

# Integrating CAD Geometry and Scaled Boundary Finite Element Analysis

**Author:**

Wang, Junchao

**Publication Date:**

2018

**DOI:**

<https://doi.org/10.26190/unsworks/20883>

**License:**

<https://creativecommons.org/licenses/by-nc-nd/3.0/au/>

Link to license to see what you are allowed to do with this resource.

Downloaded from <http://hdl.handle.net/1959.4/60921> in <https://unsworks.unsw.edu.au> on 2024-05-04

# **Integrating CAD Geometry and Scaled Boundary Finite Element Analysis**

**Junchao Wang**

*A thesis in fulfilment of the requirements for the degree of  
Doctor of Philosophy*



**UNSW**  
THE UNIVERSITY OF NEW SOUTH WALES

School of Civil & Environmental Engineering  
Faculty of Engineering

November 21, 2018



Australia's  
Global  
University

# Thesis/Dissertation Sheet

Surname/Family Name	Wang
Given Name/s	Junchao
Abbreviation for degree as give in the University calendar	Ph. D
Faculty	Faculty of Engineering
School	School of Civil & Environmental Engineering
Thesis Title	Integrating CAD geometry and scaled boundary finite element analysis

## Abstract 350 words maximum

The Finite element method (FEM) constitutes a general tool for the numerical solution in. Although it is a principle method in the engineering field, deficiency in geometric representation has been detected. Besides, it could be expensive in terms of time and human resource to create the mesh required by the FEM. The research towards integrating geometry and analysis has led to the 'Isogeometric Analysis' (IGA) (Hughes et al., 2005). However, as the CAD model provides information only of the boundary, a 2D/3D stress analysis is still one major step away. This thesis presents a simple and efficient technique based on the combination of the scales boundary finite element method (SBFEM), automatic mesh generation and adaptive refinement algorithms to reduce the human efforts in the structural analysis. This framework will also be further extended to problems with singularities and to dynamic analysis. To mode problems with complex geometries, the problems domains are divided into a mesh of scaled boundary finite elements. A quad-tree based mesh generation algorithm is developed to provide high quality mesh. Furthermore, no human efforts are required for the pre-processing as the output of the CAD will be used to determine the geometric information automatically.

To ensure a controllable accuracy and minimal computational cost, an adaptive mesh refinement algorithm is also developed. The expressions related to the eigenvalues of the SBFEM formulation representing the quantity of the error in the interpolation are adopted together with the area and other geometric properties of the Scaled Boundary Finite Element. A machine learning model using the Multilayer Perceptron (MLP) is trained to determine whether a Scaled Boundary Finite Element needs refinement or not based on all these information.

The proposed method is further extended to 3D with an initial mesh generated based on the STL file and octree algorithm. The octree mesh provides a high quality mesh in 3D for SBFEM and the IGES file from the CAD software will be adopted in order to map intersection points back to NURBS surfaces to preserve an exact geometry. The convex hull properties of the NURBS are utilized to accelerate the algorithm.

## Declaration relating to disposition of project thesis/dissertation

I hereby grant to the University of New South Wales or its agents the right to archive and to make available my thesis or dissertation in whole or in part in the University libraries in all forms of media, now or here after known, subject to the provisions of the Copyright Act 1968. I retain all property rights, such as patent rights. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation.

Microfilms to use the 350 word abstract of my thesis in

International (this is applicable to doctoral

Signature

Witness Signature

Date

The University recognises that there may be exceptional circumstances requiring restrictions on copying or conditions on use. Requests for restriction for a period of up to 2 years must be made in writing. Requests for a longer period of restriction may be considered in exceptional circumstances and require the approval of the Dean of Graduate Research.

OR OFFICE USE ONLY Date of completion of requirements for Award:

## COPYRIGHT STATEMENT

'I hereby grant the University of New South Wales or its agents the right to archive and to make available my thesis or dissertation in whole or part in the University libraries in all forms of media, now or here after known, subject to the provisions of the Copyright Act 1968. I retain all proprietary rights, such as patent rights. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation.

I also authorise University Microfilms to use the 350 word abstract of my thesis in Dissertation Abstract International (this is applicable to doctoral theses only).

I have either used no substantial portions of copyright material in my thesis or I have obtained permission to use copyright material; where permission has not been granted I have applied/will apply for a partial restriction of the digital copy of my thesis or dissertation.'

Signed

2

Date

31/08/18

## AUTHENTICITY STATEMENT

'I certify that the Library deposit digital copy is a direct equivalent of the final officially approved version of my thesis. No emendation of content has occurred and if there are any minor variations in formatting, they are the result of the conversion to digital format.'

Signed

31/08/18

Date



## INCLUSION OF PUBLICATIONS STATEMENT

UNSW is supportive of candidates publishing their research results during their candidature as detailed in the UNSW Thesis Examination Procedure.

### **Publications can be used in their thesis in lieu of a Chapter if:**

- The student contributed greater than 50% of the content in the publication and is the "primary author", ie. the student was responsible primarily for the planning, execution and preparation of the work for publication
- The student has approval to include the publication in their thesis in lieu of a Chapter from their supervisor and Postgraduate Coordinator.
- The publication is not subject to any obligations or contractual agreements with a third party that would constrain its inclusion in the thesis

Please indicate whether this thesis contains published material or not.

☐

*This thesis contains no publications, either published or submitted for publication*

☒

*Some of the work described in this thesis has been published and it has been documented in the relevant Chapters with acknowledgement*


☐

*This thesis has publications (either published or submitted for publication) incorporated into it in lieu of a chapter and the details are presented below*

### **CANDIDATE'S DECLARATION**

I declare that:

- I have complied with the Thesis Examination Procedure
- where I have used a publication in lieu of a Chapter, the listed publication(s) below meet(s) the requirements to be included in the thesis.

Name	Signature	Date (dd/mm/yy)
Junchao Wang		29/08/18

### **Postgraduate Coordinator's Declaration**

I declare that:

- the information below is accurate
- where listed publication(s) have been used in lieu of Chapter(s), their use complies with the Thesis Examination Procedure
- the minimum requirements for the format of the thesis have been met.

PGC's Name	PGC's Signature	Date (dd/mm/yy)

#### ORIGINALITY STATEMENT

'I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, or substantial proportions of material which have been accepted for the award of any other degree or diploma at UNSW or any other educational institution, except where due acknowledgement is made in the thesis. Any contribution made to the research by others, with whom I have worked at UNSW or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the project's design and conception or in style, presentation and linguistic expression is acknowledged.'

Signed .....

31/08/18

Date .....

# Acknowledgement

Foremost, I would like to express my sincere gratitude to my supervisor Prof. Chongmin Song and Dr.Sundararajan Natarajan for the continuous support of my research, for their patience, motivation, enthusiasm, and immense knowledge. their guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my Ph. D Thesis.

