

# **FREE VIBRATION ISOGEOMETRIC ANALYSIS OF FRAMED STRUCTURES**

A DISSERTATION

*Submitted in partial fulfilment of the  
requirements for the award of the degree*

*of*

**MASTER OF TECHNOLOGY**

*in*

**CIVIL ENGINEERING**

*by*

**PRASHOON GUPTA**



**DEPARTMENT OF CIVIL ENGINEERING  
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE**

**ROORKEE - 247 667 (INDIA)**

**MAY, 2019**



**©INDIAN INSTITUTE OF TECHNOLOGY ROORKEE, ROORKEE-2019  
ALL RIGHTS RESERVED**



# INDIAN INSTITUTE OF TECHNOLOGY ROORKEE ROORKEE

## CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the dissertation report entitled “Free Vibration Isogeometric Analysis of Framed Structures” in partial fulfilment of the requirements for the award of the Degree of Master of Technology and submitted in the Department of Civil Engineering of the Indian Institute of Technology Roorkee, Roorkee is an authentic record of my own work carried out during the period from July, 2018 to April, 2019 under the supervision of Dr. Rajib Chowdhury, Associate Professor, Department of Civil Engineering, Indian Institute of Technology Roorkee, Roorkee.

The matter presented in this thesis has not been submitted by me for the award of any other degree of this or any other Institute.

**(Prashoon Gupta)**

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

**(Dr. Rajib Chowdhury)**  
Supervisor

**Date:**

# *Abstract*

Department of Civil Engineering

Master of Technology

## **Free Vibration Isogeometric Analysis of Framed Structures**

by Prashoon Gupta

Isogeometric analysis (IGA) represents a recently developed technology in computational mechanics that offers the possibility of integrating methods for analysis and Computer Aided Design (CAD) into a single, unified process. The implications to practical engineering design scenarios are profound, since the time taken from design to analysis is greatly reduced, leading to dramatic gains in efficiency. In this study an introduction to Isogeometric finite element analysis on linear elasticity problems in 2D has been given using non uniform rational B-splines (NURBS) as basis functions. Theory of B-Splines and FEM have been studied and derived the equations needed to perform linear elasticity analysis. An Isogeometric finite element solver has been programmed in Python.

The IGA solver was used to compute the free vibration frequencies of bar and beam element. Both the elements formulated are 1D elements with beam element based on Euler-Bernoulli beam theory. The free vibration frequencies of both the elements were computed in the framework of NURBS based IGA and compared with analytical frequencies. The numerical frequency matched with analytical frequency for first 40 modes for both the elements. Since the higher modes frequencies are irrelevant in context of structural engineering, the results obtained can be termed as sufficiently good for further analysis.

The IGA solver was then used to analyse plate element based on Reissner-Mindlin theory of plates. Reissner-Mindlin theory was preferred instead of Kirchhoff-Love theory

to account for transverse shear deformations which are necessary to be considered for analysis of thick plates. The plate element was analysed for different boundary conditions: All sides simply supported and all sides clamped, and mesh sizes: 25, 36, 49, 64, 81, 100, 121, 256, 441. The good behaviour of the method was verified and compared with analytical results. Plot of relative error percentage and the high rate of convergence in every case considered can be seen in the results. The relative percentage error reduced to 0.00896% from an initial error of 11.3% with mesh refinement for simply supported plate subjected to UDL. This reduction in error was even more drastic in case of Clamped plate subjected to UDL with error reducing from 92.12% to 1.18% for same amount of mesh refinement. The simply supported plate produced convergent solutions in fewer mesh density and without the need for selective integration.

A good solver must be capable of automatically meshing the problem domain without consuming much computation power. Hence another IGA framework was prepared based on Polynomial over Hierarchical T-splines (PHT-splines) which is capable of being locally refined. NURBS based IGA is not suited for local refinement because of its global tensor product structure. A quad tree structure was used to construct PHT-spline elements since they are better suited to track connectivity between elements across different refinement levels. For adaptivity, Zienkiewicz-Zhu error estimator was formulated which is a recovery based a-posteriori error estimator. Dörfler marking scheme was used to mark the elements to be refined after computing the error for each element at the current refinement level. To exploit the good nature of automatic adaptive refinement, a cantilever beam was analysed, because of the presence of re-entrant corners and stress concentrations. The cantilever beam is based on Euler-Bernoulli beam theory. The promised results were obtained with the convergence plot showing good convergence. A total of 12 refinement steps were needed to reduce the error in problem domain within the prescribed limits. The mesh structure as well as the contour plots of stresses are shown in the results. A cluster of elements can be seen at the re-entrant corners as expected, increasing at each refinement level until the error reduced below the limit prescribed.

# *Acknowledgements*

Writing these acknowledgements marks the end of my adventure at the IIT Roorkee as a student. This dissertation is the result of a year of work at the Structural Engineering department. This would not have the case without the following people.

First of all, I would like to thank my supervisor Dr. Rajib Chowdhury for introducing me to Isogeometric Analysis, a novel approach to classical Finite Element Methods. Furthermore, thanks to him for his inspiration, enthusiasm, and giving me basically the freedom to perform my own research without derailing. Working on this topic was indeed beyond my professional desires but it happened and I have to say that it has been a truly fruitful and enriching experience. Next, I would like to acknowledge Abhinav Gupta for continuous encouragement, insightful discussions and his time to answer the questions. It is his confidence in me that has helped me push myself beyond the limits which I didn't knew was possible.

Thanks also Siddharth and Sumit for being such great friends and study buddies during master. Thanks to my family, for their continuous encouragement and support to help me bring the best out of myself.

**Date:**

(Prashoon Gupta)

# Contents

<b>Abstract</b>	<b>iv</b>
<b>Acknowledgements</b>	<b>vi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Need for study . . . . .	1
1.2 Objectives of the study . . . . .	2
1.3 Scope of the study . . . . .	3
1.4 Organization of dissertation . . . . .	4
<b>2 Literature Review</b>	<b>6</b>
2.1 General . . . . .	6
2.2 Literature Study . . . . .	6
2.3 Summary . . . . .	9
<b>3 2D Linear Elasticity</b>	<b>11</b>
3.1 Strain . . . . .	11
3.2 Stress . . . . .	13
3.3 Traction . . . . .	13
3.4 Hooke's law for plane stress . . . . .	15
3.5 Beam Theory . . . . .	15
3.5.1 Euler-Bernoulli beam . . . . .	16
3.5.2 Timoshenko beam . . . . .	16
3.6 Plate theory . . . . .	18
3.6.1 Kirchhoff-Love theory . . . . .	18
3.6.2 Reissner-Mindlin theory . . . . .	19
3.7 Natural vibration frequencies and modes . . . . .	21
<b>4 Finite Element Method</b>	<b>24</b>

4.1	Some definitions . . . . .	24
4.2	Galerkin method . . . . .	25
4.3	Isoparametric concept . . . . .	26
4.4	Jacobian . . . . .	27
	First order Jacobian . . . . .	27
	Second order Jacobian . . . . .	29
4.5	Quadrature . . . . .	29
4.6	Summary of FEM process . . . . .	30
<b>5</b>	<b>B-Splines and NURBS</b>	<b>32</b>
5.1	B-Splines . . . . .	32
5.1.1	Knot Vectors . . . . .	32
5.1.2	Basis Functions . . . . .	33
5.1.3	B-Spline curves . . . . .	35
5.1.4	B-Spline Surface . . . . .	36
5.1.5	Derivatives of B-Spline Functions . . . . .	38
5.1.6	Refinement; Knot insertion . . . . .	40
5.2	NURBS . . . . .	41
5.2.1	Geometric perspective . . . . .	42
5.2.2	Basis functions . . . . .	43
5.2.3	Derivatives of basis functions . . . . .	44
<b>6</b>	<b>Isogeometric Analysis</b>	<b>46</b>
6.1	Spaces and Mappings . . . . .	46
6.1.1	The physical space . . . . .	47
6.1.2	The control mesh . . . . .	47
6.1.3	The parameter space . . . . .	47
6.1.4	The index space . . . . .	49
6.1.5	The parent element . . . . .	49
6.2	Isogeometric Analysis Vs Finite Element Method . . . . .	49
<b>7</b>	<b>Adaptive IGA</b>	<b>53</b>
7.1	Introduction . . . . .	53
7.2	Mesh structure . . . . .	55
7.2.1	T-meshes . . . . .	55
7.2.2	Hierarchical T-mesh . . . . .	55



7.2.3	Tree structure	57
7.3	Basis function of PHT-splines	57
7.3.1	Definition of knot vectors	57
7.3.2	The Bernstein basis	59
7.3.3	Bézier extraction for T-splines	59
7.3.4	Modification of the basis functions at level k	60
Blossoming with De Casteljau's algorithm		60
Truncation by zeroing out Bézier coefficients		60
7.3.5	Insertion of new basis functions	62
7.4	Recovery based error estimation	62
7.5	Marking algorithm	63
<b>8</b>	<b>Isogeometric Elements</b>	<b>66</b>
8.1	Bar Element	66
8.2	Beam Element	70
8.3	Plate Element	72
<b>9</b>	<b>Numerical Results</b>	<b>75</b>
9.1	Bar element free vibration	75
9.1.1	Problem definition	75
9.1.2	Solution	76
9.2	Beam element free vibration	77
9.2.1	Problem definition	77
9.2.2	Solution	79
9.3	Plate element	80
9.3.1	Problem definition	80
9.3.2	Solution	81
9.4	Adaptive IGA	87
9.4.1	Problem definition	87
9.4.2	Solution	88
<b>10</b>	<b>Conclusion and Recommendation</b>	<b>92</b>
10.1	General	92
10.2	Conclusion	92
10.3	Recommendations for further work	95
	<b>Bibliography</b>	<b>97</b>

# List of Figures

1.1	Engineering designs are becoming increasingly complex, making analysis a time consuming and expensive endeavour. . . . .	2
3.1	Deformation of a control volume . . . . .	12
3.2	Stress Components . . . . .	13
3.3	Relationship between stress and traction . . . . .	14
3.4	Conceptual kinematic of a Euler-Bernoulli beam element . . . . .	16
3.5	Conceptual kinematic of a Timoshenko beam element . . . . .	17
3.6	Rotation of the normals about $x$ and $y$ axes considering average shear deformation. . . . .	20
4.1	Various element specifications: Circle represents the point at which coordinate is specified; Square represents the point at which field quantity is specified. (a) Isoparametric, (b) superparametric, (c) subparametric. . . . .	27
5.1	Basis functions for knot vector $\Xi = \{0, 0, 0, 0.5, 1, 1, 1\}$ of degree 2 . . . . .	34
5.2	Basis functions for knot vector $\Xi = \{0, 0, 0, 0.5, 0.5, 1, 1, 1\}$ of degree 2 . . . . .	35
5.3	B-spline curve in $\mathbf{R}^1$ for control points $\mathbf{B} = [0, -1, 1, 0]$ and knot vector $\Xi = \{0, 0, 0, 0.5, 1, 1, 1\}$ of degree 2. . . . .	36
5.4	B-spline curve corresponding to knot vector $\Xi = \{0, 0, 0, 0.5, 1, 1, 1\}$ of degree 2 and the control points $\mathbf{B}_x = [0, 3, 2, 1]$ and $\mathbf{B}_y = [0, 5, 0, 5]$ . . . . .	37
5.5	B-spline curve corresponding to knot vector $\Xi = \{0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1\}$ of degree 2 and the control points indicated by red squares. . . . .	37
5.6	Basis functions corresponding to knot vector $\mathcal{H} = \{0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1\}$ of degree 2. . . . .	39
5.7	B-spline surface for the control net $\mathbf{B}_{i,j}$ given in table 5.1 and knot vectors $\Xi = \{0, 0, 0, 0.5, 1, 1, 1\}$ and $\mathcal{H} = \{0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1\}$ of degree 2. . . . .	39
5.8	Knot refinement from $\Xi = \{0, 0, 0, 0.5, 1, 1, 1\}$ to $\tilde{\Xi} = \{0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1\}$ . Notice that the curve remains unchanged. . . . .	41

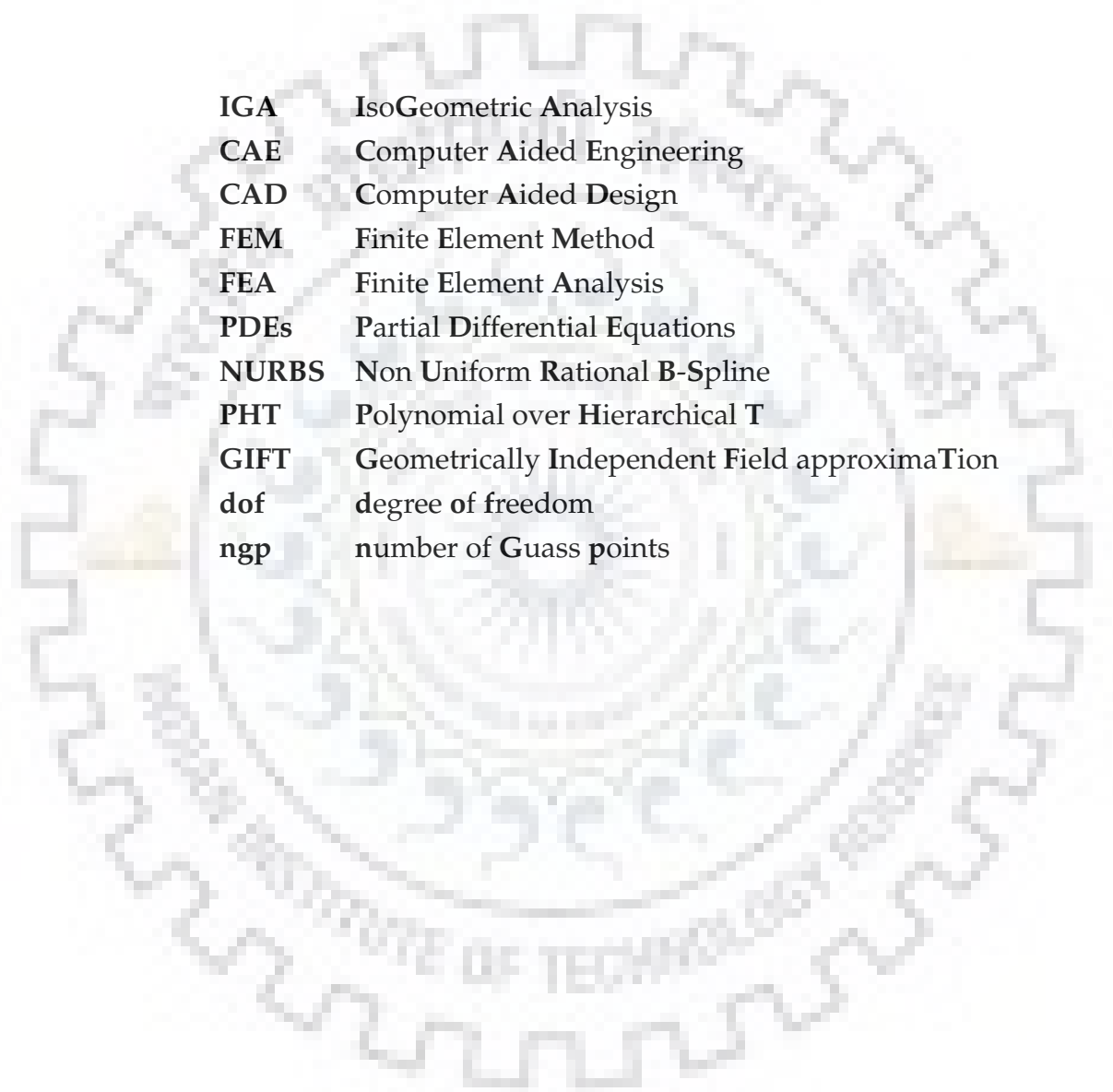
5.9	Projective transformation from a quadratic B-spline in $\mathbf{R}^3$ to a circle in $\mathbf{R}^2$ . The figure is reconstructed from Figure 2.28 in [6]. . . . .	43
6.1	Different spaces in framework of IGA . . . . .	48
6.2	Flow chart for one patch IGA. The routines in blue differ from those in classical FEA. . . . .	50
7.1	A cantilever beam and meshes: (a) a cantilever beam with parameters; (b) a uniformly coarse initial mesh used for both NURBS and PHT-splines based IGA; Mesh after one refinement: (c) NURBS and (d) PHT-splines; Mesh after two refinement: (e) NURBS and (f) PHT-splines . . . . .	54
7.2	An example of T-mesh; B1 to B10 are boundary vertices; V1 to V4 are T-junctions; C1 is a crossing vertex . . . . .	56
7.3	A hierarchical T-mesh . . . . .	56
7.4	Quadtree structure illustrating the hierarchical data organization of an adaptive mesh. Pointers represents the neighbouring relations of element (in red colour) at a particular hierarchical level. . . . .	57
7.5	Cubic B-spline basis functions with multiplicity of two of interior knots, that is, only $C^1$ continuity: $\Xi = \{0, 0, 0, 0, 1/3, 1/3, 2/3, 2/3, 1, 1, 1, 1\}$ . . . . .	58
7.6	The 9 regions into which each child element is subdivided. The axis labels show the number of Bézier coefficients in each direction. . . . .	61
7.7	Illustration of error estimation, marking and refinement procedure for adaptive PHT element. . . . .	65
8.1	Straight bar element . . . . .	67
8.2	Bar element in parametric space . . . . .	67
8.3	Support of basis functions for knot vector $\Xi = \{0, 0, 0, 1/3, 2/3, 1, 1, 1\}$ of degree 2 . . . . .	68
8.4	Straight beam element. . . . .	71
8.5	4-Noded plate bending element with 3 dof at each node . . . . .	73
9.1	Straight bar element in physical space with $L = 1$ . . . . .	76
9.2	Fixed-fixed rod problem: normalized discrete spectra using NURBS. . . . .	77
9.3	Straight beam element in physical space with $L = 1$ . . . . .	78
9.4	Fixed-fixed beam problem: normalized discrete spectra using NURBS. . . . .	79
9.5	Basic flat rectangular plate element with 3 dof at a node . . . . .	80

9.6	Various loading and support conditions on plate element: (a) Simply supported on all sides with uniformly distributed load; (b) Simply supported on all sides with uniformly varying load; (c) Clamped on all sides with uniformly distributed load; (d) Clamped on all sides with uniformly varying load . . . . .	80
9.7	The relative error for central deflection of simply supported rectangular plate under uniformly distributed loading . . . . .	82
9.8	The relative error for central deflection of simply supported rectangular plate under uniformly varying loading . . . . .	82
9.9	The relative error for central deflection of clamped rectangular plate under uniformly distributed loading . . . . .	85
9.10	The relative error for central deflection of clamped rectangular plate under uniformly varying loading . . . . .	86
9.11	Geometry description of the cantilever beam problem . . . . .	88
9.12	Components of stresses in three dimensions. . . . .	88
9.13	Relative error in energy norm vs. the degree of freedom using adaptive refinement for cantilever beam . . . . .	89
9.14	Mesh of cantilever beam at refinement level of (A) 1 <sup>st</sup> , (B) 6 <sup>th</sup> ; (C) 12 <sup>th</sup> . Different color indicates different patches used in the model. . . . .	90
9.15	Contour plots of $\sigma_{11}$ stress components of a deflected cantilever beam. (A) is the contour at 1 <sup>st</sup> refinement; (B) is the contour at 6 <sup>th</sup> refinement; (C) is the contour at 12 <sup>th</sup> refinement. . . . .	90
9.16	Contour plots of $\sigma_{12}$ stress components of a deflected cantilever beam. (A) is the contour at 1 <sup>st</sup> refinement; (B) is the contour at 6 <sup>th</sup> refinement; (C) is the contour at 12 <sup>th</sup> refinement. . . . .	91
9.17	Contour plots of $\sigma_{22}$ stress components of a deflected cantilever beam. (A) is the contour at 1 <sup>st</sup> refinement; (B) is the contour at 6 <sup>th</sup> refinement; (C) is the contour at 12 <sup>th</sup> refinement. . . . .	91

# List of Tables


4.1	Sampling point locations and weight factors for Gauss quadrature . . . .	30
5.1	Control net $\mathbf{B}_{i,j}$ . . . . .	38
6.1	Comparison of IGA and FEM summarising differences and similarities between them. Taken from [12] . . . . .	51
7.1	Super-convergent points for splines of degree $p$ and continuity $C^\alpha$ on interval $[-1, 1]$ . . . . .	63
9.1	Bar material properties . . . . .	76
9.2	Bar geometrical properties . . . . .	76
9.3	Beam material properties . . . . .	78
9.4	Beam geometrical properties . . . . .	78
9.5	Beam geometrical properties . . . . .	81
9.6	Beam material properties . . . . .	81
9.7	Central deflection for all simply supported rectangular plate with UDL .	83
9.8	Central deflection for all simply supported plate with UVL . . . . .	84
9.9	Central deflection for all clamped rectangular plate with UDL . . . . .	86
9.10	Central deflection for all clamped rectangular plate with UVL . . . . .	87
9.11	Cantilever beam material properties . . . . .	89

# List of Abbreviations



<b>IGA</b>	<b>I</b> so <b>G</b> eometric <b>A</b> nalysis
<b>CAE</b>	<b>C</b> omputer <b>A</b> ided <b>E</b> ngineering
<b>CAD</b>	<b>C</b> omputer <b>A</b> ided <b>D</b> esign
<b>FEM</b>	<b>F</b> inite <b>E</b> lement <b>M</b> ethod
<b>FEA</b>	<b>F</b> inite <b>E</b> lement <b>A</b> nalysis
<b>PDEs</b>	<b>P</b> artial <b>D</b> ifferential <b>E</b> quations
<b>NURBS</b>	<b>N</b> on <b>U</b> niform <b>R</b> ational <b>B</b> - <b>S</b> pline
<b>PHT</b>	<b>P</b> olynomial over <b>H</b> ierarchical <b>T</b>
<b>GIFT</b>	<b>G</b> eometrically <b>I</b> ndependent <b>F</b> ield approxima <b>T</b> ion
<b>dof</b>	<b>d</b> egree of <b>f</b> reedom
<b>ngp</b>	<b>n</b> umber of <b>G</b> uass <b>p</b> oints

# List of Symbols



$\nu$	poisson's ratio
$\rho$	material mass density
$\xi, \eta, \zeta$	parametric coordinates
$x, y, z$	physical coordinates
$u$	displacement vector
$\epsilon$	linear strain vector
$\sigma$	linear stress vector
$\omega_n$	natural frequency of $n^{th}$ mode
$E$	Young's modulus
$B$	strain displacement matrix
$K$	global stiffness matrix
$K^e$	element stiffness matrix
$M$	global mass matrix
$M^e$	element mass matrix
$\Xi$	knot vector in $\zeta$ direction
$N$	matrix of shape functions of a finite element
$J$	Jacobian matrix
$w$	weight in Guass quadrature
$\sigma_*$	recovered stress values in the refined mesh
$\mathbf{C}^e$	element extraction operator
$\ \cdot\ _E$	energy norm
$\ e^*\ _E$	error in energy norm

# Chapter 1

## Introduction

Isogeometric Analysis(IGA) has been introduced by T.J.R Hughes [12] in the field of structural and fluid analysis. The requirement and objectives of this study is discussed in this chapter. A brief overview is given of the current problems in the Computer Aided Engineering (CAE) industry and how it can be overcome by replacing the traditional method of analysis which is finite element method (FEM) by IGA.

### 1.1 Need for study

Usually partial differential equations(PDEs) are used to represent the space and time dependent problems. FEM, a numerical method, finds an approximate solution of these PDEs because a vast majority of these problems cannot be solved by analytical methods. Despite the fact that geometry is the foundation for an analysis, computer aided design (CAD), as is known today had its origin later than development of FEM. This is perhaps the explanation why the geometry representation in CAD and FEM differs so much. CAD files generated by designers must be translated into analysis-suitable geometries which is then meshed and becomes a suitable input to large-scale finite element analysis (FEA) codes. This task however is far from being straightforward and for engineering designs of current time, which are becoming complex in nature day by day, takes over 80% of the overall time of analysis figure 1.1. It should be noted that the CAD geometry is viewed as "exact" and a finite element mesh is only an approximation of it. Note that with "exact" it is meant "as exact the CAD modeling can be". This approximation create errors in analytical results which is termed as discretization error. FEA technique works best when the mesh is of high quality i.e. the elements are of uniform size and shape. Now this requirement can be easily satisfied for small analysis but



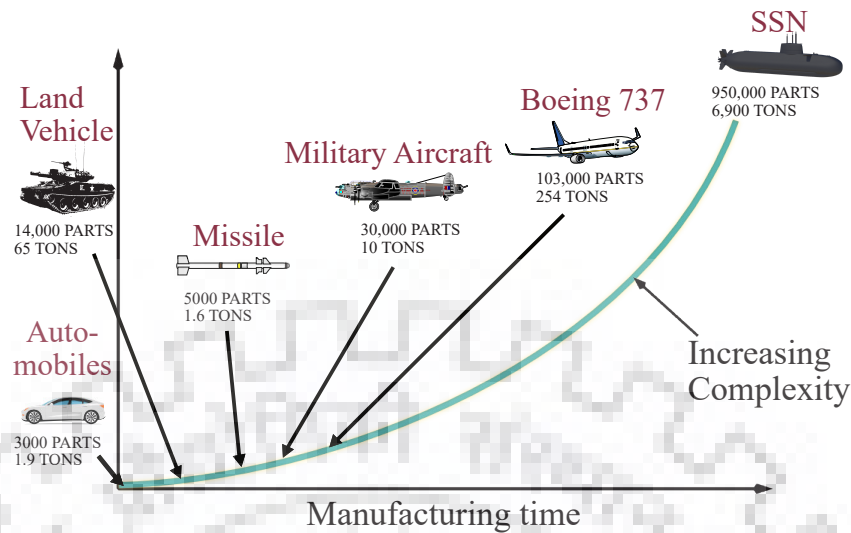


FIGURE 1.1: Engineering designs are becoming increasingly complex, making analysis a time consuming and expensive endeavour.

for complex simulations like automotive crash, this can take months of manual effort to generate a simple mesh. A way to solve this problem of such vast magnitude, ironically simple in concept, is to focus on one and only one geometry description of problem in modeling as well as analysis. IGA is an attempt by [12] which bridges the gap between CAE and CAD. The basic concept of IGA is to use the same functions between the solution space and the geometric modeling. The term (“isogeometric”) comes from the fact that the use of Non-Uniform Rational B-Splines(NURBS) leads to exact geometric description of the domain, while in standard FEA it is necessary to approximate it by means of a mesh. IGA completely reduces the need for meshing because the geometry description is now same in analysis domain and hence the geometry can simply be imported instead of meshing it again. Since the geometry can now be imported as such, the model in analysis domain is now exactly geometrically similar which improves the results and reduces the computation time. The choice of NURBS to be used is based on the fact that it is currently the most used technique to represent geometry in CAD industry.

## 1.2 Objectives of the study

The study serves to be an introduction to the IGA on linear elasticity problems. The development of IGA model for bar, beam and plate element for static and spectrum analysis was initiated with the intent to investigate the advantages of the study with respect

to the traditional finite element method used widely: these include the requirement of coarse mesh size, ability to use higher order elements, ability to represent geometry as exact and not approximate, smoother basis function, better refinement techniques and wider scope to improve the method.

The main objectives of the study can be listed as below

1. Understanding FEM which is a numerical method for solving problems of engineering.
2. Understanding mathematical method used to represent geometry such as NURBS, T-Splines etc.
3. Understanding the new analysis framework which is known as IGA.
4. Applying IGA framework to various elements and understanding the problems underlying them.
5. Exploring the domain of automatic adaptive local refinement using the framework of PHT based IGA.
6. Applying the PHT based IGA framework to a problem with stress concentration.

