quartic basis functions are less efficient than cubic ones. The final k-Refinement for bending, containing cubic basis functions and 3.234 degrees of freedom, forms a very accurate solution to the problem.

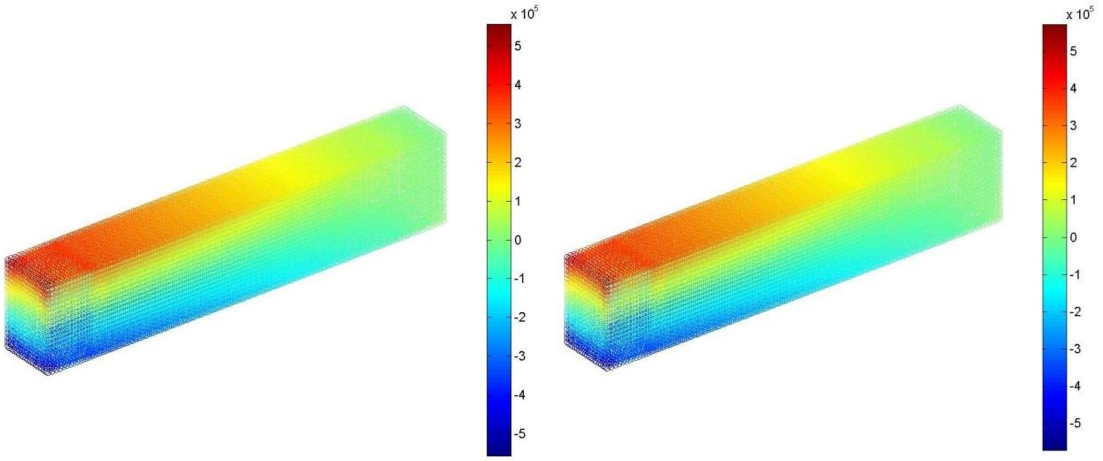


(a) Bending



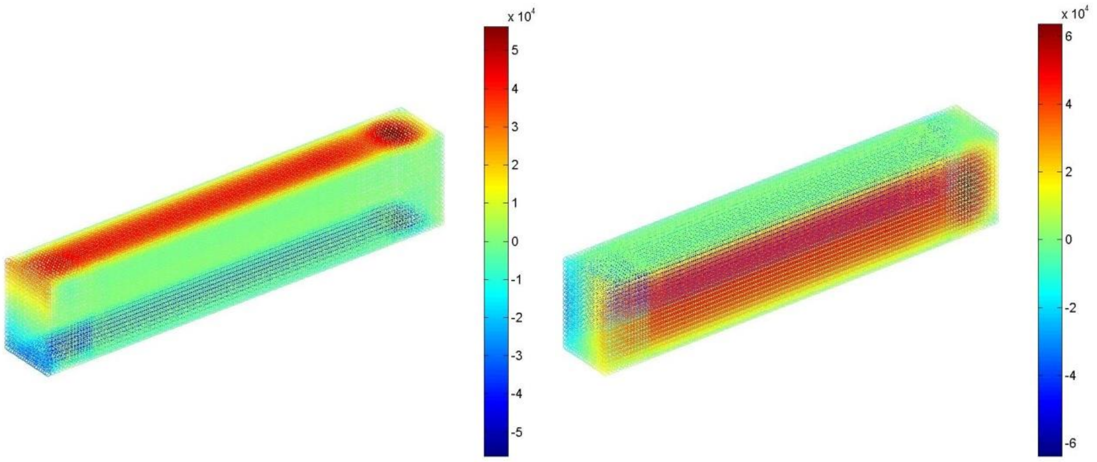
(b) Torsion

Figure 6.29. Control Point Displacement in comparison with total Degrees of Freedom, for Bending and Torsion.



(a): k-Refinement with $p = 3$. **(b):** k-Refinement with $p = 4$.

Figure 6.30. σ_{xx} Contour for Bending.

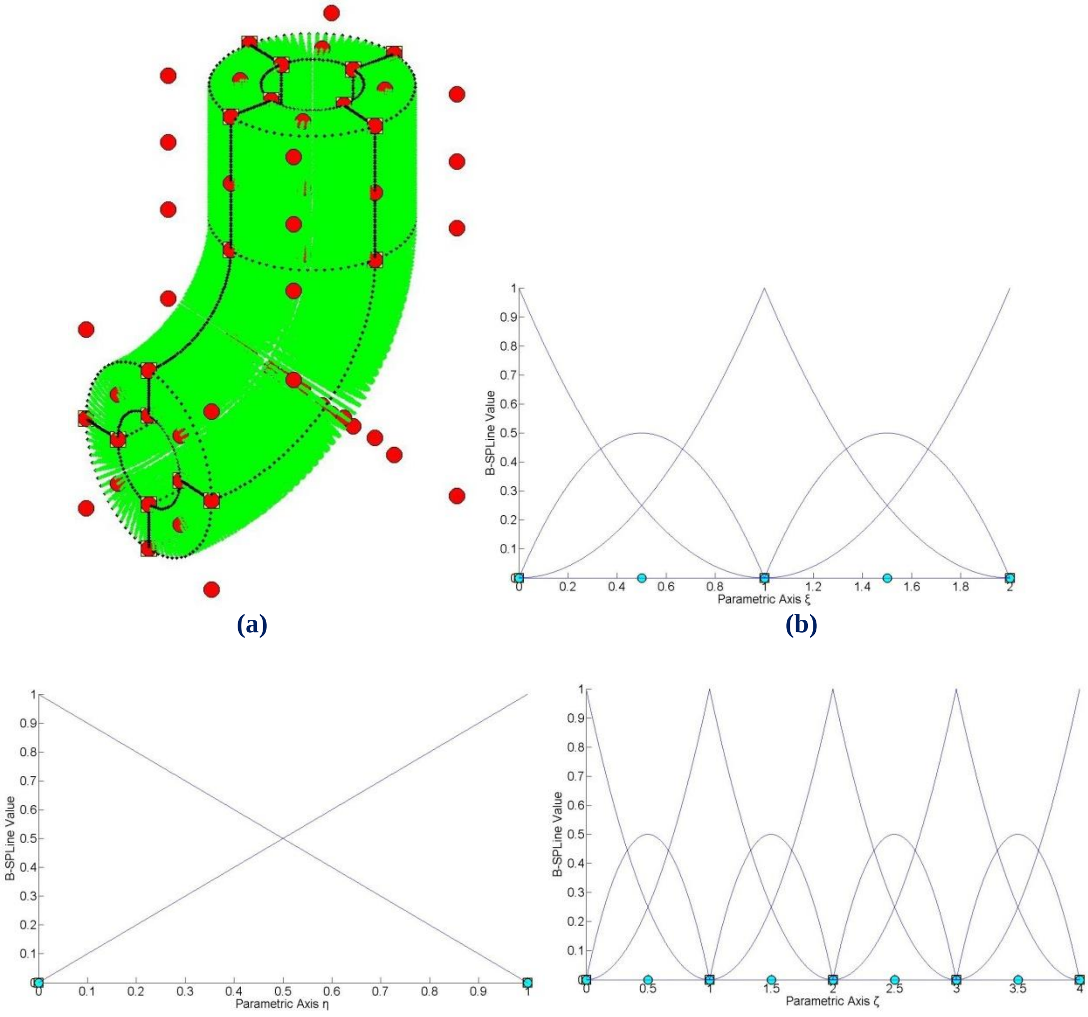


(a): τ_{xy} Contour. **(b):** τ_{zx} Contour.

Figure 6.31. Shear Stress Contour for Torsion,
for the final k-Refinement.

6.6 Bent Pipe

The bent pipe was first presented by [1]. It consists of a straight and a circular section, joined together. Parametric direction ξ represents the longitudinal direction of the pipe. Parametric direction η represents the pipe thickness and ζ the creation of the circular shape.



(a) (b)

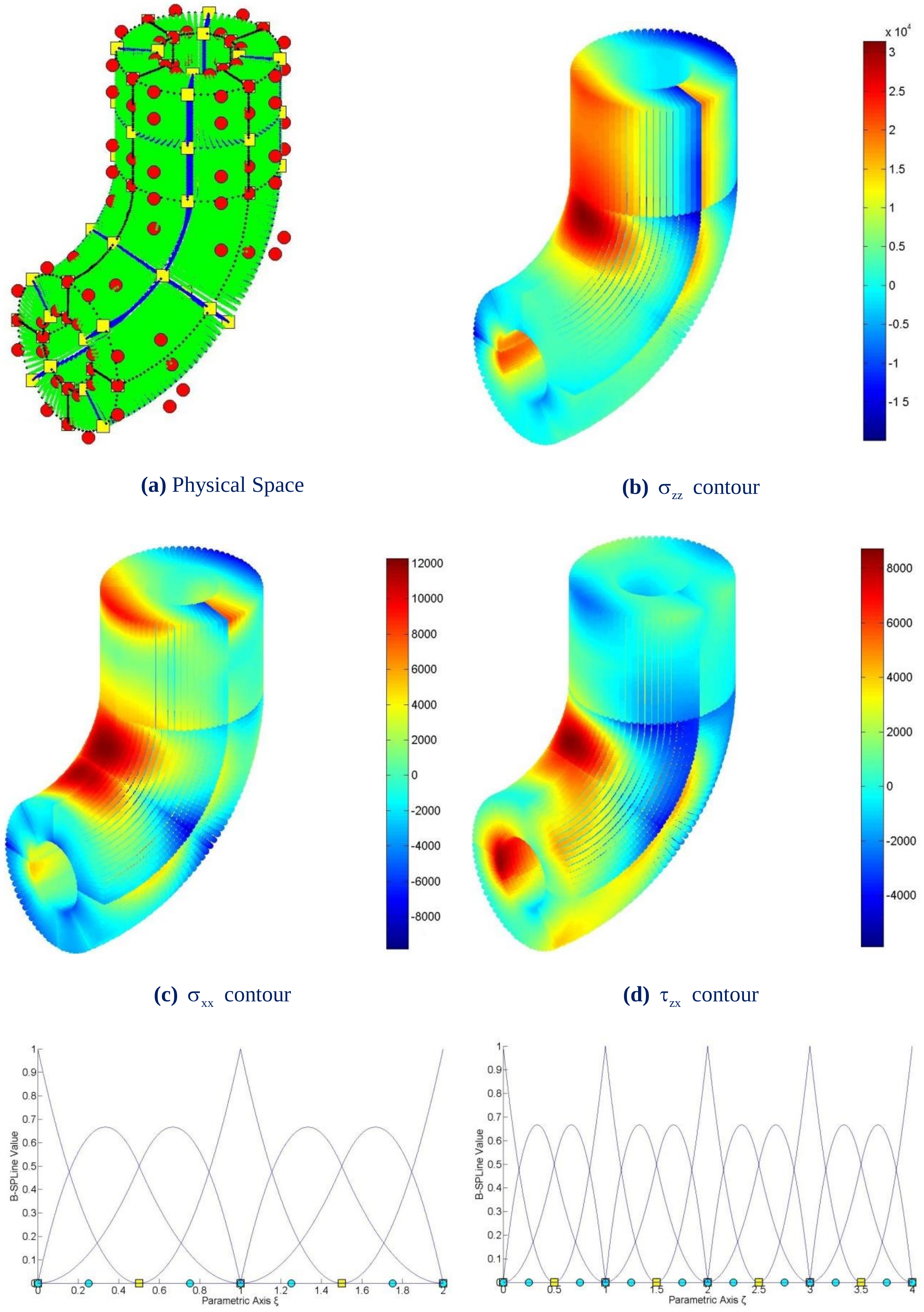
(c) (d)

Figure 6.32. (a) Physical Space and **(b), (c), (d)** basis functions for bent pipe.