# Abstract

The Finite element method (FEM) constitutes a general tool for the numerical solution of partial differential equations in engineering and applied science. Great amount of research has been conducted on FEM in terms of mathematics and applications, contributing to its dominance over numerical method in solid mechanics and structural analysis. Although it is a principle method for solving complex problems in the engineering field, deficiency in geometric representation has been detected. Besides, it could be expensive in terms of time and human resource to create the mesh required by the FEM. The research towards integrating geometry and analysis has led to the Isogeometric Analysis (IGA) (Hughes et al., 2005). However, as the CAD model provides information only of the boundary, a 2D/3D stress analysis is still one major step away.

This thesis presents a simple and efficient technique based on the combination of the scales boundary finite element method (SBFEM), automatic mesh generation and adaptive refinement algorithms to reduce the

---

human efforts in the structural analysis. In the SBFEM, only the boundary information is required and hence a seamless integration can be provided with the CAD modelling. The NURBS basis functions are adopted to discretize the unknown fields in the circumferential direction within the proposed framework, whilst analytical solution is sought in the radial direction. This framework will also be further extended to problems with singularities and to dynamic analysis.

To mode problems with complex geometries, the problems domains are divided into a mesh of scaled boundary finite elements. A quad-tree based mesh generation algorithm is developed. High quality mesh will be generated with the help of the algorithm and the computational cost will also be improved due to the utilization of the patterns in the quad-tree. Furthermore, no human efforts are required for the pre-processing as the output of the CAD software (i.e. IGES file) will be used to determine the geometric information automatically. Any mismatch between the geometric representation in design and in numerical analysis may be prevented as the design is used directly.

To ensure a controllable accuracy and minimal computational cost, an adaptive and robust mesh refinement algorithm is also developed to prevent unnecessary refinement in the region which contributes little to the improvement to the accuracy. The expressions related to the eigenvalues of the SBFEM formulation representing the quantity of the error in

---

the interpolation are adopted as one of the error indicators, together with the area and other geometric properties of the Scaled Boundary Finite Element. A machine learning model using the Multilayer Perceptron (MLP) is trained to determine whether a Scaled Boundary Finite Element needs refinement or not based on all these information.

The proposed method is further extended to 3D with an initial mesh generated based on the STL file and octree algorithm. The octree mesh provides a high quality mesh in 3D for SBFEM and the IGES file from the CAD software will be adopted in order to map intersection points back to NURBS surfaces to preserve an exact geometry. The convex hull properties of the NURBS are utilized to accelerate the algorithm.

Numerical examples are presented to verify the proposed technique with the results from the literature and the numerical results obtained using the commercial software ANSYS. The accuracy and the convergence properties of the proposed method are demonstrated with benchmark problems in the context of linear elasticity and linear elastic fracture mechanics. The presented results show a higher accuracy and rate of convergence of the proposed method.

# Publications

## Journal papers

1. Sundararajan Natarajan, **JunChao Wang** , Chongmin Song , Carolin Birk (2015). Isogeometric analysis enhanced by the scaled boundary finite element method. *Computer Methods in Applied Mechanics and Engineering*, 283:733-762
2. Yan Liu , Albert A. Saputra, **Junchao Wang**, Francis Tin-Loi, Chongmin Song (2017). Automatic polyhedral mesh generation and scaled boundary finite element analysis of STL models. *Computer Methods in Applied Mechanics and Engineering*, 313:106-132

# Nomenclature

## *Greek Letters*

$\epsilon$	Threshold
$\{\epsilon\}$	Strain tensor
$\zeta(x)$	Softplus function
$\kappa$	Kolosov constants
$\lambda$	Eigenvalue
$\nu$	Poisson's ratio
$(\xi, \eta, \zeta)$	Scaled boundary finite element coordinates
$\{\Xi\}$	NURBS knot vector
$\rho$	Density
$\sigma$	Logistic sigmoid
$\{\sigma\}$	Stress tensor
$\{\Phi\}$	Surface tractions
$\{\psi_\sigma\}$	Stress mode



---

### *Latin Letters*

$\{B\}$	Minibatch
$\{c\}$	Integrating constants
$\{C(u)\}$	NURBS curve
$E$	Young's modulus
$E_{a \sim b}$	Cross entropy between data set $a$ and $b$
$[E_0], [E_1], [E_2]$	Element coefficient matrices
$\{F\}$	Nodal forces
$G$	Shear modulus
$J(\theta)$	Cost function
$ J $	Determinant of Jacobian matrix
$[J(\xi, \eta, \zeta)]$	Jacobian matrix
$K_I, K_{II}$	Stress intensity factors for mode I and II
$[K]$	Stiffness matrix
$L(x, y, \theta)$	Pre-example Loss function
$[L]$	Linear differential operator
$[N(\eta)]$	Shape function

---

$p$	Order of the shape function
$[P]$	NURBS control points
$\{q(\xi)\}$	Internal force vector
$(r, \theta)$	Polar coordinates
$S(u, v)$	NURBS surface
$\{u\}$	Displacement vector
$[w]$	NURBS surface weight matrix
$\{w\}$	NURBS curve weight vector
$(x, y)$	Nodal coordinates
$[Z]$	Hamiltonian coefficient matrix

# Contents

<b>Acknowledgement</b>	<b>iv</b>
<b>Abstract</b>	<b>v</b>
<b>Nomenclature</b>	<b>ix</b>
<b>List of Tables</b>	<b>xvii</b>
<b>List of Figures</b>	<b>xxiv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	1
1.2 State of the problem . . . . .	6
1.3 Objective and significance . . . . .	8
1.4 Outline of the thesis . . . . .	9
<b>2 Literature review</b>	<b>13</b>
2.1 Isogeometric analysis . . . . .	13
2.1.1 Initial Graphics Exchange Specification (IGES) file .	16
2.1.2 Non-Uniform Rational B-Spline (NURBS) . . . . .	17

## CONTENTS

---

2.2	Scaled boundary finite element method . . . . .	25
2.3	Adaptive mesh refinement . . . . .	29
2.3.1	Machine learning . . . . .	30
2.3.2	Multilayer perceptron(MLP) . . . . .	32
2.3.3	Gradient based optimization . . . . .	35
2.3.4	Stochastic gradient descent (SGD) . . . . .	38
2.3.5	Cost function . . . . .	39
2.3.6	Output units . . . . .	40
2.3.7	Hidden units . . . . .	43
2.3.8	Performance indicators . . . . .	45
2.4	Automatic mesh generation in 3D . . . . .	48
2.4.1	STL file . . . . .	51
2.5	Conclusions . . . . .	52
<b>3</b>	<b>Isogeometric enhanced SBFEM in 2D</b>	<b>53</b>
3.1	Introduction . . . . .	53
3.2	Formulation of the isogeometric SBFEM . . . . .	54
3.3	Application of isogeometric SBFEM to linear elastic fracture mechanics . . . . .	62
3.4	Surface traction . . . . .	65
3.5	Displacement interpolation . . . . .	68
3.6	Numerical integrations . . . . .	68
3.7	Numerical examples . . . . .	70
3.7.1	Cantilever beam . . . . .	71
3.7.2	Infinite plate with a circular hole . . . . .	75