### 1.3 Scope of the study

In this study IGA framework (based on NURBS) prepared in Python was first used in the domain of frequency analysis. The responses of basic elements like bar and beam, which makes up a structure, to free vibration has been found using IGA. A comparison of these results with results of existing methods is made to demonstrate the effectiveness of this method. After obtaining valid results the framework was extended for the analysis of plate element. The plate element used was based on Reissner-Mindlin plate theory, which is an extension of Kirchhoff-Love plate theory, that takes into account the shear deformations throughout the thickness of a plate. Different boundary conditions for the plate element was used: all four sides simply supported or all four sides clamped. Effect of mesh size and boundary conditions on the analysis results of a 4-noded plate element by taking elastic properties and subjecting to point load is presented, and finally a comparative study of these results with analytical results [20] and FEM analysis results is presented.

NURBS based IGA is based on tensor product on basis functions because of which it is not viable to do local refinement in its framework. Lack of adaptive local refinement capability doesn't make NURBS based IGA a worthy contender to traditional FEA based softwares. Hence, PHT-Splines(Polynomial over Hierarchical T-Splines) based IGA is developed which is capable of doing local refinement in the problem domain. This framework coupled with Zienkiewicz-Zhu error estimator [23] makes the analysis procedure adaptive. The framework is applied to the problem of cantilever beam in which re-entrant corners required local refinement.

## 1.4 Organization of dissertation

This dissertation is divided into one introductory chapter, one chapter on review of existing work done on the topic, seven chapters detailing the concepts used in the work done and one concluding chapter. Chapter 2 after introduction covers a brief review of the work done already in field of IGA. Chapter 3 describes the basics underlying the whole analysis procedure. Concepts related to 2D elasticity are discussed in this chapter with various beam and plate theories relevant to the work done. Chapter 4 serves as an introduction to the FEM process with definition of the terms mentioned in the chapter showing results. Since IGA is based on similar concepts as isoparametric formulation, brief literature related to the concept is discussed. Concepts of jacobian and its formulation for first and second order derivatives are mentioned also. First order jacobian is required for free vibration analysis of bar element and bending analysis of plate element, whereas second order jacobian is required for the free vibration analysis of beam element. This chapter serves as a reference to the topics regularly mentioned throughout the dissertation.

Chapter 5 introduces the concepts of B-splines and NURBS detailing the procedure of formation of basis and curves and surfaces. The chapter starts with description of basis functions of B-splines and then proceeds to the procedure of forming B-spline curves and surfaces with knot refinement. Later, NURBS basis is derived using B-spline basis and its properties are discussed. It is the NURBS basis functions which will replace the shape functions used in FEM to form an IGA framework. The sixth chapter introduces the concepts of different spaces in the framework of IGA and how numerical integration is carried out in the framework of IGA. Chapter 6 marks the end of the introduction to NURBS based IGA. After this adaptive refinement is discussed in the next chapter.

Chapter 7 details the concepts and procedure underlying the PHT-splines based adaptive framework. The description of basis functions used and their representation using Bezier coordinates is explained first. After this the process of refinement is explained in which the procedure to form new basis functions and updating the quad tree data structure is given. Recovery based error estimator is explained which is based on Zienkiewicz-Zhu error estimator. Then the suitable marking scheme is explained which will serve the purpose of marking the elements to be refined at the next refinement step.

Chapter 8 describes the formulation of elements used in this study. Chapter starts with the formulation of bar element detailing its geometrical properties and its description in physical as well in parametric space used in NURBS based IGA. The procedure is mentioned in the form of steps, the first being its description in physical space and last being the formation of matrices using which numerical integration is carried out to compute the result. Same is discussed for beam element and plate element as well. Apart from differences in the underlying theories, the elements also differs from the description in parametric space. For example, the plate element is a  $2D$  element and is therefore formed using a tensor product structure of two knot vectors defined as perpendicular to each other. Bar and beam element are  $1D$  element and therefore only one knot vector is required to discretize the physical element.

Chapter 9 includes results of all the work done. First the results of free vibration analysis is shown of bar and beam element. Numerically computed frequencies are compared with analytically computed frequencies. After this bending analysis of plate element is shown for different boundary conditions and loading conditions. Central deflection for each of these cases is compared with analytically computed deflections and the error computed is shown in the form of graph and tables. Convergence studies is done for different mesh density and the results obtained exemplify the good nature of the proposed method. Adaptive refinement framework prepared using PHT-splines is applied to the problem of cantilever beam. The results in the form of convergence plot, meshes and stresses at different refinement levels shows good convergence while using lesser number of elements.

The dissertation ends with conclusions and suggestions for future work.

## Chapter 2

# Literature Review

### 2.1 General

IGA is integration of FEM and CAD to solve the problems underlying CAE industry. The basic concept behind IGA is to use the basis functions, used to represent geometry in CAD, to replace the shape functions used in FEM for analysis. NURBS is used in CAD to model geometries. It is the B-Spline basis functions which replaces the standard FEM Lagrange basis functions. However, replacing them is not straightforward because of their different characteristics. The first successful integration was achieved by Hughes, Cottrell, and Bazilevs [12]. Later on the research in the field of IGA increased tremendously with an aim to replace the FEA used in industries today with IGA.

A brief review of some of the previous studies is presented in this section with an aim to highlight the beginning and direction of progress of the research.

### 2.2 Literature Study

Work on IGA began in 2003 after Tom Hughes came privy to a conversation concerning the creation of finite element models from CAD representations. The first paper on this topic was first published in 2005 by Hughes, Cottrell, and Bazilevs [12]. In this study analysis framework based on NURBS was applied to linear structural and fluid problems with investigations on h-, p-, k-refinement techniques. k-refinement is only possible in IGA because of the nature of basis functions used. It also presented the



results of shell analysis with three-dimensional solid elements with convergence occurring in all the cases. The wide array of scope of work was presented which served as a pivot for all the research work that followed.

Cottrell et al. [7] were the first to successfully apply the IGA to the problem of structural vibrations. A number of elementary model problems were solved numerically and, in some cases analytically. The model consisted of rods, beams, membranes, plates, and three-dimensional solids. Rotation-less bending elements were also presented. The study found out that k-refinement provides more robust and accurate frequency spectra than typical higher order finite elements (i.e., the p-refinement technique). The study also found out that “optical” branches of frequency spectra can be eliminated through the use of nonlinear parameterizations of the geometrical mapping (Optical branches has been identified as the cause of severe degradation in accuracy in higher modes and Gibbs phenomenon in wave propagation).

Kiendl et al. [13] were the first to develop a shell element in IGA framework bases on Kirchhoff-Love theory. They found out that by using IGA the necessary continuities between elements was easily achieved. The element was formulated geometrically non-linear and discretized by displacement degree of freedoms only. Even the rotational dofs were formulated in terms of displacement dofs. Rhino (NURBS based CAD program) was used as a pre-processor for IGA. Since the shell element was defined on the shell’s middle surface only, the CAD model was directly used as a model for the analysis. The daunting task of building a 3D model from a 2D model was omitted.

Traditional IGA is based on NURBS which has a tensor product structure. Thus any attempt to refine the mesh in NURBS based IGA results in global refinement with overload of control points and dofs. One way to do local refinement is to use multiple patches which poses problems with patch continuity and patch conformance. Another way is to use suitable adaptive splines which allows for local refinement. T-Spline was a step towards this direction and was introduced by Sederberg et al. [19] in 2003. T-Spline is a generalization of B-Spline and is capable of permitting a T-junction in its control grid. Having a T-junction implies that the lines of control points need not cross the entire control grid. It breaks the rigid tensor-product structure of NURBS by inserting extra vertices into the tensor product structure. Hence, T-splines was capable of creating a single patch geometry which was water-tight and can be locally refined or coarsened.

T-Splines based IGA was first developed by Bazilevs et al. [2]. Use of T-splines made

local refinement possible. The study used the IGA using T-splines on some Fluid and structural problem. The structural problems involved numerical solutions for thin shells. But the shells were modelled as 3D solid and no shell assumptions were employed. The results obtained successfully showed the viability of T-Splines in IGA framework. Additional advantages such as local refinement and water-tight geometry was also discussed.

Borden et al. [4] presented the Bezier extraction operator and isogeometric Bezier elements for NURBS based IGA. The extraction operator allowed for numerical integration of smooth functions to be performed on  $C^0$  Bezier elements. The Bezier extraction operator and Bezier elements provided an element structure for IGA which can be easily used into finite element codes without any changes to element form and assembly algorithms, and standard data processing array. Bezier extraction operator also enabled for global basis information to be defined locally, a technique which is used in construction of PHT-spline basis. Another advantage of this is also the description of continuity between multiple patches without the need for additional arrays.

The first major implementation of the use of Bézier extraction operator was done by Lai et al. [15] by proposing the first implementation of IGA in a commercial software package, Abaqus. Although not all the modules were utilised but this marked the beginning of the implementation of CAD and a commercial analysis software which was the main aim behind the research of IGA.

After the successful implementation of T-Splines several other adaptive spline technology was developed such as Bézier extraction based T-Splines, polynomial splines over hierarchical T-meshes (PHT-splines), truncated hierarchical B-splines (THB) and locally refined (LR) B-splines. PHT-splines by [8] was a generalization of B-splines over hierarchical T-meshes. The study revealed the advantage of highly reduced requirement of control points when NURBS surface was represented by PHT-spline. Since PHT-splines are polynomial rather than rational, cross-insertion and removal of PHT-Spline was shown to be local and simple. This proved to be an additional advantage over T-Splines [19].

PHT-Splines in framework of IGA was first applied by Wang et al. [21] in 2011. The study showcased the local refinement property of PHT-splines. Residual-based posteriori error estimator was used to drive adaptivity.

Study by Anitescu, Hossain, and Rabczuk [1] developed a higher-order adaptive method

using PHT-splines. The study improved upon the existing PHT based IGA by reducing the number of control points (by truncating out Bézier coordinates) and using more efficient algorithm. The study also introduced an efficient recovery-based error estimator. The “recovered solution” produced by error estimator was shown to be a more accurate approximation than the computed numerical solution. The study was applied to various 2D and 3D problems which concluded significantly improved accuracy per dof obtained by adaptivity.

Gondegaon and Voruganti [11] presented static structural and modal analysis of basic elements: bar, beam and plate, and compared the results with that of FEM. The results verified the advantages of IGA like better solution per degree of freedom.

Study of advantages of local adaptivity in the domain of frequency analysis was done by Yu et al. [22]. The study used Geometry Independent Field approximation (GIFT) framework to describe the geometry. Within GIFT, NURBS was used to describe the exact geometry. On the other hand, PHT-spline was used to approximate the solution field. Thus solution field was described using more control points and geometry using lesser. A posteriori error estimator was used to drive adaptivity. The study achieved significantly faster convergence rates than a uniform h-refinement.

Conventionally, rotational dofs are used to represent the motion of fibers in the Reissner-Mindlin shear deformable shell theory, which results in an element with five or six dofs per node which is computationally expensive. These additional dofs becomes the source of convergence difficulties in structural analyses. Therefore, formulations based on only translational dofs are a better choice. A  $C^k$  – continuous, NURBS based IGA for shell for large deformations was presented by Benson et al. [3].

## 2.3 Summary

In spite of an extensive number of previous studies on the IGA of structural vibrations and plate element, only some conclusions of general applicability have been arrived at. The problem of local refinement is a case in point. Only a few papers have proposed a solution to these problems. An example of this being the use of PHT-splines, an adaptive refinement technique. Assessing exactly the number of aspects of behavior and their contribution in the study becomes impractical. Computational cost consideration can be termed as the main motivating factor behind the varying level of proposed



methods, but an experienced and skilled analyst is the most important aspect of the study besides the selection of an appropriate element to solve the problem.



## Chapter 3

# 2D Linear Elasticity

Most of the real life problems in the domain of structural analysis consists of materials which undergo small deformations. It is assumed that a linear relationship exists between stress and strain with material not reaching the yield point. The assumptions specified are feasible for numerous design situations, which is why using theory of linear elasticity is very practical collectively with FEM. This chapter describes with the most basic of terminology seen in the context of linear elasticity. It starts by defining quantities like stress, strain, and traction and then proceeds to discuss basic analysis theories related to beam and plate. After that, some basics about Eigenfrequency analysis is discussed.

### 3.1 Strain

Strain describes the relative amount of deformation the body has when subjected to load. It is a dimensionless quantity and is termed as fundamental quantity since it can be measured and used to derive other quantities such as stress, moments, etc.. It measures the relative displacement between particles of the material neglecting the component of displacement due to rigid body motion. The strain is derived from displacement and a displacement vector is written as

$$u = \begin{bmatrix} u_x \\ u_y \end{bmatrix} \quad (3.1)$$

Strain is defined as the amount of deformation divided by the original length. Engineering strains are therefore given by

$$\varepsilon_{xx} = \lim_{\Delta x \rightarrow 0} \frac{u_x(x + \Delta x, y) - u_x(x, y)}{\Delta x} = \frac{\partial u_x}{\partial x} \quad (3.2)$$

$$\varepsilon_{yy} = \lim_{\Delta y \rightarrow 0} \frac{u_y(x, y + \Delta y) - u_y(x, y)}{\Delta y} = \frac{\partial u_y}{\partial y} \quad (3.3)$$

whereas shear strain,  $\gamma_{xy}$ , is given by

$$\gamma_{xy} = \frac{\partial u_y}{\partial x} + \frac{\partial u_x}{\partial y} = \alpha_1 + \alpha_2 \quad (3.4)$$

Shear strain is defined as the change in angle in deformed body between unit vectors

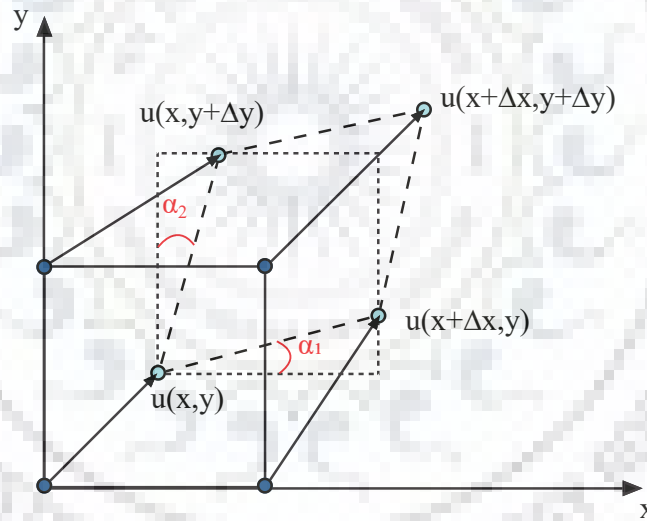


FIGURE 3.1: Deformation of a control volume

in x- and y-directions figure 3.1. The total strain is expressed as

$$\varepsilon = \begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \gamma_{xy} \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix} \quad (3.5)$$

Note that these equations are linearized and are only valid for small deformations where higher order terms can be ignored.

## 3.2 Stress

Stress is a quantity which measures force across a boundary per unit area of that boundary. Stresses in two dimensions can be specified to forces per unit area acting on the planes normal to the x-axis and y-axis. Stress can be expressed in vector form as

$$\sigma = \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{bmatrix} \quad (3.6)$$

where the first subscript denotes the boundary across which stress is being measured which is direction of the normal to the plane and the second subscript denotes the direction of the force, see figure 3.2. Note that  $\sigma_{xy} = \sigma_{yx}$ .

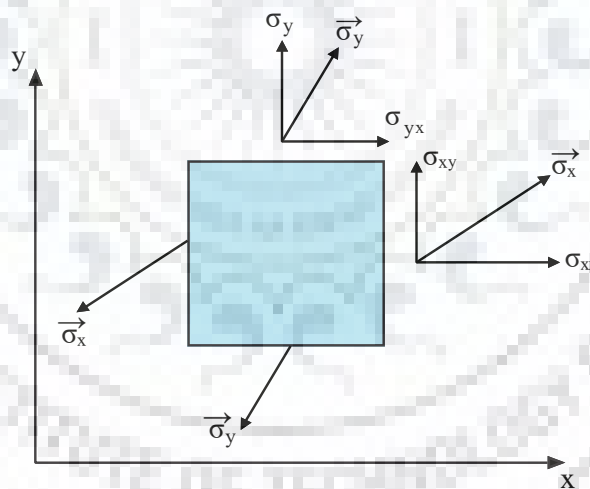


FIGURE 3.2: Stress Components

## 3.3 Traction

Traction is similar to stress and is defined as a quantity measured in force per unit area. However traction is only associated with a specific surface, such as the outer boundary

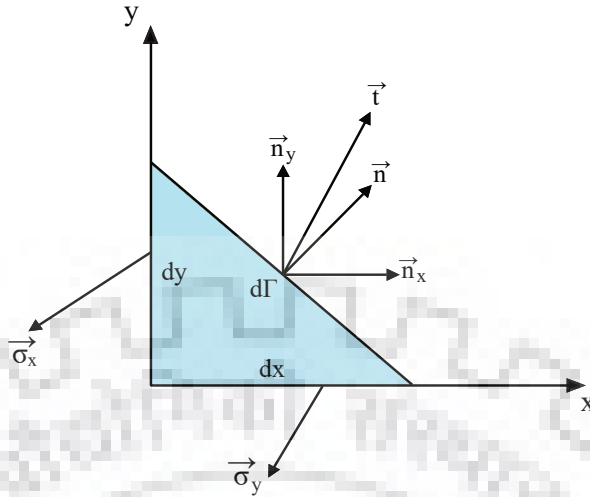


FIGURE 3.3: Relationship between stress and traction

of a domain. Traction at any point can be obtained using stress vectors at that particular point on the surface.

Relationship between traction and stress is shown in figure 3.3. In order for the forces in figure 3.3 to be in equilibrium

$$t d\Gamma - \sigma_x dy - \sigma_y dx = 0 \quad (3.7)$$

Using that  $dy = n_x d\Gamma$ ,  $dx = n_y d\Gamma$ , dividing by  $d\Gamma$  and multiplying it by unit vectors gives the expressions

$$\begin{aligned} t_x &= \sigma_{xx} n_x + \sigma_{xy} n_y = \sigma_x n \\ t_y &= \sigma_{xy} n_x + \sigma_{yy} n_y = \sigma_y n \end{aligned} \quad (3.8)$$

Above equations represent the stress boundary conditions that are given by traction condition for all points which lie on the part of boundary denoted by  $\Gamma$ .

### 3.4 Hooke's law for plane stress

Stress and strain are proportional when they are small enough quantities. The linear relationship between them is called Hooke's law and is expressed as

$$\frac{\text{Stress}}{\text{Strain}} = \text{Elastic Modulus.}$$

For problems of linear elasticity, Hooke's law is given as

$$\sigma = D\varepsilon \quad (3.9)$$

In two dimensions  $D$  will be a  $3 \times 3$  matrix, defined differently for plane strain and plane stress condition. Plane stress assumes the body to be thin compared to its dimension in the  $xy$ -plane. In other words,  $\sigma_{zz}$  can be neglected. For this dissertation plane stress condition and isotropic material is assumed using the following expression for  $D$ :

$$D = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & (1-2\nu)/2 \end{bmatrix} \quad (3.10)$$

where  $E$  and  $\nu$  are Young's modulus and Poisson's ratio respectively. Young's modulus and Poisson's ratio are constants depending on the properties of material. Note that  $D$  is symmetric positive definite. When  $\nu \neq 0$   $D$  will couple the different directions.

### 3.5 Beam Theory

Before the introduction of numerical techniques for computation, simplified theories were used to analyze some structures or components of structures. In these theories, assumptions are made about the strain and stress variation in the cross-section and thus making it possible to perform necessary approximations to the general equations. For analyzing beam two main theories exist depending on the consideration of the effects of shear deformation. Brief description about these theories in the following subsection.

### 3.5.1 Euler-Bernoulli beam

Plane perpendicular to the axis of beam before deformation is assumed to remain perpendicular after deformation. Implication of this being the rotation of cross-section of beam to be equal to the slope of the axis of beam. As explained in figure 3.4 this assumption leads to the well-recognized governing differential equation of bending of beam in which transversal displacement  $v$  is only variable. Therefore for ensuring continuity between two connected elements at point  $O$ ,  $v$  and  $\frac{dv}{dx}$  must be continuous. Point  $O$  is resting on the beam axis. From figure 3.4, during the deformation of the beam, the

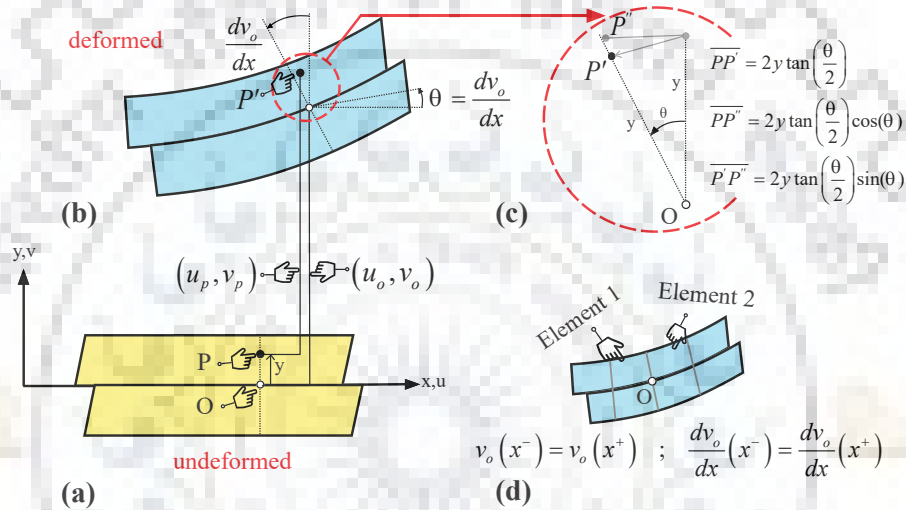


FIGURE 3.4: Conceptual kinematic of a Euler-Bernoulli beam element

displaced position and rotation of an arbitrary point  $P$  is given by:

$$\begin{aligned} u_p &= u_o - 2y \tan\left(\frac{\theta}{2}\right) \cos(\theta) \approx u_o - 2y \frac{\theta}{2} 1 \approx u_o - y\theta \\ v_p &= v_o + y - y - 2y \tan\left(\frac{\theta}{2}\right) \sin(\theta) \approx v_o - 2y \frac{\theta}{2} \theta \approx v_o \end{aligned} \quad (3.11)$$

where  $u_p$  and  $u_o$  are the horizontal displacement components of points  $P$  and  $O$ ,  $v_p$  and  $v_o$  are the transversal displacement components,  $x$  and  $y$  are the coordinates before the deformation of point  $P$ , and  $\theta$  is the rotation of the plane perpendicular to the axis of beam which in turn is equal to slope of the beam axis after deformation at point  $O$ .

### 3.5.2 Timoshenko beam

Unlike Euler-Bernoulli beam theory, the plane of cross section after deformation is not assumed to remain perpendicular to the axis of the beam. Hence, the slope of beam

axis is no longer the same as the rotation of cross-section. This assumption leads to another beam theory which is known as Timoshenko beam theory. Shear deformation is considered as a mechanism to relieve shear of the deformed cross-section. A physical interpretation of shear deformation can be considered as the resistance of the cross-section towards bending, i.e, tendency of cross-section to rotate itself back from the perpendicular to beam axis.

Continuity between two connected elements at point  $O$  can be ensured by making  $v$  and  $\theta$  to be continuous (figure 3.5d).

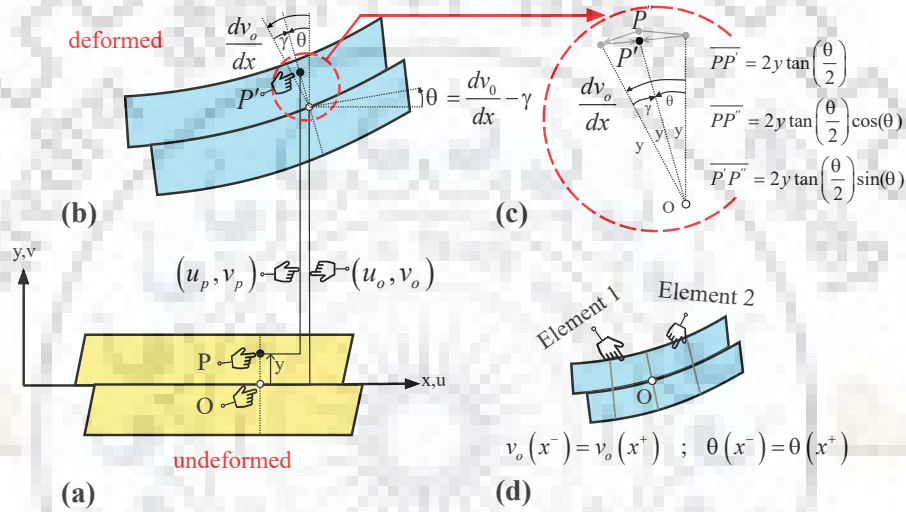


FIGURE 3.5: Conceptual kinematic of a Timoshenko beam element

From figure 3.5c, during the deformation of the beam, the displaced position and rotation of an arbitrary point  $P$  is given by:

$$\begin{aligned}
 u_p &= u_o - 2y \tan\left(\frac{\theta}{2}\right) \cos(\theta) \approx u_o - 2y \frac{\theta}{2} 1 \approx u_o - y\theta \\
 v_p &= v_o + y - y - 2y \tan\left(\frac{\theta}{2}\right) \sin(\theta) \approx v_o - 2y \frac{\theta}{2} \theta \approx v_o \\
 \theta &= \frac{dv_o}{dx} - \gamma
 \end{aligned} \tag{3.12}$$

where  $u_p$  and  $u_o$  are the horizontal displacement components of points  $P$  and  $O$ ,  $v_p$  and  $v_o$  are the transversal displacement components,  $x$  and  $y$  are the coordinates before the deformation of point  $P$ , and  $\theta$  is the rotation of the cross section which is no longer equal to slope of the beam and  $\gamma$  is the shear deformation.



## 3.6 Plate theory

For better understanding of finite element formulation, a brief review is made of the two most widely accepted and used theory of plates

1. Kirchhoff – Love theory (classical plate theory)
2. Reissner-Mindlin theory (first-order shear plate theory)

From the perspective of FEA, for Reissner-Mindlin theory, the elements with only  $C^0$  continuity are required whereas the Kirchhoff plate formulation requires elements with at least  $C^1$  continuity. Reissner-Mindlin theory is easier to implement by standard polynomial basis functions but suffers from shear locking effects, whereas Kirchhoff theory needs higher order elements. In this study Reissner-Mindlin theory is used. The formulation of Kirchhoff elements in IGA can be simplified by using k-refinement technique in IGA, which result in higher order NURBS basis functions with increased inter-element continuity. This can be done in the later stages.

### 3.6.1 Kirchhoff-Love theory

Kirchhoff's theory, also known as classical thin plate theory, is similar to beam theory by Bernoulli because of the assumptions which were then applied to plates and shells by Love and Kirchhoff. The assumptions are as follows:

1. The line initially straight and normal to the neutral axis before bending remains straight and normal after bending.
2. The normal stress in thickness direction is neglected. i.e.,  $\sigma_z = 0$ . This assumption converts the 3D problem into a 2D problem.
3. The transverse shear strains are assumed to be zero. i.e., shear strains  $\gamma_{xz}$  and  $\gamma_{yz}$  will be zero. Thus, thickness of the plate does not change during bending. The stresses normal to the plate can be neglected.

Even though Kirchhoff theory provides simple analytical solution but it has its own flaws. The incapability of plate element to rotate independently of its position of mid surface being one. This poses a problem for analysis of thick plates, for which at

boundaries transverse shear stresses are required. As higher order terms of strain-displacement relationship are necessary to be considered for problems with large deformations, the Kirchhoff theory can only be applied for problems with small deformations. Further, transverse stiffness changes as the plate deflect. Hence, the transverse stiffness can be assumed to be constant only for small deformation.