There are four patches across ζ, as required for the representation of a circle, and one patch across η, necessary for the transition from straight to circular section of the pipe.

This coarse mesh cannot produce accurate results. These are achieved, as usual, by Refinement. h- and k- Refinement are applied on the bent pipe. The corresponding basis functions on ξ, ζ are represented in Figures 6.33 (e),(f) and 6.34 (e),(f).

Refinements lead to increased continuity and element interconnectivity. The stress discontinuity across patch boundary on ξ is visible as well.



(a) Physical Space

(b) σ_{zz} contour

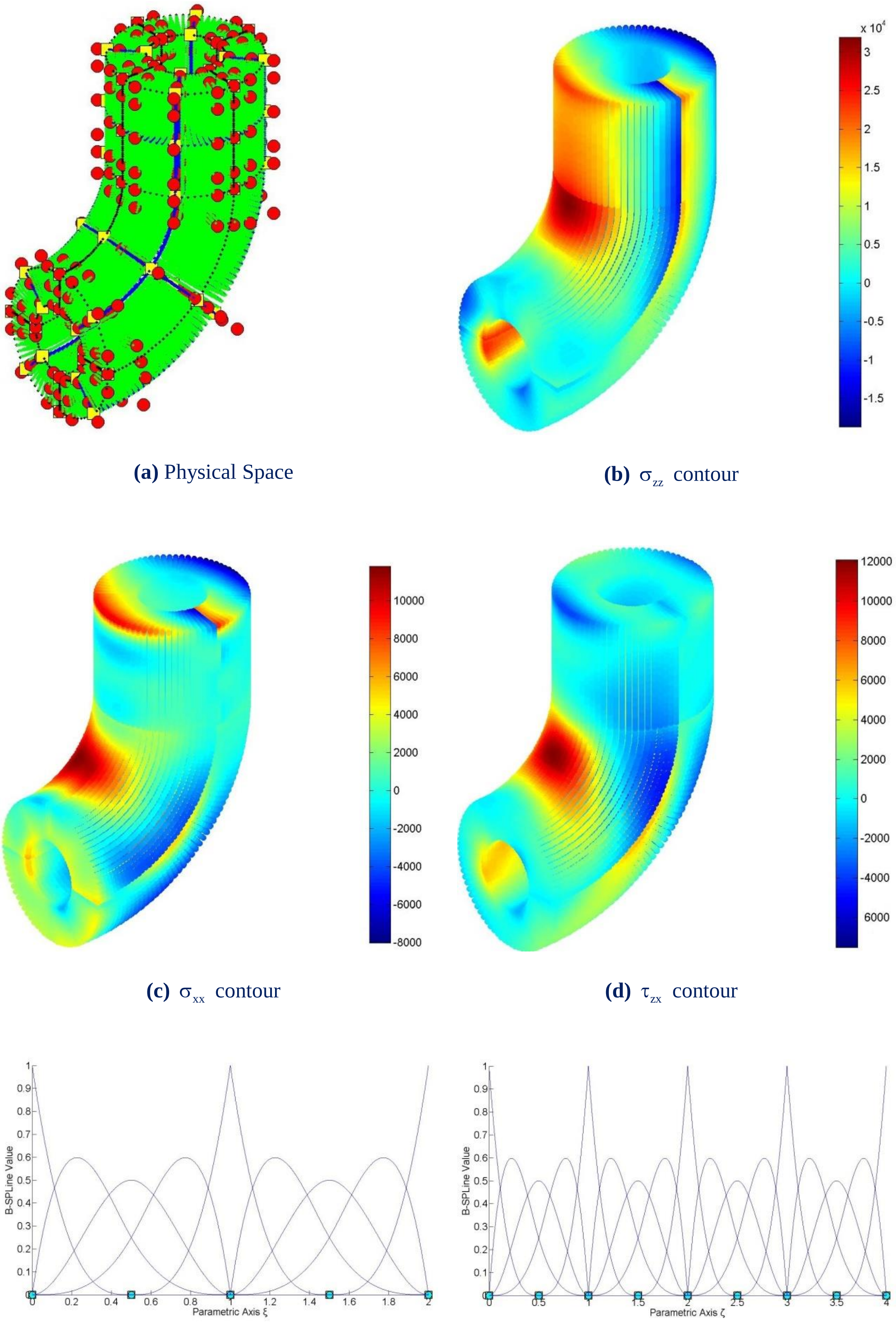
(c) σ_{xx} contour

(d) τ_{zx} contour

(e) B-SPLine basis function ξ

(f) B-SPLine basis function η

Figure 6.33. h-Refinement for Bent Pipe



(a) Physical Space

(b) σ_{zz} contour

(c) σ_{xx} contour

(d) τ_{zx} contour

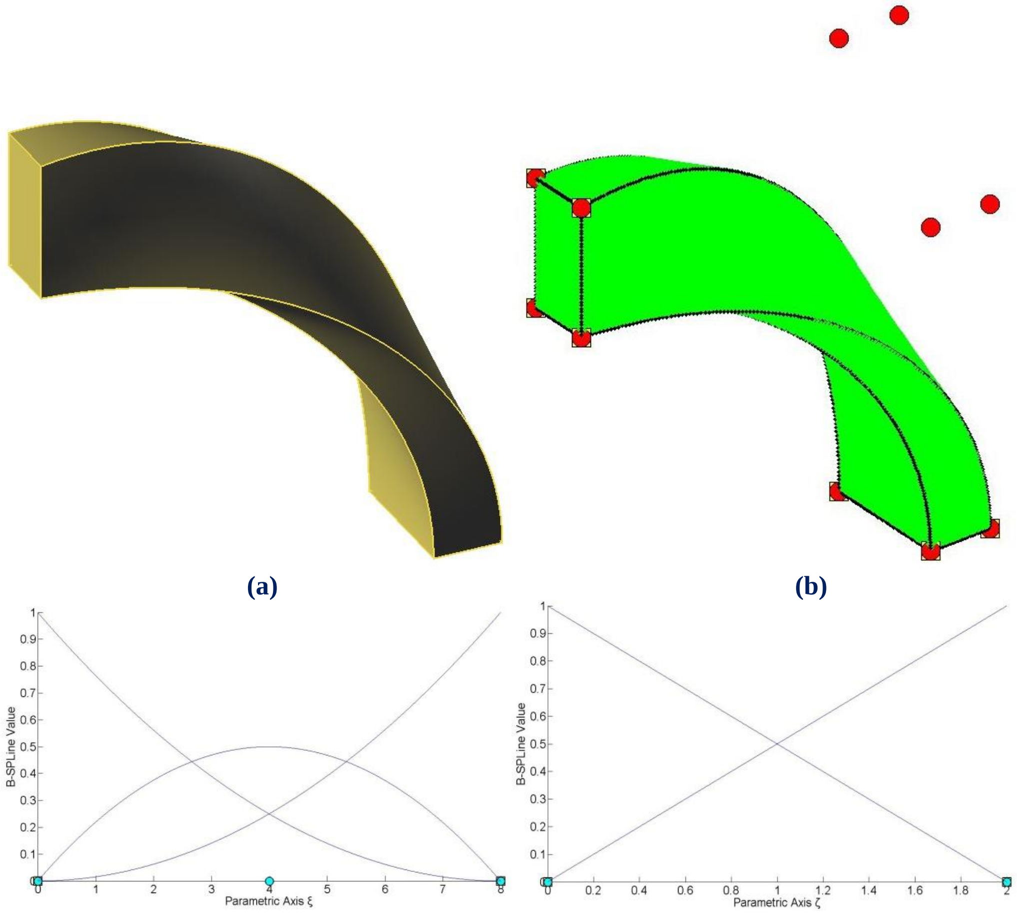
(e) B-SPLine basis function ξ

(f) B-SPLine basis function η

Figure 6.34. k-Refinement for Bent Pipe

6.7 NURBS Rod

The next application consists of an orthogonal cross-section swept across a quadrant and rotated by 90 at the other end. This bizarre shape makes it extremely difficult to mesh with classical Finite Element approach. On the other hand, only 12 Control Points are required for the NURBS representation. Quadratic basis functions are required for the longitudinal direction and linear for the other two. Coarse mesh representation and basis functions are displayed in the Figure below.