## CONTENTS

---

3.7.3	L-shaped bracket . . . . .	78
3.7.4	Circular disk with an edge crack in tension . . . . .	86
3.7.5	Edge crack in tension . . . . .	90
3.7.6	Angled crack in an orthotropic body . . . . .	93
3.7.7	Transient analysis of bimaterial plate with a notch . .	96
3.8	Conclusions . . . . .	99
<b>4</b>	<b>Quad-tree mesh in 2D analysis</b>	<b>102</b>
4.1	Introduction . . . . .	102
4.2	CAD output in 2D . . . . .	103
4.2.1	Parse geometry in IGES file . . . . .	104
4.2.2	Output to mesh generator . . . . .	104
4.2.3	Discretization of the circular arc . . . . .	106
4.2.4	Discretization of the NURBS curve . . . . .	106
4.3	Quad-tree structure . . . . .	111
4.3.1	Background mesh . . . . .	111
4.3.2	Hard point treatment . . . . .	116
4.3.3	Bucket sort algorithm . . . . .	119
4.3.4	Cutting with boundary . . . . .	120
4.3.5	Points moving . . . . .	123
4.3.6	Color the region . . . . .	127
4.4	Points projection . . . . .	133
4.4.1	Projection algorithm . . . . .	133
4.4.2	Convex hull in 2D . . . . .	136
4.5	Numerical Examples . . . . .	137

## CONTENTS

---

4.5.1	Cantilever beam . . . . .	137
4.5.2	Infinite plate with a circular hole . . . . .	141
4.5.3	Plane strain bracket . . . . .	147
4.5.4	Buildings on the ground . . . . .	153
4.5.5	Other meshes . . . . .	154
4.6	Conclusions . . . . .	156
<b>5</b>	<b>Adaptivity</b>	<b>161</b>
5.1	Introduction . . . . .	161
5.2	Error indicator . . . . .	162
5.2.1	Mesh Size . . . . .	163
5.2.2	Mesh Quality . . . . .	164
5.2.3	Eigenvalue in SBFEM . . . . .	164
5.3	Machine learning . . . . .	166
5.3.1	Training Set . . . . .	167
5.3.2	Regularization for MLP . . . . .	168
5.3.3	Result . . . . .	177
5.4	Merging triangle . . . . .	178
5.5	Matrix representation of NURBS Curves . . . . .	182
5.5.1	Implicit Matrix Representation . . . . .	182
5.5.2	Matrix Representation for Rational Bzier Curves . . . . .	183
5.5.3	Intersection . . . . .	185
5.6	Numerical examples . . . . .	189
5.6.1	Short cantilever beam . . . . .	189
5.6.2	Infinite plate with a circular hole . . . . .	190

## CONTENTS

---

5.6.3	Plane strain bracket . . . . .	192
5.7	Conclusions . . . . .	194
<b>6</b>	<b>Isogeometric enhanced SBFEM in 3D</b>	<b>201</b>
6.1	Introduction . . . . .	201
6.2	Projection . . . . .	202
6.2.1	Surfaces division . . . . .	202
6.2.2	Projection algorithm . . . . .	204
6.2.3	Convex hull in 3D . . . . .	209
6.3	Intersection . . . . .	210
6.3.1	Matrix Representation for Rational Bzier Surface . .	211
6.3.2	Properties of the $M_v$ Matrix . . . . .	211
6.4	Introduction of SBFEM in 3D . . . . .	217
6.5	Numerical examples . . . . .	222
6.5.1	Pressurized hollow sphere . . . . .	222
6.5.2	Capsule Cutting From the Cuboid with Bending . . .	224
6.5.3	Wall lamp . . . . .	230
6.5.4	Meshing examples . . . . .	231
6.6	Conclusions . . . . .	232
<b>7</b>	<b>Conclusions and recommendation</b>	<b>237</b>
7.1	Summary . . . . .	237
7.2	Future work . . . . .	242
	<b>Bibliography</b>	<b>243</b>

# List of Tables

2.1	Confusion matrix . . . . .	45
3.1	T-stress and stress intensity factors for circular disk with an edge crack. . . . .	89
3.2	Convergence of the mode I SIF and T-stress for an edge crack in tension . . . . .	93
3.3	Convergence of the generalized stress intensity factors $(\overline{K_I}, \overline{K_{II}}) = (K_I, K_{II})/P\sqrt{(\pi a)}$ . . . . .	95
3.4	Generalized stress intensity factors $K_I/P\sqrt{\pi a}$ of angled crack in rectangular orthotropic body . . . . .	95
3.5	Generalized stress intensity factors $K_{II}/P\sqrt{\pi a}$ of angled crack in rectangular orthotropic body . . . . .	96
4.1	Minimal $\epsilon_2$ with different $\epsilon_1$ . . . . .	129
5.1	Result of cross validation . . . . .	178



# List of Figures

2.1	File structure of an IGES file . . . . .	17
2.2	B-spline functions: knot insertion . . . . .	22
2.3	B-spline functions: order elevation . . . . .	23
2.4	3rd NURBS basis function . . . . .	24
2.5	Two dimensional scaled boundary coordinates . . . . .	26
2.6	A typical topology of a multilayer perceptron . . . . .	34
2.7	Example of gradient descent algorithm on a parabola . . . . .	36
2.8	Rectified linear units(RELU) . . . . .	43
2.9	Receiver operating characteristic(ROC) . . . . .	48
3.1	Cantilever beam: Geometry and boundary conditions. . . . .	72
3.2	Bending of thick cantilever beam: Convergence results . . . . .	76
3.3	Infinite plate with a circular hole: geometry and boundary conditions . . . . .	77
3.4	Plate with circular hole: mesh . . . . .	78
3.5	Bending of thick cantilever beam: Convergence results . . . . .	79
3.6	NURBS basis functions and Lagrange basis functions . . . . .	80
3.7	Infinite plate with a circular hole: Convergence results . . . . .	80

## LIST OF FIGURES

---

3.8	Geometry and boundary conditions . . . . .	81
3.9	Control net where ‘filled’ circles represents control points .	82
3.10	Von Mises equivalent stress contours for L-shaped bracket without and with fillet . . . . .	83
3.11	L-shaped bracket: geometry and boundary conditions for transient analysis . . . . .	84
3.12	Vertical displacement of L-shaped bracket at Point A: com- parison with conventional FE solution . . . . .	86
3.13	Circular disk with an edge crack . . . . .	87
3.14	Meshing of the circular disk with an edge notch . . . . .	89
3.15	Displacement and stress modes of circular disk with an edge crack using cubic NURBS functions . . . . .	91
3.16	Circular disk with an edge crack: convergence of the dis- placement mode and stress mode . . . . .	92
3.17	Plate with an edge crack under tension . . . . .	93
3.18	Angled crack in a rectangular orthotropic body: geometry .	94
3.19	Bimaterial plate with a notch: geometry, boundary condi- tions and force history . . . . .	97
3.20	Non-dimensional natural frequencies of a bimaterial plate with a notch . . . . .	98
3.21	Vertical displacements of bimaterial plate with a notch . . .	100
4.1	Chord length for circular arc . . . . .	106
4.2	Type of sub-devided NURBS curves: convex and concave with an inflection point . . . . .	107