### 3.6.2 Reissner-Mindlin theory

Reissner–Mindlin plate theory is applied for analysis of thick plates (but can be applied to thin plates also), where the shear deformations are considered. By considering shear deformations it is possible to decouple rotation and lateral deflections (figure 3.6). Cross section no longer remains perpendicular to the mid surface of plate element. Following assumptions are made,

1. Plate deflections are assumed to be small.
2. Straight line normal to the plate mid-surface before deformation remains straight but not necessarily normal after deformation.
3. Stresses normal to the mid-surface are negligible.

Thus the deformation in the plane of undeformed mid surface,  $u$  and  $v$ , at a distance  $z$  from the centroidal axis are expressed by

$$u = z\theta_y \quad (3.13)$$

$$v = -z\theta_x \quad (3.14)$$

where  $\theta_y$  and  $\theta_x$  are the respective rotations of the cross-section to the neutral axis of the plate with respect to the  $y$  and  $x$  axes after deformation. The curvatures are expressed by

$$\begin{aligned} \chi_x &= \frac{\partial \theta_y}{\partial x} \\ \chi_y &= -\frac{\partial \theta_x}{\partial y} \end{aligned} \quad (3.15)$$

Further, the transverse shear strains are determined as

$$\begin{aligned}\gamma_{xz} &= \theta_y + \frac{\partial w}{\partial x} \\ \gamma_{yz} &= -\theta_x + \frac{\partial w}{\partial y}\end{aligned}\quad (3.16)$$

The constitutive relation between stresses and strain obtained [14]:

$$\begin{Bmatrix} M_x \\ M_y \\ M_{xy} \\ Q_x \\ Q_y \end{Bmatrix} = \begin{bmatrix} \frac{Et^3}{12(1-\nu^2)} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} -\frac{\partial \theta_y}{\partial x} \\ -\frac{\partial \theta_x}{\partial y} \\ \frac{\partial \theta_y}{\partial y} - \frac{\partial \theta_x}{\partial x} \\ \theta_y + \frac{\partial w}{\partial x} \\ -\theta_x + \frac{\partial w}{\partial y} \end{Bmatrix}^T + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{Et}{2(1+\nu)} \begin{bmatrix} \alpha & 0 \\ 0 & \alpha \end{bmatrix} \end{bmatrix} \begin{Bmatrix} \alpha \\ \alpha \end{Bmatrix} \quad (3.17)$$

The above relation is similar to the stress-strain relationship. Thus, the stress resultants becomes similar to stresses and the curvature and shear deformations become similar to strain.

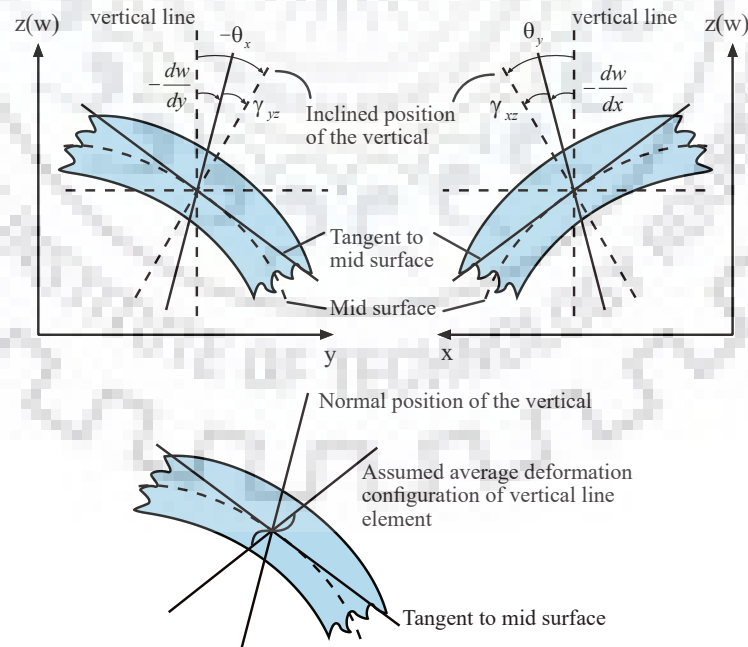


FIGURE 3.6: Rotation of the normals about  $x$  and  $y$  axes considering average shear deformation.

The strain displacement relation can be obtained as:

$$\{\varepsilon\}_p = \begin{Bmatrix} \chi_x \\ \chi_y \\ \chi_{xy} \\ \gamma_{xz} \\ \gamma_{yz} \end{Bmatrix} = \begin{bmatrix} 0 & 0 & \frac{\partial N_i}{\partial x} \\ 0 & -\frac{\partial N_i}{\partial y} & 0 \\ 0 & -\frac{\partial N_i}{\partial x} & \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial x} & 0 & N_i \\ \frac{\partial N_i}{\partial y} & -N_i & 0 \end{bmatrix} \{d\}_i \quad (3.18)$$

Here " $\alpha$ " characterizes the restraint of cross section against warping. If section completely restrains against warping then it means that there is no warping and value of  $\alpha = 1$ . If there is no restraint by cross section then  $\alpha = 2/3$ . Here it is assumed for the cross section to restrain the warping partially and hence  $\alpha = 5/6$ .

### 3.7 Natural vibration frequencies and modes

The aim of this section is to concisely revive the main equations for structural vibrations. Through Newton's second law of motion, the combination of inertia with elasticity gives rise to differential equations with time derivatives of second order. These equations possess typically wave type solutions. The resulting equation system presents an eigenvalue problem when suitable boundary condition and a harmonic solution is assumed. Solving the eigenvalue problem gives a set of eigenvalues called Eigen frequencies or natural frequencies which are the frequencies at which an elastic structure tends to vibrate. For each Eigen frequencies, the corresponding deformation pattern is called Eigen-mode. Given a mdof linear structural system, the undamped, unforced, equations of motion is:

$$M\ddot{u} + Ku = 0 \quad (3.19)$$

where  $K$  and  $M$  represents the global stiffness and mass matrix respectively.  $u = u(x, t)$  is the displacement vector and  $\ddot{u} = \frac{d^2u}{dt^2}$  is the acceleration vector. The free vibrations of the system in its  $n^{th}$  natural mode can be described by:

$$u(x, t) = \phi_n(x)q_n(t) \quad (3.20)$$

where, depending on the  $n^{\text{th}}$  natural frequency  $\omega_n$ ,  $q_n(t)$  is a harmonic function and  $\phi_n$  is the  $n^{\text{th}}$  natural mode vector.  $q_n(t)$  is of the form:

$$q_n(t) = A_n \cos(\omega_n t) + B_n \sin(\omega_n t) \quad (3.21)$$

Combining equations (3.20) and (3.21) gives:

$$u(x, t) = \phi_n(x) (A_n \cos(\omega_n t) + B_n \sin(\omega_n t)) \quad (3.22)$$

which yields:

$$\ddot{u} = -\omega_n^2 u \quad (3.23)$$

Substituting equation (3.23) into the equations of motion equation (3.19) gives the following linear system:

$$\left( K - \omega_n^2 M \right) \phi_n q_n = 0 \quad (3.24)$$

Asking for nontrivial solutions of this linear system gives rise to the generalized eigenvalue problem:

$$\det \left( K - \omega_n^2 M \right) = 0 \quad (3.25)$$

Solving equation (3.25) yields the natural frequencies  $\omega_n$  where  $n = 1, \dots, N$ ,  $N$  being the number of dofs of the system, associated to the natural modes  $\phi_n$ . After finding out a natural frequency  $\omega_n$ , the corresponding natural mode is found out by solving the following linear system for  $\phi_n$ :

$$\left( K - \omega_n^2 M \right) \phi_n = 0 \quad (3.26)$$

In conclusion, in order to apply the concepts of IGA to solve problems of structural vibrations, the following steps are needed to perform:

1. Assembling  $K$  as proposed;
2. Assembling  $M$  in a similar manner;

3. Solving the eigenvalue problem.

A more detailed description of the above steps in context of IGA can be found in section 8.1.



## Chapter 4

# Finite Element Method

The chapter starts with basic theory behind the displacement based version of FEM, in which the primary unknown is the displacement. In the first section some definitions are reviewed that are essential to the process of FEA. In the next section technique to convert a strong form to weak form is discussed. Then basic concept of Isoparametric element, Jacobian and Quadrature are discussed which is essential to the process of numerical integration in FEM. At last a basic summary of the whole process of FEM is presented.

### 4.1 Some definitions

*Strong form of equation:* A governing differential equation in its original form together with boundary conditions state the problem in strong form. Strong form states that the conditions must be satisfied at every point in the problem domain. There may exist a solution which satisfies the requirement of sufficiently smooth problem domain in addition to restrictions to boundary conditions, but getting an analytical expression for such a solution is impossible. Hence it is preferred to convert strong form to weak form by a class of techniques known as numerical methods. One of such technique is discussed in detail in section 4.3.

*Weak form of equation:* Weak form of a problem is an integral expression that implicitly contains the governing differential equation. Instead of satisfying condition at every point, weak form requires the solution to satisfy the problem only in an average or integral sense. Solving the weak form of problem gives an approximate solution and is the preferred way in context of FEM.

Boundary conditions: To complete the formulation of the problem, appropriate boundary conditions must be applied. These are:

1. Dirichlet Condition – It is also known as essential boundary condition. Essential boundary conditions are prescribed values of nodal dofs. In context of a solid mechanics problem, displacements are usually the field variable on which Dirichlet conditions are imposed. If Dirichlet boundary conditions are required to be imposed at any point other than end points/curves of the domain, special techniques are required which are penalty method, Lagrange multiplier method and least squares minimization.
2. Neumann Condition – It is also known as non-essential (or natural) boundary condition. Non-essential boundary conditions are prescribed values of higher derivatives of field quantity. They are called natural because they automatically arise in the variational statement of a problem. In context of solid mechanics, conditions are imposed on values of forces or stress.

For example, for a pipe fixed at both ends with an internal pressure acting on it, the fixed displacements and rotations at both ends will be Dirichlet condition and internal pressure will be Neumann condition.

## 4.2 Galerkin method

The approach to solving a problem with FEM is based on writing it in weak or variational form. There are two ways by which a strong form of problem can be converted to a weak form. One is using applying virtual work or variational arguments to functional such as potential energy. Second is using weighted residual methods, of which Galerkin method is discussed here. Similar to a variational statement of a problem, a Galerkin statement also incorporates governing differential equation in its weak form, which means they are satisfied over a domain in an average or integral sense. For ease, the following notation is adopted to explain Galerkin method [5].

$x$	independent variables, for example, coordinates of a material point
$u = u(x)$	dependent variables, for example, displacements of a material point
$\tilde{u} = \tilde{u}(x)$	an approximate solution
$f$	a function of $x$
$D$	a differential operator



The mathematical statement of a physical problem is

$$\text{In domain } V : Du - f = 0 \quad (4.1)$$

Equation (4.1) states the problem in strong form. In general an approximating function  $\tilde{u}$  does not satisfy equation (4.1) at every point. Thus a residual  $R = R(x)$  remains:

$$\text{Residual in domain } V : R = D\tilde{u} - f \quad (4.2)$$

$\tilde{u}$  is taken as a linear combination of basis functions, that is,  $\tilde{u} = \sum_{i=1}^n N_i \tilde{u}_i$ , where  $N_i$  is the  $i^{\text{th}}$  basis function on an element and  $\tilde{u}_i$  is the nodal dof. According to weighted residual method, values of  $N_i$  that are best satisfy the following expression of governing equations in their weak form.

$$\int W_i R dV = 0 \quad \text{for } i = 1, 2, \dots, n \quad (4.3)$$

where each  $W_i = W_i(x)$  is a weight function. In the Galerkin weighted residual method each  $W_i$  is same as  $N_i$ . Equation (4.3) is then integrated by parts which serves the purpose of reducing the order of differentiation as well as introducing natural or Neumann boundary condition in the formulation.

### 4.3 Isoparametric concept

As already discussed FEA starts with approximating solution as a polynomial but being approximated as a polynomial has its disadvantages. Hence shape functions are then used to interpolate nodal values of dofs at any point in the problem. These shape functions can be defined in the original coordinate itself, that is, physical space or it can be defined using parameters which is parametric space. Defining shape functions using parameters enables to use elements which are curved, thus better approximating geometry. The analysis is called Isoparametric when same shape functions are used to approximate solution as well as geometry in the analysis domain. If more nodes are used to define variation of field quantity than geometry, then it is called sub-parametric and if more nodes are used to approximate geometry than it is called super-parametric. The use of parametric space to define space functions leads to another step in the numerical integration process which is called Jacobian. Jacobian is discussed in more detail in the next section.

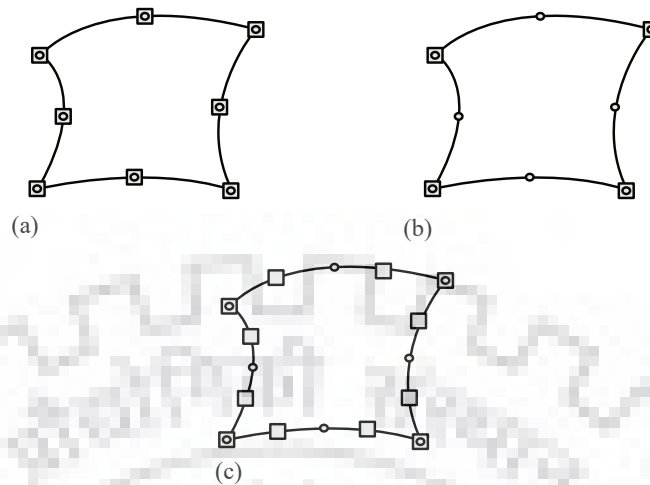


FIGURE 4.1: Various element specifications: Circle represents the point at which coordinate is specified; Square represents the point at which field quantity is specified. (a) Isoparametric, (b) superparametric, (c) subparametric.

## 4.4 Jacobian

FEA involves two spaces. One is physical space (also called global space) where the actual approximated geometry is defined in, let's say, Cartesian coordinates. Second is parametric space (also called local space) where the parameters which are used to define shape functions are defined. In elasticity problems shape functions as well as their derivatives are needed to find the solution. In general, two transformations are needed. In the first place, as shape functions are defined in terms of local (parametric) coordinates, it is necessary to devise some means of expressing global derivatives in terms of local derivatives. In the second place the element of volume (or surface) over which integration has to be carried out needs to be expressed in terms of local coordinates with an appropriate change of limits of integration. The process of this conversion is called Jacobian. Jacobian gives a relation between how space changes around a point defined in different coordinate system. Finding this relation is essential because the problem being solved is a partial differential equation.

### First order Jacobian

Consider, for instance, the set of parametric coordinates  $\xi, \eta, \zeta$  and a corresponding set of physical coordinates  $x, y, z$ . By using partial differentiation the  $\xi$  derivative can

be written as

$$\frac{\partial N_i}{d\zeta} = \frac{\partial N_i}{dx} \frac{\partial x}{d\zeta} + \frac{\partial N_i}{dy} \frac{\partial y}{d\zeta} + \frac{\partial N_i}{dz} \frac{\partial z}{d\zeta} \quad (4.4)$$

Performing the partial differentiation with respect to the other two parametric coordinates gives

$$\begin{pmatrix} \frac{\partial N_i}{d\zeta} \\ \frac{\partial N_i}{d\eta} \\ \frac{\partial N_i}{d\zeta} \end{pmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \zeta'} & \frac{\partial y}{\partial \zeta'} & \frac{\partial z}{\partial \zeta'} \\ \frac{\partial x}{\partial \eta'} & \frac{\partial y}{\partial \eta'} & \frac{\partial z}{\partial \eta'} \\ \frac{\partial x}{\partial \zeta'} & \frac{\partial y}{\partial \zeta'} & \frac{\partial z}{\partial \zeta'} \end{bmatrix} \begin{pmatrix} \frac{\partial N_i}{dx} \\ \frac{\partial N_i}{dy} \\ \frac{\partial N_i}{dz} \end{pmatrix} = J \begin{pmatrix} \frac{\partial N_i}{dx} \\ \frac{\partial N_i}{dy} \\ \frac{\partial N_i}{dz} \end{pmatrix} \quad (4.5)$$

In the above, the left hand side equation is possible to directly evaluate because the functions  $N_i$  are specified in parametric coordinates. Further, as a explicit relationship exists between  $x$ ,  $y$ ,  $z$  and the parametric coordinates, the matrix  $J$  can be found explicitly in terms of parametric coordinates. This matrix is known as the Jacobian matrix. It can also be regarded as a scale factor that multiplies to the infinitesimal element in parametric space,  $d\zeta d\eta$  to produce physical area increment in physical space,  $dx dy$ .

To find the derivatives in physical space  $J$  is inverted and written as

$$\begin{pmatrix} \frac{\partial N_i}{dx} \\ \frac{\partial N_i}{dy} \\ \frac{\partial N_i}{dz} \end{pmatrix} = J^{-1} \begin{pmatrix} \frac{\partial N_i}{d\zeta} \\ \frac{\partial N_i}{d\eta} \\ \frac{\partial N_i}{d\zeta} \end{pmatrix} \quad (4.6)$$

To transform the variables and the space with respect to which the integration is made, determinant of jacobian needs to be found out. Hence to convert a infinitesimal volume element in parametric space to a volume element in physical space, determinant of jacobian is multiplied. Thus a volume element becomes

$$dx dy dz = \det(J) d\zeta d\eta d\zeta \quad (4.7)$$

This type of transformation is valid irrespective of the number of coordinates used.

### Second order Jacobian

Consider the set of parametric coordinates  $\xi, \eta$  and a corresponding set of physical coordinates  $x, y$ . By the usual rules of partial differentiation the second order derivative of a function  $f$  can be written as

$$\begin{Bmatrix} \frac{\partial^2 f}{\partial x^2} \\ \frac{\partial^2 f}{\partial y^2} \\ \frac{\partial^2 f}{\partial x \partial y} \end{Bmatrix} = \begin{bmatrix} \left(\frac{\partial x}{\partial \xi}\right)^2 & \left(\frac{\partial y}{\partial \xi}\right)^2 & 2\frac{\partial x}{\partial \xi} \frac{\partial y}{\partial \xi} \\ \left(\frac{\partial x}{\partial \eta}\right)^2 & \left(\frac{\partial y}{\partial \eta}\right)^2 & 2\frac{\partial x}{\partial \eta} \frac{\partial y}{\partial \eta} \\ \frac{\partial x}{\partial \xi} \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \xi} \frac{\partial y}{\partial \eta} & \frac{\partial x}{\partial \eta} \frac{\partial y}{\partial \xi} + \frac{\partial y}{\partial \eta} \frac{\partial x}{\partial \xi} \end{bmatrix}^{-1} \quad (4.8)$$

$$\begin{pmatrix} \left\{ \frac{\partial^2 f}{\partial \xi^2} \right\} \\ \left\{ \frac{\partial^2 f}{\partial \eta^2} \right\} \\ \left\{ \frac{\partial^2 f}{\partial \xi \partial \eta} \right\} \end{pmatrix} - \begin{bmatrix} \frac{\partial^2 x}{\partial \xi^2} & \frac{\partial^2 y}{\partial \xi^2} \\ \frac{\partial^2 x}{\partial \eta^2} & \frac{\partial^2 y}{\partial \eta^2} \\ \frac{\partial^2 x}{\partial \xi \partial \eta} & \frac{\partial^2 y}{\partial \xi \partial \eta} \end{bmatrix} \begin{Bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{Bmatrix}$$

If differentiation has to be carried out only in 1 direction then the above equation can be simplified to

$$\left\{ \frac{\partial^2 f}{\partial x^2} \right\} = \left[ \left( \frac{\partial x}{\partial \xi} \right)^2 \right]^{-1} \left( \left[ \frac{\partial^2 f}{\partial \xi^2} \right] - \left[ \frac{\partial^2 x}{\partial \xi^2} \right] \left\{ \frac{\partial f}{\partial x} \right\} \right) \quad (4.9)$$

## 4.5 Quadrature

Integrations are needed to compute and assemble the stiffness and mass matrix and the technique to integrate numerically is called quadrature. There are many types of quadrature techniques but the most accurate for polynomial expressions is Gauss-Legendre quadrature. Gauss quadrature tables are tabulated over the range of  $-1 \leq \xi \leq 1$  (hence the reason why many shape functions are chosen in this interval).

Gaussian quadrature integrates a function as

$$\int_{-1}^1 \int_{-1}^1 \int_{-1}^1 f(\xi, \eta, \zeta) d\xi d\eta d\zeta = \sum_{j=1}^n \sum_{k=1}^n \sum_{l=1}^n f(\xi_j, \eta_k, \zeta_l) w_j w_k w_l \quad (4.10)$$

which comes as exact if polynomials being integrated are less than order  $2n$ . Table 4.1 lists the location of sampling points and weights.

TABLE 4.1: Sampling point locations and weight factors for Gauss quadrature

Order	Sampling point locations	Weight factors
$n$	$\xi_j$	$w_j$
1	0	2
2	$\pm 1/\sqrt{3}$	1
3	$\pm\sqrt{0.6}$	5/9
	0	8/9
4	$\pm\sqrt{3/7 + (2/7)\sqrt{6/5}}$	$(18 - \sqrt{30})/36$
	$\pm\sqrt{3/7 - (2/7)\sqrt{6/5}}$	$(18 + \sqrt{30})/36$

As a general rule, a polynomial of degree  $2n - 1$  is exactly integrated by  $n$  point gauss quadrature in FEM. In IGA  $n$  degree polynomial is exactly integrated by  $n + 1$  point gauss quadrature [6].

## 4.6 Summary of FEM process

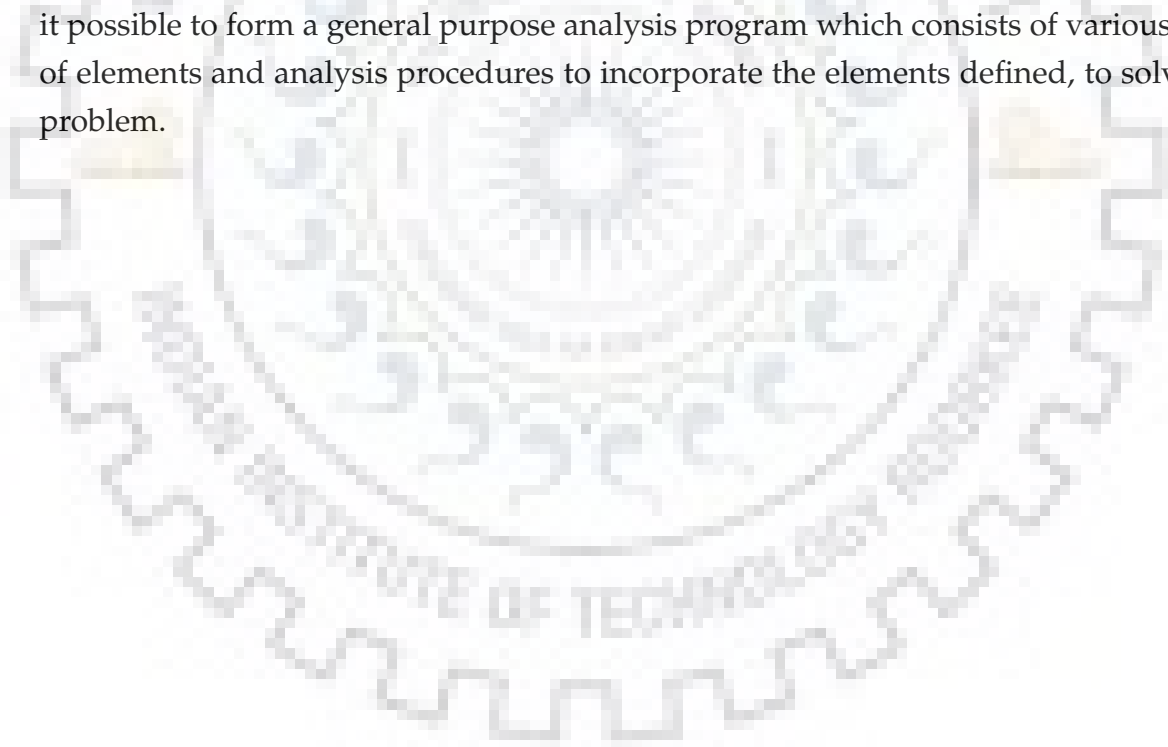
The fundamental concept of FEM, as the name suggests, is that a structure may be broken into smaller elements of finite dimensions called 'finite elements'. Each of these finite elements is assumed to be connected at a finite number of points called 'Nodes' (or nodal points). The structure is considered an assemble of these finite elements connected together at nodes. This step discretizes the original domain and becomes the source of error in numerical computation.

To obtain a solution for the whole domain, which earlier was not possible, the properties of finite elements are defined and computed. These formulations are linked together to obtain a formulation for the whole structure. Hence formulating property for an elements becomes an essential task. Properties of elements are defined at nodes. To approximate these properties throughout the element domain, 'shape functions' are required. For example, displacement within an element is approximated in terms of displacement value assumed at nodes. Nodal displacements then can be used to find

out the values of strains and stresses within an element. Then the principle of minimum potential energy or method of weighted residuals is used to derive the equations of equilibrium for the element in which displacements at nodes will be the unknowns.

To obtain the equation of equilibrium for the entire structure the equilibrium equation defined for the finite elements are combined in a sense that a certain continuity of field variable is maintained at the inter-element boundary. The amount of continuity depends on the problem being solved and the theory being used to solve the problem. The necessary Dirichlet and Neumann boundary conditions are imposed and the equations of equilibrium are solved for nodal displacements. Having obtained the values of nodal displacements for each element, the strains and stresses can be evaluated.

Thus, rather than solving the problem for the entire structure in one operation, interest is mainly devoted to the formulation of properties of the constituent elements. The steps to combine the elements, solve the global equilibrium equation, and evaluate the strains and stresses are the same for all types of structural systems. Hence FEM makes it possible to form a general purpose analysis program which consists of various types of elements and analysis procedures to incorporate the elements defined, to solve any problem.



## Chapter 5

# B-Splines and NURBS

In CAD, NURBS are generally used to draw geometries, and in CAE, the classical FEM is commonly used as the analysis tool. The main idea in IGA is to use NURBS as basis functions in both design and in the FEM. NURBS are, as the name indicates, built from B-splines. To get a better understanding of NURBS the chapter starts with B-splines and its definition. Toward the end of this chapter, NURBS will be discussed which is the technique to be used in the IGA framework. The degree and the order of a polynomial are referred to as the same quantity, that is, a quadratic polynomial is of both order and degree two.

### 5.1 B-Splines

B-splines are created from piece-wise polynomial functions which are defined on different connected intervals which are non-overlapping. B-splines are smooth, continuous and differentiable functions within each subinterval. But on the other hand, across connected subintervals, they are continuous to a certain extent but may not be differentiable.

#### 5.1.1 Knot Vectors

A B-spline is made up using  $n$  piece-wise polynomial basis functions of degree  $p$ . Knot vector is used to define the piece-wise polynomial basis function. A knot vector,  $\Xi$ , is made up of knots, which represents the sub-intervals connected using knots. The knots represents the boundaries of these sub-intervals. A knot vector is a 1D set of knots, non-decreasing in nature,  $\Xi = \{\xi_1, \xi_2, \dots, \xi_{n+p+1}\}$ , where  $\xi_i \in R$  is the  $i^{th}$  knot,  $i$  is the



knot index,  $i = 1, 2, \dots, n + p + 1$ ,  $p$  is the polynomial order and  $n$  is the number of basis function [6].

The knot-spans marks the boundaries of elements inside which the basis functions constructed for that and supporting knot interval are smooth, that is,  $C^\infty$ . Across knots or element boundaries, however the basis functions are  $C^{p-m}$ , where  $p$  is the degree of the polynomial and  $m$  is the times that particular knot is repeated. Repeating a knot leads to reduction in continuity at that particular knot value, and if sufficient times the knot is repeated the curve can be split into two at that knot. This property is frequently used in CAD. An open knot vector is a knot vector in which the first and last knot is repeated  $p + 1$  times. An open knot vector forces the basis functions to interpolate at the ends of intervals. In general, the basis functions do not interpolate the knots on interior.