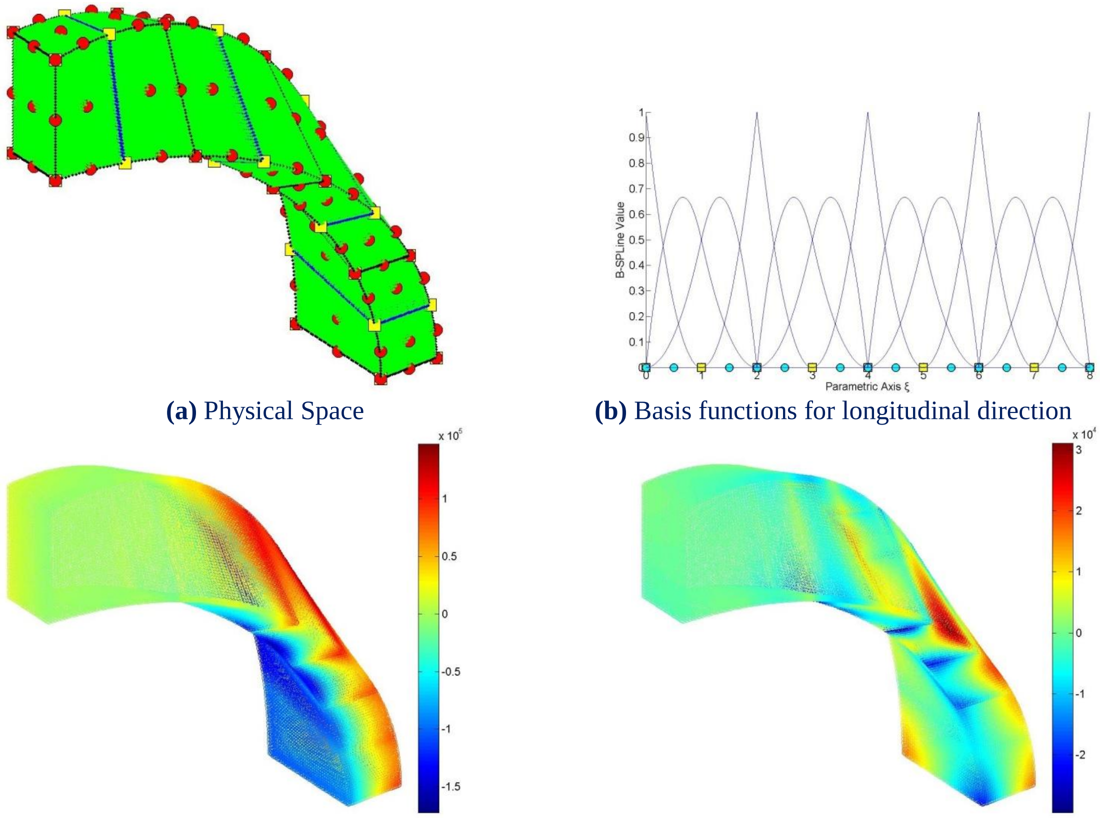


(a) **(b)**

(c) **(d)**

Figure 6.35. **(a)** Rhino Model **(b)** Coarse Mesh representation and **(c), (d)** basis functions on ξ, ζ.

Once the coarse mesh has been designed, Refinement and convergence to an accurate solution is relatively easy. The first fine mesh selected is represented by quadratic basis functions for all directions and 4 patches along ξ (longitudinal direction). C^0 Continuity across patch boundaries is reflected on the results. Shape function derivatives and, as a result, stress distribution are not Continuous at those positions. This is a major problem for finite elements, as C^{-1} is the maximum Continuity any element has. In Isogeometric analysis, the solution is simple. Patch boundaries and C^0 are used only when necessary. For the NURBS Rod application, a second Refinement is applied and basis functions with greater Continuity are selected. The new stress field is free of discontinuities.



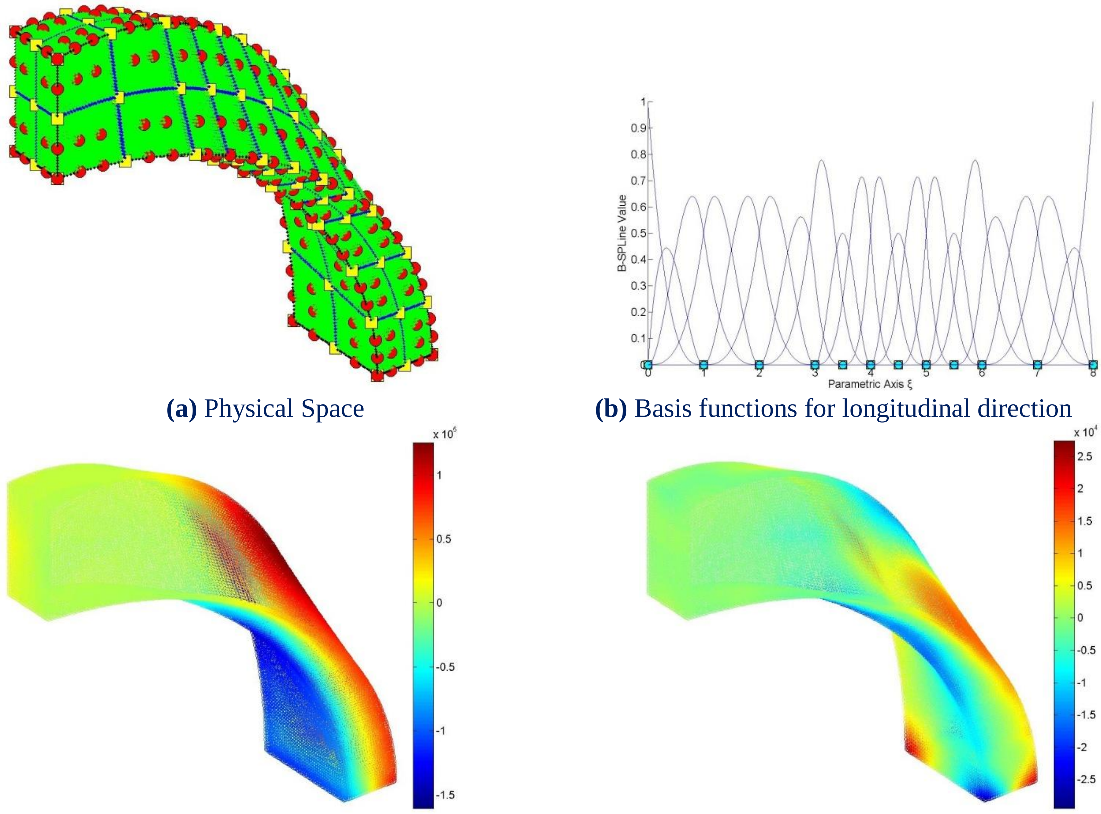
(a) Physical Space

(b) Basis functions for longitudinal direction

(c) Stress contour for σ_{zz}

(d) Stress contour for τ_{yz}

Figure 6.36. Patch representation.



(a) Physical Space

(b) Basis functions for longitudinal direction

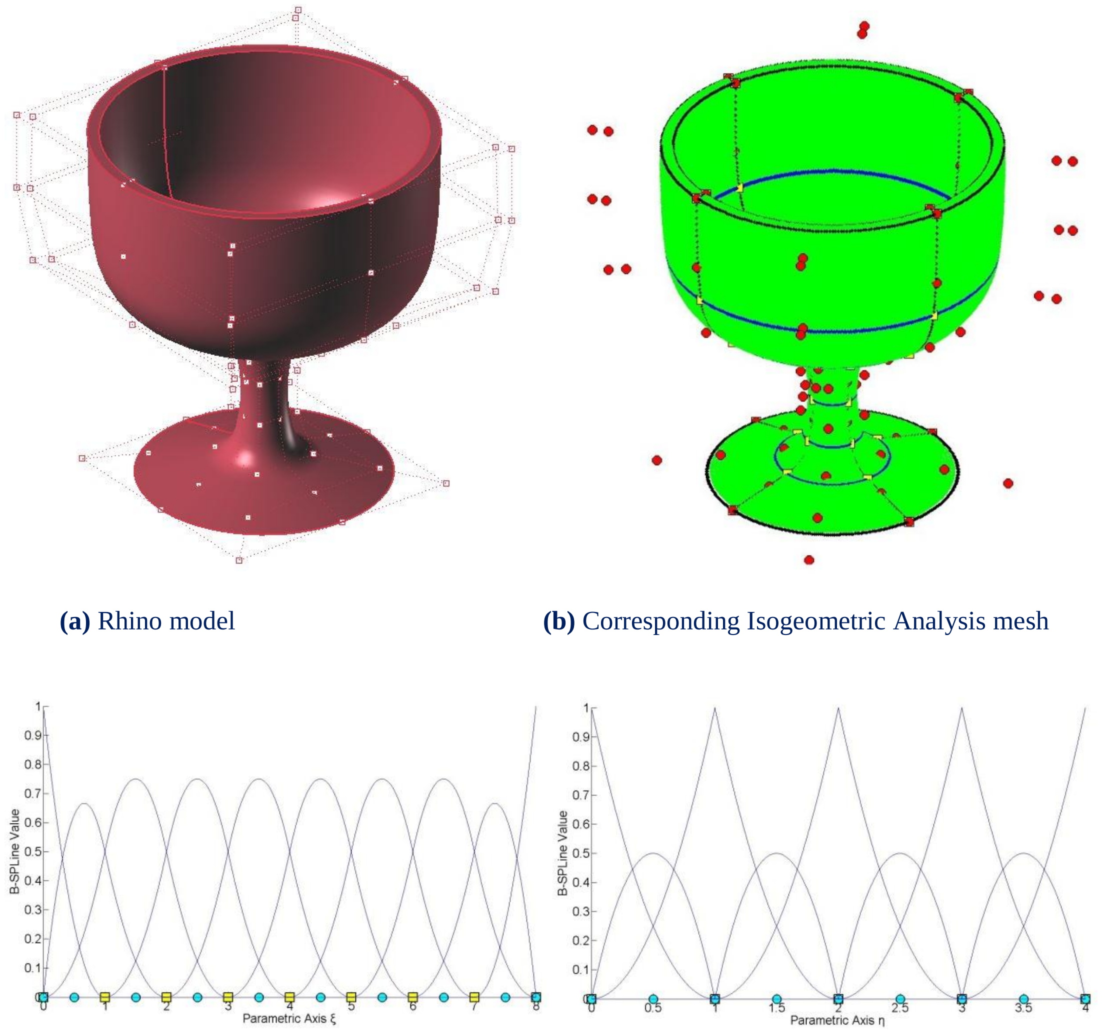
c) Stress contour for σ_{zz}

(d) Stress contour for τ_{yz}

Figure 6.37. Increased Continuity representation.

6.8 Wine Glass

The final application of this thesis is set to demonstrate the remarkable potential of Isogeometric Analysis. A wine glass is designed with the help of the platform Rhinoceros. Quadratic B-SPLines are used for the perpendicular and the circular directions, linear B-SPLines for the thickness. The geometrical data is transferred into the Isogeometric Analysis code. This is all it takes. The new model is ready for loads, boundary conditions and analysis.