## LIST OF FIGURES

---

4.3	Discretization for convex NURBS curve . . . . .	108
4.4	Initial background mesh . . . . .	112
4.5	Limitation of the seed points . . . . .	113
4.6	Balance of the AVL tree (Roura, 2013) . . . . .	114
4.7	Balance of quadtree: cell A is refined in order to balance the quadtree. . . . .	114
4.8	Small angle between element and scaling center may re- duce the mesh quality . . . . .	115
4.9	Types of the cell in self-balancing quadtree . . . . .	116
4.10	Example of a hard point . . . . .	117
4.11	Hard point treatment step 1 . . . . .	119
4.12	Bucket sort in quadtree . . . . .	121
4.13	Situations in cutting . . . . .	122
4.14	Hard point treatment in cutting . . . . .	122
4.15	Poor quality of mesh . . . . .	124
4.16	Treatment of cell cutting when point is moved . . . . .	125
4.17	Problem in marking intersection . . . . .	126
4.18	Concave cell resulted from points moving . . . . .	126
4.19	Coloring problem using location of geometric center in con- cave polygon . . . . .	128
4.20	Coloring problem using location of scaling center . . . . .	128
4.21	Example for coloring . . . . .	130
4.22	Example for coloring: step 1 . . . . .	131
4.23	Example for coloring: step 2 . . . . .	132
4.24	Example for coloring: final tree . . . . .	132

## LIST OF FIGURES

---

4.25 Cantilever beam: Geometry and boundary conditions. . . .	137
4.26 Background mesh of cantilever beam . . . . .	139
4.27 Mesh coloring of cantilever beam . . . . .	139
4.28 Final mesh of cantilever beam . . . . .	139
4.29 Mesh of the cantilever beam . . . . .	140
4.30 Convergence of the cantilever beam . . . . .	141
4.31 Infinite plate with a circular hole: geometry and boundary conditions . . . . .	142
4.32 Infinite plate with a circular hole: CAD drawing . . . . .	142
4.33 Background mesh of infinite plate with a circular hole . . . .	143
4.34 Mesh coloring of infinite plate with a circular hole . . . . .	144
4.35 Final mesh of infinite plate with a circular hole . . . . .	144
4.36 Mesh of the infinite plate with a circular hole . . . . .	146
4.37 Convergence of the infinite plate with a circular hole . . . .	146
4.38 Plane strain bracket: geometry and boundary conditions . .	147
4.39 ANSYS mesh for plane strain bracket (62004 DOF) . . . . .	148
4.40 CAD drawing of plane strain bracket . . . . .	149
4.41 Background mesh of the bracket . . . . .	150
4.42 Mesh coloring of the bracket . . . . .	150
4.43 Final mesh of the bracket . . . . .	151
4.44 Mesh of the plane strain bracket . . . . .	152
4.45 Convergence of the plane strain bracket . . . . .	153
4.46 Von Mises equivalent stress contours for plane strain bracket	154
4.47 Geometry and boundary condition of the problem . . . . .	155
4.48 Geometry and boundary condition of the problem . . . . .	155

## LIST OF FIGURES

---

4.49 Design in AutoCAD . . . . .	156
4.50 Meshing of the buildings . . . . .	157
4.51 Von-mises stress contour for the buidlings on the ground . .	158
4.52 Mesh of flower . . . . .	159
4.53 Mesh of wolli logo . . . . .	160
5.1 A polygon with $n$ vertexes . . . . .	163
5.2 Mesh quality in SBFEM . . . . .	164
5.3 Mesh refinement for square plate with circular hole (Duval et al., 2018) . . . . .	168
5.4 Mesh refinement of a short cantilever beam (Zienkiewicz et al., 2005) . . . . .	169
5.5 Training data for SVM . . . . .	170
5.6 Accuracy, precision, recall rate and F1 score vs different class weight . . . . .	177
5.7 Merging triangle . . . . .	179
5.8 Choose the cell to merge with . . . . .	180
5.9 Scaling center for concave polygon . . . . .	182
5.10 Short cantilever beam: Geometry and boundary conditions.	189
5.11 the relative error in the energy norm . . . . .	190
5.12 Short cantilever beam: mesh development (SBFEM) . . . .	191
5.13 Short cantilever beam: mesh development (Ansys) . . . . .	192
5.14 Von-mises stress contour using 9-node quadrilateral ele- ment in ANSYS . . . . .	193

## LIST OF FIGURES

---

5.15 Infinite plate with a circular hole: geometry and boundary conditions . . . . .	193
5.16 Infinite plate with a circular hole: Convergence study . . . .	194
5.17 Infinite plate with a circular hole: Mesh development (SBFEM)	195
5.18 Infinite plate with a circular hole: Mesh development (Ansys)	196
5.19 Von-mises stress contour using 9-node quadrilateral element in ANSYS . . . . .	197
5.20 Plane strain bracket: geometry and boundary conditions . .	198
5.21 Plane strain bracket: Convergence study . . . . .	198
5.22 Plane strain bracket: Mesh development (SBFEM) . . . . .	199
5.23 Plane strain bracket: Mesh development (Ansys) . . . . .	200
6.1 NURBS surface subdivision . . . . .	204
6.2 Polyhedral SBFEM subdomain in 3D . . . . .	217
6.3 Pressurized hollow sphere . . . . .	222
6.4 Coordinate transformation . . . . .	224
6.5 Mesh of the hollow sphere with 1716 DOFs . . . . .	225
6.6 Mesh of the hollow sphere with 3896 DOFs . . . . .	226
6.7 Mesh of the hollow sphere with 12078 DOFs . . . . .	227
6.8 Convergence of displacement error . . . . .	228
6.9 Problem layout . . . . .	228
6.10 Geometry of the capsule . . . . .	229
6.11 Mesh of the Capsule . . . . .	229
6.12 Boundary condition of the wall lamp . . . . .	231
6.13 CAD drawing of the wall lamp . . . . .	231

## LIST OF FIGURES

---

6.14 Background mesh of the wall lamp . . . . .	232
6.15 Mesh of the wall lamp . . . . .	233
6.16 Deformed shape of the wall lamp . . . . .	234
6.17 CAD design for spinner . . . . .	234
6.18 Mesh for the spinner . . . . .	235
6.19 CAD design for the Egypt Sphinx . . . . .	235
6.20 Mesh for the Egypt Sphinx . . . . .	236

# Chapter 1

## Introduction

### 1.1 Background

Structural engineering is one of the aspects in civil engineering that aims at the design of the structure which supports the loads without failure. The loads imposed may be any physical forces due to gravity, wind action, vibration, temperature change, etc. In the ancient age, it helped the people to build and maintain mega-structures such as pyramid and sphinx in Egypt. Nowadays, thanks to the tremendous development in mathematics and material science, an increasingly wider range of different complex structures becomes possible to analysis.