### 5.1.2 Basis Functions

The B-spline basis functions are defined in a recursive fashion [16] by

$$N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi) \quad (5.1)$$

for  $p = 1, 2, 3, \dots$ . For  $p = 0$

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

with the condition that

$$\frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} \equiv 0 \quad \text{if } \xi_{i+p} - \xi_i = 0, \quad (5.2)$$

$$\frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} \equiv 0 \quad \text{if } \xi_{i+p+1} - \xi_{i+1} = 0. \quad (5.3)$$

The above formula is also known as the *Cox-de Boor recursion formula*.

$i^{\text{th}}$  basis function of order  $p$  is represented by  $N_{i,p}$ , where  $i \in [1, n]$ .  $n + p + 1$  knots results in  $n$  number of basis functions. The number of basis functions can be increased



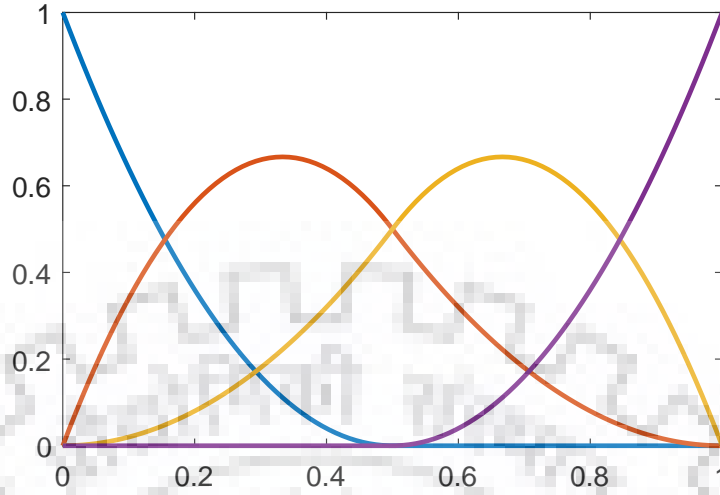


FIGURE 5.1: Basis functions for knot vector  $\Xi = \{0, 0, 0, 0.5, 1, 1, 1\}$  of degree 2

by increasing the number of knots. The basis functions formed are non-negative in magnitude and form a partition of unity, i.e.  $N_{i,p}(\xi) \geq 0 \forall \xi$ , which is a requirement to be applicable as a FEM shape function. Partition of unity states that

$$\sum_{i=1}^n N_{i,p}(\xi) = 1 \quad (5.4)$$

These basis functions also have local support and local knots.  $N_p^i(\xi) = 0$ , for knot vector  $\Xi = \{\xi_1, \xi_2, \dots, \xi_{n+p+1}\}$ , if  $\xi$  is outside the interval  $[\xi_i, \xi_{i+p+1})$ . Thus, the  $i^{\text{th}}$  B-spline  $N_p^i(\xi)$  depends only on the knots  $[\xi_i, \xi_{i+p+1})$ .

### An Example

Let the knot vector of degree  $p = 2$  be  $\Xi = \{0, 0, 0, 0.5, 1, 1, 1\}$ . With number of knots equal to 7 and polynomial degree of 2,  $n = 7 - 2 - 1 = 4$  basis functions will be required to make the curve. The knot vector being open, forces the basis functions to interpolate the boundary knots of the domain. Basis functions will be  $C^{p-m} = C^1$ -continuous across the knot of 0.5 in the interior since it is not repeated. The basis functions are shown in figure 5.1.

If the multiplicity of knots is increased by inserting another knot at 0.5, resulting in the knot vector  $\Xi = \{0, 0, 0, 0.5, 0.5, 1, 1, 1\}$ , there will be a change in continuity. The  $n = 8 - 2 - 1 = 5$  basis functions will now be  $C^{p-m} = C^0$ -continuous across knot

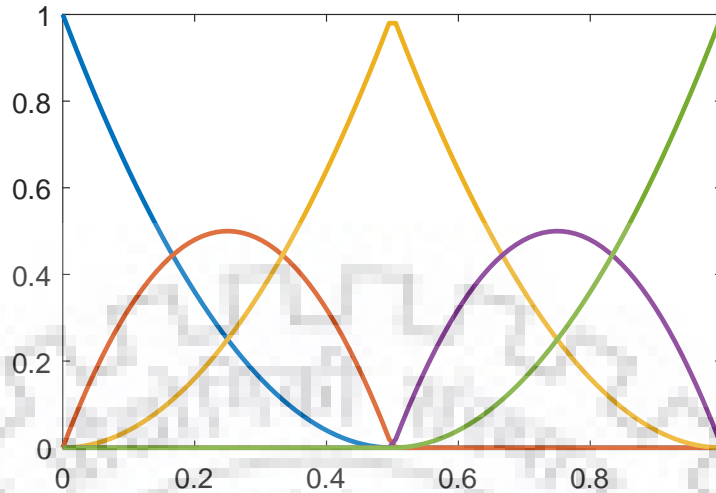


FIGURE 5.2: Basis functions for knot vector  $\Xi = \{0, 0, 0, 0.5, 0.5, 1, 1, 1\}$  of degree 2

of 0.5 in the interior of the domain. The corresponding basis functions can be seen in figure 5.2.

### 5.1.3 B-Spline curves

B-spline curves in  $\mathbf{R}^d$  are made by linear combining the B-spline basis function. The coefficients that are multiplied to the basis functions prior to adding are called control points. The resulting B-spline curve does not need to interpolate the control points. However, curve can be forced to do so by using a knot vector with sufficient multiplicity to ensure that the basis functions as well as the B-spline curve will be  $C^{p-m} = C^0$ -continuous across that particular knot. For instance, the open knot vector of degree 2,  $\Xi = \{0, 0, 0, 0.5, 1, 1, 1\}$  will force the B-spline curve to interpolate the first and the last control point. Whereas the knot vector  $\Xi = \{0, 0, 0, 0.5, 0.5, 1, 1, 1\}$  of degree  $p = 2$  the B-spline curve will interpolate the control point belonging to  $\Xi = 0.5$  too.

For  $n$  basis functions  $N_{i,p}(\xi)$ ,  $i = 1, 2, \dots, n$ , with control points  $\mathbf{B}_i \in \mathbf{R}^d$ . The corresponding B-spline curve is then given by

$$C(\xi) = \sum_{i=1}^n N_{i,p}(\xi) \mathbf{B}_i \quad (5.5)$$

This expression is equivalent to a mapping from a parameter space which is spanned by the knot vector to a physical space which is defined by the control points.

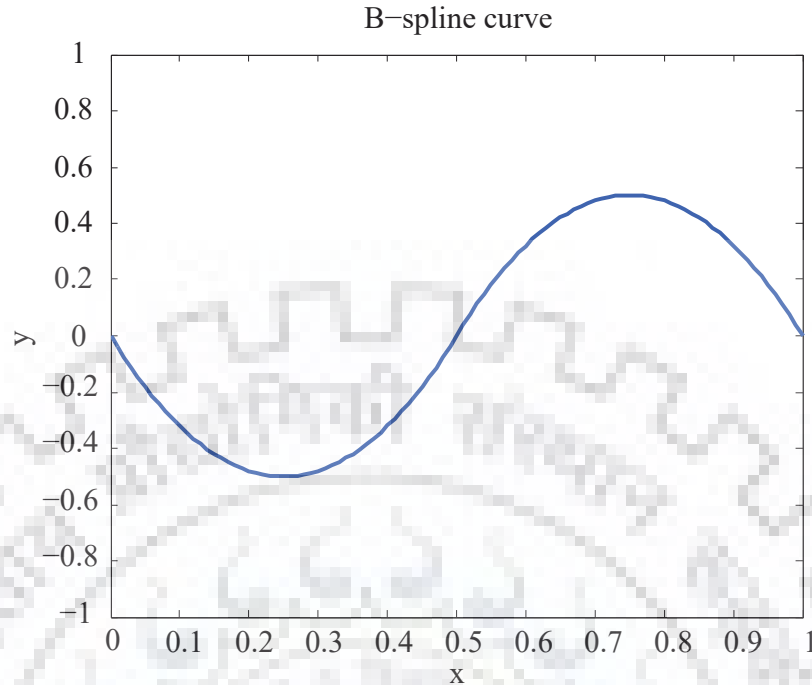


FIGURE 5.3: B-spline curve in  $\mathbf{R}^1$  for control points  $\mathbf{B} = [0, -1, 1, 0]$  and knot vector  $\Xi = \{0, 0, 0, 0.5, 1, 1, 1\}$  of degree 2.

For knot vector  $\Xi = \{0, 0, 0, 0.5, 1, 1\}$  of degree 2 and control points  $B = [0, -1, 1, 0]$  in  $\mathbf{R}^1$  the curve in figure 5.3 is obtained. Using the same knot vector but with different control points  $\mathbf{B}_x = [0, 3, 2, 1]$  and  $\mathbf{B}_y = [0, 5, 0, 5]$  in  $\mathbf{R}^2$  gives the curve which is shown in figure 5.4. The control points are indicated by red squares.

As the basis functions have local support only a particular and small part of the curve changes if the control point is changed. Figure 5.5 illustrates this change. Here the x-coordinate of first control point is changed from 2 to 0. As can be seen, only the part of the curve which is closest to the moved control point has changed. This is why B-splines technique is used predominantly in CAD programs; due to its local support of basis functions it is easy to manipulate curves by dragging the control points.

#### 5.1.4 B-Spline Surface

A B-spline surface is formed using the tensor product of basis functions in each direction. Knot vector  $\Xi$  has been defined earlier and knot vector in another orthogonal direction  $\mathcal{H} = \{\eta_1, \eta_2, \dots, \eta_{m+q+1}\}$ , where  $\eta_j \in \mathbf{R}$  is the  $j^{\text{th}}$  knot,  $m$  is the polynomial order and  $q$  is the number of basis function. In order to create a surface, a tensor product of the basis functions  $N_{i,p}(\xi)$ ,  $i = 1, 2, \dots, n$  and  $M_{j,q}(\eta)$ ,  $j = 1, 2, \dots, m$  is taken.

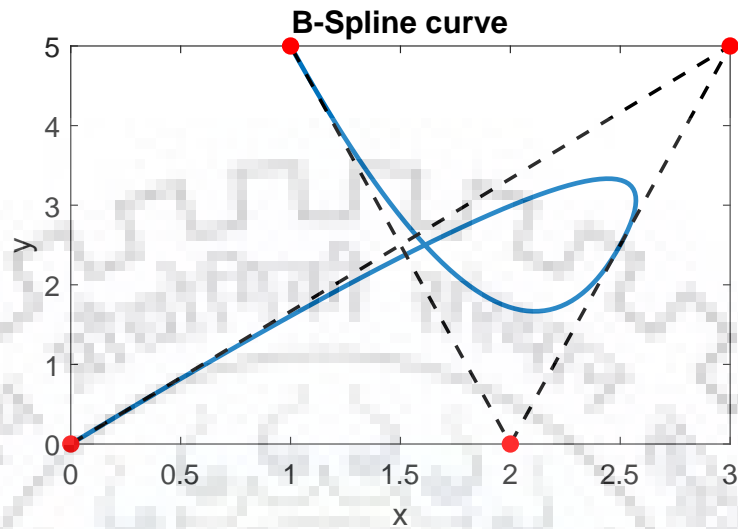


FIGURE 5.4: B-spline curve corresponding to knot vector  $\Xi = \{0,0,0,0.5,1,1,1\}$  of degree 2 and the control points  $\mathbf{B}_x = [0,3,2,1]$  and  $\mathbf{B}_y = [0,5,0,5]$ .

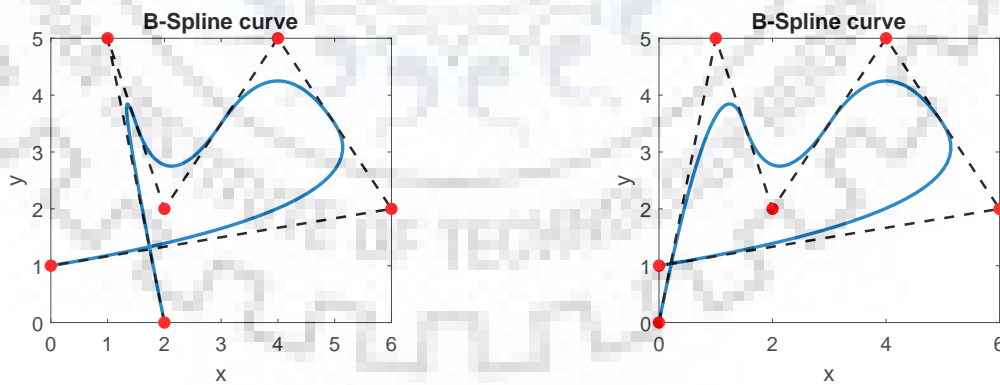


FIGURE 5.5: B-spline curve corresponding to knot vector  $\Xi = \{0,0,0,0.25,0.5,0.75,1,1,1\}$  of degree 2 and the control points indicated by red squares.

Together with the control net  $\mathbf{B}_{ij} \in \mathbf{R}^d$  the B-spline surface is given by

$$\mathbf{S}(\xi, \eta) = \sum_{i=1}^n \sum_{j=1}^m N_{i,p}(\xi) M_{j,q}(\eta) \mathbf{B}_{ij} \quad (5.6)$$

The tensor product will form a partition of unity;  $\forall(\xi, \eta) \in [\xi_1, \xi_{n+p+1}] \times [\eta_1, \eta_{m+q+1}]$

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

### An Example

For knot vector  $\mathcal{H} = \{0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1\}$  of degree  $p = 2$  the corresponding basis functions are shown in figure 5.6. These basis functions with the basis functions made by the knot vector  $\Xi = \{0, 0, 0, 0.5, 1, 1, 1\}$  of degree  $q = 2$  (figure 5.1) create a B-spline surface using the equation (5.6) for the control net given in table 5.1. The corresponding surface is shown in figure 5.7.

TABLE 5.1: Control net  $\mathbf{B}_{i,j}$

(i,j)	1	2	3	4	5	6
1	(3,0)	(3.5,1)	(3,2)	(3,3)	(3.5,4)	(3,5)
2	(2,1)	(2.5,2)	(2,3)	(2,4)	(2.5,5)	(2,6)
3	(1,0)	(1.5,1)	(1,2)	(0.5,3)	(1,4)	(1,5)
4	(0,1)	(0.5,2)	(0,3)	(0.4,4)	(0,5)	(0.2,6)

### 5.1.5 Derivatives of B-Spline Functions

Analysis using B-splines expression will require it derivatives. First and  $k^{th}$  order derivatives is given below:

$$\frac{d}{d\xi} N_{i,p}(\xi) = \frac{p}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) - \frac{p}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi) \quad (5.7)$$

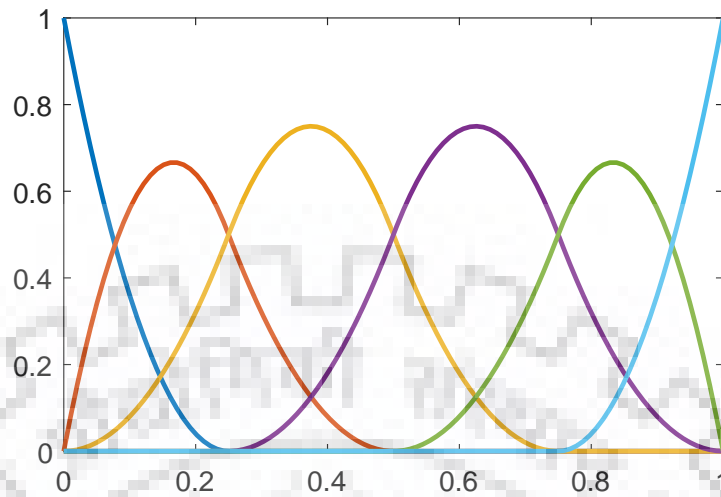


FIGURE 5.6: Basis functions corresponding to knot vector  $\mathcal{H} = \{0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1\}$  of degree 2.

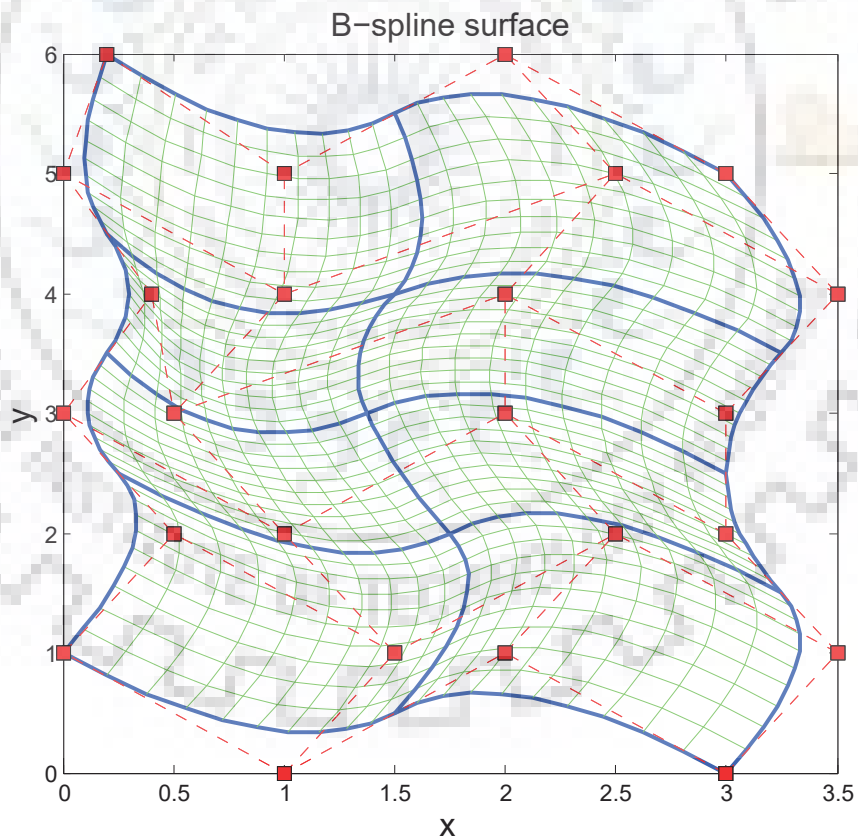


FIGURE 5.7: B-spline surface for the control net  $\mathbf{B}_{i,j}$  given in table 5.1 and knot vectors  $\Xi = \{0, 0, 0, 0.5, 1, 1, 1\}$  and  $\mathcal{H} = \{0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1\}$  of degree 2.

By differentiating equation (5.7)  $k$  times

$$\begin{aligned} \frac{d^k}{d^k \xi} N_{i,p}(\xi) &= \frac{p}{\xi_{i+p} - \xi_i} \left( \frac{d^{k-1}}{d^{k-1} \xi} N_{i,p-1}(\xi) \right) \\ &\quad - \frac{p}{\xi_{i+p+1} - \xi_{i+1}} \left( \frac{d^{k-1}}{d^{k-1} \xi} N_{i+1,p-1}(\xi) \right) \end{aligned} \quad (5.8)$$

Using equation (5.7) the right side of equation (5.8) can be expressed by lower order basis functions. A generalized expression for higher derivatives of B-splines is thus given by

$$\frac{d^k}{d^k \xi} N_{i,p}(\xi) = \frac{p!}{(p-k)!} \sum_{j=0}^k \alpha_{k,j} N_{i+j,p-k}(\xi) \quad (5.9)$$

where

$$\begin{aligned} \alpha_{0,0} &= 1 \\ \alpha_{k,0} &= \frac{\alpha_{k-1,0}}{\xi_{i+p-k+1} - \xi_i} \\ \alpha_{k,j} &= \frac{\alpha_{k-1,j} - \alpha_{k-1,j-1}}{\xi_{i+p+j-k+1} - \xi_{i+j}} \quad j = 1, \dots, k-1 \\ \alpha_{k,k} &= \frac{-\alpha_{k-1,k-1}}{\xi_{i+p+1} - \xi_{i+k}} \end{aligned}$$

### 5.1.6 Refinement; Knot insertion

Refining a B-spline curve means increasing the smoothness of the curve. Refinement can be done by either inserting additional knots, increasing the order of basis functions or both. The method of inserting additional knots is called h-refinement and increasing the order of basis functions is referred to as p-refinement. The method in which the order as well knots are increased is known as k-refinement. In this dissertation only knot insertion is considered. [6] contains in more detail the 3 types of refinement.

It is possible to improve the basis function without changing the geometry of the curve. This is done using the technique of knot insertion. Let the starting knot vector is  $\Xi = \{\xi_1, \xi_2, \dots, \xi_{n+p+1}\}$  with  $n$  respective basis function and the control points  $\mathbf{B} = \{\mathbf{B}_1, \mathbf{B}_2, \dots, \mathbf{B}_n\}$ . By inserting more knots,  $\Xi$  can be extended to a knot vector  $\tilde{\Xi} = \{\tilde{\xi}_1 = \xi_1, \tilde{\xi}_2, \dots, \tilde{\xi}_{n+m+p+1} = \xi_{n+p+1}\}$ . The set of basis functions will now be extended to consist of  $n + m$  basis functions.  $\mathbf{B}$  also needs to be extended to contain  $n + m$  control

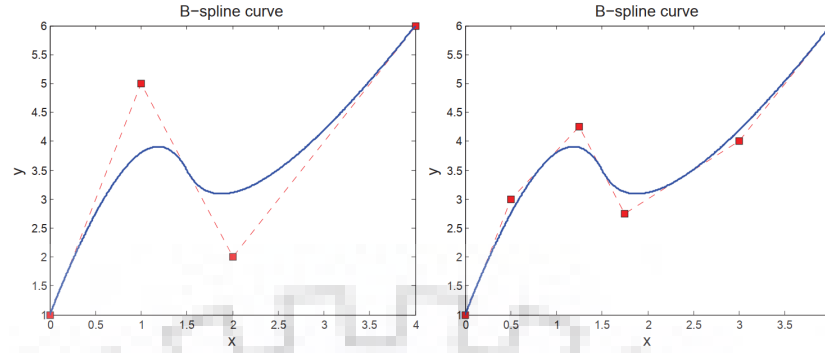


FIGURE 5.8: Knot refinement from  $\Xi = \{0, 0, 0, 0.5, 1, 1, 1\}$  to  $\tilde{\Xi} = \{0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1\}$ . Notice that the curve remains unchanged.

points. The new control points  $\tilde{\mathbf{B}} = \{\tilde{\mathbf{B}}_1, \tilde{\mathbf{B}}_2, \dots, \tilde{\mathbf{B}}_{n+m}\}$  are created by a linear combination of the previous control points,  $\mathbf{B}$ . The new control points  $\tilde{\mathbf{B}}$  will be given by

$$\tilde{\mathbf{B}} = \mathbf{T}^p \mathbf{B} \quad (5.10)$$

where

$$T_{i,j}^{q+1} = \frac{\tilde{\zeta}_{i+q} - \zeta_j}{\zeta_{j+q} - \zeta_j} T_{i,j}^q + \frac{\zeta_{j+q+1} - \tilde{\zeta}_{i+q}}{\zeta_{j+q+1} - \zeta_{j+1}} T_{i,j+1}^q$$

for  $q = 0, 1, 2, \dots, p-1$  and

$$T_{i,j}^0 = \begin{cases} 1 & \zeta_i \in [\zeta_j, \zeta_{j+1}) \\ 0 & \text{otherwise} \end{cases}$$

Knot insertion is also used to create repeated knots. The continuity of the basis will therefore be reduced while the curve remains the same. In figure 5.8 knot insertion is exploited, increasing the knot vector  $\Xi = \{0, 0, 0, 0.5, 1, 1, 1\}$  of order 2 to an extended knot vector  $\tilde{\Xi} = \{0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1\}$ . Observe that the B-spline curve is similar before and after additional knots were inserted.

## 5.2 NURBS

NURBS, non-uniform rational B-splines, are piece wise rational polynomials which is built from B-splines. The term *nonuniform* refers to using non uniform knot vectors.



The term *rational* refers to the fact that the NURBS basis functions are represented as rational. That is they are a combination of B-splines basis functions which is multiplied by a weight, and divided by the sum of the same B-spline basis functions multiplied by the same weights. If all the weights are equal to one, NURBS and B-splines are same.

### 5.2.1 Geometric perspective

NURBS are regular B-spline curves which are projected to a space of one dimension smaller. NURBS in  $\mathbf{R}^d$  are made by projecting a transformation of B-spline curves in  $\mathbf{R}^{d+1}$ . Conic sections like circles or ellipses are made by projecting quadratic B-spline curves from the  $(x, y, z)$ -plane to the  $(x, y, z = 1)$ -plane.

Assume that the B-spline curve is  $\mathbf{C}^w(\xi)$  with control points  $\mathbf{B}_i^w \in \mathbf{R}^{d+1}$ . The projected NURBS control points  $\mathbf{B}_i$  are therefore given from the control points of B-spline by

$$\begin{aligned} (\mathbf{B}_i)_j &= \frac{(\mathbf{B}_i^w)_j}{w_i} \quad j = 1, \dots, d \\ w_i &= (\mathbf{B}_i)_{d+1} \end{aligned} \quad (5.11)$$

Here  $(\mathbf{B}_i)_j$  is the  $j^{\text{th}}$  component of the vector  $\mathbf{B}_i$  and  $w_i$  is the  $i^{\text{th}}$  weight. In  $\mathbf{R}^3$ , the weights resemble the  $z$ -coordinates of the B-spline curve. Dividing the NURBS control points by the weight are thus the equivalent to applying a projective transformation on the control points. The same transformations are needed to be exploited on every point on the curve. With B-spline curves in  $\mathbf{R}^3$  that implies dividing all points by its height, see figure 5.9. To do the same transformation on all points we divide all points by

$$W(\xi) = \sum_{i=1}^n N_{i,p}(\xi) w_i \quad (5.12)$$

where  $N_{i,p}(\xi)$  are the B-spline basis functions. The NURBS curve,  $\mathbf{C}(\xi)$ , can now be defined as

$$(\mathbf{C}(\xi))_j = \frac{(\mathbf{C}^w(\xi))_j}{W(\xi)} \quad j = 1, \dots, d \quad (5.13)$$

where  $\mathbf{C}(\xi)^w = \sum_{i=1}^n N_{i,p}(\xi) \mathbf{B}_i^w = \sum_{i=1}^n N_{i,p}(\xi) \mathbf{B}_i w_i$ .

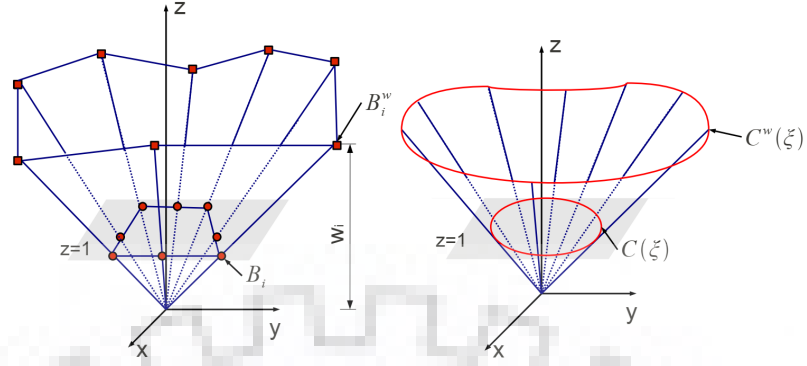


FIGURE 5.9: Projective transformation from a quadratic B-spline in  $\mathbf{R}^3$  to a circle in  $\mathbf{R}^2$ . The figure is reconstructed from Figure 2.28 in [6].

## 5.2.2 Basis functions

With the geometric point of view established NURBS basis functions can be defined as

$$R_i^p(\xi) = \frac{N_{i,p}(\xi)w_i}{W(\xi)} \quad (5.14)$$

where

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

as before. NURBS curve are then given by

$$C(\xi) = \sum_{i=1}^n R_i^p(\xi)B_i \quad (5.15)$$

This is the mapping from the parameter space to the physical space, as we will discuss in section 6.1.

