

Isogeometric analysis based on Geometry Independent Field approximation (GIFT) and Polynomial Splines over Hierarchical T-meshes

**Die isogeometrische Analysis basierend auf der
geometrieunabhängigen Feldnäherung (GIFT) und
polynomialen Splines über hierarchischen T-Netzen**

DISSERTATION

By

Md Naim Hossain

zur Erlangung des akademischen Grades eines
Doktor-Ingenieur (Dr.-Ing.)
an der Fakultät Bauingenieurwesen
der
Bauhaus Universität Weimar

vorgelegt von
M.Sc. Md Naim Hossain
(externe Doktorand)
geboren am 19. December 1984
in Dhaka, Bangladesch

Mentor:

Prof. Dr.-Ing. Timon Rabczuk, Bauhaus Universität Weimer

September, 2018

Dedication

To my family for their endless love and support.

Abstract

This thesis addresses an adaptive higher-order method based on a Geometry Independent Field approximation (GIFT) of polynomial/rational splines over hierarchical T-meshes (PHT/RHT-splines).

In isogeometric analysis, basis functions used for constructing geometric models in computer-aided design (CAD) are also employed to discretize the partial differential equations (PDEs) for numerical analysis. Non-uniform rational B-Splines (NURBS) are the most commonly used basis functions in CAD. However, they may not be ideal for numerical analysis where local refinement is required.

The alternative method GIFT deploys different splines for geometry and numerical analysis. NURBS are utilized for the geometry representation, while for the field solution, PHT/RHT-splines are used.

PHT-splines not only inherit the useful properties of B-splines and NURBS, but also possess the capabilities of local refinement and hierarchical structure. The smooth basis function properties of PHT-splines make them suitable for analysis purposes. While most problems considered in isogeometric analysis can be solved efficiently when the solution is smooth, many non-trivial problems have rough solutions. For example, this can be caused by the presence of re-entrant corners in the domain. For such problems, a tensor-product basis (as in the case of NURBS) is less suitable for resolving the singularities that appear since refinement propagates throughout the computational domain. Hierarchical bases and local refinement (as in the case of PHT-splines) allow for a more efficient way to resolve these singularities by adding more degrees of freedom where they are necessary.

In order to drive the adaptive refinement, an efficient recovery-based error estimator is proposed in this thesis. The estimator produces a recovery solution which is a more accurate approximation than the computed numerical solution. Several two- and three-dimensional numerical investigations with PHT-splines of higher order and continuity prove that the proposed method is capable of obtaining results with higher accuracy, better convergence, fewer degrees of freedom and less computational cost than NURBS for smooth solution problems. The adaptive GIFT method utilizing PHT-splines with the recovery based error estimator is used for solutions with discontinuities or singularities where adaptive local refinement in particular domains of interest achieves higher accuracy with fewer degrees of freedom. This method also proves that it can handle complicated multi-patch domains for two- and three-dimensional problems outperforming uniform refinement in terms of degrees of freedom and computational cost.

Acknowledgement

I would like to acknowledge my PhD thesis supervisor, Prof. Timon Rabczuk, whose expertise, patience, encouragement and insightful guidance helped me a lot during the entire time of my thesis work.

Additionally, I would like to acknowledge and special thanks for constant help, opportune advice and continuous support provided by Dr. Cosmin Anitescu. Without his help, it won't be possible for me to finish the thesis. It is my pleasure to work with him throughout the entire time in ITN INSIST project. He always came up with new ideas and found an exciting research direction. His explanations in different numerical problems and Matlab code helped me a lot to implement the PHT and GIFT method in Diffpack. There are no other words to thank him.

In addition, a very special thanks to Prof. Stephane P.A. Bordas to encourage me to be enrolled as a PhD student and also for his insightful guidance at the starting of my PhD work and also to work on the GIFT idea facilitated my thesis work during my stay in Cardiff.

I also would like to thank my mentor Frank Vogel in inuTech GmbH who always believed in me and gave me the opportunity to work in the ITN INSIST project. It will not be possible for me to work in this project without his constant support.

I would like to special thanks all of my colleagues in inuTech GmbH during my thesis work who helped me a lot at various times whenever I faced any kind of problem.

I would also like to thank the whole ITN INSIST training network and all the colleagues with whom I had a very nice time working together and at different times of 3 years of ITN project, we had meetings in Valencia, Weimar, Strobl and in Cardiff. It was a nice journey working with all the people within this ITN network.

I was in Cardiff University for 3 months as an academic exchange researcher due to the ITN INSIST project and there I had a nice time with some of my colleagues there specially Daniel Alves Paladim whose expertise in C++ coding encouraged me a lot to learn Diffpack specially with XFEM toolbox. Also, I would like to thank Pierre Kerfriede, Xuan Peng for the Rhino3D and nice discussion over PHT-spline, and all the colleagues in Cardiff University for their nice insightful research ideas sharing with me and also for the nice 3 months, I spent there in Cardiff University.

A very special thanks to Dr. Gang Xu and Dr. Elena Atroshchenko to work with them for this GIFT topic and nice discussion over this topic.

No words are not enough to thank my wife Fahima Akter. Without her endless love and support it could not be possible for me to focus on the research. It is her faith in me that encourages me to go through the various difficulties I faced throughout my PhD thesis work.

I would also like to stress the help of my parent specially my father and mother who always believed in me and prayed for me. My sister and other family members who encouraged me a lot in different times to carry on my research works. Also, my other family and friends, from the distance or in person, were very supportive right from the start of this thesis.

Finally, I would like to express my gratitude to all the people that compose the ITN INSIST Network who were always helpful and supportive. I also acknowledge ITN INSIST Marie-Curie Initial Training Network Funding which is provided by the EU Seventh Framework Programme (FP7) under grant agreement number 289361.

Md Naim Hossain
Nuremberg, Germany.

Contents

Nomenclature	ii
List of Figures	iv
List of Tables	viii
1 Introduction	2
1.1 Brief history of the important topics covered in this thesis	3
1.1.1 Isogeometric analysis	3
1.1.2 Geometry Independent Field approximaTion (GIFT)	4
1.1.3 Error estimation	5
1.1.4 Local refinement in isogeometric analysis	7
1.1.5 Diffpack	7
1.2 Objectives	9
1.3 Challenges	10
1.4 Contributions	10
1.5 Chapter Overview	11
2 B-splines, NURBS and Bézier extraction	13
2.1 B-splines	13
2.1.1 Knot vector	13
2.1.2 B-spline basis functions	14
2.1.3 B-spline geometries	15
2.1.4 Knot insertion in B-splines	17
2.2 NURBS	17
2.2.1 NURBS basis functions	18
2.2.2 NURBS surfaces and volumes	19
2.2.3 Property of NURBS geometries	19
2.3 Bézier extraction	20
2.3.1 Bézier decomposition	21
2.3.2 Computing the Bézier extraction operator	21
3 Isogeometric Analysis toolbox using Diffpack	23
3.1 Isogeometric Analysis	23
3.1.1 Advantages of Isogeometric Analysis	23
3.1.2 Advantage of IGA in comparison to FEM	24
3.1.3 Drawbacks of NURBS based Isogeometric analysis	25
3.1.4 Mathematical background of linear elasticity using isogeometric analysis	26
3.2 Isogeometric simulator in Diffpack	27
3.2.1 Advantage of Diffpack	27

3.2.2	Application of Diffpack	28
3.2.3	Solving linear elasticity of isogeometric analysis using Diffpack	28
3.2.4	Isogeometric analysis based on Bézier extraction of NURBS in Diffpack	28
3.3	Numerical examples of IGA using NURBS	30
3.3.1	Cylinder subjected to internal pressure	31
3.3.2	Infinite plate circular hole constant in plane tension	31
4	Adaptive Geometry Independent Field approximaTion using PHT-spline and Error es- timation	38
4.1	Geometry Independent Field approximaTion (GIFT)	38
4.1.1	Main idea of GIFT	39
4.1.2	A general framework for GIFT	40
4.2	GIFT for Linear elasticity	42
4.3	Patch test	45
4.4	Adaptive GIFT using PHT-splines	46
4.4.1	Solving and adaptive local refinement with PHT-spline	47
4.5	Recovery-based error estimation	48
4.5.1	Determining the superconvergent point locations	49
4.5.2	Superconvergent patch recovery	51
4.5.3	Marking algorithm	54
5	Polynomial spline over hierarchical T-meshes (PHT-spline)	55
5.1	1D Hierarchical T-meshes	55
5.2	1D PHT-spline	56
5.2.1	1D PHT-spline (degree, $p=3$)	56
5.2.2	1D extension of PHT-spline (degree $p > 3$)	58
5.3	Numerical Example 1D GIFT PHT-spline	60
6	Two Dimensional Polynomial Splines Over Hierarchical T-meshes	64
6.1	2D T-mesh	64
6.2	Tree structure	64
6.3	2D Hierarchical T-meshes	64
6.4	A dimension formula	66
6.5	Basis function representation and refinement in 2D	67
6.5.1	Blossoming with De Casteljaou's algorithm	67
6.5.2	Truncation by zeroing out Bézier coefficients	68
6.5.3	Insertion of new basis functions	69
6.5.4	Geometry mappings	70
6.6	2D higher-order PHT-spline (degree, $p > 3, q > 3$ and continuity $C^\alpha \subseteq C^1, C^\beta \subseteq C^1$)	72
6.6.1	2D refinement algorithm of higher-order PHT-spline	72
6.6.2	Higer order PHT-spline basis function	75
6.7	Numerical Examples	76
6.7.1	Thick cylinder subjected to internal pressure	76
7	3D adaptive GIFT PHT-spline	81
7.1	Polynomial spline over 3D hierarchical T-meshes, degree $p = 3, C^1$ continuity	81
7.1.1	3D T-meshes	82
7.1.2	3D hierarchical T-meshes	82
7.1.3	A dimensional formula for 3D T-meshes	83
7.1.4	Basis function representation and refinement in 3D	84
7.2	Numerical example of 3D adaptive GIFT PHT-spline	84

7.2.1	Hollow sphere subjected to internal pressure	85
7.2.2	Cube with an internal spherical hole in an infinite domain subjected to uni- form tension	86
7.2.3	Solid C^1 elastic “horseshoe” subjected to equal and opposite in-plane flat edge displacements on the soles of the shoe	87
8	Multi-patch analysis of adaptive GIFT	94
8.1	Handling geometries in multipatch	94
8.1.1	Multi-patch domains	94
8.1.2	Multi-patch approach by knot insertion 2D PHT-mesh	94
8.1.3	Multi-patch approach by knot insertion of 3D PHT-mesh	98
8.2	GIFT 2D multi-patch analysis using PHT-spline with the approach by knot insertion .	99
8.3	GIFT 3D multi-patch analysis using PHT-spline with the approach by knot insertion .	99
8.4	2D Numerical Examples of multi-patch analysis	100
8.4.1	Plate with a circular hole	100
8.4.2	L-shaped wedge	103
8.4.3	Edge crack using Multi-patch	104
8.4.4	L-shaped Bracket	107
8.5	3D Numerical Examples of multi-patch analysis	107
8.5.1	3D edge crack with Mode I and Mode II loading condition	108
8.5.2	Penny crack	109
8.5.3	3D Connecting rod	110
9	Summary	128
9.1	Future work	129
	References	131

Nomenclature

Latin Symbols

J	Jacobian Matrix
B	Gradient Matrix
b_i^k	PHT-spline basis function at level k
D	Matrix of Material
I	Identity tensor
K	Global stiffness Matrix
P	Control points
R	Rational basis function
N, M	Basis function in parametric
E	Young's modulus
w	Weights
S	Surface
V	Volume
C	Bézier extraction operator
t	Traction
\mathbf{n}	Normal
\mathbf{b}	Body force
p, q, r	Order of the basis functions

Greek Symbols

\mathcal{S}	Dimensional space
ν	Poisson's ratio
\mathcal{L}	Linear operator
ξ, η, ζ	Parametric coordinates
Ξ	Knot vector in ξ direction
σ^h	Numerical stress
σ^*	Recovered stress
σ	Exact stress
α, β, γ	order of the continuity
Ω	Domain

Γ Boundary

Subscripts

h Associated with approximation solution
 i, j, k Associated with node index number

Superscripts

p Power of integrable

Abbreviations and Acronyms

<i>CAD</i>	Computer Aided Design
<i>CAE</i>	Computer Aided Engineering
<i>FEA</i>	Finite Element Analysis
<i>FEM</i>	Finite Element Method
<i>IGA</i>	Isogeometric Analysis
<i>BEM</i>	Boundary Element Method
<i>IGFEA</i>	Isogeometric Finite Element Analysis
<i>GIFT</i>	Geometry Independent Field approximaTion
<i>NURBS</i>	Non-Uniform Rational B-splines
<i>PHT – splines</i>	Polynomial splines over hierarchical T-meshes
<i>THB – splines</i>	Truncated Hierarchical B-splines
<i>LRB – splines</i>	Locally Refined B-splines
<i>SPR</i>	Superconvergent Patch Recovery
<i>DOF</i>	Degree of freedoms
<i>TP</i>	Tensor-product
<i>c – edge</i>	Composite edge
<i>p – edge</i>	Plus edge
<i>p – vertex</i>	Plus vertex
<i>GP</i>	Gauss points

List of Figures

1.1	Comparison in between isoparametric and isogeometric .	3
1.2	The source of errors in simulation process.	6
1.3	Diffpack application in different types of numerical simulation problem.	8
1.4	Diffpack Kernal with different features.	8
1.5	Diffpack main philosophy to solve PDEs or ODEs.	9
2.1	Knot vector in a parametric mesh	13
2.2	B-spline basis functions ($p = 2$) for knot vector $=\{0, 0, 0, 1, 2, 3, 4, 5, 5, 5\}$.	14
2.3	B-spline curve	16
2.4	The comparison of NURBS curve with different weights, $\omega_A = 1, 0.6, 0.3$	18
2.5	NURBS refinement	20
2.6	NURBS plane with multiple patches	20
2.7	Plot of cubic NURBS curve and basis functions with knot vector $\{0,0,0,0,1,2,3,4,5,5,5,5\}$.	21
3.1	Local basis and coordinate system of T-spline	26
3.2	Flow diagram of isogeometric finite element implementation using NURBS in Diffpack [67].	29
3.3	Geometry and material specification of a thin cylinder subjected to internal pressure.	31
3.4	Cylinder subjected to internal pressure meshes at different step where uniform mesh refinement performed.	32
3.5	Contour plots of displacement and stress components (x,y) of the cylinder subjected to internal pressure.	33
3.6	Error norms for pressurized cylinder quadratic IGA NURBS and FEM comparison.	34
3.7	Plate circular hole subjected to infinite tension meshes at different step where uniform mesh refinement performed.	35
3.8	Contour plots of displacement and stress components (x,y) of the plate circular hole subjected to infinite tension.	36
3.9	H^1 error norms for plate circular hole quadratic IGA NURBS and FEM analysis comparison.	37
4.1	Main idea of isogeometric analysis	39
4.2	Main idea of GIFT	40
4.3	Geometry with the boundary condition of Timoshenko beam.	46
4.4	Error plot L^2 displacement norm of Timoshenko beam.	46
4.5	Error norms of Timoshenko beam higher-order PHT-splines.	47
4.6	Plots of $(u - u_h)'(x)$ for $u = x^{p+1}$ and uniform meshes with knot-span length $1/10$. The red dots correspond to the coordinates of the superconvergent points, i.e. the roots of ψ' computed on the reference interval and whose coordinates are scaled and translated to each knotspan in the discretization: a) $p = 3$, b) $p = 4$, c) $p = 5$	51

4.7	A recovery subdomain on an annular shaped domain together with the superconvergent points for $p = 3$: a) Parameter space b) Physical space	53
5.1	Mapping 1D B-splines basis function.	56
5.2	B-spline basis function.	57
5.3	Bernstein basis function for a 1D elements $[\mathbf{u}] = [-1, 1]$ for degree $p = 3$	58
5.4	PHT-spline basis function for 1D two elements after knot insertion at knot $x^* = 0.75$	58
5.5	B-spline basis function for $p = 5$ and C^2	59
5.6	Initial PHT-spline basis function for two 1D elements for degree $p = 5$ using Bernstein polynomial and Bézier extraction operator.	59
5.7	PHT-spline basis function refinement in the second 1D element at $x^* = 0.75$ for $p = 5$ and C^2	60
5.8	Estimated recovery based error in different refinement steps for degree $p = 3$. Here red lines are $E_{rel} * \sqrt{H^1}$ (where E_{rel} =target relative error and H^1 = energy norm global).	61
5.9	Estimated recovery based error in different refinement steps for degree, $p = 7$. Here red lines are $E_{rel} * \sqrt{H^1}$ (where E_{rel} =target relative error and H^1 = energy norm global).	62
5.10	L^2 -error and H^1 -error norms for 1D PHT-splines for different degree.	63
5.11	Estimated error Vs. Exact error H^1 -seminorm convergence plot.	63
6.1	An illustration of boundary, crossing and T-junctional vertices: b_j are boundary vertices, v_i^+ are crossing vertices and v_i^T are T-junctional vertices.	65
6.2	Quadtree example illustrating the hierarchical data organization of part of an adaptive mesh. The neighboring relations within each hierarchical level are established by pointers, which are shown here for one element of the finest level (in red color).	65
6.3	Example showing boundary, crossing and T-junction vertices on: a) the initial mesh, b) after the first refinement, c) after the second refinement. The black dots denote boundary vertices, the red dots represent crossing vertices and the green triangles are T-junctions.	66
6.4	The 9 regions into which each child element is subdivided. The axis labels show the number of Bézier coefficients in each direction.	68
6.5	Modification of a quartic ($p=4$) basis function: a) before truncation b) after truncation	69
6.6	Modification of the Bézier ordinates corresponding to a quartic basis function: a) initial Bézier ordinates on the unrefined element, b) the Bézier ordinates obtained by applying Algorithm 1 to each row and then to each resulting column of the ordinates set in a). The shaded areas corresponding to the new basis vertices indicate the Bézier ordinates that should be set to zero to obtain the truncated basis function.	69
6.7	Global indices corresponding to the basis functions on each element (IEN arrays) for $p = 4, \alpha = 1$ (quartic $C^{1,1}$ polynomials). a) A 4 element mesh before refinement. b) The resulting mesh and node indices after refining one element. The colored node indices are the new basis functions. The purple nodes are deleted from the original mesh and reassigned to the refine mesh; the red node indices are new basis functions and indices.	71
6.8	9 vertices where blue b_i is boundary vertices, red v_i is crossing vertices and e_i are elements.	74
6.9	Inserting a cross in a parent element at the left figure, $e=4$ creates child element $e=\{5,6,7,8\}$ at the figure right.	75
6.10	B-spline and Bernstein basis representation for degree $p = q = 5$ with C^2 continuity.	75

6.11	2D B-spline representation using Bézier extraction and PHT spline basis function representation on refined mesh with degree $p = q = 5$, C^2 continuity.	76
6.12	Thick pressurized cylinder.	77
6.13	The refined meshes for different polynomial degrees p in the thick cylinder example.	78
6.14	Contour plots of displacement and stress components of the thick cylinder.	79
6.15	Relative error (exact and estimated) in energy H^1 norm vs. the number of degree of freedom for $p = 3, 4$ and 5 , using adaptive and uniform refinements for thick cylinder example.	80
7.1	Example of 3D T-mesh.	82
7.2	(a) 3D Hierarchical T-Mesh at level \mathcal{S}_0 , (b) 3D Hierarchical T-Mesh at level \mathcal{S}_1 , (c) 3D Hierarchical T-Mesh at level \mathcal{S}_2	83
7.3	Geometry and material specification of a hollow sphere subjected to internal pressure.	85
7.4	The refined meshes for different polynomial degrees p in the hemisphere example.	86
7.5	Contour plots of displacement and stress components (x,y,z) of the Hollow sphere subjected to internal pressure.	88
7.6	Relative error (exact and estimated) in energy H^1 norm vs. the number of degree of freedom for $p = 3, 4, 5$, using adaptive and uniform refinements for Hollow sphere example.	89
7.7	Geometry and material specification of cube with a spherical hole in a infinite domain subjected to uniform tension.	89
7.8	The z -direction stress and the refined meshes corresponding to the different polynomial degrees p for the cube with a hole example.	90
7.9	Relative error (exact and estimated) in energy norm vs. the number of degree of freedom for $p = 3, 4$ and 5 , using adaptive and uniform refinements for the cube with a hole example.	91
7.10	Geometry of elastic horseshoe subjected to equal and opposite in-plane flat edge displacements on the soles of the shoe.	91
7.11	The refined meshes corresponding to the different polynomial degrees p for the horse shoe example.	92
7.12	The von-mises stress and the displacement contour plot for $p = 3$ for the horse shoe example.	93
7.13	Relative error (exact and estimated) in energy norm vs. the number of degree of freedom for $p = 3, 4$ and 5 , using adaptive and uniform refinements for the horse shoe example.	93
8.1	Example of patch boundary configuration. Here in the box the numbering for edges are: 1-bottom edge, 2-right edge, 3-top edge, 4-left edge. Patch numbering is shown without box.	95
8.2	Example of patch boundary configuration in the coarsest mesh where every patch consists of 2×2 elements. Here the numbering for edges are: [2, 3]-bottom edge, [3, 5]-right edge, [4, 5]-top edge, [2, 4]-left edge.	96
8.3	Example of non conforming patches.	97
8.4	Making conforming patches by a cross insertion at patch B	97
8.5	Global node indices numbering in between patches.	97
8.6	Patch boundary configuration of 3D PHT-mesh. Here the numbering for faces are: 1-front face, 2-left face, 3-back face, 4-right, 5-down face, 6-up face.	98
8.7	(a) Patch A and B are non conforming patches, (b) Making conforming patches by a cross insertion at patch B	98
8.8	Problem description for the plate with a circular hole example	101

8.9	The refined meshes for different polynomial degrees p .	101
8.10	Relative error (exact and estimated) in the energy norm vs. the number of degree of freedom for $p = 3, 4$ and 5 , using adaptive and uniform refinements for the plate with a hole example.	102
8.11	Geometry, boundary conditions and material properties for the L-shaped wedge problem.	103
8.12	The refined meshes for different polynomial degrees p in the L-Shaped wedge example.	104
8.13	Relative error (exact and estimated) in energy norm vs. the number of degree of freedom for $p = 3, 4$ and 5 , using adaptive and uniform refinements for the L-Shaped wedge example.	105
8.14	Relative error (exact) in energy norm vs. computation time for $p = 3, 4$ and 5 , using adaptive GIFT and uniform refinements using PHT-spline for the L-shaped wedge example.	106
8.15	Geometry, boundary conditions and material properties for Edge crack multipatch problem.	106
8.32	Geometry and material properties of multi-patch penny shaped crack domain.	110
8.16	Meshes at different hierarchical level from initial level 0 with the NURBS geometry and the solution field is defined by PHT-spline with adaptive local refinement for Edge crack multi-patch domain.	112
8.17	Contour plots of displacement and stress components of the Edge crack with Multi-Patch.	113
8.18	Relative error (exact and estimated) in energy norm vs. the number of degree of freedom for $p = 3, 4$ and 5 , using adaptive GIFT PHT-spline and uniform refinements for the Edge crack multi-patch analysis.	114
8.19	Geometry, boundary conditions and material properties for L-shaped bracket.	115
8.20	The refined meshes for different polynomial degrees p in the L-Shaped bracket example.	116
8.21	Contour plots of displacement and stress components of the L-shaped bracket with Multi-Patch.	117
8.22	Relative error (exact and estimated) in energy norm vs. the number of degree of freedom for $p = 3, 4$ and 5 , using adaptive GIFT PHT-spline and uniform refinements for the L-shaped bracket multi-patch analysis.	118
8.23	FEM solution of Von mises displacement of L-shaped Bracket using Matlab PDE toolbox.	118
8.24	(a)Edge crack with Mode I loading condition. (b) 4 patches to define the edge crack: patches 1-2,2-3, 3-4 connected and 4-1 is not connected.	119
8.25	(a) Edge crack with Mode II loading condition. (b) 4 patches to define the edge crack: patches 1-2,2-3, 3-4 connected and 4-1 is not connected.	119
8.26	Meshes at different hierarchical level from initial level 0 with the NURBS geometry and at the solution field PHT-spline used with adaptive local refinement for Edge crack Mode I loading condition in multi-patch domain.	120
8.27	Contour plots of displacement and stress magnitude of the Edge Crack Mode I loading condition with Multi-Patch.	121
8.28	Relative error (exact and estimated) in energy norm vs. the number of degree of freedom for $p = 3, 4$ and 5 , using adaptive and uniform refinements for the 3D Edge crack mode I loading condition example.	121
8.29	Meshes at different hierarchical level from initial level 0 with the NURBS geometry and at the solution field PHT-spline used with adaptive local refinement for Edge crack Mode II loading condition in multi-patch domain.	122

8.30	Contour plots of displacement and stress magnitude of the Edge Crack Mode II loading condition with Multi-Patch.	123
8.31	Relative error (exact and estimated) in energy norm vs. the number of degree of freedom for $p = 3, 4$ and 5 , using adaptive and uniform refinements for the 3D Edge crack mode II loading condition example.	123
8.33	The von Mises stresses and the refined meshes corresponding to the different polynomial degrees p for the penny crack example.	124
8.34	Relative error (exact and estimated) in energy norm vs. the number of degree of freedom for $p = 3, 4$ and 5 , using adaptive and uniform refinements for the penny crack example.	125
8.35	Geometry of the connecting rod example.	126
8.36	The von Mises stresses and the refined meshes corresponding to the different polynomial degrees p for the connecting rod example.	126
8.37	Relative error (exact and estimated) in energy norm vs. the number of degree of freedom for $p = 3, 4$ and 5 , using adaptive GIFT PHT-spline and uniform refinements for the connecting rod example.	127

List of Tables

4.1	Local refinement scheme in GIFT using PHT-splines.	48
4.2	The superconvergent points for splines of degree p with continuity C^α on interval $[-1,1]$	50
4.3	Values of p^* and α^* for given p and α with the size of the resulting matrix $A^{(k)}$. The last column shows the location of the superconvergent points from Table 4.2 in one dimension ($d = 1$).	53
8.1	Example of patch boundary connection configuration. Edge numbering: 1-bottom edge, 2-right edge, 3-top edge, 4-left edge.	95

Chapter 1

Introduction

Classical finite element analysis is a commonly used method in many fields of numerical simulation and engineering applications. It has a vast field of application from solid mechanics to fluid mechanics, wave propagation, structural mechanics and dynamics and material sciences. But there is always a gap in between Computer Aided Engineering (CAE) and Computer Aided Design (CAD) because of discretization of complex geometries require mesh generation for subsequent numerical analysis. This extra step, which leads to an imperfect approximation of geometry, is rather costly, particularly in terms of human intervention. Moreover, due to this mesh generation process, the geometric properties from CAD are often altered, and some details may be discarded in order to simplify the numerical analysis using FEM.

Several numerical methods have been introduced over the past couple of years to incorporate the geometric information in the approximation of the solution field for solving partial differential equations (PDEs). Among the proposed numerical methods noteworthy are: subdivision surfaces [41], implicit surfaces and non-fitted meshes [92, 96]. In all simulation approaches, the geometric information is always transferred to the numerical approximation to discretize the field variables. Due to this transformation, the geometric exactness is sometimes lost and the simulation process is started with the approximated geometry which is prone to the beginning errors in the solution of the numerical problems. A breakthrough was the method of **Isogeometric analysis**(IGA) by [70] which enables a more seamless integration from CAD to analysis (CAE).

Initially, the key idea of isogeometric analysis was based on Non-Uniform Rational B-splines (NURBS) [70], and then the idea was extended by other spline types which are T-splines [24], locally refined (LR) B-splines [74], truncated hierarchical B-splines (THB-splines) [59, 60] and polynomial/rational splines over hierarchical T-meshes (PHT/RHT-splines) [99, 6, 128]. T-splines are commonly used in CAD to create unstructured surface representations, however, the linear dependency is not in general guaranteed and may be particularly difficult to enforce for 3D domains for which T-splines are not standard CAD modeling tools. Nevertheless, we mention recent advances regarding analysis-suitable T-splines [112], and truncated T-splines [129]. The THB-splines and some other related methods have been the subject of a significant amount of recent research including studies of the mathematical properties and CAD applications [65, 35, 78] in particular due to the improved sparsity and conditioning which results from the truncation procedure while maintaining the inherent efficiency of hierarchical spaces. For a detailed comparison between the LR B-splines and (T)HB-splines we refer to [75]. PHT and RHT splines belong to a class of Hermite finite elements over T-meshes where the basis is given in terms of Bernstein polynomials as discussed in [79, 109].

The key idea of IGA is to use the CAD geometry directly in the analysis by using the same

spline basis functions without converting the geometry through a preprocessor routine into a different form. The main advantage is that the meshing procedure can be eliminated because an existing CAD geometry is used directly in the analysis by keeping the exactness of the geometry.

1.1 Brief history of the important topics covered in this thesis

In this section the brief history of the important topics which have been taken into account to cover this thesis is discussed.

1.1.1 Isogeometric analysis

Design and analysis always interrelated to each other. Each has its own geometric representation and description, embodied in CAD system which needs to be translated as an analysis-suitable geometry for the simulation process in a FEA code. This task is not always trivial as most of the time in a complex design is consumed by mesh generation, and engineering design is becoming increasingly involved. Complex engineering objects like automobiles, fighter jets and airplanes consist of hundreds of thousands of parts. It can take almost 80 percent of the analysis time to convert the design to an analysis suitable. Engineering design and analysis in a sophisticated system are based on a wide range of computational analysis and simulation methods.

To break down the discrepancies between the design and analysis process, isogeometric analysis stepped forward to focus on one and only geometric model which can be directly utilized as an analysis model. The requirement of changing from classical finite element analysis to an analysis procedure based on exact CAD representation has been fulfilled by isogeometric analysis proposed by [70] and the method has been further developed in other several papers.

Although the term *isoparametric* is well established, which is based on the idea that one should use the same basis for the geometry and the unknown field approximation, the term *isogeometric analysis* is not restricted only to one type of basis functions. It just indicates the geometric description which can be used for FEA will be the same as was used in CAD design before. In analogy to the well-known term **isoparametric** in FEA, the name **isogeometric** is chosen. The comparison between **isogeometric analysis** and **isoparametric** is given in Fig. 1.1 side by side.

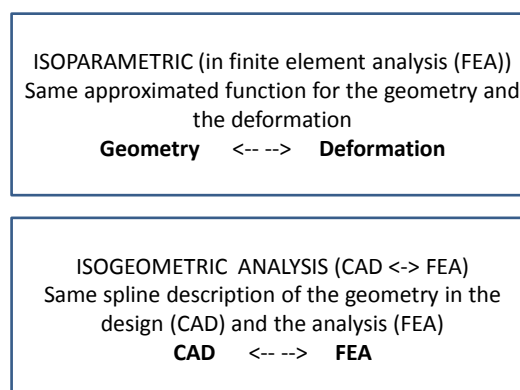


Fig. 1.1: Comparison in between **isoparametric** and **isogeometric**.

Isogeometric analysis has so far been successfully used in variety fields of application such as:

- Structural optimization [127, 115].
- Applications such as Fluid-Structure interaction and fluid flows [8, 23].
- Structural elements like plate and shell problems [29, 30, 123].
- Electromagnetism [38].
- Biomechanics [25, 131].

It has been quickly discovered that using smooth spline functions for design and analysis offered a simplified way for analysis and it is an emerging technology capable of:

- Directly interacting with the CAD system.
- Greatly simplifying the refinement processes.
- smooth basis functions with compact support.
- Improve the solution accuracy.
- Reducing the computational cost.
- Integrating the design and analysis.

It has been shown to be a more accurate per degree of freedom (dof), and it allows for the discretization of high-order differential equations, in addition to providing stable solutions when using higher-order polynomial approximations [25]. Initially, isogeometric analysis was considered equivalent to B-splines and NURBS as a basis in the FEM, but in later years many extensions have been proposed. It has been applied to:

- Collocation method [14, 10].
- Boundary element method [116, 111].
- Multigrid methods [58].
- Finite volume methods [64].
- Immersed Boundary method and finite cell method [107, 103].
- Phase-field modeling [106, 61].

1.1.2 Geometry Independent Field approximation (GIFT)

Isogeometric analysis enables the analysis of some CAD models in FEA without mesh generation and it has removed somehow the burden of preprocessing. But still, in the standard form, it is constrained to use the same spline space in the geometry as well as in the solution which might be in some cases unnecessary if the same exact geometry can be described in a simpler way on a different basis. The strict dependence of the use of the same basis functions for the geometry representation and field approximation in some particular cases is more hindrance than an advantage. For example, it can be possible that the spline space is not well-suited to approximate the solution field of Partial Differential Equations (PDEs), in particular when local mesh refinement is required to capture the solution of particular interest with a minimal computational cost. To alleviate this

difficulty **Geometry Independent Field approximaTion (GIFT)** has further developed from the idea originally proposed by [90] based on the principle of *generalized IGA* which was discussed at the 2013 CISM course on *Iso-Geometric Methods for Numerical Simulation* by Prof. Gernot Beer [27].

So far it has been applied in some specific methods, among which are:

- In Boundary Element Method (BEM), boundary displacements and tractions have different continuity properties, which cannot be accurately approximated by the same basis in IGA, but with weakening this dependency by using GIFT, independently h- and p- refinements can be performed of the displacement and traction fields while keeping the exact NURBS geometry (presented by [28, 12]).
- GIFT presented by [130] applied on PHT-spline descriptions in the solution field and NURBS for the geometric field for a two-dimensional heat conduction problem.
- GIFT implemented by [68] in a flexible object-oriented Diffpack program with adaptive PHT-spline for two-dimensional linear elasticity problem.

1.1.3 Error estimation

Local refinement and error estimation both are very important topics for adaptive local refinement in IGA and GIFT. Simulation based engineering is more concerned with solving physical problems of interest to engineers as the computational power is becoming cheaper and more prevalent. Several questions may arise for this topic:

- How can one model the physical problem?
- What are the equations describing the phenomenon which leads to a mathematical model?
- How can these equations be worked with the computational model (e.g. FEM, BEM, IGA, mesh-free methods)?
- Are these equations solved correctly and what is the error?

Related to local refinement, error estimation procedures are needed to drive adaptivity. We note there are two types of error estimators depending on whether the error being considered results from the construction of the *mathematical model* or from the *numerical model* (the discretized version of the mathematical model). This [117] suggests that knowledge of the error is essential to be able to correlate experimental and numerical results. One must ensure that the numerical results are close to the exact solution of the mathematical model, to guarantee that any discrepancy between the numerical and experimental results can be ascribed to the instability of the mathematical model (see Fig. 1.2).

Since the 1970s several methods had been developed to estimate the discretization error in finite element solutions. The first *posteriori* error estimator was introduced by [16, 20] and since then many others have been proposed. The existing techniques of the error estimators can be divided into two main categories:

- **Residual based a posteriori error estimator:** The main idea of **Residual based** error estimator based on to solve a physical problem by mathematical modeling suppose to find a solution of a stress field of a body which is in equilibrium. Suppose FEM is considered to solve this problem to find the error in each finite element and to see how far is the numerical stress from the equilibrium state. However, it does not require an enhanced solution in this error estimator. There are two types of residual based error estimator

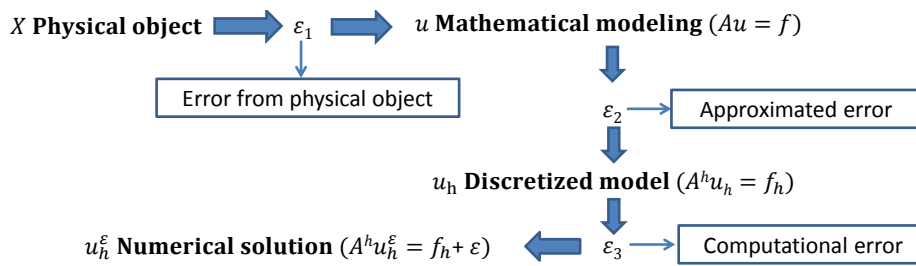


Fig. 1.2: The source of errors in simulation process.

1. **Explicit residual based error estimator:** **Explicit** residual based error estimator estimates [7] to measure accurately the boundary value problem to solve in each finite element. It involves the computation of the interior elements directly with the estimation of calculating the error in the energy norm of jumps at element boundaries [16, 20, 15, 76].
 2. **Implicit residual based error estimator:** **Implicit** residual based error estimator requires the solution of a local (in each element or over a group of elements) boundary value problem approximating the equation for the local error itself. The norm of the local error estimation is then used as a solution to the local boundary value problem. In this error estimator with the residual as data can be used to solve an auxiliary boundary value problem. This method has the advantage of alleviating the difficulty arise by the relative weighting of the residuals, but the disadvantage of this error estimation technique is to solve an additional boundary value problem which requires to choose an appropriate approximation method.
- **Recovery-based a posteriori error estimator:** The recovery-based error estimator which involves constructing an enhanced solution from a suitable, usually smoothing, post-processing of the numerical solution. The objective is to employ this improved solution as a substitute for the unknown exact solution. A remarkable and often surprising feature of the recovery-based error estimation is that, under some assumptions, by using just the raw numerical solution, an enhanced solution can be derived for use in error estimation. This technique was developed by Zienkiewicz and Zhu and reported in his famous papers [133, 134].

So far the application of a *posteriori* error estimation in isogeometric analysis is still not as mature as traditional FEM. There are several studies in IGA such as [52, 80, 125, 128, 121]. A multi-level error estimator using bubble functions was employed for T-splines in [52], as well as for reduced continuity order of T-meshes [36] and other adaptive hierarchical local refinement approaches in IGA [125]. A residual-based estimator was used for RHT splines in [128]. Recently, a functional type a posteriori error estimator for isogeometric analysis was introduced in [80]. The method provides guaranteed and sharp bounds for the error, but it requires the solution of a global problem.

In this thesis, we explore a different approach to construct a recovery-based on a posteriori error estimator proposed by [133, 134], with the aim of obtaining a recovered solution that is more accurate than the computed solution. Recovery-based error estimators are more commonly used in finite elements, as they are relatively simple to implement, computationally inexpensive and usually provide reliable error estimation. In [18, 19, 17], it was shown that the Superconvergent Patch Recovery (SPR) technique developed in [133, 134] results in robust error estimators, particularly when the exact solution is smooth. Moreover, it was shown in [132] that for any finite element residual error estimator an equivalent recovery estimator can be constructed. Recently, a detailed analysis of recovery-based estimators for LR B-splines has been conducted in [81].

In this thesis, we develop a higher polynomial degrees $p \geq 3$ and as well as increased continuity α , with $p \geq 2\alpha + 1$. While the continuity is less than the maximal C^{p-1} offered by standard B-Splines and NURBS, the reduced smoothness simplifies the construction, refinement and presents some advantages in terms of computational time for direct and iterative solvers [43, 42]. Moreover we develop a recovery-based error estimate for PHT-splines with IGA and GIFT discretization based on the superconvergence theory and verify their effectiveness for cubic (C^1 continuity) and arbitrary higher-order PHT-splines with adaptivity.

1.1.4 Local refinement in isogeometric analysis

Refinement is an important element in IGA. In the first paper of IGA proposed by [70], only NURBS was presented, which has certain shortcomings for local refinement due to the tensor product structure. These shortcomings can be overcome with the T-splines developed by [114, 113] as a generalization of NURBS. They overcome the geometric limitations using NURBS to generate local refinements in the mesh. Although T-splines was initially a CAD endeavor and a widespread technology in CAD, a few shortcomings have made them inconvenient to use in FEA, such as linear dependence [37] and refinement propagation [52]. A new sub-class of T-splines which is analysis-suitable (AS) T-splines [112, 85] emerged, which is a significant step towards more versatility in IGA.

There are other several local refinement strategies introduced before T-splines. The hierarchically refinable B-splines introduced by [57], which had been applied in isogeometric analysis in different applications [108, 107, 33, 125]. Later a generalization of Hierarchical B-splines presented by [60, 59] which is called Truncated Hierarchical B-splines recaptured several properties of NURBS, by which lack in the standard Hierarchical B-splines, for example, partition of unity, thereby having improved stability.

Another recently proposed spline called Locally Refined B-splines (LR B-splines) by [50] has the potential to do local h -refinement. This new approach operates directly on the spline spaces, and in this way, a broad spectrum of piecewise spline functions may be obtained. LR B-splines consist of smooth, piecewise polynomial basis functions that also constitute a partition of unity.

A generalized hierarchical approach of T-splines proposed by [48, 49] called Polynomial Splines over Hierarchical T-meshes has the potential of performing local refinement hierarchically. It has some other certain advantages like piecewise polynomial, hierarchical structure which can be adopted to adaptive local refinement, linear independence. Later it has been used in isogeometric analysis by [128, 99, 6]. We will also show in this thesis how it can be easier to utilize this PHT-splines with IGA and GIFT method.

1.1.5 Diffpack

Diffpack is an object-oriented development framework for the solution of partial differential equations. It is based on the latest developments in Object-Oriented Numerics and a programming environment for developing simulation software for scientific and engineering applications. Diffpack has its main focus on the numerical modeling and solution of partial differential equations, in particular by the finite element method and the finite difference method (finite volume method is also supported to some extent). The Diffpack software consists of a family of C++ libraries for general tasks related to the numerical solution of partial differential equations, plus a set of Perl and Python scripts that ease the development of simulation programs and problem-solving environments for

scientific or engineering research. The package was one of the very first to explore object-oriented programming and the C++ language for advanced, high-performance computing. For customers, this means unsurpassed modeling flexibility while satisfying the strictest demand for computational efficiency. There are more than 350 Diffpack users in more than 30 countries, including major industrial enterprises, consulting companies, software vendors, research institutions and universities in such diverse areas as oil and gas, mechanical engineering, telecommunication, medicine and finance.

Diffpack [1] has been actively developed since 1991, with main contributions from the University of Oslo and the research institutes SINTEF [4] and Simula Research Laboratory [3]. The initiators and main contributors to Diffpack in the 1990s were Hans Petter Langtangen (see in [82]) and Are Magnus Bruaset. Version 1.0 of the software was released in the public domain in 1995, with a new version in 1997. The Norwegian company Numerical Objects as took over the rights of Diffpack 1997 and commercialized the product. In 2003, the German company inuTech GmbH [2] purchased Diffpack and is now the principal maintainer and developer of the software. It has several applications in different fields solving the PDEs with various numerical methods like FEM, FVM and IGA which has shown in Fig. 1.3.

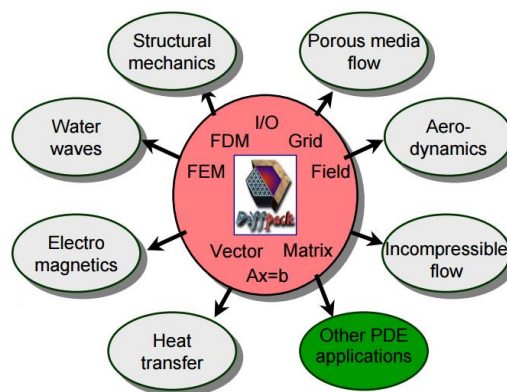


Fig. 1.3: Diffpack application in different types of numerical simulation problem.

The Diffpack Kernel (see in Fig.1.4) contains a substantial collection of data structures and algorithms such as e.g. vectors, matrices, strings, input-output (I/O), databases, reporting tools, GUIs, pre- and post-processing tools, linear and non-linear systems and solvers, finite element and finite difference algorithms, grids, fields, numerical integration, real and complex arithmetic, etc. Functionality including e.g., adaptive meshing, multigrid methods and parallel computing is provided in separate toolboxes.

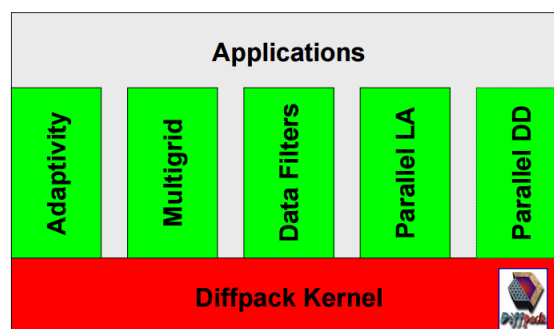


Fig. 1.4: Diffpack Kernel with different features.

The main advantage (see Fig. 1.5) of working with Diffpack is that it allows the implementation of any kind of Partial Differential Equations (PDEs) or Ordinary Differential Equations (ODEs) or other numerical problems using the Diffpack Kernel. Other toolboxes create an environment for the users to create their own flexible solvers. So Diffpack is a development environment, not an application that allows a user to generate a simulator or to use an already developed simulator. Diffpack tailored simulation provides the following benefits:

- Flexible building blocks to develop solvers for non-standard or specialized simulation problems.
- Complementary to standard FEM-programs (i.e. ANSYS, CFX, etc.) to fill the gaps in functionality.
- Building highly customized trimmed-down solvers with the minimum overhead of unused functionality and unnecessary complexity in the simulation process.
- Applicable to all simulation problems that can be modeled by Differential Equations, exact modeling of physical effects and control of the numerical simulation algorithm.

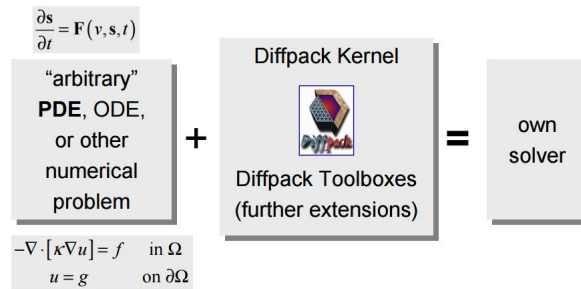


Fig. 1.5: Diffpack main philosophy to solve PDEs or ODEs.

1.2 Objectives

In isogeometric analysis as the same spline representation for both geometry and solution space is used, it creates certain constraints in specific simulation process e.g. where local refinement is a necessary part to obtain a good approximation of the numerical solution of PDEs. To alleviate this constraint, we can represent the solution field independent of the geometric spline space, which provides more flexibility in the simulation process by allowing the choice of appropriate splines in the solution field. As Geometry Independent Field approximaTION (GIFT) provides the user's flexibility in using a simpler mapping than the spline space without altering the initial CAD geometry, it becomes a new paradigm in the simulation process for designers and analysts. We know that most of the geometries are made of NURBS by CAD community and therefore, it is not directly possible to do local refinement at the solution space, as it does not have the local refinement property due to the tensor product structure. For this reason, our motivation of this thesis is to use PHT-splines at the solution space due to its local refinement property and to develop adaptive higher-order isogeometric methods using PHT-splines. For using adaptivity with PHT-splines, we need to use error estimators and recovery-based error estimation is the most robust error estimator in the class of a posteriori error estimation. Our objective is the usage of more flexible numerical tools in the simulation process which could be more beneficial in both the design and analysis process.

Most of the problems considered in the standard IGA have some very nice properties when the solution is smooth (without any discontinuities), but most of the real world simulation problems are

considered domains with certain discontinuities or not smooth solutions, such as re-entrant corners, cracks, etc, for which NURBS are not very efficient due to the singularities that appear. Adaptivity allows users to more efficiently resolve these singularities, by adding more degrees of freedom where they are needed. So our main concern is to focus on certain simulation problems where higher-order adaptive PHT-splines could be more efficient using the flexible GIFT method in particular for Linear Elasticity problems. For this reason, PHT-splines, the GIFT method, recovery-based error estimation, Diffpack with the flexible object oriented C++ environment etc, are just the tools that are used towards the fulfillment of this goal.

1.3 Challenges

While working with this thesis, some challenges came across in various implementation stages. Most of the challenges can be mentioned in the following order:

- The first challenge is to implement the NURBS in isogeometric analysis. It is always hard to utilize the geometry directly from CAD software to analysis and to use the CAD information in analysis is also a great concern in the numerical simulation process. We overcome this challenge by creating an interface and writing a python script which can read the NURBS geometric data and T-spline Bézier meshes easily from CAD software such as Rhino3D.
- Implementation of PHT-splines is a challenge because of its hierarchical data structures for 2-dimensions and especially for 3-dimensions.
- For recovery-based error estimation it is necessary to carefully choose the sampling points that are used for the particular type of PDE. We have chosen the super-convergent points appropriately to get a more accurate recovered solution than the original computed solution.
- Handling multi-patch geometries in CAD design to analysis makes numerical simulation a more challenging and tedious process. Our simple knot insertion algorithm can handle complicated CAD geometries having several patches efficiently.
- Doing numerical analysis with the geometries which have discontinuities is more difficult, in particular when the spline space consists of smooth functions especially in 3D. We show how to overcome this challenge by doing some numerical examples with certain geometric discontinuities (cracks, holes, re-entrant corners) that can handle multi-patch geometries.

1.4 Contributions

The contributions from this thesis can be summarized in the following points:

- With the method we used here, one can directly do analysis on the geometry created from CAD software (e.g. Rhino3d or the NURBS toolbox) as most of the CAD designs are NURBS or T-splines. The coarse NURBS mesh created by the CAD software can represent exactly the geometry while the solution space is approximated with PHT-splines. One can do adaptive local refinements, where it is necessary to focus the particular point of interest in the numerical simulation process.
- For adaptivity, we used recovery-based error estimator, by choosing superconvergent points which have more accuracy in approximating the gradient of the exact solutions, this implies that the error estimator is more robust and efficient.

- We used higher-order and continuous PHT-splines which are more efficient in terms of using less dofs to achieve greater convergence rate to reach a target relative error.
- We applied adaptivity for numerical examples which have smooth solutions and also the problems with certain discontinuities (re-entrant corners, cracks). For both type of numerical applications, we exhibited the advantage of using higher-order adaptive GIFT PHT-splines for both 2D and 3D cases. Also, we demonstrated which order and continuity of PHT-splines are more accurate in terms of utilizing more or fewer degree of freedoms (DOFs).
- Most of the CAD geometries are created with multiple patches. So we demonstrated a simple method how to handle more general multi-patch geometries containing C^0 continuity in between patches by using GIFT and higher-order PHT-splines.

1.5 Chapter Overview

This thesis will deal with different aspects of isogeometric analysis and an alternative method (GIFT) where different spline applications (NURBS, T-splines, PHT-splines) are presented. In this introduction, we gave a general overview of this work which shows the motivation of the various topics of this thesis, the contributions and the challenges as they are taken into account to fulfill the thesis objectives. Further detailed background information and the state of the art of each topic will be presented in the corresponding chapters. The thesis consists of 9 chapters including the concluding remarks and future works at the end. The organization of work is as follows:

- In chapter 2 we introduce in detail various splines like B-splines, NURBS and Bézier extraction method which are familiar in CAD design and useful for isogeometric analysis both for geometry and field approximation.
- In chapter 3 we present in detail isogeometric analysis and its data structure as it is implemented in Diffpack using NURBS and how to utilize the Bézier extraction operator to work on existing FEM code with minor modification using NURBS for analysis in the isogeometric toolbox in Diffpack for linear elasticity.
- In chapter 4 we present a generalization of the isogeometric analysis to a new discretization method named Geometry Independent Field approximation (GIFT) in detail where a general overview, the main principle with a mathematical framework of this method for linear elasticity problem. In this chapter, we also present how to use adaptivity with GIFT method together with the PHT-spline and the recovery-based error estimation techniques and the marking algorithm which has been used for adaptivity.
- In chapter 5 we introduce the one-dimensional polynomial spline over hierarchical T-meshes (PHT-spline) with cubic C^1 continuity and higher-order PHT-spline. We present here the refinement algorithm of the higher-order and continuity for 1D PHT-spline basis with a simple 1D numerical example.
- In chapter 6 we introduce two-dimensional PHT-spline in details with its dimensional formula and the structure of the basis function in different hierarchical level for cubic and higher-order PHT-spline. A 2D numerical example is also demonstrated showing the performance of adaptive GIFT PHT-spline basis with different cubic and higher-order polynomials.
- In chapter 7 we extend the 2D PHT-spline to 3D by showing the 3D hierarchical T-meshes with the dimensional formula. We also show the 3D PHT-spline structure for cubic and higher-order PHT-spline basis functions. We present some 3D numerical examples with adaptive GIFT PHT-splines using recovery-based error estimation.

- In chapter 8 we discuss the multipatch analysis of 2D and 3D higher-order adaptive GIFT PHT-spline providing some numerical examples that have smooth solutions with some discontinuity (cracks, re-entrant corners, etc.).
- In chapter 9 will summarize the conclusions of this work.

Chapter 2

B-splines, NURBS and Bézier extraction

In this chapter, we describe the two important splines for geometric modeling in CAD (Computer Aided Design) which are B-splines and NURBS. Due to the need for integrating CAD and CAE (Computer-Aided Engineering) analysis, these splines play important roles in the Isogeometric Analysis (IGA) and Geometric Independent Field approximation (GIFT). In this chapter, we will review the formulations of B-splines and NURBS. In addition, the Bézier extraction technique is also explained, which enables the implementation of IGA and GIFT in the existing finite element codes to incorporate with B-splines and NURBS easily.

2.1 B-splines

B-spline or basis spline is defined as a spline function that has minimal support with respect to a given degree and smoothness. B-spline curve can be created with a linear combination of control points and B-spline basis functions.

2.1.1 Knot vector

A knot vector is defined in the parametric space as a sequence of non-decreasing real numbers and it must always be in ascending order:

$$\mathbf{U} = \{u_1, u_2, \dots, u_{n+p+1}\}, \quad u_A \in \mathbb{R}, \quad (2.1)$$

Here A denotes the knot index, p the polynomial order, and n is the number of basis functions or control points. Each real number of u_A is called a knot. The number of knots in a valid knot vector is always $n + p + 1$. The half-open interval $[u_i, u_{i+1})$ is called a knot span (See Fig. 2.1).

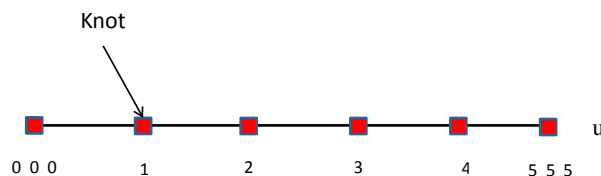


Fig. 2.1: Knot vector in a parametric mesh

2.1.2 B-spline basis functions

B-spline basis functions for a given degree, p can be defined recursively over the parametric domain by the knot vector. Piecewise constants are first defined by using the **Cox-de Boor recursion formula** [46, 31] as:

$$N_{A,0}(u) = \begin{cases} 1 & \text{if } u_A \leq u < u_{A+1} \\ 0 & \text{otherwise} \end{cases} \quad (2.2)$$

For $p > 0$, the basis functions are defined as:

$$N_{A,p}(u) = \frac{u - u_A}{u_{A+p} - u_A} N_{A,p-1}(u) + \frac{u_{A+p+1} - u}{u_{A+p+1} - u_{A+1}} N_{A+1,p-1}(u) \quad (2.3)$$

In essence, a B-spline basis function is a piecewise polynomial function. The functions are C^∞ within elements and C^{p-k} on element boundaries, where k is the number of knot repetitions. A B-spline curve of degree p in \mathbb{R}^d is defined by a set of B-spline basis functions $N(u) = \{N_{A,p}(u)\}_{A=1}^n$, and control points $\mathbf{P} = \{\mathbf{P}_A\}_{A=1}^n$ as:

$$T(u) = \sum_{A=1}^n \mathbf{P}_A N_{A,p}(u) = \mathbf{P}^T N(u) \quad (2.4)$$

B-spline basis functions possess the following properties (Fig. 2.2):

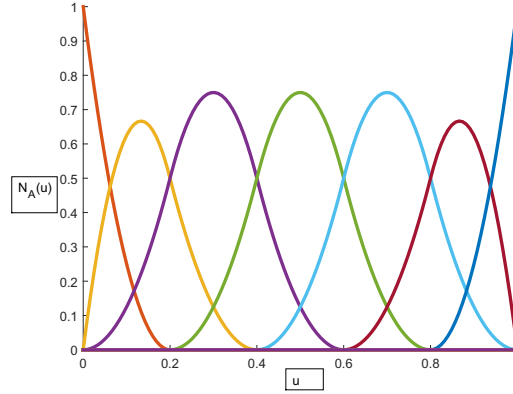


Fig. 2.2: B-spline basis functions ($p = 2$) for knot vector $=\{0, 0, 0, 1, 2, 3, 4, 5, 5, 5\}$.

- **Local support:** The B-spline basis function $N_{A,p}$ is always non-negative in the knot span of $[u_A, u_{A+p+1})$. This is significant for geometry manipulation: the change of one control point only affects the local part of the curve, providing flexibility in curve modifications.
- **Partition of unity.** $\sum_{A=1}^n N_{A,p}(u) = 1$.
- **Pointwise non-negativity.**
- **Linear independence.** This property is essential to construct the approximation space for numerical analysis.

The first-order derivative of the B-spline basis function is:

$$\frac{d}{du} N_{A,p}(u) = \frac{p}{u_{A+p} - u_A} N_{A,p-1}(u) - \frac{p}{u_{A+p+1} - u_{A+1}} N_{A+1,p-1}(u) \quad (2.5)$$

The k -th order derivatives of the B-spline basis function is given by:

$$\frac{d^k}{d^k u} N_{A,p}(u) = \frac{p}{u_{A+p} - u_A} \left(\frac{d^{k-1}}{d^{k-1} u} N_{A,p-1}(u) \right) - \frac{p}{u_{A+p+1} - u_{A+1}} \left(\frac{d^{k-1}}{d^{k-1} u} N_{A+1,p-1}(u) \right) \quad (2.6)$$

To construct B-spline surfaces and solids, the basis functions can be obtained from the tensor product. B-spline surface basis function is given by

$$N_A(\tilde{u}) = \sum_{i=1}^{d_A} N_A(u^i) \quad (2.7)$$

where i denotes the direction index, d_A is the number of dimensions. So the B-spline two-dimensional surface basis function can be defined as

$$N_{A,B}^{p,q}(u, v) = \sum_{A=1}^n \sum_{B=1}^m N_{A,p}(u) M_{B,q}(v) \quad (2.8)$$

and three-dimensional solid basis function can be defined as

$$N_{A,B,C}^{p,q,r}(u, v, w) = \sum_{A=1}^n \sum_{B=1}^m \sum_{C=1}^l N_{A,p}(u) M_{B,q}(v) L_{C,r}(w) \quad (2.9)$$

2.1.3 B-spline geometries

A B-spline geometry is a mapping from parametric space to physical space through a linear combination of B-spline basis functions, which are defined in parametric space. The corresponding coefficients are called control points because they can be represented by actual points scattered in physical space. Here we will discuss how to define the B-spline curve, surface and solid using the B-spline basis functions and control points.

2.1.3.1 B-spline curves, surfaces and solids

A B-spline curve can be expressed as

$$\mathbf{x}(u) = \sum_A^n N_{A,p}(u) \mathbf{P}_A, \quad (2.10)$$

where $\mathbf{x}(u)$ denotes the physical curve of interest, u is the spatial coordinate in parametric space, \mathbf{P}_A the control points, $N_{A,p}$ the B-spline basis functions of order p . See Fig. 2.3.