As a result of the increasingly complicated geometry in structural engineering and the birth of the computer, Computer Aided Design (CAD)



## 1.1. BACKGROUND

---

has been developed in 1980s (Weisberg, 2008). The first Unigraphics System (for 2D modelling and drafting) was sold by United Computing in 1975 (Stefano and Politecnico, 2010). Nowadays, CAD has been widespread and won significant popularity. A predominant amount of the designs delivered to structural engineers are generated by the help of commercial CAD softwares. Besides, the CAD has also been extended to other fields especially mechanical and aerospace engineering where extreme complex geometry are treated.

The traditional structural analysis method which depends on a closed form mathematical solution becomes incapable to handle the highly complicated geometric input. This motivated the Finite element method (FEM) which constitutes a general tool for the numerical solution of partial differential equations in engineering and applied science to be proposed (Berkeley, 2010). It did not achieve enormous popularity until early 1950s, when digital computer was developed. After that, the method was refined with the help of variational methods from Lord Rayleigh (1870) and W. Ritz (1909) as well as the Galerkin's weighted-residual approach (Felippa, 1994). Then, great amount of research has been conducted on the FEM in terms of mathematics and applications, contributing to its dominance over numerical method in solid mechanics and structural analysis by the beginning of the 1990s (Clough, 1980). Although it is a principle method for solving complex problems in the engineering field, lack of the local mesh refinement and the inability to formulate unbounded domain restrict

## 1.1. BACKGROUND

---

the usage of the FEM in some area. As a consequence, other numerical methods such as Boundary Element Method (BEM) (Li and Qian, 2011; Wardle, 1984) and the Extended FEM (X-FEM) (Moës et al., 1999) were proposed. In the BEM, only the boundary was discretised, contributing to a reduction of the spatial dimension by one. Besides, the problem involving the unbounded domain can be solved naturally. Nevertheless, the fundamental solution satisfying the governing differential equations in the domain must be available. Unfortunately, this fundamental solution may be extremely complex. The X-FEM extended the FEM to solve problems with localized features that are not efficiently resolved by mesh refinement. Compared to the traditional FEM, the X-FEM exhibits a strong ability of modeling the fractures in the material.

The FEM could be one of the most popular numerical methods in engineering analysis. In the FEM, a meshing procedure that discretize the problem domain into individual elementary components or “elements” is necessary. The solution of the whole system is calculated by assembling its discretized elements. However, creation of mesh in FEM can be expensive in terms of time and human resource. Numerous studies have been conducted for an automatic mesh generation algorithm (Blacker and Meyers, 1993; Löhner and Parikh; Owen, 2000; Watson, 1981).

When complex geometric input is involved, chances are that a considerably fine mesh is required to capture the localized phenomena. How-

## 1.1. BACKGROUND

---

ever, a naive implemented mesh generation algorithm usually produce s a uniform mesh where small elements are created even though they are only necessary in limited areas. Hence the adaptive finite element analysis was proposed to maximize the quality of the numerical solution for a given amount of the computational effort. In this method, tiny elements are generated at the required areas only and coarse mesh is expected for the rest. Since the stress or the displacement is unknown before the numerical method is conducted, an adaptive mesh can be difficult to be generated based on the geometry and boundary condition only. As a consequence, the concept of the “posteriori error estimator” was introduced to estimate the error contribution of each element based on the solution calculated from the FEM (Bauman et al., 2009; Bernardi et al.; Duval et al., 2018; Prudhomme et al., 2009).

In the existing computational methods, the geometry is interpolated by high order polynomials and the exact geometry is neglected due to its intricacy. However, the accuracy of adopting it into computational mechanics seems to be limited (Szabo et al., 2004). Tensor product Non-Uniform Rational B-spline (NURBS) is a well-known curve and surface representation method and has been adopted as a standard in computer graphics, computer-aided-design (CAD) (Nasri, 2003) and Initial Graphics Exchange Specification (IGES) since 1983 (IGES, 1986). Nowadays, the employment of rational polynomial functions in description of geometry in CAD/CAM applications is becoming more and more extensive (Piegl and

## 1.1. BACKGROUND

---

Tiller, 1987) and the CAD has been widespread and won even more popularity than the FEM does. While, the geometric descriptions adopted by engineers today for CAD and that for analysis are totally different. Furthermore, frequent design modifications in fast pace, modern society restricts the usage of analysis if a new mesh cannot be created in a short duration.

The creation of mesh in FEM can be expensive in terms of time and human resource. One possible solution is trying to replace the geometric modelling tool in FEM by something more CAD-like. NURBS, for example, is a standard mathematical model utilized in CAD industry. Exact CAD geometric boundary is achieved with the help of the NURBS curve and surface. This idea to be geometrically exact with a minimum discretization was adopted in the Isogeometric analysis developed in 2005 (Hughes et al., 2005) and further refined recently (Bazilevs et al., 2006a,b; Cottrell et al., 2006, 2009; Hughes et al., 2005; Zhang et al., 2007). Furthermore, a simplified mesh refinement method by omitting the necessity for communication with the CAD geometry once the initial shape was received is also targeted (Cottrell et al., 2007). It shows advantage in the structure analysis, fluid mechanics (Buffa et al., 2011) and dynamics (Cottrell et al., 2006).

A Scaled Boundary finite element method (SBFEM) which has similarity with both the FEM and the BEM is proposed to eradicate the necessity of fundamental solution in the BEM. SBFEM is a novel semi-

## **1.2. STATE OF THE PROBLEM**

---

analytical approach developed by Wolf and Song (Wolf and Song, 1999). As a method developed based on the FEM and the BEM, the SBFEM is a fundamental-solution-less boundary element method which keeps the benefits of the both as well as provides some effective solutions to the limitations to the FEM and the BEM (Wolf and Song, 1999). The fundamental solution is no longer required, spatial dimension is reduced by one as only the boundary is meshed with surface elements which leads to a decline in the number of unknowns and achievement of infinite boundary (Wolf, 2003).

## **1.2 State of the problem**

In current Isogeometric analysis, accuracy problems with numerical integration of a rational polynomials attract significant attention (Auricchio et al., 2012; Hughes et al., 2010; Sevilla and Fernández-Méndez, 2011). Furthermore, incapability to create a set of control points to fit an inhomogeneous essential boundary condition may lead to considerable errors and lower converge speed due to the non-interpolatory characteristics of the NURBS (Koo et al., 2013; Wang and Xuan, 2010; Wolfgang and Sven, 2011). Besides, although numerous amount of research has been conducted on improving the algorithm efficiency (Boor De, 1972; Choi et al., 1990; Grabowski and Li, 1992; Pan, 2001; Qin, 1996; Wang et al., 2012), most of the time is devoted to calculate the basis function which restricts the usage of high order basis function in 3D problems. Moreover, one of

## 1.2. STATE OF THE PROBLEM

---

the most critical problems in the existing Isogeometric analysis lies in the dimension incompatibility. It is based on FEM where 3D NURBS solids are required for meshing in 3D problems but only the boundary is described in CAD system. Further meshing process for converting input surface data to higher dimension physical geometry in isogeometric analysis has been referred to as “analysis-aware modelling” (Cohen et al., 2010). Considerable research on solving this incompatibility by domain parameterization has been performed using a variety of methods (Aigner et al., 2009; Martin et al., 2009; Qian and O., 2011; Yang and Qian, 2007).