In 2D the basis function is given as a tensor product by

$$R_{i,j}^{p,q}(\xi, \eta) = \frac{N_{i,p}(\xi)M_{j,q}(\eta)w_{i,j}}{W(\xi, \eta)} \quad (5.16)$$

with

$$W(\xi, \eta) = \sum_{i=1}^n \sum_{j=1}^m N_{i,p}(\xi)M_{j,q}(\eta)w_{i,j} \quad (5.17)$$

Surfaces are given by

$$\mathbf{S}(\xi, \eta) = \sum_{i=1}^n \sum_{j=1}^m R_{i,j}^{p,q}(\xi, \eta) \mathbf{B}_{i,j} \quad (5.18)$$

Extension to three dimensions are done analogously. Notice that if all weights are equal to one, NURBS and B-spline are the same.

### 5.2.3 Derivatives of basis functions

Derivatives of basis functions is vital to the process of numerical integration if NURBS is used in analysis. First derivative is obtained by differentiating equation (5.14) with respect to  $\xi$  using the quotient rule.

$$\frac{d}{d\xi} R_i^p(\xi) = w_i \frac{N_{i,p}(\xi)' W(\xi) - N_{i,p}(\xi) W(\xi)'}{(W(\xi))^2} \quad (5.19)$$

where

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

Generalizing to higher order terms results in

$$\frac{d^k}{d\xi^k} R_i^p(\xi) = \frac{w_i \frac{d^k}{d\xi^k} N_{i,p}(\xi) - \sum_{j=1}^k \frac{k!}{j!(k-j)!} W^{(j)}(\xi) \frac{d^{k-1}}{d\xi^{k-1}} R_i^p(\xi)}{W(\xi)} \quad (5.20)$$

In 2d, differentiating equation (5.16) with respect to  $\xi$  yields

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

and with respect to  $\eta$  yields

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

Here

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

and

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

.



## Chapter 6

# Isogeometric Analysis

Isoparametric concept involves using the same basis functions to represent geometry as well as design in the domain of analysis, whereas IGA also implies letting to do the same but in the analysis of domain as well as design. The main idea in IGA is to use NURBS as basis functions in both design and in the FEM. B-Spline and NURBS basis function and their derivatives are already discussed in chapter 5. The chapter starts with the concept of different spaces in the context of FEM as well as IGA. Finally, differences between IGA and FEM are discussed with basic methodology.

### 6.1 Spaces and Mappings

In classical FEA there is a concept of different spaces when doing numerical computation; that is, physical mesh, the physical elements, and the parent domain. During the analysis procedure, parameters are regularly transformed from one space to another for convenience and because of the way the various parameters are defined. The physical mesh is where the geometry is represented with help from nodes. The physical mesh is the space where the original geometry is discretized into non-overlapping physical elements. Each of these physical elements is linked to one parent element in the parent space where numerical integration is performed by using Gaussian quadrature. After performing the integration, inverse mapping can be used to go back to physical space. The physical elements are defined using nodal coordinates, and the dofs are the values of the basis function at the nodes. The local basis functions have support only on neighboring elements, due to compact support. The basis functions interpolate the nodes and are also called shape functions. Generally basis functions are defined in parametric space and Gauss quadrature can only be performed in parent space.

Different domains exist in IGA too; physical mesh, the control mesh, the parameter space, the index space and the parent element, all shown in figure 6.1.

### 6.1.1 The physical space

The physical space is where the actual geometry lies which is formed by linearly combining the basis functions defined in parametric space and the control points which lie in physical space. The basis functions usually do not interpolate the control points. The physical mesh is a decomposition of the geometry and can be discretized in two different ways; either by patches or by knot spans.

Sometimes it may not be possible to discretize a domain by only using elements. Patches are used to discretize complex domains or problems with discontinuous solutions. They can be thought of as subdomains which themselves are discretized using elements. Hence a patch can be a curve in 1D or a surface in 2D. To discretize a patch, each of them is divided by knot spans. Knot spans are defined by knots that define element domains. The basis functions are smooth at a particular knot, that is inside an element, but at the boundary, that is, across knots, are  $C^{p-m}$  continuous.  $m$  is times a particular knot is repeated. Knot spans are the lowest level and cannot be discretized further. In the parent domain if knots represent points in 1D, lines in 2D and planes in 3D, in the physical space they represent points in 1D, curves in 2D and surfaces in 3D.

### 6.1.2 The control mesh

The control mesh consists of details of control points. Think of control mesh as an overlay over the physical mesh which uses basis functions from parametric space to construct geometry. It may not coincide with the physical mesh even though it controls the geometry which lies in the physical mesh. An example of a control mesh in 2D is a bilinear quadrilateral defined by four control points [6]. The control variables, which are dof, are located at the control points. Control meshes can be severely distorted while physical geometry still remaining well defined.

### 6.1.3 The parameter space

The parameter space consists of definitions of the NURBS basis functions  $R_{i,j}^{p,q}$  and the knot vector. The knot vector is termed as uniform if all the knots are equally spaced. Each parameter space is local to patches. Therefore one has to loop over all the patches

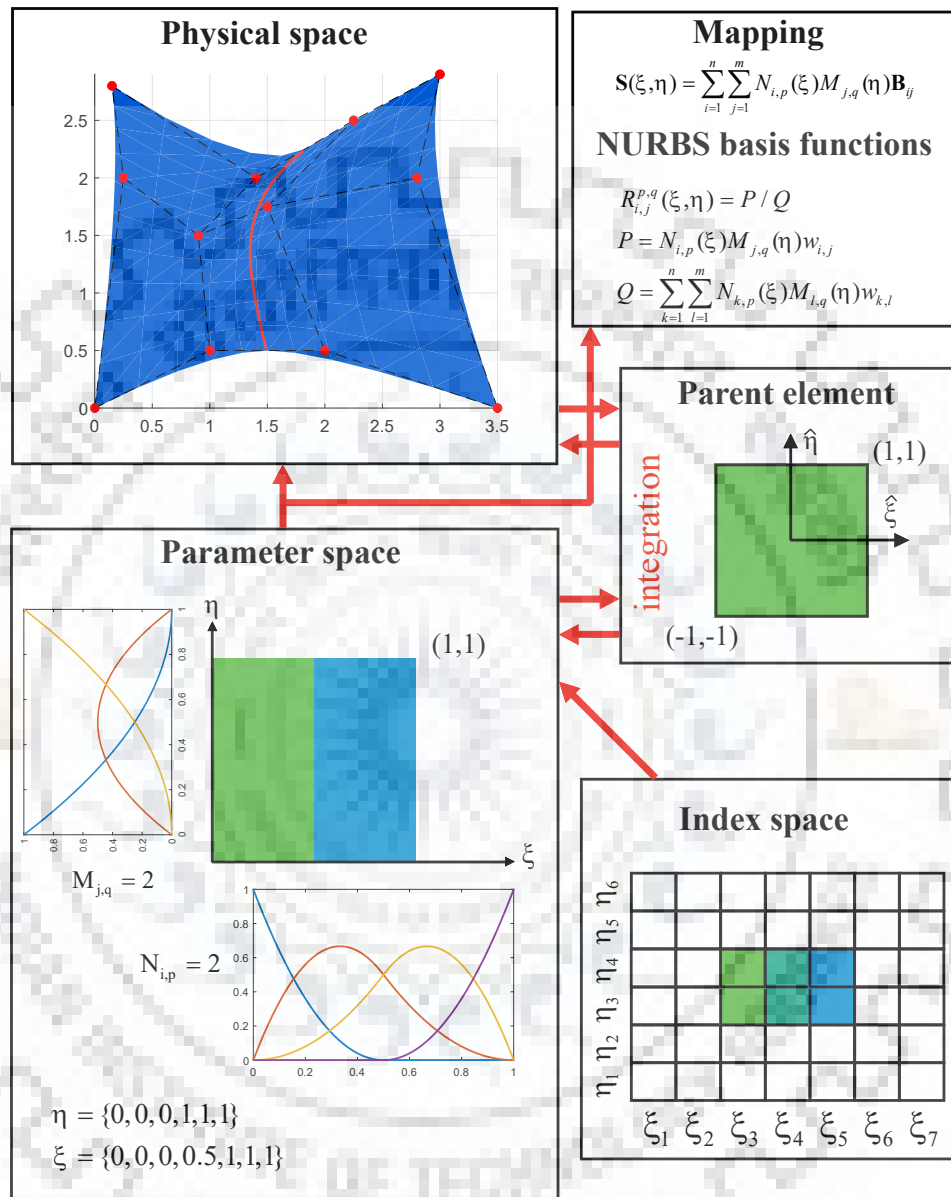


FIGURE 6.1: Different spaces in the context of IGA. A surface is shown in physical space with red dots representing control points. The red line in the physical space marks the element boundary in the physical space which is well defined in the parameter space. The two elements are shown in different colors in parametric space with the support of each element is shown in the respective color in the index space.

while looping over all the elements in the current patch. Each element in parametric space is mapped to an element in physical space and hence it can be said that each element in the physical space, therefore, replicates the corresponding element in the parameter space. The mapping from the parametric space to the physical space for surfaces is given by

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

a mapping that is global to whole patch.

### 6.1.4 The index space

Index space of a patch consists of knot values in a knot vector defined for that patch. Hence elements can be identified depending on knot values as well as continuity across elements. The index space is, for a 2D case, spanned by the area  $[1, n + p + 1]$  in the  $i$ -direction and  $[1, m + q + 1]$  in the  $j$ -direction.

### 6.1.5 The parent element

The parent element is global to all the patches and is of area  $[-1, 1] \times [-1, 1]$ . It is the parent element where the integration is performed.  $\xi$  and  $\eta$  in the parameter space are mapped to  $\hat{\xi}$  and  $\hat{\eta}$  in the parent element to do Gaussian quadrature for numerical integration. The mapping from the parent element to the parameter space is given by

$$\begin{aligned} \xi(\hat{\xi}) &= \frac{(\xi_{i+1} - \xi_i) \hat{\xi} + (\xi_{i+1} + \xi_i)}{2} \\ \eta(\hat{\eta}) &= \frac{(\eta_{i+1} - \eta_i) \hat{\eta} + (\eta_{i+1} + \eta_i)}{2} \end{aligned} \quad (6.1)$$

## 6.2 Isogeometric Analysis Vs Finite Element Method

In this section the differences and similarities between IGA and standard FEA are mentioned. The concept of IGA is that the shape functions that are used to model the exact geometry are used to approximate the solution field. In FEA, on the other hand, the basis functions chosen to approximate the solution field is used to approximate the known geometry. NURBS based Galerkin FEM is similar to classical FEA but with different basis functions being used.



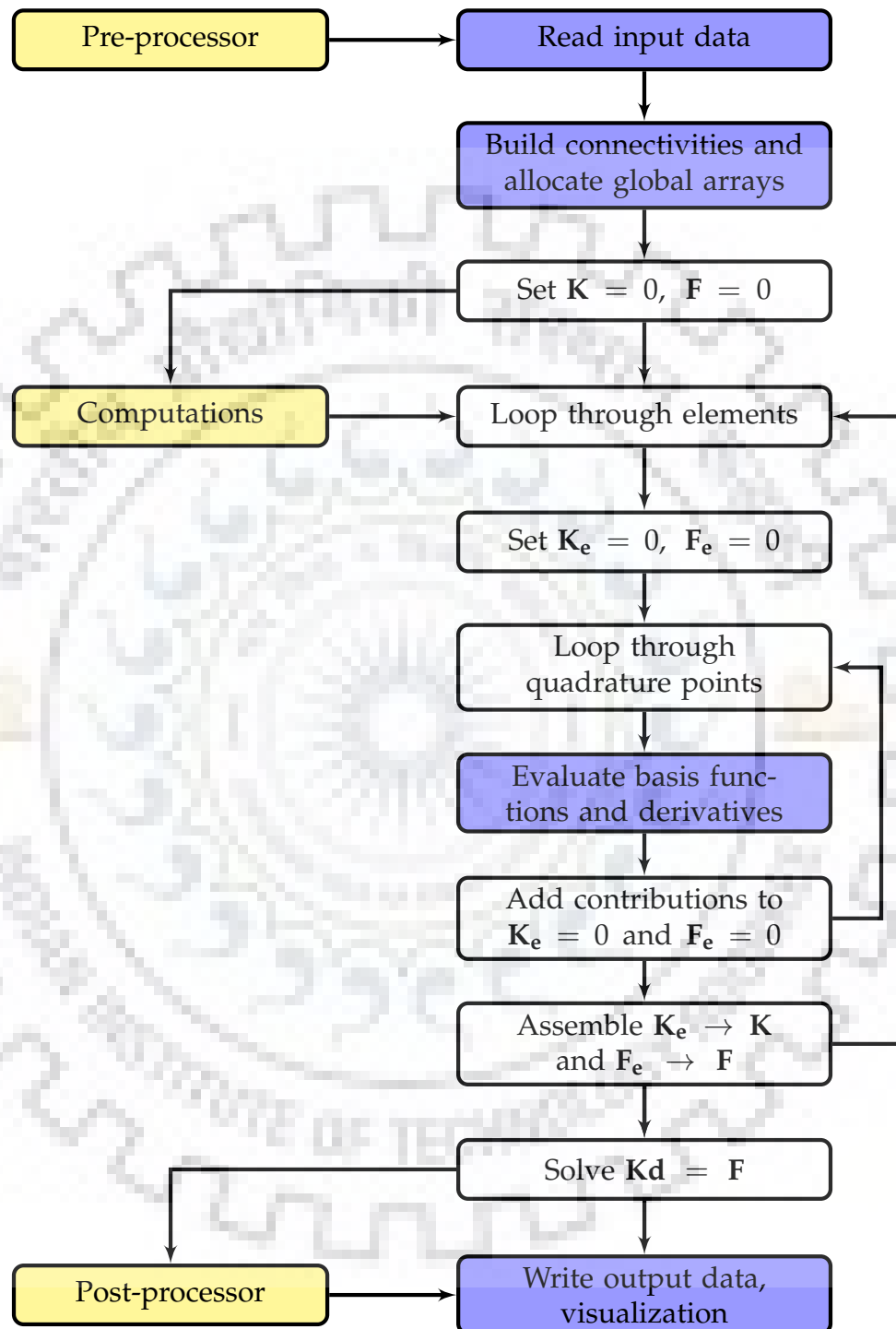


FIGURE 6.2: Flow chart for one patch IGA. The routines in blue differ from those in classical FEA.

TABLE 6.1: Comparison of IGA and FEM summarising differences and similarities between them. Taken from [12]

IGA	FEM
Control points	Nodal points
Control variables	Nodal variables
Knots	Mesh
Knot Span	Element
Exact geometry	Approximated geometry
NURBS basis functions	Lagrange basis functions
Basis not interpolating control points	Basis interpolating nodes
Patches	Subdomains
Compact support	
Partition of unity	
Isoparametric concept	
Affine covariance	
Patch tests satisfied	

The code architecture of IGA is shown in figure 6.2. The routines displayed in blue differ from the ones used in classical FEA. A different input is needed to convert a FEA code to an IGA one. FE mesh and nodal points are no longer serves as an input to describe the geometry, but rather knot vectors (which defines the basis functions) and control points (which in combination with basis functions construct geometry). Connectivity array, which links the local entities to global entities, changes. In IGA, the connectivity array is calculated using the input given to the code. Changing the input to code changes the nature and definition of basis functions which in turn changes the connectivities between elements and size of the element property matrices. Hence the global stiffness and mass matrix are no longer the same as that in FEA. In FEA the initial geometry is discretized using an approximate geometry whereas, in IGA, exact geometry is used at every level of discretization. This enables to have greater accuracy in solution as well as geometry description independent refinement procedure in IGA. In both FEA and IGA, the weak form's solution is a linear combination of the basis functions. In FEA, the coefficients used for linear combination with basis functions are the nodal variables, while in IGA they are the control variables. Both methods are based on isoparametric implementations of the Galerkin method. Both methods also utilize

an element approach with basis functions having compact support. In both approaches, the bases being used form a partition of unity. In FEA, the basis function can be both positive or negative, while NURBS basis functions cannot be negative. In FEA, the dofs are positioned at the nodes, while in IGA they are positioned at the control points. In FEA the continuity of the basis functions is fixed at the times of definition, while in IGA continuity can be easily changed as desired. Table 6.1 lists up some of the differences and similarities.



## Chapter 7

# Adaptive IGA

In the previous chapters the basics of NURBS based IGA has been discussed. Ideally, IGA should pair exact geometry representation capability with automatic adaptive local refinement to provide truly local h-refinement. NURBS based IGA is based on tensor product structure of basis functions and is thus incapable of doing local refinement. For this purpose T-splines was introduced in 2003 which has local refinement properties. Since then many spline technology has been introduced which has made local refinement possible in framework of IGA. Adaptive hierarchical refinement has recently gained considerable attention. PHT-Spline based adaptive hierarchical refinement is discussed in this chapter. The chapter starts with definition of T-meshes and then proceeds to procedure of evaluating basis functions over PHT-meshes. After that the procedure of cross insertion and removal is discussed. Chapter ends with the description of error estimator being used and the marking scheme to mark the elements to be refined.

### 7.1 Introduction

Finite element mesh is directly responsible for the accuracy of the solution being obtained using that mesh. Higher the number of elements in a mesh, lower is the error. This is a fundamental property of the FEM. Of course, more computational resources, both time and hardware, are required to solve finer meshes. However, resources can be used efficiently if mesh is made fine only in the region where error is high. Mesh refinement is the technique of making the mesh finer, i.e., increasing the solution accuracy across the mesh. The mesh is refined by comparing the solution domain with that of true value. This comparison can be done by either comparing the field quantity at

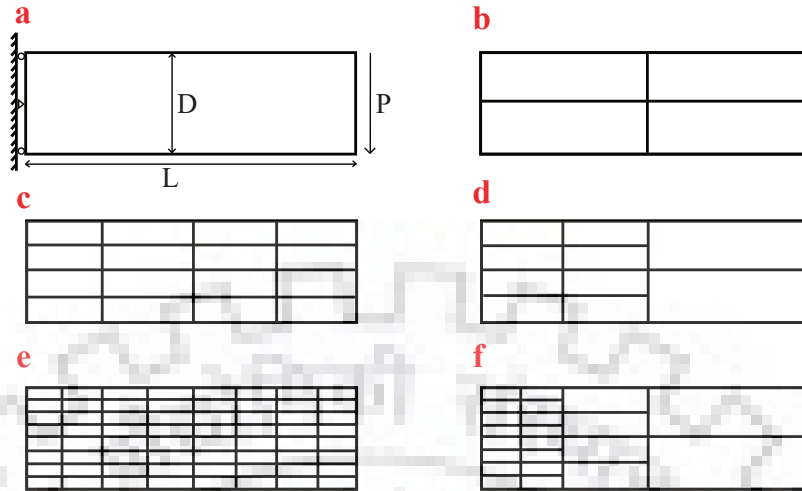


FIGURE 7.1: A cantilever beam and meshes: (a) a cantilever beam with parameters; (b) a uniformly coarse initial mesh used for both NURBS and PHT-splines based IGA; Mesh after one refinement: (c) NURBS and (d) PHT-splines; Mesh after two refinement: (e) NURBS and (f) PHT-splines

one or more points in the model or by evaluating the integral of field quantity over the problem domain.

Convergency studies requires choosing an approximate mesh refinement *metric*. This metric can be either global or local depending on the definition of metric. If the metric is defined as an integral over the entire model domain, integral of strain energy density, then the metric is called global. If the metric is defined at one location, like displacement or stress at a point, then it is called local. Different convergence rate is expected for different metrics. Refinement can be done by either increasing the approximating polynomial order,  $p$ -refinement, or by increasing the number of elements,  $h$ -refinement. IGA has one more type of refinement,  $k$ -refinement, which increases inter-element continuity as opposed to standard  $C^0$  continuity of conventional FEM. Increasing the order of element suffers from the drawback of requiring more computational resources than its counterparts. An example of comparison to global refinement to local refinement is shown in figure 7.1. The stark difference in the number of element being used with each refinement, while converging to the actual solution, is clearly visible in the figure.

Adaptivity in FEA requires three components. Firstly, an algorithm for refining or enriching the solution in a particular domain of solution is required. This is generally achieved by increasing the mesh size in particular part of model. However T- splines

[19] were originally developed for the purpose of local refinement but it suffers with the problem of inefficient T-junction insertion. Hence, in this study, PHT-splines are used as they are better known to handle adaptive local refinement. Secondly, to locate the area where refinement is needed, an error estimator is required which indicates the elements responsible for large contribution to error. In this study, a posteriori error estimator based on work by [23] is used for calculating the elements which needs to be refined. Third is a remeshing criterion which translates the output of error analysis and marking scheme to execute the local refinement procedure.

## 7.2 Mesh structure

### 7.2.1 T-meshes

A T-mesh is a mesh which allows T-junctions in a rectangular mesh [19]. The mesh can be said to be made up of grid lines. It is required that the ends of each grid lines lie on other grid lines. These grid lines forms a quadrilateral which is similar to an element. Figure 7.2 illustrates a typical T-mesh. The intersection of grid lines is called vertex and is classified on the basis of its location. If a vertex lies on the boundary of the domain then it is called a boundary vertex. Similarly if it happens to lie anywhere inside the domain, it is called interior vertex.

### 7.2.2 Hierarchical T-mesh

Instead of considering general T-meshes, hierarchical T-meshes are considered, since general T-meshes lack properties required for adaptivity.

A *hierarchical T-mesh* has a natural level structure. The structure are defined recursively. The lowest level in the mesh is level 0. From level  $k$  to level  $k + 1$ , subdividing an element(called cross-insertion) at level  $k$  results into four sub-elements, called children elements of the parent element. These children elements are at level  $k + 1$ . For simplicity, each element is subdivided equally simply by dividing it into four equal rectangular parts. Figure 7.3 illustrates the natural level structure at different refinement levels.

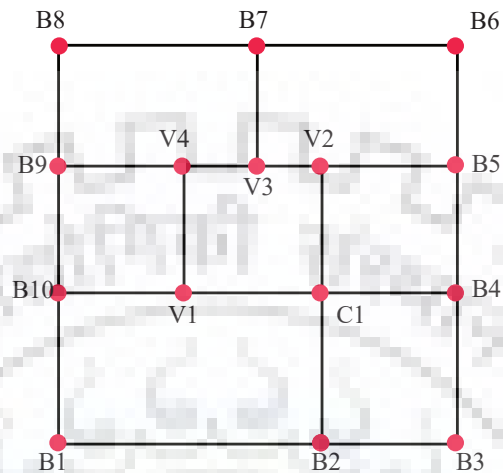


FIGURE 7.2: An example of T-mesh; B1 to B10 are boundary vertices; V1 to V4 are T-junctions; C1 is a crossing vertex

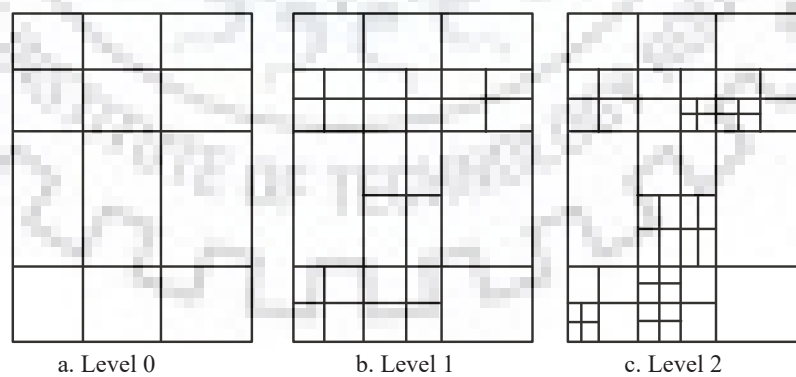


FIGURE 7.3: A hierarchical T-mesh

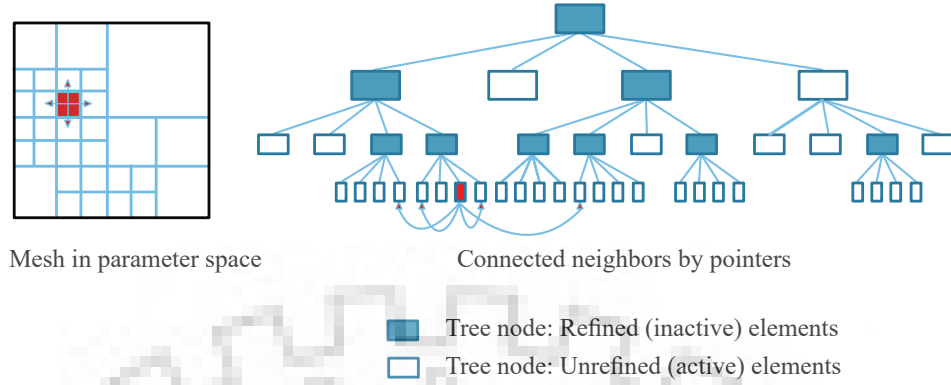


FIGURE 7.4: Quadtree structure illustrating the hierarchical data organization of an adaptive mesh. Pointers represents the neighbouring relations of element (in red colour) at a particular hierarchical level.

### 7.2.3 Tree structure

Efficient and straightforward implementation of hierarchical refinement is possible by using quadtree and octree structure [17]. In this study quadtree structure has been used to construct PHT-spline elements. Tree structure is better suited for tracking connectivities between the elements at different levels of refinement. Figure 7.4 illustrates the analogy between a tree structure and an adaptive hierarchical mesh. Each leaf of the tree contains information of the placement of elements in the parametric space, their associated local basis functions, and element to node connectivities [18]. Each leaf also stores information regarding the neighboring elements so that they can be placed with little computational effort.

## 7.3 Basis function of PHT-splines

### 7.3.1 Definition of knot vectors

Knot vector with multiplicity of two can be defined as

$$\Xi = \{\xi_0, \xi_0, \xi_1, \xi_1, \xi_2, \xi_2, \xi_3, \xi_3, \dots, \xi_{m-2}, \xi_{m-2}, \xi_{m-1}, \xi_{m-1}, \xi_m, \xi_m, \xi_m\} \quad (7.1)$$



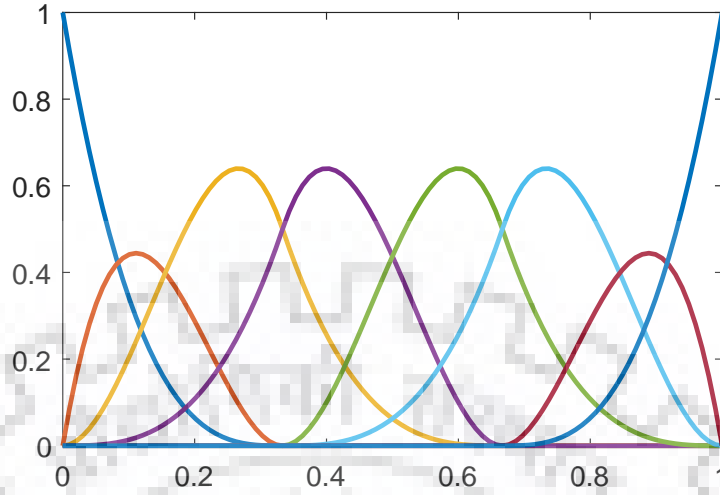


FIGURE 7.5: Cubic B-spline basis functions with multiplicity of two of interior knots, that is, only  $C^1$  continuity:  $\Xi = \{0, 0, 0, 0, 1/3, 1/3, 2/3, 2/3, 1, 1, 1, 1\}$

such that  $\xi_i < \xi_{i+1}$ ,  $1 \leq i \leq m-2$ , and  $\xi_{m-1} = \xi_m$ . Equation (7.1) can be rearranged as:

$$\Xi = \left\{ \underbrace{\xi_0, \xi_0, \xi_1, \xi_1}_{p+1}, \underbrace{\xi_2, \xi_2}_{k=2}, \dots, \xi_{m-2}, \xi_{m-2}, \underbrace{\xi_{m-1}, \xi_{m-1}, \xi_{m-1}, \xi_m, \xi_m}_{p+1} \right\} \quad (7.2)$$

As already mentioned in [8], the construction of PHT-spline basis functions is initiated from  $C^1$  continuous cubic B-splines. Equation (7.2) implies that cubic B-splines ( $p = 3$ ) are  $C^{p-k} = C^1$ -continuous at every interior knots (figure 7.5). It is seen that, for the line case, only two B-spline basis functions in  $[\xi_{i-1}, \xi_{i+1}]$  are nonzero at each interior knot  $\xi_i$ . These two basis functions are incorporated with knot vectors  $[\xi_{i-1}, \xi_{i-1}, \xi_i, \xi_i, \xi_{i+1}]$  and  $[\xi_{i-1}, \xi_i, \xi_i, \xi_{i+1}, \xi_{i+1}]$ , respectively. Also, two remaining B-spline basis functions vanish at  $\xi_i$ , see figure 7.5. This property also is satisfied by any B-spline basis function of degree  $p \geq 3$ . In addition, every interior knot is of multiplicity two, the derivatives of the basis functions also vanish at  $\xi_i$ . Extending further this fact to the surface case, there are four B-spline basis functions in  $[\xi_{i-1}, \xi_{i+1}] \times [\eta_{i-1}, \eta_{i+1}]$  that are nonzero at each interior vertex  $(\xi_i, \eta_i)$ .