(a) Rhino model

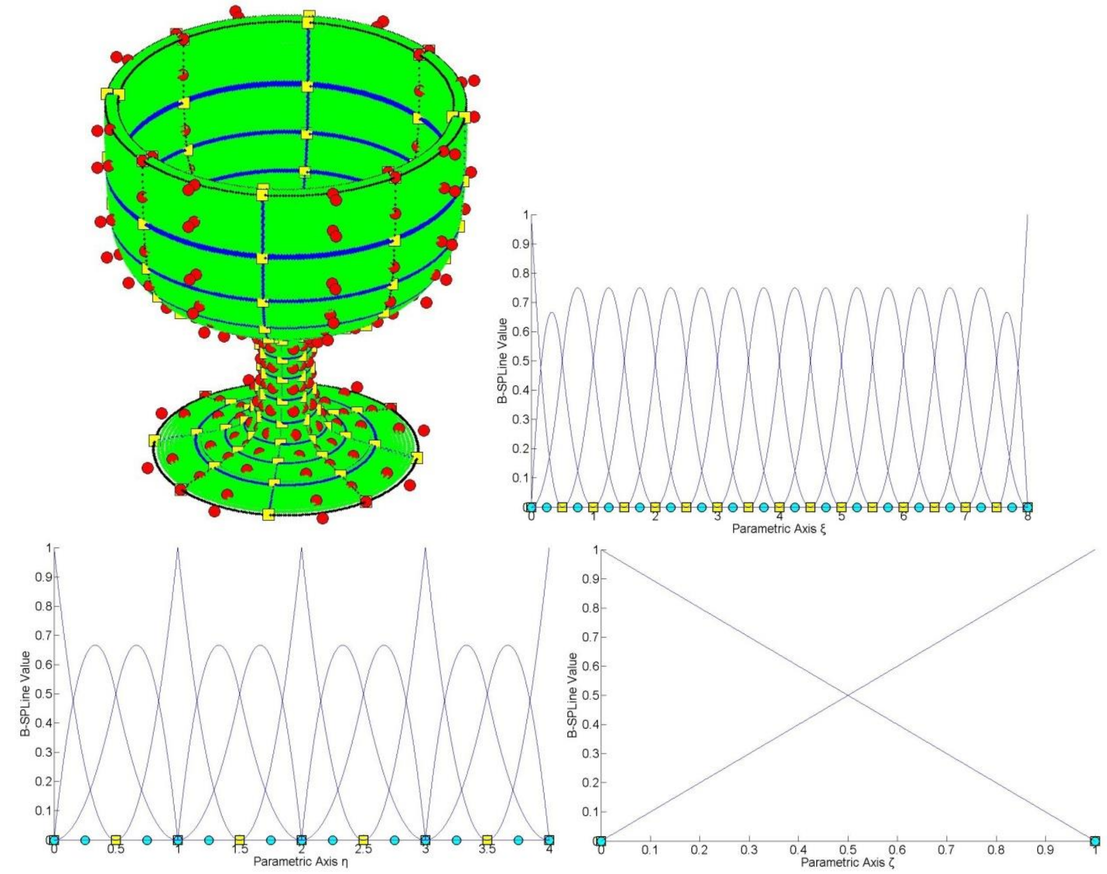
(b) Corresponding Isogeometric Analysis mesh

(c) Basis functions on ξ. Sweeps from the bottom to the top of the glass.

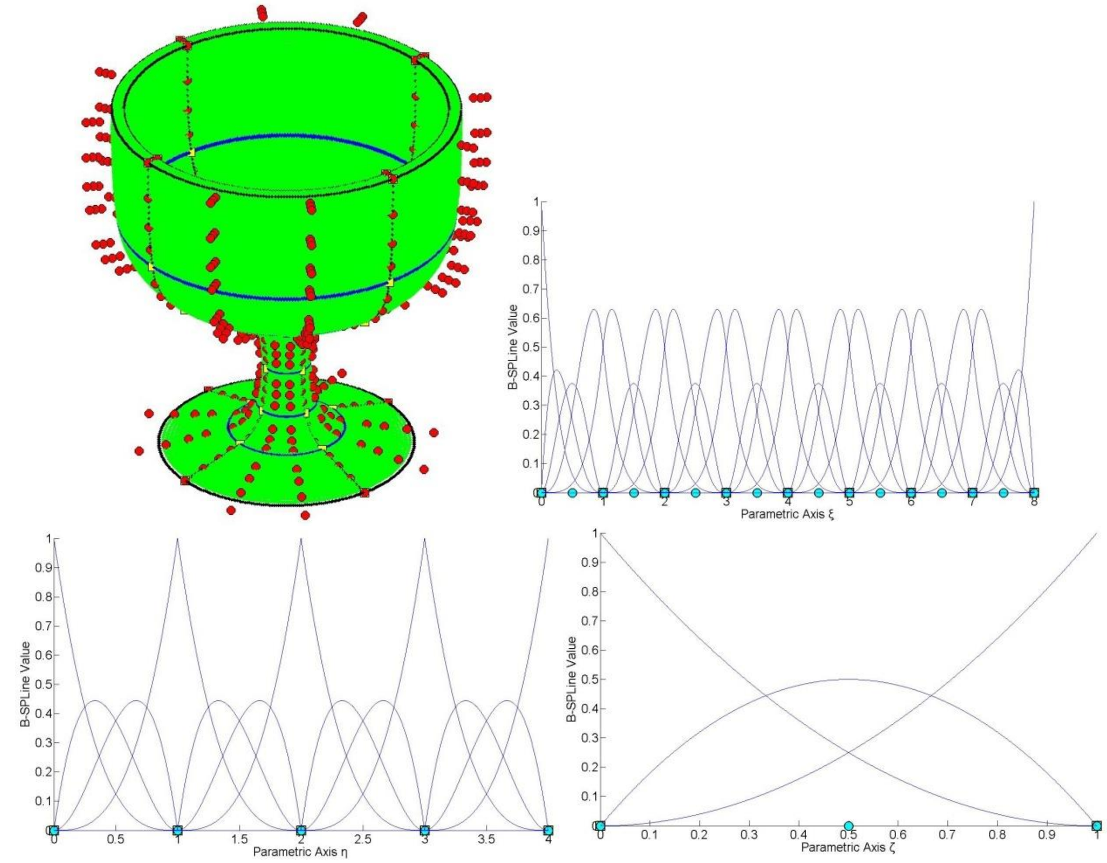
(d) Basis functions on η. Representation of circle

Figure 6.38. Coarse mesh representation and basis functions.

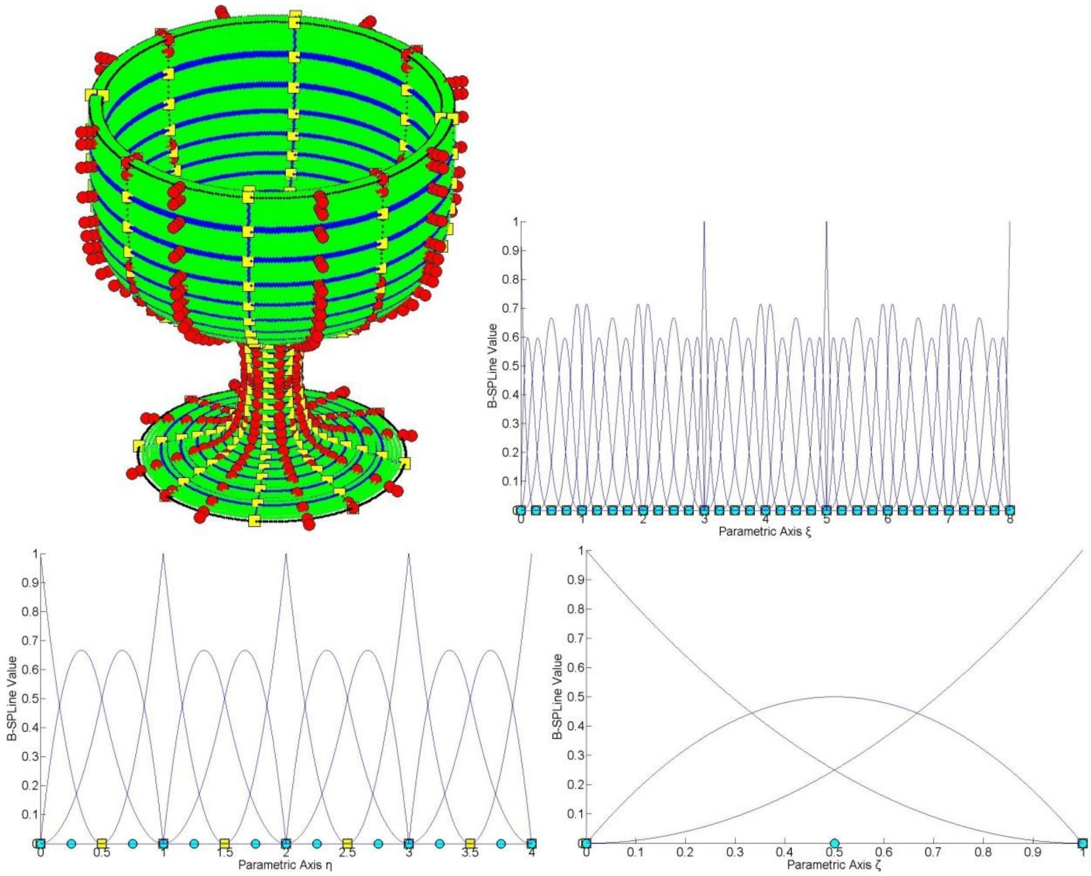
Three finer meshes are generated through h-, p- and k- Refinement. Physical space and basis functions are displayed in Figure 6.39. The neck of the glass is subjected to the maximum stress values. Therefore, Refinement aims to create regular-sized elements in that particular area.