B-spline surfaces are constructed from a linear combination of bivariate B-spline basis functions $N_{A,B}^{p,q}(u, v)$ and control points $\mathbf{P}_{A,B} \in \mathbb{R}^{d_p}$, $d_p = 2$ as

$$S(u, v) = \sum_{A=1}^n \sum_{B=1}^m \mathbf{P}_{A,B} N_{A,B}^{p,q}(u, v), \quad (2.11)$$

where the bivariate B-spline basis functions are defined as Eq. 2.8.

B-spline volumes are constructed from basis functions $N_{A,B,C}^{p,q,r}(u, v, w)$ and control points $\mathbf{P}_{A,B,C} \in \mathbb{R}^{d_p}$, $d_p = 3$ as

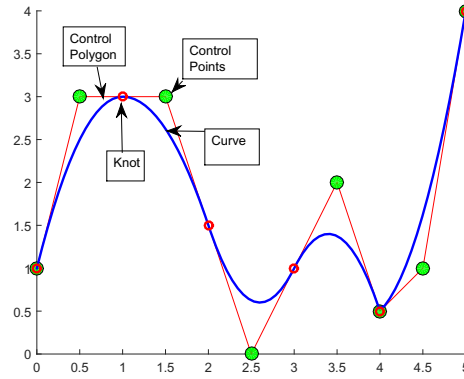


Fig. 2.3: B-spline curve

$$V(u, v, w) = \sum_{A=1}^n \sum_{B=1}^m \sum_{C=1}^l \mathbf{P}_{A,B,C} N_{A,B,C}^{p,q,r}(u, v, w), \quad (2.12)$$

where the trivariate B-spline basis functions are defined as Eq. 2.9.

2.1.3.2 B-spline geometrical properties

The continuity and differentiability of a B-spline curve are inherited directly from its basis functions and the continuity of a B-spline curve is at least C^{p-k} . Within the knot vector, knots can be repeated to make changes in the continuity. For example in the knot vector $U = \{0, 0, 0, 1, 2, 3, 4, 5, 5, 5\}$ the order is $p = 2$ and continuity is $C^{p-k} = C^{2-1} = C^1$, where k is the number of knot repetition numbers for an inner knots in this knot vector U . The knots with different values can be viewed as different break points which divide the one-dimensional parameter space into different elements. Hence, the physical interpretation of the knots can be explained as the parametric coordinates of the element edges, while the “knot span” between two knots with different values can be defined as elements in the parametric space u . The insertion of a new knot will split an element similar to h -refinement in the Finite Element Method. However, the repetition of existing knots will not increase the number of elements, but to decrease the continuity of the B-splines basis functions. For example, the knot vector $U = \{0, 0, 0, 1, 2, 3, 4, 5, 5, 5\}$ has 10 knot values with 9 knot spans, $[0, 0)$, $[0, 0)$, $[0, 1)$, $[1, 2)$, $[2, 3)$, $[3, 4)$, $[4, 5)$, $[5, 5)$, $[5, 5)$ but it has only 5 elements, $[0, 1)$, $[1, 2)$, $[2, 3)$, $[3, 4)$, $[4, 5)$.

An open knot vector can be defined if its first and last knot values are repeated $p + 1$ times, such as for the knot vector $U = \{0, 0, 0, 1, 2, 3, 4, 5, 5, 5\}$ with $p = 2$. The open knot vector is the standard in CAD, so all the examples in this thesis work will be presented with open knot vectors. The knot vector values can be normalized without affecting the resulting B-splines basis functions. Therefore the knot vector $U = \{0, 0, 0, 1, 2, 3, 4, 5, 5, 5\}$ is equivalent to the knot vector $U = \{0, 0, 0, 1/5, 2/5, 3/5, 4/5, 1, 1, 1\}$.

It is called a uniform knot vector if the intermittent knots are uniformly spaced, for example, $\{0, 0, 0, 1, 2, 3, 4, 5, 5, 5\}$ in Fig. 2.1.

B-spline geometries contain the following properties:

- **The convex hull property:** A B-spline curve is contained in the convex hull of its control polygon.

More specifically, if u is in knot span $[u_i, u_{i+1})$, then $C(u)$ is in the convex hull of control points $P_{i-p}, P_{i-p+1}, \dots, P_i$.

- **The variation diminishing property:** No plane has more intersections with the curve than it has with the control grid. This property renders B-splines less oscillatory than Lagrangian polynomials.
- **The transformation invariance property:** An affine transformation of a B-splines curve can be achieved by applying an affine transformation to the control points.
- **Non-interpolatory behavior of control points:** The B-spline geometry does not interpolate the control points except at the starting point of the curve, the endpoint of the curve and any point whose knot value is repeated p times.

2.1.4 Knot insertion in B-splines

Knots may be inserted into a knot vector without changing the geometric or parametric properties of the curve. Let $U = \{u_1, u_2, \dots, u_{n+p+1}\}$ be a given knot vector. Inserting a new knot $\bar{u} \in [u_A, u_{A+1}]$ with $A > p$ into the knot vector requires that $n + 1$ new basis functions be defined using Eq. 2.2 and 2.3 with the new knot vector $U = \{u_1, u_2, \dots, u_A, \bar{u}, u_{A+1}, u_{n+p+1}\}$. The $m = n + 1$ new control points, $\{\bar{\mathbf{P}}_A\}_{A=1}^m$, are formed from the original control points, $\{\mathbf{P}_A\}_{A=1}^n$ by

$$\bar{\mathbf{P}}_A = \begin{cases} \mathbf{P}_1 & A = 1 \\ \alpha_A \mathbf{P}_A + (1 - \alpha_A) \mathbf{P}_{A-1} & 1 < A < m \\ \mathbf{P}_n & A = m \end{cases} \quad (2.13)$$

$$\alpha_A = \begin{cases} 1 & 1 \leq A \leq k - p \\ \frac{\bar{u}_A - u_A}{u_{A+p} - u_A} & k - p + 1 \leq A \leq k \\ 0 & A \geq k + 1 \end{cases} \quad (2.14)$$

Knot values may be inserted multiple times, but the continuity of the basis is decreased by one for each repetition of a given knot value. However, by choosing control variables as in Eq. 2.13 and 2.14 the continuity of the curve is preserved.

2.2 NURBS

Non-uniform rational B-splines (NURBS) [102] are industry standard tools for the representation and design of geometry which are developed from B-splines. It can offer significant advantages due to their ability to represent a wide variety of geometric entities such as circular and conic sections. Moreover, the reasons for the use of NURBS are:

- Most of the CAD designs constructed by NURBS because of its flexibility.
- It can be evaluated reasonably fast by numerically stable and accurate algorithms.
- NURBS are invariant under affine mapping as well as perspective transformations.
- It is a generalization of non-rational B-splines and non-rational and rational Bézier curves and surfaces.

2.2.1 NURBS basis functions

NURBS can be defined by a knot vector $U = \{u_1, u_2, \dots, u_{n+p+1}\}$, a set of rational basis functions $\mathbf{R} = \{R_{A,p}\}_{A=1}^n$, and a set of control points $\mathbf{P} = \{\mathbf{P}_A\}_{A=1}^n$ as

$$\mathbf{x}(u) = \sum_A^n R_{A,p}(u) \mathbf{P}_A \quad (2.15)$$

Here, \mathbf{P}_A is the set of control point coordinates and $R_{A,p}$ are NURBS basis functions, defines as

$$R_{A,p}(u) = \frac{N_{A,p}(u)\omega_A}{W(u)} \quad (2.16)$$

with

$$W(u) = \sum_{A=1}^n \omega_A N_{A,p}(u), \quad (2.17)$$

where ω_A denotes a weight associated with each basis function or control point. It can influence the distance between the associated control point and the NURBS geometry, with higher values drawing the curve closer to that point (Fig. 2.4). When all of the weights are equal to 1, the NURBS curve becomes to a B-spline curve.

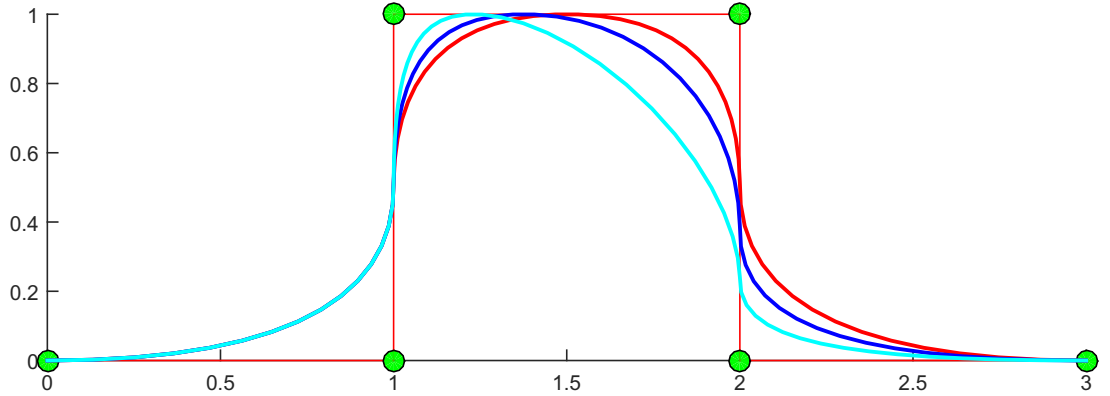


Fig. 2.4: The comparison of NURBS curve with different weights, $\omega_A = 1, 0.6, 0.3$

The derivatives of a NURBS basis function are expressed by

$$\frac{d}{du} R_{A,p}(u) = \omega_A \frac{W(u) \frac{d}{du} N_{A,p}(u) - \frac{d}{du} W(u) N_{A,p}(u)}{(W(u))^2} \quad (2.18)$$

and

$$\frac{d}{du} W(u) = \sum_{A=1}^n \frac{d}{du} N_{A,p}(u) \omega_A \quad (2.19)$$

NURBS geometries can be interpreted by a linear combination of standard B-spline basis functions and weighted control points

$$\mathbf{x}(u) = \sum_{A=1}^n N_{A,p}(u) \tilde{\mathbf{P}}_A, \quad (2.20)$$

where $\tilde{\mathbf{P}}_A = \omega_A \mathbf{P}_A$ and ω_A^T are the weighted control points in projective space. $N_{A,p}$ is the standard B-spline basis function.

A NURBS basis function in multi-dimensions can be obtained using a tensor product such as

$$R_A(\mathbf{u}|\mathbf{U}_A) = \prod_{i=1}^{d_p} N_A^i(u_A^i|U_A^i), \quad (2.21)$$

where i denotes the direction index and d_p is the dimension number. Therefore, NURBS basis functions in two-dimensions and three dimensions can be represented as follows:

$$R_{A,B}^{p,q}(u, v) = \frac{N_{A,p}(u)M_{B,q}(v)\omega_{A,B}}{\sum_{A=1}^n \sum_{B=1}^m N_{A,p}(u)M_{B,q}(v)\omega_{A,B}} \quad (2.22)$$

$$R_{A,B,C}^{p,q,r}(u, v, w) = \frac{N_{A,p}(u)M_{B,q}(v)L_{C,r}(w)\omega_{A,B,C}}{\sum_{A=1}^n \sum_{B=1}^m \sum_{C=1}^l N_{A,p}(u)M_{B,q}(v)L_{C,r}(w)\omega_{A,B,C}} \quad (2.23)$$

2.2.2 NURBS surfaces and volumes

Similar to the NURBS curve formulation, NURBS surfaces can also be represented as a linear combination of bivariate NURBS basis functions $R_{A,B}^{p,q}(u, v)$, control points $\mathbf{P}_{A,B} \in \mathbb{R}^{d_p}$, $d_p = 2$ and weights $\omega_{A,B} > 0$ as

$$S(u, v) = \sum_{A=1}^n \sum_{B=1}^m \mathbf{P}_{A,B} R_{A,B}^{p,q}(u, v), \quad (2.24)$$

where the bivariate NURBS basis functions are defined as Eq. 2.22.

NURBS volumes can be represented as trivariate basis functions $R_{A,B,C}^{p,q,r}(u, v, w)$, control points $\mathbf{P}_{A,B,C} \in \mathbb{R}^{d_p}$, $d_p = 3$ and weights $\omega_{A,B,C} > 0$ as

$$V(u, v, w) = \sum_{A=1}^n \sum_{B=1}^m \sum_{C=1}^l \mathbf{P}_{A,B,C} R_{A,B,C}^{p,q,r}(u, v, w), \quad (2.25)$$

where the trivariate NURBS basis functions are defined as Eq. 2.23.

2.2.3 Property of NURBS geometries

NURBS inherits the aforementioned properties of B-splines, but still have some drawbacks:

Rational functions NURBS are not polynomial functions and integration can not be done with Gauss quadrature.

Tensor product The parametric space and control points rely on a structured grid due to the tensor product property of NURBS and thus does not allow local refinement, which increases the redundancy of the degrees of freedom (Fig. 2.5).

Geometry repair From a computational geometry point of view, a NURBS based geometry always requires some level of repair due to gaps or overlaps of the various patches making up the

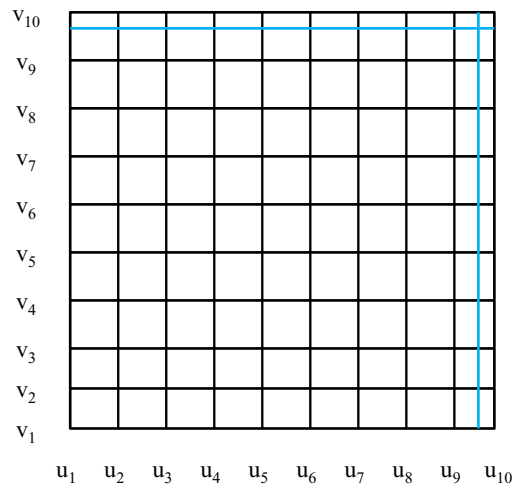


Fig. 2.5: NURBS refinement

geometry.

Continuity For Complex geometry, NURBS normally needs multiple patches. Each patch is associated with a parametric space (Fig. 2.6) and achieve only C^0 continuity between the patches.

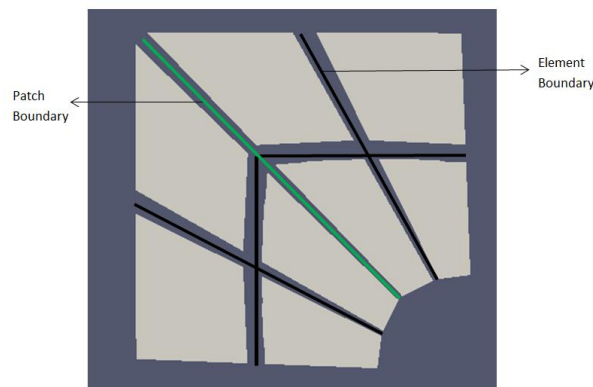


Fig. 2.6: NURBS plane with multiple patches

2.3 Bézier extraction

The idea to use Bézier extraction operator is to map a piecewise Bernstein polynomial basis onto a B-spline basis. This transformation makes it possible to use piecewise C^0 Bézier elements as the finite element representation of a NURBS, T-spline or PHT-spline. The use of Bézier extraction operator in IGA with existing FEM code was introduced, first for NURBS by [32] and then extended for T-splines in [110]. Bézier extraction provides an element data structure suitable for analysis. That is, similar to Lagrangian polynomial elements in traditional FEM, Bernstein basis does not change from element to element. In this section, we will show the representation of NURBS by Bézier elements and how it provides a mapping from a piecewise Bernstein polynomial basis onto a NURBS basis.

Later in 6 the use of Bézier extraction is shown for PHT-spline.

2.3.1 Bézier decomposition

To illustrate Bézier decomposition, we need to compute the Bézier elements of NURBS which can be accomplished by repeating all interior knots of a knot vector until they have a multiplicity equal to $p + 1$. For this purpose a multiplicity of p is sufficient. Bézier decomposition of a univariate B-spline curve is illustrated as follows.

We begin with the cubic B-spline curve shown in Fig. 2.7 and its associated knot vector as

$$U = \{0, 0, 0, 0, 1, 2, 3, 4, 5, 5, 5, 5\} \quad (2.26)$$

To decompose the curve into its Bézier elements we perform repeated knot insertion on all interior knots, beginning from the left, until they have a multiplicity equal to p , the degree of the curve. Thus, we will be performing knot refinement by inserting the knots $\{1, 1, 2, 2, 3, 3, 4, 4\}$ into the knot vector.

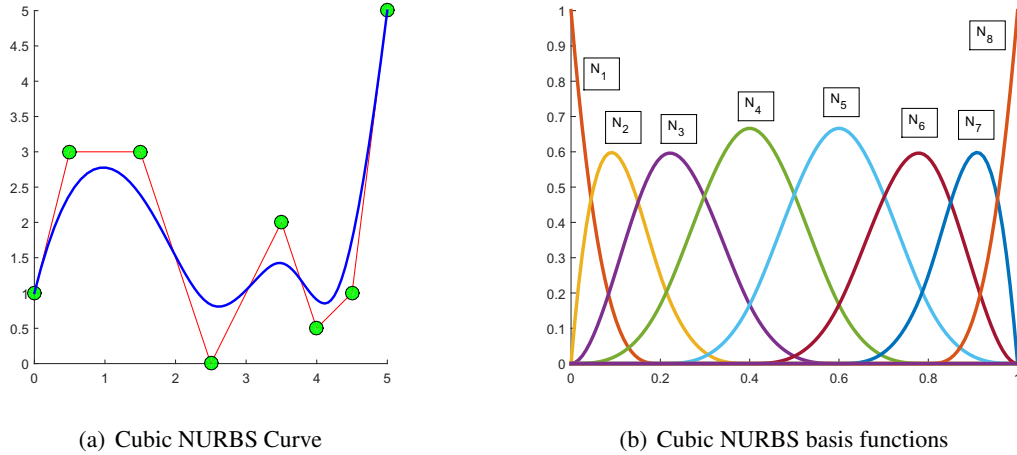


Fig. 2.7: Plot of cubic NURBS curve and basis functions with knot vector $\{0,0,0,0,1,2,3,4,5,5,5,5\}$.

Fig. 2.7 shows the sequence of basis functions and control variables created by inserting the new knots into the knot vector. Each inserted knot reduces the continuity of the basis by one at the knot location. The resulting basis has been decomposed into a set of C^0 Bézier elements with each element corresponding to a knot span in the original knot vector. The control points of the Bézier elements are computed by Eq. 2.13 and 2.14 each time a knot is inserted. Thus, the continuity of the curve is unchanged.

2.3.2 Computing the Bézier extraction operator

The Bézier extraction operator for a NURBS curve may be computed in order to represent it in terms of a set of Bézier elements. Assume that we are given a knot vector $U = \{u_1, u_2, \dots, u_{n+p+1}\}$ and a set of control points, $\mathbf{P} = \{\mathbf{P}_A\}_{A=1}^n$, that define a B-spline curve (possibly in projective space). Let be the set of knots that are required to produce the Bézier decomposition of the B-spline. Then for each new knot, we define to be the A_{th} alpha as defined in 2.14. Now, defining to be

$$\bar{\mathbf{P}} = (C^1)\mathbf{P}, \quad (2.27)$$

where C^1 is a matrix form representing the linear map defined in Eq. (2.12) while keeping the geometry the same. If the knots are replicated multiple times m , then

$$\bar{\mathbf{P}}^{m+1} = C^T \mathbf{P}, \quad (2.28)$$

where $C^T = (C^m)^T (C^{m-1})^T \dots (C^1)^T$. If $m = p$, a Bézier curve is obtained, and C is called the Bézier extraction operator. Hence, the geometry can be expressed by the Bézier basis function $\mathbf{B}(u)$

$$x(u) = (\bar{\mathbf{P}}^{m+1})^T \mathbf{B}(u) = (C^T \mathbf{P})^T \mathbf{B}(u) = \mathbf{P}^T \mathbf{C} \mathbf{B}(u) \quad (2.29)$$

After the rearrangement above, the B-spline basis functions can be represented by the linear combination of the Bézier basis $\mathbf{B}(u)$ and Bézier extraction operator \mathbf{C} ,

$$\mathbf{N}(u) = \mathbf{C} \mathbf{B}(u) \quad (2.30)$$

The Bézier basis $\mathbf{B}(u)$ is also called Bernstein basis, defined as

$$B_{i,p}^k(u^k) = \frac{1}{2^p} \binom{p}{i-1} (1-u^k)^{p-(i-1)} (1+u^k)^{i-1}, \quad (2.31)$$

where

$$\binom{p}{i-1} = \frac{p!}{(i-1)!(p+1-i)!}, \quad 1 \leq i \leq p+1 \quad (2.32)$$

And Bernstein basis satisfies the following properties

- Partition of unity

$$\sum_{i=1}^{p+1} B_{i,p}^k(u^k) = 1 \quad (2.33)$$

- Pointwise non-negativity

$$B_{i,p}^k(u^k) \geq 0 \quad (2.34)$$

- Endpoint interpolation

$$B_{1,p}^k(-1) = B_{p+1,p}^k(1) = 1 \quad (2.35)$$

- Symmetry

$$B_{i,p}^k(u^k) = B_{p+1-i,p}^k(-u^k) \quad (2.36)$$

- Linear independence

It is noteworthy that Bézier extraction operator is only determined by the knot vector, independent on the positions of control points.

From Eq. (2.30), it can be used to extend Bézier extraction from B-splines to NURBS. Writing Eq. (2.15) and Eq. (2.16) in matrix form as

$$\mathbf{R}(u) = \frac{1}{\mathbf{w}^T \mathbf{N}(u)} \mathbf{W} \mathbf{N}(u), \quad (2.37)$$

where \mathbf{W} is the diagonal matrix of weights, and \mathbf{N} is the matrix form of B-spline basis. Now we can write NURBS in terms of Bernstein basis as

$$x(u) = \mathbf{P}^T \mathbf{R}(u) = \frac{1}{\mathbf{w}^T \mathbf{N}(u)} \mathbf{P}^T \mathbf{W} \mathbf{N}(u) = \frac{1}{\mathbf{w}^T \mathbf{C} \mathbf{B}(u)} \mathbf{P}^T \mathbf{W} \mathbf{C} \mathbf{B}(u) = \frac{1}{\mathbf{w}^T \mathbf{C} \mathbf{B}(u)} (\mathbf{C}^T \mathbf{W} \mathbf{P})^T \mathbf{B}(u) \quad (2.38)$$

The above equation has represented NURBS with Bézier elements.

Chapter 3

Isogeometric Analysis toolbox using Diffpack

Diffpack [82] is an object-oriented problem-solving environment for the numerical solution of partial differential equations (PDEs). For a programmer, Diffpack eases the programming task using its several classes and methods which are written in C++ programming language. Diffpack demonstrates how programming using C++ classes is easier and more flexible than shuffling data in or out of the subroutine of traditional Fortran or C programming language. The C++ classes can be combined into application building blocks to solve problems in diverse fields, including Engineering, Natural Sciences, Economics and Medicine. Since using application codes extensively use of well-tested libraries and high-level abstractions, the time spent writing and debugging the code is significantly reduced.

In this chapter, we will discuss how isogeometric analysis is implemented using object-oriented flexible Diffpack FEM libraries. Some numerical examples will be demonstrated here with isogeometric analysis using NURBS.

3.1 Isogeometric Analysis

Isogeometric Analysis (IGA) is a numerical method introduced first by [70] closing the gap between Computer Aided Design (CAD) and Computer Aided Engineering (CAE) by using the same basis functions which are used to describe the geometry of the object and to describe the approximated solution fields. In most cases, IGA reduces the complexity of the iterative design to the analysis cycle. For CAD Engineers creating mesh is always a burden to apply the geometry for CAE analysis. This burdens can be eliminated using isogeometric analysis where 80 percent of the time requires an analyst to create the mesh for FEM analysis and this allows most certain geometries to be represented exactly that are only approximated by polynomial functions, including conic and circular sections. IGA applied in several fields of study including structural mechanics, solid mechanics, fluid mechanics and contact mechanics. We will provide in this section an overview of the advantage of using IGA in different applications. Here we will also demonstrate the advantage of IGA in comparison to FEM with presenting some shortcomings of IGA.

3.1.1 Advantages of Isogeometric Analysis

Isogeometric analysis has advantages in different applications comparing to conventional Finite Element Method and Boundary Element Method. They are:

- **Contact formulation** is an area where with the conventional faceted geometry discretizations, the presence of faceted surfaces can lead to oscillation and jumps in traction responses unless very refined mesh can be used in the analysis. The benefits of using NURBS-based geometry over such an application are evident as smooth contact surfaces are obtained, which leads to more physically accurate contact stresses due to smooth higher-order basis functions. Recently some applications in this area have been shown by [73, 118, 119, 88].
- **Optimization** is another field of an area where IGA has some greater advantages over traditional numerical methods where the tight coupling with CAD models offers an extremely alternative approach for industrial applications. In Shape Optimization with NURBS based boundary integral method shows superior computational advantages over the usual Lagrange based BEM [83]. Topology optimization in IGA has shown in a coupled problem with Phase field modeling by [47] which is particularly suitable for phase field problems and allows exact CAD geometry can be suitable to describe the design domain and can also be used in the optimization procedure.
- **Boundary discretization** is also another attractive approach using IGA where it can be applied to direct coupling with CAD model. In this case, isogeometric boundary element methods for elastostatic analysis were presented by [116, 111, 86], demonstrating the advantage of IGA to eliminate mesh generation completely by using CAD discretization directly for analysis.
- **Plate and Shell** analysis is another application where IGA has demonstrated a greater advantage over conventional FEM and BEM methods [72, 29]. The smoothness of the NURBS basis functions allows for a straightforward construction of plates or shell elements. For this shell analysis, particularly for rotation-free shells can be generated easily [72, 77] using smooth spline basis functions. We know that multi-patch NURBS surfaces, rotation-free IGA elements require special treatment at patch boundaries where the basis functions are found to be C^0 continuous. Moreover, isogeometric shell/plate elements demonstrate less shear locking compared to standard FEM shell/plate analysis.
- **Vibration** analysis is an important topic in structural analysis where the advantage of NURBS based IGA [71, 45, 120] using unique k-refinement can provide more robustness and accurate frequency spectrum than typical higher-order FEM p -methods. Particularly, the optical branches of the frequency spectrum, which have been identified as a contributors to **Gibbs** phenomena in wave propagation problem (for the cause of rapid degradation of higher modes in the p -version of FEM) which are eliminated by IGA method. However when lumped mass matrices are used, the accuracy is limited to second order for any basis order. Mathematical properties of IGA were studied vastly by [54].
- **Fluid and fluid-structure interaction** problems is another application of IGA where the smooth NURBS basis functions is attractive for analysis [62, 101, 24, 23, 26]. Constructing higher-order continuous basis functions is easier in IGA to solve PDEs that can incorporate fourth-order (or higher) derivatives of the field variable such as the CahnHilliard equation [61], gradient elasticity [56] and explicit gradient damage models [124]. There is a potential applications in the Kohn-Sham equation for electronic structure modeling of semiconducting materials using higher-order NURBS basis functions with IGA [91].

3.1.2 Advantage of IGA in comparison to FEM

As from the previous subsection 3.1.1 it is shown that Isogeometric analysis has several advantages in comparison to FEM in different applications. The major advantages of CAD spline based basis functions (e.g. B-spline, NURBS, T-spline, PHT-spline) over finite elements are their ability

to describe a larger class of geometric objects, e.g. circular, conic sections. However, the requirements of basis functions in CAE analysis are higher than in CAD design. Besides their potentiality to integrate CAD and CAE analysis, therefore to reduce the computational cost, NURBS-based isogeometric finite element formulations have other advantages over (Lagrange) polynomial based Finite Elements(FEs):

- **Higher-order and continuity** of NURBS basis in many examples of IGA has shown an outperform result comparing to traditional FEM. This IGFEA devoted to higher smoothness and continuity because of the CAD basis functions (B-splines, NURBS, PHT-splines). Higher-order and continuous spline basis functions do not exhibit any jumps or oscillation in derivatives e.g in the stress or strain field during the analysis which usually exhibits in traditional FEM.
- **Large deformation analysis** in initial studies [29] conjecture that IGFEA based on NURBS show less sensitivity with respect to excessive mesh distortion compared to Lagrange polynomial based FEM making them particularly attractive for problems with large deformations analysis such as shear band formation, sheet metal stamping or crashworthiness etc (though mesh distortion can also be a consequence of poor mesh generation). This was again attributed to higher-order and higher-continuity of the approximation.
- **Eigen value** problem where using IGA, it was found that the natural eigen frequencies of higher-order NURBS-based are much lower compared to higher-order Lagrange polynomial based FEM [5]. For explicit time integration analysis it has the greater advantage where the stable time step is inversely proportional to the maximum eigen frequency.
- *hpk*– **refinement** is the basic refinement procedure in Isogeometric analysis. Besides the conventional h-refinement and p-refinement in FEM, NURBS-based IGFEA offers a more flexible k-refinement procedure. The k-refinement procedure is suited for higher-order approximations as it maintains the higher-order continuity and the polynomial order under refinement. Many examples have been shown (see in [70, 71]) demonstrating the superiority of the *k*-refinement over the *p*-refinement.

3.1.3 Drawbacks of NURBS based Isogeometric analysis

Isogeometric analysis based on NURBS also have some certain drawbacks in the context of numerical analysis:

- **Tensor-product** property of NURBS, control points are required to lie in a structured grid (e.g. in a rectangle in 2D). This leads to an excessive overhead of control points with increasing refinement Fig. 3.1(a). A local refinement strategy proposed by [5, 57] , Fig. 3.1(b), but constraint equations require increasing complexity and implementational effort. Moreover, refinement still propagates through a given patch.
- C^0 **continuity across patch boundaries** is another disadvantage of NURBS when two NURBS surfaces do not share a common boundary, they cannot even achieve C^0 continuity without disturbing at least one of the surfaces. It is noted that Lagrange polynomial based FEM also possesses only C^0 continuity across a common boundary.
- **Gaps and overlaps** are probably the most striking drawback in NURBS based geometries. It often unacceptable for analysis and requires additional repairing of the geometry in between patch boundaries to make it analysis suitable.

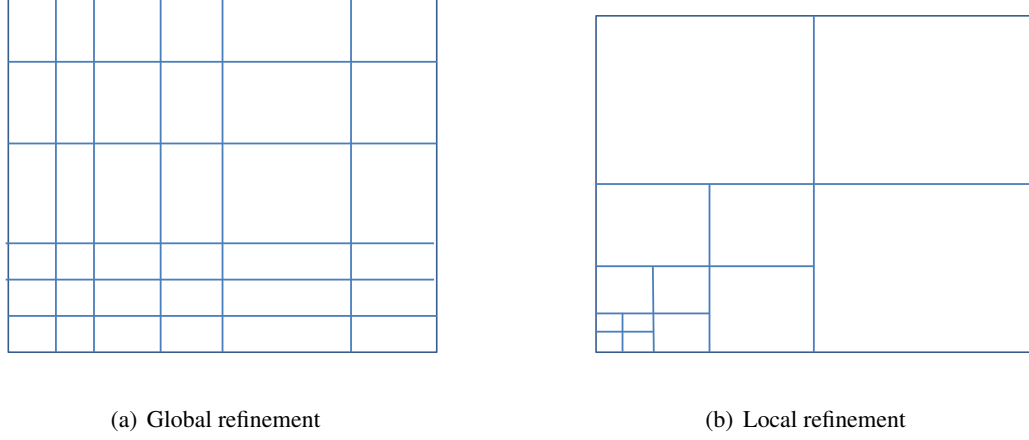


Fig. 3.1: Local basis and coordinate system of T-spline

3.1.4 Mathematical background of linear elasticity using isogeometric analysis

We consider a linear elasto static problem in a bounded domain $\Omega \subset \mathbb{R}^d, d = \{2, 3\}$ and with a Lipschitz boundary bounded by Γ . The boundary is partitioned into two sets, Γ_u and Γ_t with displacements prescribed on Γ_u and tractions $\bar{\mathbf{t}}$ prescribed on Γ_t , where $\Gamma = \Gamma_u \cup \Gamma_t$ and $\Gamma_u \cap \Gamma_t = \emptyset$. The governing equations express equilibrium between the Cauchy stresses σ and the applied body forces \mathbf{b} .

$$-\nabla \cdot \sigma = \mathbf{b} \quad \text{in } \Omega \quad (3.1)$$

The displacement \mathbf{u} is prescribed on the boundary Γ_u and Γ_t by

$$\mathbf{u} = 0 \quad \text{on } \Gamma_u \quad (3.2)$$

$$\sigma \cdot \mathbf{n} = \bar{\mathbf{t}} \quad \text{on } \Gamma_t \quad (3.3)$$

The weak form of a linear elastostatics problem is to find \mathbf{u} in the trial space

$$\mathcal{V} = \{\mathbf{u} \in (H^1(\Omega))^2, \quad \mathbf{u} = 0 \quad \text{on } \Gamma_u\} \quad (3.4)$$

such that for all test functions $\delta \mathbf{u}$ in the test space,

$$\mathcal{V}_0 = \{\delta \mathbf{u} \in (H^1(\Omega))^2, \quad \delta \mathbf{u} = 0 \quad \text{on } \Gamma_u\} \quad (3.5)$$

So the weak form gets in the following form:

$$\int_{\Omega} \epsilon(\mathbf{u}) : \mathbf{D} : \epsilon(\delta \mathbf{u}) d\Omega = \int_{\Gamma_t} \bar{\mathbf{t}} \cdot \delta \mathbf{u} d\Gamma + \int_{\Omega} \mathbf{b} \cdot \delta \mathbf{u} d\Omega \quad (3.6)$$

where the elasticity matrix is denoted by \mathbf{D} and \mathbf{b} refers to a body force. Using the Galerkin method where the same shape functions $R_A(\tilde{\xi})$ are used for both \mathbf{u} and $\delta \mathbf{u}$, we can write

$$\mathbf{u}(\mathbf{x}) = \sum_{A=1}^{n_{np}} R_A(\tilde{\xi}) \mathbf{u}_A, \quad \delta \mathbf{u}(\mathbf{x}) = \sum_{A=1}^{n_{np}} R_A(\tilde{\xi}) \delta \mathbf{u}_A \quad (3.7)$$

where \mathbf{u}_A , $\delta\mathbf{u}_A$ denote the nodal displacements and it's variation respectively and n_{np} is the total number of control points. In 2D, each control points has two unknowns dofs in x and y directions, hence we write as $\mathbf{u}_A = \{u_{xA}, u_{yA}\}$.

Substitution of these approximations from Eq. 3.7 into Eq. 3.6 and using the arbitrariness of the nodal variations give the standard discrete set of equations $\mathbf{K}\mathbf{u} = \mathbf{f}$ with

$$\mathbf{K}_{AB} = \int_{\Omega} \mathbf{B}_A^T \mathbf{D} \mathbf{B}_B d\Omega, \quad \mathbf{f}_A = \int_{\gamma_t} R_A \mathbf{b} d\Omega, \quad A, B = 1, 2, \dots, n_{np}. \quad (3.8)$$

In two dimensions, the strain-displacement matrix B_A is given by,

$$\mathbf{B}_A = \begin{bmatrix} R_{A,x} & 0 \\ 0 & R_{A,y} \\ R_{A,y} & R_{A,x} \end{bmatrix} \quad (3.9)$$

where the shape functions derivative $R_{A,x} = dR_A/dx$ from Eq. 2.18 and can be determined by:

$$\left[\frac{\partial R_{A,x}}{\partial x} \right] = \left[\frac{\partial R_{A,x}}{\partial \xi} \right] \begin{bmatrix} \frac{\partial \tilde{\xi}}{\partial x} & \frac{\partial \tilde{\xi}}{\partial y} \\ \frac{\partial \tilde{\eta}}{\partial x} & \frac{\partial \tilde{\eta}}{\partial y} \end{bmatrix} = \left[\frac{\partial R_a^e}{\partial \xi} \right] \mathbf{J}_{\tilde{\xi}}^{-1} \quad (3.10)$$

where $\mathbf{J}_{\tilde{\xi}}^{-1}$ is the inverse of the Jacobian of mapping $\mathbf{x} : \tilde{\Omega} \mapsto \Omega$ from the parameter space to the physical space.

3.2 Isogeometric simulator in Diffpack

Diffpack is a set of libraries containing building blocks of numerical methods for solving Partial Differential Equations (PDEs). Diffpack utilizes object-oriented (OO) programming techniques to a large extent and is coded in the C++ programming language. To understand how Diffpack-features work a user need to have good programming skills in object-oriented C++. Using Diffpack makes dramatic reduction of implementation efforts for solving PDEs and enables the user to solve real-world problems. It relies on programming and scripting and organize the libraries in terms of C++ classes and class hierarchies (OO design). Diffpack let the simulator mainly contains problem-dependent code and acts as a computational engine with a layered design. Implementation of IGA using Diffpack eases to reuse the Diffpack FEM classes and using Bézier extraction makes the user efforts very less to implement IGA using existing FEM classes in Diffpack.

3.2.1 Advantage of Diffpack

- **Program maintenance:** Programs are easier to read and understand using Diffpack object-oriented programming structure.
- **Program alteration:** User can often make additions and deletions to programs, such as in a database programs, by simply adding or deleting objects.
- **Use objects numerous times:** Using Diffpack a user can save well-designed objects in a toolkit of useful features that can easily insert into new code, with few or no changes.
- **Programming flexibility:** The design of Diffpack supports flexibility of a user to implement complicated programs and allows to take complete control and writes application-specific optimized code.

- **Extensive support:** The Diffpack libraries offer extensive support for numerical experimentation and experienced users can develop their own tailored scripts, e.g. with a full GUI (Graphical User Interface), calling up Diffpack scripts and C++ code.

3.2.2 Application of Diffpack

Some useful application areas of Diffpack are:

- Basic model equations in applied mathematics (Laplace, heat and wave equations.)
- Navier-Stokes equations.
- Water wave equations.
- Multi-phase porous media flow.
- Heat transfer, including phase changes.
- Thermo-elasticity.
- Large deformation plasticity.
- Stochastic PDEs.
- Stochastic ODEs.
- Electrical activity in the heart.

3.2.3 Solving linear elasticity of isogeometric analysis using Diffpack

The application of the FEM to linear elasticity is common and represents a familiar language to many researchers. Here therefore, outline a two-dimensional implementation of IGA for linear elasticity using Diffpack classes, highlighting the differences over conventional Diffpack FEM discretizations.

3.2.4 Isogeometric analysis based on Bézier extraction of NURBS in Diffpack

Implementation of IGA with NURBS as basis function in Diffpack based on Bézier extraction is easy to use in the existing Diffpack FEM kernel. It is also flexible to use the existing Diffpack FEM classes using Bézier extraction with less effort of programming and easy for users as implementation aspects. But as there is no Diffpack CAD software available, so the geometry can be built by Autodesk Rhino3D or NURBS toolbox. We import the CAD geometry in the pre-processing step from Autodesk Rhino3D using the python script to get the geometric information (control points, knot vectors, the order of the basis and side of the geometries etc). The full process of IGA NURBS analysis in several steps is shown in a flow diagram in Fig. 3.2 with the overall implementation procedure in Diffpack.

The implementation procedure of IGA NURBS in Diffpack is described according to the flow diagram from Fig. 3.2 in details below:

Preprocessing:

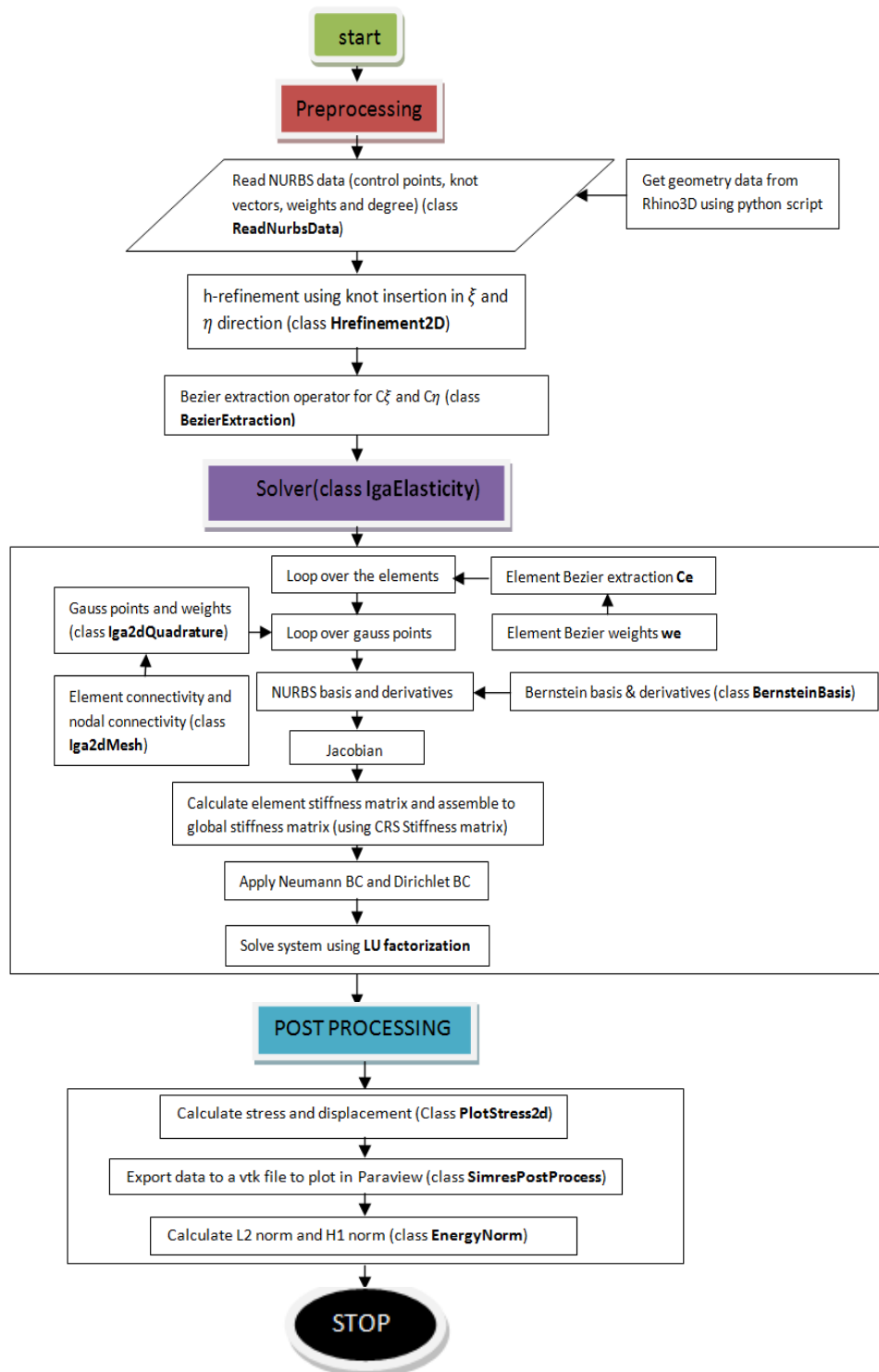


Fig. 3.2: Flow diagram of isogeometric finite element implementation using NURBS in Diffpack [67].

- Design the geometry in Autodesk Rhino3d using NURBS.
- Import the CAD NURBS geometry from Autodesk Rhino3D using a python script or get the geometric information using NURBS toolbox.
- Reading the NURBS geometric information (control points, knot vectors, order of the basis and side of the geometry etc) from the implemented class **ReadBezierExtraction**.
- Calculate the Bézier extraction operators C_ξ and C_η .

Assembly routine (IgaElasticity class):

- Loop over the elements.
- Loop over the gauss points (get the gauss points and weights from class **Iga2dQuadrature**).
- Get the element and nodal connectivity (class **Iga2dMesh**).
- Calculate the elemental Bézier extraction operator C^e and weights w^e .
- Calculate the Bernstein basis and derivatives from Eq. 2.31 (in class **BernsteinBasis**).
- Calculate the NURBS basis (Eq. 2.37) and derivatives (Eq. 2.18) and jacobian matrix (according to Eq. 3.10).
- Calculate the strain displacement matrix B.
- After calculating the element stiffness matrix, assemble it in a global system of equation with a sparse system using a Compressed row storage system (CRS-pattern).
- Apply Neumann and Dirichlet boundary conditions.

Solve the system $KU = f$:

- Solve the sparse system by LU factorization.

Post-processing:

- Calculate the displacements and stress components from the solution field to plot (in class **PlotStress2d**).
- Export data to a vtk/vtu file to plot in Paraview (class **SimresPostProcess**).
- Calculate the L^2 -error norm and H^1 -error norm to get the convergence plots (class **EnergyNorm**).

3.3 Numerical examples of IGA using NURBS

Here we will show some numerical examples of isogeometric analysis using NURBS where uniform refinement has been performed and L^2 and H^1 -error norms are calculated to compare the results with FEM using Diffpack FEM toolbox (see in [82, 66]). At first numerical example, we will show a cylinder problem subjected to internal pressure and another example is a plate with a circular hole constant in-plane tension.

3.3.1 Cylinder subjected to internal pressure

In this example, we will consider a 2D domain which is a pressurized thin cylinder, in Fig. 3.3. Here we select internal and external radius equal to $R_1 = 0.5$ and $R_2 = 1$, and the internal pressure of the cylinder is $P = 30000\text{N}/\text{m}^2$ respectively. The Youngs modulus and Poisson ratios are $E = 3 \times 10^7\text{N}/\text{m}^2$, $\nu = 0.25$ and plane stress condition is considered in this problem. Due to symmetric geometry $\frac{1}{4}$ -th of the geometry is considered, symmetry conditions are imposed on the left and bottom edges, the outer boundary is traction free and internal pressure P imposed on the inner boundary. The exact solution for the stress and displacement components are given in Eq. 6.15 and Eq.6.16.

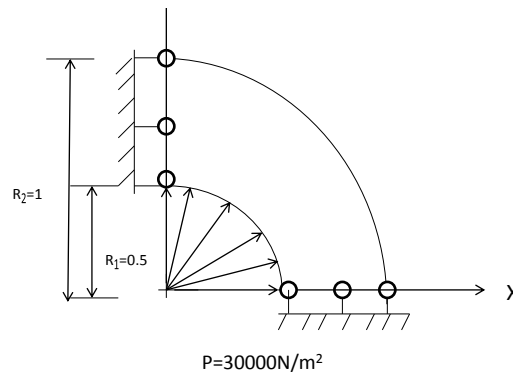


Fig. 3.3: Geometry and material specification of a thin cylinder subjected to internal pressure.

In Fig. 3.4 uniform mesh refinement performed by NURBS due to tensor product structure.

In Fig. 3.5 contour plots of displacement and stress components have been shown.

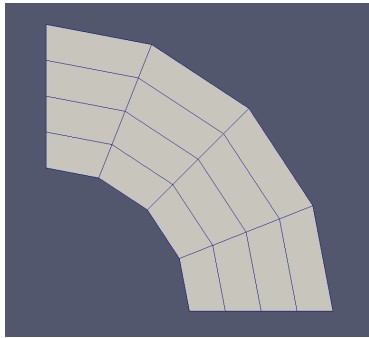
L^2 and H^1 error norms have been compared between IGA NURBS and traditional FEM. Here we considered a quadratic basis in both IGA NURBS and FEM analysis. From Fig. 3.6 we can see that IGA NURBS outperforms comparing to traditional lagrange based FEM due to higher efficiency and accuracy.

3.3.2 Infinite plate circular hole constant in plane tension

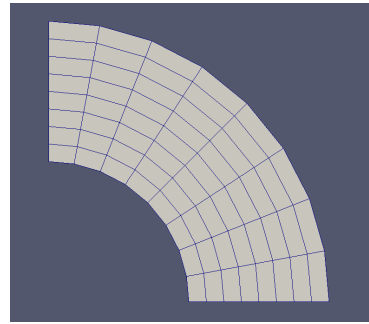
This is an example of an infinite plate with a circular hole under constant load. In Fig. 8.8 the geometry, boundary conditions and material properties of the problem along with the exact solution have been shown. It has an advantage of using symmetrical geometry and for this reason, only a quarter of the model is considered and symmetric boundary conditions imposed at the bottom and right, while exact traction imposed on the left and top edges. No boundary conditions imposed around the hole. The exact solutions of displacements and stresses are in Eq.8.1, 8.2.

In Fig.3.7 the uniform refinement by knot insertion has been shown at different steps with NURBS.

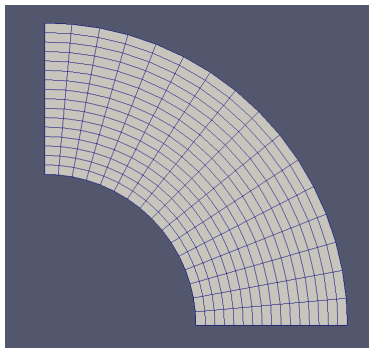
Fig. 3.8 shows the contour plots of displacement and stress components at x, y -direction. It is seen that stress concentration due to the hole at the corner of the plate appears. The applied stress is $T_x = 10$ and the stress contour plots show that the stress concentration of $\sigma_{xx} = 30$ appears at the edge of the hole (at $r = R$ and $\theta = 3\pi/2$).



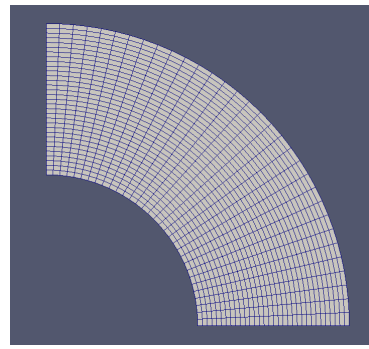
(a) A uniform mesh refinement with NURBS at step 1.



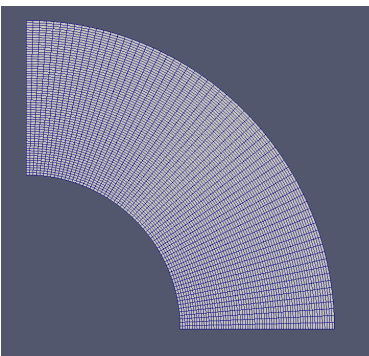
(b) A uniform mesh refinement with NURBS at step 2.



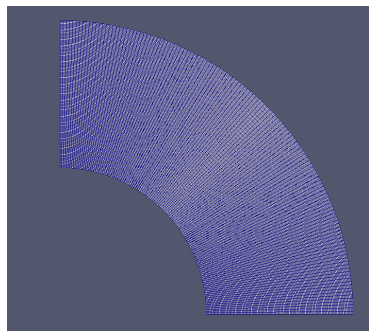
(c) A uniform mesh refinement with NURBS at step 3.



(d) A uniform mesh refinement with NURBS at step 4.



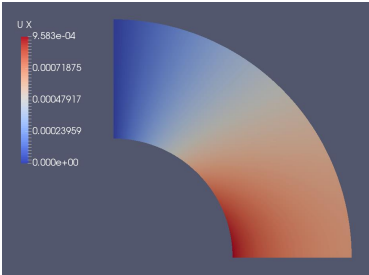
(e) A uniform mesh refinement with NURBS at step 5.



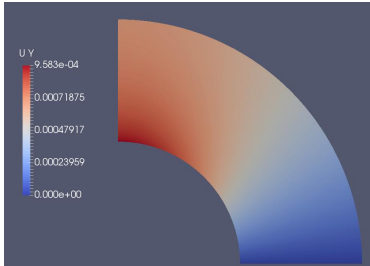
(f) A uniform mesh refinement with NURBS at step 6.

Fig. 3.4: Cylinder subjected to internal pressure meshes at different step where uniform mesh refinement performed.

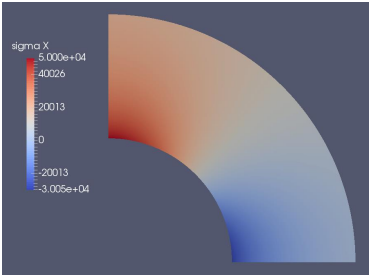
The convergence plot of energy norm (H^1 -norm) of stresses is shown in Fig. 3.9. Here both for isogeometric analysis (IGA) using NURBS and FEM analysis, quadratic basis functions have been used and IGA NURBS outperforms comparing to FEM analysis (although the rate of convergence almost same but the accuracy is higher in the case of IGA NURBS).



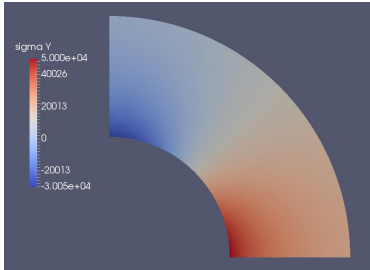
(a) Displacement component in X-direction.



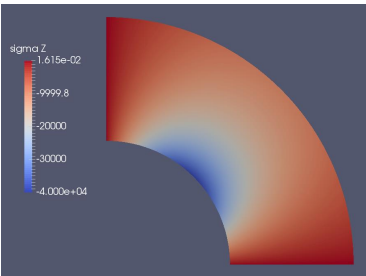
(b) Displacement component in Y-direction.



(c) Stress component in X-direction.

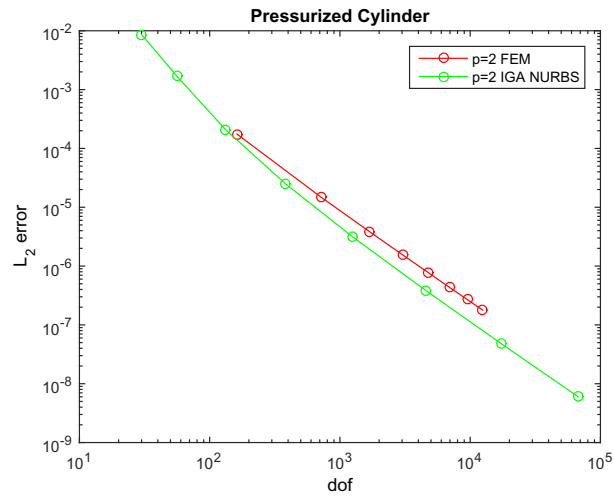


(d) Stress component in Y-direction.

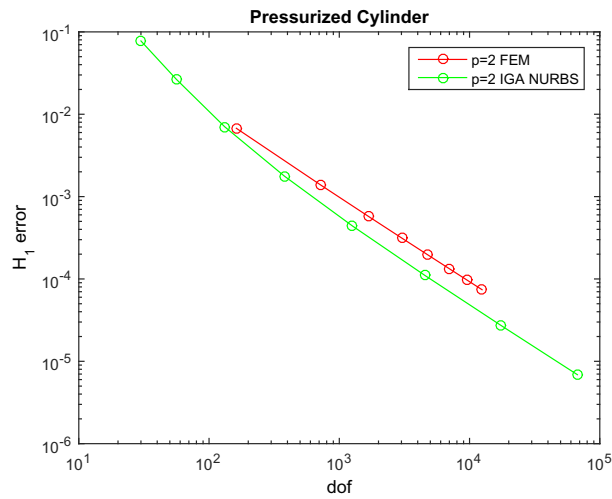


(e) Stress component in XY-direction.

Fig. 3.5: Contour plots of displacement and stress components (x,y) of the cylinder subjected to internal pressure.

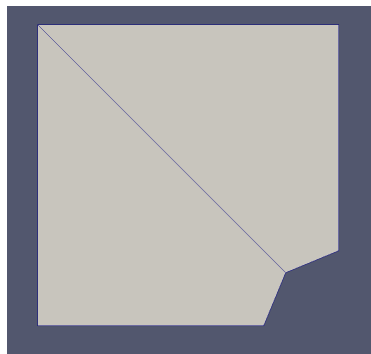


(a) L^2 -error norm for pressurized cylinder quadratic IGA NURBS and FEM.

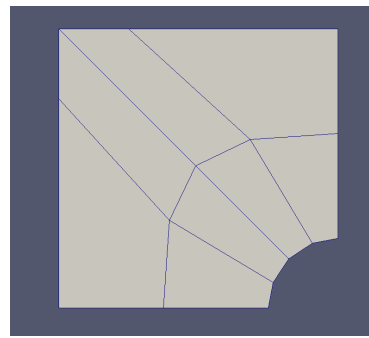


(b) H^1 -error norm for pressurized cylinder quadratic IGA NURBS and FEM.

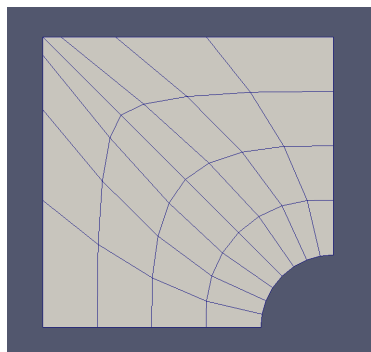
Fig. 3.6: Error norms for pressurized cylinder quadratic IGA NURBS and FEM comparison.



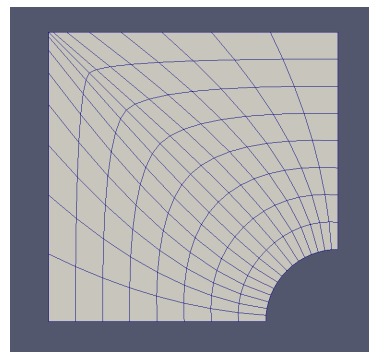
(a) Initial mesh with NURBS at step 1.



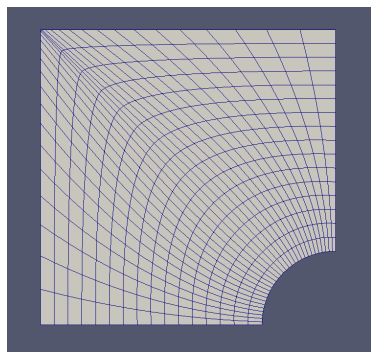
(b) A uniform mesh refinement with NURBS at step 2.



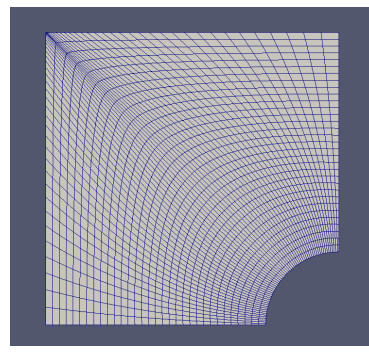
(c) A uniform mesh refinement with NURBS at step 3.



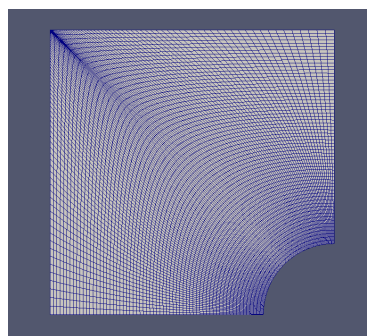
(d) A uniform mesh refinement with NURBS at step 4.



(e) A uniform mesh refinement with NURBS at step 5.

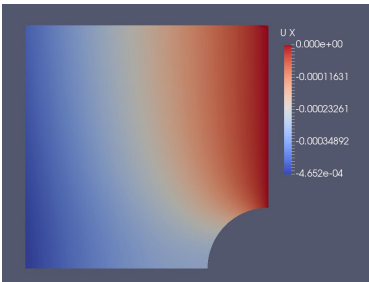


(f) A uniform mesh refinement with NURBS at step 6.