### 7.3.2 The Bernstein basis

The Bernstein polynomials for the basis for the Bézier elements. The Bernstein polynomials are usually defined over the unit interval  $[0, 1]$ , but in FEA the bi-unit interval,  $[-1, 1]$ , is preferred to take advantage of the Gauss quadrature. The Bernstein basis are defined over the interval  $[-1, 1]$  as

$$B_{i,p}^k(\tilde{\xi}^k) = \frac{1}{2^p} \binom{p}{i-1} (1 - \tilde{\xi}^k)^{p-(i-1)} (1 + \tilde{\xi}^k)^{i-1} \quad (7.3)$$

where the binomial coefficient

$$\binom{p}{i-1} = \frac{p!}{(i-1)!(p+1-i)!}, \quad 1 \leq i \leq p+1$$

### 7.3.3 Bézier extraction for T-splines

The PHT-spline element construction can be simplified by the use of Bézier extraction [4], [18]. For each PHT-spline element  $e$ , Bézier extraction determines the exact representation of PHT-spline basis in terms of a set of Bernstein polynomials,  $\mathbf{B}(\tilde{\xi})$ . Every localised PHT-spline basis function,  $N^e(\tilde{\xi})$ , can be written as a linear combination of these Bernstein polynomials. In other words,

$$N^e(\tilde{\xi}) = \sum_{b=1}^{(p+1)^d} c_b^e B_b(\tilde{\xi}) \quad (7.4)$$

In matrix-vector form equation (7.4) is written as

$$\mathbf{N}^e(\tilde{\xi}) = \mathbf{C}^e \mathbf{B}(\tilde{\xi}) \quad (7.5)$$

where  $\mathbf{C}^e$  is the element extraction operator. The element defined by Bernstein polynomials is called Bézier element.

Note that, in contrast with the T-spline basis functions  $\mathbf{N}^e$ , same number of Bernstein basis functions exists for all the elements. That is, each Bézier element is defined in terms of the exact same set of Bernstein basis functions. Hence only the extraction operator differs for each element which needs to be calculated to construct the PHT-spline basis functions. More

### 7.3.4 Modification of the basis functions at level $k$

Basis functions needs to be modified if an element at level  $k$ ,  $T_{k,e}$ , is refined to 4 children elements at level  $k + 1$ ,  $T_{k+1,e_1}$ ,  $T_{k+1,e_2}$ ,  $T_{k+1,e_3}$ ,  $T_{k+1,e_4}$ , (This procedure of refinement will be referred as "cross insertion"). Basis functions are formed using Bézier representation and hence Bézier coefficients  $C_{ij}$  needs to be calculated again for the new mesh. The procedure for calculating the coefficients are described below.

#### Blossoming with De Casteljau's algorithm

The first step is to calculate the Bézier coefficients of the original basis functions on the elements  $T_{k+1,e_1}$ ,  $T_{k+1,e_2}$ ,  $T_{k+1,e_3}$ ,  $T_{k+1,e_4}$ . This subdivision is realized using De Casteljau's algorithm [10]. In one dimension, the algorithm (see ) takes input as  $p + 1$  Bézier coefficients  $c_1^{(0)}, c_2^{(0)}, \dots, c_{p+1}^{(0)}$  and computes the following coefficients,  $c_1^{(0)}, c_1^{(1)}, \dots, c_1^{(p)}$  and  $c_1^{(p)}, c_2^{(p-1)}, \dots, c_{p+1}^{(0)}$ , according to recurrence relation:

$$c_i^{(j)} := \frac{(c_i^{(j-1)} + c_{i+1}^{(j-1)})}{2}, \quad i = 1, \dots, p + 1 - j, \quad j = 1, \dots, p$$

These two sets of coefficients,  $c_1^{(0)}, c_1^{(1)}, \dots, c_1^{(p)}$  and  $c_1^{(p)}, c_2^{(p-1)}, \dots, c_{p+1}^{(0)}$ , serves the purpose of splitting the original Bézier polynomial defined on the segment  $[u_k, u_{k+1}]$  into two segments  $[\xi_k, (\xi_k + \xi_{k+1})/2]$  and  $[\xi_k, \xi_{k+1}]$ . In two dimensions  $(p + 1)^2$  coefficients  $C_{ij}^{(E)}$  corresponding to a single basis function  $N$  are arranged in a  $(p + 1) \times (p + 1)$  array and compute the  $(2p + 2) \times (2p + 2)$  array that holds the Bézier coefficients corresponding to the 4 child elements. This is accomplished by applying Algorithm 1  $3p + 3$  times: first  $p + 1$  times to each row of the  $C_{ij}^{(E)}$  array, and then  $2p + 2$  times to each column of the resulting  $(p + 1) \times (2p + 2)$  array. After this procedure, a  $(2p + 2) \times (2p + 2)$  array is obtained which is split into 4 subcells of size  $(p + 1) \times (p + 1)$ , each containing the Bézier representation of the basis function  $N$  on the 4 child elements. Do note that only Bézier ordinates are changed after its subdivision into four sub-elements while the function  $N_i^k(\xi, \eta)$  has not changed, but is now defined over the refined mesh.

#### Truncation by zeroing out Bézier coefficients

A truncation procedure is employed for basis functions on the elements being refined to ensure linear independence of basis and better sparsity in the resulting linear system. It

**Algorithm 1** De Casteljau Algorithm in 1D**Input:** Bézier coefficients  $c_1^{(0)}, c_2^{(0)}, \dots, c_{p+1}^{(0)}$ **Output:** Bézier coefficients  $c_1^{(0)}, c_1^{(1)}, \dots, c_1^{(p)}$  and  $c_1^{(p)}, c_2^{(p-1)}, \dots, c_{p+1}^{(0)}$ 

- 1: **for**  $j = 1, \dots, p$  **do**
- 2:   **for**  $i = 1, \dots, p + 1 - j$  **do**
- 3:     Compute  $b_i^j = (b_i^{(j-1)} + b_{i+1}^{(j-1)}) / 2$
- 4:   **end for**
- 5: **end for**

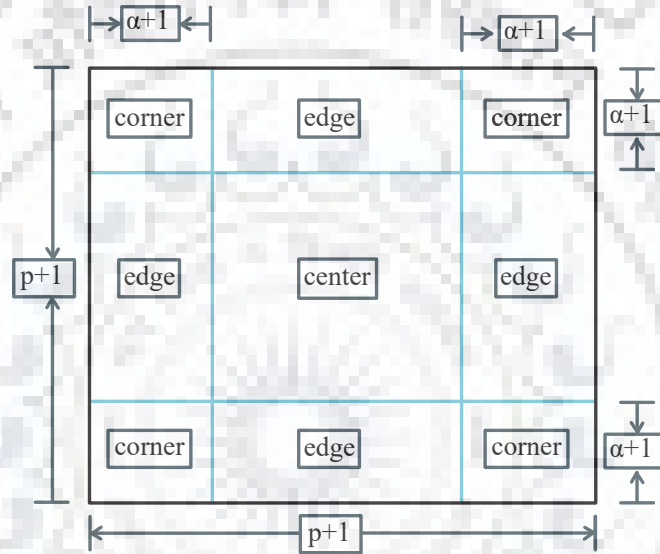


FIGURE 7.6: The 9 regions into which each child element is subdivided. The axis labels show the number of Bézier coefficients in each direction.

is assumed that the basis functions of polynomial degree  $p$  have  $C^{\alpha, \alpha}$  continuity, where  $2\alpha + 1 \leq p$ .

For the truncation procedure, each of the 4 child elements are split into 9 regions, as shown in figure 7.6. Then, the Bézier coefficients in the corner regions, as well as the two edge and center regions closest to the new basis vertex in each element are zeroed out. The aim of this step is to reduce the support of basis functions, which results in less overlap and sparser linear system. Since all Bézier ordinates are set to zeros, any basis function which could result in a linear dependency with refined basis are deactivated.

### 7.3.5 Insertion of new basis functions

Cross insertion involves calculation of new basis functions for the element which is to be refined into four new elements. The refinement procedure is completed when these new basis functions are inserted in the tree data structure defined which contains information about the hierarchical mesh. For removal of T-junction, the new basis functions are inserted in the neighbor elements in a way that they have support on both the refined element as well as its neighbor.

## 7.4 Recovery based error estimation

Error estimator manages the evolution of a FEA mesh in an adaptive refinement approach. In most cases, error is calculated with respect to exact solution. In recovery based error estimation, error is calculated with respect to a recovered solution instead of exact solution, which has shown convergence towards true error under certain assumptions [23] [24]. This type of error estimator based on superconvergent patch recovery was first proposed by [23] and is called Zienkiewicz-Zhu error estimator or  $Z^2$  error estimator. It was shown that if the recovered derivatives (in this case stresses which are derivatives of displacement) are superconvergent, the  $Z^2$  error estimator will always be asymptotically exact in the energy norm. Hence the accuracy of the procedure is dependent on the accuracy of the recovered solution.

To calculate derivatives from the finite element approximation, obvious technique is to directly differentiate the resulting solution at points of interests. For example, stresses are obtained by differentiating displacement solution and using appropriate constitutive relations. This direct calculation however suffers from inferior accuracy because the derivatives obtained are discontinuous at boundaries of elements and at inter-element nodes where accurate values of stresses may be desired. To overcome this, the original procedure, the  $Z^2$  error estimator, computed more accurate solution at carefully chosen points (called superconvergent points) that are capable of generating better approximation of desired quantity. At those points a single and continuous polynomial expansion of the function describing the derivatives on an element patch surrounding the nodes at which recovery is desired is made to fit locally in a least square manner. Table 7.1 shows the super-convergent points on an interval  $[-1, 1]$  for several values of spline degree  $p$  and continuity order  $\alpha$ . As the process is local the computation cost is negligible.

The solution is recovered at each refinement level since the error needs to be calculated at each refinement level. The mesh at refinement level  $k$ ,  $T_k$ , is divided into a set of non-overlapping domains, where each subdomain consists of four elements because of the way the elements are refined. Let  $\Omega_m, m = 1, \dots, N_{domain}$  be a set of  $n$  non-overlapping patches such that together they form  $T_k$ . These domains are chosen such that the elements inside share same parent element. Recovered stress is superapproximate if certain properties such as consistency, locality, boundedness and linearity are satisfied. In each of these subdomains the stresses are calculated again,  $\sigma_*^m$  where star\* denotes enhanced solution, at superconvergent points using locally defined PHT-splines. The exact error in energy norm for entire patch domain is given by

$$\|e^*\|_{E(\Omega)} = \sqrt{\sum_{m=1}^{N_{domain}} \int_{\Omega^m} (\sigma_*^m - \sigma_h^m)^T (\mathbf{D}^m)^{-1} (\sigma_*^m - \sigma_h^m) d\Omega} \quad (7.6)$$

TABLE 7.1: Super-convergent points for splines of degree  $p$  and continuity  $C^\alpha$  on interval  $[-1, 1]$ .

p	$\alpha$	Super-convergent points
3	1	$\pm 1, 0$
4	1	$\pm \sqrt{(3/7) \pm (2/7)\sqrt{6/5}}$
5	2	$\pm 1, \pm \sqrt{1/3}, 0$
6	2	$\pm 0.790208564, \pm 0.2800702925$

To obtain error across multiple patches after obtaining error at each patch individually, the following equation can be used

$$\|e_*\|_{E(T_k)} = \left( \sum_{n=1}^{N_{patch}} \left( \|e_*\|_{E(\Omega^n)} \right)^2 \right)^{\frac{1}{2}} \quad (7.7)$$

## 7.5 Marking algorithm

After calculating error for each subdomain for the whole mesh, the errors are arranged in the descending order of their magnitude for each subdomain. Remember each subdomain consist of four elements. The simplest refinement strategy is to compare these errors with a particular threshold and mark those elements whose error exceeds the threshold. This is called as "absolute threshold" marking strategy. However this method

does not take into account overall distribution of error and hence in this study, "Dörfler marking" strategy [9] is used. In this strategy, the elements with largest contribution to the total estimated error (in terms of percentage  $\rho$ ) are selected. The parameter  $\rho = 1$  results in uniform refinement,  $\rho \ll 1$  results in smaller refinement steps. The whole refinement procedure is illustrated in figure 7.7. The initial mesh is refined uniformly for the purpose of error estimation. The calculated error is used to mark the elements (shown in blue) to be refined. Then the marked elements are refined where new basis functions, connectivity between elements, element properties are calculated again for the whole mesh. The same procedure keeps repeating until the error drops below the threshold value specified.



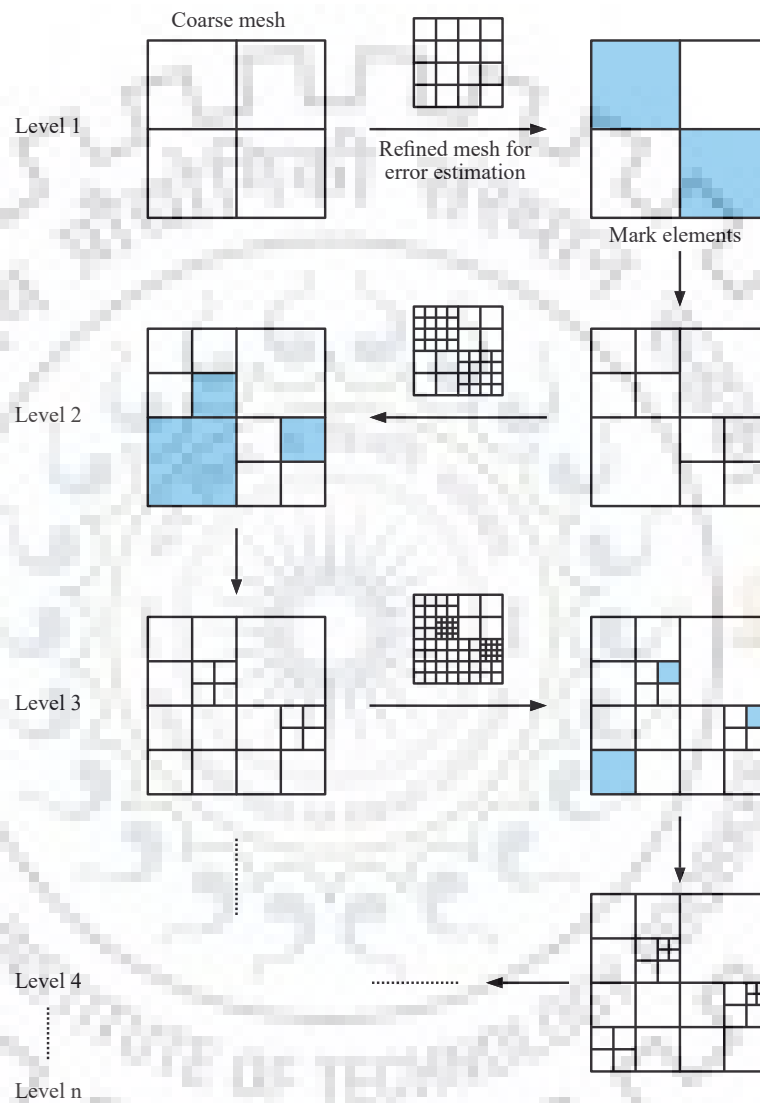


FIGURE 7.7: Illustration of error estimation, marking and refinement procedure for adaptive PHT element.



## Chapter 8

# Isogeometric Elements

After studying the theoretical background of IGA and linear elasticity, formulation of some IGA elements are given in this chapter. Same description of elements are used in the IGA framework, developed in Python, to compute the results mentioned in chapter 9. Three basic types of elements are discussed which comprises almost whole structural analysis domain. The chapter starts with description of bar element, beam element (based on Euler theory), and ends with plate element (based on Reissner-Mindlin theory). For the sake of simplicity, process of IGA has been described for bar element only.

### 8.1 Bar Element

Bars are members are that are subjected to axial force only. These members have length considerably large compared to their cross-sectional dimension. Typical member considered for explaining the procedure is shown in figure 8.1.

$$R_i^p(\xi) = \frac{N_{i,p}(\xi)w_i}{W(\xi)} \quad (8.1)$$

**Step 1:** Selecting suitable field variable

In stress analysis or free vibration analysis, displacements are selected as field variable. In a bar there is only one component of displacement to be considered which is displacement along its axis, i.e, x direction. NURBS curve is used to represent the

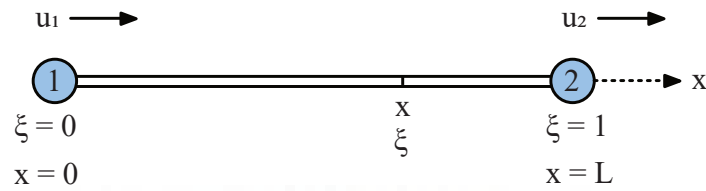


FIGURE 8.1: Straight bar element

coordinates of the geometry of the bar (section 5.2) by:

$$\mathbf{C}(\xi) = \sum_{i=1}^n R_i^p(\xi) \mathbf{B}_i \quad (8.2)$$

where  $R_i^p(\xi)$  is given by equation (5.14)

### Step 2: Discretizing the element

In IGA the element is discretized by knot insertion (p-refinement). Increasing the number of knot spans increases the number of elements. Hence the initial knot vector discretizes the element as well as forms shape functions defined for the element. The procedure of forming the shaping functions as well as their support at control points is mentioned in chapter 5. For example, if the initial knot vector is  $\Xi = \{0, 0, 0, 1/3, 2/3, 1, 1, 1\}$  then 5 control points are needed to make a NURBS curve of degree 2 as shown in figure 8.2. The support of control points for each element can be seen in the figure.

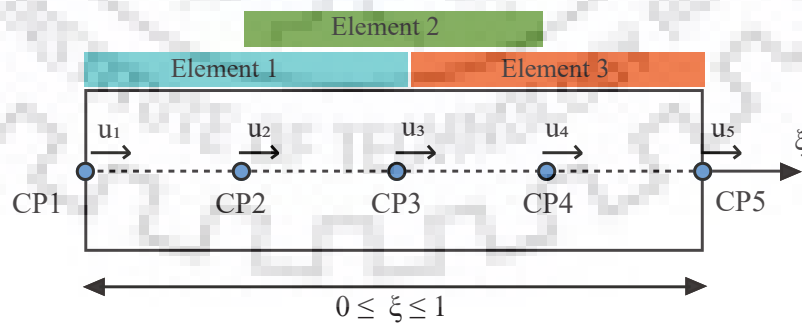


FIGURE 8.2: Bar element in parametric space

Since the number of knot spans in  $\Xi$  is 3, the number of elements is equal to 3. Nodal displacement vector for an element is given as:

$$\{\delta\} = \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix} \quad (8.3)$$

There is one degree of freedom at each control point. Hence total dof in the problem is  $5 \times 1 = 5$ . Hence nodal displacement vector for the problem is  $\{\delta^T\} = \{u_1 \ u_2 \ u_3 \ u_4 \ u_5\}$

**Step 3:** Selecting interpolation functions The basis functions needed to interpolate the geometry as well as solution space is shown in figure 8.3

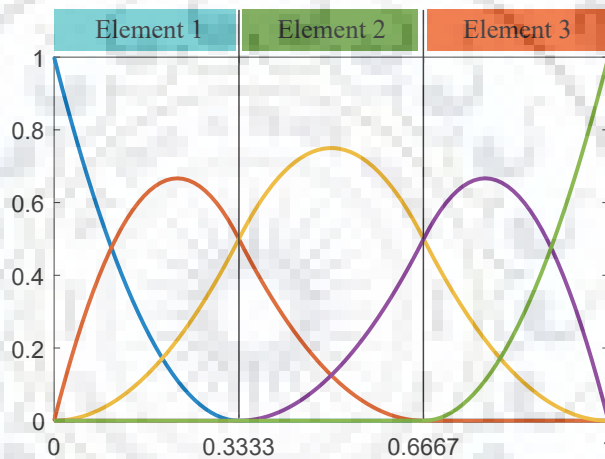


FIGURE 8.3: Support of basis functions for knot vector  $\Xi = \{0, 0, 0, 1/3, 2/3, 1, 1, 1\}$  of degree 2

Hence generalised displacement for the element is

$$\{u\} = [N]\{\delta\}_e \quad (8.4)$$

where  $[N]^T = [N_1 \ N_2 \ N_3 \ N_4 \ N_5]$  which are shown in figure 8.3.

**Step 4:** Element properties

In this step element stiffness matrix and mass matrix is assembled for the element. Since in this study bar element is used for finding out free vibration frequencies of the

element the governing PDE is given by

$$\rho A(x) \frac{\partial^2 u}{\partial t^2} - \frac{\partial}{\partial x} \left( EA(x) \frac{\partial u}{\partial x} \right) = 0 \quad (8.5)$$

where  $A(x)$  is the cross section area,  $E$  is the Young's modulus,  $\rho$  is the specific mass and  $t$  is the time. The problem of free vibration consists of finding the axial displacement  $u = u(x, t)$  which must satisfy boundary and initial conditions defined in the problem.

Equation (8.5) is converted to weak form using Galerkin method (section 4.2). The resulting matrix form of equation in terms of stiffness and mass is given by (section 3.7)

$$\det \left( K - \omega_n^2 M \right) = 0 \quad (8.6)$$

where  $\omega_n$  is the natural frequency, and  $K$ , global stiffness matrix, is formed by assembling  $K_e$ , element stiffness matrix, for each element which itself is given as

$$K_e = \int_0^L EA \frac{\partial N_1}{\partial x} \frac{\partial N_2}{\partial x} dx \quad (8.7)$$

and  $M$  is the global mass matrix formed by assembling individual element mass matrices given as

$$M_e = \int_0^L \rho A N_1 N_2 dx \quad (8.8)$$

Since the description of basis functions is in parametric space while integration is being carried out in physical space, Jacobian (section 4.4) has to be used. To find out the derivatives of shape function in equations (4.6) and (8.7) is used. To transform the infinitesimal length element in physical space to parametric space equation (4.7) is used. By use of Jacobian the integration which was supposed to be carried out in physical space is now being carried out in parametric space. Further, integration is carried out by the use of Gauss quadrature (section 4.5). Since the order of NURBS curve chosen is 2, the number of Gauss points used are 3. Gauss points weight and locations are given in table 4.1.

#### Step 5: Global properties

After computing stiffness matrix of an element, global stiffness matrix is assembled

in which each element stiffness matrix has its contributions corresponding to the dof. In the assumed case, for element 2 (figure 8.2) with dof 2, 3 and 4 the entries of the element stiffness matrix will be reflected at row and column location 2nd, 3rd and 4th of the global stiffness matrix. In the same way mass matrix is computed.

#### Step 6: Boundary conditions

Boundary condition is applied at control points (in this case at the ends of bars) by deleting the respective row and column from the global stiffness and global mass matrix.

#### Step 7: Solution of simultaneous equations

After imposing the boundary conditions, equation (8.6) is solved. If the problem to be solved is of static nature, like finding deflections under loading, then equation to be solved is as follows:

$$KU = F \quad (8.9)$$

$$U = K^{-1}F \quad (8.10)$$

where  $K$  is the global stiffness matrix,  $U$  is the displacement vector to be found and  $F$  is the force vector consisting of forces applied to the body. Then the computed result is compared with theoretical results to judge the accuracy and efficiency of the proposed method.

## 8.2 Beam Element

Bending problems of thin beams and shells does not require independent rotation fields. Note that by being 'thin', transverse shear deformations are being neglected. Since slope dofs are necessary in order to satisfy continuity requirements, classical FEA approach does not consider the rotation free formulation.

In IGA however the model is represented using control points and each control point has one dof for 1D cases. Hence slope as a dof cannot be captured separately in IGA formulation of beam. Therefore in IGA, beam analysis is carried out with a rotation-free

element [7]. Hence in this study a rotation-free beam element based on Euler-Bernoulli beam (section 3.5.1) is formulated. Typical member considered is shown in figure 8.4.

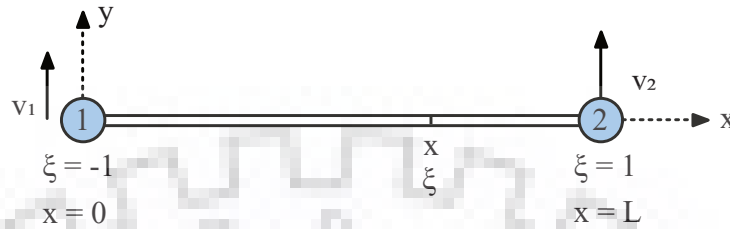


FIGURE 8.4: Straight beam element.

### Element displacement vector

There is one dof at each node which is the transverse deflection of beam. Hence the displacement vector for element is

$$\{\delta\} = \begin{Bmatrix} v_1 \\ v_2 \end{Bmatrix} \quad (8.11)$$

### Element properties

Beam element is used for finding out free vibration frequencies of the element the governing PDE is given by

$$EI(x) \frac{\partial^4 v}{\partial x^4} + \rho A(x) \frac{\partial^2 v}{\partial t^2} = 0 \quad (8.12)$$

where  $A(x)$  is the cross section area,  $E$  is the Young's modulus,  $\rho$  is the specific mass and  $t$  is the time. The problem of free vibration consists of finding the transverse displacement  $v = v(x, t)$  which must satisfy boundary and initial conditions defined in the problem. The terms of stiffness and mass, equation is given by

$$\det(\mathbf{K} - \omega_n^2 \mathbf{M}) = 0 \quad (8.13)$$

where  $\omega_n$  is the natural frequency, and  $K$ , global stiffness matrix, is formed by assembling  $K_e$ , element stiffness matrix, for each element which itself is given as

$$K_e = \int_0^L EI \frac{\partial^2 N_1}{\partial x^2} \frac{\partial^2 N_2}{\partial x^2} dx \quad (8.14)$$

and  $M$  is the global mass matrix formed by assembling individual element mass matrices given as

$$M_e = \int_0^L \rho A N_1 N_2 dx \quad (8.15)$$

Since in this case  $2^{nd}$  order derivatives of basis function is required, use of  $2^{nd}$  order jacobian is essential. Hence the second order derivative of basis functions in physical space is given by equation (4.9) which is shown below:

$$\left\{ \frac{\partial^2 f}{\partial x^2} \right\} = \left[ \left( \frac{\partial x}{\partial \xi} \right)^2 \right]^{-1} \left( \left[ \frac{\partial^2 f}{\partial \xi^2} \right] - \left[ \frac{\partial^2 x}{\partial \xi^2} \right] \left\{ \frac{\partial f}{\partial x} \right\} \right) \quad (8.16)$$

### 8.3 Plate Element

Plates and shells are a particular form of 3D solids having thickness very small compared to its other dimensions. A plate element only carries transversal loads which produces deflection and rotation of the normals of the middle plane which can be deduced as bending deformation. Theory of plates (section 3.6) is used to derive the strain matrix for the formulation of plate element. Plate element formulated (see figure 8.5) is a 2D 4 noded bending element with 3 dof at each node.