(a) h-Refinement

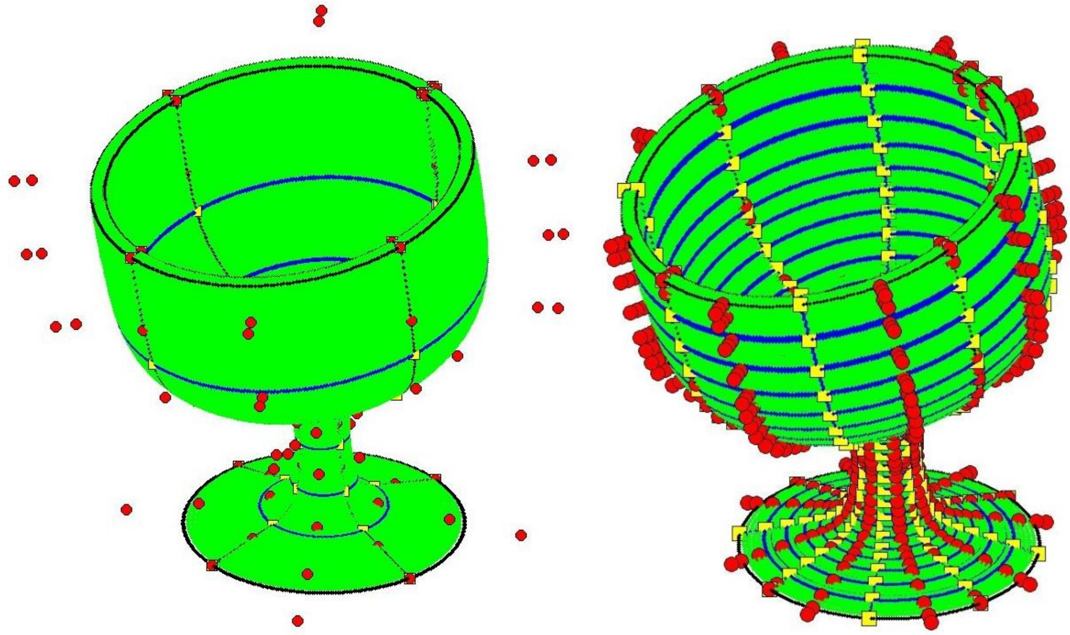


(b) p-Refinement



(c) k-Refinement

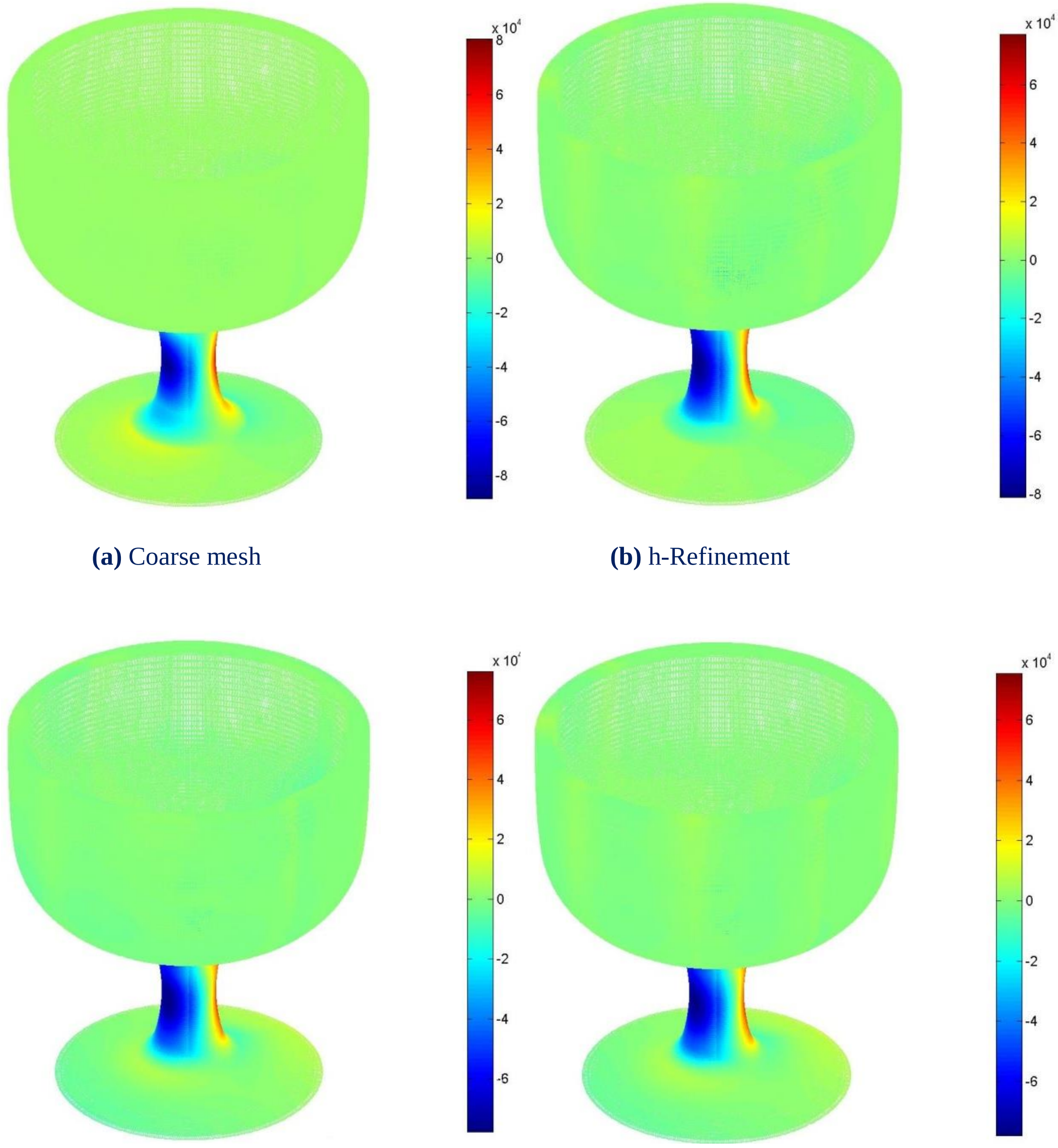
Figure 6.39. Physical Space and basis functions for the Refinements.



(a) Coarse Mesh

(b) k-Refinement

Figure 6.40. Deformed Physical Space



(a) Coarse mesh

(b) h-Refinement

(c): p-Refinement

(d): k-Refinement

Figure 6.41. σ_{zz} Contour for all representations.

Appendix A

A.1 B-SPLine Basis Functions Creation

In this example, B-SPLine basis functions are produced for $p = 2$ and Knot Value Vector $\Xi = \{\xi_1 \;\; \xi_2 \;\; \xi_3 \;\; \xi_4 \;\; \xi_5 \;\; \xi_6 \;\; \xi_7 \;\; \xi_8 \;\; \xi_9\} = \{0 \;\; 0 \;\; 0 \;\; 1 \;\; 2 \;\; 2 \;\; 3 \;\; 3 \;\; 3\}$. For $k = 9$ Knot Values and $p = 2$, $n = k - p - 1 = 6$ B-SPLine basis functions are produced. Therefore, $n + 1 = 7$ B-SPLine basis functions of degree $p = 1$ and $n + 2 = 8$ B-SPLine basis functions of degree $p = 0$ have to be evaluated. In general, for n B-SPLines of degree p, n+p-q B-SPLines of degree q are required.

A.1.1 First Iteration. Order $p = 0$.

8 box functions are required.

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

For example, $i = 4$ yields:

$$N_{4,0}(\xi) = \begin{cases} 1, & \xi_4 \le \xi < \xi_5 \\ 0, & \text{otherwise} \end{cases} \Rightarrow$$

$$N_{4,0}(\xi) = \begin{cases} 1, & 1 \le \xi < 2 \\ 0, & \text{otherwise} \end{cases}$$

Basis functions for $p = 0$ are:

$$N_{1,0}(\xi) = 0, \quad 0 \le \xi \le 3$$

$$N_{2,0}(\xi) = 0, \quad 0 \le \xi \le 3$$

$$N_{3,0}(\xi) = \begin{cases} 1, & 0 \le \xi < 1 \\ 0, & \text{otherwise} \end{cases}$$

$$N_{4,0}(\xi) = \begin{cases} 1, & 1 \le \xi < 2 \\ 0, & \text{otherwise} \end{cases}$$

$$N_{5,0}(\xi) = 0, \quad 0 \le \xi \le 3$$

$$N_{6,0}(\xi) = \begin{cases} 1, & 2 \le \xi < 3 \\ 0, & \text{otherwise} \end{cases}$$

$$N_{7,0}(\xi) = 0, \quad 0 \le \xi \le 3$$

$$N_{8,0}(\xi) = 0, \quad 0 \le \xi \le 3$$

A.1.2 Second Iteration. Order $p = 1$.

7 linear B-SPLine basis functions are produced. Cox-de Boor recursive formula for $p \geq 1$ yields:

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

In this particular example, solved for $i = 3$ and $p = 1$:

$$N_{3,1}(\xi) = \frac{\xi - \xi_3}{\xi_4 - \xi_3} N_{3,0}(\xi) \; + \; \frac{\xi_5 - \xi}{\xi_5 - \xi_4} N_{4,0}(\xi)$$

$$N_{3,1}(\xi) = \frac{\xi - 0}{1 - 0} N_{3,0}(\xi) \; + \; \frac{2 - \xi}{2 - 1} N_{4,0}(\xi)$$

$$N_{3,1}(\xi) = \xi \cdot N_{3,0}(\xi) + (2 - \xi) \cdot N_{4,0}(\xi)$$

Box functions are obtained from the previous cycle:

$$N_{3,0}(\xi) = \begin{cases} 1, & 0 \leq \xi < 1 \\ 0, & \text{otherwise} \end{cases}$$

$$N_{4,0}(\xi) = \begin{cases} 1, & 1 \leq \xi < 2 \\ 0, & \text{otherwise} \end{cases}$$

therefore,

$$N_{3,1}(\xi) = \begin{cases} 1\xi, & 0 \leq \xi < 1 \\ 1(2 - \xi), & 1 \leq \xi < 2 \\ 0, & \text{otherwise} \end{cases}$$

$$N_{3,1}(\xi) = \begin{cases} \xi, & 0 \leq \xi < 1 \\ 2 - \xi, & 1 \leq \xi < 2 \\ 0, & \text{otherwise} \end{cases}$$

Basis functions for $p = 1$:

$$N_{1,1}(\xi) = 0, \quad 0 \le \xi \le 3$$

$$N_{2,1}(\xi) = \begin{cases} 1-\xi, & 0 \le \xi < 1 \\ 0, & \text{otherwise} \end{cases}$$

$$N_{3,1}(\xi) = \begin{cases} \xi, & 0 \le \xi < 1 \\ 2-\xi, & 1 \le \xi < 2 \\ 0, & \text{otherwise} \end{cases}$$

$$N_{4,1}(\xi) = \begin{cases} \xi-1, & 1 \le \xi < 2 \\ 0, & \text{otherwise} \end{cases}$$

$$N_{5,1}(\xi) = \begin{cases} 3-\xi, & 2 \le \xi < 3 \\ 0, & \text{otherwise} \end{cases}$$

$$N_{6,1}(\xi) = \begin{cases} \xi-2, & 2 \le \xi < 3 \\ 0, & \text{otherwise} \end{cases}$$

$$N_{7,1}(\xi) = 0, \quad 0 \le \xi \le 3$$

A.1.3 Third Iteration. Order $p = 2$.

6 quadratic B-SPLine Basis functions are produced.