The conventional FEM allows only hexahedron, tetrahedron, wedge and pyramid in 3D and triangular and quadrilateral elements in 2D which poses a heavy burden on mesh generation. In order to achieve a reasonably accurate result, the mesh of the traditional FEM is required to conform to the boundary of the problem domain. One rough estimate provided by Hughes et al. (2004) suggests that more than half of the overall analysis time is spent on meshing in the industries such as automotive and aerospace where complex shapes are involved. As a consequence, it could be necessary to develop an automatic mesh generation algorithm using limited types of shapes (Frey and George, 2007).

Numerous research has been conducted on the “posteriori error estimator” to achieve an adaptive analysis and it has been developed in the FEM (Ainsworth and Oden, 1993; Boroomand and Zienkiewicz, 1999;

### **1.3. OBJECTIVE AND SIGNIFICANCE**

---

Zienkiewicz and Zhu, b,c; Zienkiewicz et al., 1999), the BEM (Guiggiani, 1990; Kamiya and Kawaguchi, 1992; Zhao, 1998) and the SBFEM (Deeks and Wolf, 2002c). However, some of these error estimators require extra works such as stress recovery. Besides, it could be difficult to determine the most suitable error indicator to a given problem. Furthermore, the threshold is taken manually which limits the usage of several indicators as the number of the threshold grows quadratically as the number of the estimators increases.

## **1.3 Objective and significance**

As discussed in Sec. 1.2, there are several limitations associated with the existing numerical methods. This thesis aims at developing a complete and systematic numerical method where all procedures are conducted without human involvement for an arbitrary geometric input in both 2D and 3D situations. After the design files (IGES file and STL file in 3D) are delivered in electronic form, the proposed method shall be able to parse the geometric input, generate the mesh (quadtree in 2D and octree in 3D), determine the result using SBFEM and refine the mesh based on the error estimator automatically. The SBFEM is adopted as it requires only the boundary information and hence provides a seamless integration with the CAD modeling. The main objectives of this thesis are as follows

1. Minimize human efforts spent on structural analysis

#### **1.4. OUTLINE OF THE THESIS**

---

2. Be compatible with arbitrary geometric input in both 2D and 3D
3. Generate high quality mesh
4. Retain exact geometry
5. Develop a robust, extensible and flexible error estimator

Accomplishing tasks mentioned above makes significant contributions to solve the practical engineering problems automatically. Furthermore, a new error estimator trained using machine learning algorithm allows unlimited number of indicators to be used.

### **1.4 Outline of the thesis**

In the next chapter, a literature review on the existing numerical methods including the Isogeometric analysis and the SBFEM, the automatic mesh generation algorithm and the adaptive mesh refinement is presented. The advantages and disadvantages of different methods are critically discussed.

In Chapter 3, the idea of the isogeometric analysis is extended to the SBFEM to solve 2D problems including linear elasticity and linear elastic fracture mechanics. The NURBS basis functions are adopted as the shape functions in the SBFEM instead of the conventional Legendre polynomials. Some key formulations in linear elasticity and linear elastic



#### 1.4. OUTLINE OF THE THESIS

---

fracture mechanics in the SBFEM using the NURBS shape functions are derived. In order to perform a numerical integration on piecewise polynomials, a knot insertion is adopted to convert a piecewise polynomial to multiple non-piecewise polynomials. NURBS curve fitting is also introduced to enforce the stress boundary condition.

Chapter 4 implements an IGES adaptor which can convert the geometric information stored in IGES file into polylines. A quad-tree based mesh generation algorithm is developed to handle arbitrary geometric input from these polylines. In order to retain the exact geometry, the intersection point on the polyline is projected back onto the original NURBS curve. The projection is accelerated using the convex hull property and the quick hull algorithm is hence introduced. Hard point treatment is developed in order to handle the multiple material interfaces, sharp edges or cracks. Optimization algorithms such as bucket sort are adopted to improve the computational efficiency of the mesh generation. Stress analysis is conducted on 2D linear elasticity problems.

In Chapter 5, an adaptive mesh refinement algorithm is proposed by the help of machine learning. Expressions related to the eigenvalues of the SBFEM formulation together with some key geometric properties of the scaled boundary finite element are obtained as the error indicators. The models trained by the Multilayer Perceptron (MLP), the Support Vector Machine (SVM) using radial kernel function and the random forest are

#### 1.4. OUTLINE OF THE THESIS

---

compared and the MLP model which achieves the best performance is used. In order to improve the accuracy of the MLP, regularization methods including bagging and dropout are adopted. Due to the lack of eigenvalue error indicator in first order triangular elements, the method that eliminates these situations is produced. A matrix representation of NURBS curves is presented to achieve a higher efficiency and stability in calculating the intersections between a line and a NURBS curve. Stress analysis is conducted on 2D linear elasticity problems.

Chapter 6 references an octree based automatic mesh generation algorithm based on the STL file in 3D (Liu et al., 2017). In order to retain the exact geometry, a method that can project intersection points on the triangular surfaces back to their origin NURBS surfaces is developed. Another method that can calculate the intersection point between an edge in the scaled boundary finite element and the NURBS surface is also presented for the purpose of exact geometry. The matrix representation of the NURBS surface in 3D is introduced to improve the computational efficiency and stability of the calculation of the intersection. Splitting NURBS surfaces is adopted to accelerate the algorithm in both point projection and intersection calculation. The computational efficiency of them are further improved by utilizing the strong hull properties of the NURBS surface and hence the quick hull algorithm in 3D is introduced.

#### **1.4. OUTLINE OF THE THESIS**

---

Chapter 7 presents conclusions to the research. Possible future works are proposed.

# Chapter 2

## Literature review

### 2.1 Isogeometric analysis

In the traditional approach, geometric design and analysis are treated as separate modules requiring different methods and interpretations. For example, the geometric design module employed non-uniform rational B-splines (NURBS) introduced in Sec. 2.1.2 to describe the geometry, whilst the analysis module consisted of one of the following

1. Mesh based discrete models, such as the finite element method (FEM) (Hughes)
2. Boundary based methods, such as the boundary element method (BEM) (Sutradhar et al., 2008), scaled boundary finite element method (SBFEM) (Song and Wolf, 1997)

## 2.1. ISOGEOMETRIC ANALYSIS

---

### 3. Meshless methods (Liu, 2003)

The approximation space employed in the analysis module to describe both the geometry and the fields is different from that used in the CAD system. Hence it requires repetitive conversion between the CAD and the analysis and in this process errors are inevitable. Moreover, the analysis module employs polynomials that do not lead to exact representation of the geometry, whilst the geometric module employs Bzier representations that use Bernstein polynomials or B-splines and NURBS that employ de Boor polynomials (Piegl and Tiller, 1997). The above representations utilize basis functions and control points to represent the geometry, in addition to this, B-splines and NURBS also utilize a vector of knots. NURBS further use weights to control points to model intricate shapes.