#### Element displacement vector

There are 3 dof at each node which in matrix form can be represented as:

$$\{\delta\} = \begin{Bmatrix} w \\ \theta_x \\ \theta_y \end{Bmatrix} \quad (8.17)$$

#### Element properties

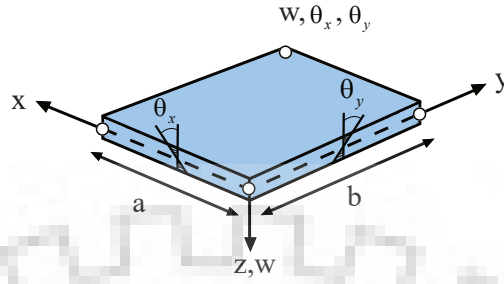


FIGURE 8.5: 4-Noded plate bending element with 3 dof at each node

In this study plate element is formulated for the purpose of finding static displacements under various conditions of load and support conditions. The governing PDE is given as:

$$\frac{\partial^4 w}{\partial x^4} + 2 \frac{\partial^4 w}{\partial x^2 \partial y^2} + \frac{\partial^4 w}{\partial y^4} = \frac{q}{D} \quad (8.18)$$

where  $w$  is the displacement in  $z$  direction,  $q$  is the transverse loading,  $D$  is the flexural rigidity given by  $D = Et^3 / [12(1 - \nu^2)]$  in which  $h$  is the thickness of the plate and  $E$  is Young's modulus.

Although in this study the governing equations are derived using minimization of potential energy and constitutive matrix is found by writing moment curvature relationship (equation (3.17)). Thus for finding the deflection of plate under a certain loading condition following equation is solved

$$KU = F \quad (8.19)$$

where  $K$  is the global stiffness matrix,  $U$  is the displacement vector to be found and  $F$  is the force vector consisting of forces applied to the body.

$K$  is given as

$$[K] = \iint_A [B]^T [D] [B] dx dy$$

The above expression in local coordinates is written as

$$[K] = \iint_A [B]^T [D] [B] |J| d\xi d\eta \quad (8.20)$$



where  $D$  is given by

$$[D] = \begin{bmatrix} \frac{Et^3}{12(1-\nu^2)} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} & \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & \frac{Et}{2(1+\nu)} \begin{bmatrix} \alpha & 0 \\ 0 & \alpha \end{bmatrix} \end{bmatrix} \quad (8.21)$$

$B$  matrix is given by curvature-displacement relationship (equation (3.18)).

$$[B_i] = \begin{bmatrix} 0 & 0 & \frac{\partial N_i}{\partial x} \\ 0 & -\frac{\partial N_i}{\partial y} & 0 \\ 0 & -\frac{\partial N_i}{\partial x} & \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial x} & 0 & N_i \\ \frac{\partial N_i}{\partial y} & -N_i & 0 \end{bmatrix} \quad (8.22)$$

## Chapter 9

# Numerical Results

As stated in the previous chapters the initial concept of IGA has been extended and generalized in many ways. In this section the deflection of plate element subjected to static loading using the framework of IGA is discussed. After that spectrum analysis is done for bar and beam element. The physical domain is described by NURBS and also discretized by NURBS. Multipatch cantilever beam is solved in framework of PHT-splines to demonstrate adaptive refinement advantages of IGA.

### 9.1 Bar element free vibration

Spectrum Analysis is the comparison between computed natural frequencies,  $\omega_n^h$ , with the analytically computed natural frequencies,  $\omega_n$ . To start with spectrum analysis simplest vibrational model is considered, longitudinal vibrations of an elastic rod. To model an elastic bar is not a problem of geometrical accuracy. Both, FEA and NURBS based IGA, are equally capable of representing the problem domain exactly. Hence, the difference in results will be from the approximating properties of the basis functions used.

#### 9.1.1 Problem definition

Bar element formulated is a 1D element with single dof at each node. The axial displacement is selected as the field variable. The bar is fixed at both ends. Material and geometrical properties of the element is given in tables 9.1 and 9.2 respectively. Section 8.1 describes in more detail the process of formation of bar element.

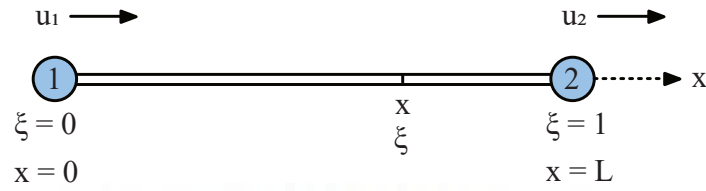
FIGURE 9.1: Straight bar element in physical space with  $L = 1$ .

TABLE 9.1: Bar material properties

Young's Modulus	$E$	1
Density	$\rho$	1
Length	$L$	1
Area	$A$	1

TABLE 9.2: Bar geometrical properties

NURBS basis order	$p$	2
Knot vector	$\xi$	$\{0,0,0,1/3,2/3,1,1,1\}$
Guass point	$ngp$	3

### 9.1.2 Solution

Solving equation (3.25) using NURBS based IGA yields natural frequencies  $\omega_n^h$ .  $\omega_n^h$  are normalized with respect to exact solution and plotted versus the corresponding modes  $n$ , normalized with respect to the total number of dofs  $N$ . To produce the figure 9.2 a number of dof  $N = 100$  is employed to produce a smooth curve. The exact solution for free vibration of bar is given by

$$\omega_n = (n\pi), \text{ with } n = 1, 2, 3 \dots \quad (9.1)$$

$(\omega_n^h/\omega_n) = 1$  means that the numerical frequency and the analytical frequency are the same. In practice, the frequencies will always obey the relationship

$$\omega_n \leq \omega_n^h \quad \text{for } n = 1, \dots, n_{eq} \quad (9.2)$$

and so it is expected for  $(\omega_n^h/\omega_n)$  to be greater than 1, with larger values indicating decreased accuracy.

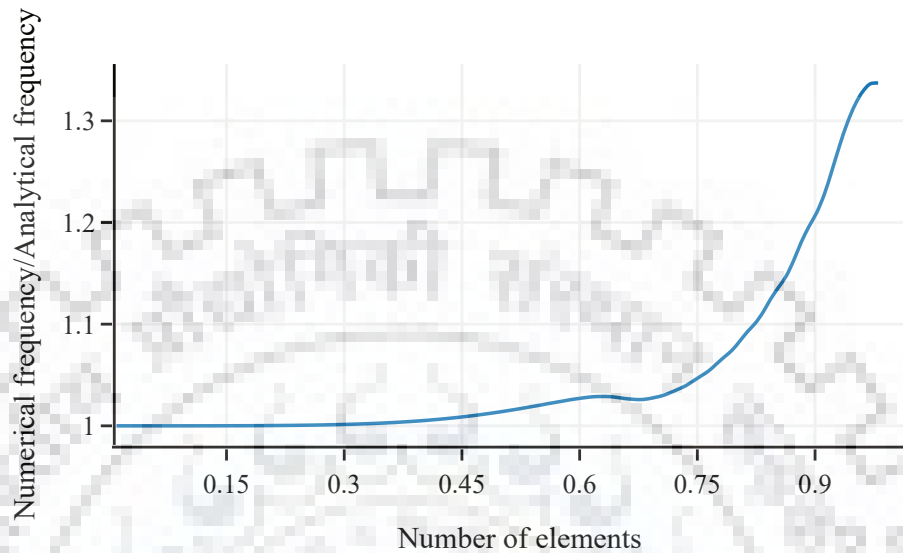


FIGURE 9.2: Fixed-fixed rod problem: normalized discrete spectra using NURBS.

As can be seen in figure 9.2,  $\omega_n^h/\omega_n = 1$ , implying that numerical frequency is identical to analytical frequency, for first 40 modes. Considering how insignificant higher mode frequencies are in context of structural dynamics of structures the results can be termed as good enough for further studies and modifications.

## 9.2 Beam element free vibration

### 9.2.1 Problem definition

Another attractive 1D problem is the natural transversal vibrations of an Euler-Bernoulli beam. Beam element formulated is a 1D element with single dof and is rotation free. Since no boundary condition is being applied to rotations at the ends, beam is simply supported at ends. Material and geometrical properties of the element is given in tables 9.3 and 9.4 respectively. Section 8.2 describes in more detail the process of formation of bar element.

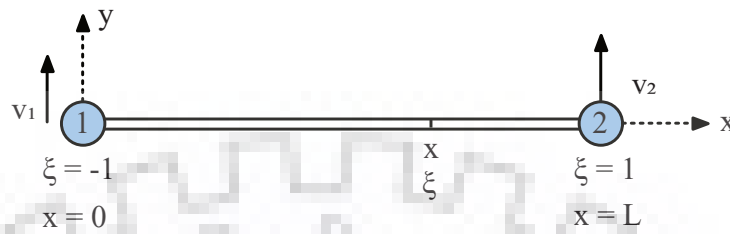
FIGURE 9.3: Straight beam element in physical space with  $L = 1$ .

TABLE 9.3: Beam material properties

Young's Modulus	$E$	1
Density	$\rho$	1
Length	$L$	1
Area	$A$	1

TABLE 9.4: Beam geometrical properties

NURBS basis order	$p$	3
Knot vector	$\xi$	$\{0,0,0,0,1/3,2/3,1,1,1,1\}$
Guass point	$ngp$	4

## 9.2.2 Solution

Solving equation (3.25) using NURBS based IGA yields natural frequencies  $\omega_n^h$ .  $\omega_n^h$  are normalized with respect to exact solution and plotted versus the corresponding modes  $n$ , normalized with respect to the total number of dofs  $N$ . To produce the spectra of figure 9.2 a number of dof  $N = 100$  is employed to produce a smooth curve. The exact solution for free vibration of bar is given by

$$\omega_n = (n\pi)^2, \text{ with } n = 1, 2, 3 \dots \quad (9.3)$$

$(\omega_n^h/\omega_n) = 1$  means that the numerical frequency and the analytical frequency are the same. In practice, the frequencies will always obey the relationship

$$\omega_n \leq \omega_n^h \quad \text{for } n = 1, \dots, n_{eq} \quad (9.4)$$

and so it is expected for  $(\omega_n^h/\omega_n)$  to be greater than 1, with larger values indicating decreased accuracy. As can be seen in figure 9.4,  $\omega_n^h/\omega_n = 1$ , implying that numerical

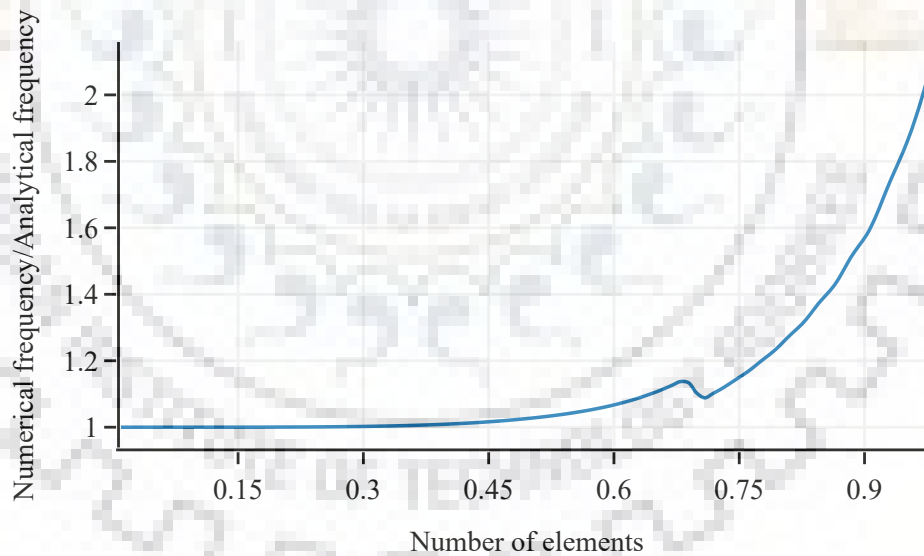


FIGURE 9.4: Fixed-fixed beam problem: normalized discrete spectra using NURBS.

frequency is identical to analytical frequency, for first 40 modes. Considering how insignificant higher mode frequencies are in context of structural dynamics of structures the results can be termed as good enough for further studies and modifications.

## 9.3 Plate element

### 9.3.1 Problem definition

A four noded rectangular plate element shown in figure 9.5 is formulated for the bending analysis purpose. The element is based on Reissner-Mindlin theory and its formulation has been discussed in sections 3.6 and 8.3. Material and geometrical properties of the element is given in tables 9.5 and 9.6 respectively.

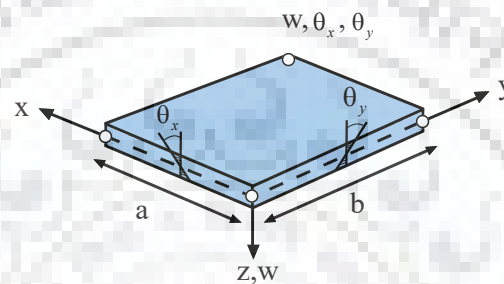


FIGURE 9.5: Basic flat rectangular plate element with 3 dof at a node

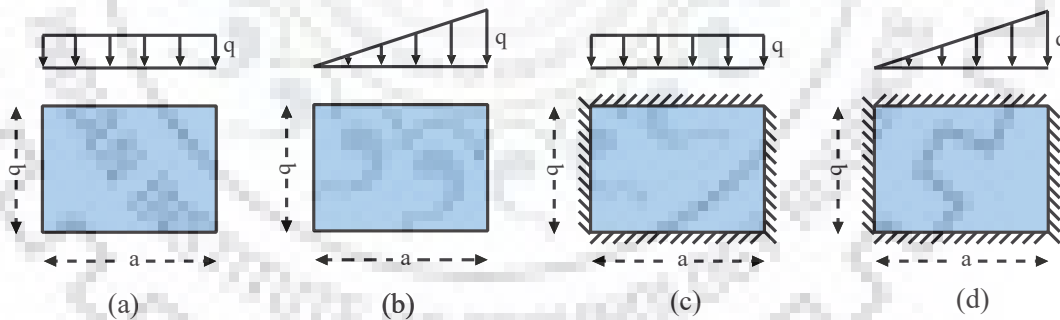


FIGURE 9.6: Various loading and support conditions on plate element: (a) Simply supported on all sides with uniformly distributed load; (b) Simply supported on all sides with uniformly varying load; (c) Clamped on all sides with uniformly distributed load; (d) Clamped on all sides with uniformly varying load

To validate the objective, several numerical examples of rectangular plate with different aspect ratio, different boundary conditions and different loading conditions are solved and the results are compared with available analytical or other numerical solution.

TABLE 9.5: Beam geometrical properties

NURBS basis order	$p$	2
Knot vector	$\xi$	{0,0,0,1,1,1}
Knot vector	$\eta$	{0,0,0,1,1,1}
Guass point	$ngp$	3

TABLE 9.6: Beam material properties

Young's Modulus	$E$	$10.92 \times 10^6$
Poisson's ratio	$\nu$	0.3

Convergence studies were carried out for both simply supported (SSSS) and clamped plate (CCCC) which were subjected to uniformly distributed load and uniformly varying load. This problem has been computed with different mesh refinements (25, 36, 49, 64, 81, 100, 121, 256, 441).

### 9.3.2 Solution

The solution for different boundary conditions and loading conditions are as follows:

#### **Simply supported on all sides subjected to uniformly distributed loading**

Table 9.7 shows the results of dimensionless central deflections for different mesh density, obtained by NURBS based IGA. These results are compared with analytical solution for the same boundary and loading conditions obtained from [20]. Relative error is calculated for different mesh density between numerical and analytical solution and plotted in figure 9.7.

#### **Simply supported on all sides subjected to uniformly varying loading**

Table 9.8 shows the results of dimensionless central deflections for different mesh density, obtained by NURBS based IGA. These results are compared with analytical solution for the same boundary and loading conditions obtained from [20]. Relative error is calculated for different mesh density between numerical and analytical solution and plotted in figure 9.8.

#### **Clamped on all sides subjected to uniformly distributed loading**



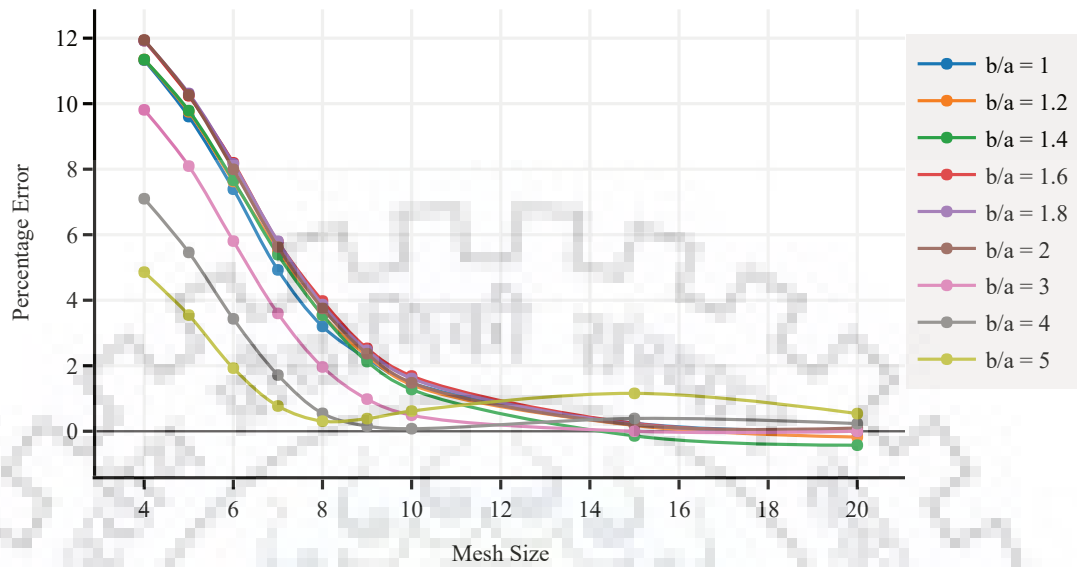


FIGURE 9.7: The relative error for central deflection of simply supported rectangular plate under uniformly distributed loading

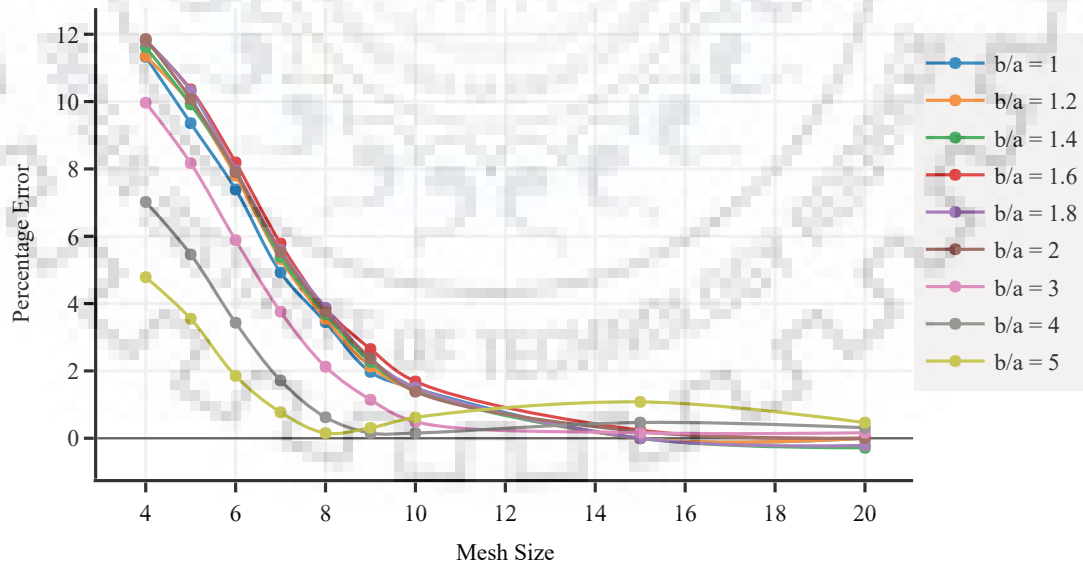


FIGURE 9.8: The relative error for central deflection of simply supported rectangular plate under uniformly varying loading

TABLE 9.7: Central deflection for all simply supported rectangular plate with UDL

Mesh	Aspect ratio (b/a)									
	1	1.2	1.4	1.6	1.8	2	3	4	5	
4	0.003601	0.004995	0.006246	0.007312	0.008196	0.008922	0.011029	0.011911	0.012336	
5	0.003673	0.005087	0.006359	0.007446	0.008349	0.009092	0.011241	0.012116	0.012507	
6	0.003764	0.005207	0.006509	0.007625	0.008555	0.009320	0.011521	0.012381	0.012721	
7	0.003856	0.005335	0.006671	0.007816	0.008771	0.009557	0.011790	0.012604	0.012866	
8	0.003926	0.005437	0.006802	0.007972	0.008948	0.009749	0.011989	0.012746	0.012934	
9	0.003975	0.005510	0.006899	0.008088	0.009077	0.009887	0.012109	0.012798	0.012918	
10	0.004005	0.005558	0.006961	0.008164	0.009162	0.009976	0.012175	0.012810	0.012886	
15	0.004052	0.005634	0.007063	0.008284	0.009292	0.010108	0.012226	0.012766	0.012823	
20	0.004060	0.005647	0.007080	0.008302	0.009311	0.010125	0.012230	0.012788	0.012901	
<b>Analytical Result</b>	0.004060	0.005640	0.007050	0.008300	0.009310	0.010130	0.012230	0.012820	0.012970	
<b>% Relative Error</b>	-0.008958	-0.116567	-0.419004	-0.027294	-0.006079	0.054035	0.002189	0.245719	0.529962	

TABLE 9.8: Central deflection for all simply supported plate with UVL

Mesh	Aspect ratio (b/a)									
	1	1.2	1.4	1.6	1.8	2	3	4	5	
4	0.001801	0.002498	0.003123	0.003656	0.004098	0.004461	0.005514	0.005955	0.006168	
5	0.001837	0.002543	0.003180	0.003723	0.004175	0.004546	0.005620	0.006058	0.006254	
6	0.001882	0.002604	0.003255	0.003812	0.004277	0.004660	0.005761	0.006190	0.006361	
7	0.001928	0.002667	0.003335	0.003908	0.004386	0.004779	0.005895	0.006302	0.006433	
8	0.001963	0.002718	0.003401	0.003986	0.004474	0.004875	0.005995	0.006373	0.006467	
9	0.001987	0.002755	0.003449	0.004044	0.004539	0.004943	0.006054	0.006399	0.006459	
10	0.002002	0.002779	0.003481	0.004082	0.004581	0.004988	0.006087	0.006405	0.006443	
15	0.002026	0.002817	0.003531	0.004142	0.004646	0.005054	0.006113	0.006383	0.006412	
20	0.002030	0.002823	0.003540	0.004151	0.004655	0.005062	0.006115	0.006394	0.006451	
<b>Analytical Result</b>	0.002030	0.002820	0.003530	0.004150	0.004650	0.005060	0.006120	0.006410	0.006480	
<b>% Relative Error</b>	-0.008958	-0.116567	-0.276767	-0.027294	-0.113612	-0.044726	0.083886	0.245719	0.453210	

Table 9.9 shows the results of dimensionless central deflections for different mesh density, obtained by NURBS based IGA. These results are compared with analytical solution for the same boundary and loading conditions obtained from [20]. Relative error is calculated for different mesh density between numerical and analytical solution and plotted in figure 9.9.

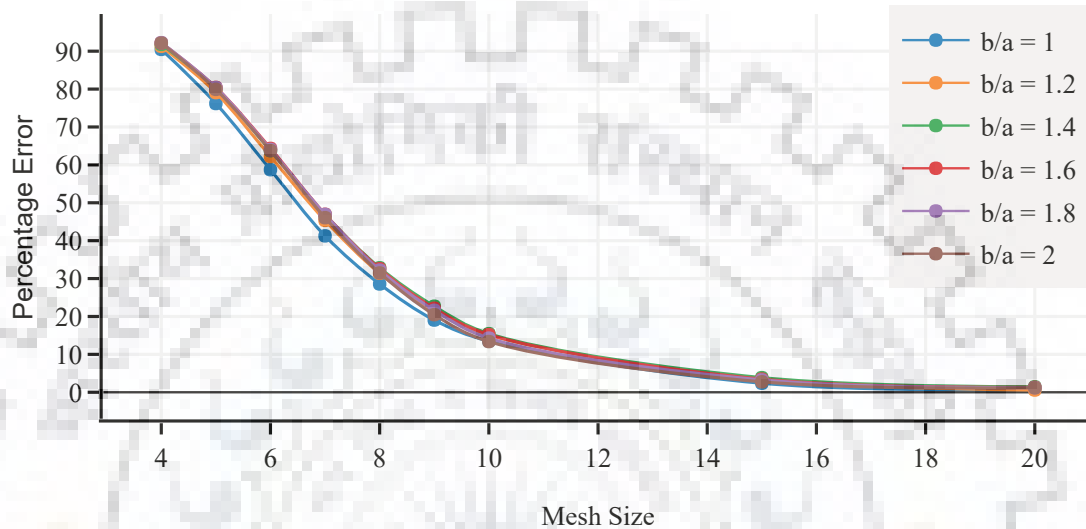


FIGURE 9.9: The relative error for central deflection of clamped rectangular plate under uniformly distributed loading

### **Clamped on all sides subjected to uniformly varying loading**

Table 9.10 shows the results of dimensionless central deflections for different mesh density, obtained by NURBS based IGA. These results are compared with analytical solution for the same boundary and loading conditions obtained from [20]. Relative error is calculated for different mesh density between numerical and analytical solution and plotted in figure 9.10.

Do note that the percentage relative error given in the tables are only for the last mesh size i.e. 21 by 21. As can be seen in the graphs, the solution converges rapidly for all support and loading conditions with increasing mesh size and almost approaches zero for the mesh size of 21 by 21. Having successfully applied NURBS based IGA to the static problem the next problem is of adaptive refinement in the framework of PHT-splines based IGA.