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

For example, basis function for $i = 2$ and $p = 2$ is calculated by:

$$N_{2,2}(\xi) = \frac{\xi - \xi_2}{\xi_4 - \xi_2} N_{2,1}(\xi) \ + \ \frac{\xi_5 - \xi}{\xi_5 - \xi_3} N_{3,1}(\xi)$$

$$N_{2,2}(\xi) = \frac{\xi - 0}{1 - 0} N_{2,1}(\xi) \ + \ \frac{2 - \xi}{2 - 0} N_{3,1}(\xi)$$

$$N_{2,2}(\xi) = \xi \cdot N_{2,1}(\xi) \ + \ \frac{2 - \xi}{2} N_{3,1}(\xi)$$

Linear basis functions are obtained from the previous cycle:

$$N_{2,1}(\xi) = \begin{cases} 1-\xi, & 0 \le \xi < 1 \\ 0, & \text{otherwise} \end{cases}$$

$$N_{3,1}(\xi) = \begin{cases} \xi, & 0 \le \xi < 1 \\ 2-\xi, & 1 \le \xi < 2 \\ 0, & \text{otherwise} \end{cases}$$

Therefore,

$$N_{2,2}(\xi)=\begin{cases}\xi(1-\xi)+\dfrac{(2-\xi)}{2}\xi, & 0\le\xi<1\\ \dfrac{(2-\xi)}{2}(2-\xi), & 1\le\xi<2\\ 0, & \text{otherwise}\end{cases}$$

$$N_{2,2}(\xi)=\begin{cases}2\xi-\dfrac{3\xi^2}{2}, & 0\le\xi<1\\ \dfrac{(2-\xi)^2}{2}, & 1\le\xi<2\\ 0, & \text{otherwise}\end{cases}$$

Basis functions for $p=2$:

$$N_{1,2}(\xi)=\begin{cases}(1-\xi)^2, & 0\le\xi<1\\ 0, & \text{otherwise}\end{cases}\qquad N_{4,2}(\xi)=\begin{cases}(\xi-1)^2, & 1\le\xi<2\\ (3-\xi)^2, & 2\le\xi<3\\ 0, & \text{otherwise}\end{cases}$$

$$N_{2,2}(\xi)=\begin{cases}2\xi-\dfrac{3\xi^2}{2}, & 0\le\xi<1\\ \dfrac{(2-\xi)^2}{2}, & 1\le\xi<2\\ 0, & \text{otherwise}\end{cases}\qquad N_{5,2}(\xi)=\begin{cases}-2\xi^2+10\xi-12, & 2\le\xi<3\\ 0, & \text{otherwise}\end{cases}$$

$$N_{3,2}(\xi)=\begin{cases}\dfrac{\xi^2}{2}, & 0\le\xi<1\\ 4\xi-\dfrac{3\xi^2}{2}-2, & 1\le\xi<2\\ 0, & \text{otherwise}\end{cases}\qquad N_{6,2}(\xi)=\begin{cases}(\xi-2)^2, & 2\le\xi<3\\ 0, & \text{otherwise}\end{cases}$$

A.2 Stiffness Matrix Formulation

An application for the creation of Stiffness Matrix in 1D is presented. 1D Elasticity is only suitable for truss elements (axial tension) and it is only used as a simplified example in this Appendix. The scope is to review and understand numerical integration techniques rather than formulate a complicated Stiffness Matrix for a more realistic problem. Shape Functions are obtained from Appendix A.1.

A.2.1 Gauss Points

For 1D problems, p Gauss Points are required per Knot Span.

$$p = 2$$
$$\xi = \pm 0.57735 \text{ and } w_{\xi}^{R} = 1.0$$

The Knot Spans are $[0,1)$, $[1,2)$ and $[2,3]$. Therefore, six Gauss Points are calculated.

Transformation of coordinates and weights from reference Knot Span to $[0,1)$:

$$\xi = \frac{(\xi_{i+1} - \xi_i) \cdot \xi^R + (\xi_{i+1} + \xi_i)}{2}$$

$$\xi^{GP1} = \frac{(1-0) \cdot (-0.57735) + (1+0)}{2} \Rightarrow \xi^{GP1} = 0.211325$$

and

$$\xi^{GP2} = \frac{(1-0) \cdot 0.57735 + (1+0)}{2} \Rightarrow \xi^{GP2} = 0.788675$$

$$w^{GP\xi} = \frac{(\xi_{i+1} - \xi_i)}{2} \cdot w_{\xi}^{R} \Rightarrow w^{GP\xi} = \frac{(1-0)}{2} \cdot 1 = 0.5$$

Accordingly, for the Knot Span $[1,2)$:

$$\xi^{GP3} = \frac{(2-1) \cdot (-0.57735) + (2+1)}{2} \Rightarrow \xi^{GP3} = 1.211325$$

and

$$\xi^{GP4} = \frac{(2-1) \cdot 0.57735 + (2+1)}{2} \Rightarrow \xi^{GP4} = 1.788675$$

$$w^{GP\xi} = \frac{(2-1)}{2} \cdot 1 = 0.5$$

For the last Knot Span $\left[2,3\right]$:

$$\xi^{GP5} = \frac{(3-2)\cdot(-0.57735)+(3+2)}{2} \Rightarrow \xi^{GP5} = 2.211325$$

and

$$\xi^{GP6} = \frac{(3-2)\cdot 0.57735+(3+2)}{2} \Rightarrow \xi^{GP6} = 2.788675$$

$$w^{GP\xi} = \frac{(3-2)}{2}\cdot 1 = 0.5$$

A.2.2 Deformation Matrix

In order to evaluate the Deformation Matrix, the derivatives of the basis functions are calculated.

$$R_{1,\xi}(\xi) = \begin{cases} -2\cdot(1-\xi), & 0\le\xi<1 \\ 0, & \text{otherwise} \end{cases}$$

$$R_{4,\xi}(\xi) = \begin{cases} 2(\xi-1), & 1\le\xi<2 \\ -2(3-\xi), & 2\le\xi<3 \\ 0, & \text{otherwise} \end{cases}$$

$$R_{2,\xi}(\xi) = \begin{cases} 2-3\cdot\xi, & 0\le\xi<1 \\ \xi-2, & 1\le\xi<2 \\ 0, & \text{otherwise} \end{cases}$$

$$R_{5,\xi}(\xi) = \begin{cases} -4\cdot\xi+10, & 2\le\xi<3 \\ 0, & \text{otherwise} \end{cases}$$

$$R_{3,\xi}(\xi) = \begin{cases} \xi, & 0\le\xi<1 \\ 4-3\cdot\xi, & 1\le\xi<2 \\ 0, & \text{otherwise} \end{cases}$$

$$R_{6,\xi}(\xi) = \begin{cases} 2\cdot(\xi-2), & 2\le\xi<3 \\ 0, & \text{otherwise} \end{cases}$$

The coordinates are set as

$$\underset{(1\times 6)}{\{X_1 \quad X_2 \quad X_3 \quad X_4 \quad X_5 \quad X_6\}^T} = \underset{(1\times 6)}{\{2 \quad 4 \quad 5 \quad 7 \quad 11 \quad 15\}^T}$$

The Jacobian Matrix is given by the equation below:

$$\underset{(1\times1)}{[J]} = \underset{(1\times n)}{\left[R_{1,\xi}(\xi) \quad R_{2,\xi}(\xi) \quad \dots \quad \dots \quad \dots \quad R_{n,\xi}(\xi)\right]}\cdot \underset{(n\times1)}{\begin{bmatrix} X_1 \\ X_2 \\ \cdot \\ \cdot \\ \cdot \\ X_n \end{bmatrix}}$$

For Gauss Point $\xi^{GP1} = 0.211325$ in Knot Span $\left[0,1\right)$:

$$R_{1,\xi}(\xi^{GP1}) = -2\cdot\left(1-\xi^{GP1}\right) = -2\cdot\left(1-0.211325\right) = -1.57735$$

$$R_{2,\xi}(\xi^{GP1}) = 2-3\cdot\xi^{GP1} = 2-3\cdot 0.211325 = 1.366025$$

$$R_{3,\xi}(\xi^{GP1}) = \xi^{GP1} = 0.211325$$

$$R_{4,\xi}(\xi^{GP1}) = 0$$

$$R_{5,\xi}(\xi^{GP1}) = 0$$

$$R_{6,\xi}(\xi^{GP1}) = 0$$

$$\underset{(1\times1)}{\left[J_{GP1}\right]} = \underset{(1\times6)}{\left[-1.57735 \quad 1.366025 \quad 0.211325 \quad 0 \quad 0 \quad 0\right]}\cdot \underset{(6\times1)}{\begin{bmatrix}2\\4\\5\\7\\11\\15\end{bmatrix}} \Rightarrow$$

$$\left[J_{GP1}\right] = 3.366025 \Rightarrow$$

$$\left[J_{GP1}\right]^{-1} = \left[\frac{1}{J_{GP1}}\right] = 0.297086326 \Rightarrow$$

$$\underset{(1\times1)}{\left[B_1^{GP1}\right]} = \left[\frac{1}{J_{GP1}}\right] = 0.297086326$$

and

$$\underset{(1\times n)}{\left[B_2\left(\xi\right)\right]} = \left[R_{1,\xi}(\xi) \quad R_{2,\xi}(\xi) \quad \dots \quad \dots \quad \dots \quad R_{n,\xi}(\xi)\right] \Rightarrow$$

$$\underset{(1\times6)}{\left[B_2^{GP1}\right]} = \left[-1.57735 \quad 1.366025 \quad 0.211325 \quad 0 \quad 0 \quad 0\right]$$

Deformation Matrix is evaluated by

$$\underset{(1\times n)}{\left[B(\xi)\right]} = \underset{(1\times 1)}{\left[B_1(\xi)\right]} \cdot \underset{(1\times n)}{\left[B_2(\xi)\right]}$$

$$\underset{(1\times 6)}{\left[B^{GP1}\right]} = [0.297086326] \cdot [-1.57735 \quad 1.366025 \quad 0.211325 \quad 0 \quad 0 \quad 0] \Rightarrow$$

$$\underset{(1\times 6)}{\left[B^{GP1}\right]} = [-0.4686 \quad 0.4058 \quad 0.0628 \quad 0 \quad 0 \quad 0]$$

For Gauss Point $\xi^{GP2} = 0.788675$ in Knot Span $[0,1)$:

$R_{1,\xi}(\xi^{GP2}) = -0.42265$ $R_{4,\xi}(\xi^{GP2}) = 0$

$R_{2,\xi}(\xi^{GP2}) = -0.366025$ $R_{5,\xi}(\xi^{GP2}) = 0$

$R_{3,\xi}(\xi^{GP2}) = 0.788675$ $R_{6,\xi}(\xi^{GP2}) = 0$

$$\left[J_{GP2}\right] = [1.633975] \Rightarrow$$

$$\underset{(1\times 1)}{\left[B_1^{GP2}\right]} = \left[\frac{1}{J_{GP2}}\right] = [0.612004468]$$

and

$$\underset{(1\times 6)}{\left[B_2^{GP2}\right]} = [-0.42265 \quad -0.366025 \quad 0.788675 \quad 0 \quad 0 \quad 0]$$

$$\underset{(1\times 6)}{\left[B^{GP2}\right]} = [-0.2587 \quad -0.2240 \quad 0.4827 \quad 0 \quad 0 \quad 0]$$

For $\xi^{GP3} = 1.211325$ in $[1,2)$:

$R_{1,\xi}(\xi^{GP3}) = 0$ $R_{4,\xi}(\xi^{GP3}) = 0.42265$

$R_{2,\xi}(\xi^{GP3}) = -0.788675$ $R_{5,\xi}(\xi^{GP3}) = 0$

$R_{3,\xi}(\xi^{GP3}) = 0.366025$ $R_{6,\xi}(\xi^{GP3}) = 0$

$$\left[J_{GP3}\right] = 1.633975 \Rightarrow$$

$$\underset{(1\times 1)}{\left[B_1^{GP3}\right]} = \left[\frac{1}{J_{GP3}}\right] = 0.612004468$$

$$\underset{(1\times 6)}{\left[B_2^{GP3}\right]}=\begin{bmatrix}0 & -0.788675 & 0.366025 & 0.42265 & 0 & 0\end{bmatrix}$$

$$\underset{(1\times 6)}{\left[B^{GP3}\right]}=\begin{bmatrix}0 & -0.4827 & 0.2240 & 0.2587 & 0 & 0\end{bmatrix}$$

For the fourth Gauss Point $\xi^{GP4}=1.788675$ and the Knot Span $[1,2)$:

$R_{1,\xi}(\xi^{GP4})=0$	$R_{4,\xi}(\xi^{GP4})=1.57735$
$R_{2,\xi}(\xi^{GP4})=-0.211325$	$R_{5,\xi}(\xi^{GP4})=0$
$R_{3,\xi}(\xi^{GP4})=-1.366025$	$R_{6,\xi}(\xi^{GP4})=0$

$$\left[J_{GP4}\right]=3.366025\Rightarrow$$

$$\underset{(1\times 1)}{\left[B_1^{GP4}\right]}=\left[\frac{1}{J_{GP4}}\right]=0.297086326$$

and

$$\underset{(1\times 6)}{\left[B_2^{GP4}\right]}=\begin{bmatrix}0 & -0.211325 & -1.366025 & 1.57735 & 0 & 0\end{bmatrix}$$

$$\underset{(1\times 6)}{\left[B^{GP4}\right]}=\begin{bmatrix}0 & -0.0628 & -0.4058 & 0.4686 & 0 & 0\end{bmatrix}$$

For $\xi^{GP5}=2.211325$ in $[2,3]$:

$R_{1,\xi}(\xi^{GP5})=0$	$R_{4,\xi}(\xi^{GP5})=-1.57735$
$R_{2,\xi}(\xi^{GP5})=0$	$R_{5,\xi}(\xi^{GP5})=1.1547$
$R_{3,\xi}(\xi^{GP5})=0$	$R_{6,\xi}(\xi^{GP5})=0.42265$

$$\left[J_{GP5}\right]=8\Rightarrow$$

$$\underset{(1\times 1)}{\left[B_1^{GP5}\right]}=\left[\frac{1}{J_{GP5}}\right]=0.125$$

$$\underset{(1\times6)}{\left[B_2^{GP5}\right]} = \left[0 \quad 0 \quad 0 \quad -1.57735 \quad 1.1547 \quad 0.42265\right]$$

$$\underset{(1\times6)}{\left[B^{GP5}\right]} = \left[0 \quad 0 \quad 0 \quad -0.1972 \quad 0.1443 \quad 0.0528\right]$$

For $\xi^{GP6} = 2.788675$ in $[2,3]$:

$R_{1,\xi}(\xi^{GP6}) = 0$ $R_{4,\xi}(\xi^{GP6}) = -0.42265$

$R_{2,\xi}(\xi^{GP6}) = 0$ $R_{5,\xi}(\xi^{GP6}) = -1.1547$

$R_{3,\xi}(\xi^{GP6}) = 0$ $R_{6,\xi}(\xi^{GP6}) = 1.57735$

$$\left[J_{GP6}\right] = [8] \Rightarrow$$

$$\underset{(1\times1)}{\left[B_1^{GP6}\right]} = \left[\frac{1}{J_{GP6}}\right] = [0.125]$$

and

$$\underset{(1\times6)}{\left[B_2^{GP6}\right]} = \left[0 \quad 0 \quad 0 \quad -0.42265 \quad -1.1547 \quad 1.57735\right]$$

$$\underset{(1\times6)}{\left[B^{GP6}\right]} = \left[0 \quad 0 \quad 0 \quad -0.0528 \quad -0.1443 \quad 0.1972\right]$$

A.2.3 Numerical Integration

Stiffness Matrix is evaluated by:

$$\underset{(n\times n)}{[K]} = \sum_{i=1}^{GP\xi} \left\{ \underset{(n\times1)}{\left[B(\xi_i)\right]^T} \cdot \underset{(1x1)}{[E]} \cdot \underset{(1\times n)}{\left[B(\xi_i)\right]} \cdot A \cdot \det[J] \cdot w_i^{GP\xi} \right\}$$

where,

$A = 1\ m^2$

$\underset{(1x1)}{[E]} = E = 210 \cdot 10^6 \ \dfrac{KN}{m^2}$

$\det[J] = [J]$

$$\underset{(6\times 6)}{\left[K^{GP1}\right]} = \begin{bmatrix} -0.4686 \\ 0.4058 \\ 0.0628 \\ 0 \\ 0 \\ 0 \end{bmatrix} \cdot \left[210\cdot 10^{6}\right] \cdot \left[-0.4686 \quad 0.4058 \quad 0.0628 \quad 0 \quad 0 \quad 0\right] \cdot 1 \cdot 3.366025 \cdot 0.5 \Rightarrow$$

The contribution of the first Gauss Point to the Stiffness Matrix is:

$$\underset{(6\times 6)}{\left[K^{GP1}\right]} = 10^{6} \cdot \begin{bmatrix} 77.6088 & -67.2080 & -10.4008 & 0 & 0 & 0 \\ -67.2080 & 58.2010 & 9.0070 & 0 & 0 & 0 \\ -10.4008 & 9.0070 & 1.3939 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Respectively, for the other Gauss Points:

$$\underset{(6\times 6)}{\left[K^{GP2}\right]} = 10^{6} \cdot \begin{bmatrix} 11.4823 & 9.9421 & -21.4244 & 0 & 0 & 0 \\ 9.9421 & 8.6086 & -18.5507 & 0 & 0 & 0 \\ -21.4244 & -18.5507 & 39.9751 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\underset{(6\times 6)}{\left[K^{GP3}\right]} = 10^{6} \cdot \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 39.9751 & -18.5507 & -21.4244 & 0 & 0 \\ 0 & -18.5507 & 8.6086 & 9.9421 & 0 & 0 \\ 0 & -21.4244 & 9.9421 & 11.4823 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\underset{(6\times 6)}{\left[K^{GP4}\right]} = 10^{6} \cdot \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.3939 & 9.0070 & -10.4008 & 0 & 0 \\ 0 & 9.0070 & 58.2010 & -67.2080 & 0 & 0 \\ 0 & -10.4008 & -67.2080 & 77.6088 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\underset{(6\times6)}{\left[K^{GP5}\right]}=10^6\cdot\begin{bmatrix}0&0&0&0&0&0\\0&0&0&0&0&0\\0&0&0&0&0&0\\0&0&0&32.6658&-23.9030&-8.7462\\0&0&0&-23.9030&17.4909&6.4000\\0&0&0&-8.7462&6.400&2.3418\end{bmatrix}$$

$$\underset{(6\times6)}{\left[K^{GP6}\right]}=10^6\cdot\begin{bmatrix}0&0&0&0&0&0\\0&0&0&0&0&0\\0&0&0&0&0&0\\0&0&0&2.3418&6.4000&-8.7462\\0&0&0&6.4000&17.4909&-23.9030\\0&0&0&-8.7462&-23.9030&32.6658\end{bmatrix}$$

The Stiffness Matrix is obtained as the sum of the separate Gauss Points' Stiffness Matrices:

$$\underset{(6\times6)}{\left[K\right]}=\sum_{GPi=1}^{6}\underset{(6\times6)}{\left[K^{GPi}\right]}$$

The final result is:

$$\underset{(6\times6)}{\left[K\right]}=10^6\cdot\begin{bmatrix}89.0911&-57.2659&-31.8252&0&0&0\\-57.2659&108.1786&-19.0875&-31.8252&0&0\\-31.8252&-19.0875&108.1786&-57.2659&0&0\\0&-31.8252&-57.2659&124.0987&-17.5030&-17.4924\\0&0&0&-17.5030&34.9818&-17.5030\\0&0&0&-17.4924&-17.5030&35.0076\end{bmatrix}$$

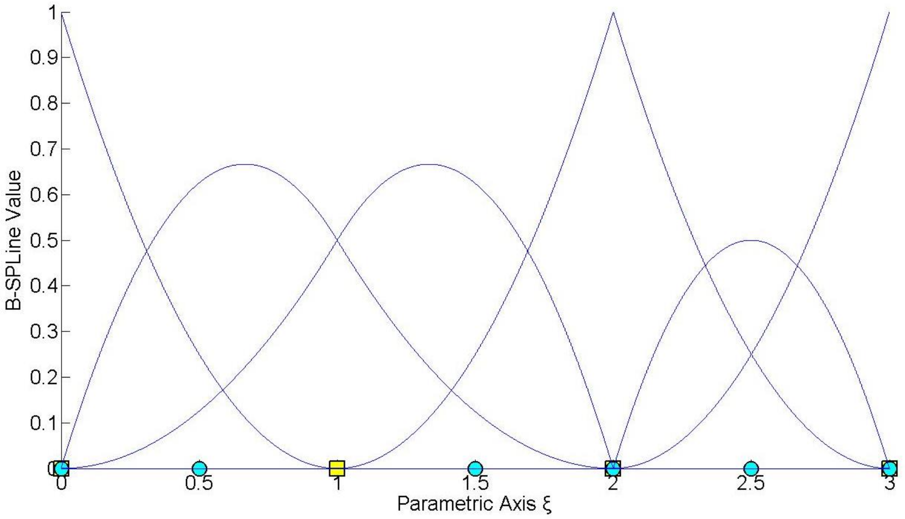


Figure A.1. Basis Functions for Appendix A.

7 Conclusions

Exact Geometry

Isogeometric Analysis enables exact representation for the analysis process. The mathematical model has the exact same features as the design model. Therefore, transition from design to analysis model does not produce any geometrical errors and the accuracy is instantly increased.

Improved refinement schemes

Analysis is carried out without the creation of an approximate mesh. The analysis mesh is directly connected with the accurate geometrical representation of the model. Refinements preserve the exact geometry and parametric mapping while providing greater accuracy for analysis. FEM, on the contrary, creates a new approximation with each refinement; thus, IGA refinement is much faster and more efficient. In addition to the classical h- and p- refinement, a new combination also applies in IGA. k-Refinement is proving to be more competitive than its counterparts.