As a consequence, the concept of isogeometric analysis is proposed (Hughes et al., 2004), in which the conventional Lagrange polynomials are replaced with the NURBS basis functions. The concept of isogeometric analysis (IGA) has revolutionized the analysis procedure. The IGA provides a natural link with the CAD model. A key feature of this framework is that the geometry is represented exactly by NURBS and the isoparametric concept is invoked to define the field variables. Since its inception, the method has been applied to a variety of problems such as plates and shells (Hosseini et al., 2014; Nguyen-Thanh et al., 2011a; Nguyen-Xuan et al., 2014), as cohesive elements (Nguyen et al., 2014), for shape optimization (Wall et al., 2008), fluidstructure interaction problems

## 2.1. ISOGEOMETRIC ANALYSIS

---

(Bazilevs et al., 2012), problems with strong discontinuities and singularities (Bazilevs et al., 2012; Jia et al., 2015; Oh et al.), optimization problems (Ghasemi et al., 2014) to name a few. Jia et al. (Jia et al., 2013) by incorporating reproducing kernel approximation methods, alleviated the instabilities of the conventional triangular B-spline element. The new approach yielded improved convergence rate and accuracy when compared to the conventional triangular B-spline element. This seems to be a promising alternative to NURBS and T-splines where considerable efforts are required for local refinements. In the conventional IGA, the surfaces/volumes are represented by the tensor product of the corresponding knot vectors. This requires the domain to be discretized with standard shapes and leads to a restricted number of boundary curves/surfaces. Also, this leads to excessive overhead of control points with refinement. This can be circumvented by adopting local refinement as proposed (Nguyen-Thanh et al., 2011b) or by employing T-splines (Sederberg et al., 2003). Recently, Simpson et al. (Simpson et al., 2012, 2013) proposed the isogeometric boundary element method (IGABEM), in which the NURBS functions were used to approximate the unknown fields. This framework circumvents the need to discretize the domain, as required by the IGAFEM. It was shown that the IGABEM is more accurate than the conventional BEM with polynomial interpolations. Furthermore, Scott et al. (Scott et al., 2013) and Simpson et al. (Simpson et al., 2014) combined the collocated IGABEM with T-splines for linear elastostatics and acoustic analysis, respectively. The concept of IGABEM was further extended to damage tolerance assessment (Peng, Xuan; Atroshchenko, Elena; Bordas, 2014) and shape sensitivity analysis

## **2.1. ISOGEOMETRIC ANALYSIS**

---

(Lian, Haojie;Simpson, Robert;Bordas, 2013).

### **2.1.1 Initial Graphics Exchange Specification (IGES) file**

The Initial Graphics Exchange Specification (IGES) is one of the standard file formats that allows the digital data exchange among the CAD industries, proposed by the United States Air Force (USAF) Integrated Computer Aided Manufacturing (ICAM) project (U.S. Product Data Association;, 2006). The initial propose of the ICAM was to significantly decrease the expenditure of data exchange in Aerospace manufacturing industry by developing procedures processes and CAD softwares. In order to minimize the gap between parts design and manufacturing in Aerospace industry, ICAM planed to improve the CAD software so that it can produce the numerical control programs automatically for the complex Computer Numerically Controlled (CNC) machine tools. One of the significant problem of it is that the data structure of the output of different CAD softwares can be different between each others. Later, this problem was solved by the introduction of the IGES.

The IGES had not gain popularity until 1988, when United State Department of Defense (USDoD) accepted the IGES file only for the contract of all weapon system s if the products are delivered in digital form. Consequently, any CAD software whose targeting customers want to sell their products to USDoD must support this file specification in terms of reading

## 2.1. ISOGEOMETRIC ANALYSIS

and writing.

A typical IGES file will be divided into 5 sections: start section, global section, directory entry section, parameter data section and terminate section as shown in Fig. 2.1. The mathematical tool that represents most of geometric shapes in the IGES file is the NURBS curve in 2D and the NURBS surface in 3D which will be introduced in Sec. 2.1.2

1	8 9	16 17	24 25	32 33	40 41	48 49	56 57	64 65	72	73	80
<b>Start Section</b> – a human readable prologue to the file.										S0000001 S0000002 S0000003 ⋮ S000000N	
It contains one or more lines ⋮ using ASCII characters in columns 1–72.											
<b>Global Section</b> – sending system and file information.										G0000001 G0000002 G0000003 ⋮ G000000N	
It contains the number of lines needed to hold the parameter fields, separated by ⋮ parameter delimiters, and terminated by one record delimiter, in columns 1–72.											
<b>Directory Entry Section</b> - contains one pair of lines for each entity Directory entry fields 1-9 in nine 8-column-wide fields Directory entry fields 10-18 in nine 8-column-wide fields										D0000001 D0000002	
<b>Parameter Data Section</b> – values and parameter delimiters terminated by one record delimiter, in columns 1-64; column 65 is unused									DE back Pointer	P0000001 P0000002	
S0000020 G0000003D0000500 P0000261				<b>Terminate Section</b> – record counts for preceding sections; columns 33–72 unused						T0000001	

**Figure 2.1:** File structure of an IGES file

### 2.1.2 Non-Uniform Rational B-Spline (NURBS)

In the proposed approach, the geometry and the unknown fields are represented by the NURBS. In this section, a brief overview is given. For



## 2.1. ISOGEOMETRIC ANALYSIS

---

more detailed description and implementation aspects, interested readers can refer to (Nguyen et al., 2015; Piegl and Tiller, 1997).

NURBS are the superset of B-spline functions. B-spline is short for basis spline and is a generalization of Bzier curves. A spline function is a piecewise polynomial function of degree  $p$  and the points of intersection of such functions are called knots. The number of knots must be equal to or greater than  $p + 1$ . One of the salient features of spline functions is that the functions are continuous at the knots, however, the continuity of the functions can be altered by repeating the knots. The B-spline functions are parametric functions of the form  $F(\eta)$  in which the parameter  $\eta$  lies in the parametric space. The key ingredients in the construction of B-spline functions are: the knot vector (a non decreasing sequence of parameter values,  $\eta_i \leq \eta_{i+1}$ ,  $i = 0, 1, \dots, m - 1$ ) and the degree of the curve  $p$ . The  $i$ th B-spline basis function of degree  $p$ , denoted by  $N_{i,p}$  is defined as (Piegl and Tiller, 1997):

$$N_{i,0}(\eta) = \begin{cases} 1 & \eta \in [\eta_i, \eta_{i+1}] \\ 0 & \eta \in (-\infty, \eta_i) \cup (\eta_{i+1}, \infty) \end{cases} \quad (2.1)$$
$$N_{i,p}(\eta) = \frac{\eta - \eta_i}{\eta_{i+p} - \eta_i} N_{i,p-1}(\eta) - \frac{\eta_{i+p+1} - \eta}{\eta_{i+p+1} - \eta_{i+1}} N_{i+1,p-1}(\eta)$$

## 2.1. ISOGEOMETRIC ANALYSIS

---

The first derivative of the B-spline basis function can be computed recursively from lower order basis functions as:

$$\frac{d}{d\eta} N_{i,p}(\eta) = \frac{p}{\eta_{i+p} - \eta_i} N_{i,p-1}(\eta) - \frac{p}{\eta_{i+p+1} - \eta_{i+1}} N_{i+1,p-1}(\eta) \quad (2.2)$$

The B-spline basis functions has the following properties:

1. Non-negativity
2. Partition of unity,  $\sum_i N_{i,p} = 1$
3. Interpolatory at the end points. The last point requires special treatment when imposing non-homogeneous Dirichlet boundary conditions (Nguyen et al., 2015).