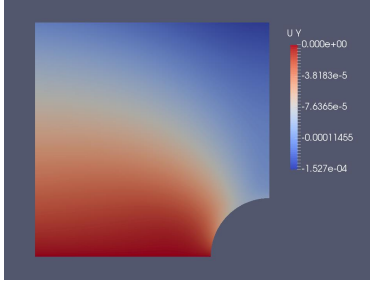


(g) A uniform mesh refinement with NURBS at step 7.

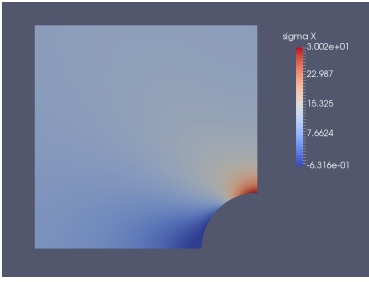
Fig. 3.7: Plate circular hole subjected to infinite tension meshes at different step where uniform mesh refinement performed.



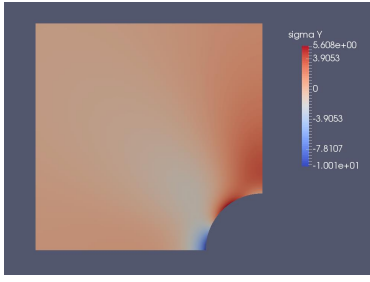
(a) Displacement component in X-direction.



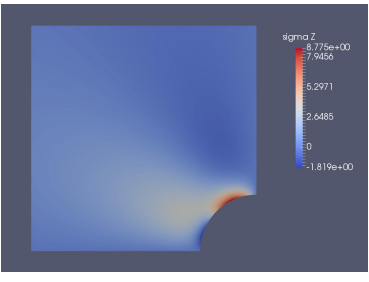
(b) Displacement component in Y-direction.



(c) Stress component in X-direction.



(d) Stress component in Y-direction.



(e) Stress component in XY-direction.

Fig. 3.8: Contour plots of displacement and stress components (x,y) of the plate circular hole subjected to infinite tension.

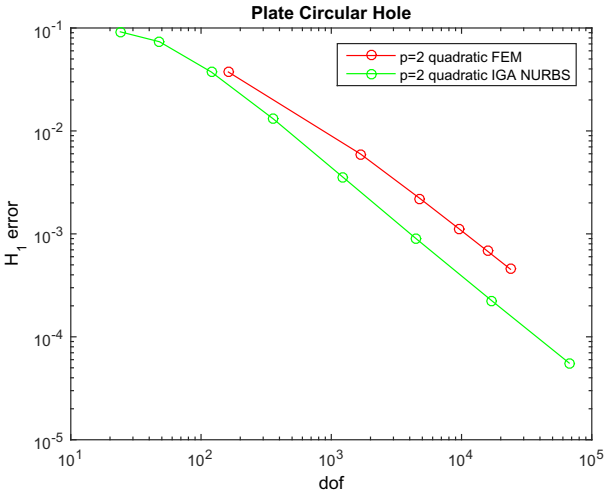


Fig. 3.9: H^1 error norms for plate circular hole quadratic IGA NURBS and FEM analysis comparison.

Chapter 4

Adaptive Geometry Independent Field approximation using PHT-spline and Error estimation

In this chapter, a new discretization scheme will be discussed which is a generalization of the isogeometric analysis method. The main concept in this method is to use a different kind of spline spaces for the geometry and for the solution field. The solution field can be chosen independently of the geometric design space. This method is known as *Geometry Independent Field approximation* (GIFT). It is based on “super-parametric” or “generalized” IGA approach incepted in [90] and presented in [89, 28, 130]. There are several cases of user-defined chosen geometry and field approximation but here we will mainly focus on choosing the geometry defined by NURBS and solution field is to be chosen by PHT-spline where adaptive refinement method can be used using recovery based error estimation.

4.1 Geometry Independent Field approximation (GIFT)

As the main concept of isogeometric analysis (IGA) [70] is to utilize the same spline representation for both the geometry and the unknown solution fields (see Fig. 4.1), considering an advantage for the integration of Computer Aided Geometric Design (CAGD) and subsequent analysis in Computer Aided Engineering (CAE). But using the same spline spaces for both geometry and field approximation creates some cases a major constraint when the geometry spline space is not well-suited to approximate the solution for solving the Partial Differential Equation (PDE). The main problem arises when we have a singularity in the solution field which needs to be captured locally by mesh refinement. For this reason a new discretization method, called **Geometry Independent Field approximation (GIFT)** proposed by [130], according to which the spaces used for the geometry and the field variables can be selected by different splines and can be adapted independently. This can allow more flexibility to the users or the analyst to choose different spline space for the solution field approximation by preserving geometric exactness and tight CAD integration.

In the proposed method, for a given computational domain with spline form, the solution field may have a different kind of spline representation, such as PHT-splines, T-splines and generalized B-splines or B-splines of different degrees. Importantly, the geometry of the computational domain has the same spline representation as that of the given CAD boundary, which allows the method to preserve the potential to realize the seamless integration of CAD and CAE analysis. There are several shortcomings can overcome by GIFT which are:

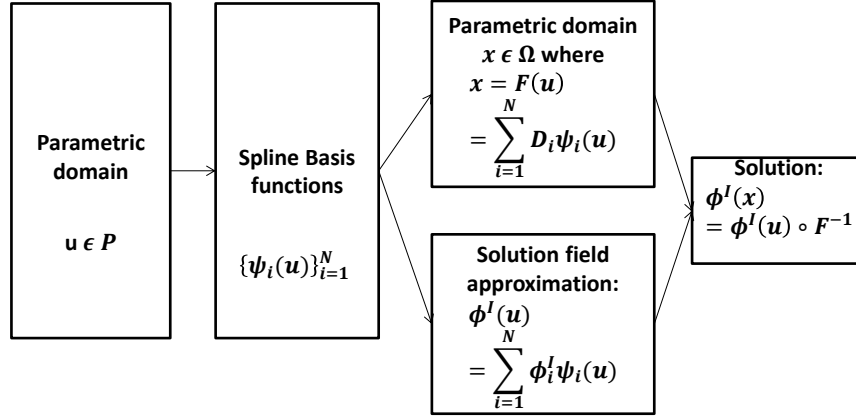


Fig. 4.1: Main idea of isogeometric analysis

- The flexibility of choosing the solution spline field by another kind of spline independent of the geometric field represented by spline.
- Multi-patch configuration could be easier without additional constraint.
- The refinement of the solution field is independent.

4.1.1 Main idea of GIFT

In this sub-section, we describe the main idea behind this GIFT method. Suppose the computational domain $\Omega \in \mathbb{R}^d$, $d \geq 2$, is defined on parametric domain \mathcal{P} has the following spline representation:

$$\Omega = \{x : x = \mathbf{F}(\mathbf{u}), \mathbf{u} \in \mathcal{P}\}, \quad \mathbf{F}(\mathbf{u}) = \sum_{i=1}^N D_i \psi_i(\mathbf{u}) \quad (4.1)$$

where $D_i \in \mathbb{R}^d$ are the d -dimensional control points, $\psi_i(u)$ are tensor product isogeometric spline basis functions defined over d given knot vectors.

The basic idea of IGA (Fig.4.1) is to represent the solution field $\phi(\mathbf{u})$ in the parameter space using the same spline representation of the geometry defined spline, i.e

$$\phi^I(\mathbf{u}) = \sum_{i=1}^N \phi_i^I \psi_i(\mathbf{u}) \quad (4.2)$$

where $\phi_i^I(\mathbf{u}), i = 1, \dots, N$ are the control variables that need to be solved for the solution. When h -refinement and p -refinement are performed on the solution field, the same parameterization remains of the refined computational domain.

In the proposed GIFT method, the solution field $\phi^G(\mathbf{u})$ can have different spline representation as shown in Fig. 4.2:

$$\phi^G(\mathbf{u}) = \sum_{j=1}^M \phi_j^G \chi_j(\mathbf{u}) \quad (4.3)$$

where $\phi_j^G(\mathbf{u}), j = 1, \dots, M$ are the control variables which need to be solved in the solution space

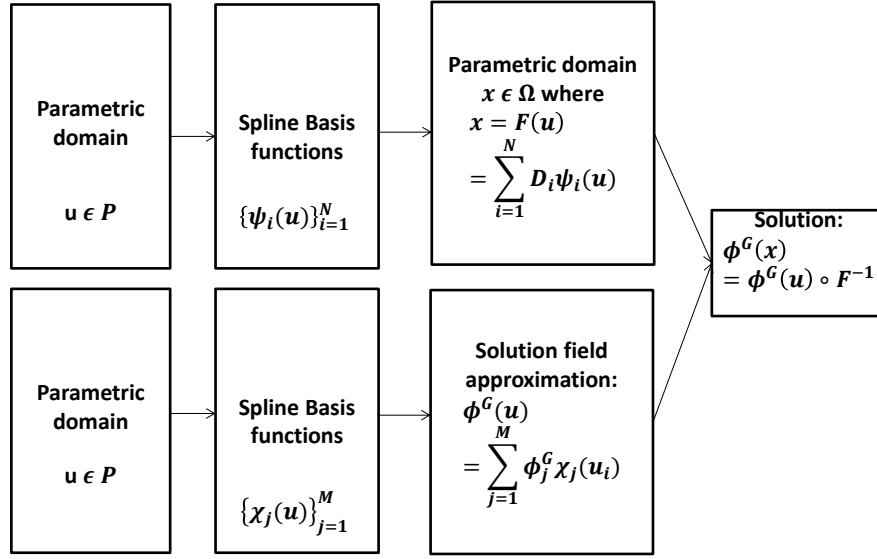


Fig. 4.2: Main idea of GIFT

and $\chi_j(\mathbf{u})$ are the basis functions of the specified solution spline space defined on the parametric domain \mathcal{P} , which can be PHT-splines, T-splines and generalized B-splines or NURBS of different degree.

In GIFT, the computational domain Ω has the same spline representation as that of the given CAD boundary, hence it does not violate the main idea of integration of CAD and CAE.

If $\chi_j(\mathbf{u}) = \psi_i(\mathbf{u})$, the GIFT method works same as IGA or, in other words, it can be said that GIFT is a generalization process of IGA.

4.1.2 A general framework for GIFT

In this sub-section, the mathematical foundation of the GIFT method will be formulated. In what follows, we consider an open domain $\Omega \in \mathbb{R}^d$, $d \geq 2$ with boundary $\Gamma = \partial\Omega$, consisting of two parts Γ_D and Γ_N , such that: $\Gamma = \overline{\Gamma_D} \cup \overline{\Gamma_N}$, $\Gamma_D \cap \Gamma_N = \emptyset$. The boundary problem (strong form) for an elliptical partial differential operator of the second-order \mathcal{A} consists in finding the function $u : \overline{\Omega} \mapsto \mathbb{R}$ such that:

$$\begin{aligned} \mathcal{A}u &= f, & x \in \Omega \\ u &= 0, & x \in \Gamma_D \\ \frac{\partial u}{\partial n_A} &= g, & x \in \Gamma_N \end{aligned} \tag{4.4}$$

where \mathcal{A} is given by

$$\mathcal{A}u = -\partial_j(a_{ij}(x)\partial_i u) + b_j(x)\partial_j u + c(x)u \tag{4.5}$$

and Neumann boundary condition $\frac{\partial u}{\partial \mathbf{n}_A}$ is defined by

$$\frac{\partial u}{\partial \mathbf{n}_A} = n_j a_{ij}(x) \partial_i u \quad (4.6)$$

and \mathbf{n} is a unit normal to Γ_N .

Next, we can define the functional space in the following way:

$$H_0^1(\Omega) = \{u \in H^1(\Omega), u|_{\Gamma_D} = 0\} \quad (4.7)$$

The weak form of the problem Eq. 4.4 consists in finding $u \in H_0^1(\Omega)$, such that for any $v \in H_0^1(\Omega)$

$$a(u, v) = l(v), \quad (4.8)$$

where

$$\begin{aligned} a(u, v) &= \int_{\Omega} \{a_{ij}(x) \partial_i u \partial_j v + b_j(x) \partial_j u v + c(x) u v\} d\Omega, \\ l(v) &= \int_{\Omega} f(x) v(x) d\Omega + \int_{\Gamma_N} g(x) v(x) d\Gamma \end{aligned} \quad (4.9)$$

Next we introduce the parameterization of the physical domain Ω on a parametric domain \mathcal{P} :

$$\mathbf{F} : \mathcal{P} \rightarrow \Omega, \quad \mathbf{x} = \mathbf{F}(\mathbf{u}), \quad \mathbf{x} \in \Omega, \mathbf{u} \in \mathcal{P}. \quad (4.10)$$

Typically, geometrical map \mathbf{F} is given by a set of spline basis functions $\psi_{i_1, i_2, \dots, i_d}(\mathbf{u})$ and a set of control points D_{i_1, i_2, \dots, i_d} as

$$\mathbf{F}(\mathbf{u}) = \sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \dots \sum_{i_d=1}^{n_d} D_{i_1, i_2, \dots, i_d} \psi_{i_1, i_2, \dots, i_d}(\mathbf{u}), \quad (4.11)$$

where $\psi_{i_1, i_2, \dots, i_d}(\mathbf{u})$ can be a tensor product of NURBS, B-splines, T-splines, PHT-splines, LR-spline, THB-spline, etc. For brevity Eq. 4.11 is written as

$$\mathbf{F}(\mathbf{u}) = \sum_{i=1}^N D_i \psi_i(\mathbf{u}), \quad N = n_1, n_2, \dots, n_d \quad (4.12)$$

In what follows, we need to calculate the Jacobian matrix $\mathbf{J}(\mathbf{u})$ of the mapping \mathbf{F} given by

$$\mathbf{J}(\mathbf{u}) = \sum_{i=1}^N D_i \frac{\partial \psi_i(\mathbf{u})}{\partial \mathbf{u}} \quad (4.13)$$

The main idea of the GIFT method is to seek for a solution independent of the geometry spline space

$$V = \{u \in \text{span}\{\chi_{i_1, i_2, \dots, i_d}(\mathbf{u})\} \circ \mathbf{F}^{-1}, u|_{\Gamma_D} = 0\}, \quad (4.14)$$

where $i_1 = 1..m_1, i_2 = 1..m_2, \dots, i_d = 1..m_d$ and $\chi_{i_1, i_2, \dots, i_d}(\mathbf{u})$ can be a tensor product of B-splines, T-splines, THB-spline, PHT-splines, LR-spline or NURBS of different degree etc.

The problem then consists in finding $u^G \in V$ such that for any $v^G \in V$

$$a(u^G, v^G) = l(v^G) \quad (4.15)$$

The solution can be sought in the form

$$\phi^G(x) = \sum_{j=1}^M \phi_j^G \chi_j(\mathbf{u}) \quad (4.16)$$

where $\chi_j(\mathbf{u}) = \chi_{i_1, i_2, \dots, i_d}(\chi_j(\mathbf{u})) \circ \mathbf{F}^{-1}$, $\chi = \chi_1, \chi_2 \dots \chi_d$ and ϕ_j^G are unknown control variables. Then Eq. 4.15 is transformed into a linear system

$$\mathbf{K}\mathbf{u} = \mathbf{f} \quad (4.17)$$

where the stiffness matrix \mathbf{K} and the force vector \mathbf{f} are given by:

$$K_{\gamma\beta} = a(\chi_\gamma(\mathbf{u}), \chi_\beta(\mathbf{u})) \quad (4.18)$$

$$f_\gamma = l(\chi_\gamma(\mathbf{u})) \quad (4.19)$$

and vector ϕ consists of all unknown control variables ϕ_j^G .

4.2 GIFT for Linear elasticity

In this sub-section, we demonstrate applications of the GIFT method for two-dimensional linear elasticity problems [68], in which the computational domain is parameterized by a planar NURBS surface, and the solution is sought by PHT-spline space. Although the presentation is done in the univariate case (see chapter. 5), bivariate case (see chapter. 6), the trivariate case follows identical principles that have been discussed in the chapter. 7.

The governing equations are the two-dimensional linear elastostatics equations under small deformation assumption. The linear elastic solid is modeled as a planar domain Ω and a body force g within the domain. The displacement field \mathbf{u} satisfies the Dirichlet boundary condition $\mathbf{u} = \bar{\mathbf{u}}$ on Γ_u and Neumann boundary condition \mathbf{t} on Γ_t . An arbitrary virtual displacement $\delta\mathbf{d}$ result in compatible virtual strains $\delta\epsilon$ and internal displacements $\delta\mathbf{u}$. The relation between the displacement field \mathbf{u} , the infinitesimal strain tensor $\epsilon = [\epsilon_{ij}]$ and the Cartesian components of the Cauchy *stress tensor* $\sigma = [\sigma_{ij}]$ are:

The compatibility relations

$$\epsilon_{i,j} = \frac{1}{2}(\delta_j u_i + \delta_i u_j) \quad \text{in } \Omega \quad (\forall i, j \in 1, 2, 3) \quad (4.20)$$

The constitutive relations comes from this strain tensor to the aforementioned stress tensor which is the generalized Hooke's law governed by,

$$\sigma_{ij} = C_{ijkl} \epsilon_{kl} \quad \text{in } \Omega \quad (4.21)$$

we will assume that the body is homogeneous and isotropic. That is, the elastic coefficients have the form

$$C_{ijkl} = \lambda \delta_{ij} \delta_{kl} + \mu (\delta_{ik} \delta_{jl} + \delta_{il} \delta_{jk}), \quad (4.22)$$

where the **Kronecker delta** is defined by

$$\delta_{ij} = \begin{cases} 1 & \text{if } i=j \\ 0 & \text{otherwise} \end{cases}$$

Constants λ and μ are the Lamé parameters of the material and These are frequently expressed in terms of the Youngs modulus, E , and Poissons ratio, ν , as

$$\lambda = \frac{\nu E}{(1 + \nu)(1 - 2\nu)} \quad (4.23)$$

$$\mu = \frac{E}{2(1 + \nu)}$$

We can state the strong form of the boundary value equilibrium equations:

$$\begin{aligned} \sigma_{ij,j} + f_i &= 0 \quad \text{in } \Omega \\ u_i &= g_i \quad \text{on } \Gamma_u \\ \sigma_{ij}n_j &= \bar{t}_i \quad \text{on } \Gamma_t \end{aligned} \quad (4.24)$$

where given $f_i : \Omega \rightarrow \mathbb{R}^2$, $g_i : \Gamma_u \rightarrow \mathbb{R}^2$, and $\bar{t}_i : \Gamma_t \rightarrow \mathbb{R}^2$. σ_{ij} is defined in terms of u_i by Eq. 4.20 and 4.21 respectively.

We are applying these Dirichlet and Neumann boundary conditions in each direction independently and thus, $\overline{\Gamma_u \cup \Gamma_t} = \Gamma$ and $\Gamma_u \cap \Gamma_t = \phi$ for $i = 1, \dots, d$. In this context, g_i and \bar{t}_i are referred to as “prescribed boundary displacements” and “tractions” respectively.

Let the two spaces of kinematically admissible displacements, denoted by \mathcal{V} and \mathcal{V}_0 , respectively

$$\mathcal{V} = \{\mathbf{u} \in (H^1(\Omega))^2, \quad \mathbf{u} = \mathbf{g} \quad \text{on } \Gamma_u\} \quad (4.25)$$

$$\mathcal{V}_0 = \{\mathbf{v} \in (H^1(\Omega))^2, \quad \mathbf{v} = 0 \quad \text{on } \Gamma_v\} \quad (4.26)$$

Here, $H^1(\Omega)$ denotes the Hilbert space of order 1 defined as,

$$H^1(\Omega) = \{u \mid \|u\|_{H^1(\Omega)} \leq \infty\} \quad (4.27)$$

with

$$(u, v)_{H^1(\Omega)} = (u, v)_{L^2(\Omega)} + (\nabla u, \nabla v)_{L^2(\Omega)} \quad ; \quad \|u\|_{H^1(\Omega)} = [(u, v)_{H^1(\Omega)}]^{1/2} \quad (4.28)$$

and ∇ is the gradient vector.

The spaces, \mathcal{V} and \mathcal{V}_0 equipped to a bounded energy in a stable solid

$$\int_{\Omega} C_{ijkl} \epsilon_{ij}(\mathbf{u}) \epsilon_{kl}(\mathbf{v}) d\Omega < \infty \quad (4.29)$$

From Eq. 4.29, both \mathcal{V} and \mathcal{V}_0 spaces define the energy norm

$$\|\mathbf{u}\|_E = \left(\int_{\Omega} C_{ijkl} \epsilon_{ij}(\mathbf{u}) \epsilon_{kl}(\mathbf{v}) d\Omega \right)^{1/2} \quad (4.30)$$

where C is a bounded positive definite matrix. The problem in strong form can be stated as weak

form which is given by: find $\mathbf{u} \in \mathcal{V}$ such that

$$a(\mathbf{u}, \mathbf{v}) = l(\mathbf{v}) \quad \forall \mathbf{v} \in \mathcal{V}_0 \quad (4.31)$$

as like Eq. 4.8, where

$$\begin{aligned} a(\mathbf{u}, \mathbf{v}) &= \int_{\Omega} \epsilon^T(\mathbf{u}) \mathbf{D} \epsilon(\mathbf{v}) d\Omega \\ l(\mathbf{v}) &= \int_{\Omega} \mathbf{f}^T \mathbf{v} d\Omega + \int_{\Gamma_t} \bar{\mathbf{t}}^T \mathbf{v} d\Gamma \end{aligned} \quad (4.32)$$

By projecting the solution \mathbf{u} and the test functions \mathbf{v} to a finite-dimensional subspace \mathcal{V}^h of \mathcal{V} . The statement of the discrete problem becomes finding a discrete solution $\mathbf{u}^h \in \mathcal{V}^h$ that satisfies like Eq. 4.15

$$a(\mathbf{u}^h, \mathbf{v}^h) = l(\mathbf{v}^h) \quad \forall \mathbf{v}^h \in \mathcal{V}_0^h \quad (4.33)$$

where

$$\mathcal{V}_0^h = \{\mathbf{v}^h \in \mathcal{V}^h, \quad \mathbf{v} = 0 \quad \text{on} \quad \Gamma_v\} \quad (4.34)$$

The principle of virtual work can be written as follows:

$$\int_{\Omega} \delta \epsilon^T \epsilon^T(\mathbf{u}) \mathbf{D} \epsilon(\mathbf{v}) d\Omega - \int_{\Omega} \delta \mathbf{u}^T \mathbf{b} d\Omega - \int_{\Gamma_t} \delta \mathbf{u}^T \bar{\mathbf{t}} d\Gamma = 0 \quad (4.35)$$

where \mathbf{b} is the body force vector. The virtual displacements and the compatible strains $\epsilon = \nabla_s \mathbf{u}$ within any element can be written like Eq. 4.16 as follows:

$$\delta \mathbf{u}^h = \sum_{I=1}^N \mathbf{M}_I \delta \mathbf{d}_I, \quad \delta \epsilon^h = \sum_{I=1}^N \mathbf{B}_I \delta \mathbf{d}_I \quad (4.36)$$

where $N = np$ total number of control variables and $\mathbf{d}_I = [u_I \quad v_I]^T$ is the displacement vector and M_I is the shape function. Then Eq. 4.35 gives the assembly process like:

$$\int_{\Omega} \delta \mathbf{d}^T \mathbf{B}^T \mathbf{D} \mathbf{B} d\Omega - \int_{\Omega} \delta \mathbf{d}^T \mathbf{M}^T \mathbf{b} d\Omega - \int_{\Gamma_t} \delta \mathbf{d}^T \mathbf{M}^T \bar{\mathbf{t}} d\Gamma = 0 \quad (4.37)$$

Where \mathbf{B} is the standard strain-displacement matrix and \mathbf{D} is a matrix of material constraints. From the expressions related to arbitrary virtual displacements $\delta \mathbf{d}^T$, we have

$$\int_{\Omega} \mathbf{B}^T \mathbf{D} \mathbf{B} d\Omega - \int_{\Omega} \mathbf{M}^T \mathbf{b} d\Omega - \int_{\Gamma_t} \mathbf{M}^T \bar{\mathbf{t}} d\Gamma = 0 \quad (4.38)$$

The resulting governing Eq. 4.38 can be written like Eq. 4.17 as a linear system as follows:

$$\mathbf{K} \mathbf{u} = \mathbf{f} \quad (4.39)$$

where,

$$\begin{aligned} \mathbf{K}_{\gamma\beta} &= \int_{\Omega} \mathbf{B}_{\gamma}^T \mathbf{D} \mathbf{B}_{\beta} d\Omega = \int_{\mathcal{P}} \mathbf{B}(\mathbf{u})^T \nabla_{\mathbf{u}} M_{\gamma}(\mathbf{u}) \mathbf{D} \nabla_{\mathbf{u}} M_{\beta}(\mathbf{u}) \mathbf{B}(\mathbf{u}) \det|\mathbf{J}(\mathbf{u})| d\mathbf{u} \\ \mathbf{f}_{\gamma} &= \int_{\Omega} \mathbf{M}_{\gamma}^T \mathbf{b} d\Omega + \int_{\Gamma_t} \mathbf{M}_{\gamma}^T \bar{\mathbf{t}} d\Gamma = \int_{\mathcal{P}} \mathbf{M}_{\gamma}^T(\mathbf{u}) \mathbf{b}(\mathbf{u}) \det|\mathbf{J}(\mathbf{u})| d\mathbf{u} + \int_{\Gamma_t} \mathbf{M}_{\gamma}^T(\mathbf{u}) \bar{\mathbf{t}} \det|\mathbf{J}(\mathbf{u})| d\mathbf{u} \end{aligned} \quad (4.40)$$

In two dimension, the strain-displacement matrix \mathbf{B}_{γ} can be written as

$$\mathbf{B}_{\gamma} = \begin{bmatrix} M_{\gamma,x} & 0 \\ 0 & M_{\gamma,y} \\ M_{\gamma,y} & M_{\gamma,x} \end{bmatrix} \quad (4.41)$$

where, $M_{\gamma,x} = \nabla M_{\gamma}(\mathbf{x})$ is the derivative of the shape function. The Jacobian $\mathbf{J}(\mathbf{u})$ corresponds to the parameterization of domain Ω , as given by Eq. 4.13 and $\mathbf{B}(\mathbf{u}) = \mathbf{J}^{-1}(\mathbf{u})$. Note, that these two matrices fully depend on the parameterization of the domain and refinement operations (h-,p-refinement) are performed only on the solution space.

4.3 Patch test

To study the behavior of the GIFT method, we first perform standard patch tests [69] in chapter 3 and 4. In finite element methods, patch tests are used to determine whether an arbitrary “patch” of elements can exactly represent all rigid body motions and constant strain states exactly. Also, patch test verifies a “patch” can reproduce basic constant and linear deformation states. From a more practical standpoint, patch tests are also used to assess the correctness of a finite element implementation. They are considered by engineers to be practical ways of assessing whether or not finite elements are convergent and whether or not they have been programmed correctly.

“Patch-test” can not be guaranteed for all the cases in GIFT as mentioned in detail in [13], as for the patch test to pass it is required for the geometry and the field bases to be equivalent. It is also dependent upon certain operations like degree elevation or knot insertion to have a common **parent basis**. In this section, we will perform the patch test for 2D static linear elasticity problem to investigate the accuracy and the convergence properties of the proposed GIFT method.

For a two-dimensional static linear elasticity problem, we took the Timoshenko beam example where the geometry is defined by NURBS and the solution will be represented by PHT-spline.

A Timoshenko beam with length L and height H and unit thickness is studied as a benchmark here. The beam is subjected to parabolic traction at the free end as shown in Fig. 4.3. The parameters are: length $L = 48\text{m}$, height $H = 12\text{m}$, Young's modulus $E = 3.0 \times 10^7 \text{kPa}$, Poisson's ratio $\nu = 0.3$, $P = 1000N$. The analytical solution by [122] is given below:

$$\begin{aligned} u_x &= \frac{Py}{6EI} \left[(6L - 3x)x + (2 + \nu)(y^2 - \frac{H^2}{4}) \right] \\ u_x &= -\frac{Py}{6EI} \left[3\nu y^2(L - x) + (4 + 5\nu)\frac{H^2 x}{4} + (3L - x)x^2 \right] \end{aligned} \quad (4.42)$$

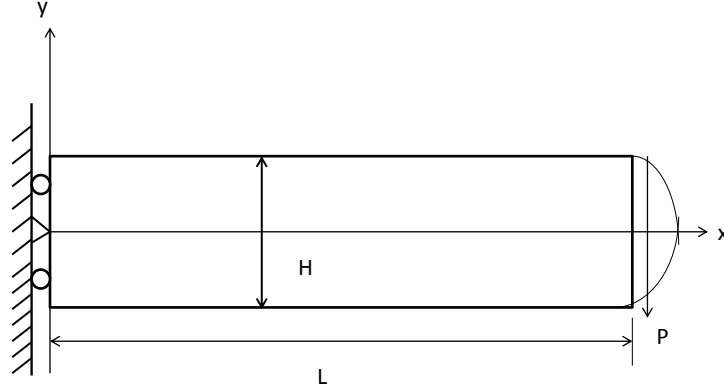


Fig. 4.3: Geometry with the boundary condition of Timoshenko beam.

where the moment of inertia I for a beam with a rectangular cross-section and unit thickness is given by $I = H^3/12$ and plane stress condition is considered here. The stresses corresponding to the displacements Eq. 4.42 are:

$$\begin{aligned}\sigma_{xx} &= \frac{P(L-x)y}{I}, \\ \sigma_{yy} &= 0, \\ \tau_{xy} &= \frac{P}{2I}\left(y^2 - \frac{H^2}{4}\right)\end{aligned}\tag{4.43}$$

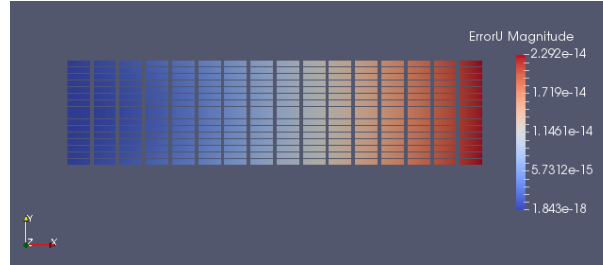
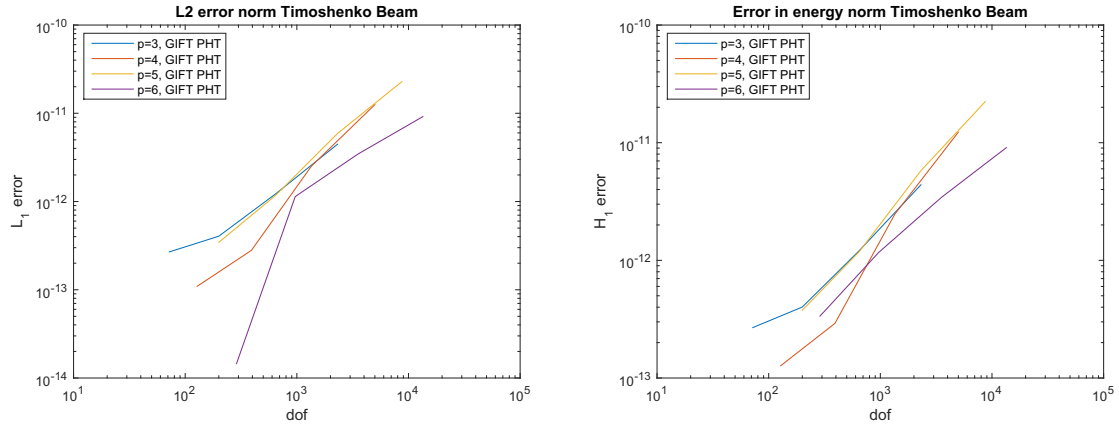


Fig. 4.4: Error plot L^2 displacement norm of Timoshenko beam.

Fig. 4.4 shows the contour plot of the error in L^2 displacement norm. The L^2 and H^1 error norms have been shown in Fig 4.5. It is seen that the proposed GIFT method passes the patch test to machine precision for higher-order PHT-splines.

4.4 Adaptive GIFT using PHT-splines

In this chapter, we study the adaptive GIFT approach where geometry is defined with NURBS and the solution space is defined by PHT-splines. Both NURBS 2.2 and PHT-splines have been discussed elaborately in the chapter. 5, 6, 7. Within this GIFT framework, the computational domain is chosen in NURBS space whereas the solution field is defined by PHT-spline. Firstly, we construct the parametric domain of the PHT-spline model to represent the solution field. We can then get the analytic expression of the initial solution field, in which the unknown control variables can be solved by the method presented in section 4.2. By using a recovery based a-posteriori error estimation technique, the supporting cell with larger errors in the parametric domain of the solution field can be marked, and local h-refinement (cross-insertion) is performed only on the solution field. Several local refinement steps can be performed until the desired error level is achieved.



(a) L^2 -error norm for Timoshenko beam higher-order PHT-splines. (b) H^1 -error norm for Timoshenko beam higher-order PHT-splines.

Fig. 4.5: Error norms of Timoshenko beam higher-order PHT-splines.

4.4.1 Solving and adaptive local refinement with PHT-spline

An initial construction of the solution field can be represented by PHT-spline which is independent of geometrical NURBS representation. The control variables of the solution field are obtained according to Eq. 4.16 by solving the linear elasticity problem as introduced in section 4.2.

In order to obtain a solution with required target accuracy, refinement operations are often needed to increase the approximation power of the solution space locally and thereby optimize the computational expense required for a given accuracy level. Contrary to the classical isogeometric analysis, the refinement operation in GIFT is only performed on the solution field, while the parameterization of the computational domain remains the same during the refinement process. Choosing a PHT-spline approximation offers a natural local refinement. Local refinement is performed on the sub-patches of the solution field on which the local error indicator is larger than a given marking threshold. The corresponding working flow is summarized in an Algorithm in Table. 4.1.

Tab. 4.1: Local refinement scheme in GIFT using PHT-splines.

Process	Parameter	Description of the process
Preprocessor	Geometric space (NURBS geometry field)	Planar NURBS parameterization $\mathbf{P}(\mathbf{u}, \mathbf{v})$ of computational domain Ω .
Solving	Approximate space (PHT-spline solution field)	Compute the PHT-spline solution $\mathcal{U}(\mathbf{u}, \mathbf{v})$ of model problem 4.24 over the given NURBS parameterization $\mathbf{F}(\mathbf{u}, \mathbf{v})$ by GIFT mapping.
Error Estimation	Approximate space (PHT-spline solution field)	Calculate recovery based error estimator for refinement patch by patch on the solution field (subsection 4.5)
Marking algorithm	Approximate space (PHT-spline solution field)	<ol style="list-style-type: none"> 1. Mark in the parametric cell of the PHT-spline solution $\mathcal{U}(\mathbf{u}, \mathbf{v})$ (subsection 4.5.3) in the patches where the elements with the largest contribution to the total error. 2. Insert cross in the marking cells.
Resolving the PHT-spline solution	Approximate space (PHT-spline solution field)	<ol style="list-style-type: none"> 1. Construct the PHT-spline basis functions over the refined T-mesh of the parametric domain. 2. Recompute the PHT-spline solution $\mathcal{U}(\mathbf{u}, \mathbf{v})$ in the refined PHT-spline solution space. 3. Repeat the above refinement steps until the estimated global error $<$ given threshold (target relative error).

4.5 Recovery-based error estimation

In this section, we will present the application of superconvergent recovery-based error estimators to hierarchical splines and a practical way to perform the patch-recovery, including the marking and refinement algorithms. The method proposed is based on exploiting the hierarchical structure of the PHT mesh to define the recovery patches and it can be used for 2D as well as 3D domains. While this procedure can be applied to various PDEs and quantities of interest, we focus here on stress-recovery and error estimation in energy norm for linear elasticity problems.

The key idea of recovery-based error estimation is to derive a smoother and more accurate approximation of the quantity of interest (e.g. stress fields) by sampling the computed solution at carefully chosen points and performing a higher-order polynomial fit through the sample points. This makes use of some assumptions, particularly regarding the continuity of the exact solution. In practice, even when some of these assumptions are not satisfied, the recovered solution can still be used as an *error*

indicator to mark for refinement the regions of domain where the solution is rough.

4.5.1 Determining the superconvergent point locations

To find the locations of the superconvergent points, we proceed similarly as in [126, 10], by considering the Taylor expansion of degree $p+1$ of the exact solution. Suppose Ω is the computational domain (in the physical space) and $\Omega_1 \subset \Omega_0 \subset \Omega$ is a subdomain containing a point x_0 . Let u be the exact solution, and let u_h be an H^1 projection in the approximation space $S^h(\Omega)$, i.e. u_h satisfies

$$B(u - u_h, \chi) = 0, \quad \forall \chi \in S^h(\Omega), \quad (4.44)$$

where

$$B(u, v) = \int_{\Omega} \nabla u \nabla v \, d\Omega, \quad \forall u, v \in H^1(\Omega). \quad (4.45)$$

Furthermore, let Q be a $p+1$ degree polynomial approximation to u (e.g. the Taylor expansion of u centered at x_0). A crucial observation is that for a uniform mesh and when the exact solution is a polynomial of degree $p+1$, i.e. $u = Q$, the spline approximation of degree p to u can be locally described (up to higher-order terms) by a periodic function. More precisely, we can write,

$$(u - u_h)(x) = \psi(x) + R(x), \quad \text{with } \|R\|_{W^{i,\infty}(\Omega_1)} \leq Ch^{p-i+1+\delta} \text{ for } x \in \Omega_1, \quad (4.46)$$

where ψ is a periodic function which can be computed on a reference knotspan, $\delta > 0$ and $W^{i,\infty}$ is the usual Sobolev norm i.e. $\|u\|_{W^{i,\infty}(\Omega)} = \max |D^i u|_{L^\infty}$.

To further describe the function ψ , we first consider the *periodic subspace* of S^h . Suppose x_0, x_1, \dots, x_n are uniformly spaced knot-spans in Ω_0 . Then the periodic subspace of S^h consists of the functions which have the same values at the nodes, as well as the same derivatives up to order α , i.e.

$$S_{per}^h(\Omega_0) := \left\{ \chi : \frac{d^\beta \chi}{dx^\beta}(x_i) = \frac{d^\beta \chi}{dx^\beta}(x_{i+1}) \right\}, \quad \beta = 0, \dots, \alpha, \text{ and } i = 0, \dots, n-1. \quad (4.47)$$

We now consider $P_{per}\rho$, the *periodic projection* of a function ρ over the space $S_{per}^h(\Omega_0)$, which satisfies

$$B(\rho - P_{per}\rho, \chi) = 0, \quad \forall \chi \in S_{per}^h(\Omega_0), \text{ and} \quad (4.48)$$

$$\int_{\Omega} (\rho - P_{per}\rho) \, d\Omega = 0. \quad (4.49)$$

Now we can define ψ as in terms a spline interpolant $I^h[Q]$ of Q , and the periodic projection of the difference between Q and its interpolant as follows:

$$\psi := (Q - I^h[Q]) - P_{per}(Q - I^h[Q]). \quad (4.50)$$

Here $I^h[Q]$ is an interpolant chosen such that $Q - I^h[Q]$ is a periodic function in Ω_0 . For the space of C^α splines, $I^h[u]$ is chosen such that it matches u and the derivatives up to order α at the nodes:

$$\frac{d^\beta I^h[u]}{dx^\beta}(x_i) = \frac{d^\beta u}{dx^\beta}(x_i), \quad \beta = 0, \dots, \alpha, \text{ and } i = 0, \dots, n. \quad (4.51)$$

We emphasize that the superconvergent points, which are the roots of ψ or ψ' , only need to be computed once for a reference element and the obtained coordinates can then be used using scaling and

translations to any uniform mesh. A general procedure for obtaining the superconvergent points for spline approximations of degree p with continuity α , with $p \geq 2\alpha + 1$, on a reference knot-span $[-1, 1]$ is as follows:

1. Consider a knot vector of the form:

$$U := \left\{ \underbrace{-3, \dots, -3}_{p-\alpha \text{ times}}, \underbrace{-1, \dots, -1}_{p-\alpha \text{ times}}, \underbrace{1, \dots, 1}_{p-\alpha \text{ times}}, \underbrace{3, \dots, 3}_{p-\alpha \text{ times}} \right\} \quad (4.52)$$

From this knot vector, we only need to consider the $p + 1$ splines $\phi_1, \phi_2, \dots, \phi_{p+1}$ that have support on $[-1, 1]$ and the local approximation space for this reference element:

$$S^{h,ref} := \text{span}\{\phi_1, \phi_2, \dots, \phi_{p+1}\}. \quad (4.53)$$

2. We build an interpolant $I^h[Q] \in S^{h,ref}$ for $Q(x) = x^{p+1}$ which satisfies (4.51) at the points $x_0 := -1$ and $x_1 := 1$. We note that when $p = 2\alpha + 1$, we already have $p + 1$ constraints for the $p + 1$ unknown coefficients of ϕ_i to be solved. When $p > 2\alpha + 1$, we can interpolate $Q(x)$ at equally spaced points inside $[-1, 1]$ to impose additional constraints.
3. We compute a basis for the periodic subspace $S_{per}^{h,ref}$ of $S^{h,ref}$, which satisfies (4.47) for x_0 and x_1 . It can easily be checked that

$$S_{per}^{h,ref} = \text{span}\left\{ \{\phi_1 + \phi_{p-\alpha}, \dots, \phi_{\alpha+1} + \phi_{p+1}\} \cup \{\phi_{\alpha+2}, \dots, \phi_{p-\alpha-1}\} \right\}, \quad (4.54)$$

which shows that the basis functions in the periodic space are the first $\alpha + 1$ sum of the corresponding splines which have support at the two end-points, together with the ‘‘middle’’ $p - 2\alpha - 1$ splines that do not have support at the endpoints. Clearly, these latter basis functions only appear for $p > 2\alpha + 1$.

4. Now the function ψ can be computed according to (4.50). The superconvergent points for second-order PDEs are then the roots of ψ or rather, for gradient recovery purposes, of ψ' . The coordinates of superconvergent points for several values of p and α are given in Table 4.2.

Tab. 4.2: The superconvergent points for splines of degree p with continuity C^α on interval $[-1, 1]$

p	α	Roots of ψ'
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	$\pm 0.790208564, \pm 0.2800702925$
7	3	$\pm 1, \pm 0.5294113738, 0$

We note that the superconvergent points for $p = 4$ and $\alpha = 1$ are the same as the coordinates for the 4-point Gauss-Legendre quadrature rule. The superconvergent points for the other values of p and α are however different from the Gauss quadrature points.

It is instructive to check numerically that (4.46), or rather its derivative obtained by differentiating both sides, holds. In Figure 4.6, we plot the error in the approximation of u' for $u(x) = x^{p+1}$, highlighting the values of the difference $e(x) := (u - u_h)'(x)$ at the (scaled and translated) coordinates of the superconvergence points. It can be seen that in the interior of the domain the values of $e(x)$ are much smaller compared to the maximum error. This superconvergence property holds for other

functions besides x^{p+1} provided that it is smooth enough (the $p + 1$ degree Taylor expansion exists and is a good approximation to u). However, near the boundaries of the uniform mesh, the superconvergence property is lost (except for B-Splines of degree $p = 4$). This will lead to a lesser effectivity of the error estimator developed in the next subsection, however in general, as it will be seen later, it has a small impact on the overall estimation and refinement scheme. The results of superconvergence

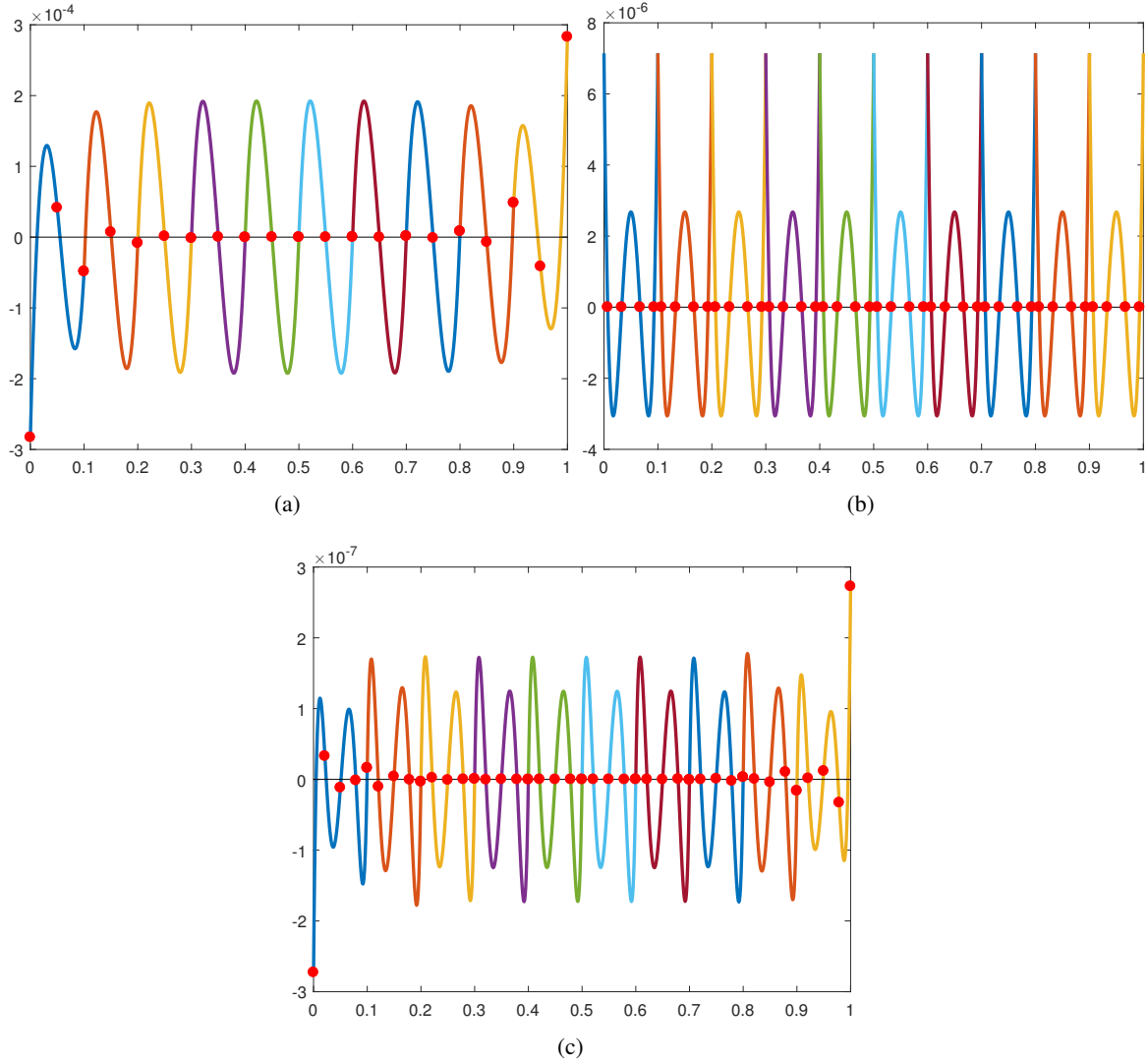


Fig. 4.6: Plots of $(u - u_h)'(x)$ for $u = x^{p+1}$ and uniform meshes with knot-span length $1/10$. The red dots correspond to the coordinates of the superconvergent points, i.e. the roots of ψ' computed on the reference interval and whose coordinates are scaled and translated to each knotspan in the discretization: a) $p = 3$, b) $p = 4$, c) $p = 5$

can also be extended to tensor-product splines in higher dimensions, as in [126, 95], where the superconvergent points are obtained using the tensor-product of the coordinates in one dimension. The points are mapped to the physical domain through the geometry mapping \mathbf{F} , though as is usual in IGA, all the computations are done in the parameter space.

4.5.2 Superconvergent patch recovery

Once the superconvergent points are determined, they can be used to build an enhanced “recovered” solution. In the following, we will particularly focus on constructing a recovered stress field

σ_h^* obtained from the computed stresses σ_h . The procedure is similar to that of the Zienkiewicz-Zhu (ZZ) patch recovery technique [133, 134].

The task at hand is constructing a stress recovery operator which outputs a better (i.e. superconvergent) approximation to the solution. In particular, we want to compute $G[\sigma_h] = \sigma_h^*$, where the recovered stress σ_h^* is obtained by fitting a polynomial spline of higher degree through the superconvergent points on local subdomains. Let Ω_k , $k = 1, \dots, n$ be a set of non-overlapping patches such that $\bigcup_{k=1}^n \Omega_k = \Omega$, where n is the number of patches in the domain. Furthermore, let $\mathbf{x}_{i,k}^*$, with $i = 1, \dots, N_k$ be the set of the N_k superconvergent points on the patch Ω_k . Then we choose the operator $G[\sigma_h]$ such that:

$$G[\sigma_h](\mathbf{x}) = \sum_{i=1}^{M_k} \phi_{i,k}^*(\mathbf{x}) \mathbf{C}_{i,k}^*, \text{ for } \mathbf{x} \in \Omega_k \text{ with } M_k \geq N_k \text{ and} \quad (4.55)$$

$$G[\sigma_h](\mathbf{x}_{i,k}^*) = \sigma_h(\mathbf{x}_{i,k}^*), \text{ for } i = 1, \dots, N_k, k = 1, \dots, n. \quad (4.56)$$

Here $\phi_{i,k}^*$ are B-Splines of degree $p^* \geq p$ and continuity $\alpha^* \geq \alpha$ with support on the local patches Ω_k . This gives rise to n linear systems of the form $\mathbf{A}^{(k)} \mathbf{C}_k^* = \mathbf{b}_k$, where

$$\mathbf{A}_{ij}^{(k)} = \phi_{j,k}^*(\mathbf{x}_{i,k}^*), \quad i = 1, \dots, N_k, j = 1, \dots, M_k, \quad (4.57)$$

and \mathbf{b}_k is a $N_k \times e$ matrix with $e = 3$ in two dimensions ($d = 2$) and $e = 6$ in three dimensions ($d = 3$) containing the stress components of σ_h evaluated at the points $\mathbf{x}_{i,k}^*$. These linear systems are overdetermined if $N_k > M_k$ and can be solved in a least-squares sense. Nevertheless, it is more computationally efficient to select Ω_k and $\phi_{i,k}^*$ such that $N_k = M_k$. A practical way to accomplish this in the context of hierarchical meshes is as follows:

- (i) We perform one level of cross-insertion on all the elements on the initial coarse mesh and let each subdomain Ω_k to be the set of 2^d elements which have a common parent. Subsequently we refine all the elements in a subdomain together, i.e. we mark all the elements in a given Ω_k for refinement and split it into 2^d subdomains which are used for patch recovery in the next step. This ensures that the number of elements in the mesh is divisible by 2^d and that every element belongs to a patch of 2^d elements which have the same refinement level.
- (ii) The polynomial degree and continuity of the recovery splines $\phi_{i,k}$ is chosen to match the number of superconvergent points in each recovery subdomain. This ensures that we have the same number of unknown coefficients as we have equations for the fitting of the recovered solution. The values of p^* and α^* for different values of p and α along with the size of the matrix $\mathbf{A}^{(k)}$ needed for computing $G[\sigma_h]$ on each recovery subdomain are given in Table 4.3. We note that for $p = 4$ and $p = 5$, other values of p^* and α^* are possible which result in a basis $\{\phi_{i,k}^*\}$ of the same dimension.

Tab. 4.3: Values of p^* and α^* for given p and α with the size of the resulting matrix $A^{(k)}$. The last column shows the location of the superconvergent points from Table 4.2 in one dimension ($d = 1$).

p	α	p^*	α^*	Size of $A^{(k)}$	Superconvergent points
3	1	3	2	$5^d \times 5^d$	
4	1	5	3	$8^d \times 8^d$	
5	2	7	6	$9^d \times 9^d$	
6	2	6	5	$8^d \times 8^d$	
7	3	7	6	$9^d \times 9^d$	

A recovery subdomain and the superconvergent points in the parameter and physical space are shown in Figure 4.7. In general, we choose these subdomains such that the elements inside share the same parent element, which ensures that they are non-overlapping and the recovery cost is minimized. Since performing the recovery is completely local, the computational cost for computing $G[\sigma_h]$ is typically less than 10% than that of computing u_h and σ_h . Moreover, this operation can be trivially parallelized since the recovery on each subdomain is computed independently of the others.

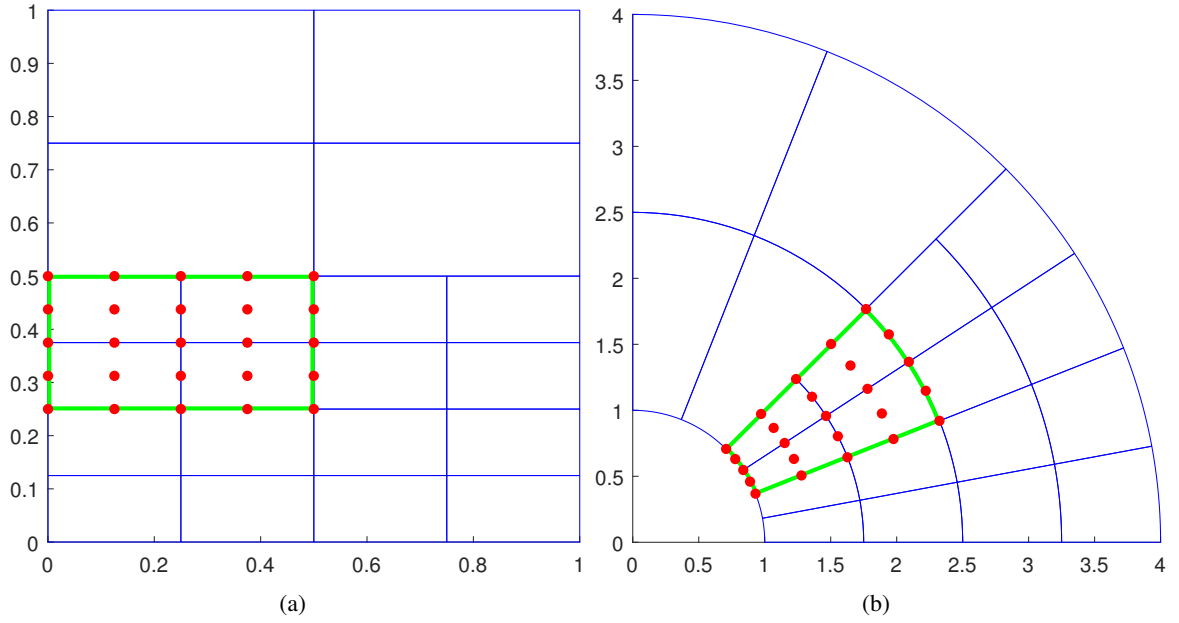


Fig. 4.7: A recovery subdomain on an annular shaped domain together with the superconvergent points for $p = 3$: a) Parameter space b) Physical space

Once $G[\sigma_h]$ has been computed, we estimate the error in the energy norm by:

$$\eta(\sigma_h; \Omega) := \left(\int_{\Omega} (G[\sigma_h] - \sigma_h)^T D^{-1} (G[\sigma_h] - \sigma_h) d\Omega \right)^{1/2}, \quad (4.58)$$

where D^{-1} is the compliance matrix. We are now interested to analyze the asymptotic behavior of $G[\sigma_h]$. It is known [7, 81] that if certain properties, which include consistency, locality, boundedness and linearity, are satisfied, then the recovery operator satisfies the *superapproximation property*:

$$e(\sigma_h; \Omega) := \left(\int_{\Omega} (\sigma - \sigma_h)^T D^{-1} (\sigma - \sigma_h) d\Omega \right)^{1/2} \leq Ch^{p+\delta}, \quad (4.59)$$

where δ is the same as in (4.46) and σ is the exact stress. If $\delta > 0$, then the error estimator $\eta(\sigma_h; \Omega)$ is asymptotically exact, i.e.

$$\theta(\sigma_h; \Omega) := \frac{\eta(\sigma_h; \Omega)}{e(\sigma_h; \Omega)} \rightarrow 1 \text{ as } h \rightarrow 0, \quad (4.60)$$

where $\theta(\sigma_h; \Omega)$ is the *effectivity index* of the error estimator $\eta(\sigma_h; \Omega)$. We note that asymptotic exactness can typically be achieved only under some rather stringent assumptions on the regularity of the solution and uniformity of the meshes. However, in many cases constructing a smoother, higher-order approximation to the stress field is enough to drive the adaptivity, though usually the error is over-estimated and finer meshes than needed are employed. This can be seen for example in [98, 100] where over-refinement is observed in some of the numerical examples considered there.

4.5.3 Marking algorithm

It is important to consider for the hierarchical refinement process not just the accuracy and robustness of the error estimator, but also the choice of the marking algorithm. In general, there is a trade-off between the number of refinement steps required to reach a certain (estimated) accuracy and the number of elements in the final mesh. In other words, refining in small increments results in “optimal” meshes that have as few as possible elements while refining more elements at each refinement, step results in fewer overall refinement steps but less optimal meshes. This is particularly true in case of problems in singularities, as for coarse meshes the presence of a singularity results in significant errors even some distance away in the domain (the so-called “pollution errors”); however these errors become less significant as the area around the singularity is refined.

Given a particular error tolerance (in percentage or as a relative error), a simple refinement strategy is to mark for refinement all the elements where the error exceeds some threshold (which can be determined as a percentage of the maximum estimated error). This is referred to as the “absolute threshold” marking strategy in [59]. Since this method does not take into account the overall distribution of the error, it is preferable to use the “Dörfler marking” strategy [53], where the elements with the largest contribution to the total error are selected. In particular we want to choose the set of marked elements \mathcal{M} of minimal cardinality such that given a parameter $\theta \in (0, 1]$:

$$\eta(\theta_h; \Omega_{\mathcal{M}}) \leq \theta \eta(\theta_h; \Omega), \quad (4.61)$$

where $\Omega_{\mathcal{M}} \subseteq \Omega$ is the region in the domain that is marked for refinement. We note that $\theta = 1$ results in uniform refinement, while $\theta \ll 1$ results in smaller refinement steps. In practice, $\theta = 0.75$ for problems with smooth solutions and $\theta = 0.5$ for problems with singularities provides a reasonable compromise between the number of refinement steps and the optimality of the meshes.

Chapter 5

Polynomial spline over hierarchical T-meshes (PHT-spline)

Although T-splines remove the burden of using the traditional tensor-product B-spline and NURBS where it is possible to do local refinement in the sharp region, without propagating from the sharp region along horizontal and vertical directions to maintain the tensor-product mesh structure and the superfluous control points which is a big burden to geometric modeling. But using T-splines for local refinement is not so simple. In the T-spline theory, the local refinement is not independent on the structure of the mesh, and its complexity is uncertain [113]. On the other hand, since the basis functions do not form a partition of unity and T-splines are rational which leads to more complicated situations in geometric operations. Deng [48] introduced polynomial spline over T-meshes where it represents the spline function on every cell to be a tensor-product polynomial to achieve the specified smoothness across the common edges and they differentiate it from T-splines as *splines over T-meshes*. It has several advantages over T-meshes. The following advantages are:

- Local refinement becomes simple.
- The splines are piecewise polynomial, instead of rational functions.
- There is a hierarchical structure of this kind of spline which can be adopted to adaptive local refinement during analysis.
- During the refinement operation, the new basis functions do not interfere with the neighboring basis functions and the existing basis functions only change on the refined elements (their support shrinks to insure linear independence).