TABLE 9.9: Central deflection for all clamped rectangular plate with UDL

Mesh	Aspect ratio (b/a)					
	1	1.2	1.4	1.6	1.8	2
4	0.000124	0.000148	0.000167	0.000182	0.000193	0.000202
5	0.000299	0.000364	0.000414	0.000453	0.000483	0.000505
6	0.000518	0.000647	0.000746	0.000820	0.000876	0.000919
7	0.000736	0.000943	0.001101	0.001218	0.001305	0.001370
8	0.000902	0.001178	0.001390	0.001547	0.001661	0.001744
9	0.001018	0.001350	0.001605	0.001791	0.001925	0.002020
10	0.001093	0.001462	0.001746	0.001952	0.002098	0.002200
15	0.001225	0.001663	0.001994	0.002223	0.002374	0.002469
20	0.001253	0.001705	0.002044	0.002275	0.002422	0.002511
<b>Analytical Result</b>	0.001260	0.001720	0.002070	0.002300	0.002450	0.002540
<b>% Relative Error</b>	0.549823	0.857703	1.251639	1.095300	1.133454	1.134758

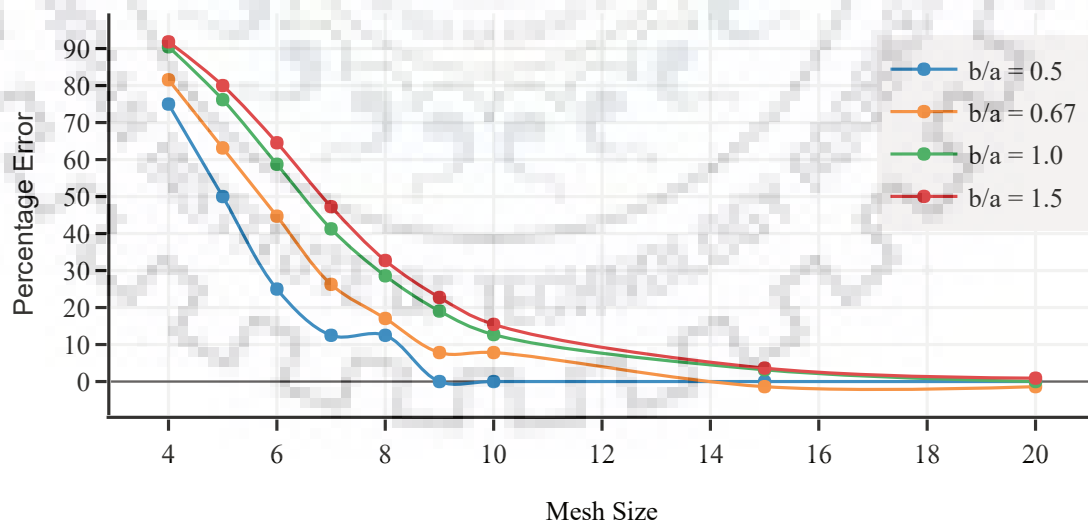


FIGURE 9.10: The relative error for central deflection of clamped rectangular plate under uniformly varying loading

TABLE 9.10: Central deflection for all clamped rectangular plate with UVL

Mesh	Aspect ratio (b/a)			
	0.5	0.67	1	1.5
4	0.000021	0.000036	0.000062	0.000087
5	0.000041	0.000080	0.000150	0.000218
6	0.000058	0.000124	0.000259	0.000393
7	0.000068	0.000160	0.000368	0.000582
8	0.000074	0.000183	0.000451	0.000737
9	0.000076	0.000197	0.000509	0.000853
10	0.000077	0.000205	0.000546	0.000929
15	0.000079	0.000217	0.000613	0.001060
20	0.000079	0.000220	0.000627	0.001086
<b>Analytical Result</b>	0.000080	0.000217	0.000630	0.001100
<b>% Relative Error</b>	0.766258	-1.308450	0.549823	1.293441

## 9.4 Adaptive IGA

Numerical analysis is no better if it lacks the capability of automatic adaptive mesh refinement. NURBS based IGA is incapable of such because of its tensor product structure in meshes (see figure 7.1). Hence in this study PHT-splines based IGA framework is prepared to exploit the automatic adaptive refinement capabilities and see if it is any better. Adaptive isogeometric method is usually based on the following loop

SOLVE  $\longrightarrow$  ESTIMATE  $\longrightarrow$  MARK  $\longrightarrow$  REFINE

### 9.4.1 Problem definition

A benchmark problem in the domain of adaptive refinement is that which has a singularity at the re-entrant corner in the domain. That problem in this study is of a cantilever beam. A cantilever beam based on Euler-Bernoulli beam theory (section 3.5.1) is analyzed in the framework of PHT spline. The geometry description is given in figure 9.11 and table 9.11.

The domain is discretized using 4 PHT patches as can be seen in figure 9.14. The patches are made to conform at their boundaries. The beam is analyzed with cubic basis functions, the initial mesh for which is shown in figure 9.14(A). The problem is computed and adaptively refined with the error estimator outlined in section 7.4. For marking which elements to refine at each consecutive steps, a Dörfler marking scheme was used with parameter  $\rho = 0.5$  which implies that each step the elements contributing at least 50% of the error are refined. Hence, lower the  $\theta$ , more refinement steps needed to solve the problem. Stresses (and strains) are evaluated and plotted for refinement steps of 1<sup>st</sup>, 6<sup>th</sup> and 12<sup>th</sup>. The stresses follows the sign convention of figure 9.12.

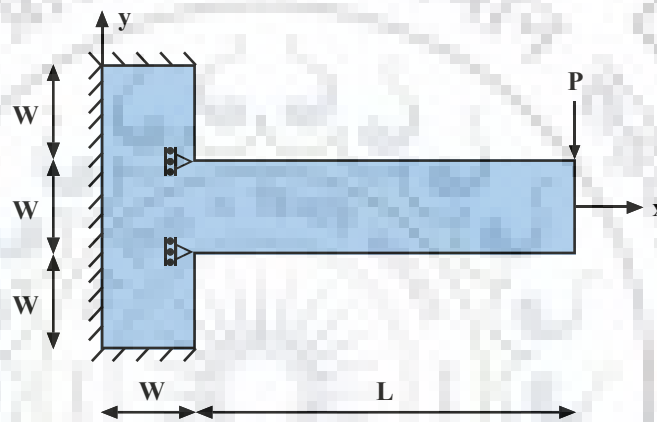


FIGURE 9.11: Geometry description of the cantilever beam problem

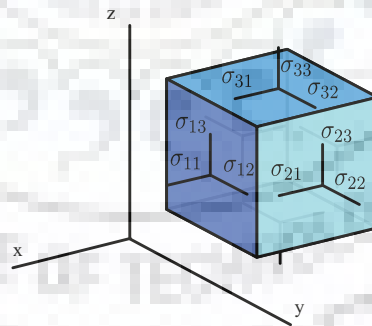


FIGURE 9.12: Components of stresses in three dimensions.

## 9.4.2 Solution

The problem statement described in the preceding section is solved in the framework of PHT-splines and results obtained show good nature. The problem takes 12 refinements to reduce the estimated relative error to  $10^{-2}$ . The refined meshes for initial

TABLE 9.11: Cantilever beam material properties

Young's Modulus	$E$	1000
Poisson's ratio	$\nu$	0.3
Length	$L$	8
Width	$W$	2

mesh, mesh at  $6^{th}$  and last,  $12^{th}$ , are shown in figure 9.14(A), (B) and (C) respectively. Convergence plot is shown in figure 9.13. As there are two locations with a stress concentration or singularities, a clustering of elements is expected and can be seen as well in figure 9.14(C). Contour plots of stresses for the different refinement levels is shown in figures 9.15 to 9.17. It can be observed that elements with large error contribution are successfully marked and refined by adaptivity procedure until the error reduces below a threshold value.

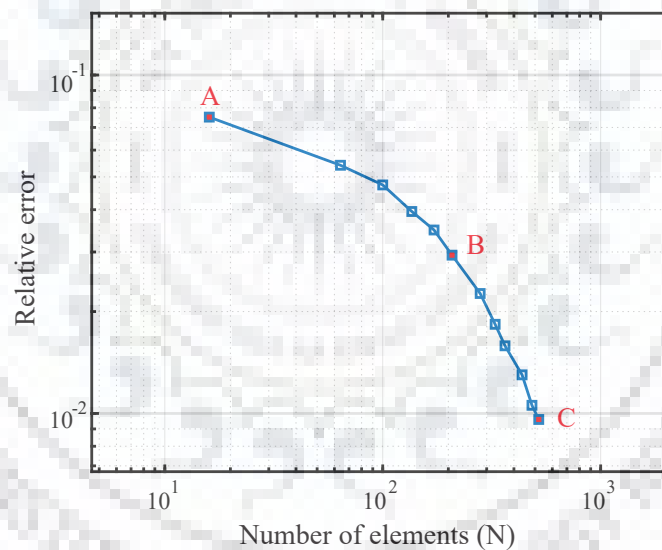


FIGURE 9.13: Relative error in energy norm vs. the degree of freedom using adaptive refinement for cantilever beam



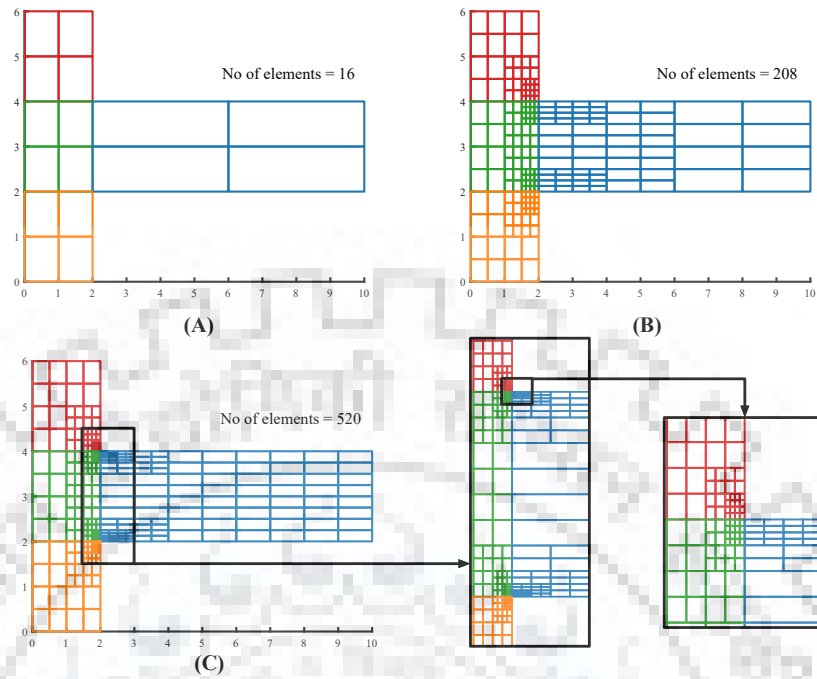


FIGURE 9.14: Mesh of cantilever beam at refinement level of (A) 1<sup>st</sup>, (B) 6<sup>th</sup>, (C) 12<sup>th</sup>. Different color indicates different patches used in the model.

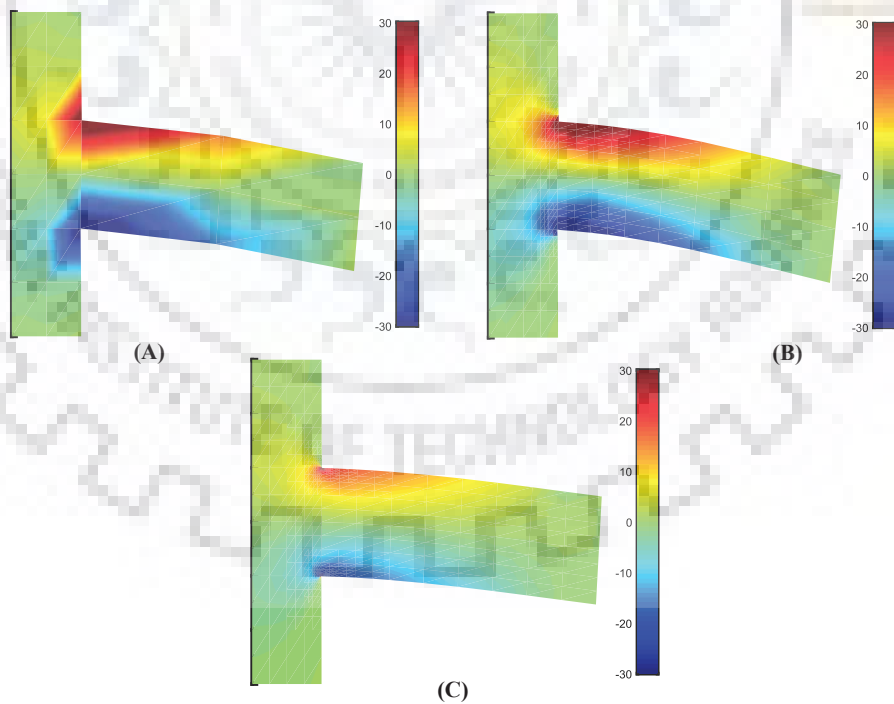


FIGURE 9.15: Contour plots of  $\sigma_{11}$  stress components of a deflected cantilever beam. (A) is the contour at 1<sup>st</sup> refinement; (B) is the contour at 6<sup>th</sup> refinement; (C) is the contour at 12<sup>th</sup> refinement.

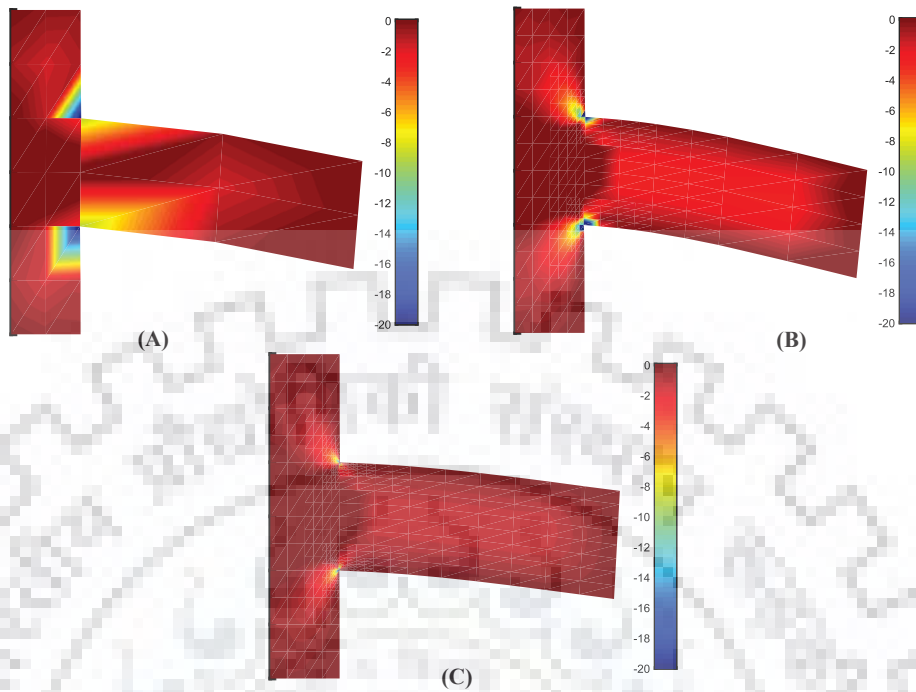


FIGURE 9.16: Contour plots of  $\sigma_{12}$  stress components of a deflected cantilever beam. (A) is the contour at 1<sup>st</sup> refinement; (B) is the contour at 6<sup>th</sup> refinement; (C) is the contour at 12<sup>th</sup> refinement.

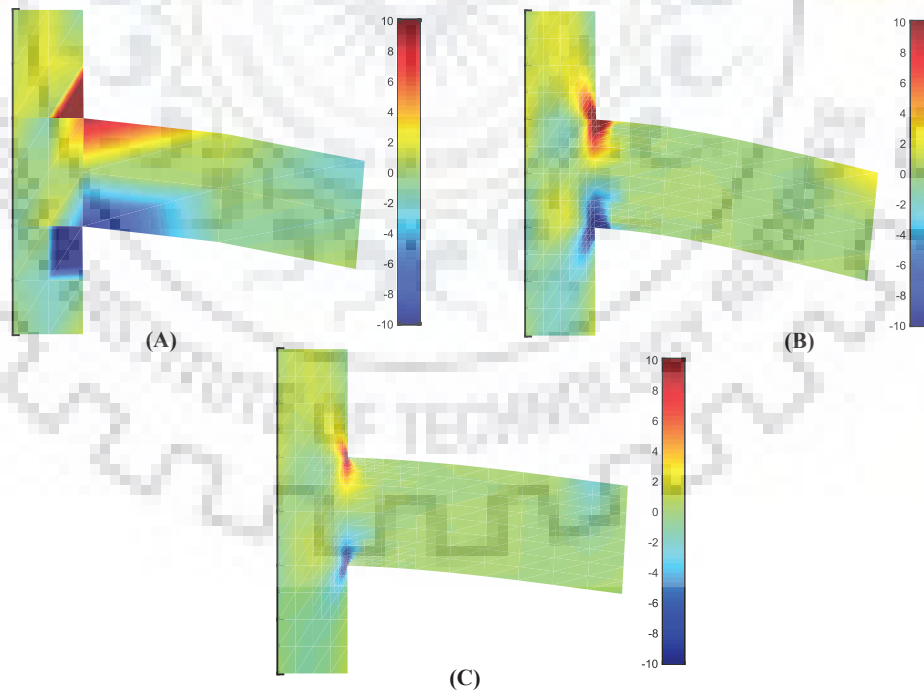


FIGURE 9.17: Contour plots of  $\sigma_{22}$  stress components of a deflected cantilever beam. (A) is the contour at 1<sup>st</sup> refinement; (B) is the contour at 6<sup>th</sup> refinement; (C) is the contour at 12<sup>th</sup> refinement.

## Chapter 10

# Conclusion and Recommendation

### 10.1 General

NURBS based IGA was developed with the purpose of reducing the analysis time by using the same geometrical description in modeling and analysis domain. However, with increasing research NURBS based IGA proved to be better than FEM. The method promised not only reduced analysis time but better results using less computation power. In this study the advantages that the NURBS based IGA offered were studied in the static and dynamic domains. Adaptive refinement technique in context of IGA gave promising results. The conclusions drawn and scope of future work that can be done are mentioned below.

### 10.2 Conclusion

In this study, the NURBS based IGA is successfully applied to solve Reissner-Mindlin static plate problems and free vibration frequency of bar and beam element. Thin plates with various shapes and boundary conditions are numerically simulated and presented to validate the proposed approach. The numerical examples are compared with available analytical methods. The free vibration frequencies of various modes of bar and beam element are compared with analytical frequencies. IGA based on PHT-splines is developed to explore the domain of adaptive local refinement. IGA based on PHT-splines is applied to a cantilever beam problem. The results demonstrate the robustness and efficiency of IGA for these problems and its high accuracy and fast convergence rate.

Following conclusions are drawn from the work:

- The relative percentage error of deflection of plate with aspect ratio of 1 for simply supported condition and uniformly distributed loading and mesh density of 5 by 5 is 11.3% which rapidly converges to 3.2% for a finer mesh of 21\*21.
- The relative percentage error of deflection of plate with aspect ratio of 5 for simply supported condition and uniformly varying loading and mesh density of 5 by 5 is 4.86% which rapidly converges to 0.308 % for a mesh density of only 9\*9.
- In this problem, the simply supported plate with higher aspect ratio produces convergent solutions in a much coarser mesh of 9 by 9 and that too without the need for selective integration.
- The relative percentage error of deflection of plate clamped on all sides and subjected to UDL is around 90% for all aspect ratio and mesh density of 5 by 5 which converges even rapidly than simply supported plate to error of around 1% for a mesh density of 21 by 21.
- The initial relative percentage error for clamped condition is much higher than that for simply supported condition. This is expected, as the plate is clamped it locks the boundary elements. This locking is propagated to inside elements as well. For a initial coarse mesh almost all elements are locked which results in an overall stiffer element and large initial relative error. As per Cook et al. [5] the effect of boundary constraints on internal elements becomes negligible after a mesh size of 12 which is verified by the results for clamped plate.
- The results of plate bending demonstrate the robustness and efficiency of IGA for plate problems and its high accuracy and fast convergence rate.
- Even though Guass-quadrature is not optimal in context of IGA but with mesh refinement the solution converges as expected.
- For bar element the numerically computed free vibration frequencies matched with analytical frequencies for almost first 40 modes.
- For beam element the numerically computed free vibration frequencies matched with analytical frequencies also for almost first 40 modes.
- Considering how insignificant the contribution of higher modes is in deformation of a solid, the results of free vibration analysis are termed as acceptable.

- PHT-spline formulation aids in adaptive refinement which is cumbersome if NURBS based IGA formulation is used.
- Cantilever beam problem solved in the PHT-spline based IGA framework yielded promising results. Elements were refined automatically at re-entrant corners and areas of stress concentrations. Contours of stresses shown clearly demonstrate the reduction in stress concentration with increasing refinement levels. A total of 12 refinement steps were needed to reduce the error below the threshold limit.
- The following inferences are made about the variation of  $\sigma_{11}$ :
  1. It is known that the variation of axial stress in  $x$  direction should be zero at the free end throughout the section and should gradually increase as one goes from the free end to the fixed end, also at any particular section it should be zero at the neutral axis, which is at the middle for a rectangular cross section and should gradually increase to maximum at the top and the bottom.
  2. The stress at the top would be tensile and that at the bottom would be compressive. The points having maximum stress would be near the re-entrant corners, at the top and bottom of the beam, the stress at the top and bottom fibre would reduce as one moves away from the fixed end towards the free end. Along the mid-section the axial stress would be zero throughout.
  3. Hence, the compressive and tensile axial stress contour in the  $x$  direction as observed in the obtained contour plots are as per expectations.
- The following inferences are made about the variation of  $\sigma_{12}$ :
  1. For the given loading, shear force remains constant from the free end up to the fixed end. The variation of shear force along a section is parabolic with zero value at the extreme fibres and maximum value at the neutral axis.
  2. Hence, the shear force  $\sigma_{12}$ , should be maximum along the central line of the beam and gradually decreasing towards the top and bottom faces.
  3. The variation of shear forces hence obtained in the contour plots, is in accordance with expectations.
- The following inferences are made about the variation of  $\sigma_{22}$ :
  1. Speaking in a strict theoretical sense, as per stress calculation procedures, no axial stress in the  $y$  direction should be obtained for this loading. But, axial

stresses in  $y$  direction are expected at points where load is getting transferred from one part of the beam to another. For example, at points just beneath the point load, the load is getting distributed in the beam through vertical load transfer and this entails an axial stress in  $y$  directions at those points.

2. At the fixed end, there is some vertical load transfer expected at the bottom re-entrant corner and this entails an axial stress in  $y$  direction at those points. Similarly, some tension as an axial stress in  $y$  direction is expected at the top re-entrant corner.
3. Hence, the compressive and tensile axial stress contour in the  $y$  direction as observed in the obtained contour plots are as per expectation.

- IGA is computationally cheap because the size of element matrices is smaller than that for FEM.
- For 1D problem any reduction in size of element matrices is nullified by the lengthy procedure of computing shape functions and transforming variables in different spaces. Hence the procedure becomes expensive for a simple 1D problem.
- IGA is capable of directly interacting with CAD system.
- Nodes in FEM are interpolatory whereas control points in IGA are not.
- Applying Dirichlet boundary condition inside a domain is not a direct technique in IGA.

### 10.3 Recommendations for further work

IGA is a promising new approach which can successfully replace the traditional FEM used in industry. In order for it to be the standard technique in the industry it has to be robust efficient. IGA needs to be enhanced by continuous research, development, and testing.

Following work is suggested for the future:

- The work presented in this dissertation can be extended to 3D problems.
- Beam element formulated in this study was rotation free and simply supported. To further enhance the results, boundary condition may be applied to rotations at

the ends of the beam by using either penalty method or the Lagrangian multiplication method.

- For vibrational analysis, a parametric study can be performed with different order of NURBS basis functions and the results can be compared with the results obtained using FEM.
- Isogeometric vibration analysis of plates based on Reissner-Mindlin plate element can be done to investigate the advantages of smoothness of NURBS basis functions in two dimensions.
- Formulating shell element in IGA will help in investigating the exact geometry description property of IGA.
- PHT-spline based IGA can be extended to problems of larger deformation, fracture mechanics and non-linear analyses.
- Gaussian quadrature is not optimal for IGA because it fits a polynomial through the results of shape functions evaluations at Gauss points. Whereas in NURBS based IGA these shape functions are rational which cannot be exactly reproduced with the help of fitted polynomial. Hence using optimal quadrature rule in context of IGA can be done as well.



# Bibliography

- [1] Cosmin Anitescu, Md Naim Hossain, and Timon Rabczuk. “Recovery-based error estimation and adaptivity using high-order splines over hierarchical T-meshes”. In: *Computer Methods in Applied Mechanics and Engineering* 328 (2018), pp. 638–662. DOI: <https://doi.org/10.1016/j.cma.2017.08.032>.
- [2] Yuri Bazilevs et al. “Isogeometric analysis using T-splines”. In: *Computer Methods in Applied Mechanics and Engineering* 199.5-8 (2010), pp. 229–263.
- [3] DJ Benson et al. “A large deformation, rotation-free, isogeometric shell”. In: *Computer Methods in Applied Mechanics and Engineering* 200.13-16 (2011), pp. 1367–1378.
- [4] Michael J Borden et al. “Isogeometric finite element data structures based on Bézier extraction of NURBS”. In: *International Journal for Numerical Methods in Engineering* 87.1-5 (2011), pp. 15–47.
- [5] Robert D Cook et al. *Concepts and applications of finite element analysis*. John Wiley & Sons, 2007.
- [6] J Austin Cottrell, Thomas JR Hughes, and Yuri Bazilevs. *Isogeometric analysis: toward integration of CAD and FEA*. John Wiley & Sons, 2009.
- [7] J Austin Cottrell et al. “Isogeometric analysis of structural vibrations”. In: *Computer methods in applied mechanics and engineering* 195.41-43 (2006), pp. 5257–5296.
- [8] Jiansong Deng et al. “Polynomial splines over hierarchical T-meshes”. In: *Graphical models* 70.4 (2008), pp. 76–86.
- [9] Willy Dörfler. “A convergent adaptive algorithm for Poisson’s equation”. In: *SIAM Journal on Numerical Analysis* 33.3 (1996), pp. 1106–1124.



- [10] Gerald E Farin and Gerald Farin. *Curves and surfaces for CAGD: a practical guide*. Morgan Kaufmann, 2002.
- [11] Sangamesh Gondegaon and Hari K Voruganti. "Static structural and modal analysis using Isogeometric analysis". In: *Journal of Theoretical and Applied Mechanics* 46.4 (2016), pp. 36–75.
- [12] T.J.R. Hughes, J.A. Cottrell, and Y. Bazilevs. "Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement". In: *Computer Methods in Applied Mechanics and Engineering* 194.39 (2005), pp. 4135–4195. DOI: <https://doi.org/10.1016/j.cma.2004.10.008>.
- [13] Josef Kiendl et al. "Isogeometric shell analysis with Kirchhoff–Love elements". In: *Computer Methods in Applied Mechanics and Engineering* 198.49-52 (2009), pp. 3902–3914.
- [14] CS Krishnamoorthy. *Finite element analysis: theory and programming*. Tata McGraw-Hill Education, 1994.
- [15] Yicong Lai et al. "Integrating CAD with Abaqus: a practical isogeometric analysis software platform for industrial applications". In: *Computers & Mathematics with Applications* 74.7 (2017), pp. 1648–1660.
- [16] David F Rogers. *An introduction to NURBS: with historical perspective*. Elsevier, 2000.
- [17] Hanan Samet. *Foundations of multidimensional and metric data structures*. Morgan Kaufmann, 2006.
- [18] Michael A Scott et al. "Isogeometric finite element data structures based on Bézier extraction of T-splines". In: *International Journal for Numerical Methods in Engineering* 88.2 (2011), pp. 126–156.
- [19] Thomas W Sederberg et al. "T-splines and T-NURCCs". In: *ACM transactions on graphics (TOG)* 22.3 (2003), pp. 477–484.
- [20] Stephen P Timoshenko and Sergius Woinowsky-Krieger. *Theory of plates and shells*. McGraw-hill, 1959.
- [21] Ping Wang et al. "Adaptive isogeometric analysis using rational PHT-splines". In: *Computer-Aided Design* 43.11 (2011), pp. 1438–1448.

- [22] Peng Yu et al. “Adaptive Isogeometric analysis for plate vibrations: An efficient approach of local refinement based on hierarchical a posteriori error estimation”. In: *Computer Methods in Applied Mechanics and Engineering* 342 (2018), pp. 251–286. DOI: <https://doi.org/10.1016/j.cma.2018.08.010>.
- [23] Olgierd Cecil Zienkiewicz and Jian Zhong Zhu. “The superconvergent patch recovery and a posteriori error estimates. Part 1: The recovery technique”. In: *International Journal for Numerical Methods in Engineering* 33.7 (1992), pp. 1331–1364.
- [24] Olgierd Cecil Zienkiewicz and Jian Zhong Zhu. “The superconvergent patch recovery and a posteriori error estimates. Part 2: Error estimates and adaptivity”. In: *International Journal for Numerical Methods in Engineering* 33.7 (1992), pp. 1365–1382.