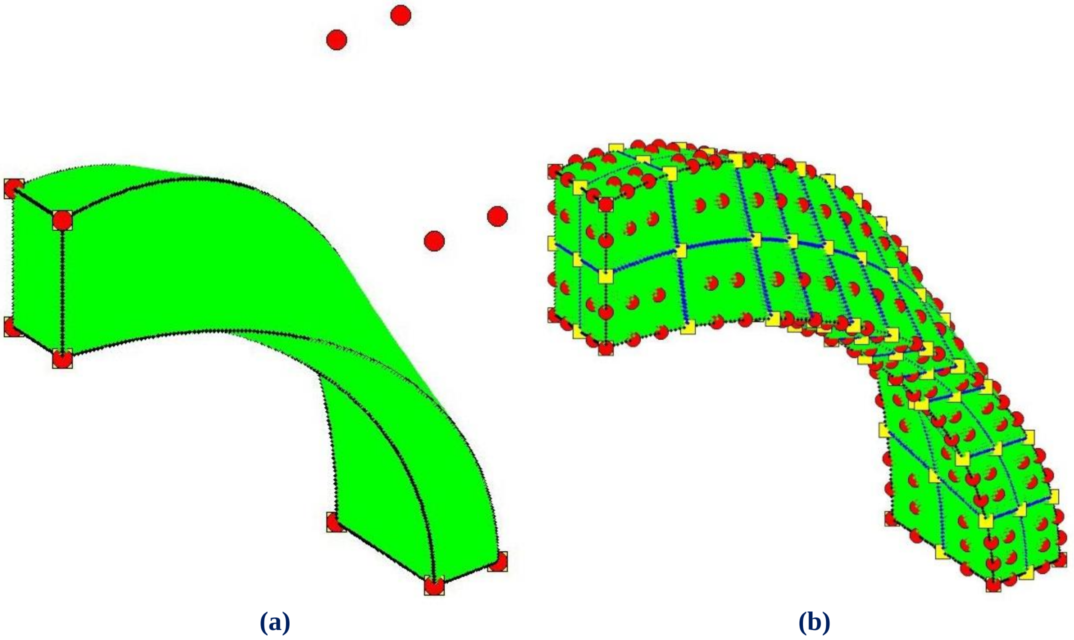
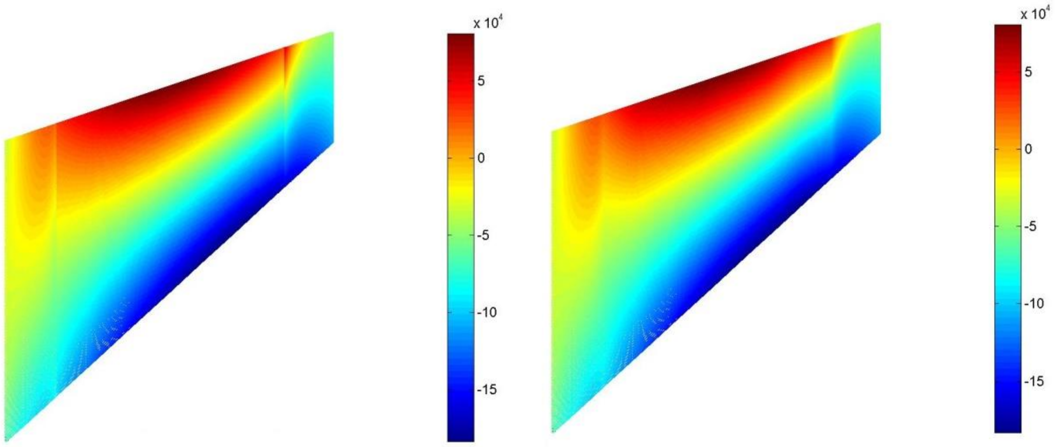


Figure 7.1. (a) Coarse mesh and **(b)** application of k-Refinement.

Element interconnectivity

Isogeometric Shape functions hold an overlapping that leads to greater interconnectivity between elements. Derivatives are continuous across element boundaries, therefore stress and strain fields are continuous as well. The use of corrective methods such as extrapolation is applied only to special cases of C^0 Continuity.



(a) **(b)**

Figure 7.2. Stress contour σ_{xy} for **(a)** C^0 **(b)** C^1 Continuity.

Patch Utilitization in IGA

Patches are used in Isogeometric Analysis when a significant change in Geometry or Material occurs. Similarly to Finite Elements, patches possess C^{-1} Continuity on the boundary and all Shape functions are contained in their interior. During Stiffness Matrix Formulation, a local Stiffness Matrix is produced for every patch. Afterwards, the global Stiffness Matrix is built through patch connectivity.

Each patch has a unique mapping from its own Parameter Space to the Physical Space. Refinement is applied separately for every patch. NURBS patch connection is not always water-tight; boundary inconsistencies are possible to exist.

Patch boundary can also be created through a single Knot Value Vector, when the separate Parameter Spaces of the patches can be merged into one. Control Points on the boundaries are fused into one. The corresponding C^{-1} Continuous Shape functions are combined into one, with C^0 Continuity across that edge. C^0 Continuity is also useful for the creation of interpolatory Control Points at a specific part of the model.

Stiffness Matrix Formulation

Stiffness Matrix creation for IGA follows the same general principles as in FEM. Enhanced Shape function overlapping leads to a matrix with greater bandwidth. As more non-zero elements are created and interconnectivity is increased, the resulting Stiffness Matrix is a more accurate representation of the natural problem. However, this overlapping makes the formulation process time-consuming.

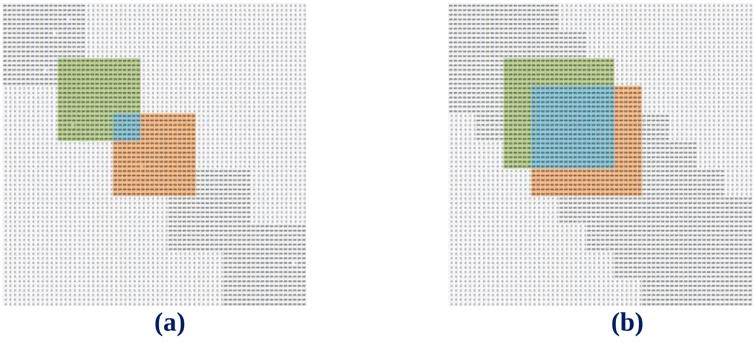


Figure 7.3. Stiffness Matrices with various interconnectivities. **(a)** C^0 and **(b)** C^2 Continuity.

The role of the Engineer

The collective experience from all the years of FEM practice is definitely useful in Isogeometric Analysis as well. The engineer should be able to maintain a thorough understanding of the methods and techniques involved, as no machine can replace the creativity and important role of the human mind.

Special care has to be taken so that the Jacobian matrix is reversible, or in other words, positive directions of the axes in Parameter and Physical Space are equivalent. Selection of distorted elements is also inefficient for analysis. Knot rectangles and cuboids of similar dimensions and nearly orthogonal shape are preferred. Reverse refinement, for analysis purposes, should be applied after careful consideration.

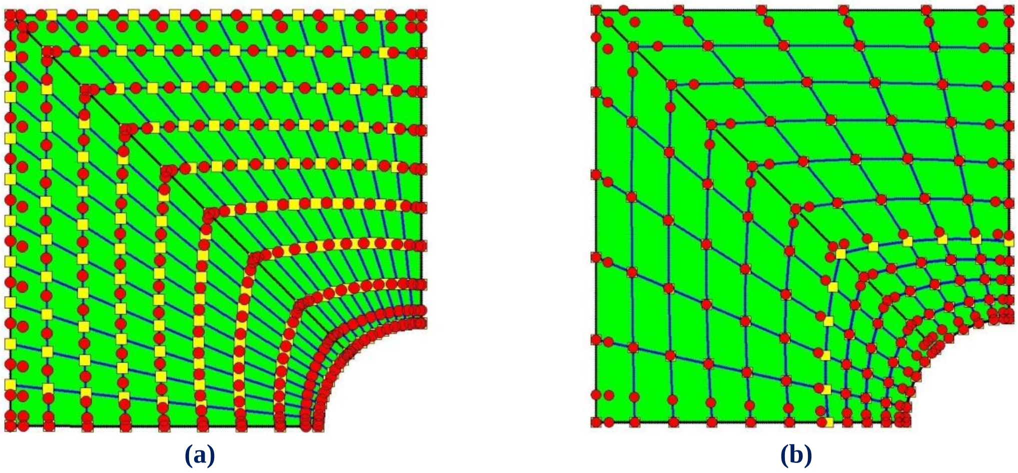


Figure 7.4. **(a)** Inefficient and **(b)** Efficient refinement.

References

Published Books and Scientific Papers

[1] Isogeometric Analysis: Toward Integration of CAD and FEA - J. Austin Cottrell, Thomas J. R. Hughes, Yuri Bazilevs - Wiley, 2009

[2] Analysis of Structures with the Finite Element Method - M. Papadrakakis - Papasotiriou, 2001

[3] The NURBS Book - L. Piegl, W. Tiller - Springer, 1997

[4] Efficient quadrature for NURBS- based Isogeometric Analysis - T. J. R. Hughes, A. Realli, G. Sangalli - 2008

[5] A fully ‘‘locking-free’’ isogeometric approach for plane linear elasticity problems - Auricchio, Veiga, Buffa, Lovadina, Reali, Sangalli - 2007

[6] A Practical Guide to Splines - de Boor - 1st Revised Edition 2001, Springer

[7] An introduction to IGA with Matlab implementation, FEM and XFEM Formulations - Nguyen, Simpson, Bordas, Rabczuk - 2012

[8] An Introduction to NURBS With Historical Perspective - Rogers - 2000, The Morgan Kaufmann Series in Computer Graphics

[9] Comparison of Numerical Quadrature Schemes in IGA - Rypl, Patzak - 2011

[10] Curves and Surfaces for CAGD, A Practical Guide - Farin - 5th Edition 2001

[11] Efficient quadrature for NURBS-based IGA - Hughes, Reali, Sangalli - 2008, Elsevier

[12] Enhancing IGA by a FEA-based local refinement strategy - Kleiss, Jüttler, Zulehner - 2011

[13] Isogeometric Analysis, CAD, finite elements, NURBS, exact geometry and mesh refinement - Hughes, Cottrell, Bazilevs - 2004

[14] The cost of continuity, Performance of using direct solvers in IGA - Collier, Pardo, Dalcin, Paszynski, Calo - 2011

Websites

[15] http://amresearch.blogspot.gr/2006/01/in-memoriam-to-professor-john-h.html

[16] http://en.wikipedia.org/wiki/Abaqus

[17] http://en.wikipedia.org/wiki/Nastran

[18] http://www.cervenka.cz/products/

[19] http://www.adina.com/products.shtml

[20] http://en.wikipedia.org/wiki/Femap

[21] http://www.pi.gr/CMSnew/index.php

[22] http://www.sofistik.com/en/

[23] http://lhlogismiki.gr/index_gr.php

[24] http://www.mactech.com/articles/develop/issue_25/schneider.html

[25] http://en.wikipedia.org/wiki/Computer-aided_geometric_design#History

[26] http://isicad.net/articles.php?article_num=14940