5.1 1D Hierarchical T-meshes

For a simple case, we at first consider hierarchical refinements of tensor-product meshes. We consider the initial sets of vertices X^i corresponding to each spatial direction in the parameter space $\Omega_{par} := [0, 1]^d$:

$$U^i = \{u_0^i, u_1^i, \dots, u_{n_i}^i\}, \quad i = 1, \dots, d, \quad (5.1)$$

where $u_0^i := 0 < u_1^i < \dots < u_{n_i-1}^i < u_{n_i}^i := 1$ and here d denotes the space dimension and n_i denotes the number of control points in each parametric direction. By setting U^i the initial tensor product mesh on the initial mesh. For example for 1D mesh let's consider $d = 1$, we can write:

$$\mathbb{T}_0 = \{E_{0,k} = [u_{k-1}^1, u_k^1], \quad k = 1, \dots, n\}, \quad (5.2)$$

where $E_{0,k}$ denotes an element in initial mesh level of 0 and index k can be defined as $k := (k_2 - 1)n_1 + k_1$. From a given mesh refinement level of l at \mathbb{T}_l , we select a subset $\mathcal{M} \subseteq \mathbb{T}_l$ of elements that are marked for refinement and subdivide each element $E_{l,k}$ from \mathcal{M} . The further detail of the hierarchy mesh will be discussed in the next chapter.

5.2 1D PHT-spline

The main idea of the higher-order PHT-spline using the hierarchy from lower level to higher level by modification of several basis functions from level k and addition of new basis functions corresponding to the new basis vertices. We will start from the simple 1D example for degree $p = 3$ and extend this example for a higher degree of the PHT-spline basis function ($p > 3$).

5.2.1 1D PHT-spline (degree, $p=3$)

As a first step, we should construct a parametric domain of the solution field. The mapping from the parameter space to physical space Fig. 5.1 can be defined as Eq. 4.12

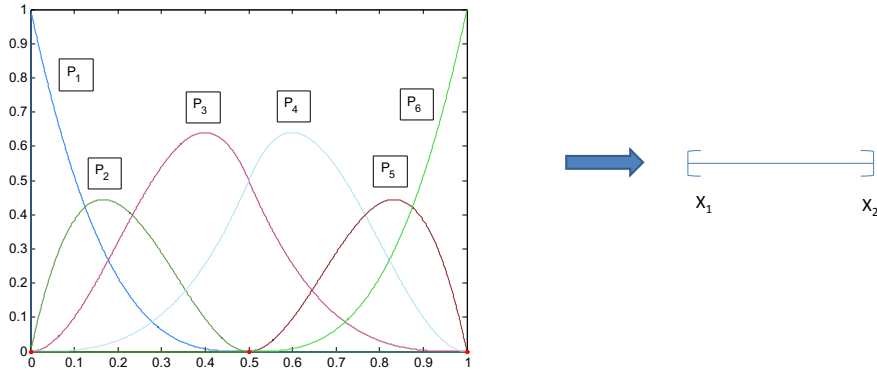


Fig. 5.1: Mapping 1D B-splines basis function.

$$F(\mathbf{u}) = \sum_{i=1}^N C_i P_i(\mathbf{u}), \quad (5.3)$$

where C_i is the PHT-control points and $P_i(\mathbf{u})$ is the PHT-basis function. This PHT-basis function can be represented in Bézier representation

$$P_i(\mathbf{u}) = \sum_{j=1}^{N+1} D_j^n \hat{B}_j^n(\mathbf{u}), \quad (5.4)$$

where $\hat{B}_j^n(\mathbf{u})$ is Bernstein basis (Eq. 2.31) scaled to the n -th element and D_j^n Bézier ordinates (control points) on the n -th element. This D_j^n computed by Bézier extraction or De Casteljau algorithm [102].

“New” mapping from the parameter space to physical space according to Eq. 4.1 can be defined,

$$\hat{F}(\mathbf{u}) = \sum_{i=1}^N C_i \hat{P}_i(\mathbf{u}) + C_{i+1} \hat{P}_{i+1}(\mathbf{u}) \dots + C_N \hat{P}_N(\mathbf{u}) = F(\mathbf{u}) = x \quad (5.5)$$

if x^* is a new basis vertex then we can set

$$\begin{aligned} \hat{F}(x^*) &= F(x^*), \text{ and} \\ \hat{F}'(x^*) &= F'(x^*) \end{aligned} \quad (5.6)$$

This x^* new basis vertex can be computed from the previous geometric mapping by,

$$\begin{aligned} C_7 \hat{P}_7(x^*) + C_8 \hat{P}_8(x^*) &= F(x^*) \\ C_7 \hat{P}'_7(x^*) + C_8 \hat{P}'_8(x^*) &= F'(x^*) \end{aligned} \quad (5.7)$$

- B-spline basis: Start with a knot vector and insert a knot into a knot vector and modify the B-spline basis functions for the new knot using the Bézier extraction. In Fig. 5.2 it shows for degree $p = 3$ the initial knot vector for C^1 continuity $\{0, 0, 0, 0, 0.5, 0.5, 1, 1, 1, 1\}$ where C^{p-k} interior knots and two 1D elements.

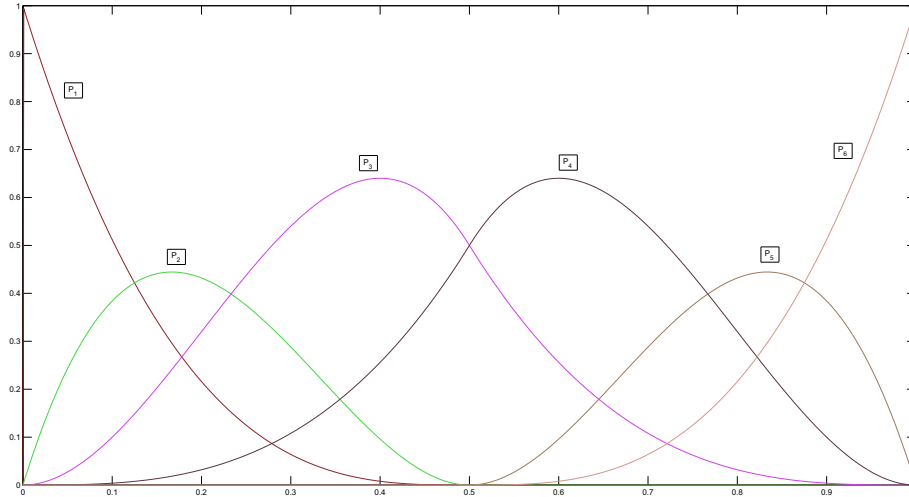


Fig. 5.2: B-spline basis function.

- Refinement of 1 element: For the refinement of the 1D element we will take the second element for refinement. Here in Fig. 5.3 shows the Bernstein basis where each element has $p + 1$ basis functions support.
- Apply De Casteljaun algorithm [102] from Eq. 6.7 to recompute the Bézier ordinates.
- Zero out the Bézier ordinates like in Fig. 6.6 from the basis functions that have support in the neighbour element (shorten the tail of the neighbouring element basis function P_3 and P_5) in Fig.5.4.
- Add the new basis functions after knot insertion in $x^* = 0.75$, we can observe that 2 new basis functions appear P_7 and P_8 which have support only on the refined element in the knot $x^* = 0.75$ from Fig. 5.4 and minor modification is needed for the neighbouring element basis functions.

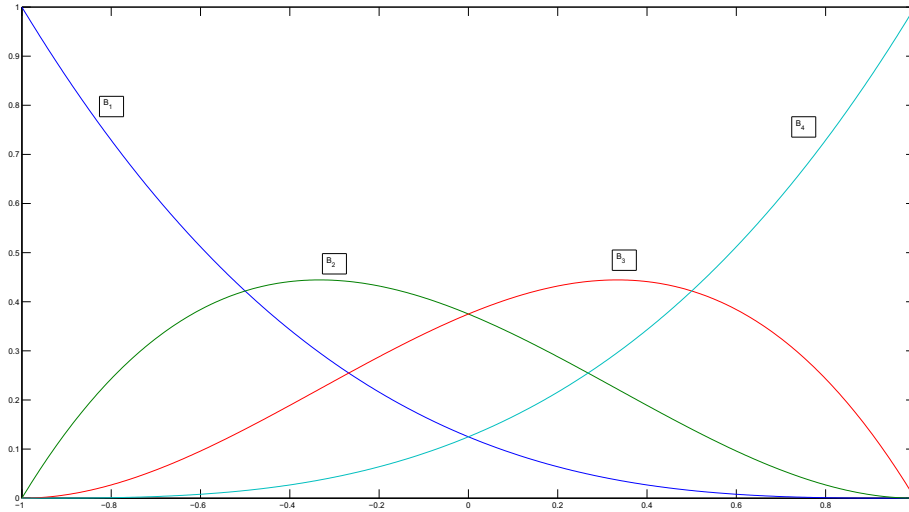


Fig. 5.3: Bernstein basis function for a 1D elements $[u] = [-1, 1]$ for degree $p = 3$.

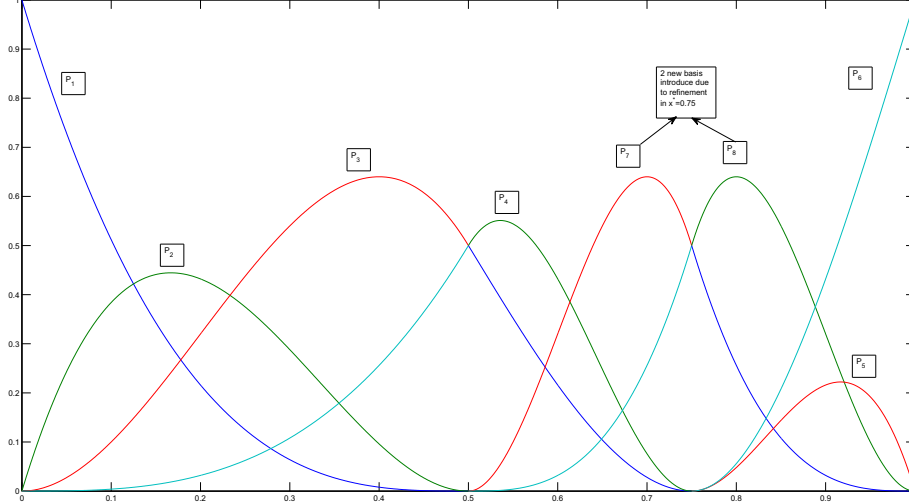


Fig. 5.4: PHT-spline basis function for 1D two elements after knot insertion at knot $x^* = 0.75$.

5.2.2 1D extension of PHT-spline (degree $p > 3$)

In this subsection, we will discuss the extension of the 1D PHT-basis function for higher degree $p > 3$ with higher continuity α , where $p \geq 2\alpha + 1$ from Eq. (6.2) and C^α is the continuity order of the PHT-spline basis. Here we consider degree $p = 5$ and the continuity order of the basis function becomes

$$\alpha = \frac{p-1}{2} = \frac{5-1}{2} = 2 \quad (5.8)$$

So we get higher continuity C^2 of the PHT-spline basis function.

- Again we start for the definition of B-spline basis with degree $p = 5$. The initial knot vector for C^2 continuity is $\{0, 0, 0, 0, 0, 0, 0.5, 0.5, 0.5, 1, 1, 1, 1, 1\}$ where $C^{p-\alpha} = 3$ interior knots and $C^{p+1} = 6$ open knots before and after the interior knots for two 1D elements. So we get $N - p - 1 = 15 - 5 - 1 = 9$ B-splines basis functions from Fig. 5.5, where N =number of the knots in a knot vector.
- Then we define the PHT-spline basis function using Bernstein polynomial and Bézier extraction operator in Fig. 5.6.

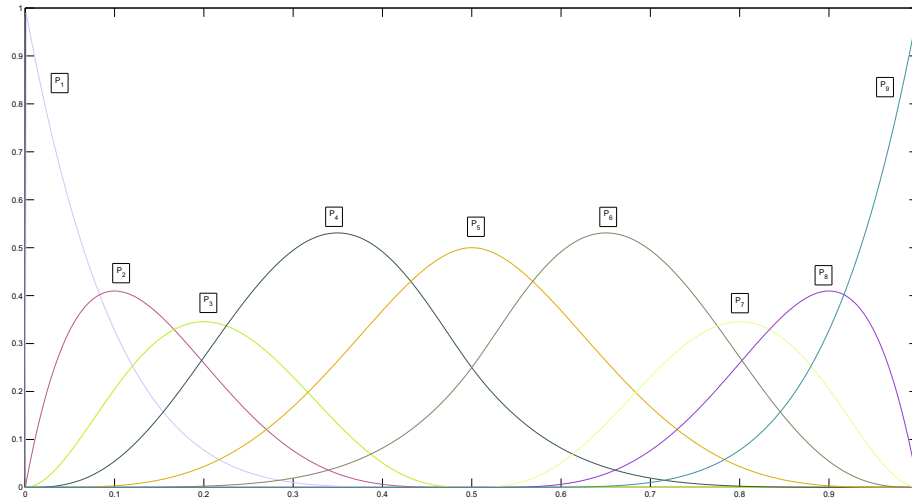


Fig. 5.5: B-spline basis function for $p = 5$ and C^2 .



Fig. 5.6: Initial PHT-spline basis function for two 1D elements for degree $p = 5$ using Bernstein polynomial and Bézier extraction operator.

- For refinement, we can again take the second 1D element and insert a knot at $x^* = 0.75$ at the knot vector.
- Apply De Casteljaun algorithm [102] from Eq. 6.7 to recompute the Bézier ordinates.
- Zero out the Bézier ordinates like in Fig. 6.6 from the basis functions that have support in the neighbour element (shorten the tail of the neighbouring element basis function P_5, P_6, P_7 and P_8) in Fig. 5.7
- Add the new basis functions after knot insertion in $x^* = 0.75$, we get 4 new basis functions P_{10}, P_{11}, P_{12} and P_{13} which have support only on the refined element in the knot $x^* = 0.75$ from Fig. 5.7 and minor modifications are needed for the neighbouring element basis functions.

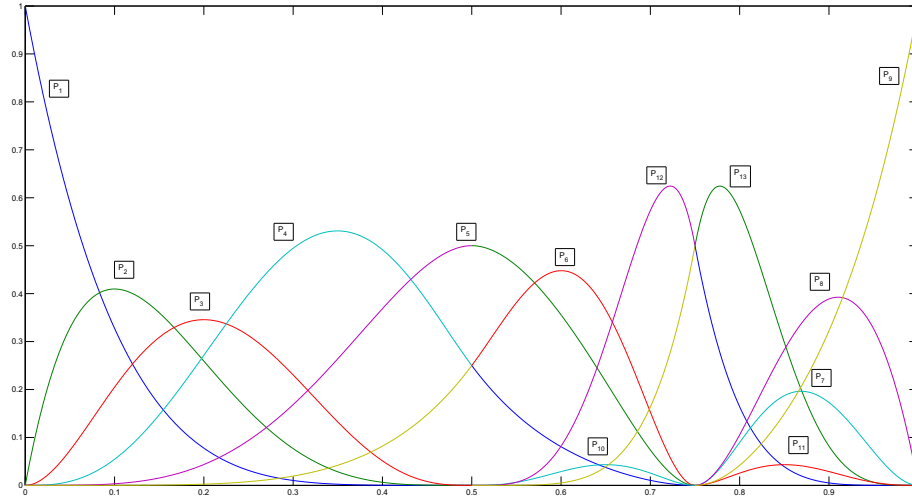


Fig. 5.7: PHT-spline basis function refinement in the second 1D element at $x^* = 0.75$ for $p = 5$ and C^2

5.3 Numerical Example 1D GIFT PHT-spline

Here we will demonstrate a simple 1D example using PHT-spline within the GIFT method. This 1D test case [14] solves the following source problem define on the domain $[0, 1]$:

$$\begin{aligned} -u''(x) + a_1 u'(x) + a_0 u(x) &= f(x) \quad x \in [0, 1] \\ u(0) &= 0, u(1) = 0; \end{aligned} \quad (5.9)$$

where $a_0 = 1$ and $a_1 = 1$ with $f(x) = (1 + 4\pi^2)\sin(2\pi x) - 2\pi\cos(2\pi x)$ and the exact solution is:

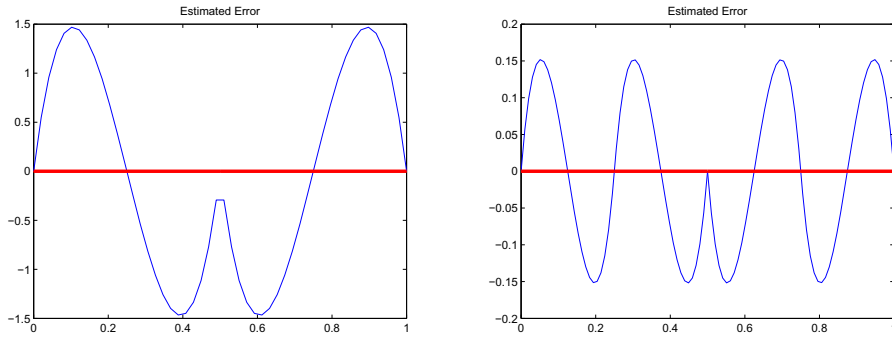
$$u(x) = \sin(2\pi x) \quad (5.10)$$

This problem is numerically solved using the GIFT method outlined in the previous chapter. Here for degree $p = 3$ we used $p + 2$ gauss quadrature and the estimated error plot in different refinement step is shown in Fig 5.8.

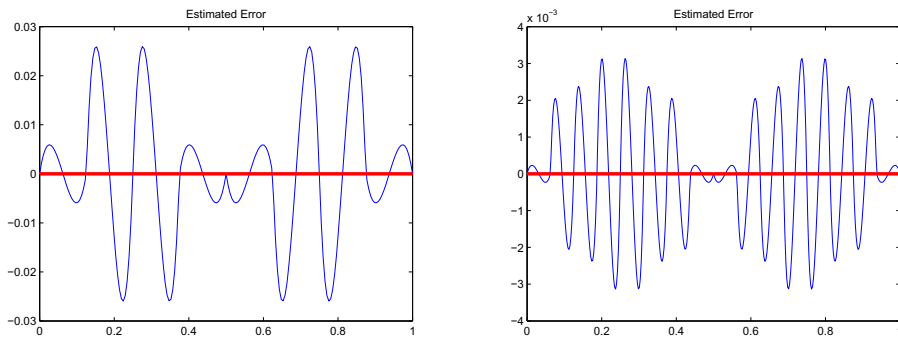
For higher degree $p = 7$ PHT-spline, it converges in 5 steps with fewer dofs to the target relative error $= 10^{-10}$. For higher degree ($p = 7$) PHT-spline, the estimated error is shown in Fig 5.9.

In Fig. 5.10 the error in the L^2 -norm and H^1 -norm versus the number of dofs of the mesh is plotted. If we compare the L^2 -error norm and H^1 -error norms it can be seen that for higher-order with greater continuity we get the higher accuracy and convergence rate.

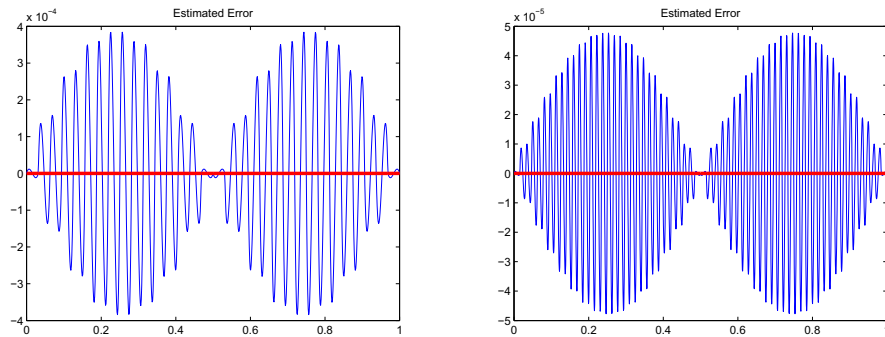
In Fig. 5.11 it has been demonstrated the estimated error by recovery based error estimator with the exact error in H^1 seminorm convergence plot for degree $p = 7$. The error estimator here shows the same convergence for higher degree PHT-spline comparing with the exact error.



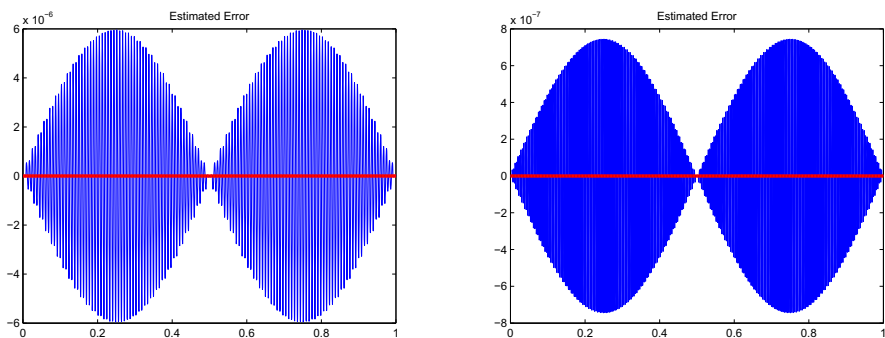
(a) Estimated error for adaptive refinement step 1. (b) Estimated error for adaptive refinement step 2.



(c) Estimated error for adaptive refinement step 3. (d) Estimated error for adaptive refinement step 4.

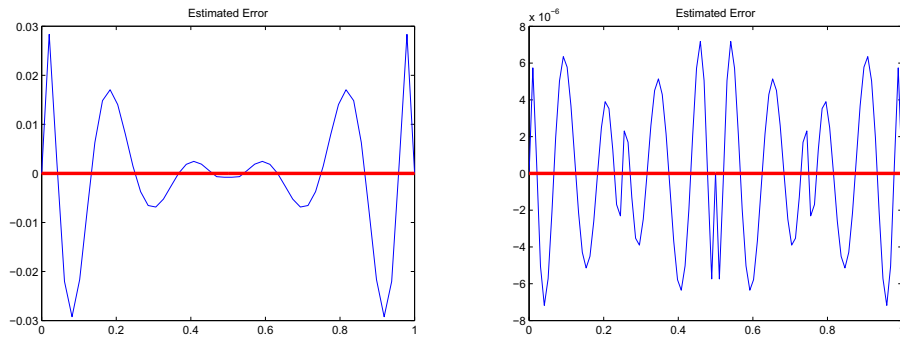


(e) Estimated error for adaptive refinement step 5. (f) Estimated error for adaptive refinement step 6.

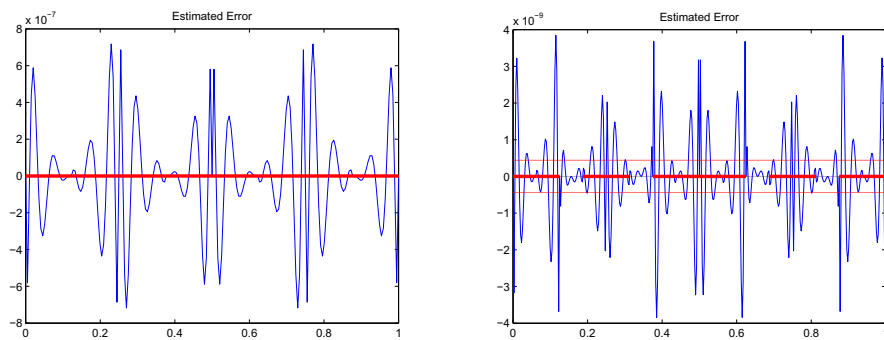


(g) Estimated error for adaptive refinement step 7. (h) Estimated error for adaptive refinement step 8.

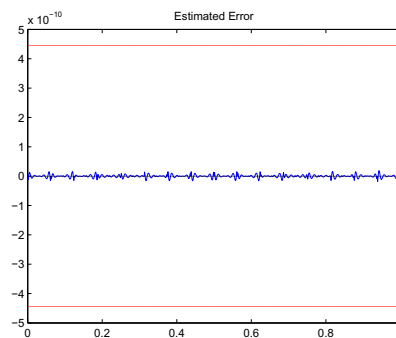
Fig. 5.8: Estimated recovery based error in different refinement steps for degree $p = 3$. Here red lines are $E_{rel} * \sqrt{H^1}$ (where E_{rel} =target relative error and H^1 = energy norm global).



(a) Estimated error for adaptive refinement step 1. (b) Estimated error for adaptive refinement step 2.



(c) Estimated error for adaptive refinement step 3. (d) Estimated error for adaptive refinement step 4.



(e) Estimated error for adaptive refinement step 5.

Fig. 5.9: Estimated recovery based error in different refinement steps for degree, $p = 7$. Here red lines are $E_{rel} * \sqrt{H^1}$ (where E_{rel} =target relative error and H^1 = energy norm global).

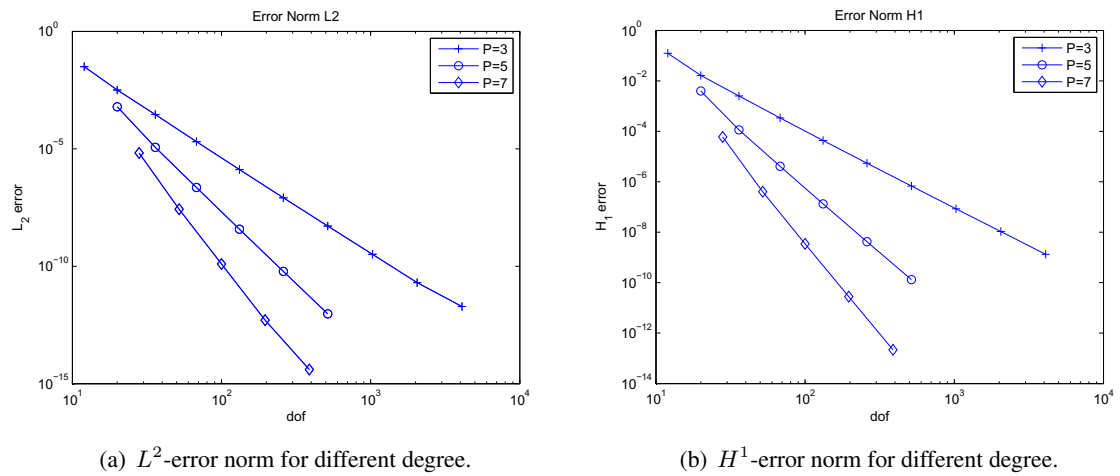


Fig. 5.10: L^2 -error and H^1 -error norms for 1D PHT-splines for different degree.

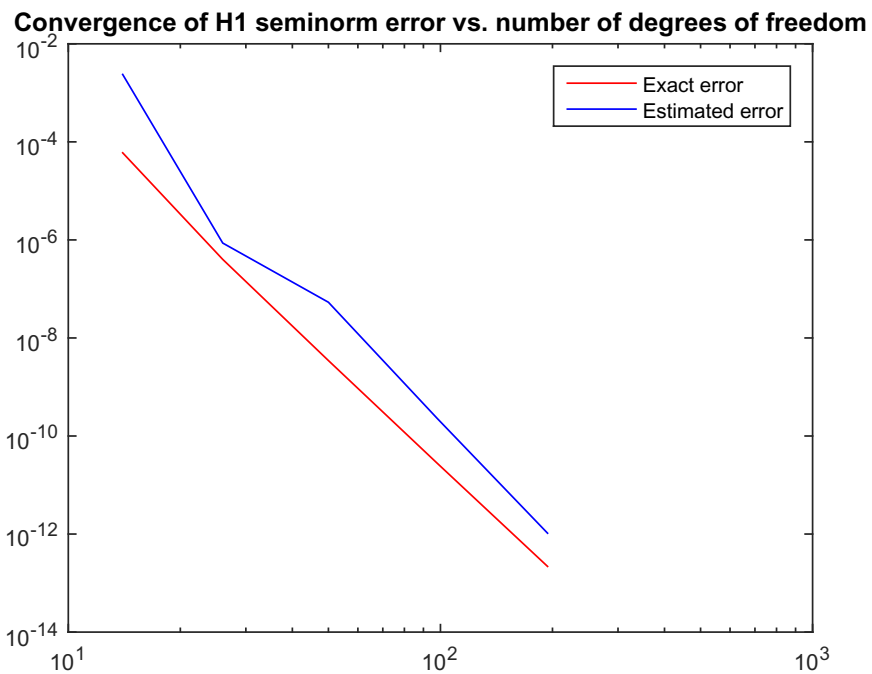


Fig. 5.11: Estimated error Vs. Exact error H^1 -seminorm convergence plot.

Chapter 6

Two Dimensional Polynomial Splines Over Hierarchical T-meshes

In this chapter, we will discuss 2D PHT-splines which are based on hierarchical 2D T-meshes. At first, we will show the Tree structures, hierarchical properties, and general dimensional formula of the PHT-spline. Then we will focus on the degree $p = 3$ polynomial spline with C^1 continuity and extend our implementation to higher-order ($p > 3$) and higher continuity $C^\alpha \subseteq C^1$ where α is the continuity order of the PHT-spline basis.

6.1 2D T-mesh

A T-mesh is a mesh based on rectangular grids that allow T-junctions [114]. In T-meshes, the end points of each grid line must lie on two other grid lines, and each cell or facet in the grid is formed by a quadrilateral. Fig. 6.1 illustrates a typical T-mesh. A vertex of the T-mesh is assigned to each grid point. If a vertex is inside the domain, it is called an interior vertex (v_1, v_2, \dots, v_i , where i = number of interior vertices). Otherwise, it is called a boundary vertex (b_1, b_2, \dots, b_j , where j = number of boundary vertices).

6.2 Tree structure

A considerable advantage of hierarchical refinement is its straightforward and efficient implementation through the quadtree and octree structure [105], which provide a natural way to decompose and organize spatial data according to different levels of complexity and offer fast access to relevant parts of a dataset [39, 21]. The quadtree concept is shown in Fig. 6.2 illustrates the analogy between an adaptive hierarchical quadrilateral mesh and the two-dimensional tree. Each node or leaf of the tree holds the information regarding the placement of the element in the parameter space and the information about the local basis functions, including the element to node (IEN) connectivities. Additionally, each node or leaf is equipped with pointers that connect it with all direct neighbors (see Fig. 6.2) so that “horizontal” neighboring elements can be located with little computational effort. More details on implementation aspects and related algorithms of tree structure can be found in [107].

6.3 2D Hierarchical T-meshes

A 2D hierarchical T-mesh can be considered as a special form of T-mesh, which has a natural level structure. We can define the 2D hierarchical T-mesh structure from the extension of 1D T-mesh

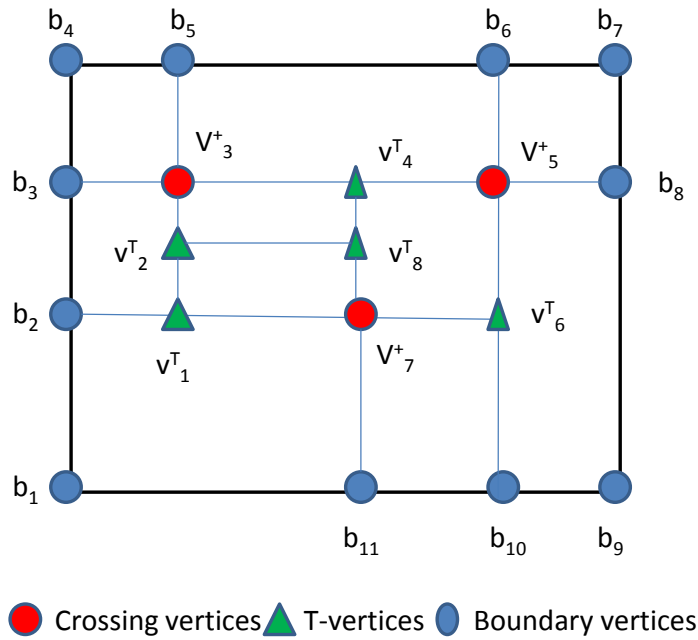


Fig. 6.1: An illustration of boundary, crossing and T-junctional vertices: b_j are boundary vertices, v_i^+ are crossing vertices and v_i^T are T-junctional vertices.

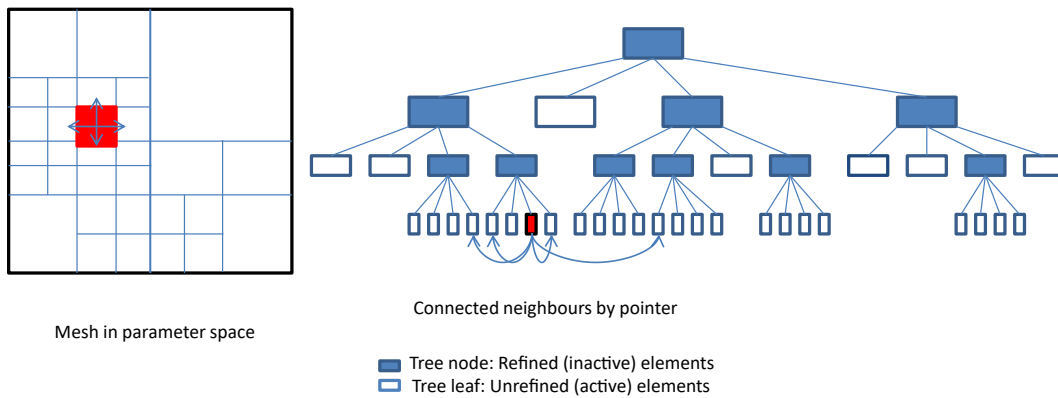


Fig. 6.2: Quadtree example illustrating the hierarchical data organization of part of an adaptive mesh. The neighboring relations within each hierarchical level are established by pointers, which are shown here for one element of the finest level (in red color).

structure of the previous chapter 5.1. From Eq. 5.1 we can define for $d = 2$ as an extension of Eq. 5.2:

$$\mathbb{T}_0 = \{E_{0,k} = [u_{k_1-1}^1, u_{k_1}^1] \times [u_{k_2-1}^2, u_{k_2}^2], \quad k_1 = 1, \dots, n_1, k_2 = 1, \dots, n_2\} \quad (6.1)$$

For each element at initial level $E_{0,k}$ we can subdivide each element $E_{l,k}$ from \mathcal{M} into 2^d children elements, $E_{l+1,j_1}, \dots, E_{l+1,j_{2^d}}$. In the T-spline mesh, we differentiate between crossing vertices, which are vertices in the interior of the domain that is shared by 2^d elements, boundary vertices, which are located on the boundary of the parameter space and are shared by 2^{d-1} elements, and T-junctions which are neither crossing vertices nor boundary vertices. The first two types of vertices are called basis vertices because they are associated with particular basis functions in the mesh. An illustration of the meshes at different refinement levels and the corresponding vertices of different types is given in Fig. 6.3. If a level- k mesh is given, then the level $(k + 1)$ mesh is obtained by subdividing some of the cells in level- k . Each cell is subdivided into four subcells by connecting the middle points of the opposite edges in the cell.

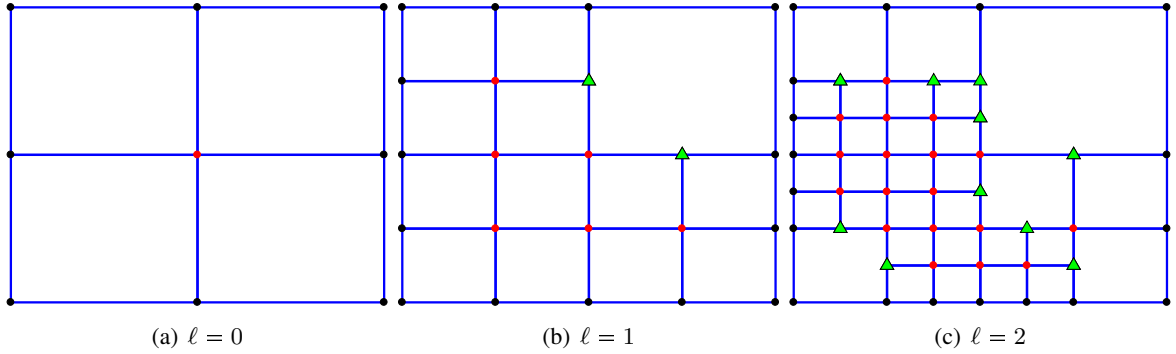


Fig. 6.3: Example showing boundary, crossing and T-junction vertices on: a) the initial mesh, b) after the first refinement, c) after the second refinement. The black dots denote boundary vertices, the red dots represent crossing vertices and the green triangles are T-junctions.

6.4 A dimension formula

As pointed out in [48], polynomial spline functions over T-meshes have advantages such as the simplification of local refinement strategy, the use of piecewise polynomials, and the utilization of hierarchical structures. Herein two important issues are addressed to be the dimension formula of spline function spaces and the construction of basis functions of splines over hierarchical T-meshes. Let \mathcal{T} be the T-mesh, \mathcal{H} be the cells in \mathcal{T} , $\Omega \in \mathbb{R}^2$ be the region occupied by \mathcal{T} . Then we write:

$$\mathcal{S}(m, n, \alpha, \beta, \mathcal{T}) = \{s(x, y) \in C^{\alpha, \beta}(\Omega) | s(x, y)|_{\phi} \in \mathbb{P}_{mn} \quad \forall \quad \phi \in \mathcal{H}\}, \quad (6.2)$$

where space, \mathbb{P}_{mn} , consists of all the bi-degree (m, n) polynomial and the space, $C^{\alpha, \beta}$, consist of all the continuously bivariate functions up to order α in the x -direction and order β in the y -direction. The dimensional formula of the spline space $\mathcal{S}(m, n, \alpha, \beta, \mathcal{T})$ when $m \geq 2\alpha + 1$ and $n \geq 2\beta + 1$ has already been proved in [48]. For a C^1 -continuous cubic spline, where every interior knots of multiplicity two, the evaluation of the dimension formula is reduced to the following form

$$\dim \mathcal{S}(3, 3, 1, 1, \mathcal{T}) = 4(V^b + V^+), \quad (6.3)$$

where V^b , V^+ are boundary vertices, and interior crossing vertices, respectively. For example, we have $V^b = 11$ and $V^+ = 3$ as illustrated in Fig. 6.1. The dimension formula is

$$\dim \mathcal{S}(3, 3, 1, 1, \mathcal{T}) = 4(11 + 3) = 56 \quad (6.4)$$

Eq. (6.4) shows the number of basis functions corresponding to boundary and crossing vertices which

need to be constructed. It is also implied that each boundary vertex or each crossing vertex is connected by four basis functions. After obtaining the dimensional formula, the task in the next section will show how to construct basis functions of splines over hierarchical T-meshes.

6.5 Basis function representation and refinement in 2D

In this subsection, we describe the representation of the basis functions in the parameter space and how the basis functions are modified in the process of refinement. We proceed inductively, assuming that a T-spline mesh \mathbb{T}_ℓ is given, and we select a particular element $E_{\ell,k}$ for refinement. We call the procedure of refining $E_{\ell,k}$ by splitting it into 4 sub-elements “cross insertion”. Different from other hierarchical spline structures and similar to the standard finite element schemes, the elements of the mesh are being refined and the new basis functions are calculated by the cross-insertion algorithm described below.

As usual, we use the *Bézier representation* of the basis functions. We define a linear mapping $\hat{\mathbf{F}}^{(E)}$ from a *reference element* $[-1, 1] \times [-1, 1]$ to an element $E \in \mathbb{T}_\ell$. Each basis function $\hat{\phi}$ of polynomial degree p on an element E , is represented as a linear combination of $(p + 1)^2$ Bernstein basis functions composed with the inverse of the mapping from the parameter space to the physical space, i.e.

$$\hat{\phi}|_E(u^{(1)}, u^{(2)}) = \sum_{i=1}^{p+1} \sum_{j=1}^{p+1} C_{ij}^{(E)} \hat{B}_{i,j} \circ \hat{\mathbf{F}}^{(E)-1}(u^{(1)}, u^{(2)}), \quad (6.5)$$

where $\hat{B}_{i,j}(\xi^{(1)}, \xi^{(2)}) = B_i(\xi^{(1)})B_j(\xi^{(2)})$ is a tensor product of Bernstein polynomials defined on the reference interval $[-1, 1]$, i.e.

$$B_i(\xi) = \frac{1}{2^p} \binom{p}{i-1} (1-\xi)^{p-i+1} (1+\xi)^{i-1}, \quad i = 1, 2, \dots, p+1. \quad (6.6)$$

The procedure for computing the (scalar) Bézier coefficients C_{ij} of the basis functions on the child elements $E_{\ell+1,j_1}, E_{\ell+1,j_2}, E_{\ell+1,j_3}$ and $E_{\ell+1,j_4}$ is described in the following subsections.

6.5.1 Blossoming with De Casteljau’s algorithm

The first step is to compute the Bézier representation of the original basis functions ϕ_j on the elements $E_{\ell+1,j_1}, E_{\ell+1,j_2}, E_{\ell+1,j_3}$ and $E_{\ell+1,j_4}$. This subdivision is realized by applying De Casteljau’s algorithm [55] for evaluating Bézier polynomials in each spatial direction. In one dimension, the algorithm (see Algorithm 1) takes as input $p + 1$ Bézier coefficients $b_1^{(0)}, b_2^{(0)}, \dots, b_{p+1}^{(0)}$ and computes the coefficients $b_1^{(0)}, b_1^{(1)}, \dots, b_1^{(p)}$ and $b_1^{(p)}, b_2^{(p-1)}, b_{p+1}^{(0)}$ according to the recurrence relation:

$$b_i^{(j)} := (b_i^{(j-1)} + b_{i+1}^{(j-1)})/2, \quad i = 1, \dots, p+1-j, \quad j = 1, \dots, p. \quad (6.7)$$

The two sets of coefficients $b_1^{(0)}, b_1^{(1)}, \dots, b_1^{(p)}$ and $b_1^{(p)}, b_2^{(p-1)}, \dots, b_{p+1}^{(0)}$ split the original Bézier polynomial defined on the segment $[u_k, u_{k+1}]$ into two segments $[u_k, (u_k + u_{k+1})/2]$ and $[(u_k + u_{k+1})/2, u_{k+1}]$. In two dimensions, we arrange the $(p + 1)^2$ coefficients $C_{ij}^{(E)}$ corresponding to a single basis function ϕ 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 $4(p+1) \times (p+1)$ subcells, each containing the Bézier representation of the basis function ϕ on the 4 child elements.

Algorithm 1: De Casteljau Algorithm in 1D

Input : Bézier coefficients $b_1^{(0)}, b_2^{(0)}, \dots, b_{p+1}^{(0)}$

Output: Bézier coefficients $b_1^{(0)}, b_1^{(1)}, \dots, b_1^{(p)}$ and $b_2^{(p-1)}, \dots, b_{p+1}^{(0)}$

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

6.5.2 Truncation by zeroing out Bézier coefficients

To insure the linear independence of the basis and better sparsity in the resulting linear system, a truncation procedure is employed for the basis functions on the elements that are being refined. We assume 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 is split into 9 regions, as shown in Figure 6.4. Then, 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. We note that the new basis vertices can correspond to the center vertex, a domain boundary vertex or a removed T-junction.

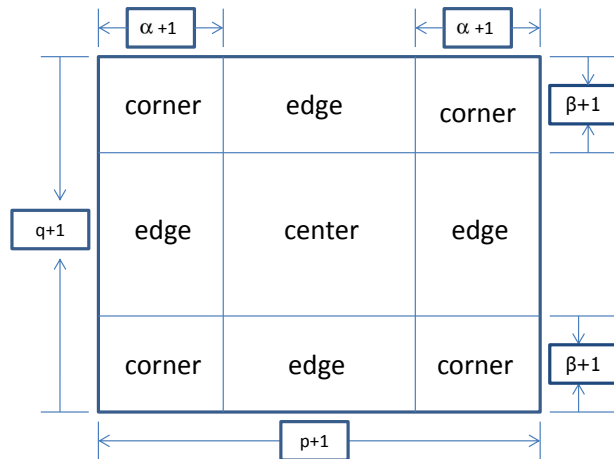


Fig. 6.4: The 9 regions into which each child element is subdivided. The axis labels show the number of Bézier coefficients in each direction.

An example of a quartic basis function before and after subdivision and truncation is plotted in Figure 6.5. The corresponding Bézier ordinates for the refined element and the child elements are shown in Figure 6.6. It can be seen that there are 3 new basis vertices that appear (two on the boundary and one in the interior) as well as 2 T-junctions. The new basis vertices are each marked with a rectangle in Figure 6.6 b), while the T-junctions are marked with a triangle each. The shaded regions surrounding the new basis vertices indicate the Bézier ordinates to be zeroed out. We note

that after this procedure, the modified basis function no longer has support on the child element in the corner of the domain.

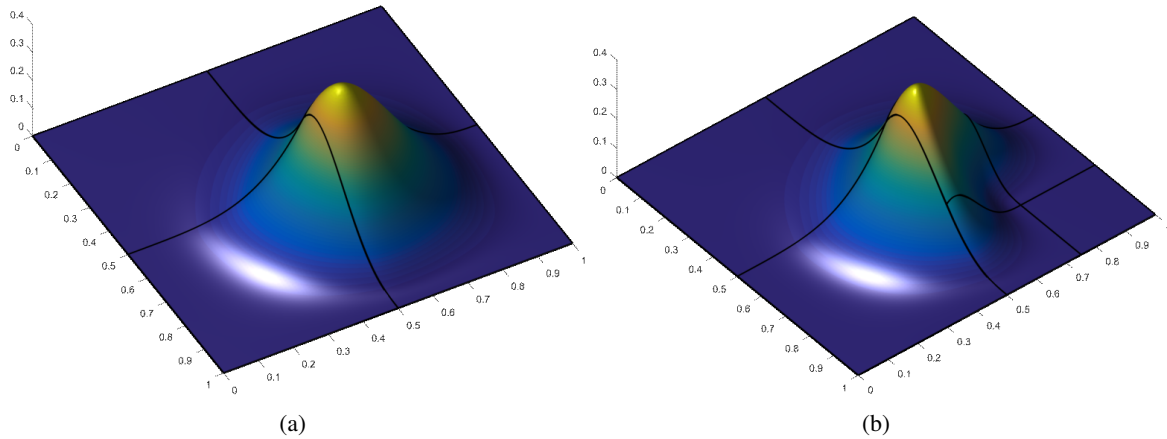


Fig. 6.5: Modification of a quartic ($p=4$) basis function: a) before truncation b) after truncation

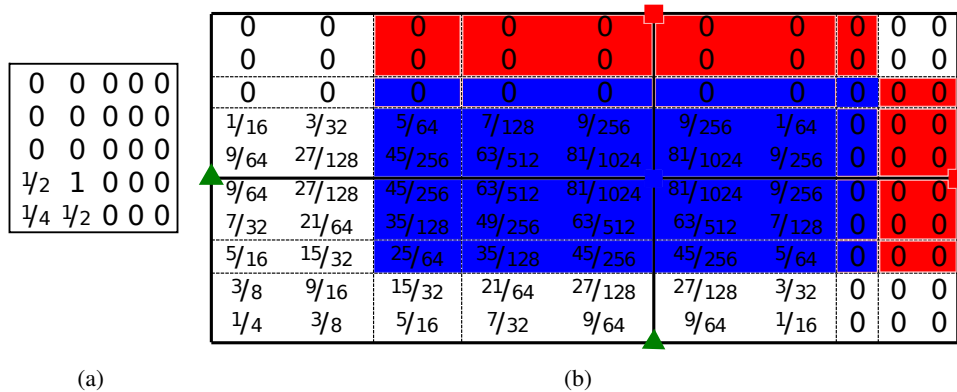


Fig. 6.6: Modification of the Bézier ordinates corresponding to a quartic basis function: a) initial Bézier ordinates on the unrefined element, b) the Bézier ordinates obtained by applying Algorithm 1 to each row and then to each resulting column of the ordinates set in a). The shaded areas corresponding to the new basis vertices indicate the Bézier ordinates that should be set to zero to obtain the truncated basis function.

We remark that the truncation step in this construction is related to the truncation of THB-splines, in the sense that in both cases the support of the basis functions is reduced, resulting in less overlap and a sparser linear system. Furthermore, at this step, any basis functions which could result in a linear dependency with the refined basis are deactivated, since all of their Bézier ordinates are set to zero. The supports of the corresponding basis functions are identical to those of THB-splines of continuity $C^{\alpha, \alpha}$ with dyadic refinement. In both constructions, the resulting approximation space is algebraically complete, meaning they can represent any piecewise polynomial of given degree and smoothness on the hierarchical mesh [94, 93].

6.5.3 Insertion of new basis functions

The last step in the refinement procedure is the insertion of new basis functions in the tree data structure that contains the information about the hierarchical mesh. For each new basis function, this involves the computation of the corresponding Bézier ordinates and the assignment of a new global

basis index. We note that the new basis functions are standard (non-truncated) B-Splines, which can be computed based on local knot-vector information. The local knot vector corresponding to each new basis vertex has three distinct knots, which include the parametric coordinates of the endpoints of the knot-spans to the left, right in the horizontal direction and up and down in the vertical direction of the vertex. For $C^{\alpha,\alpha}$ continuity, the multiplicity of the knots is $p - \alpha$ for interior knots and $p + 1$ for boundary knots (as usual for open knot vectors).

In the case of a removed T-junction, the local knot-vectors need to be computed based on the neighbor connectivity of the elements which is stored in the quad-tree structure. Moreover, the new basis functions need to be inserted in the neighbor elements as they have support on both the refined elements and their neighbors. Fortunately, this process is relatively straight-forward since the restriction to $C^{\alpha,\alpha}$ continuity ensures that each basis function has support on at most two knot-spans in each parametric direction.

Finally, the new basis indices corresponding to the new basis functions are determined. The global indices (IEN array) for the first $(p + 1)^2$ basis functions corresponding to each element can be stored in a two-dimensional array that is organized in the same way as in Figure 6.4. During refinement (cross-insertion), some basis functions are removed by the truncation step, i.e. all their Bézier ordinates are set to zero. Subsequently, new basis functions are added which are assigned either the old basis indices from the removed functions or incrementally new ones.

An example of this procedure is shown in Figure 6.7. For each of the new basis vertices, the basis functions in the closest of the nine element regions are being removed. They are transferred to the refined mesh together with the new indices that are being created. It can be seen that $\alpha + 1$ rows and columns of basis functions are repeated across element boundaries to ensure $C^{\alpha,\alpha}$ continuity. Moreover, it can be seen that corresponding to the mesh T-junctions, no new basis functions are being inserted.

An implementation issue that should be noted is that, different from NURBS or B-Splines over uniform meshes, a variable number of splines have support over a given element (knot-span). For example, in Figure 6.7 b), if the center element is being refined, then one of the resulting child elements will have more than 25 basis functions with non-zero support. This can be determined computationally by considering the basis functions inherited from the parent element whose Bézier ordinates are not all zero after truncation, together with the new (B-Spline) basis functions associated with the new basis vertices. Because of the different supports, the global linear independence of the basis is maintained, even though a *local* linear dependence may occur.

Finally, we note that a simplification occurs when $p = 2\alpha + 1$, i.e. cubic splines with C^1 continuity, quintic splines with C^2 continuity, etc. In these cases, the edge and center regions shown in Figure 6.4 become empty. Then the construction of the new basis functions is simplified and the construction algorithm becomes similar to the standard PHT-splines [49], with $(\alpha + 1)^2$ basis functions associated to each basis vertex. A more detailed overview of the theoretical aspects of the hierarchical polynomial basis with $p \geq 2\alpha + 1$, including a basis dimension formula, can be found in [48].

6.5.4 Geometry mappings

The discussion up to this point focused on the construction of the basis functions in the parameter space. As it is common in IGA, the physical geometry description is given in terms of B-Spline or

57	58	59	60	61	60	61	62	63	64	57	58	59	60	61	60	61	62	72	74	72	74	76	63	64	
49	50	51	52	53	52	53	54	55	56	49	50	51	52	53	44	45	86	87	88	87	88	89	89	69	70
41	42	43	44	45	44	45	46	47	48	41	42	43	44	45	36	37	81	83	85	83	85	91	66	68	
33	34	35	36	37	36	37	38	39	40	33	34	35	36	37	28	29	80	82	84	82	84	90	65	67	
25	26	27	28	29	28	29	30	31	32	25	26	27	28	29	60	61	81	83	85	83	85	91	66	68	
33	34	35	36	37	36	37	38	39	40	33	34	35	36	37	52	53	80	82	84	82	84	90	65	67	
25	26	27	28	29	28	29	30	31	32	25	26	27	28	29	44	45	46	77	78	77	78	79	47	48	
17	18	19	20	21	20	21	22	23	24	17	18	19	20	21	36	37	38	39	40	36	37	38	39	40	
9	10	11	12	13	12	13	14	15	16	9	10	11	12	13	28	29	30	31	32	28	29	30	31	32	
1	2	3	4	5	4	5	6	7	8	1	2	3	4	5											

(a)

(b)

Fig. 6.7: Global indices corresponding to the basis functions on each element (IEN arrays) for $p = 4, \alpha = 1$ (quartic $C^{1,1}$ polynomials). a) A 4 element mesh before refinement. b) The resulting mesh and node indices after refining one element. The colored node indices are the new basis functions. The purple nodes are deleted from the original mesh and reassigned to the refine mesh; the red node indices are new basis functions and indices.

NURBS mapping \mathbf{F} from the parameter space to the physical space, i.e.

$$\mathbf{F}(\mathbf{u}) = \sum_{\mathbf{i}} \mathbf{D}_{\mathbf{i}} \psi_{\mathbf{i}}(\mathbf{u}) = \mathbf{x}, \quad (6.8)$$

where \mathbf{i} is a d -dimensional multi-index corresponding to the elements on a coarse mesh that provides the geometry description. Moreover, \mathbf{u} is the coordinate in the parameter space, \mathbf{x} is the coordinate in the physical space, $\mathbf{D}_{\mathbf{i}}$ are the d -dimensional control points corresponding to the tensor-product isogeometric basis functions $\psi_{\mathbf{i}}$.

The hierarchical spline basis functions are then defined on the physical space with the help of the inverse mapping \mathbf{F}^{-1} as follows:

$$\phi(\mathbf{x}) = \hat{\phi} \circ \mathbf{F}^{-1}(\mathbf{x}), \quad (6.9)$$

where $\hat{\phi}$ is a basis function defined on each element in the parameter space as in (6.5).

Two approaches can be considered for the computation of the basis functions on the physical space. The simplest approach is to consider the mapping fixed, as defined in (6.8). This leads to a geometry-independent field approximation scheme, as previously considered particularly for boundary-element methods [90]. Care must be taken that the elements in the hierarchical mesh are fully contained in the elements for the geometric description and that the continuity across the element lines is the same for both meshes.

For certain applications, such as time-dependent or moving boundary problems, it is desirable to calculate the control point corresponding to each basis function in the field discretization. For the case $p = 2\alpha + 1$, the new control points can be easily computed using the *geometric information* at each new basis vertex, which is defined as the function value and derivatives up to order α in each parameter direction of the geometry mapping evaluated at the basis vertex. In particular, we proceed inductively by assuming that on a given mesh there are N basis functions and the geometry is determined by the N associated control points. When a new basis vertex is inserted, there are $(\alpha + 1)^d$ new basis functions added while we wish to keep the geometric mapping and the previous control points fixed, i.e.

$$\tilde{\mathbf{F}}(\mathbf{u}) = \sum_{i=1}^N \mathbf{D}_i \tilde{\phi}_i(\mathbf{u}) + \sum_{j=N+1}^{N+(\alpha+1)^d} \mathbf{D}_j \phi_j(\mathbf{u}) = \mathbf{F}(\mathbf{u}), \quad (6.10)$$

where $\tilde{\phi}_i$ are the (possibly truncated) basis functions from the previous level and ϕ_j are the newly inserted basis functions. Let the new basis vertex be denoted by \mathbf{u}^* . Then due to the truncation, the basis functions $\tilde{\phi}_i$ and their derivatives up to order α are zero when evaluated at \mathbf{u}^* . Therefore,

$$\frac{\partial^\beta \tilde{\mathbf{F}}}{\partial \mathbf{u}^\beta}(\mathbf{u}^*) = \sum_{j=N+1}^{N+(\alpha+1)^d} \mathbf{D}_j \frac{\partial^\beta \phi_j}{\partial \mathbf{u}^\beta}(\mathbf{u}^*) = \frac{\partial^\beta \mathbf{F}}{\partial \mathbf{u}^\beta}(\mathbf{u}^*), \quad (6.11)$$

where $\beta = (\beta_1, \dots, \beta_d)$ is a multi-index, with $0 \leq \beta_i \leq \alpha, \quad i = 1, \dots, d$. This results in a linear system of $(\alpha + 1)^d$ equations with $(\alpha + 1)^d$ unknowns for each spatial direction, where the left-hand side matrix depends only on the local knot vector coordinates and can be expressed as a tensor product of $(\alpha + 1)$ matrices. We refer to [40] for more details in the case $p = 3, \alpha = 1$.

For the case $p \geq 2\alpha + 1$, a different procedure is needed, similar to that of calculating control points after knot-insertion in standard B-Splines. We note that if an isoparametric representation is used on the initial (coarse) mesh, the calculation of new control points is not necessary assuming the geometry mapping is fixed ($\mathbf{F} = \tilde{\mathbf{F}}$) at each refinement step. In the case where the geometry is given using rational B-Splines (NURBS), there is a slight difference between using isoparametric rational PHT-splines and a geometry independent approximation, as noted also in [79] for the case of a Hermite basis. These differences are investigated numerically for the case $p = 3$ in the numerical results section (see in 8.4).

6.6 2D higher-order PHT-spline (degree, $p > 3, q > 3$ and continuity $C^\alpha \subseteq C^1, C^\beta \subseteq C^1$)

6.6.1 2D refinement algorithm of higher-order PHT-spline

The 2D refinement algorithm starts for example, a 4×4 system of 16 basis function values of a basis function $b_i^k(\xi, \eta)$. Here Bézier ordinates for the unrefined element for a bi-cubic polynomial.

$$\begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{bmatrix}$$

1. Apply De Casteljau [102] algorithm for each row and the matrix A becomes B in dimension 4×8 matrix below:

$$\begin{bmatrix} B_{11} & B_{12} & B_{13} & B_{14} & B_{15} & B_{16} & B_{17} & B_{18} \\ B_{21} & B_{22} & B_{23} & B_{24} & B_{25} & B_{26} & B_{27} & B_{28} \\ B_{31} & B_{32} & B_{33} & B_{34} & B_{35} & B_{36} & B_{37} & B_{38} \\ B_{41} & B_{42} & B_{43} & B_{44} & B_{45} & B_{46} & B_{47} & B_{48} \end{bmatrix}$$

2. After the row operation in unrefined Bézier ordinates, then De Casteljau algorithm [102] is applied for each column of the matrix B. Matrix B from 4×8 dimension becomes 8×8

dimension in the form of matrix C below:

$$\begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & C_{16} & C_{17} & C_{18} \\ C_{21} & C_{22} & C_{23} & C_{24} & C_{25} & C_{26} & C_{27} & C_{28} \\ C_{31} & C_{32} & C_{33} & C_{34} & C_{35} & C_{36} & C_{37} & C_{38} \\ C_{41} & C_{42} & C_{43} & C_{44} & C_{45} & C_{46} & C_{47} & C_{48} \\ C_{51} & C_{52} & C_{53} & C_{54} & C_{55} & C_{56} & C_{57} & C_{58} \\ C_{61} & C_{62} & C_{63} & C_{64} & C_{65} & C_{66} & C_{67} & C_{68} \\ C_{71} & C_{72} & C_{73} & C_{74} & C_{75} & C_{76} & C_{77} & C_{78} \\ C_{81} & C_{82} & C_{83} & C_{84} & C_{85} & C_{86} & C_{87} & C_{88} \end{bmatrix}$$

3. Zero out (Fig. 6.6) the entries of matrix C corresponding to new vertices. Here defined in red, orange and green region Bézier ordinates becomes zero.

$$\begin{pmatrix} C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & C_{16} & C_{17} & C_{18} \\ C_{21} & C_{22} & C_{23} & C_{24} & C_{25} & C_{26} & C_{27} & C_{28} \\ C_{31} & C_{32} & C_{33} & C_{34} & C_{35} & C_{36} & C_{37} & C_{38} \\ C_{41} & C_{42} & C_{43} & C_{44} & C_{45} & C_{46} & C_{47} & C_{48} \\ C_{51} & C_{52} & C_{53} & C_{54} & C_{55} & C_{56} & C_{57} & C_{58} \\ C_{61} & C_{62} & C_{63} & C_{64} & C_{65} & C_{66} & C_{67} & C_{68} \\ C_{71} & C_{72} & C_{73} & C_{74} & C_{75} & C_{76} & C_{77} & C_{78} \\ C_{81} & C_{82} & C_{83} & C_{84} & C_{85} & C_{86} & C_{87} & C_{88} \end{pmatrix}$$

4. Add the new basis functions in:

- Current element
- In the neighbour element if there are any T-junctions then it will be removed. This may result in more than $(p + 1) \times (q + 1)$ basis functions with support on a given element where p and q is the degree of the B-spline basis function.

5. Establish the element node index connectivities (IEN). For a C^1 -continuous cubic spline, where every interior knots is of multiplicity two, the evaluation of the dimension formula is of form:

$$\dim \mathcal{S}(3, 3, 1, 1, \mathcal{T}) = 4(V^b + V^+), \tag{6.12}$$

where V^b, V^+ are boundary vertices, and interior crossing vertices, respectively. For example, we have $V^b = 8$ and $V^+ = 1$ as illustrated in Fig. 6.8. The dimension formula is

$$\dim \mathcal{S}(3, 3, 1, 1, \mathcal{T}) = 4(8 + 1) = 36 \tag{6.13}$$

In matrix form, it can be written that every vertices is dependent on $4 \times 4 = 16$ Bézier ordinates like:

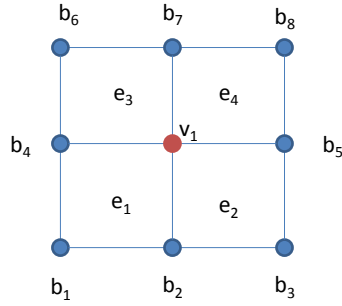


Fig. 6.8: 9 vertices where blue b_i is boundary vertices, red v_i is crossing vertices and e_i are elements.

$$IEN = \begin{pmatrix} \begin{array}{|cccc|cccc} \hline 31 & 32 & 33 & 34 & 33 & 34 & 35 & 36 \\ \hline 25 & 26 & 27 & 28 & 27 & 28 & 29 & 30 \\ \hline 19 & 20 & 21 & 22 & 21 & 22 & 23 & 24 \\ \hline 13 & 14 & 15 & 16 & 15 & 16 & 17 & 18 \\ \hline 19 & 20 & 21 & 22 & 21 & 22 & 23 & 24 \\ \hline 13 & 14 & 15 & 16 & 15 & 16 & 17 & 18 \\ \hline 7 & 8 & 9 & 10 & 9 & 10 & 11 & 12 \\ \hline 1 & 2 & 3 & 4 & 3 & 4 & 5 & 6 \\ \hline \end{array} \end{pmatrix}$$

If the order becomes higher suppose for $p = q = 4$ for C^1 -continuity then the Bézier ordinates or node index connectivities (IEN) dimension can be calculated for higher-order PHT-spline and the dimension of spline space can be found from [48]. Given a regular T-mesh and a corresponding spline space $\mathcal{S}(p, q, \alpha, \beta, \mathcal{T})$, and suppose $p \geq 2\alpha + 1$ and $q \geq 2\beta + 1$ from Eq.6.14, where $F = 4, V = 1, p = 4, q = 4, \alpha = 1, \beta = 1, E_h = 2, E_v = 2$ and we get $dim(S^4) = 64$.

$$dim \mathcal{S}(p, q, \alpha, \beta, \mathcal{T}) = F(p+1)(q+1) - E_h(p+1)(\beta+1) - E_v(\alpha+1)(q+1) + V(\alpha+1)(\beta+1) \quad (6.14)$$

It can be shown the numbering of the bézier ordinates in the following way:

$$IEN = \begin{pmatrix} \begin{array}{|ccccc|ccccc} \hline 57 & 58 & 59 & 60 & 61 & 60 & 61 & 62 & 63 & 64 \\ \hline 49 & 50 & 51 & 52 & 53 & 52 & 53 & 54 & 55 & 56 \\ \hline 41 & 42 & 43 & 44 & 45 & 44 & 45 & 46 & 47 & 48 \\ \hline 33 & 34 & 35 & 36 & 37 & 36 & 37 & 38 & 39 & 40 \\ \hline 25 & 26 & 27 & 28 & 29 & 28 & 29 & 30 & 31 & 32 \\ \hline 33 & 34 & 35 & 36 & 37 & 36 & 37 & 38 & 39 & 40 \\ \hline 25 & 26 & 27 & 28 & 29 & 28 & 29 & 30 & 31 & 32 \\ \hline 17 & 18 & 19 & 20 & 21 & 20 & 21 & 22 & 23 & 24 \\ \hline 9 & 10 & 11 & 12 & 13 & 12 & 13 & 14 & 15 & 16 \\ \hline 1 & 2 & 3 & 4 & 5 & 4 & 5 & 6 & 7 & 8 \\ \hline \end{array} \end{pmatrix}$$

Here for bi-quartic ($p = q = 4$) polynomials, every vertex belongs to $5 \times 5 = 25$ Bézier ordinates.

6. Update the neighbour connectivities. The algorithm for updating the neighbouring connectivi-

ties for element 4 in Fig. 6.9 left are given below:

- For each new child element, here for example PHT-spline element number 2 in Fig. 6.9 right has neighbouring element left = 1, right = empty, up = 5,6 and down = empty.
- if the child element has neighbour down = [5,6].
 - Update neighbour down with the new child element.
 - Remove the parent element from the neighbour down list.
- By this way, we have to update the child element neighbour up, neighbour left and neighbour right after inserting a cross.

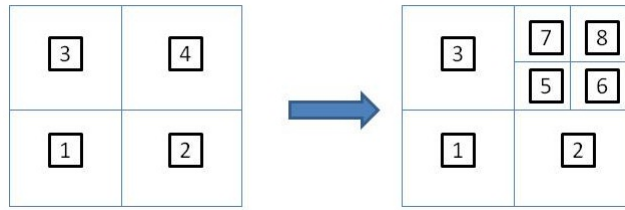


Fig. 6.9: Inserting a cross in a parent element at the left figure, $e=4$ creates child element $e=\{5,6,7,8\}$ at the figure right.

6.6.2 Higher order PHT-spline basis function

In Fig. 6.10(a) B-spline basis function on coarsest mesh in 2D has been demonstrated for degree $p=q=5$ with C^2 continuity of knot vector $U=V=\{0, 0, 0, 0, 0, 0, 0.5, 0.5, 0.5, 1, 1, 1, 1, 1\}$.

Then Bernstein basis function in reference element $[\xi, \eta] = [-1, 1] \times [-1, 1]$ in 2D in Fig. 6.10(b) and B-spline basis functions represented on the coarsest mesh using Bernstein basis function with Bézier extraction operator in Fig. 6.11(a).

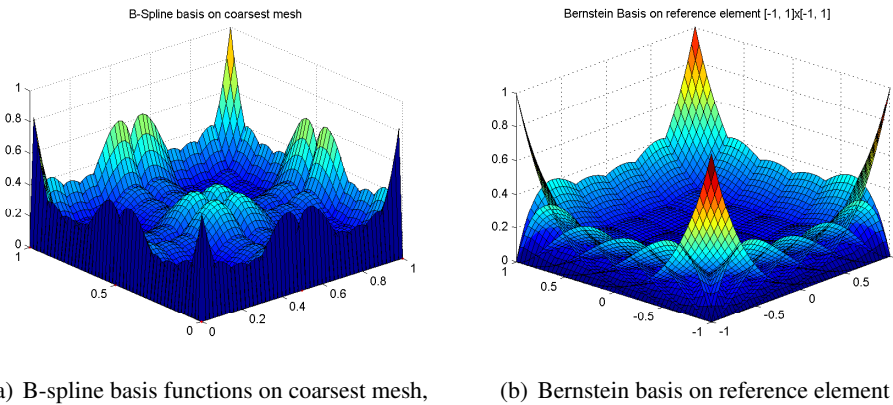
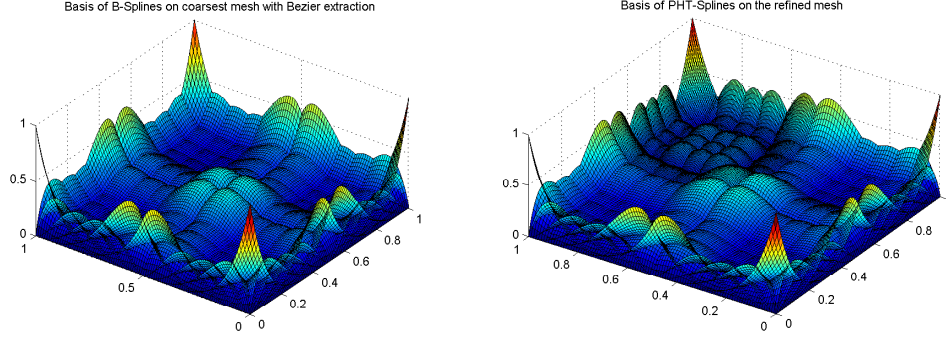


Fig. 6.10: B-spline and Bernstein basis representation for degree $p = q = 5$ with C^2 continuity.

In the above description, we can derive the higher-order PHT-spline basis function using Bézier extraction operator and it can be represented in refined elements like Fig. 6.9 demonstrated in Fig. 6.11(b). In Fig. 6.11(b) it is also noticed after knot insertion at $(\xi, \eta) = (0.75, 0.75)$ in the initial mesh $[\xi, \eta] = [0, 0.5, 1] \times [0, 0.5, 1]$ more bubble functions are added after inserting a cross in an element, $e = 4$ in Fig. 6.8.



(a) B-spline basis functions with Bézier extraction operator, (b) PHT-spline basis functions on refined elements

Fig. 6.11: 2D B-spline representation using Bézier extraction and PHT spline basis function representation on refined mesh with degree $p = q = 5$, C^2 continuity.

6.7 Numerical Examples

In this section, we present a numerical example of linear elasticity Eq. 4.24 to illustrate the effectiveness of the proposed GIFT method.

6.7.1 Thick cylinder subjected to internal pressure

We now consider a higher-dimensional problem and we start with the example of a linear elasticity problem on a 2D domain consisting of a pressurized cylinder, as sketched in Fig. 6.12. Here we select internal and external radius equal to $R_1 = 1$ and $R_2 = 4$, and the internal pressure of the cylinder $P = 30000 N/m^2$ respectively. The Young's modulus and Poisson's ratios are $E = 3 \times 10^7 N/m^2$, $\nu = 0.25$ and plane stress condition is considered in this problem. Symmetry conditions are imposed on the left and bottom edges, and the outer boundary is traction free. The exact solution for the stress components as:

$$\begin{aligned}\sigma_r(r) &= \frac{R_1^2 P}{R_2^2 - R_1^2} \left(1 - \frac{R_2^2}{r^2}\right) \\ \sigma_\phi(r) &= \frac{R_1^2 P}{R_2^2 - R_1^2} \left(1 + \frac{R_2^2}{r^2}\right) \\ \sigma_{\phi r} &= 0\end{aligned}\tag{6.15}$$

where the radial and the tangential exact displacements are given by:

$$\begin{aligned}u_r(r) &= \frac{R_1^2 P r}{E(R_2^2 - R_1^2)} \left\{1 - \nu + \frac{R_2^2}{r^2} (1 + \nu)\right\} \\ u_\phi(r) &= 0\end{aligned}\tag{6.16}$$

Here the thick cylinder computational domain in Fig. 6.13 is represented by quadratic NURBS and the solution field is represented by cubic C^1 PHT-spline in Fig. 6.13(a), quadratic in Fig. 6.13(b) and quintic in Fig. 6.13(c). we will consider initially from cubic C^1 -continuous PHT-spline which can be exactly represented by a single quadratic NURBS patch geometry. Adaptive local refinement used to capture higher stress concentration in the solution field using the hierarchical property of this PHT-spline. Later we will also compare the results with higher-order and greater continuous PHT-spline with the cubic C^1 PHT-splines.

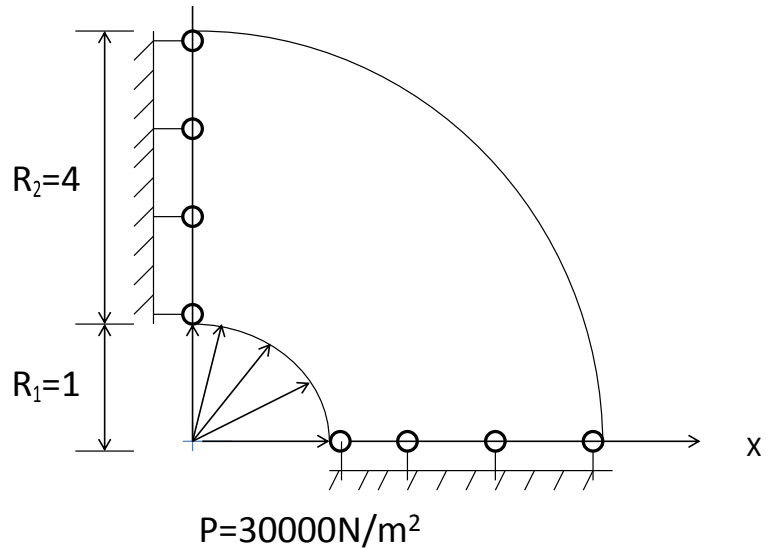


Fig. 6.12: Thick pressurized cylinder.

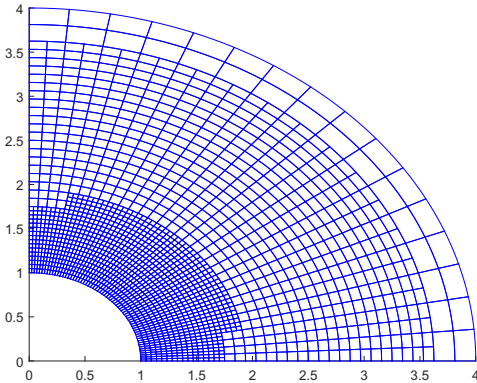
In this example, the performance of the error estimator can be analyzed for a smooth solution. For refinement, a Dörfler refinement scheme was used, with the parameter $\theta = 0.75$. Therefore refinement steps are needed for this example to obtain a mesh with as few elements as possible where the stress could be more. The refined meshes for polynomial degrees 3, 4 and 5 are shown in Fig. 6.13, where the refinement was stopped when the estimated relative error reached 10^{-5} .

For higher polynomial degrees, meshes with significantly fewer elements are required to reach the target relative error 10^{-5} . In particular, much more graded meshes are obtained for $p = 3$ and $p = 4$, with most of the refinements concentrated around the inner circular zone of the cylinder as the stress concentration is higher in this zone.

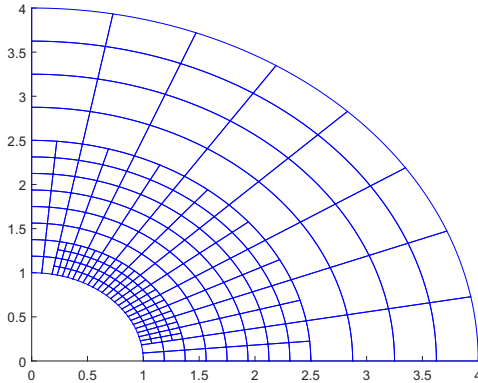
In Fig. 6.14 shows contour plots for the thick cylinder of different displacements and stress components.

From Fig. 6.15 it is shown that for higher-order with greater continuity we can get more accuracy and higher convergence rate in the energy norm. For this thick cylinder numerical problem with a smooth solution, the convergence rate does not change much between the refinement levels as we can see from the H^1 -norm convergence plots.

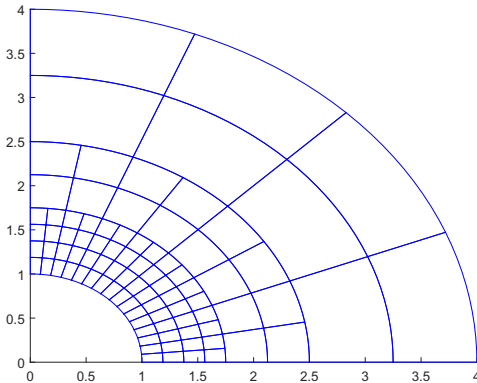
In Fig. 6.15, the convergence plots for the different polynomial degrees with uniform and adaptive refinements are shown for the energy norm. We observe good agreement between the estimated error obtained using the recovery-based estimator and the exact error computed using the analytical solution. For the adaptive refinement and the last refined mesh, the effectivity indices (defined as the ratio of the estimated error and the actual error) are 1.0367 for $p = 3$, 1.0849 for $p = 4$, and 1.2149 for $p = 5$. Here for higher polynomial degree, because of the smooth solution we do not get much advantage by using adaptive refinement as error convergence rate gets closer to uniform refinements.



(a) PHT with $p = 3$ (1840 active elements)



(b) PHT with $p = 4$ (232 active elements)



(c) PHT with $p = 5$ (88 active elements)

Fig. 6.13: The refined meshes for different polynomial degrees p in the thick cylinder example.

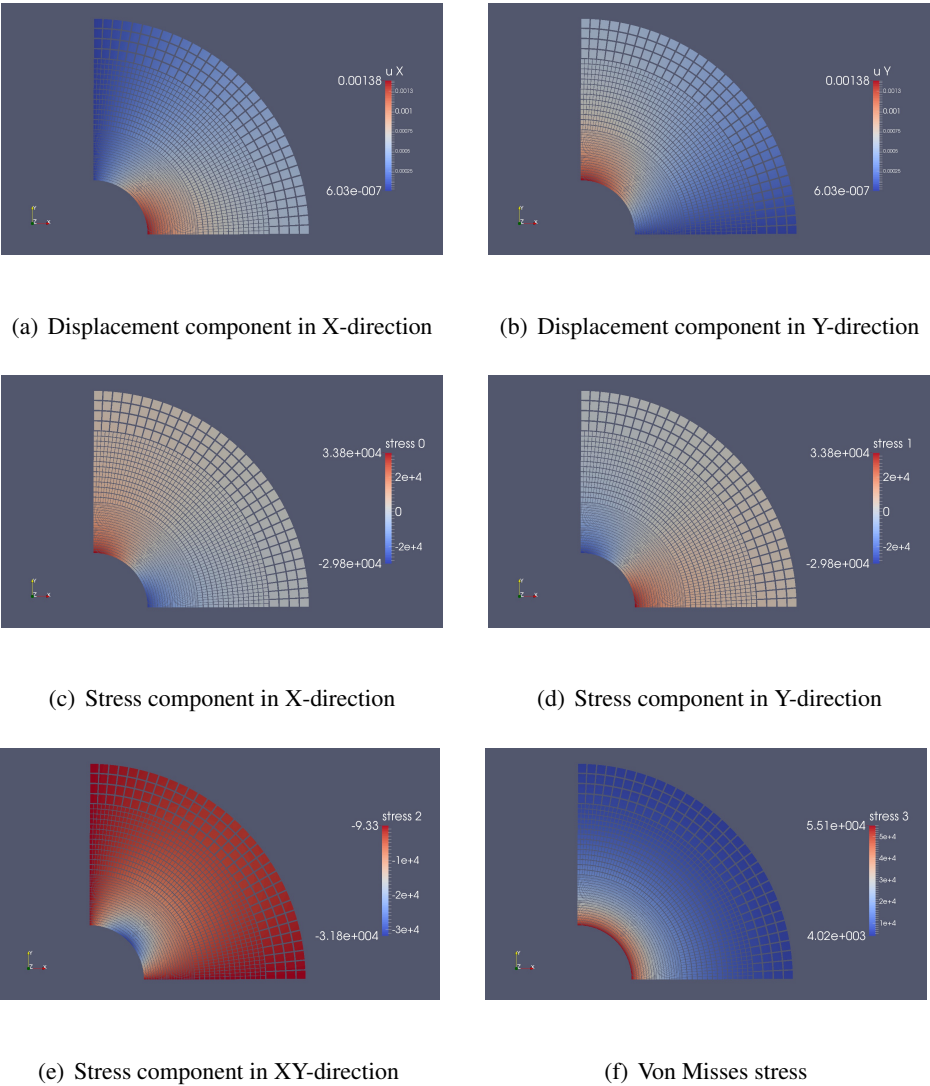


Fig. 6.14: Contour plots of displacement and stress components of the thick cylinder.

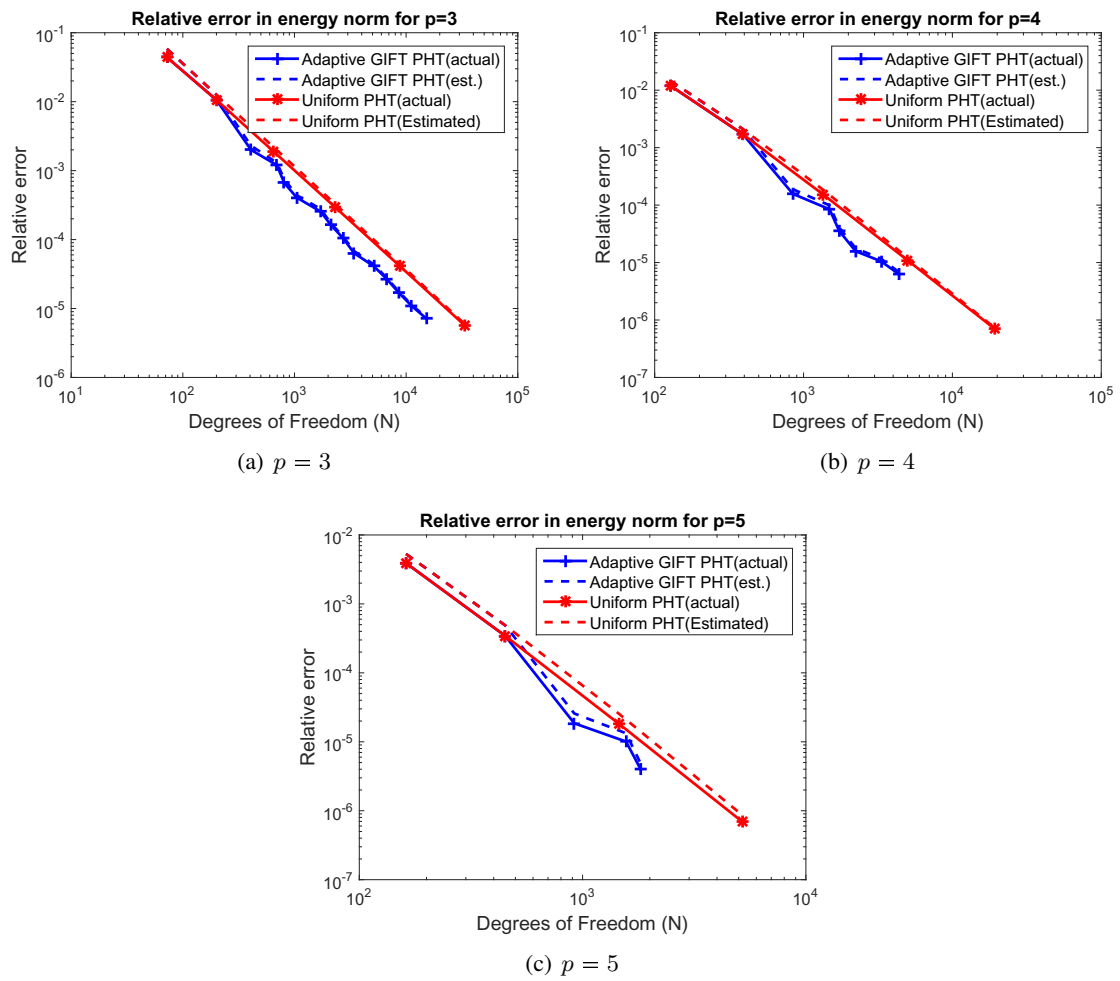


Fig. 6.15: Relative error (exact and estimated) in energy H^1 norm vs. the number of degree of freedom for $p = 3, 4$ and 5 , using adaptive and uniform refinements for thick cylinder example.

Chapter 7

3D adaptive GIFT PHT-spline

GIFT has been introduced for choosing independently the geometry space and field approximation space due to certain constraints at the geometry spline space which may not be suitable to approximate the solution spline space. So this is an alternative approach of isogeometric analysis with some certain benefits. In the CAD community, Non-Uniform Rational B-Splines (NURBS) are classically used for creating the geometry but it has certain drawbacks to use for the solution approximation in numerical analysis. The main drawback of using NURBS is the lack of local refinement and watertight properties. For this reason, recently different spline basis functions have been exploited by IGA such as locally refined splines, T-splines, polycube splines, Sabine splines, truncated hierarchical B-spline and polynomial splines over hierarchical T-meshes.

Although PHT-splines inherit all the important properties of NURBS such as the partition of unity, linear independence, non-negativity and local support. PHT-splines also have the capability of local refinement due to hierarchical properties and joining geometric objects without gaps and overlaps preserving higher continuity everywhere in comparison to NURBS. So choosing NURBS as basis spline for the geometry is the most common and popular in the CAD community and using PHT-spline for the solution field approximation is the main approach we concentrated here with GIFT mapping for the 3D cases.

In this chapter the adaptive 2D PHT-spline approach is extended to 3D PHT-spline using GIFT mapping for linear elasto-static problems. We used adaptive refinement using recovery based error estimation with super-convergent points where the idea Super convergent Patch Recovery (SPR) technique proposed by [133, 134]. Here at first, we will discuss the common cubic, C^1 continuous PHT-spline for the 3D cases and extend our implementation to the higher-order and greater continuity of PHT-spline. Using higher-order and continuity PHT-spline basis can provide more efficient and robust performance comparing to lower order and continuous PHT-spline basis.

7.1 Polynomial spline over 3D hierarchical T-meshes, degree $p = 3$, C^1 continuity

In this section, we will describe the 3D hierarchical T-meshes and definition of spline spaces for 3D case with dimension formula for cubic C^1 continuous PHT-spline basis function.

7.1.1 3D T-meshes

3D T-meshes is the extension of 2D T-meshes where a mesh on a rectangular grid allow T-junctions and the end points of each grid line lie on two other grid lines. Similar to 2D in 3D cases each cell and facets are formed by quadrilateral. A 3D T-mesh is basically a partition of unity of a cuboid domain $\Omega \in \mathbb{R}^3$ such that every part is a smaller cuboid or we can call octuples where there are no holes among them (see in Fig. 7.1). Suppose Ω is a 3D T-mesh, which is a partition of octuples Ξ . Then the smaller octuple is called cell of the T-mesh. A **vertex** of the octuples in the T-mesh is assigned to each grid point. If a vertex is on the boundary of domain Ω boundary Ξ , then it is called **boundary vertex** (see in Fig. 7.1 boundary vertex b_i) otherwise if a vertex is inside the domain, it is called **interior vertex** (v_i^+ in Fig. 7.1). If an edge is on the boundary, then it is called a **boundary edge** ($b_1 b_2$); otherwise, it is called an **interior edge** ($v_1^+ b_{24}$). Same things also go for faces (**boundary face** ($b_{16} b_{23} b_{10} b_3$) and **interior face** ($v_1^+ b_{23} b_{16} b_{26}$)). Two parallel faces are called **adjacent** faces if they share a common rectangle ($b_{13} b_{25} v_1^+ b_{23}$).

Composite face(c-face) and **Composite edge (c-edge)** are important to define dimension formula for higher-order and continuity PHT-basis. A **c-face** can be defined such as a face of some cell is adjacent to the c-face, then it belongs to the c-face, but this is not the case for any subsets of c-face. A **composite edge** can be defined such as of some direction of a set of edges which is adjacent to the c-edge, then it belongs to the c-edge. It can be said that all the boundary faces are c-faces. For example in Fig. 7.1 $b_{13} b_{25} v_1^+ b_{23}$ is a c-face of yz and $v_1^+ b_{22} b_{19} b_{26}$ is a c-face of xy. c-edges are $b_{26} v_1^+$ in y-direction and $b_{23} v_1^+$ in z-direction respectively.

Plus edge (p-edge) is called such an edge if a c-edge belongs to four c-faces, then it is only on the edges of cells in the domain Ω . **Plus vertex (p-vertex)** is a vertex, if a vertex does not lie on any faces of all the cells ($b_{26} v_1^+$ and $b_{23} v_1^+$ are **p-edges** and v_1^+, b_{23}, b_{22} are **p-vertex**). Another special kind of vertex for 3D T-mesh is **e-vertex** which lies on the **p-edge** and the vertex of four cells and also end point of six edges. For example v^+ is an **e-vertex**.

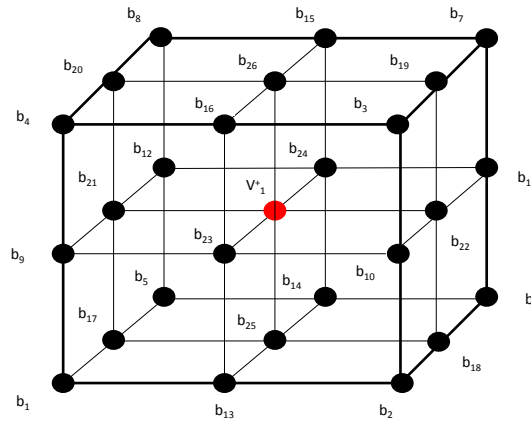


Fig. 7.1: Example of 3D T-mesh.

7.1.2 3D hierarchical T-meshes

Hierarchical T-meshes consist of a standard tensor product splines on a cube with three knot sequences $\mathbf{U} = \{u_1, u_2, \dots, u_{n+p+1}\}$, $\mathbf{V} = \{v_1, v_2, \dots, v_{m+q+1}\}$, $\mathbf{W} = \{w_1, w_2, \dots, w_{l+r+1}\}$, where p, q, r are the polynomial orders, and n, m, l are the number of basis along x, y , and z -direction respectively. We can construct 3D hierarchical T-meshes from starting level \mathcal{S}_0 to \mathcal{S}_k

where $k = 0, 1, 2, \dots$ in the following iterative way. For any $k \geq 1$, selecting some cells of level k from \mathcal{S}_k and dividing each of the selected k into eight octuples which are parallel to xy , yz and zx . Then we can get the mesh in level \mathcal{S}_{k+1} and all the new octuples are leveled with $(k + 1)$. Then we can call any mesh for $k \geq 1$, the mesh \mathcal{S}_k is called 3D hierarchical T-meshes of level k . In Fig. 7.2 we can see the different hierarchical levels of 3D hierarchical T-meshes.

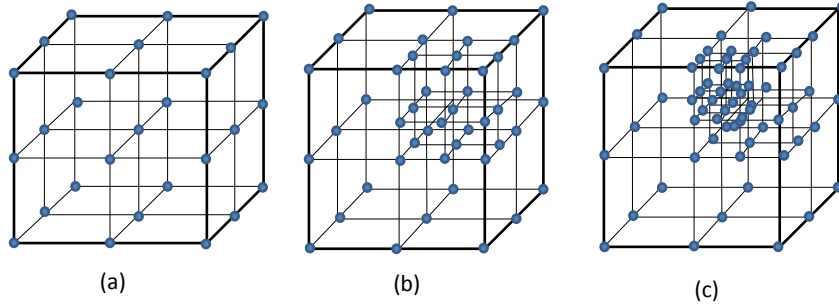


Fig. 7.2: (a) 3D Hierarchical T-Mesh at level \mathcal{S}_0 , (b) 3D Hierarchical T-Mesh at level \mathcal{S}_1 , (c) 3D Hierarchical T-Mesh at level \mathcal{S}_2 .

7.1.3 A dimensional formula for 3D T-meshes

The dimensional formula for 3D T-mesh is a generalization of 2D T-mesh described by [48]. Let \mathcal{T} be the 3D T-mesh with the octuples Ξ , and $\Omega \in \mathbb{R}^3$ be the region occupied by all the octuples in \mathcal{T} . Then, when $p \geq 2\alpha + 1$, $q \geq 2\beta + 1$ and $r \geq 2\gamma + 1$, it can be written the dimensional formula in the following way:

$$\mathcal{S}(p, q, r, \alpha, \beta, \gamma, \mathcal{T}) = \{s(x, y, z) \in C^{\alpha, \beta, \gamma}(\Omega) | s(x, y, z)|_{\phi} \in \mathbb{P}_{pqr} \quad \forall \phi \in \Xi\}, \quad (7.1)$$

where \mathbb{P}_{pqr} is space consists of all polynomials with tri-degree (p, q, r) , and the space $C^{\alpha, \beta, \gamma}(\Xi)$ consists of all tri-variate functions which are continuous in Ξ with the order α, β, γ along x, y, z -directions respectively. It can be said that the space $\mathcal{S}(p, q, r, \alpha, \beta, \gamma, \mathcal{T})$ is a linear space and we can mention it as **the spline space over the T-mesh \mathcal{T}** .

For any cubic C^1 -continuous spline, where every interior knots have multiplicity two, the dimensional formula can be written as:

$$\dim \mathcal{S}(3, 3, 3, 1, 1, 1, \mathcal{T}) = 8(V^b + V^+), \quad (7.2)$$

where, V^b, V^+ are the boundary and interior vertices respectively. From Eq. 7.2 it is shown that the number of basis functions can be constructed for each of the boundary and crossing vertices. It also implies that each of the boundary and cross vertices is connected to 8 basis functions.

For example, using the **corollary 1** in [84] it can be shown that for $p = q = r = 3$, C^1 continuity where $\alpha = \beta = \gamma = 1$ the dimensional formula for a cell with a cross (see Fig. 7.1) can be written in the following way:

$$\begin{aligned} \dim \mathcal{S}(3, 3, 3, 1, 1, 1, \mathcal{T}) &= (3 + 1)^3 + (8 - 1)(3 + 1)(1 + 1)(3 + 1 + 2) - \\ &12(3 + 1)(1 + 1)(3 - 1) + (1 - 0 - 0 - 0)(1 + 1)^2(4 - 1) = 216 \end{aligned} \quad (7.3)$$

where, number of cell $C = 8$, internal number of faces $F = 12$ and internal number of vertices $V^+ = 1, V_x = 0, V_y = 0, V_z = 0$.

Using **Corollary 2** in [84], we can also get the same dimension

$$\dim \mathcal{S}(3, 3, 3, 1, 1, 1, \mathcal{T}) = (1 + 1)(1 + 1)(1 + 1)27 = 216, \quad (7.4)$$

where the number of total p-vertices are, $V^+ = 27$.

Similarly for higher degree $p = q = r = 5$ with continuity order $\alpha = \beta = \gamma = 2$ from **corollary 2** in [84], the dimensional formula can be defined with the number of basis functions as follows:

$$\dim \mathcal{S}(5, 5, 5, 2, 2, 2, \mathcal{T}) = (2 + 1)(2 + 1)(2 + 1)27 = 729 \quad (7.5)$$

So we can construct by this way for higher-order 3D PHT-spline the number of basis functions for each of the boundary and crossing vertices.

7.1.4 Basis function representation and refinement in 3D

The construction of the basis functions in three dimensions can be performed in a similar manner as in 2D, though the proper “book-keeping” of the hierarchical structure is significantly more challenging. In terms of neighbor connectivity, each 3D element may have a variable number of face neighbors for each of the 6 faces and edge neighbors for the 12 edges of the element. As before, these are stored as pointers in the element structure. Each basis function in the parameter space is determined by $(p + 1)^3$ Bézier ordinates, which can be arranged in a three-dimensional array. However, for performance reasons, the basis functions associated with each element are stored in a two-dimensional matrix where each row contains the $(p + 1)^3$ ordinates for a particular basis function.

The refinement is performed in an element-by-element fashion, where the first step is the application of the Algorithm 1 in each parametric direction, yielding the Bézier representation of the basis on each of the eight child elements. For the truncation part, the $(p + 1) \times (p + 1) \times (p + 1)$ ordinates array is divided into 27 subregions, of which there are 8 corners, 12 edges, 6 faces and 1 center. The corner regions contain $(\alpha + 1)^3$ ordinates, while the edge, face and center region contain $(p - 2\alpha - 1)(\alpha + 1)^2$, $(p - 2\alpha - 1)^2(\alpha + 1)$, and $(p - 2\alpha - 1)^3$ ordinates respectively. For the case $p = 2\alpha + 1$, only the corner regions are non-empty.

We note that for each refined 3D element, there are 19 possible locations for new basis vertices, versus 5 for 2D (at the element center, or one of the 6 face centers, or one of 12 edge midpoints). The new basis functions that are inserted are standard B-Splines whose local knot vectors are determined by the position of basis vertex in the element and the element neighbors. The other parts of the refinement algorithm are similar to those in 2D, though more cases need to be considered due to the increased complexity. We refer to the published code for more implementation details.

7.2 Numerical example of 3D adaptive GIFT PHT-spline

Similar to the 2D adaptive GIFT PHT-spline, we will show some 3D adaptive GIFT PHT-spline numerical examples where recovery-based error estimation has been used with GIFT geometry mapping. Similar to our 2D implementation here as well geometry has been considered with NURBS and the solution field has been approximated using PHT-spline. At level 0, both NURBS and PHT-splines represented with a uniformly coarse mesh for the initial step. But we refine at the solution space

which has been defined by PHT-spline keeping the initial NURBS geometry fixed. As PHT-splines formulation allows for local refinement, we do refinement only in the solution field.

7.2.1 Hollow sphere subjected to internal pressure

The first example is performed with a hollow sphere of internal radius $R_1 = 1m$ and external radius $R_2 = 2m$, subjected to an internal pressure $P = 1N/m^2$ as illustrated in Fig. 7.3. Only 1/8 of the geometry is modeled due to symmetry and symmetry conditions are imposed on the three planes of geometry. The material parameters are Young's modulus $E = 1kPa$ and Poisson ratio $\nu = 0.3$. The exact solutions in polar coordinates by [122] are:

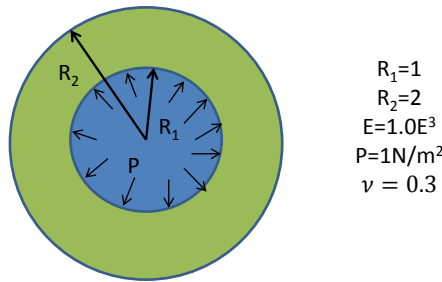


Fig. 7.3: Geometry and material specification of a hollow sphere subjected to internal pressure.

Exact displacement

$$u_r = \frac{PR_1^3 r}{E(R_2^3 - R_1^3)} \left[(1 - 2\nu) + (1 + \nu) \frac{R_2^3}{2r^3} \right] \quad (7.6)$$

Exact stresses

$$\begin{aligned} \sigma_{rr} &= \frac{PR_1^3(R_2^3 - r^3)}{r^3(R_1^3 - R_2^3)} \\ \sigma_{\theta\theta} &= \frac{PR_1^3(R_2^3 - r^3)}{r^3(R_1^3 - R_2^3)}, \\ \sigma_{\phi\phi} &= 0 \\ \sigma_{r\theta} &= 0 \\ \sigma_{r\phi} &= 0 \end{aligned} \quad (7.7)$$

where r is the radial distance from the centroid of the sphere to the point of interest in the sphere.

In Fig. 7.4 different hierarchical refined meshes at the different degrees ($p = 3, 4, 5$) are shown. The initial geometric mesh is defined by NURBS and the solution space is represented by PHT-spline. For different degree, it can be seen that for higher-order PHT-spline we need less element to reach the target relative error. Here, the target relative error is 10^{-4} .

In Fig. 7.5 the displacement and stress contour plot for degree $p = 3$ at the refinement level $k = 5$ has been demonstrated. It can be noticed that at the inside of the hollow sphere we have higher stress concentration which has been captured by adaptive refinement.

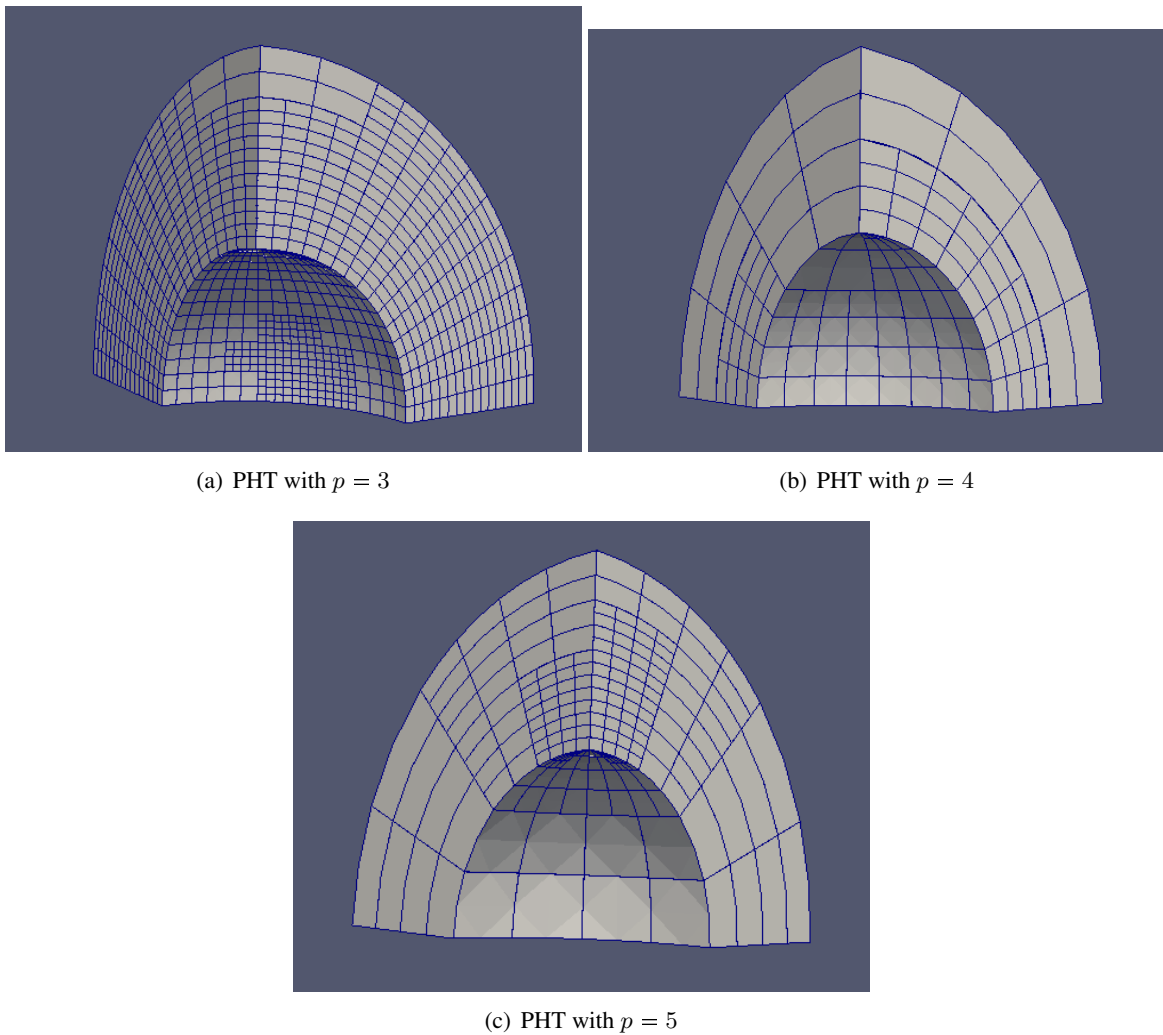


Fig. 7.4: The refined meshes for different polynomial degrees p in the hemisphere example.

In Fig. 7.6, the convergence plots for different polynomial degrees with uniform and adaptive refinements are shown in the energy norm. We observe due to smooth solution we get almost similar convergence rates for lower degree $p = 3$ with greater accuracy in adaptive refinement comparing to uniform refinement. But for the higher degree ($p = 4, 5$) the adaptive refinement at some stage decreases and then we get a better convergence rate at later steps comparing to uniform refinement. For the adaptive refinement and the last refined mesh, the effectivity indices (defined as the ratio of the estimated error and the actual error) are 1.1961 for $p = 3$, 1.1208 for $p = 4$ and 6.3354 for $p = 5$. Here for this example, the effectivity index is considered $\theta = 0.75$ and target relative error = 10^{-4} .

7.2.2 Cube with an internal spherical hole in an infinite domain subjected to uniform tension

In this problem, we are considering a cube with an internal spherical hole in an infinite domain in \mathbb{R}^3 is shown in Fig. 7.7. The exact stresses, given using spherical coordinates (r, ϕ, θ) by [22, 111],

are:

$$\begin{aligned}
\sigma_{rr} &= S \cos^2 \theta + \frac{S}{7-5\nu} \left(\frac{a^3}{r^3} (6-5(5-\nu) \cos^2 \theta) + \frac{6a^5}{r^5} (3 \cos^2 \theta - 1) \right) \\
\sigma_{\phi\phi} &= \frac{3S}{2(7-5\nu)} \left(\frac{a^3}{r^3} (5\nu-2+5(1-2\nu) \cos^2 \theta) + \frac{a^5}{r^5} (1-5 \cos^2 \theta) \right) \\
\sigma_{\theta\theta} &= S \sin^2 \theta + \frac{S}{2(7-5\nu)} \left(\frac{a^3}{r^3} (4-5\nu+5(1-2\nu) \cos^2 \theta) + \frac{3a^5}{r^5} (3-7 \cos^2 \theta) \right) \\
\sigma_{r\theta} &= S \left(-1 + \frac{1}{7-5\nu} \left(-\frac{5a^3(1+\nu)}{r^3} + \frac{12a^5}{r^5} \right) \right) \sin \theta \cos \theta,
\end{aligned} \tag{7.8}$$

where a denotes the radius of the sphere, S is the uniaxial tension applied at infinity, μ is the shear modulus, and ν is Poisson's ratio.

Because of the symmetry, $1/8$ of the problem domain in Figure 7.7 is discretized. Four geometric patches are used to minimize the distortions due to the parametrization. The stress field in the z -direction obtained for a fine mesh with $p = 3$ and the refined meshes for different polynomial degrees are plotted in Figure 7.8. We note that the maximum stress of 2.05 corresponds to the theoretical maximum stress of $\sigma_{max} = 3S(9-5\nu)/(14-10\nu)$, where $S = 1$ is the tension at infinity and $\nu = 0.3$ is the Poisson ratio. It is also observed that for a given accuracy (in this case a relative error of 10^{-4} in the energy norm was selected as the target), fewer elements are required as the polynomial degree increases.

The convergence plots for this problem are shown in Figure 7.9. We observe that the optimal convergence rate in terms of degrees of freedom ($O(N^{-p/3})$) is obtained for both adaptive and uniform refinements, though fewer degrees of freedom are needed when adaptivity is used for a given accuracy. The effectivity index on the finest meshes is 1.07 for $p = 3$, 1.19 for $p = 4$ and 1.37 for $p = 5$.

7.2.3 Solid C^1 elastic ‘‘horseshoe’’ subjected to equal and opposite in-plane flat edge displacements on the soles of the shoe

In this example, we consider a solid C^1 elastic ‘‘horseshoe’’ numerical problem in Fig. 7.10. The dimensions of the domain are considered here length $L = 1$, height $H = 1$ and width $W = 1$. The elastic properties for this example are considered Young modulus $E = 100kPa$ and Poisson ratio $\nu = 0.3$ and the equal and opposite imposed displacement on the soles of the shoe is 0.1.

The meshes for different degree $p = 3, 4, 5$ with adaptive refinement is shown in Fig. 7.11. The displacement and stress contour plot is shown in Fig. 7.12.

The H^1 error in the approximate energy norm is shown in Fig. 7.13 with adaptive GIFT PHT-spline refinement for degree $p = 3, 4, 5$ and compared with uniform PHT-spline refinement. As in this example, we have a smooth solution, so the accuracy of adaptive GIFT PHT-spline in energy norm for the lower degree ($p = 3$) is higher with the same convergence rate as uniform refinement. But for higher degrees ($p = 4, 5$) we get a higher convergence rate in adaptive refinement comparing to uniform refinement. The target relative error was considered here 0.01 and Dörfler refinement scheme was used, with the effectivity index $\theta = 0.75$ for adaptive refinement.

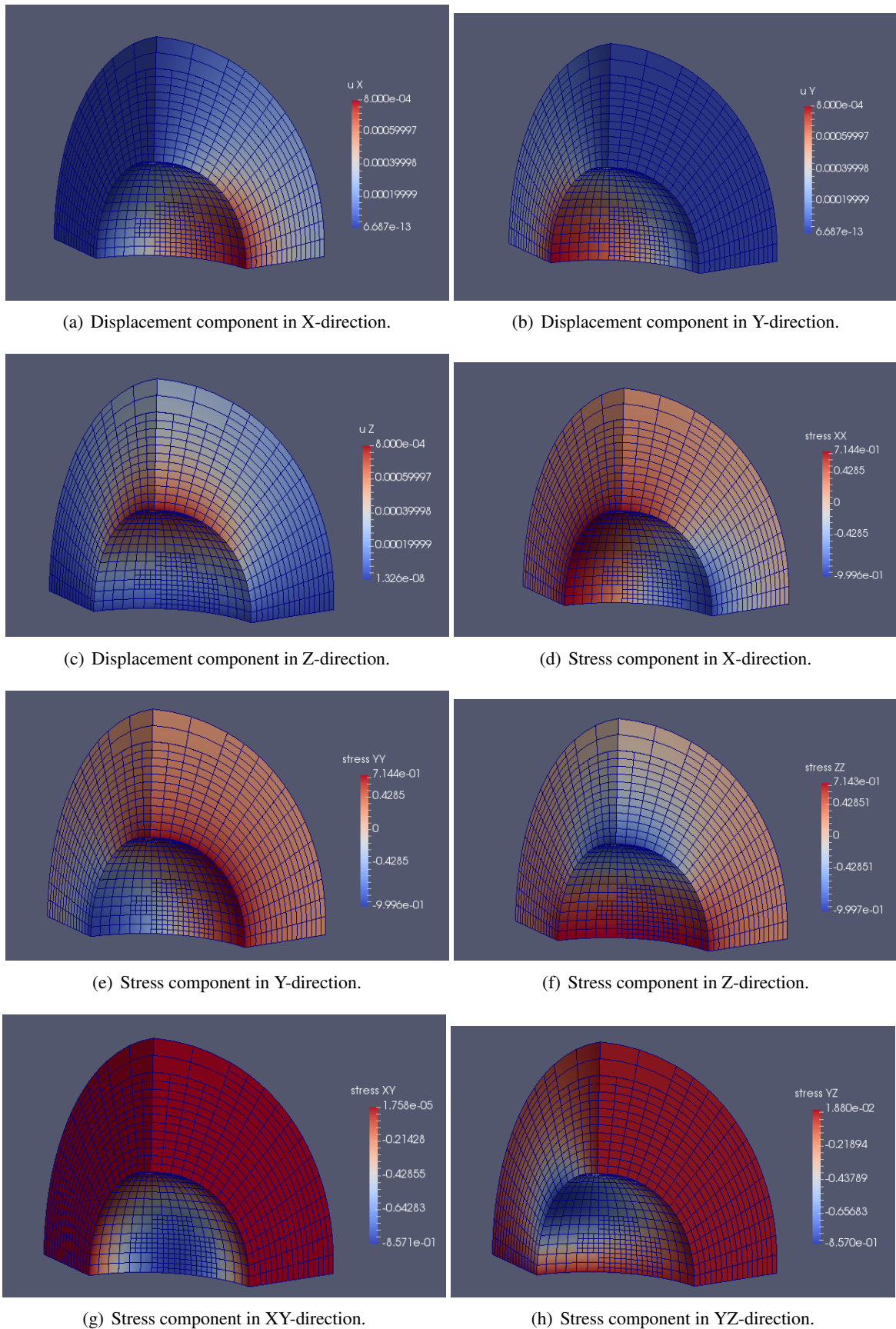


Fig. 7.5: Contour plots of displacement and stress components (x,y,z) of the Hollow sphere subjected to internal pressure.

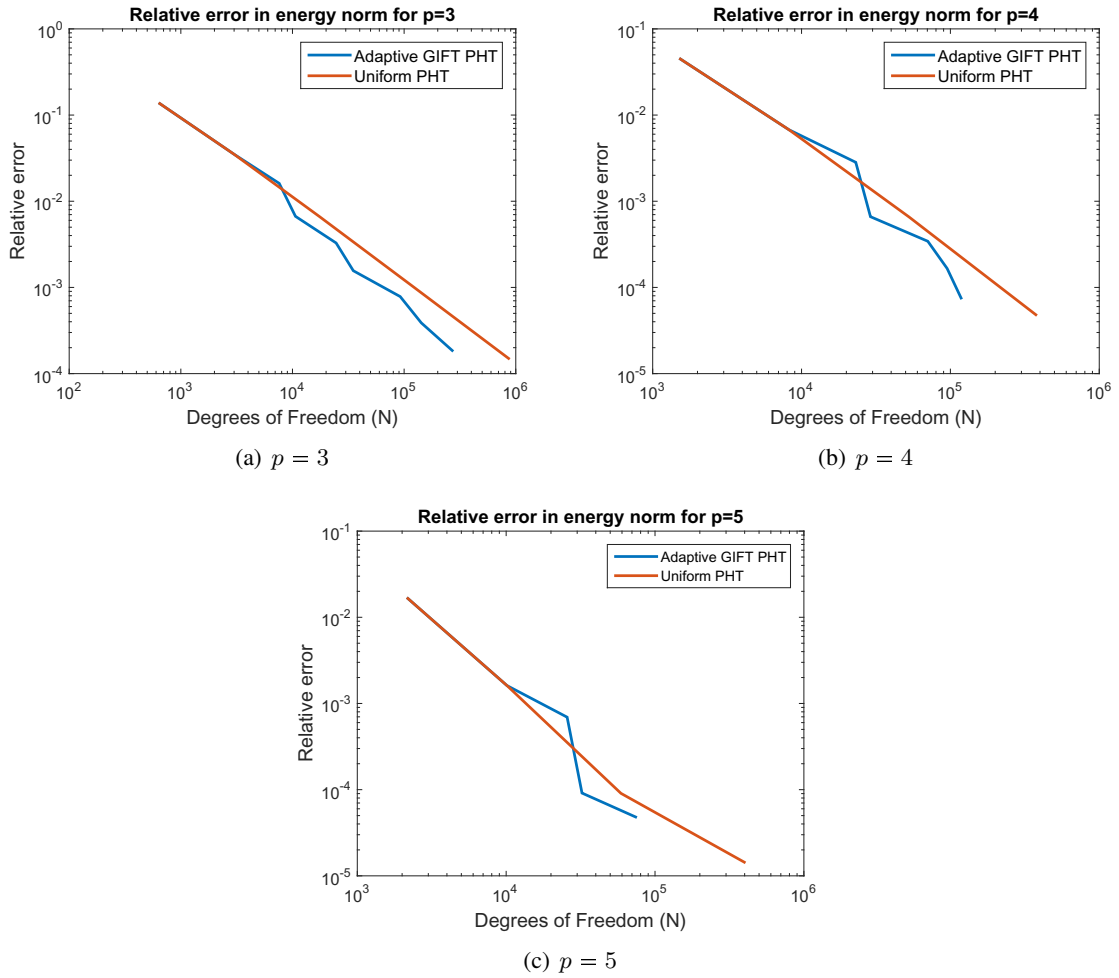


Fig. 7.6: Relative error (exact and estimated) in energy H^1 norm vs. the number of degree of freedom for $p = 3, 4, 5$, using adaptive and uniform refinements for Hollow sphere example.

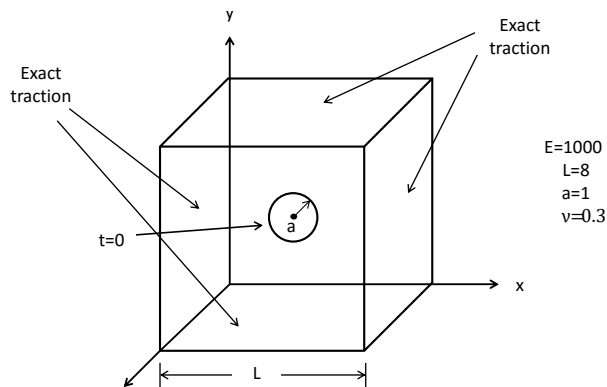


Fig. 7.7: Geometry and material specification of cube with a spherical hole in a infinite domain subjected to uniform tension.

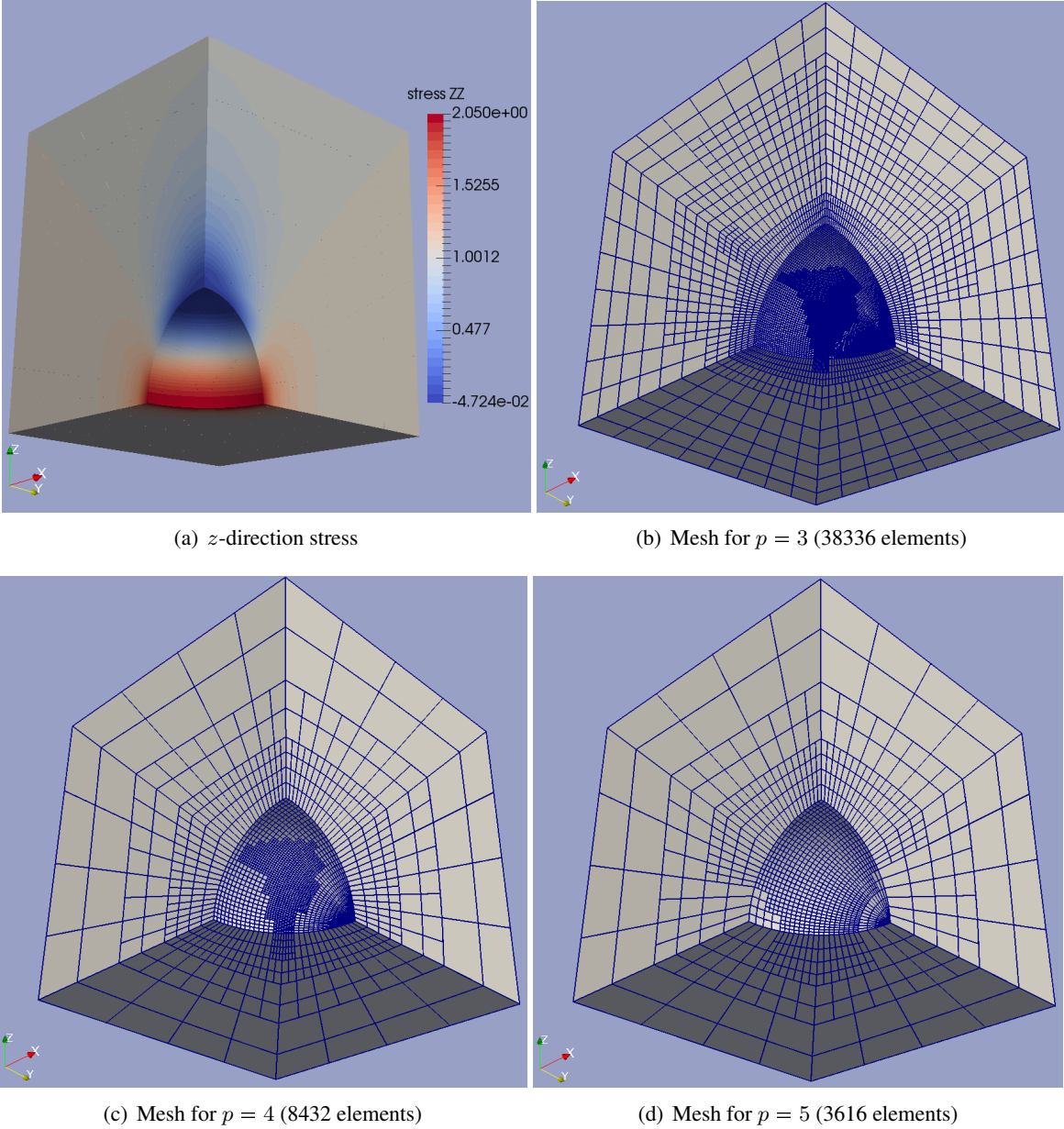


Fig. 7.8: The z -direction stress and the refined meshes corresponding to the different polynomial degrees p for the cube with a hole example.

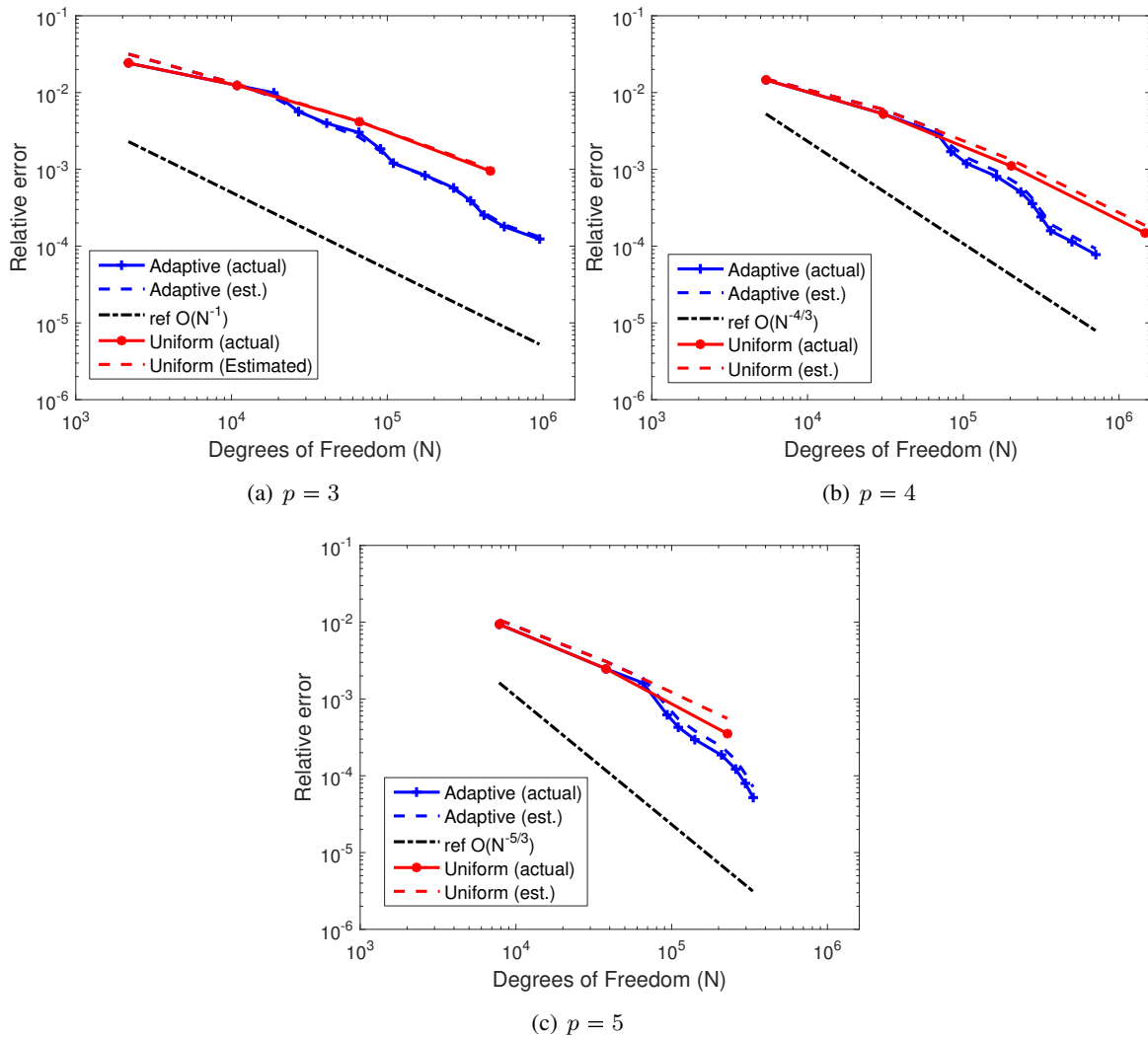


Fig. 7.9: Relative error (exact and estimated) in energy norm vs. the number of degree of freedom for $p = 3, 4$ and 5 , using adaptive and uniform refinements for the cube with a hole example.

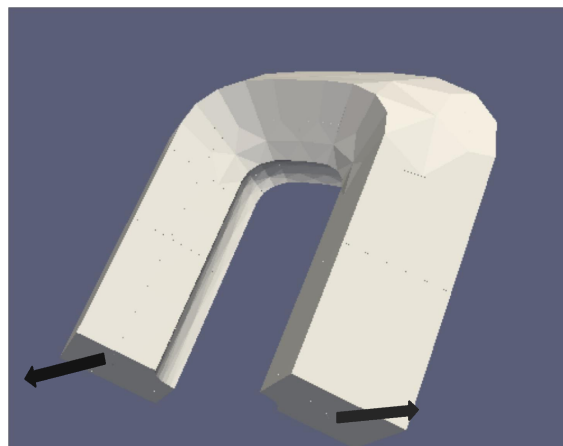


Fig. 7.10: Geometry of elastic horseshoe subjected to equal and opposite in-plane flat edge displacements on the soles of the shoe.

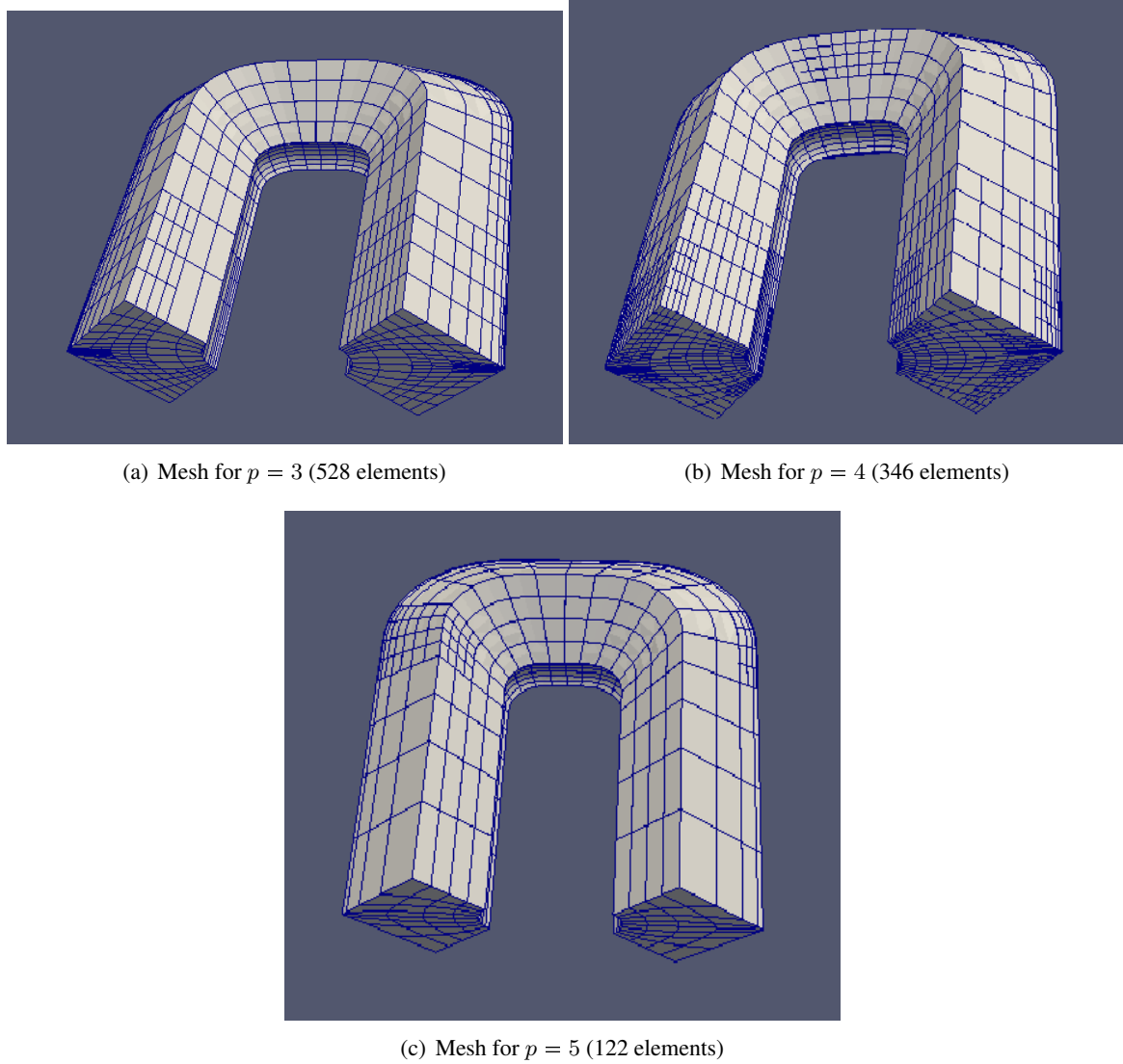


Fig. 7.11: The refined meshes corresponding to the different polynomial degrees p for the horse shoe example.

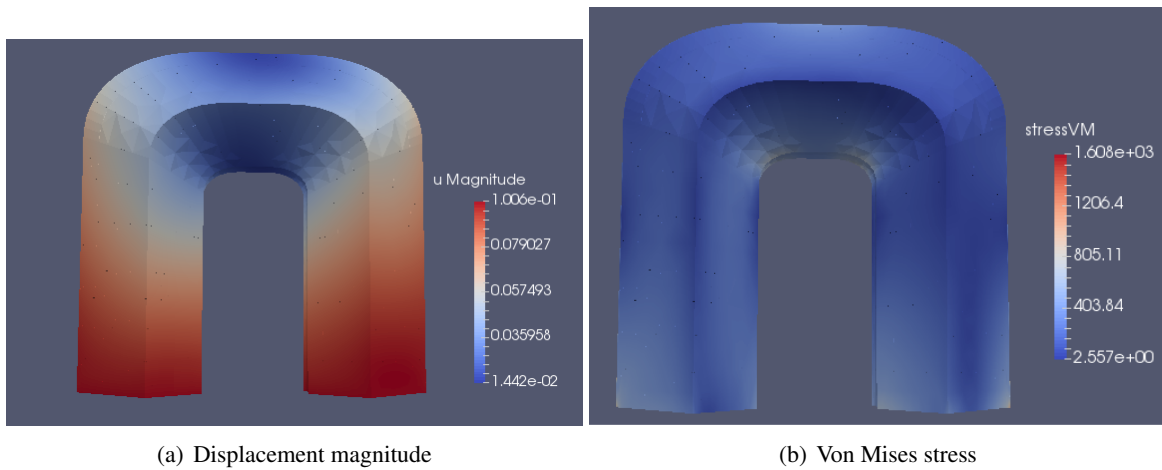


Fig. 7.12: The von-mises stress and the displacement contour plot for $p = 3$ for the horse shoe example.

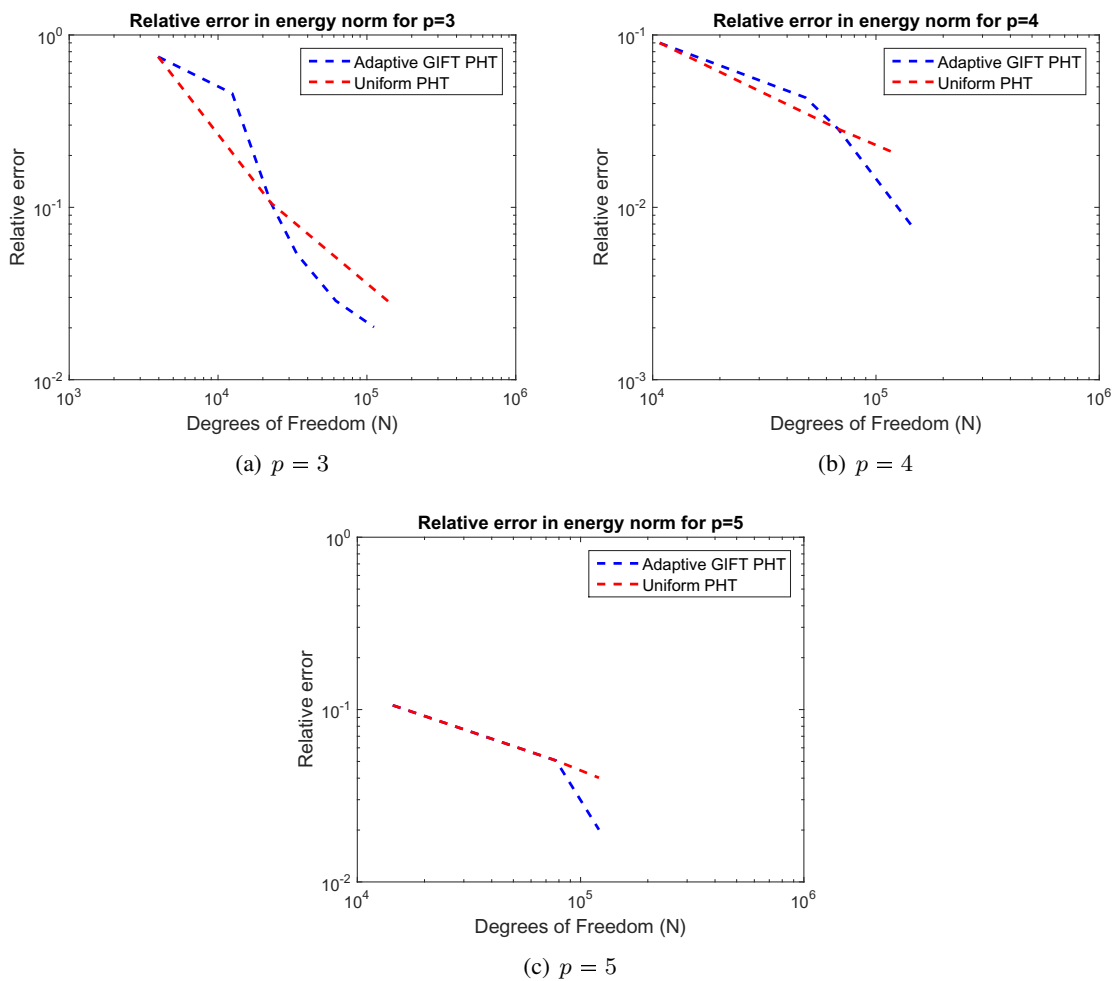


Fig. 7.13: Relative error (exact and estimated) in energy norm vs. the number of degree of freedom for $p = 3, 4$ and 5 , using adaptive and uniform refinements for the horse shoe example.

Chapter 8

Multi-patch analysis of adaptive GIFT

In this chapter, we will discuss multipatch analysis. Multipatch is necessary to handle more general geometries and using GIFT it is possible to handle C^0 continuity. Any knot that has repeated degree, p times resulting a C^0 continuity line can be modeled by splitting the parameter space into patches along the knot line. There are several approaches to handle connectivity between the patches and we will discuss the approach we considered here to handle several patches in geometry.

8.1 Handling geometries in multipatch

Multipatch analysis can be performed using GIFT as the PHT-spline basis function can handle more than C^0 -continuous basis functions across the elements comparing to classical Finite Element Analysis (FEA). Using this advantage it can be adjustable to the smoothness across the knot spans in between the element patches. Multipatch analysis is also necessary because the vast majority of CAD models comprise multipatch geometries. There are several ways of handling multipatch geometries. They are:

- Master-slave approach [44].
- Weakly coupling based on Nitsche's method [97, 104].

8.1.1 Multi-patch domains

It can be defined that for complex domains, it is generally desirable to consider multiple patches, each of which are associated with a different parameter domain. While several techniques for weak coupling have been proposed (see for example [97, 104]), the simplest and most robust way to couple multiple conforming patches is by identifying the degrees of freedom at the patch interfaces. Due to the hierarchical structure of the meshes which allow local refinements, it is relatively easy to obtain conforming patches by matching the knot-vectors along the common boundaries. This procedure ensures that C^0 continuity is strongly enforced along the patch boundaries, with a minimal amount of extra refinement or computational cost. Other proposed methods to couple multi-patch geometries can also be used without additional difficulties.

8.1.2 Multi-patch approach by knot insertion 2D PHT-mesh

In our approach as it can be mentioned before that any knot that is repeated p times resulting a C^0 lines can be modeled by splitting the parameter space into patches along the knot vector. By doing splitting the patches, we can make conforming patches by knot insertion. The approach we considered here are:

- Define the patch boundaries and configure which patch is connected by which neighboring patches with the numbering of edge orientation.
- At first check the conformity between the patches.
- If the patches are not conforming then insert a cross in case of PHT-spline to make the conformity in between the patches.
- Match up the global node indices on the neighboring patches.

Now we will describe in details the implementation of the multi-patch approach by knot insertion what we have considered for our multi-patch analysis.

8.1.2.1 Patch boundary configuration

At first to do multi-patch analysis we need to define the patch boundary connectivity and also which patch boundary edge is connected to which other patch boundary edge. It is also needed to mention the edge orientation to define the patch boundary edges to connect the patches. Here in Fig. 8.2 we will show a simple example which has also used in the numerical example 8.4.4, the patch connectivity and defining the numbering of the orientation of patch boundaries. The direction of the parameter space has to match across the patch boundaries and in Fig. 8.1 we see that each patch has 2×2 elements on the coarsest mesh, the element numbers should increase for both patches or decrease for both patches in the same direction. Note that, this is handled through an additional flag described by [51]. In the case of numbering the patches, patch boundaries cannot be set in any random order. In particular, there needs to be an ordering according to which the global node numbers are assigned.

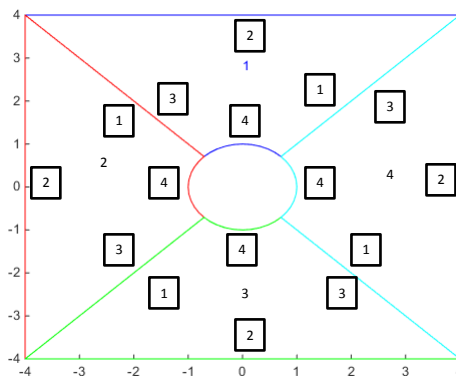


Fig. 8.1: Example of patch boundary configuration. Here in the box the numbering for edges are: 1-bottom edge, 2-right edge, 3-top edge, 4-left edge. Patch numbering is shown without box.

Tab. 8.1: Example of patch boundary connection configuration. Edge numbering: 1-bottom edge, 2-right edge, 3-top edge, 4-left edge.

Patch A	Patch B	Connected edge number of patch A	Connected edge number of patch B
1	4	1	3
4	3	1	3
[1,3]	2	[3,1]	[1,3]

In Tab. 8.1 we can see according to the example in Fig. 8.1 that patch 1 is defined as **Patch A** and patch 4 is defined as **Patch B** which are connected by the edge 1(bottom) of patch 1 and edge 3(top) of patch 4. Next patch 4 is defined as (**Patch A**) and patch 3 defined as (**Patch B**) are connected by edge 1(bottom) of patch 4 and edge 3(top) of patch 3. Here, for the last patch configuration to complete a circuit we need to define patch 1 and 3 together as **Patch A** which are connected to patch 2 as **Patch B** with the corresponding edges 3(top) and 1(bottom) for connecting patches 1 and 2 and to connect the patches 3 and 2 is done by the edges 1(bottom) and 3(top) of the respective patches.

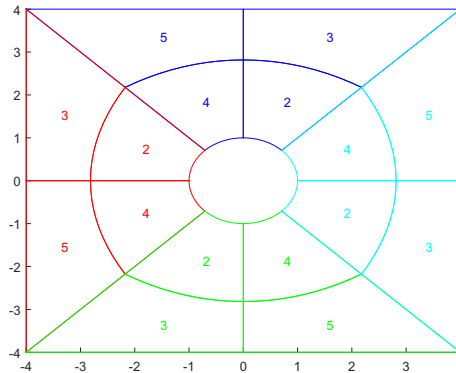


Fig. 8.2: Example of patch boundary configuration in the coarsest mesh where every patch consists of 2×2 elements. Here the numbering for edges are: $[2, 3]$ -bottom edge, $[3, 5]$ -right edge, $[4, 5]$ -top edge, $[2, 4]$ -left edge.

In Fig.8.2, we insert one cross in every patch in parametric PHT coordinate and on the coarsest mesh it can be noticed that the side which has a number of pair $[2, 3]$ along the edge is the bottom side in the parameter space, the side that has a number of pair $[3, 5]$ along the edge is the right side, the side that has a number of pair $[4, 5]$ is the top edge and the side that has a number of pair $[2, 4]$ is the left side edge. This configuration makes it easier to impose boundary conditions and check the orientation of the boundary edges in the patches.

8.1.2.2 Checking the conformity between the patches

At the first step of this multi-patch approach, we need to check if the parametric coordinates in between the patches are conforming or not. For example in Fig. 8.3 we have 2 patches which are defined as patch **A** and patch **B** respectively. Here we can see that patch **A** consists of 4 elements and the right side has 2 edges $[0, 0.5]$ and $[0.5, 1]$ in parametric coordinates. Patch **B** consists of only 1 element and on the left side it has 1 edge in parametric coordinate $[0, 1]$. So it is necessary to check at first the patch boundaries (here right side for patch **A** and left side for patch **B**) whether the edges match the coordinates of the boundaries in between the patches or not. If it does not match the edges of the patch boundaries then for making the patches conforming it will insert a cross at the patch **B** to make a conforming patch in between patch boundaries **A** and **B**.

8.1.2.3 Make conforming patch by cross insertion

After checking in every patch boundaries of several patches, it marks which patch boundaries are conforming and non-conforming. When it finds non-conforming patch boundaries in between patches, it inserts a cross to make conforming in between the patches at the patch boundary. Here in Fig. 8.4, we can see that as it finds patch **B** boundary does not match with the patch **A** boundary, it inserts a cross in patch **B** to make conforming patches.

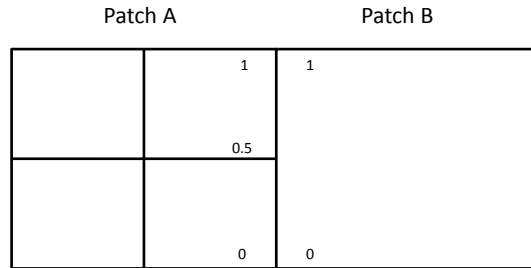


Fig. 8.3: Example of non conforming patches.

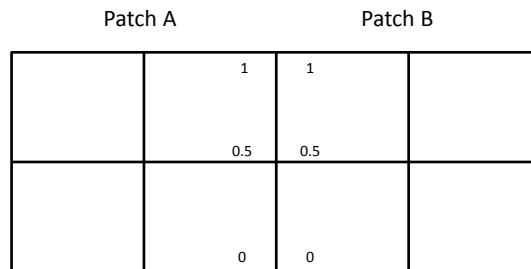


Fig. 8.4: Making conforming patches by a cross insertion at patch **B**.

So after insertion of a cross in patch **B**, the right side of patch **A** and left side of patch **B** consist of 2 edges $[0, 0.5]$ and $[0.5, 1]$ in parametric coordinates.

8.1.2.4 Match up the global node indices on the neighboring patches

To match up the patch boundaries between the neighboring patches we need to make changes the numbering at the global node indices in case of using PHT-spline for multi-patch analysis. We need to connect two conforming patches by changing the global node indices entry in between the patches. In Fig. 8.5 we can see the global node indices numbering in between patches patch **A**, **B**, **C** and **D** respectively.

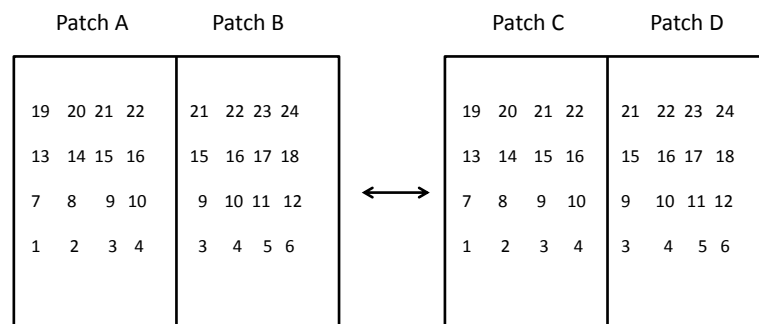


Fig. 8.5: Global node indices numbering in between patches.

At first, it checks the nodal indices along the patch boundary of patches **B** and **C**. Here the nodal indices at the edge of patch **B** are $[6, 12, 18, 24]$ and in patch **C** are $[1, 7, 13, 19]$. As the numbering of nodal indices is the same in these edges of two patches, so they are conforming patches, otherwise, it will be non-conforming patches. It creates a node pattern for the patch **C** to make a conforming global node index and to glue together with patch **B**.

8.1.3 Multi-patch approach by knot insertion of 3D PHT-mesh

The multi-patch approach by knot insertion of 3D PHT-mesh is the extension of 2D PHT-spline mesh. Here instead of edges, the patches are connected by faces.

8.1.3.1 Patch configuration 3D PHT-mesh

The patch boundary configuration for 3D PHT-mesh is similar to the 2D configuration of PHT-mesh. Here in Fig. 8.6 it can be seen that the Patch **A** and Patch **B** are connected by faces 2(left) and 4(right) while in 2D cases of PHT-mesh the patches are connected by edges. The numbering of the faces are 1 for the front face, 2 for left face, 3 for back face, 4 for right face, 5 for down face and 6 for the top face.

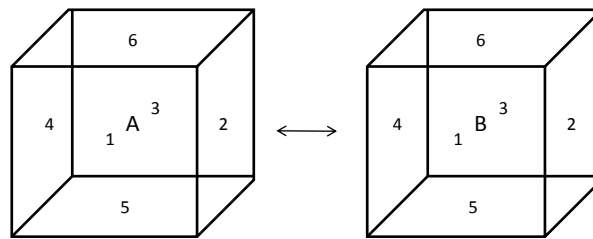


Fig. 8.6: Patch boundary configuration of 3D PHT-mesh. Here the numbering for faces are: 1-front face, 2-left face, 3-back face, 4-right, 5-down face, 6-up face.

8.1.3.2 Make a conforming patch of 3D PHT-mesh by knot insertion

For the 3D PHT-mesh similar to 2D PHT-mesh at first, it checks the mesh conformity in between the patches **A** and **B**. For example in Fig. 8.7 at first it checks the two 3D patches whether the faces in between the two patches **A** and **B** the vertices of each edge match or not. If the configuration of vertices in between two patches mismatch Fig. 8.7(a) then it inserts a cross to the corresponding patch to make the patches conforming (see in Fig. 8.7(b)).

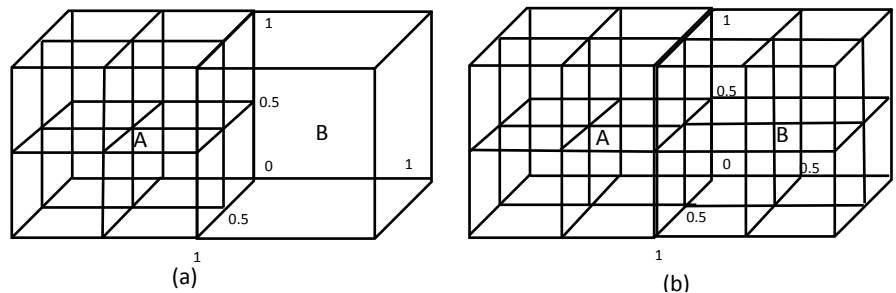


Fig. 8.7: (a) Patch **A** and **B** are non conforming patches, (b) Making conforming patches by a cross insertion at patch **B**.

8.1.3.3 Match up the global node indices on the neighboring patches for 3D PHT-mesh

After making the patches of 3D PHT-mesh conforming, the next step is to match the global node indices which are similar to the 2D case. But here instead of checking nodal indices of edges, it checks the nodal indices of faces in between the patches. It creates a node pattern for the two 3D

patches to make conforming global node indices when it finds the two patches are conforming.

8.2 GIFT 2D multi-patch analysis using PHT-spline with the approach by knot insertion

We will describe the process how the GIFT method has been applied for doing 2D multi-patch analysis using PHT-spline with the approach by knot insertion.

- Get the initial 2D GIFT NURBS geometry with coarse mesh associated with control points, knot vectors, Bézier extraction operators, etc for using GIFT mapping.
- Initialize the PHT-spline geometry on a coarse mesh.
- Define the connectivity in between the patches by the edge orientation numbering (1-bottom, 2-right, 3-up, 4-left).
- Check the conformity between the patches.
- If it is found a non-conforming patch in between patch boundaries then it makes conforming patches by knot insertion (cross insertion in case of PHT-spline).
- Connects two conforming patches by changing the global node indices.
- Do the assembly of the linear system of equations and impose the Dirichlet and Neumann boundary conditions.
- Solve the system of equations.
- Estimate the error using a recovery-based error estimator for every patch (see 4.5).
- In adaptive refinement steps (see 4.4.1), it does the local refinement in the marked patches (see 4.5.3) using PHT-spline cross insertion at the solution field.
- It keeps doing the refinement until the **global error estimator** < **target relative error**.

8.3 GIFT 3D multi-patch analysis using PHT-spline with the approach by knot insertion

We extend the 2D adaptive GIFT PHT-spline multi-patch analysis for the 3D case. For the 3D GIFT PHT-spline multi-patch analysis we define the patch boundaries by face numbering and check patch conformity in between faces of every 3D PHT patches instead of edges in the 2D case. The algorithm here for 3D PHT-spline multi-patch analysis is similar to 2D which is described below:

- Get the initial 3D GIFT NURBS geometry with coarse mesh associated with control points, knot vectors, Bézier extraction operators, etc to do GIFT mapping.
- Initialize the 3D PHT-spline geometry on a coarse mesh.
- Define the connectivity in between the patches by face orientation numbering (1-front, 2-left, 3-back, 4-right, 5-down, 6-up).
- Check the conformity between the 3D patches. At first, needs to check that the patches are conforming or not and if needed, makes them conforming through mesh refinement.

- If it is a non-conforming patch in between patches then it makes conforming patches by knot insertion (cross insertion in case of 3D PHT-spline which makes 8 octuples).
- Connects two conforming patches by changing the global node indices. Shift the basis function indices in nodes pattern so that globally the patches become conforming.
- Do the assembly of the linear system of equations and impose the Dirichlet and Neumann boundary conditions.
- Solve the system of equations.
- Estimate the error using a recovery-based error estimator for every patch of octuples(see 4.5).
- In adaptive refinement steps (see 4.4.1), it does the local refinement in the marked patches (see 4.5.3) using PHT-spline cross insertion at the solution field.
- It keeps doing the refinement until **global error estimator** < **target relative error**.

8.4 2D Numerical Examples of multi-patch analysis

In this section, we will demonstrate several benchmark numerical examples using GIFT method for multi-patch analysis with NURBS for geometry and PHT-splines in the solution field to do adaptive local refinement. We will show numerical examples like Plate with a circular hole, L-shaped wedge and Edge crack using several patches where it can handle C^0 continuity in between the patches.

8.4.1 Plate with a circular hole

This is an example of an infinite plate with a circular hole under constant in-plane tension. It involves two conforming patches where initial coarse mesh has been taken with NURBS geometric description, while solution space has been chosen by PHT-spline using the GIFT mapping. In Fig. 8.8 the geometry, boundary conditions and material properties of the problem along with the exact solution have been shown. It has an advantage of symmetrical geometry and for this reason, only a quarter of the model is considered here with 2 conforming patches. The exact solutions of displacements and stresses are given below:

Exact displacements:

$$\begin{aligned}
 u_r(r, \theta) &= \frac{(1 + \nu)T_x}{E} \left(\frac{r \cos \theta}{1 + \nu} + \frac{2R^2 \cos \theta}{(1 + \nu)r} + \frac{R^2 \cos 3\theta}{2r} - \frac{R^4 \cos 3\theta}{2r^3} \right) \\
 u_\theta(r, \theta) &= \frac{(1 + \nu)T_x}{E} \left(-\frac{\nu r \sin \theta}{1 + \nu} - (1 - \nu) \frac{R^2 \sin \theta}{(1 + \nu)r} + \frac{R^2 \sin 3\theta}{2r} - \frac{R^4 \sin 3\theta}{2r^3} \right)
 \end{aligned} \tag{8.1}$$

Exact stresses:

$$\begin{aligned}
 \sigma_{rr}(r, \theta) &= \frac{T_x}{2} \left(1 - \frac{R^2}{r^2} \right) + \frac{T_x}{2} \left(1 + 3\frac{R^4}{r^4} - 4\frac{R^2}{r^2} \right) \cos 2\theta \\
 \sigma_{\theta\theta}(r, \theta) &= \frac{T_x}{2} \left(1 + \frac{R^2}{r^2} \right) - \frac{T_x}{2} \left(1 + 3\frac{R^4}{r^4} \right) \cos 2\theta \\
 \sigma_{r\theta}(r, \theta) &= -\frac{T_x}{2} \left(1 + 2\frac{R^2}{r^2} - 3\frac{R^4}{r^4} \right) \sin 2\theta
 \end{aligned} \tag{8.2}$$

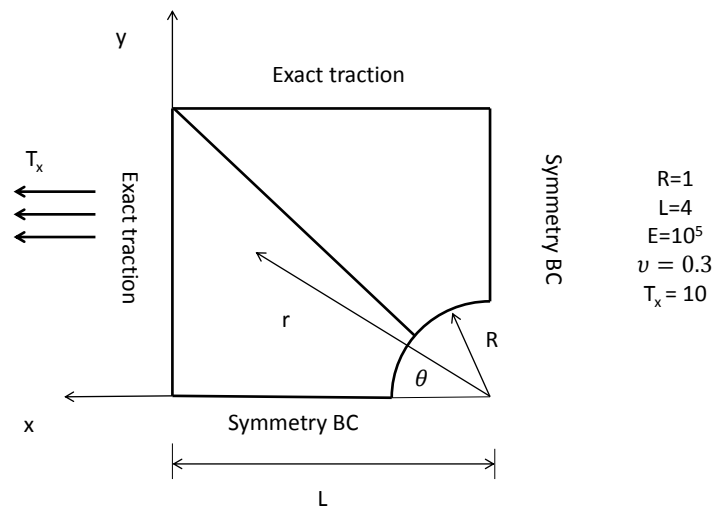
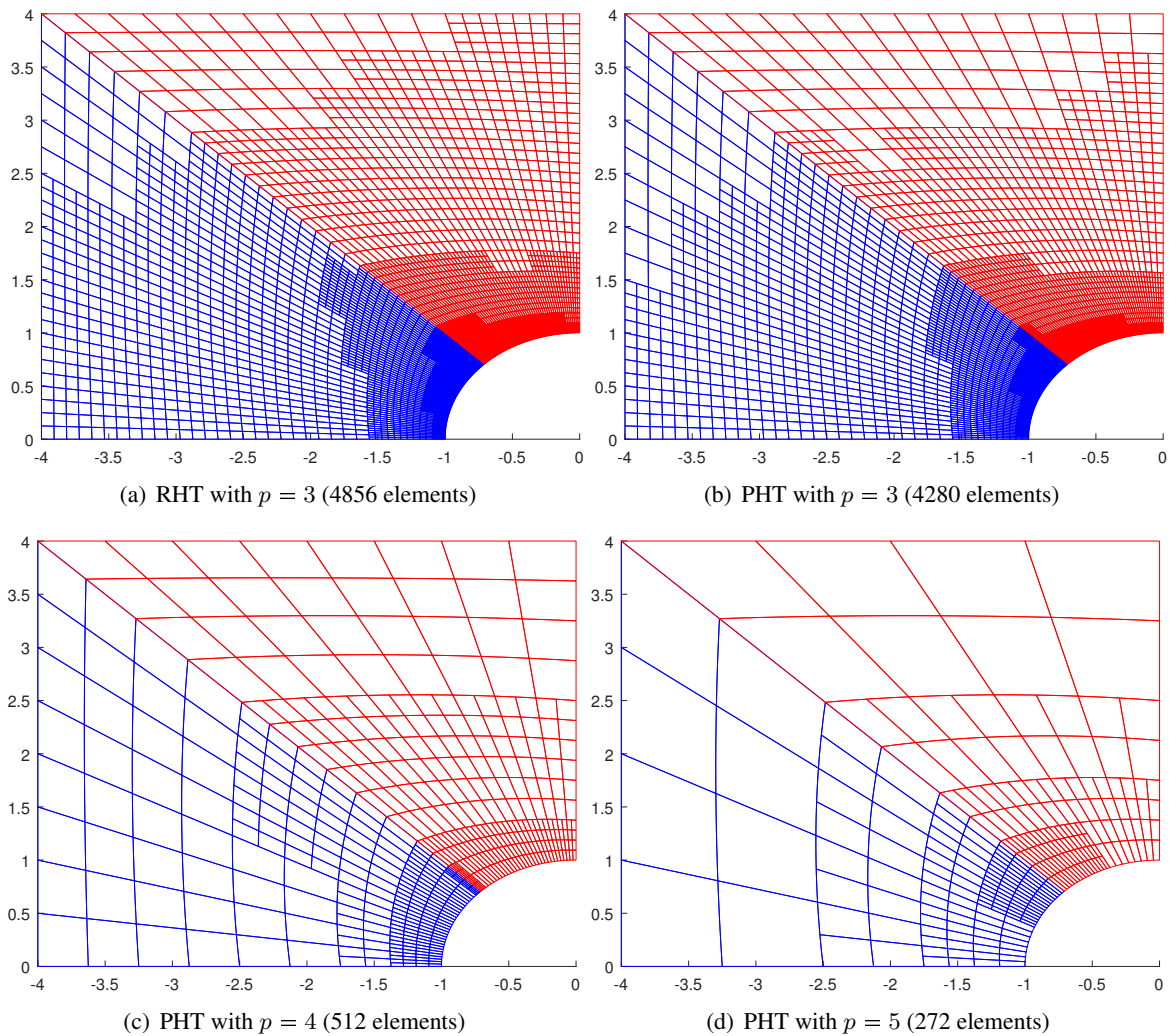


Fig. 8.8: Problem description for the plate with a circular hole example

Fig. 8.9: The refined meshes for different polynomial degrees p .

The refined meshes for the polynomial degrees 3, 4 and 5 are shown in Figure 8.9, where the refinement is stopped when the relative error in energy norm reaches 10^{-5} . The Dörfler marking

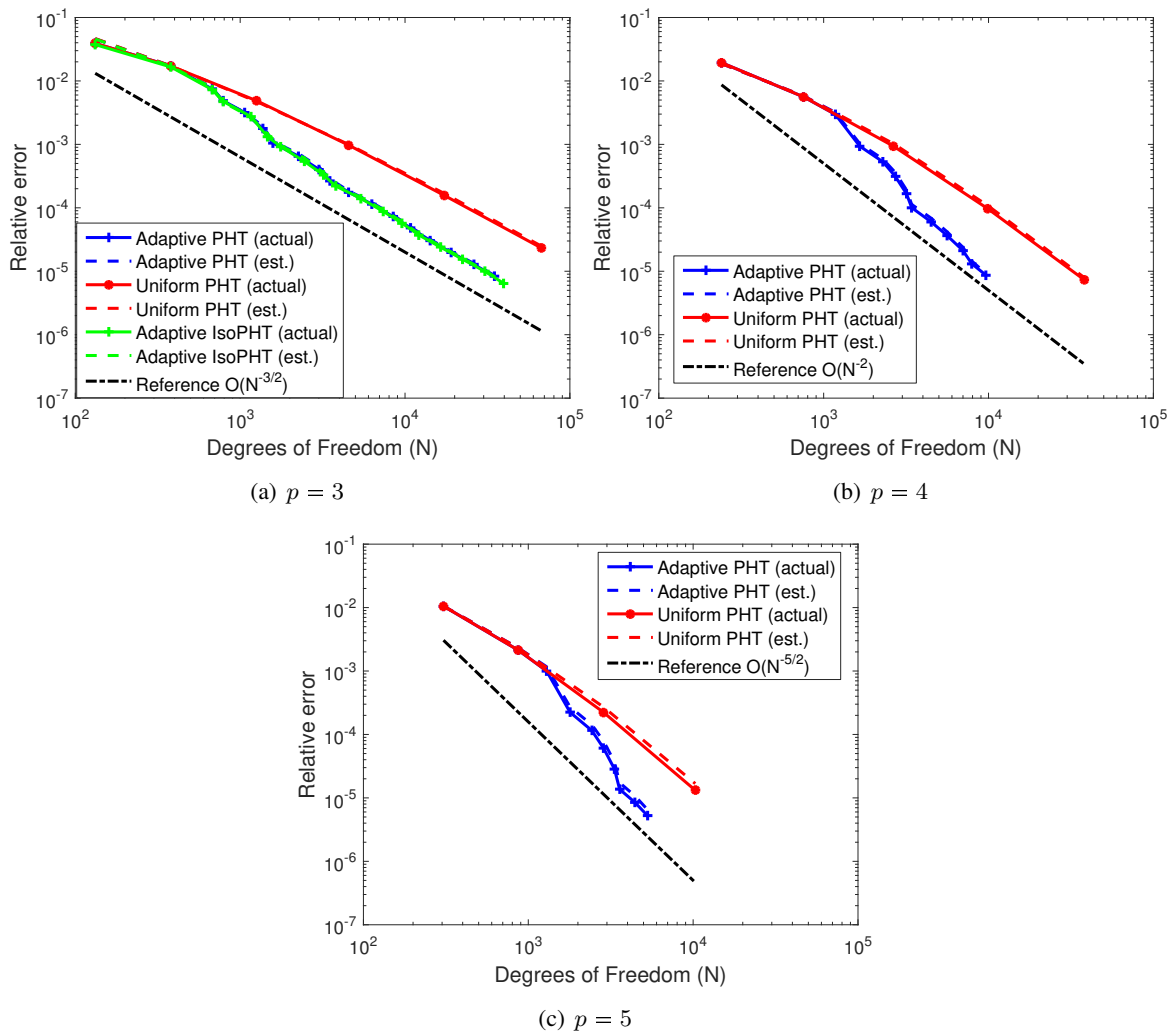


Fig. 8.10: Relative error (exact and estimated) in the energy norm vs. the number of degree of freedom for $p = 3, 4$ and 5 , using adaptive and uniform refinements for the plate with a hole example.

parameter has the value $\theta = 0.75$, meaning that at each step the elements contributing to at least 75% of the error are refined. We note that for the cubic degree, the resulting refinements for polynomial spline basis with a rational mapping and the rational isoparametric splines are similar. For higher polynomial degrees, the meshes have considerably fewer elements and the number of refinement steps for reaching the target relative error is also significantly smaller.

In Figure 8.10, the convergence plots for the different polynomial degrees with uniform and adaptive refinements are shown for the energy norm. It can be seen that for the cubic degree, the results for the rational basis and the polynomial splines are almost identical. We observe good agreement between the estimated error obtained using the recovery-based estimator and the exact error computed using the analytical solution. For the adaptive refinement and the last refined mesh, the effectivity indices (defined as the ratio of the estimated error and the actual error) are 1.03 for $p = 3$, 1.06 for $p = 4$, and 1.24 for $p = 5$. We also note that the optimal convergence rate in terms of degrees of freedom N , i.e. $O(N^{-p/2})$ is obtained for both uniform and adaptive refinements.

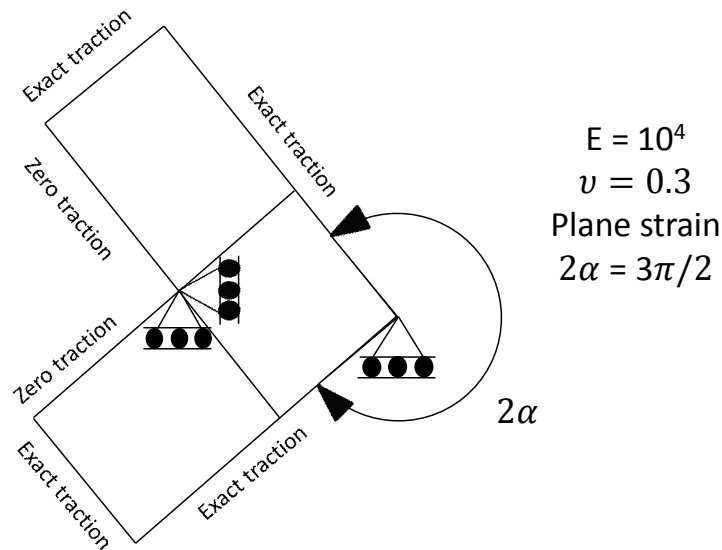


Fig. 8.11: Geometry, boundary conditions and material properties for the L-shaped wedge problem.

8.4.2 L-shaped wedge

The next example was considered was the L-shaped wedge taken from [116], which exhibits singularity at the wedge apex. The analytical solution to this problem is given by [117] where a wedge angle of $2\alpha = 3\pi/2$ was used. We have considered the mode 1 loading case, exact tractions were applied along all faces with appropriate displacement constraints as shown in Fig. 8.11. Material properties $E = 1e4$ and $\nu = 0.3$ were used under plane strain conditions.

The domain is discretized using 3 PHT patches, each of which is a linear transformation (dilation and rotation) of a unit square. In this example, the performance of the error estimator can be analyzed for a non-smooth solution, without the possible interference from a non-linear geometric mapping. For refinement, a Dörfler refinement scheme was used, with the parameter $\theta = 0.5$. A lower θ and therefore more refinement steps are needed for this example to obtain a mesh with as few elements as possible due to the error pollution effect from the singularity at the re-entrant corner. The refined meshes for polynomial degrees 3, 4 and 5 are shown in Figure 8.12, where the refinement was stopped when the estimated relative error reached 10^{-5} .

Similarly to the previous example, for higher polynomial degrees, meshes with significantly fewer elements are required to reach the given precision. In particular, much more graded meshes are obtained for $p = 3$ and $p = 4$, with most of the refinements concentrated around the re-entrant corner.

The convergence plots for this example are shown in Figure 8.13. For uniform refinement, a non-optimal rate of convergence is observed, as expected due to the non-smooth solution. Because the initial mesh is very coarse, with just 2×2 elements in each patch, uniform refinement is selected in the first two steps. Afterwards, most of the refinement is concentrated near the singularity at the re-entrant corner. Finally, the refinement zone is progressively enlarged with some refinements farther away in the domain as the error decreases. We observe that for sufficiently fine meshes, an optimal convergence rate is obtained for adaptive refinements. The effectivity indices for the error estimator approach 1 as the mesh is refined, reaching 1.003 for $p = 3$, 1.04 for $p = 4$ and 1.06 for $p = 5$. For uniform refinement, the effectivity of the estimator is significantly worse even on the most refined meshes as the error near the singularity is not as well controlled.

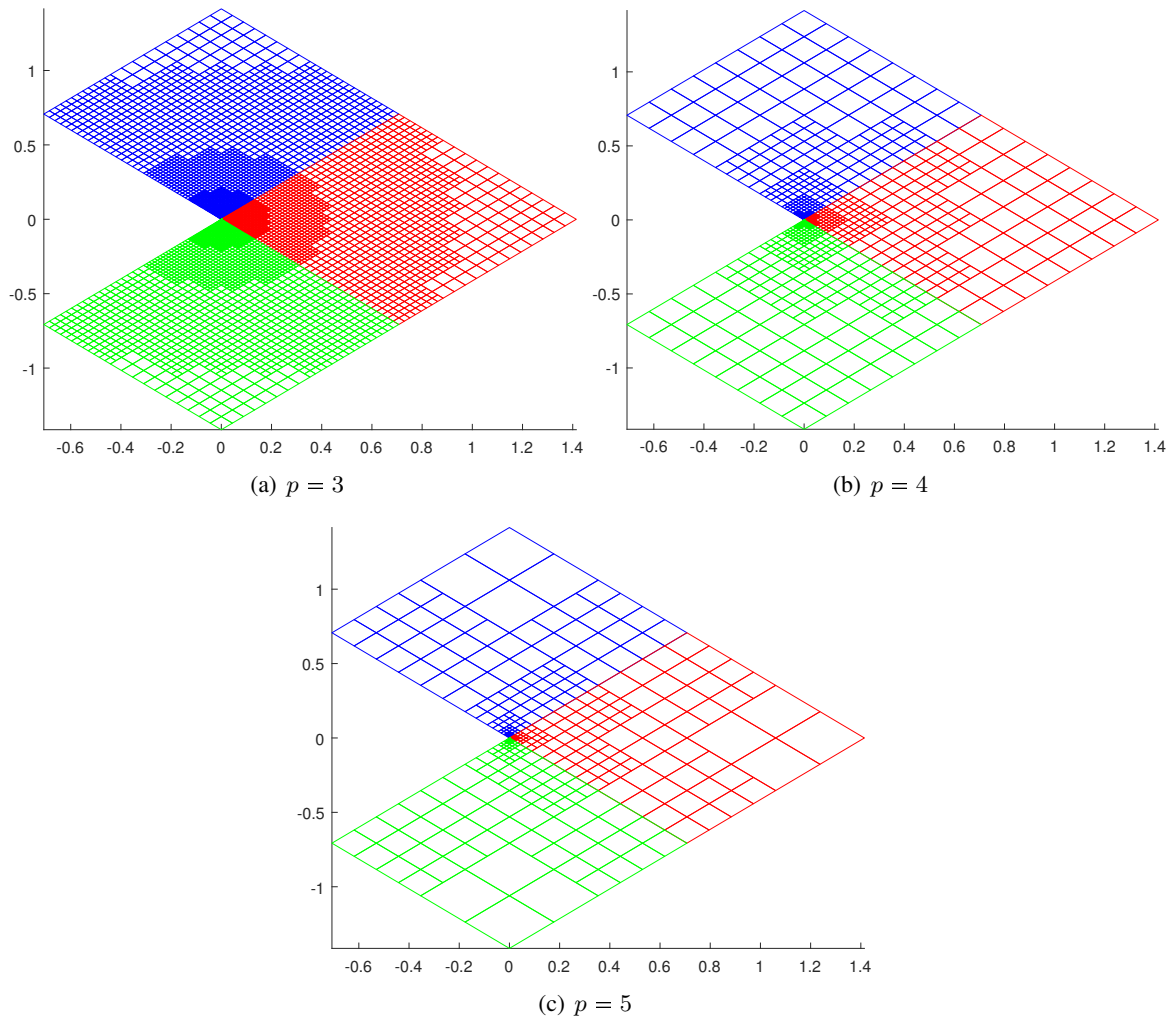


Fig. 8.12: The refined meshes for different polynomial degrees p in the L-Shaped wedge example.

Computational times for the corresponding adaptive GIFT and uniform PHT-spline refinements for each polynomial degree are shown in Fig. 8.14. The computational times shown include the post-processing stages such as error estimation and refinement. We note the substantial speed-up obtained by adaptive GIFT PHT-spline refinement in the case where the approximation error is dominated by a singularity in the solution.

8.4.3 Edge crack using Multi-patch

In this example, we will describe an edge crack problem discretized with 4 patches. In Fig. 8.15 patch 1 is connected to patch 2, patch 2 is connected with patch 3, and patch 3 is connected with patch 4 where patches 1 and 4 are not connected to make an open edge crack. At the end of patches 1 and 4, we can see the crack tip. The edge crack plate dimension $b \times 2h$ is loaded by tensile stress at the top edge with $\sigma = 1$. We impose zero displacements at the crack tip (in x, y -direction) and zero-displacement in the y -direction at the midpoint of the right edge (as in the L-Shaped wedge example). The material parameters are $E = 10^3$ and $\nu = 0.3$. A plane strain condition is assumed. Here a is the crack length, b is the plate width and h is the plate height.

In Fig. 8.16 the meshes at the different hierarchical level has been plotted using degree $p = 3$, C^1 continuity PHT-spline for the Edge crack where in the first mesh the geometry is represented with

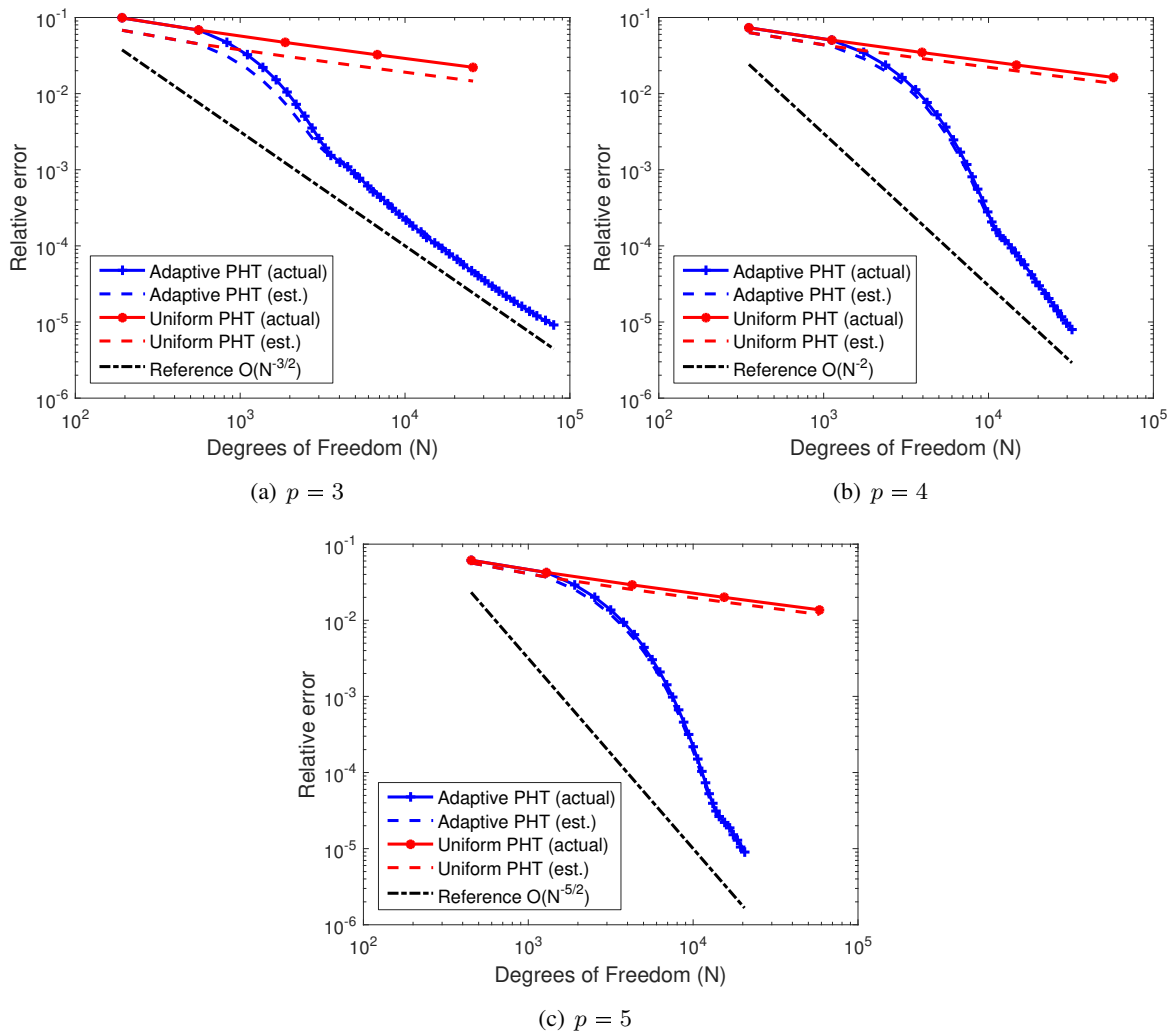


Fig. 8.13: Relative error (exact and estimated) in energy norm vs. the number of degree of freedom for $p = 3, 4$ and 5 , using adaptive and uniform refinements for the L-Shaped wedge example.

linear NURBS and in level 1 refinement is performed with uniform refinement. Adaptive refinement is activated from level 2 and due to singularity at the crack tip, it performs the refinement towards the crack singularity.

In Fig. 8.17 shows contour plots for the Edge crack with the multi-patch configuration of different displacements and stress components. It has shown that at the crack tip it has singularity and using the adaptive refinement with PHT-spline, it can capture the stress concentration from level 2 to 6 (see also Fig. 8.16).

The convergence plots for this example are shown in Figure 8.18 wherewith uniform refinement, a non-optimal rate of convergence is observed, as expected due to the non-smooth solution for edge crack. Because the initial mesh is very coarse, with just 2×2 elements in each patch, uniform refinement is selected in the first two steps. Afterwards, most of the refinement is concentrated near the singularity at the crack tip. Finally, the refinement zone is progressively enlarged with some refinements farther away from the crack tip as the error decreases. We observe that for sufficiently fine meshes, an optimal convergence rate is obtained for adaptive refinements. The effectivity indices for the error estimator approach to 1 as the mesh is refined, reaching 1.1053 for $p = 3$, but

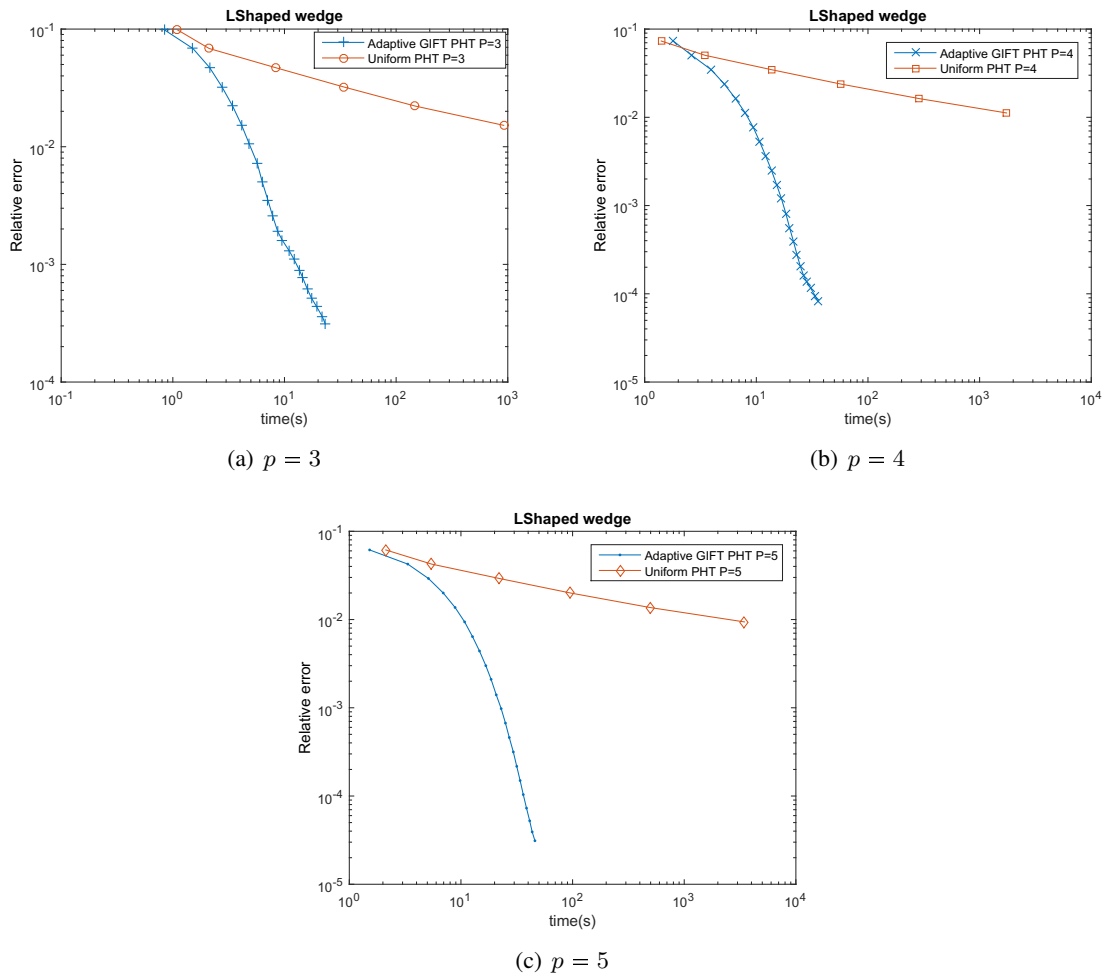


Fig. 8.14: Relative error (exact) in energy norm vs. computation time for $p = 3, 4$ and 5 , using adaptive GIFT and uniform refinements using PHT-spline for the L-shaped wedge example.

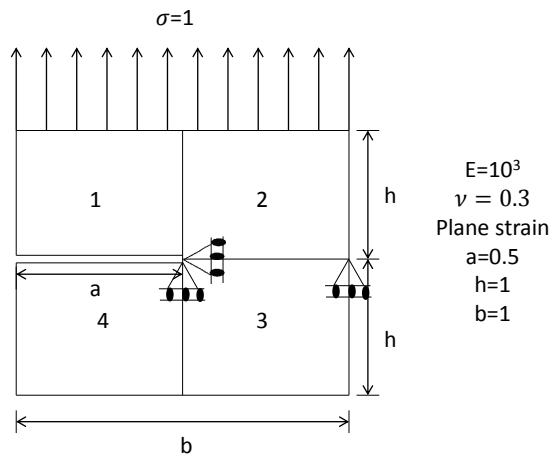


Fig. 8.15: Geometry, boundary conditions and material properties for Edge crack multipatch problem. decreases for higher-order 0.926 for $p = 4$ and 0.7845 for $p = 5$. Here, we can observe that the estimated and actual error are becoming close as it does more adaptive refinement. So similar to the previous L-shaped wedge example, this edge crack problem also shows that using the higher-order

PHT-splines we get much benefit in adaptive refinement compared to uniform refinement.

8.4.4 L-shaped Bracket

In this numerical example, we will show a complex geometry with multiple patches and compare the estimated energy norms with different orders and continuity. In Fig. 8.19 an L-shaped bracket with 18 patches has been shown with the dimensions and boundary conditions. 4 symmetric brackets each of them has 4 patches are connected at the top right and bottom left with a 2 patches quarter plate hole geometry at the middle. The Young modulus, $E = 10^5$ and Poisson ratio $\nu = 0.3$ with a fixed edge boundary condition at the bottom edge of the bottom patch Bracket and in the first patch Bracket at the right-hand side edge a traction $t = 2$ imposed in y -direction.

The domain is discretized using 18 PHT patches, each of which is a linear transformation (dilation and rotation) of a quarter and full circular geometry. In this example, the performance of the error estimator can be analyzed for a non-smooth solution, without the possible interference from a non-linear geometric mapping. For refinement, a Dörfler refinement scheme was used, with the parameter $\theta = 0.75$. A lower θ and therefore more refinement steps are needed for this example to obtain a mesh with as few elements as possible due to the error pollution effect from the singularity at the L-shaped corner. The refined meshes for polynomial degrees 3, 4 and 5 are shown in Fig. 8.20, where the refinement was stopped until the estimated relative error reached 10^{-3} .

In Fig. 8.21 the displacement and stress contour plots are shown. As the traction is imposed at the right edge of the first bracket at the corner of the right edge stress concentration is higher and also at the hole of the brackets. Adaptive mesh refinement performs at the particular stress singularity at the corner of the brackets and holes of each bracket. After a few refinement steps, the error is dominated by the corner stress singularities near the bottom fixed boundary.

The convergence plots for $p = 3, 4, 5$ are presented in Figure 8.22. We note that for this case only the estimated errors are shown, as an analytical solution is not available. It can be seen that adaptive GIFT PHT-spline refinement results in smaller errors compared to uniform refinement. Because of the stress singularities near the corner of the L-shape and the circular zone, the optimal rate of convergence is not achieved in this example. However, it would be expected that further refinement would result in a more drastic error reduction as the stress singularities are resolved in adaptive refinement, similarly to the 2D L-shaped example (see in 8.4.2).

We also validate the results using Matlab PDEtoolbox and it can be seen from Fig. 8.23 that at the corner near the bottom fixed boundary stress singularities appear and the solution in between FEM and adaptive PHT-spline is in good agreement in this case.

8.5 3D Numerical Examples of multi-patch analysis

Here we present some 3D numerical examples of adaptive GIFT PHT-spline with multi-patch geometry. We will show an example of a 3D edge crack of mode I and mode II loading conditions. Another example is the penny shaped crack and the last example is a connecting rod with complex geometry multi-patch.

8.5.1 3D edge crack with Mode I and Mode II loading condition

In this numerical example, we will show an edge crack domain with PHT-splines where asymptotic Mode I and Mode II loading conditions are considered with exact solutions. In Fig.8.24 an infinitely large linear elastic solid, the 'Mode I' loading condition has been shown which signifies stress intensity factor K_I quantifies the crack opening displacements and stresses. The 'Mode II' stress intensity factor characterizes in-plane shear displacements and stress in Fig. 8.25. Here the dimension of the geometry is considered: length, $L = 1$, width $W = 1$, height $H = 1$ and crack length $a = 0.5$. The applied force at infinity is considered $\sigma = 1$. The material conditions are: Young's modulus, $E = 1.0 \times 10^5$, Poisson's ratio $\nu = 0$ and plane strain condition. The solid is loaded at infinity.

For both mode I and mode II loading edge crack are discretized with 4 patches. In Fig. 8.24(b), the patch 1 is connected to patch 2, patch 2 is connected with patch 3, and patch 3 is connected with patch 4 where patches 1 and 4 are not connected to make an open edge crack.

Crack tip fields are most conveniently expressed in terms of cylindrical-polar coordinates (r, θ, z) with an origin at the crack tip. The displacement and stress near the crack tip can be characterized by three numbers K_I, K_{II}, K_{III} , known as 'stress intensity factors'. By definition:

$$\begin{aligned} K_I &= \lim_{r \rightarrow 0} \sqrt{2\pi r} \sigma_{22} \\ K_{II} &= \lim_{r \rightarrow 0} \sqrt{2\pi r} \sigma_{12} \\ K_{III} &= \lim_{r \rightarrow 0} \sqrt{2\pi r} \sigma_{32} \end{aligned} \quad (8.3)$$

with the limit taken along $\theta = 0$. The stress field near the crack tip is:

$$\begin{aligned} \sigma_{rr} &= \frac{1}{\sqrt{2\pi r}} \left(K_I \cos \frac{\theta}{2} (1 + \sin^2 \frac{\theta}{2}) + K_{II} \sin \frac{\theta}{2} (1 - 3 \sin^2 \frac{\theta}{2}) \right) \\ \sigma_{\theta\theta} &= \frac{1}{\sqrt{2\pi r}} \cos^2 \frac{\theta}{2} \left(K_I \cos \frac{\theta}{2} - 3K_{II} \sin \frac{\theta}{2} \right) \\ \sigma_{r\theta} &= \frac{1}{\sqrt{2\pi r}} \cos \frac{\theta}{2} \left(K_I \sin \frac{\theta}{2} \cos \frac{\theta}{2} + K_{II} (1 - \sin^2 \frac{\theta}{2}) \right) \end{aligned} \quad (8.4)$$

The adaptive mesh refinement steps are shown in Fig. 8.26 for Mode I loading condition. It can be seen that at the initial step NURBS geometry is taken and the solution field is represented by PHT-spline in parametric space with GIFT mapping and at the beginning 2 steps, uniform mesh refinement is performed. Later from step 3, adaptivity mesh refinement starts in particular at the crack tip, it refines more.

In Fig. 8.27 contour plots of displacement and stress of Edge crack Mode I loading condition are shown and due to asymptotic crack-tip higher stress concentration appears at the crack-tip. As the force applied at the top and bottom surface of the solid, displacement magnitude is higher at the edge due to Mode I loading condition.

In Fig. 8.28 H^1 energy error norm has been shown for higher-order PHT-spline and we can see that due to the presence of singularity at the crack-tip, the rate of convergence with uniform refinement is lower compared to adaptive refinement. By using adaptive refinement, the convergence rate becomes clearly higher, although for this example, there is a clear advantage in terms of degrees

of freedom in the favor of higher degree bases. However, the estimated error is not closer to the actual error in energy norm as the polynomial degree increases. The effectivity indices (defined as the ratio of the estimated error and the actual error) are 0.56136 for $p = 3$, 0.71935 for $p = 4$, and 0.66204 for $p = 5$. Here for a higher degree because of the edge crack the adaptive refinement error convergence rate increases in comparison to uniform PHT-refinement.

For the mode II loading condition, the adaptive mesh refinement steps are shown in Fig. 8.29. At the initial step, NURBS geometry is taken and solution field is represented by PHT-spline in parametric space with GIFT mapping and at the beginning 2 steps uniform mesh refinement are performed. Later from the step 3, adaptivity mesh refinement starts in particular at the crack tip where it refines more.

Fig. 8.30 contour plots of displacement and stress of Edge crack Mode II loading condition are shown and due to asymptotic crack tip higher stress concentration appears at the crack tip similar to mode I condition. Due to in-plane shear displacements at the edge of the front side, the displacement is higher.

In Fig. 8.31, the convergence plots for the different polynomial degrees with uniform and adaptive refinements are shown for the energy norm. We observe good agreement between the estimated error obtained using the recovery-based estimator and the exact error computed using the analytical solution. For the adaptive refinement and the last refined mesh, the effectivity indices (defined as the ratio of the estimated error and the actual error) are 0.56026 for $p = 3$, 0.64738 for $p = 4$, and 0.59941 for $p = 5$. Here for higher degree because of the edge crack the adaptive refinement error convergence rate increases in comparison to uniform PHT-refinement.

8.5.2 Penny crack

Here we will study a penny shaped crack domain which is a benchmark problem in 3D fracture mechanics. This allows us to analyze the behavior of the adaptive refinement scheme in the presence of a strong discontinuity. Due to symmetry, the geometry is modeled by considering $1/8$ of the cracked domain, using an annular patch and a quarter of a cylinder patch with C^0 continuity at the interface. The geometry and material properties are shown in Fig. 8.32. The radius of the crack is $a = 1$, the radius of the considered domain is $a + b = 4$ and the height is $c = 4$. Symmetry boundary conditions are imposed on the bottom of the annular patch and the force acting at infinity is set as $\sigma = 10^4$. The exact solutions for the stress and displacement fields can be found in [34].

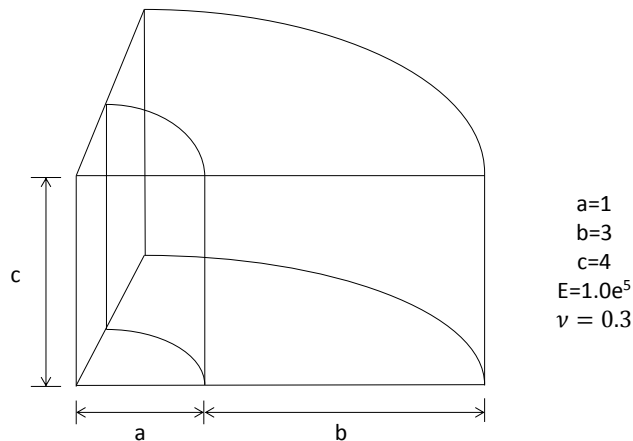


Fig. 8.32: Geometry and material properties of multi-patch penny shaped crack domain.

The von Mises stress field obtained for this example is shown in Figure 8.33 a). A logarithmic scale is used to show the stress singularity at the crack tip. The meshes obtained the last step of refinement for each polynomial degree, corresponding to a relative error of less than 0.0015, are shown in Figures 8.33 b)-d). We note that for all meshes shown, the refinement is concentrated in a narrow area near the crack tip. As before, the higher degree polynomial bases can be used to obtain similar accuracy with significantly fewer elements.

The convergence plots for this example are shown in Figure 8.34. It is noted that due to the presence of singularity at the crack tip, the rate of convergence with uniform refinement is $O(N^{-1/6})$ for all polynomial degrees considered. When adaptive refinement is used, the convergence rate is clearly higher, though for this example there is no longer a clear advantage in terms of degrees of freedom in the favor of higher degree bases. However, the estimated error is closer to the actual error in energy norm as the polynomial degree increases. The effectivity index on the finest meshes is 1.21 for $p = 3$, 0.96 for $p = 4$ and 0.99 for $p = 5$.

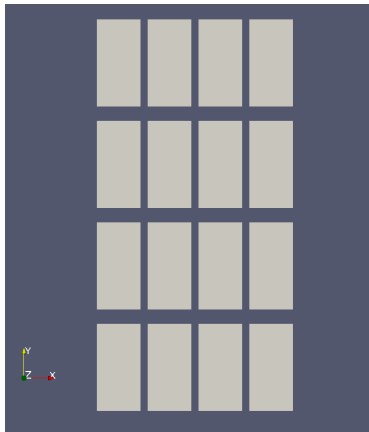
8.5.3 3D Connecting rod

In this example, we study a problem involving more complex geometry, namely that of a connecting rod assembly. The domain of this problem is illustrated in Figure 8.35. We note that a similar problem geometry was also studied in the context of weakly-coupled patches in [97]. A force $F = 1$ kN pointing downward is applied to the surface of the internal ring on the right side of the domain.

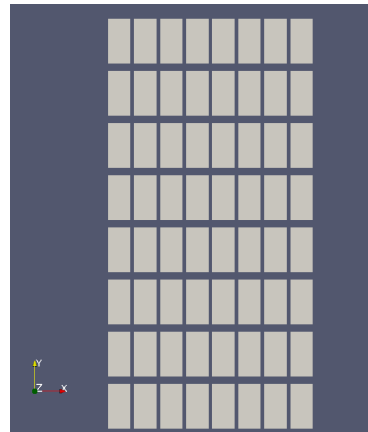
The deformed configuration, von Mises stresses and meshes resulting in the last step of adaptive refinement are shown in Figure 8.36. The refinement is conducted until an estimated relative error in the energy norm of 0.005 is reached. We observe that better accuracy is obtained on coarse meshes as the polynomial degree increases and, as expected, most of the refinements are concentrated in the area where the stem is connected to the left and right rings.

The convergence plots for $p = 3, 4, 5$ are presented in Figure 8.37. We note that for this case only the estimated errors are shown, as an analytical solution is not available. It can be seen as before that adaptive GIFT PHT-spline refinement results in smaller errors compared to uniform refinement. Because of the stress singularities near the junction of the ring elements with the stem, the optimal rate

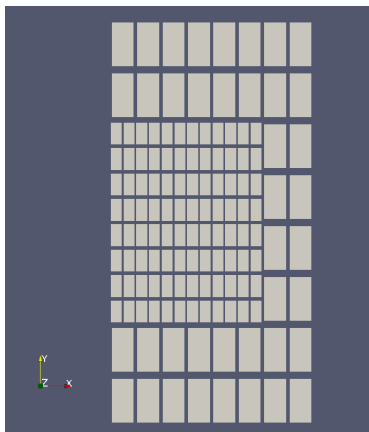
of convergence is not achieved in this example. However, it would be expected that further refinement would result in a more drastic error reduction as the stress singularities are resolved, similarly to the 2D L-shaped example (see in 8.4.2).



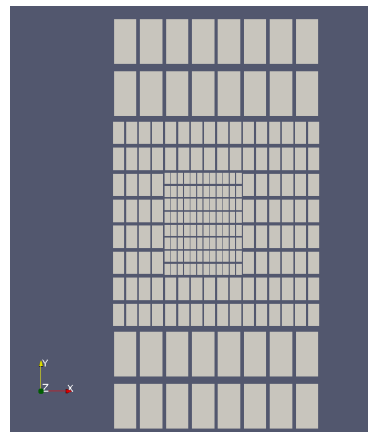
(a) Initial NURBS geometry and PHT-spline in the solution field at level 0.



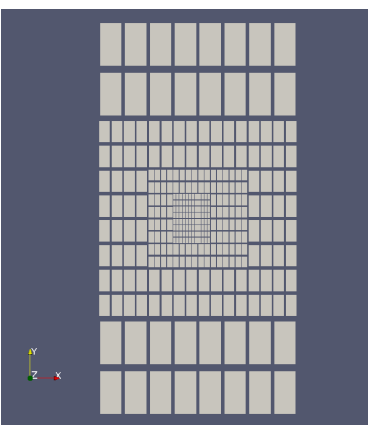
(b) A uniform mesh refinement with PHT-spline solution field at level 1.



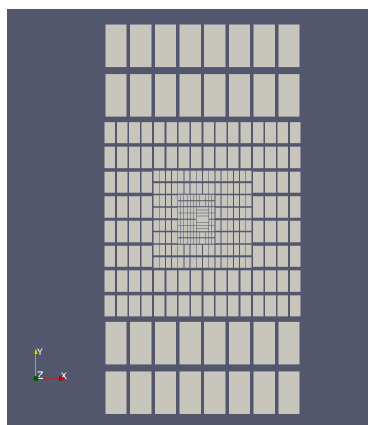
(c) Adaptive local refinement with PHT-spline solution field at level 2.



(d) Adaptive local refinement with PHT-spline solution field at level 3.

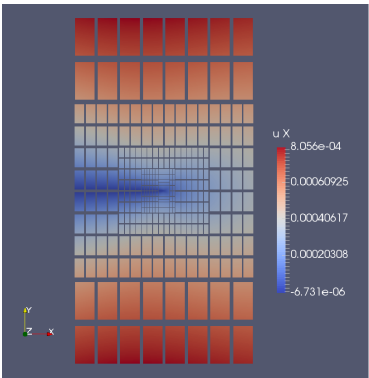


(e) Adaptive local refinement with PHT-spline solution field at level 4.

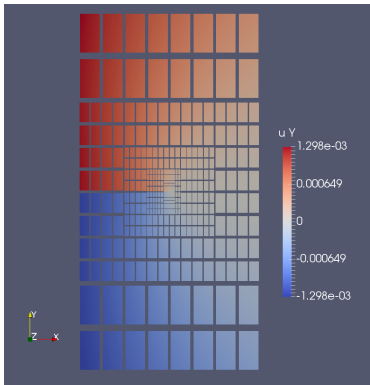


(f) Adaptive local refinement with PHT-spline solution field at level 5.

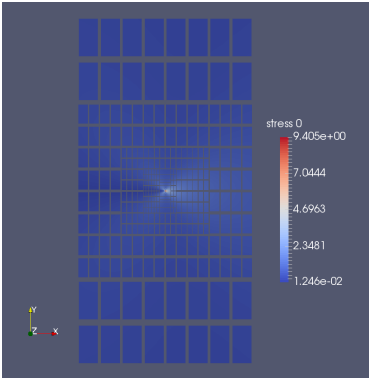
Fig. 8.16: Meshes at different hierarchical level from initial level 0 with the NURBS geometry and the solution field is defined by PHT-spline with adaptive local refinement for Edge crack multi-patch domain.



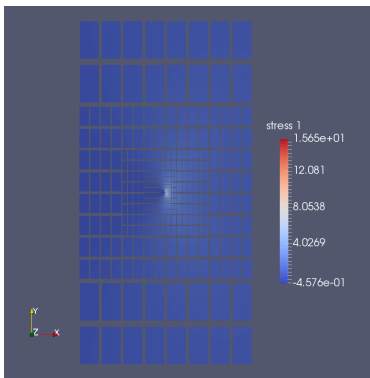
(a) Displacement component in X-direction.



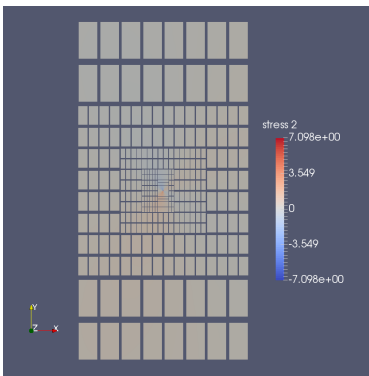
(b) Displacement component in Y-direction.



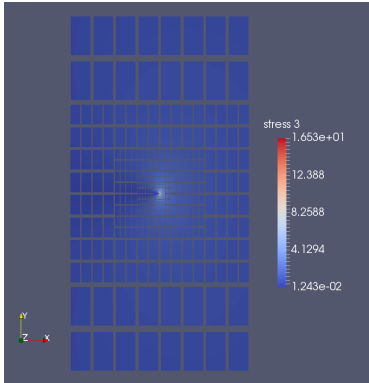
(c) Stress component in X-direction.



(d) Stress component in Y-direction.



(e) Stress component in XY-direction.



(f) Von Mises stress

Fig. 8.17: Contour plots of displacement and stress components of the Edge crack with Multi-Patch.

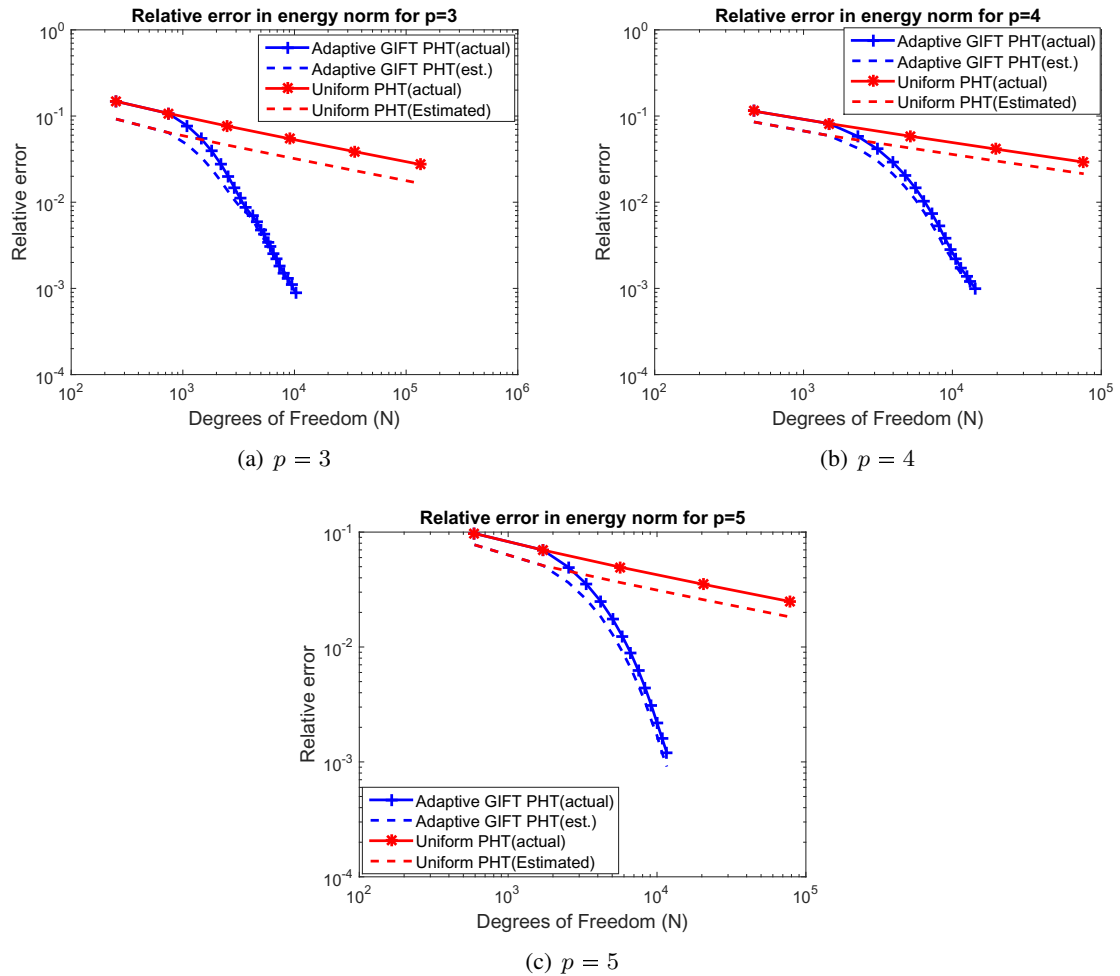


Fig. 8.18: Relative error (exact and estimated) in energy norm vs. the number of degree of freedom for $p = 3, 4$ and 5 , using adaptive GIFT PHT-spline and uniform refinements for the Edge crack multi-patch analysis.

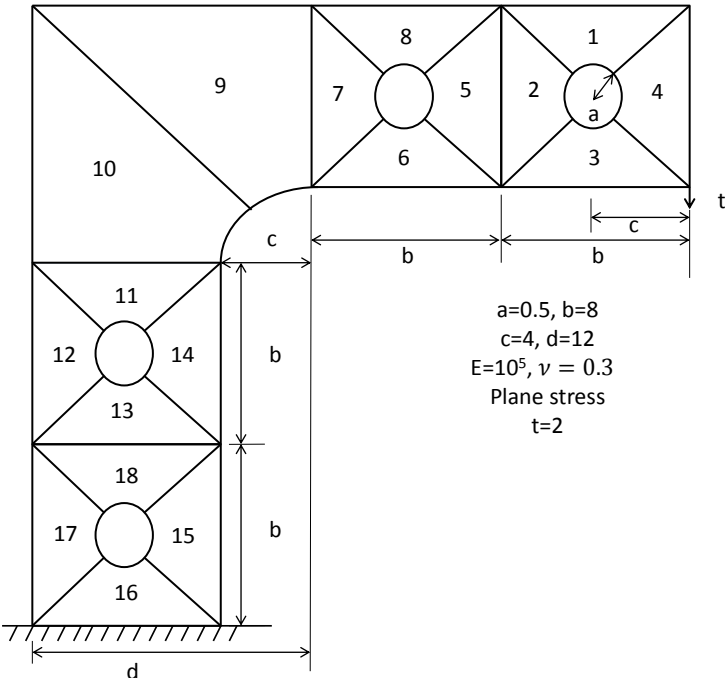


Fig. 8.19: Geometry, boundary conditions and material properties for L-shaped bracket.

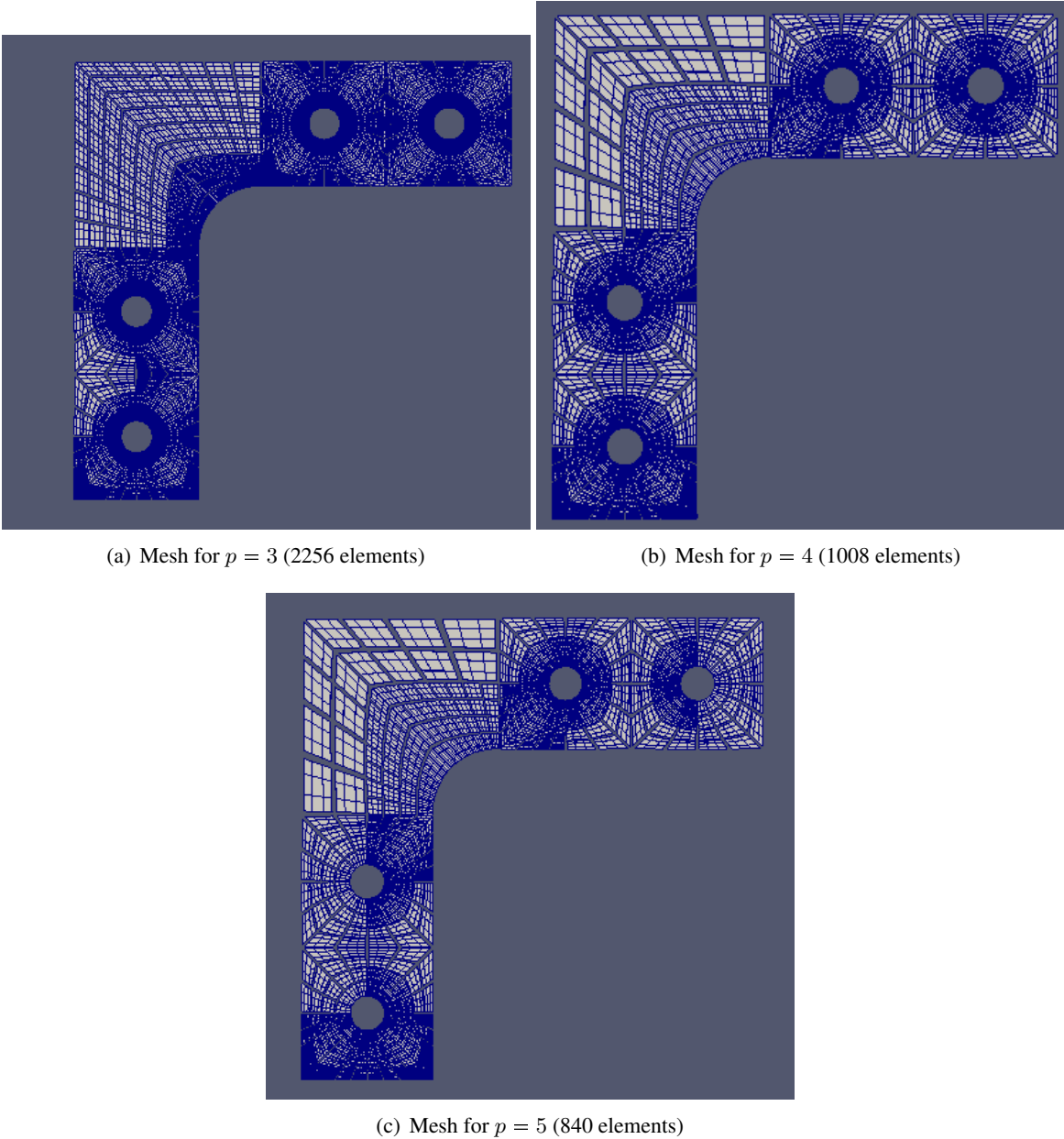
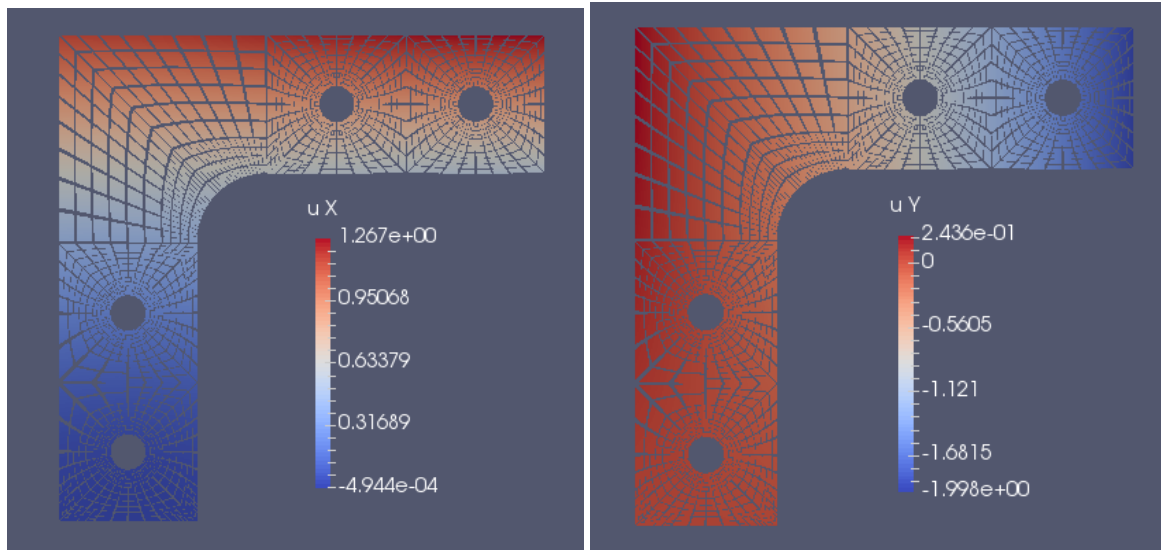
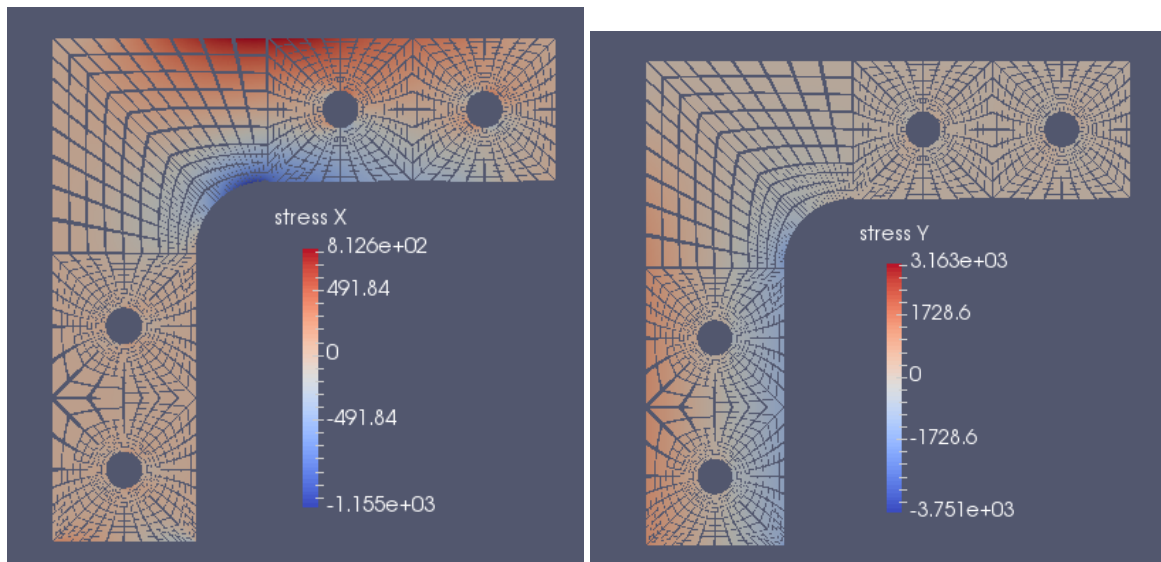


Fig. 8.20: The refined meshes for different polynomial degrees p in the L-Shaped bracket example.



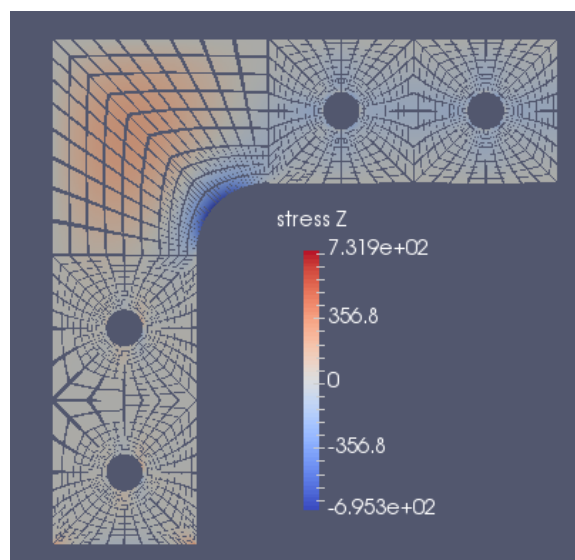
(a) Displacement component in X-direction.

(b) Displacement component in Y-direction.



(c) Stress component in X-direction.

(d) Stress component in Y-direction.



(e) Stress component in XY-direction.

Fig. 8.21: Contour plots of displacement and stress components of the L-shaped bracket with Multi-Patch.

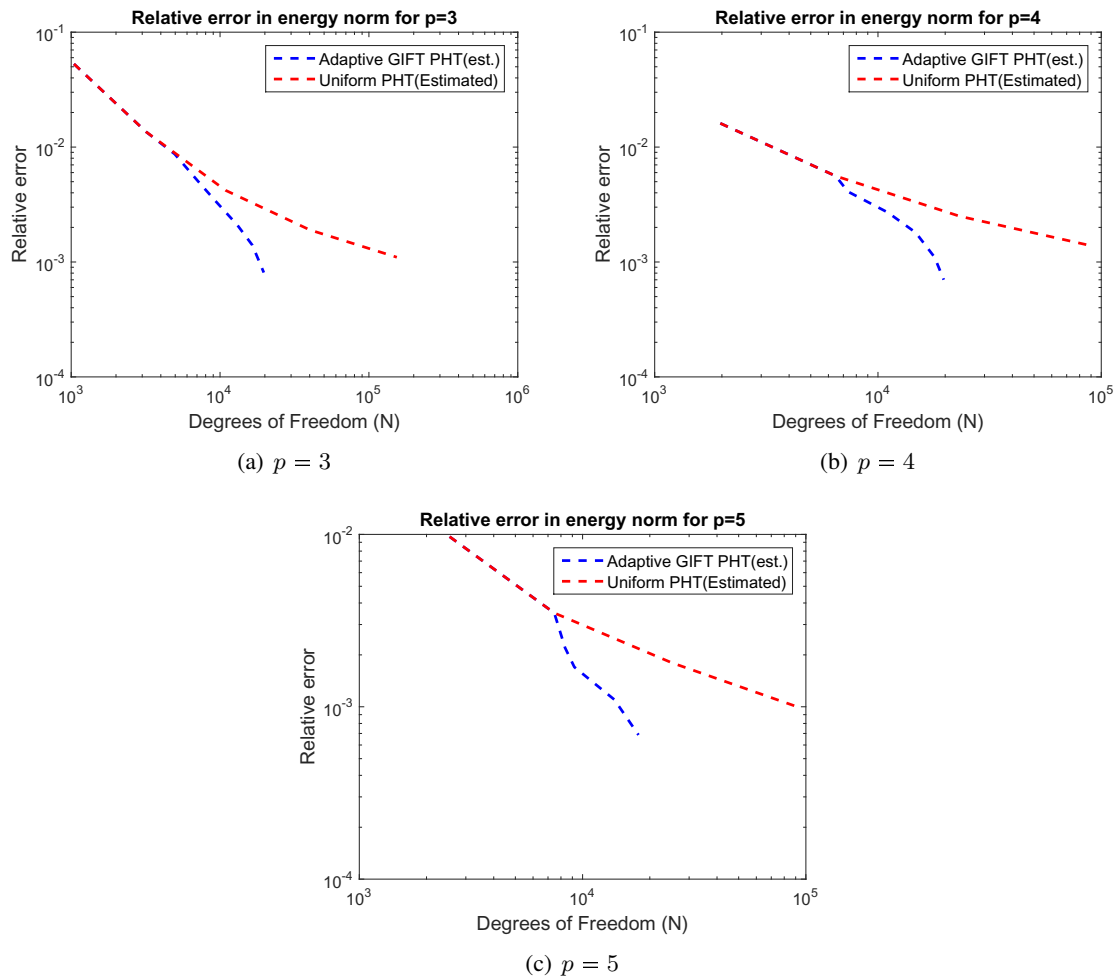


Fig. 8.22: Relative error (exact and estimated) in energy norm vs. the number of degree of freedom for $p = 3, 4$ and 5 , using adaptive GIFT PHT-spline and uniform refinements for the L-shaped bracket multi-patch analysis.

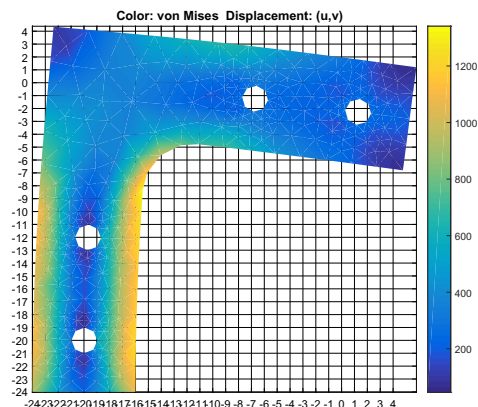


Fig. 8.23: FEM solution of Von mises displacement of L-shaped Bracket using Matlab PDE toolbox.

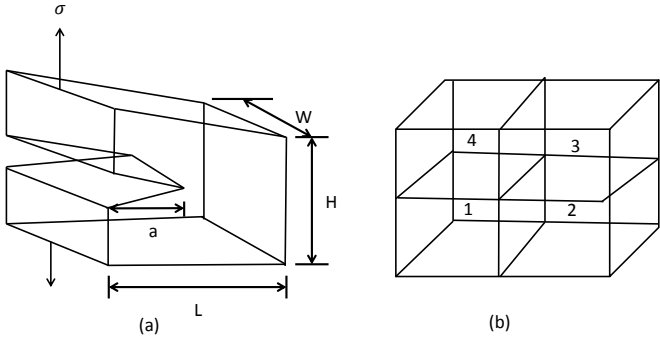


Fig. 8.24: (a)Edge crack with Mode I loading condition. (b) 4 patches to define the edge crack: patches 1-2,2-3, 3-4 connected and 4-1 is not connected.

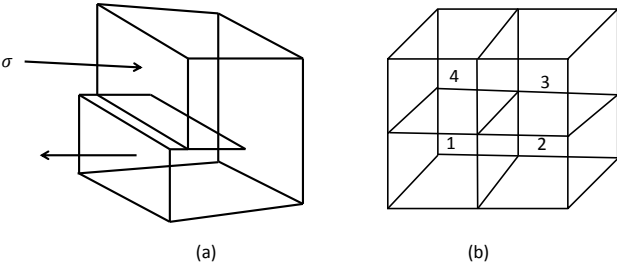
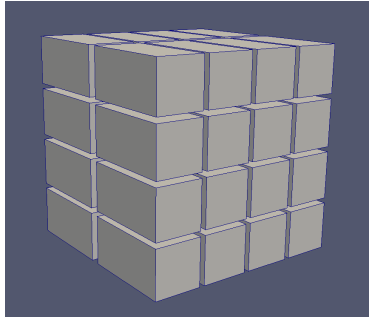
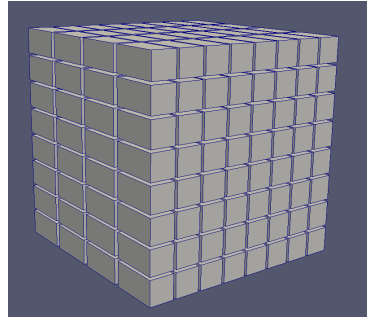


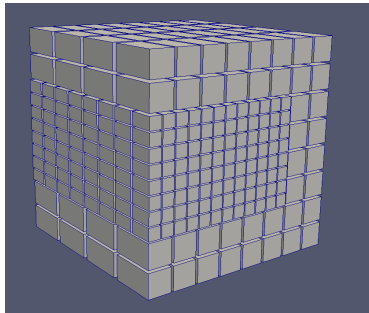
Fig. 8.25: (a) Edge crack with Mode II loading condition. (b) 4 patches to define the edge crack: patches 1-2,2-3, 3-4 connected and 4-1 is not connected.



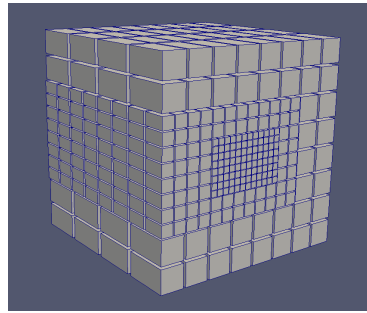
(a) Initial NURBS geometry and PHT-spline in the solution field at level 0.



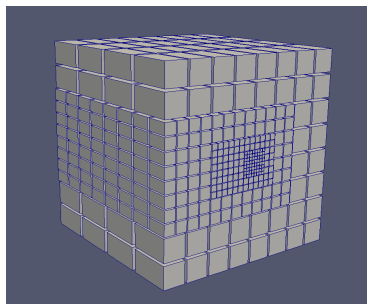
(b) A uniform mesh refinement with PHT-spline solution field at level 1.



(c) A uniform mesh refinement with PHT-spline solution field at level 2.

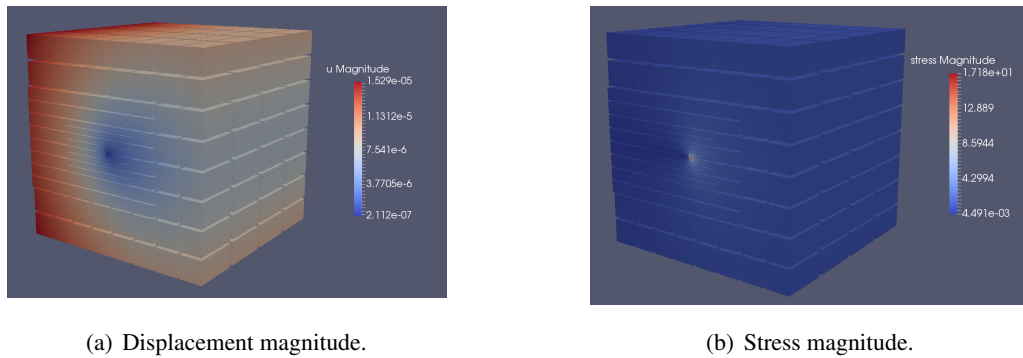


(d) Adaptive local refinement with PHT-spline solution field at level 3.



(e) Adaptive local refinement with PHT-spline solution field at level 4.

Fig. 8.26: Meshes at different hierarchical level from initial level 0 with the NURBS geometry and at the solution field PHT-spline used with adaptive local refinement for Edge crack Mode I loading condition in multi-patch domain.



(a) Displacement magnitude.

(b) Stress magnitude.

Fig. 8.27: Contour plots of displacement and stress magnitude of the Edge Crack Mode I loading condition with Multi-Patch.

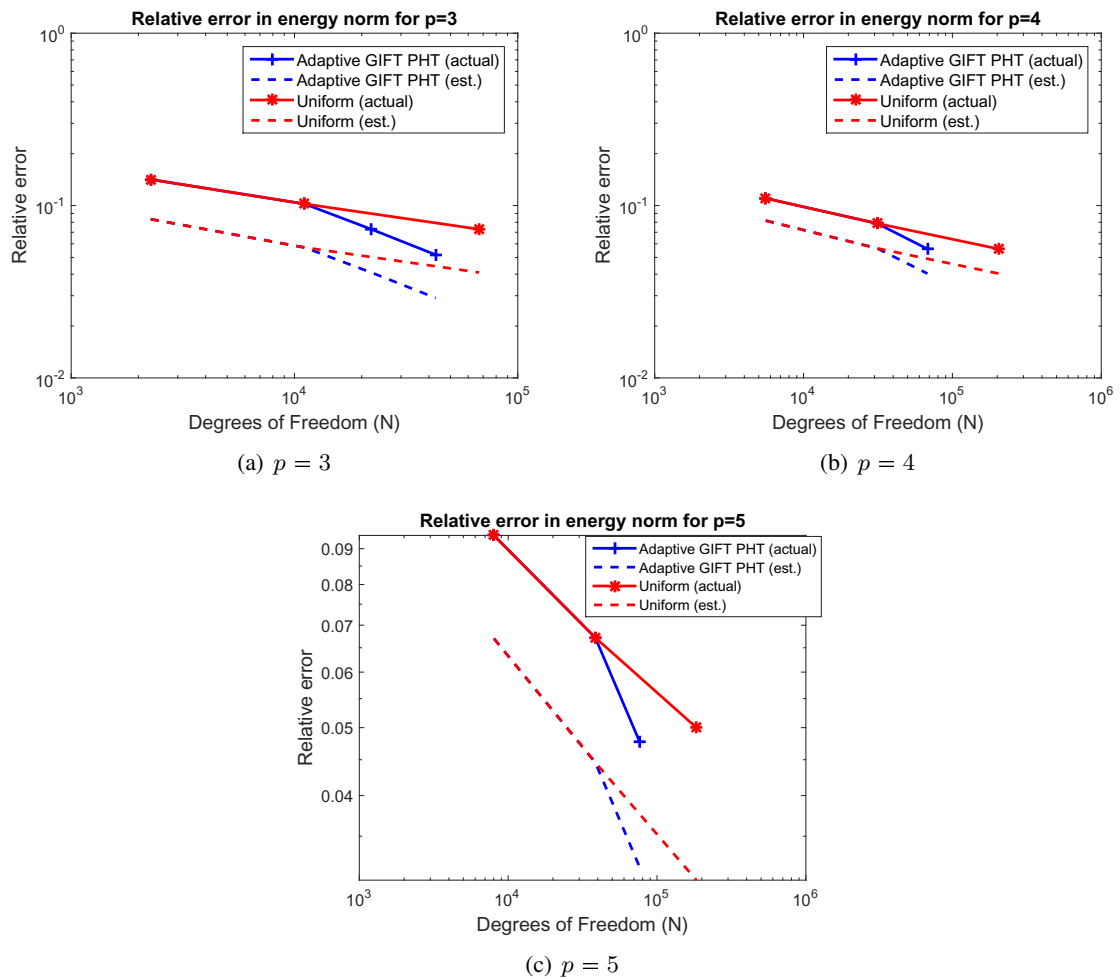
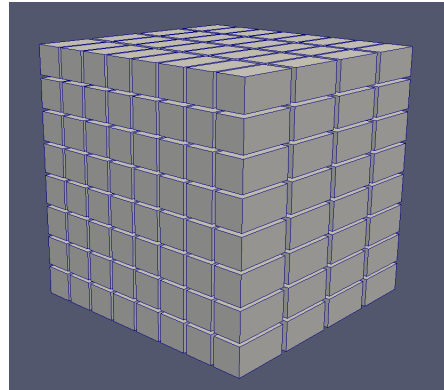
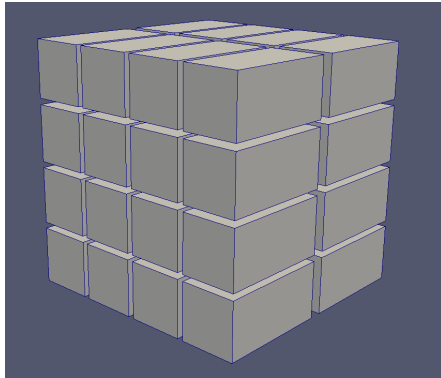
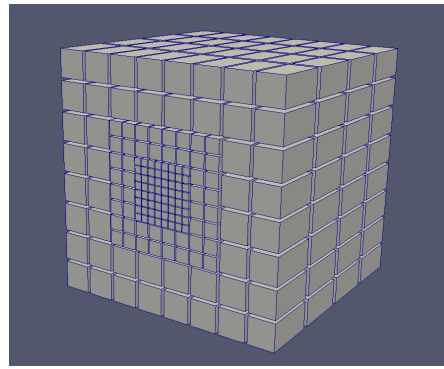
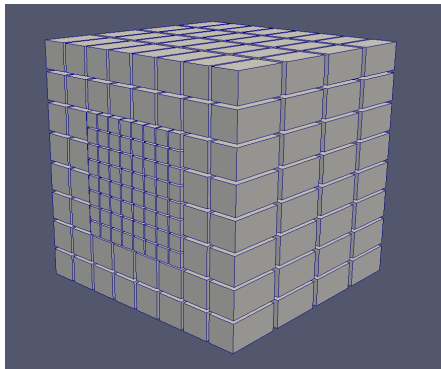


Fig. 8.28: Relative error (exact and estimated) in energy norm vs. the number of degree of freedom for $p = 3, 4$ and 5 , using adaptive and uniform refinements for the 3D Edge crack mode I loading condition example.



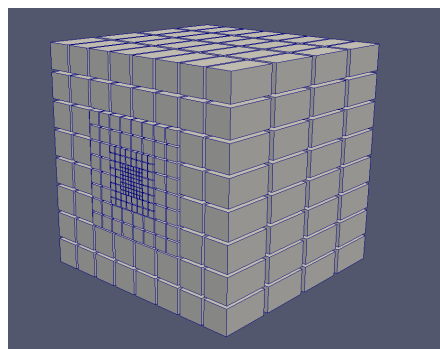
(a) Initial NURBS geometry and PHT-spline in the solution field at level 0.

(b) A uniform mesh refinement with PHT-spline solution field at level 1.



(c) A uniform mesh refinement with PHT-spline solution field at level 2.

(d) Adaptive local refinement with PHT-spline solution field at level 3.



(e) Adaptive local refinement with PHT-spline solution field at level 4.

Fig. 8.29: Meshes at different hierarchical level from initial level 0 with the NURBS geometry and at the solution field PHT-spline used with adaptive local refinement for Edge crack Mode II loading condition in multi-patch domain.

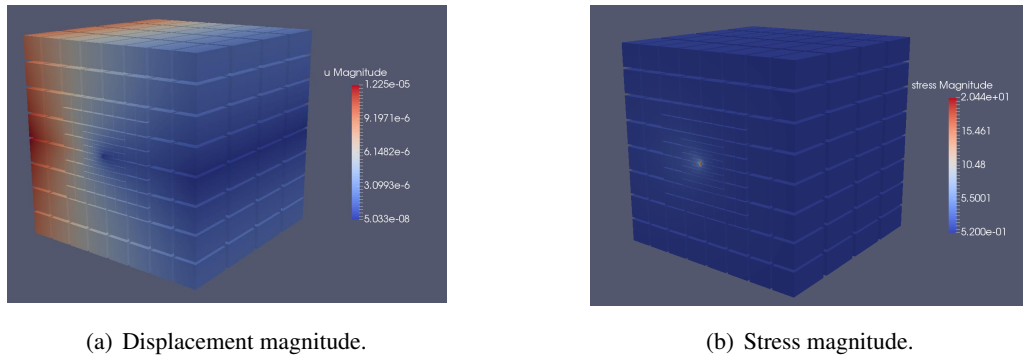


Fig. 8.30: Contour plots of displacement and stress magnitude of the Edge Crack Mode II loading condition with Multi-Patch.

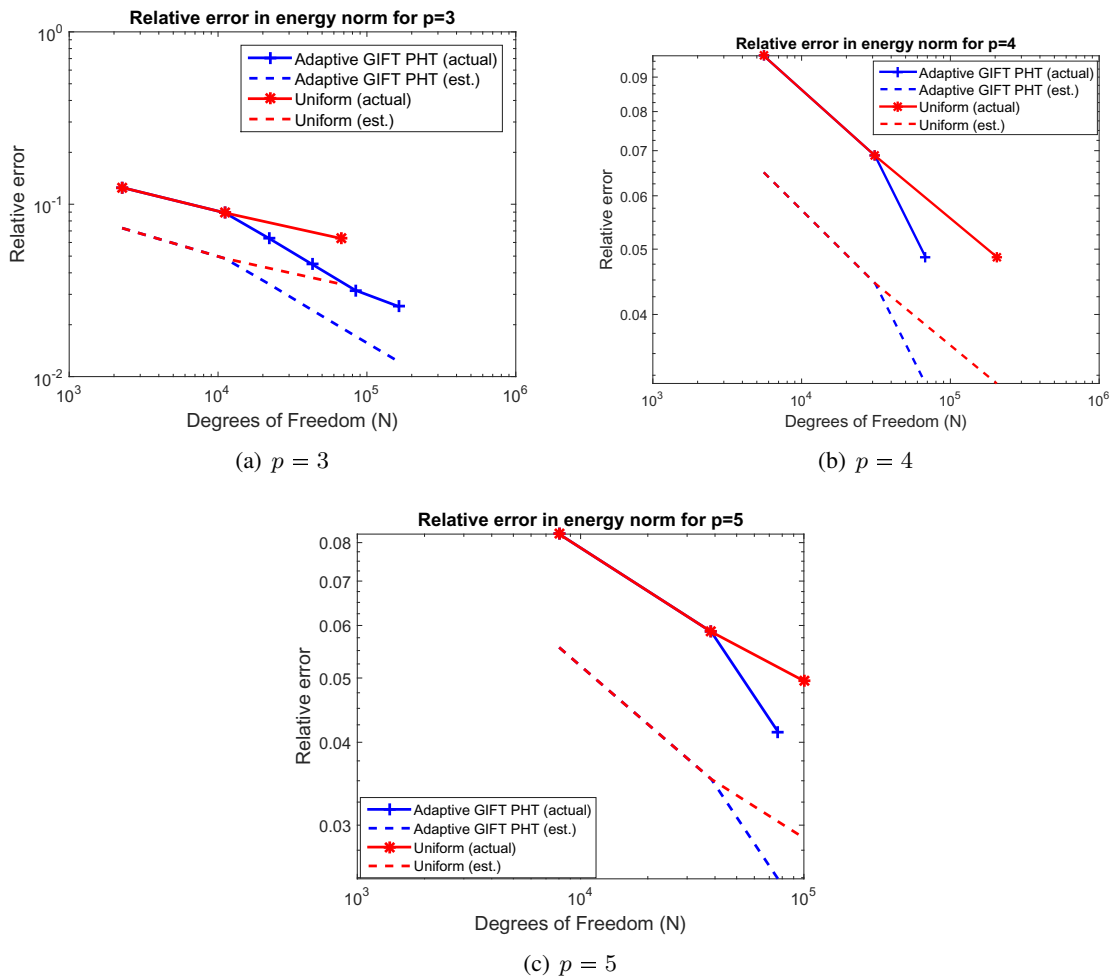


Fig. 8.31: Relative error (exact and estimated) in energy norm vs. the number of degree of freedom for $p = 3, 4$ and 5 , using adaptive and uniform refinements for the 3D Edge crack mode II loading condition example.

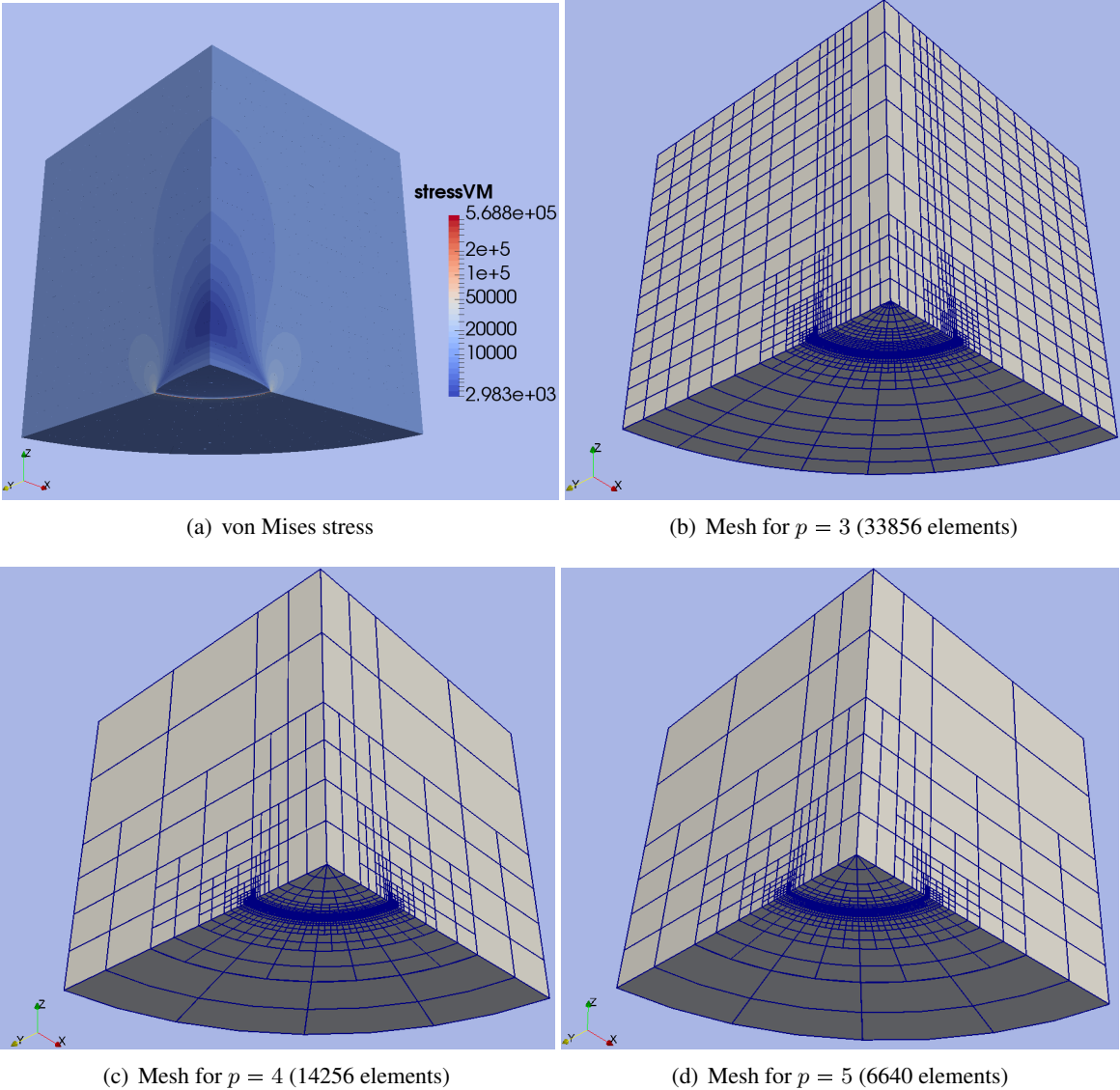


Fig. 8.33: The von Mises stresses and the refined meshes corresponding to the different polynomial degrees p for the penny crack example.

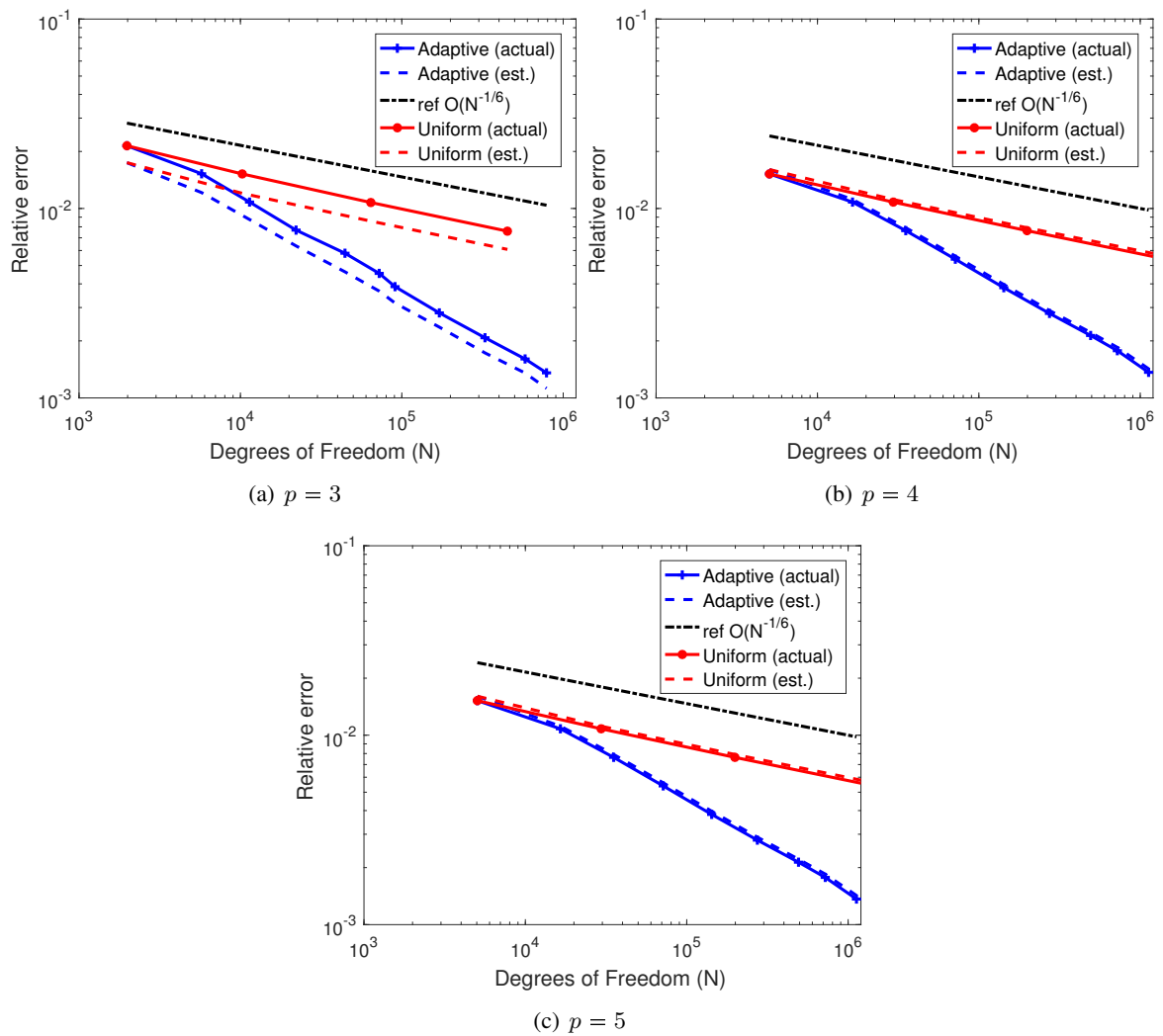


Fig. 8.34: Relative error (exact and estimated) in energy norm vs. the number of degree of freedom for $p = 3, 4$ and 5 , using adaptive and uniform refinements for the penny crack example.

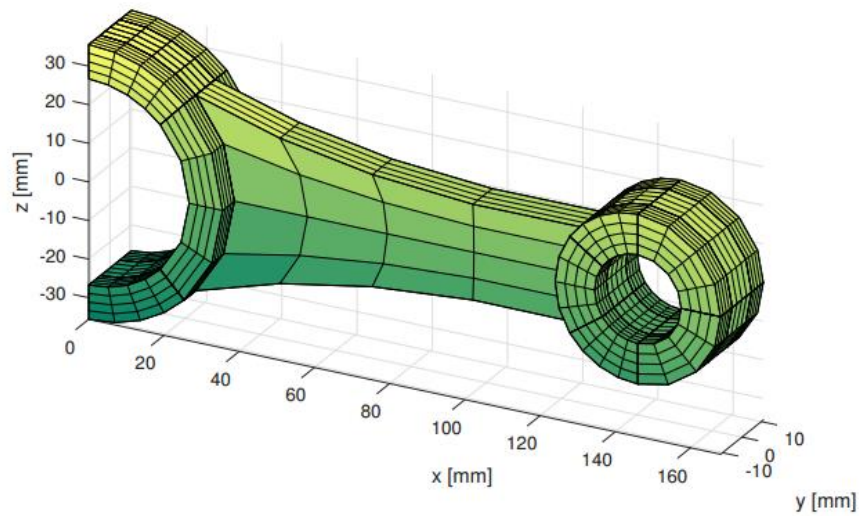
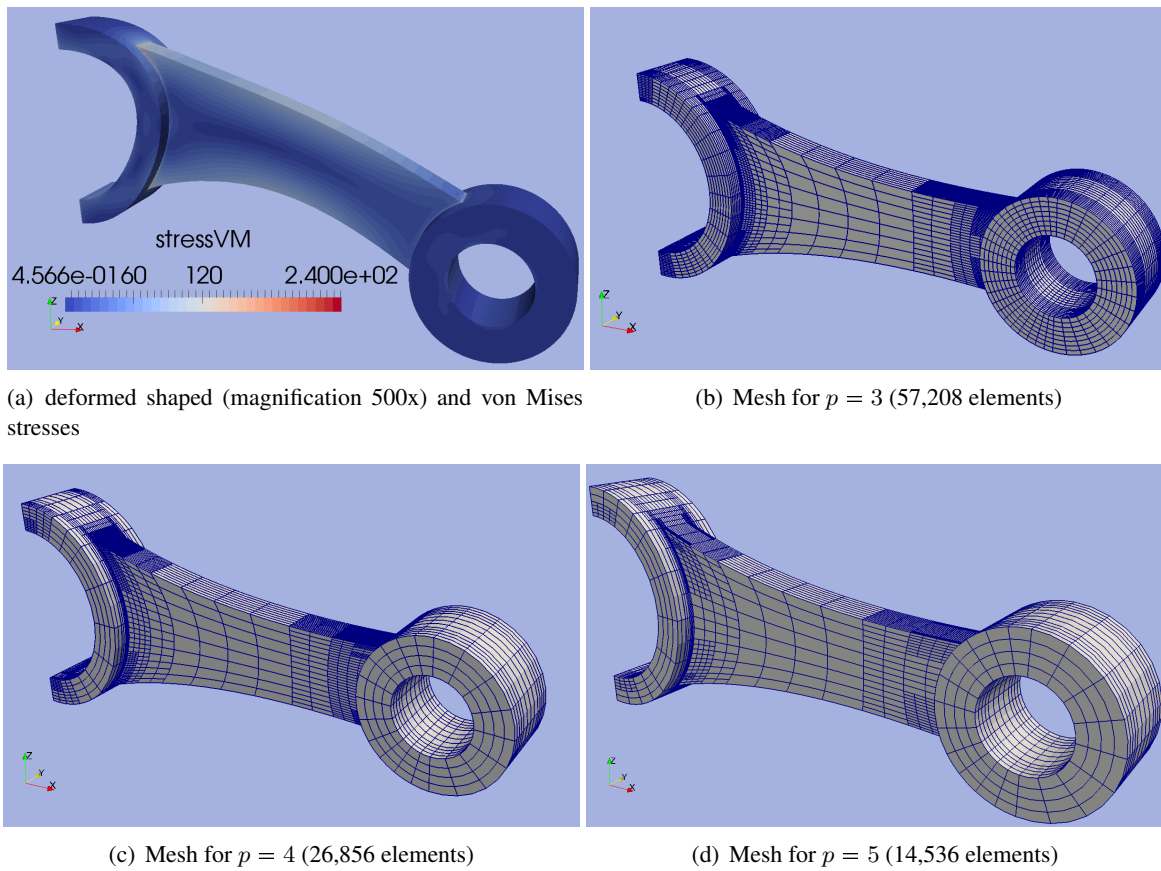


Fig. 8.35: Geometry of the connecting rod example.

Fig. 8.36: The von Mises stresses and the refined meshes corresponding to the different polynomial degrees p for the connecting rod example.

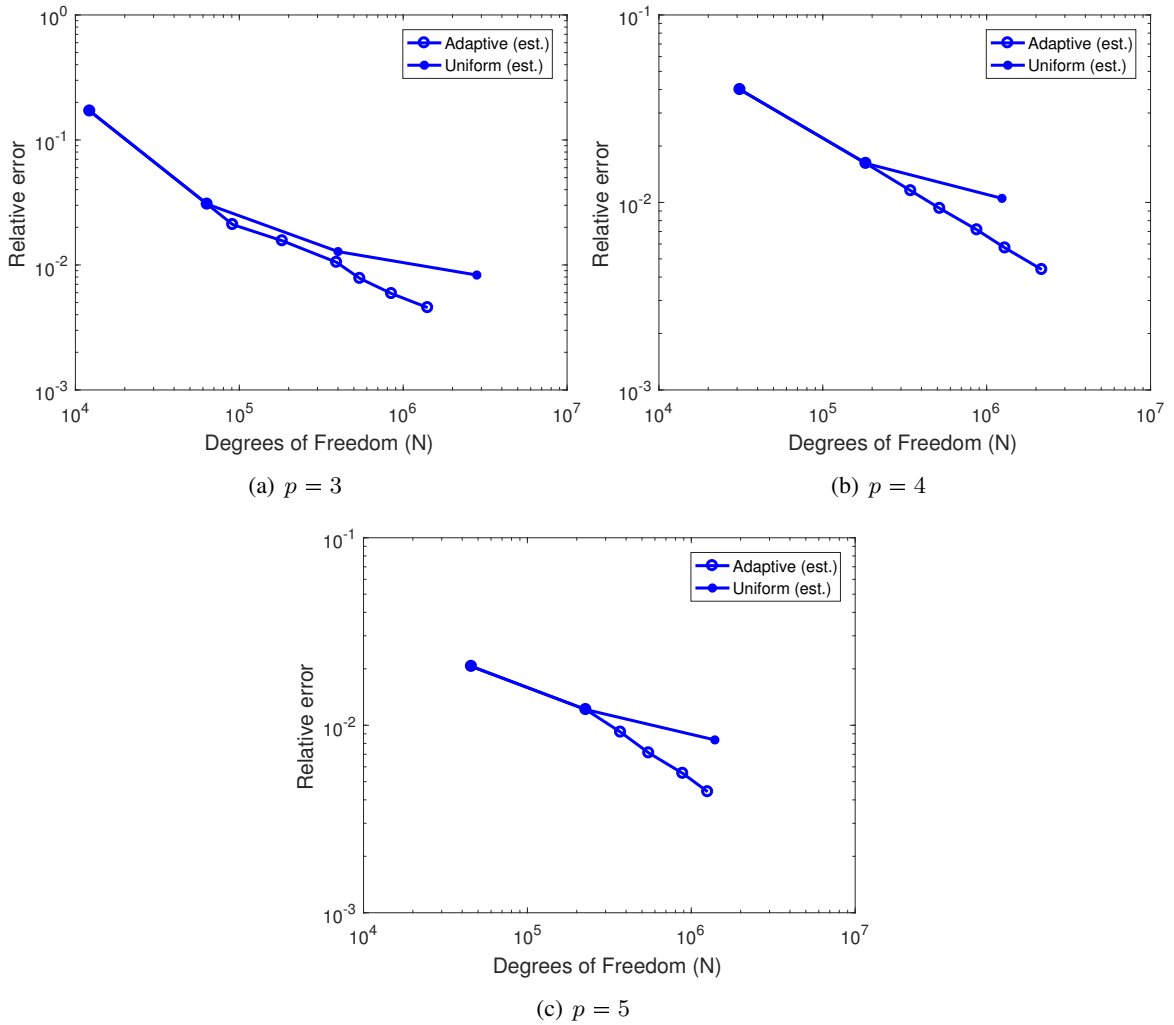


Fig. 8.37: Relative error (exact and estimated) in energy norm vs. the number of degree of freedom for $p = 3, 4$ and 5 , using adaptive GIFT PHT-spline and uniform refinements for the connecting rod example.

Chapter 9

Summary

In this work, the Geometry Independent Field approximaTion (GIFT) method was applied with recovery-based error estimation using NURBS for geometry representation and higher-order PHT-spline for solution approximation in two-dimensional and three-dimensional linear elasticity problems. For IGA and GIFT analysis our main focus was to use IGA-FEM or GIFT-FEM based method to utilize the main advantage of CAD geometry with NURBS and higher-order PHT-splines for the solution field using the flexible GIFT mapping. The IGA adopts the same spline basis functions as used in the CAD to discretize the elliptic boundary value problem in case of linear elasticity. Compared with the IGA in the Finite Element form, the main advantage using GIFT is to represent the solution field independent of the geometry to alleviate the constraint of using the same spline representation which for some numerical analysis may not be suitable (where local refinement is needed), and thus compatible with the CAD geometric models. We present here in this work a flexible way to do the numerical analysis so that we can do adaptive local refinement using recovery based error estimation. We have derived an error estimator based on superconvergent patch recovery and an associated refinement procedure which takes advantage of the hierarchical mesh structure. Hence, IGA and GIFT can achieve a true integration of analysis and CAD, which plays an important role in automating and accelerating the numerical simulation process. To facilitate the implementation of IGA and GIFT in Diffpack for different splines, the present work made the following contributions:

- The IGA-FEM formulation are presented by discretizing the elliptic boundary value problem in case of linear elasticity where NURBS based IGA is implemented in an object-oriented C++ environment called Diffpack. The main focus of this part is to integrate the CAD software Rhino3D with Diffpack so that geometry designed by Rhino3D (NURBS) can be analyzed in the IGA Diffpack toolbox. It has also been shown that the IGA NURBS is more efficient, accurate and robust compared to traditional FEM by some numerical examples in Chapter 3. We also present in Chapter 3 how to use a Bézier mesh created from Rhino3D in Diffpack for analysis where the refinement property can be utilized manually defined by the design process in CAD software.
- As most of the CAD designs are based on NURBS and it has been used in IGA extensively so far in many applications, but still using IGA has some constraints in some cases of numerical problems. The same basis spline space for the geometry and solution field may not be suitable specially in case of the problems with discontinuity and singularity where local refinement is necessary. Due to uniform refinement property of NURBS, we used here a flexible method GIFT because of the representation of solution field spline space independent of the geometric spline space which is usually represented by the NURBS. The solution space with higher-order and greater continuity ($p = 3, C^1$ to $p = 5, C^2$) is defined by PHT-spline suitable for local refinement with the most robust recovery-based error estimation. In Chapter 4 we present the

GIFT method elaborately along with the adaptivity procedure and recovery based error estimation using superconvergent patch recovery technique in detail to make the process automatic. We also presented a simple "Patch-test" in Chapter 4 which passes even for the higher-order PHT-spline. But "Patch-test" can not be guaranteed as it is explained in details by [13] and it depends on the various choices of geometry and solution field spline approximation. The marking algorithm with "Dörfler marking" strategy has been shown to select which elements need to mark for refinement and mathematical formulation of determining the location of superconvergent points have also presented in 4.

- For local refinement, we represent the solution spline space by the PHT-spline and we start our presentation in at first with basic 1D PHT-spline approach and show the PHT-spline basis function properties and refinement algorithms with a simple 1D example by De Casteljau algorithm in Chapter 5. With this simple 1D example we also have shown that the higher-order PHT-spline can be a useful property in PHT-spline for IGA and GIFT analysis.
- To make the analysis deeper we then present the 2D PHT-splines properties with the dimensional formula, 2D T-mesh structure and also the 2D PHT-spline basis functions refinement procedures in details in Chapter 6. We also present a simple 2D numerical example showing the advantage of using higher-order PHT-splines and comparing the results with the adaptive and uniform refinement of GIFT method. After demonstrating the numerical examples it can be said that 2D adaptive GIFT PHT-spline analysis in comparison to uniform refinement shows robust performance. Although comparing the cubic PHT-spline in adaptive GIFT, we evaluate the performance of the higher-order PHT-spline and it has been explained that with higher-order PHT-spline, greater convergence rate can be achieved with a minimum degree of freedoms (Dofs).
- We extended the 2D PHT-spline formulation to 3D PHT-spline to present the adaptive GIFT performance in the 3D numerical case. In Chapter 7, we presented the 3D PHT-spline dimensional formula, 3D hierarchical T-mesh and showed how to calculate the 3D PHT-spline basis for higher degree and continuity using the dimensional formula. Furthermore, some 3D GIFT PHT-spline numerical examples presented in Chapter 7. By studying the numerical examples it can be concluded that for the 3D GIFT PHT-spline in comparison of uniform refinement in energy error norms for the problems with smooth solutions do not show very effective convergence rate for higher-order and in some steps it diverges but after certain refinement steps the convergence rate gets similar comparing to uniform refinement.
- To get the main advantage of adaptive GIFT PHT-spline we presented in Chapter 8, some numerical examples with discontinuity and singularity where adaptivity is more effective. We presented in Chapter 8 a simple multi-patch analysis of adaptive GIFT PHT-spline where it is possible to handle the geometries with C^0 continuity in between the patches and to represent the discontinuity for e.g re-entrant corner or cracks at the geometry. After investigating from the examples which have a discontinuity in 2D and 3D GIFT PHT-spline, it can be concluded that significant improvement and accuracy per degree of freedom is obtained by using adaptive local refinement, capturing stress singularities at the re-entrant corners and significant reduction of computational cost can be achieved.

9.1 Future work

The work can be extended in the future as follows:

- This work can be extended by coupling of spline approximations for the geometry with non-spline

approximation for the field variables, including meshless methods. It would be particularly attractive to work with Maximum Entropy Shape functions (see [11, 9]).

- GIFT within isogeometric collocation and isogeometric boundary element framework.
- In this thesis, we assumed the field approximation spline space to be arbitrary, independent of the geometry approximation. It would be desirable for sensitivity analysis, e.g. through adjoint methods to drive the automatic selection of the most suitable field spline space based on goal-oriented error estimators (see. [63]).
- GIFT for isogeometric shape and topology optimization [127], where the constraint of using the same space for the geometry and the approximation is particularly undesired as described by [87].
- A possible topic for future work is the development of more efficient $hp\alpha$ -refinement schemes which would require the integration of splines of different degrees and different continuity of orders on the same hierarchical mesh and the development of h - and p - refinement strategies. The latter would require a more in-depth study of various error estimation methods.

Bibliography

- [1] Differential package. <http://www.diffpack.com/>.
- [2] Innovative Numerical Technologies. <https://inutech.de/>.
- [3] Simula research laboratory. <https://www.simula.no/>.
- [4] SINTEF. <https://www.sintef.no/en/>.
- [5] Studies of refinement and continuity in geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 196:4160–4183, 2007.
- [6] Rotation free isogeometric thin shell analysis using PHT-splines. *Computer Methods in Applied Mechanics and Engineering*, 200(4748):3410 – 3424, 2011.
- [7] Mark Ainsworth and J.Tinsley Oden. A posteriori error estimation in finite element analysis. *Computer Methods in Applied Mechanics and Engineering*, 142(12):1 – 88, 1997.
- [8] I. Akkerman, Y. Bazilevs, V.M. Calo, T.J.R. Hughes, and S. Hulshoff. The role of continuity in residual-based variational multiscale modeling of turbulence. *Computational Mechanics*, 41(3):371–378, 2008.
- [9] F. Amiri, C. Anitescu, M. Arroyo, S.P.A. Bordas, and T. Rabczuk. XLME interpolants, a seamless bridge between XFEM and enriched meshless methods. *Computational Mechanics*, 53(1):45–57, 2014.
- [10] Cosmin Anitescu, Yue Jia, Yongjie Jessica Zhang, and Timon Rabczuk. An isogeometric collocation method using superconvergent points. *Computer Methods in Applied Mechanics and Engineering*, 284:1073 – 1097, 2015. Isogeometric Analysis Special Issue.
- [11] M. Arroyo and M. Ortiz. Local maximum-entropy approximation schemes: a seamless bridge between finite elements and meshfree methods. *International Journal for Numerical Methods in Engineering*, 65(13):2167–2202, 2006.
- [12] E. Atroshchenko, X. Peng, J. S. Hale, S. K. Tomar, G. Xu, and S. P. A. Bordas. Boundary Element Method with NURBS-geometry and independent field approximations in plane elasticity. In *1st Pan-American Congress on Computational Mechanics*, 2015.
- [13] Elena Atroshchenko, Satyendra Tomar, Gang Xu, and S.P.A. Bordas. Weakening the tight coupling between geometry and simulation in isogeometric analysis: from sub- and super-geometric analysis to Geometry Independent Field approximation (GIFT). *International Journal for Numerical Methods in Engineering*, 114:1131 – 1159, 2018.
- [14] F. AURICCHIO, L. BEIRO DA VEIGA, T. J. R. HUGHES, A. REALI, and G. SANGALLI. ISOGEOOMETRIC COLLOCATION METHODS. *Mathematical Models and Methods in Applied Sciences*, 20(11):2075–2107, 2010.

- [15] I. Babuška and A. Miller. A feedback finite element method with a posteriori error estimation: Part I. The finite element method and some basic properties of the a posteriori error estimator. *Computer Methods in Applied Mechanics and Engineering*, 61(1):1 – 40, 1987.
- [16] I. Babuška and W. C. Rheinboldt. A-posteriori error estimates for the finite element method. *International Journal for Numerical Methods in Engineering*, 12(10):1597–1615, 1978.
- [17] I. BABUŠKA, T. STROUBOULIS, and C. S. UPADHYAY. A MODEL STUDY OF THE QUALITY OF A POSTERIORI ERROR ESTIMATORS FOR FINITE ELEMENT SOLUTIONS OF LINEAR ELLIPTIC PROBLEMS, WITH PARTICULAR REFERENCE TO THE BEHAVIOR NEAR THE BOUNDARY. *International Journal for Numerical Methods in Engineering*, 40(14):2521–2577, 1997.
- [18] I. Babuška, T. Strouboulis, C. S. Upadhyay, S. K. Gangaraj, and K. Copps. Validation of a posteriori error estimators by numerical approach. *International Journal for Numerical Methods in Engineering*, 37(7):1073–1123, 1994.
- [19] I Babuška, T Strouboulis, and C.S Upadhyay. A model study of the quality of a posteriori error estimators for linear elliptic problems. Error estimation in the interior of patchwise uniform grids of triangles. *Computer Methods in Applied Mechanics and Engineering*, 114(3):307 – 378, 1994.
- [20] I. Babuška and W. C. Rheinboldt. Error Estimates for Adaptive Finite Element Computations. *SIAM Journal on Numerical Analysis*, 15(4):736–754, 1978.
- [21] W. Bangerth, C. Burstedde, T. Heister, and M. Kronbichler. Algorithms and data structures for massively parallel generic adaptive finite element codes. *ACM Trans. Math. Softw*, 38(2):14, 2011.
- [22] J. R. Barber. *Elasticity*. Springer-Verlag, New York, 2010.
- [23] Y. Bazilevs, V. M. Calo, T. J. R. Hughes, and Y. Zhang. Isogeometric fluid–structure interaction: theory, algorithms, and computations. *Computational Mechanics*, 43:3–37, 2008.
- [24] Y. Bazilevs, V.M. Calo, J.A. Cottrell, J.A. Evans, T.J.R. Hughes, S. Lipton, M.A. Scott, and T.W. Sederberg. Isogeometric analysis using T–splines. *Computer Methods in Applied Mechanics and Engineering*, 199(58):229 – 263, 2010. Computational Geometry and Analysis.
- [25] Y. Bazilevs, L. Beirao da Veiga, J.A. Cottrell, T.J.R. Hughes, and G. Sangalli. Isogeometric Analysis: Approximation, stability and error estimates for h-refined meshes. *Mathematical Models and Methods in Applied Sciences*, 16:1031–1090, 2006.
- [26] Y. Bazilevs, J.R. Gohean, T.J.R. Hughes, R.D. Moser, and Y. Zhang. Patient-specific isogeometric fluid-structure interaction analysis of thoracic aortic blood flow due to implantation of the Jarvik 2000 left ventricular assist device. *Computer Methods in Applied Mechanics and Engineering*, 198(45-46):3534–3550, 2008.
- [27] G. Beer. Iso-Geometric Methods for Numerical Simulation, May 2013.
- [28] G. Beer, B. Marussig, J. Zechner, C. Duenser, and T. P. Fries. Boundary Element Analysis with trimmed NURBS and a generalized IGA approach. In *11th World Congress on Computational Mechanics (WCCM XI)*, 2014.
- [29] D.J. Benson, Y. Bazilevs, M.C. Hsu, and T.J.R. Hughes. Isogeometric shell analysis: The Reissner–Mindlin shell. *Computer Methods in Applied Mechanics and Engineering*, 199(5-8):276 – 289, 2010.

- [30] D.J. Benson, S. Hartmann, Y. Bazilevs, M.-C. Hsu, and T.J.R. Hughes. Blended isogeometric shells. *Computer Methods in Applied Mechanics and Engineering*, 255:133 – 146, 2013.
- [31] C. De Boor. On calculating with B-splines. *Journal of Approximation Theory*, 6(1):50–62, 1972.
- [32] M. J. Borden, M. A. Scott, J. A. Evans, and T. J. R. Hughes. Isogeometric finite element data structures based on Bézier extraction of NURBS. *International Journal for Numerical Methods in Engineering*, 87(1-5):15–47, 2011.
- [33] B. Bornemann and F. Cirak. A subdivision-based implementation of the hierarchical B-spline finite element method. *Computer Methods in Applied Mechanics and Engineering*, 253:584–598, 2013.
- [34] A. Bower. *Applied Mechanics of Solids*. CRC Press, 2009.
- [35] Cesare Bracco, Carlotta Giannelli, and Alessandra Sestini. Adaptive scattered data fitting by extension of local approximations to hierarchical splines. *Computer Aided Geometric Design*, 52-53(Supplement C):90 – 105, 2017. Geometric Modeling and Processing 2017.
- [36] A. Buffa, D. Cho, and M. Kumar. Characterization of T-splines with reduced continuity order on T-meshes. *Computer Methods in Applied Mechanics and Engineering*, 201204:112 – 126, 2012.
- [37] A. Buffa, D. Cho, and G. Sangalli. Linear independence of the T-spline blending functions associated with some particular T-meshes. *Computer Methods in Applied Mechanics and Engineering*, 199(2324):1437 – 1445, 2010.
- [38] A. Buffa, G. Sangalli, and R. Vázquez. Isogeometric analysis in electromagnetics: B-splines approximation. *Computer Methods in Applied Mechanics and Engineering*, 199(1720):1143 – 1152, 2010.
- [39] C. Burstedde, L. Wilcox, and O. Ghattas. p4est: Scalable Algorithms for Parallel Adaptive Mesh Refinement on Forests of Octrees. *SIAM J. Scient. Compu*, 33(3):1103–1133, 2011.
- [40] Chiu Ling Chan, Cosmin Anitescu, and Timon Rabczuk. Volumetric parametrization from a level set boundary representation with PHT-splines. *Computer-Aided Design*, 82:29 – 41, 2017. Isogeometric Design and Analysis.
- [41] Fehmi Cirak, Michael Ortiz, and Peter Schröder. Subdivision surfaces: a new paradigm for thin-shell finite-element analysis. *International Journal for Numerical Methods in Engineering*, 47(12):2039–2072, 2000.
- [42] Nathan Collier, Lisandro Dalcin, David Pardo, and V. M. Calo. The Cost of Continuity: Performance of Iterative Solvers on Isogeometric Finite Elements. *SIAM Journal on Scientific Computing*, 35(2):A767–A784, 2013.
- [43] Nathan Collier, David Pardo, Lisandro Dalcin, Maciej Paszynski, and V.M. Calo. The cost of continuity: A study of the performance of isogeometric finite elements using direct solvers. *Computer Methods in Applied Mechanics and Engineering*, 213-216(Supplement C):353 – 361, 2012.
- [44] J. Austin Cottrell, Thomas J. R. Hughes, and Yuri Bazilevs. *Isogeometric Analysis: Toward Integration of CAD and FEA*. Wiley Publishing, 1st edition, 2009.

- [45] J.A. Cottrell, A. Reali, Y. Bazilevs, and T.J.R. Hughes. Isogeometric analysis of structural vibrations. *Computer Methods in Applied Mechanics and Engineering*, 195(41-43):5257–5296, 2006.
- [46] M. G. Cox. The numerical evaluation of B-splines. *IMA Journal of Applied Mathematics*, 10(2):134–149, 1972.
- [47] Luca Dedé, Micheal J. Borden, and Thomas J. R. Hughes. Isogeometric Analysis for Topology Optimization with a Phase Field Model. *Archives of Computational Methods in Engineering*, 19(3):427–465, Sep 2012.
- [48] Jiansong Deng, Falai Chen, and Yuyu Feng. Dimensions of spline spaces over T-meshes. *Journal of Computational and Applied Mathematics*, 194(3):267–283, 2006.
- [49] Jiansong Deng, Falai Chen, Xin Li, Changqi Hu, Weihua Tong, Zhouwang Yang, and Yuyu Feng. Polynomial splines over hierarchical T-meshes. *Graphical Models*, 70:76–86, 2008.
- [50] Tor Dokken, Tom Lyche, and Kjell Fredrik Pettersen. Polynomial splines over locally refined box-partitions. *Computer Aided Geometric Design*, 30(3):331 – 356, 2013.
- [51] Tor Dokken, Ewald Quak, and Vibeke Skytt. Requirements from isogeometric analysis for changes in product design ontologies. In *FOCUS K3D CONFERENCE ON SEMANTIC 3D MEDIA AND CONTENT, February 11-12 2010, INRIA Sophia Antipolis - Mediterranee, France*.
- [52] M. Dörfel, B. Jüttler, and B. Simeon. Adaptive isogeometric analysis by local h-refinement with T-splines. *Computers and Mathematics with Applications*, 199(5-8):264–275, 2010.
- [53] Willy Dörfler. A Convergent Adaptive Algorithm for Poissons Equation. *SIAM Journal on Numerical Analysis*, 33(3):1106–1124, 1996.
- [54] John A. Evans, Yuri Bazilevs, Ivo Babuška, and Thomas J.R. Hughes. n-Widths, sup-infs, and optimality ratios for the k-version of the isogeometric finite element method. *Computer Methods in Applied Mechanics and Engineering*, 198(2126):1726 – 1741, 2009. Advances in Simulation-Based Engineering Sciences Honoring J. Tinsley Oden.
- [55] G. Farin. *Curves and Surfaces for CAGD A Practical Guide*. Morgan Kaufmann Publishers, 2002.
- [56] Paul Fischer, Markus Klassen, Julia Mergheim, Paul Steinmann, and Ralf Müller. Isogeometric analysis of 2D gradient elasticity. *Computational Mechanics*, 47(3):325–334, 2011.
- [57] D. Forsey and R. Bartels. Hierarchical B-spline refinement. *Comput. Graphics*, 22:205–212, 1998.
- [58] K.P.S. Gahalaut, J.K. Kraus, and S.K. Tomar. Multigrid methods for isogeometric discretization. *Computer Methods in Applied Mechanics and Engineering*, 253:413 – 425, 2013.
- [59] Carlotta Giannelli, Bert Jüttler, Stefan K. Kleiss, Angelos Mantzaflaris, Bernd Simeon, and Jaka Špeh. THB-splines: An effective mathematical technology for adaptive refinement in geometric design and isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 299:337 – 365, 2016.
- [60] Carlotta Giannelli, Bert Jüttler, and Hendrik Speleers. THB-splines: The truncated basis for hierarchical splines. *Computer Aided Geometric Design*, 29(7):485 – 498, 2012. Geometric Modeling and Processing 2012.

- [61] Héctor Gómez, Victor M. Calo, Yuri Bazilevs, and Thomas J.R. Hughes. Isogeometric analysis of the Cahn–Hilliard phase-field model. *Computer Methods in Applied Mechanics and Engineering*, 197(4950):4333 – 4352, 2008.
- [62] Héctor Gómez, Thomas J.R. Hughes, Xesus Nogueira, and Victor M. Calo. Isogeometric analysis of the isothermal Navier–Stokes–Korteweg equations. *Computer Methods in Applied Mechanics and Engineering*, 199(2528):1828 – 1840, 2010.
- [63] O.A. González-Estrada, E. Nadal, J.J. Ródenas, P. Kerfriden, S.P.A. Bordas, and F.J. Fuenmayor. Mesh adaptivity driven by goal-oriented locally equilibrated superconvergent patch recovery. *Computational Mechanics*, 53(5):957–976, 2014.
- [64] Ch. Heinrich, B. Simeon, and St. Boschert. A finite volume method on NURBS geometries and its application in isogeometric fluid-structure interaction. *Mathematics and Computers in Simulation*, 82(9):1645 – 1666, 2012.
- [65] Clemens Hofreither, Bert Jüttler, Gábor Kiss, and Walter Zulehner. Multigrid methods for isogeometric analysis with THB-splines. *Computer Methods in Applied Mechanics and Engineering*, 308(Supplement C):96 – 112, 2016.
- [66] Md Naim Hossain, Daniel Alves Paladim, Frank Vogel, and Stéphane P. A. Bordas. Implementation of a XFEM toolbox in Diffpack. In *International Conference on Extended Finite Element Methods - XFEM 2013*, Lyon, France, 2013.
- [67] Md Naim Hossain, Frank Vogel, Daniel Alves Paladim, Vinh Phu Nguyen, and Stéphane P. A. Bordas. Implementation of an isogeometric finite element toolbox in Diffpack. In *11th World Congress on Computational Mechanics (WCCM XI)*, Barcelona, Spain, 2014.
- [68] Md Naim Hossain, Gang Xu, Stéphane P. A. Bordas, Elena Atroshchenko, Xuan Peng, Frank Vogel, and Timon Rabczuk. Geometry Independent Field approximaTion (GIFT) for spline based FEM for Linear Elasticity: a Diffpack implementation. In *3rd International Conference on Isogeometric Analysis*, Trondheim, Norway, 2015.
- [69] T. J. R. Hughes. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Dover Publications, Mineola, New York, 2000.
- [70] T.J.R. Hughes, J.A. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194(39-41):41354195, 2005.
- [71] T.J.R. Hughes, A. Reali, and G. Sangalli. Duality and unified analysis of discrete approximations in structural dynamics and wave propagation : Comparison of pmethod finite elements with kmethod NURBS. *Computer Methods in Applied Mechanics and Engineering*, 197(4950):4104 – 4124, 2008.
- [72] J. Linhard J. Kiendl, K.-U. Bletzinger and R. Wuchner. Isogeometric shell analysis with Kirchhoff Love elements. *Computer Methods in Applied Mechanics and Engineering*, 198(49-52):3902 – 3914, 2009.
- [73] L. Jia. Isogeometric contact analysis: Geometric basis and formulation for frictionless contact. *Computer Methods in Applied Mechanics and Engineering*, 200(5-8):726–741, 2011.
- [74] Kjetil André Johannessen. *Isogeometric Analysis Using Locally Refined B-splines*. PhD thesis, Norwegian University of Science and Technology, Faculty of Information Technology, Mathematics and Electrical Engineering, Department of Mathematical Sciences, Trondheim, Norway, 2014.

- [75] Kjetil André Johannessen, Filippo Remonato, and Trond Kvamsdal. On the similarities and differences between Classical Hierarchical, Truncated Hierarchical and LR B-splines. *Computer Methods in Applied Mechanics and Engineering*, 291(Supplement C):64 – 101, 2015.
- [76] D. W. Kelly, J. P. De S. R. Gago, O. C. Zienkiewicz, and I. Babuška. A posteriori error analysis and adaptive processes in the finite element method: Part I error analysis. *International Journal for Numerical Methods in Engineering*, 19(11):1593–1619, 1983.
- [77] J. Kiendl, Y. Bazilevs, M.-C. Hsu, R. Wuchner, and K.-U. Bletzinger. The bending strip method for isogeometric analysis of Kirchhoff – Love shell structures comprised of multiple patches. *Computer Methods in Applied Mechanics and Engineering*, 199(37-40):2403 – 2416, 2010.
- [78] Gábor Kiss, Carlotta Giannelli, Urška Zore, Bert Jüttler, David Gromann, and Johannes Barner. Adaptive CAD model (re-)construction with THB-splines. *Graphical Models*, 76(5):273 – 288, 2014. Geometric Modeling and Processing 2014.
- [79] Stefan K. Kleiss, Bert Jüttler, and Walter Zulehner. Enhancing isogeometric analysis by a finite element-based local refinement strategy. *Computer Methods in Applied Mechanics and Engineering*, 213-216(Supplement C):168 – 182, 2012.
- [80] Stefan K. Kleiss and Satyendra K. Tomar. Guaranteed and sharp a posteriori error estimates in isogeometric analysis. *Computers Mathematics with Applications*, 70(3):167 – 190, 2015.
- [81] Mukesh Kumar, Trond Kvamsdal, and Kjetil André Johannessen. Superconvergent patch recovery and a posteriori error estimation technique in adaptive isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 316:1086 – 1156, 2017. Special Issue on Isogeometric Analysis: Progress and Challenges.
- [82] Hans Petter Langtangen. *Computational Partial Differential Equations Numerical Methods and Diffpack Programming*. Springer, New York, 1999.
- [83] Kang Li and Xiaoping Qian. Isogeometric analysis and shape optimization via boundary integral. *Computer-Aided Design*, 43(11):1427 – 1437, 2011. Solid and Physical Modeling 2011.
- [84] Xin Li, Jiansong Deng, and Falai Chen. Dimensions of Spline spaces over 3D Hierarchical T-meshes. *Journal of Information and Computational Science*, 3(3):487–501, 2006.
- [85] Xin Li and M. A. Scott. Analysis-suitable T-splines: Characterization, refineability, and approximation. *Mathematical Models and Methods in Applied Sciences*, 24(06):1141–1164, 2014.
- [86] H. Lian. *Shape Optimization Directly from CAD: an Isogeometric Boundary Element Approach*. PhD thesis, Cardiff University, 2014.
- [87] H. Lian, P. Kerfriden, and S.P.A. Bordas. Shape sensitivity analysis and optimization using isogeometric boundary element methods in two-dimensional linear elasticity, 2014.
- [88] L. De Lorenzis, I. Temizer, P. Wriggers, and G. Zavarise. A large deformation frictional contact formulation using NURBS–bases isogeometric analysis. *International Journal for Numerical Methods in Engineering*, 87(13):1278–1300, 2011.
- [89] B. Marussig, J. Zechner, G. Beer, C. Duenser, and T. P. Fries. Fast isogeometric boundary element method based on super-parametric representation. In *Proceeding of IGA 2014*, 2014.

- [90] Benjamin Marussig, J. Zechner, Gernot Beer, and Thomas-Peter Fries. Fast isogeometric boundary element method based on independent field approximation. *Computer Methods in Applied Mechanics and Engineering*, 284:458 – 488, 2015. Isogeometric Analysis Special Issue.
- [91] Arif Masud and Raguraman Kannan. B-splines and NURBS based finite element methods for Kohn–Sham equations. *Computer Methods in Applied Mechanics and Engineering*, 241244:112 – 127, 2012.
- [92] N. Moës, M. Cloirec, P. Cartraud, and J.-F. Remacle. A computational approach to handle complex microstructure geometries. *Computer Methods in Applied Mechanics and Engineering*, 192(2830):3163 – 3177, 2003. Multiscale Computational Mechanics for Materials and Structures.
- [93] Dominik Mokriš and Bert Jüttler. Tdhb-splines: The truncated decoupled basis of hierarchical tensor-product splines. *Computer Aided Geometric Design*, 31(7):531 – 544, 2014. Recent Trends in Theoretical and Applied Geometry.
- [94] Dominik Mokriš, Bert Jüttler, and Carlotta Giannelli. On the completeness of hierarchical tensor-product B-splines. *Journal of Computational and Applied Mathematics*, 271(Supplement C):53 – 70, 2014.
- [95] M. Montardini, G. Sangalli, and L. Tamellini. Optimal-order isogeometric collocation at Galerkin superconvergent points. *Computer Methods in Applied Mechanics and Engineering*, 316:741 – 757, 2017. Special Issue on Isogeometric Analysis: Progress and Challenges.
- [96] Mohammed Moumnassi, Salim Belouettar, Éric Béchet, Stéphane P.A. Bordas, Didier Quoirin, and Michel Potier-Ferry. Finite element analysis on implicitly defined domains: An accurate representation based on arbitrary parametric surfaces. *Computer Methods in Applied Mechanics and Engineering*, 200(58):774 – 796, 2011.
- [97] Vinh Phu Nguyen, Pierre Kerfriden, Marco Brino, Stéphane P.A. Bordas, and Elvio Bonisoli. Nitsches method for two and three dimensional NURBS patch coupling. *Computational Mechanics*, 53(6):1163–1182, 2014.
- [98] N. Nguyen-Thanh, J. Muthu, X. Zhuang, and T. Rabczuk. An adaptive three-dimensional RHT-splines formulation in linear elasto-statics and elasto-dynamics. *Computational Mechanics*, 53(2):369–385, 2014.
- [99] N. Nguyen-Thanh, H. Nguyen-Xuan, S.P.A. Bordas, and T. Rabczuk. Isogeometric analysis using polynomial splines over hierarchical T-meshes for two-dimensional elastic solids. *Computer Methods in Applied Mechanics and Engineering*, 200(2122):1892 – 1908, 2011.
- [100] N. Nguyen-Thanh, K. Zhou, X. Zhuang, P. Areias, H. Nguyen-Xuan, Y. Bazilevs, and T. Rabczuk. Isogeometric analysis of large-deformation thin shells using RHT-splines for multiple-patch coupling. *Computer Methods in Applied Mechanics and Engineering*, 316:1157 – 1178, 2017. Special Issue on Isogeometric Analysis: Progress and Challenges.
- [101] P. N. Nielsen, Allan Roulund Gersborg, Jens Gravesen, and Niels Leergaard Pedersen. Discretizations in isogeometric analysis of Navier–Stokes flow. *Computer Methods in Applied Mechanics and Engineering*, 200(4546):3242 – 3253, 2011.
- [102] L. Piegl and W. Tiller. *The NURBS Book*. Springer-Verlag, New York, 1997.

- [103] E. Rank, M. Ruess, S. Kollmannsberger, D. Schillinger, and A. Düster. Geometric modeling, isogeometric analysis and the finite cell method. *Computer Methods in Applied Mechanics and Engineering*, 249252:104 – 115, 2012. Higher Order Finite Element and Isogeometric Methods.
- [104] Martin Ruess, Dominik Schillinger, Ali I. Ozcan, and Ernst Rank. Weak coupling for isogeometric analysis of non-matching and trimmed multi-patch geometries. *Computer Methods in Applied Mechanics and Engineering*, 269:46 – 71, 2014.
- [105] H. Samet. *Foundations of Multidimensional and Metric Data Structures*. Morgan Kaufmann Publishers, 2006.
- [106] Dominik Schillinger, Michael J. Borden, and Henryk K. Stolarski. Isogeometric collocation for phase-field fracture models. *Computer Methods in Applied Mechanics and Engineering*, 284:583 – 610, 2015. Isogeometric Analysis Special Issue.
- [107] Dominik Schillinger, Luca Dedé, Michael A. Scott, John A. Evans, Michael J. Borden, Ernst Rank, and Thomas J.R. Hughes. An isogeometric design-through-analysis methodology based on adaptive hierarchical refinement of NURBS, immersed boundary methods, and T-spline CAD surfaces. *Computer Methods in Applied Mechanics and Engineering*, 249252(0):116 – 150, 2012.
- [108] Dominik Schillinger and Ernst Rank. An unfitted hp-adaptive finite element method based on hierarchical B-splines for interface problems of complex geometry. *Computer Methods in Applied Mechanics and Engineering*, 200(4748):3358 – 3380, 2011.
- [109] Larry L. Schumaker and Lujun Wang. Approximation power of polynomial splines on T-meshes. *Computer Aided Geometric Design*, 29(8):599 – 612, 2012.
- [110] M. A. Scott, M. J. Borden, C. V. Verhoosel, T. W. Sederberg, and T. J. Hughes. Isogeometric finite element data structures based on Bézier extraction of T-splines. *International Journal for Numerical Methods in Engineering*, 88(2):126–156, 2011.
- [111] M. A. Scott, R. N. Simpson, J. A. Evans, S. Lipton, S. P. A. Bordas, T. J. R. Hughes, and T. W. Sederberg. Isogeometric boundary element analysis using unstructured T-splines. *Computer Methods in Applied Mechanics and Engineering*, 254:197–221, 2013.
- [112] M.A. Scott, X. Li, T.W. Sederberg, and T.J.R. Hughes. Local refinement of analysis-suitable T-splines. *Computer Methods in Applied Mechanics and Engineering*, 213216:206 – 222, 2012.
- [113] Thomas W. Sederberg, David L. Cardon, G. Thomas Finnigan, Nicholas S. North, Jianmin Zheng, and Tom Lyche. T-spline Simplification and Local Refinement. *ACM Trans. Graph.*, 23(3):276–283, August 2004.
- [114] Thomas W. Sederberg, Jianmin Zheng, Almaz Bakenov, and Ahmad Nasri. T-splines and T-nurccs. *ACM Trans. Graph.*, 22(3):477–484, July 2003.
- [115] Yu-Deok Seo, Hyun-Jung Kim, and Sung-Kie Youn. Shape optimization and its extension to topological design based on isogeometric analysis. *International Journal of Solids and Structures*, 47(1112):1618 – 1640, 2010.
- [116] R.N. Simpson, S.P.A. Bordas, J. Trevelyan, and T. Rabczuk. A two-dimensional isogeometric boundary element method for elastostatic analysis. *Computer Methods in Applied Mechanics and Engineering*, 209-212:87–100, 2012.

- [117] B. Szabó and I. Babuška. *Finite Element Analysis*. John Wiley and Sons, 1991.
- [118] I. Temizer, P. Wriggers, and T.J.R. Hughes. Contact treatment in isogeometric analysis with NURBS. *Computer Methods in Applied Mechanics and Engineering*, 200(9-12):11001112, 2011.
- [119] I. Temizer, P. Wriggers, and T.J.R. Hughes. Three-Dimensional Mortar-Based frictional contact treatment in isogeometric analysis with NURBS. *Computer Methods in Applied Mechanics and Engineering*, 209(12):115–128, 2012.
- [120] Chien H. Thai, H. Nguyen-Xuan, N. Nguyen-Thanh, T-H. Le, T. Nguyen-Thoi, and T. Rabczuk. Static, free vibration, and buckling analysis of laminated composite Reissner-Mindlin plates using NURBS-based isogeometric approach. *International Journal for Numerical Methods in Engineering*, 91(6):571–603, 2012.
- [121] Li Tian, Falai Chen, and Qiang Du. Adaptive finite element methods for elliptic equations over hierarchical t-meshes. *Journal of Computational and Applied Mathematics*, 236(5):878 – 891, 2011. The 7th International Conference on Scientific Computing and Applications, June 1316, 2010, Dalian, China.
- [122] S.P. Timoshenko and J.N. Goodier. *Theory of Elasticity (3rd edn)*. McGraw, New York, 1970.
- [123] Navid Valizadeh, Sundararajan Natarajan, Octavio A. Gonzalez-Estrada, Timon Rabczuk, Tinh Quoc Bui, and Stéphane P.A. Bordas. Nurbs-based finite element analysis of functionally graded plates: Static bending, vibration, buckling and flutter. *Composite Structures*, 99:309 – 326, 2013.
- [124] Clemens V. Verhoosel, Michael A. Scott, Thomas J. R. Hughes, and René de Borst. An isogeometric analysis approach to gradient damage models. *International Journal for Numerical Methods in Engineering*, 86(1):115–134, 2011.
- [125] A.-V. Vuong, C. Giannelli, B. Jüttler, and B. Simeon. A hierarchical approach to adaptive local refinement in isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 200(4952):3554 – 3567, 2011.
- [126] L.B. Wahlbin. *Superconvergence in Galerkin Finite Element Methods*. 1995.
- [127] Wolfgang A. Wall, Moritz A. Frenzel, and Christian Cyron. Isogeometric structural shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 197(3340):2976 – 2988, 2008.
- [128] Ping Wang, Jinlan Xu, Jiansong Deng, and Falai Chen. Adaptive isogeometric analysis using rational PHT-splines. *Computer-Aided Design*, 43(11):1438 – 1448, 2011. Solid and Physical Modeling 2011.
- [129] Xiaodong Wei, Yongjie Zhang, Lei Liu, and Thomas J.R. Hughes. Truncated T-splines: Fundamentals and methods. *Computer Methods in Applied Mechanics and Engineering*, 316(Supplement C):349 – 372, 2017. Special Issue on Isogeometric Analysis: Progress and Challenges.
- [130] G. Xu, E. Atroshchenko, and S. P. A. Bordas. Geometry-independent field approximation for spline-based finite element methods. In *11th World Congress on Computational Mechanics (WCCM XI)*, 2014.
- [131] Yongjie Zhang, Yuri Bazilevs, Samrat Goswami, Chandrajit L. Bajaj, and Thomas J.R. Hughes. Patient-specific vascular NURBS modeling for isogeometric analysis of blood flow. *Computer Methods in Applied Mechanics and Engineering*, 196(2930):2943 – 2959, 2007.

-
- [132] J.Z. Zhu. A posteriori error estimationthe relationship between different procedures. *Computer Methods in Applied Mechanics and Engineering*, 150(1):411 – 422, 1997. Symposium on Advances in Computational Mechanics.
- [133] O. C. Zienkiewicz and J. Z. Zhu. The superconvergence Patch Recovery and a Posteriori Error Estimation in the Finite Element Method, Part 1: The Recovery Technique. *International Journal for Numerical Methods in Engineering*, 33(7):1331–1364, 1992.
- [134] O. C. Zienkiewicz and J. Z. Zhu. The Superconvergence Patch Recovery and a Posteriori Error Estimation in the Finite Element Method, Part 2: Error Estimates and Adaptivity. *International Journal for Numerical Methods in Engineering*, 33(7):1364–1382, 1992.

Curriculum Vitae

Md Naim Hossain

inuTech GmbH

Fürther str. 212

90429, Nuremberg, Germany

Email: mdnaim.hossain@inutech.de

Date of Birth: 19th December, 1984

Place of Birth: Dhaka, Bangladesh

Citizenship: Bangladeshi

Current position

Research associate and software developer

inuTech GmbH

Fürther str. 212

90429, Nuremberg, Germany

Education

- External Ph.D student: Institute of Structural Mechanics, Bauhaus-Universität Weimar, Germany, 2014-2018.
- M.Sc: Erasmus Mundus Master of Science in Computational Mechanics, Swansea University, Swansea, UK, 2010-2011.
- M.Sc: Erasmus Mundus Master of Science in Computational Mechanics, Technical University of Catalunya, Barcelona, Spain, 2011-2012.
- B.Sc: Bachelor of Science in Mechanical Engineering, Bangladesh University of Engineering and Technology, Dhaka, Bangladesh, 2004-2009.

Research Interest

Computational Mechanics, Finite Element Method, Isogeometric Analysis, Fracture Mechanics, Numerical Methods

Skills and Expertise

Matlab, C++, C#, Fortran, Latex, Ansys, Abaqus, Qt, Numerical Analysis, Finite Element Analysis, CAD.

Awards & Scholarships

- Marie-Curie Initial Training Network (ITN) funding provided by EU Seventh Framework Program (FP7) as an Early Stage Researcher.
- European Union Erasmus Mundus scholarship for the master's program in Computational Mechanics funded by Erasmus Mundus Consortium (led by CIMNE, Universitat Politècnica de Catalunya, Barcelona, Spain).

Publications & talks

Journal articles

1. Cosmin Anitescu, **Md Naim Hossain**, Timon Rabczuk. Recovery-based error estimation and adaptivity using high-order splines over hierarchical T-meshes. *Computer Methods in Applied Mechanics and Engineering*. 328(638-662), 2018.

Conference Talks

1. **Md Naim Hossain**, Daniel Alves Paladim, Frank Vogel, Stephane P. A. Bordas. Implementation of XFEM toolbox in Diffpack. International Conference on Extended Finite Element Methods, XFEM 2013, Lyon, France, 2013.
2. **Md Naim Hossain**, Frank Vogel, Daniel Alves Paladim, Vinh Phu Nguyen, Stephane P. A. Bordas. Implementation of an isogeometric finite element toolbox in Diffpack. 11th World Congress on Computational Mechanics, Barcelona, Spain, 2014.
3. **Md Naim Hossain**, Gang Xu, Stephane P.A. Bordas, Elena Atroshchenko, Xuan Peng, Frank Vogel, Timon Rabczuk. Geometry-Independent Field approximation (GIFT) for spline based FEM for Linear Elasticity: a Diffpack implementation, 3rd International Conference on Isogeometric Analysis IGA 2015, Trondheim, Norway, 2015.
4. Cosmin Anitescu, **Md Naim Hossain**, Timon Rabczuk. IGAPack: an isogeometric hierarchical spline package using recovery-based error estimation and adaptivity, EMI 2017 Engineering Mechanics Institute Conference, San Diego, USA, 2017.

Workshop Presentations

1. Implementation of h-refinement convergency of isogeometric analysis. Presentation, ITN-INSIST workshop, 23 - 27 September, 2013, Weimer, Germany.
2. Isogeometric Finite Element Toolbox: a Diffpack implementation. Presentation, ITN-INSIST seminar, 26 - 30 Januar, 2015, Strobl, Austria.
3. Higher order and higher continuity Polynomial Spline over hierarchical T-Meshes. Presentation, ITN-INSIST Conference, 15 - 17 September, 2015, Cardiff, UK.