Moreover, the spline function has limited support. Given  $n + 1$  control points  $(P_0, P_1, \dots, P_n)$  and a knot vector  $\Xi = \{\eta_0, \eta_1, \dots, \eta_m\}$ , the piecewise polynomial B-spline curve of degree  $p$  is defined as:

$$C(\eta) = \sum_{i=0}^n P_i N_{i,p}(\eta) \quad (2.3)$$

where  $P_i$  are the control points. A B-spline curve has the following information:  $n + 1$  control points,  $m + 1$  knots and a degree  $p$ . It is noted that  $n, m$  and  $p$  must satisfy  $m = n + p + 1$ . The B-spline functions also provide a variety of refinement algorithms, which are essential when employing B-spline functions to discretize the unknown fields. The analogous  $h$  and  $p$  refinement can be done by the process of ‘knot insertion’ and ‘order el-

## 2.1. ISOGEOMETRIC ANALYSIS

---

evation'. Another unique feature of the B-spline basis function is that, it is possible to combine the knot insertion and the degree elevation, commonly referred to as k-refinement in the literature (Hughes et al., 2005). Here we briefly discuss the knot insertion and the degree elevation. For more details, interested readers are referred to (Hughes et al., 2005; Piegl and Tiller, 1997) and references therein.

### 2.1.2.1 Knot insertion

Consider a B-spline basis functions defined on  $\Xi = \{\eta_0, \eta_1, \dots, \eta_m\}$ , let  $\bar{\eta} \in [\eta_k, \eta_{k+1})$ , and insert  $\bar{\eta}$  into  $\Xi$  to form a new knot vector  $\bar{\Xi} = \{\eta_0, \dots, \bar{\eta}_k = \eta_k, \bar{\eta}_{k+1} = \bar{\eta}, \bar{\eta}_{k+2} = \eta_{k+1}, \dots, \eta_{m+1} = \eta_m\}$ . Simultaneously, the size of the control points is increased by one. Thus  $C(\eta)$  has a representation on  $\bar{\Xi}$  of the form

$$\mathbf{C}(\eta) = \sum_{i=0}^{n+1} \bar{N}_{i,p}(\eta) \mathbf{Q}_i \quad (2.4)$$

Where  $\mathbf{Q}_i$  is:

$$\mathbf{Q}_i = \alpha_i \mathbf{P}_i + (1 - \alpha_i) \mathbf{P}_{i-1} \quad (2.5)$$

where

$$\alpha_i = \begin{cases} 1 & i \leq k - p \\ \frac{\bar{\eta} - \eta_i}{\eta_{i+p} - \eta_i} & k - p \leq i \leq k \\ 0 & i \geq k + 1 \end{cases} \quad (2.6)$$

## 2.1. ISOGEOMETRIC ANALYSIS

---

### 2.1.2.2 Order elevation

Let  $C(\eta) = \sum_{i=0}^n N_{i,p}(\eta) \mathbf{P}_i$  be a  $p$ th-degree B-spline curve on the knot vector  $\Xi$ . As a piecewise polynomial curve with  $p + 1$  order,  $C(\eta)$  is expected to be expressed in higher order basis functions. In other words, another set of control points  $\mathbf{Q}_i$  and knot vector  $\Xi$  should exist such that

$$C(\eta) = \sum_{i=0}^{\bar{n}} N_{i,p+1}(\eta) \mathbf{Q}_i \quad (2.7)$$

The procedure to elevate the order of a B-spline is listed as follows (Piegl and Tiller, 1997):

1. Extract each Bzier segment from the curve
2. Elevate the order of each Bzier segment
3. Remove unnecessary knots separating the  $(i - 1)$ th and  $i$ th segments

When elevating a  $p$ th Bzier curve, a new set of control points can be determined from:

$$\mathbf{Q}_i = (1 - \alpha_i) \mathbf{P}_i + \alpha_i \mathbf{P}_{i-1} \quad (2.8)$$

where  $\alpha_i = \frac{i}{p+i}$ ,  $i = 0, \dots, p + 1$ . Fig. 2.2 and Fig. 2.3 show an example of basis function when performing a knot insertion and order elevation, respectively.

Besides, it is found that if the order is elevated to  $q$  and only then inserted a unique knot value, the basis would have  $q - 1$  continuous derivatives at the knot we inserted and this process is called  $k$ -refinement (Hughes

## 2.1. ISOGEOMETRIC ANALYSIS

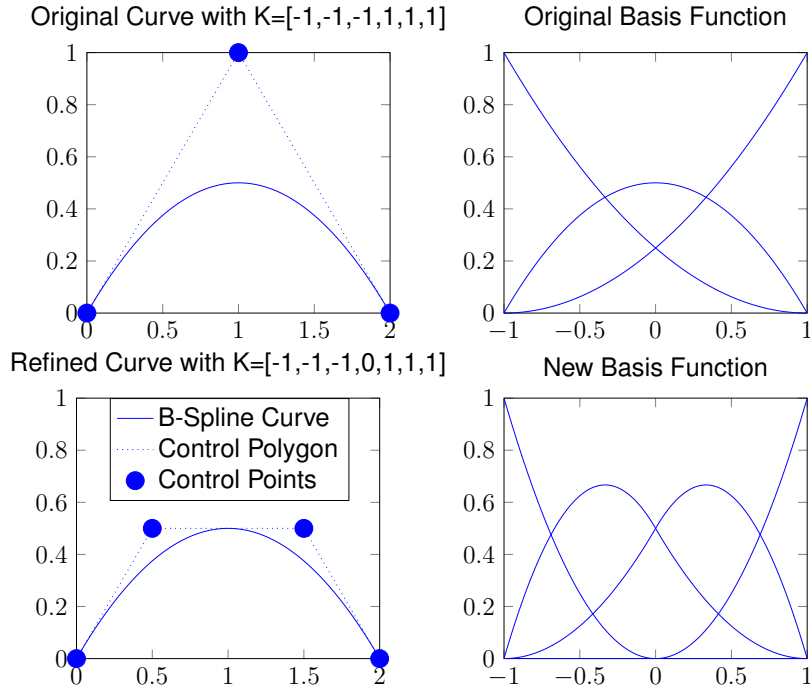
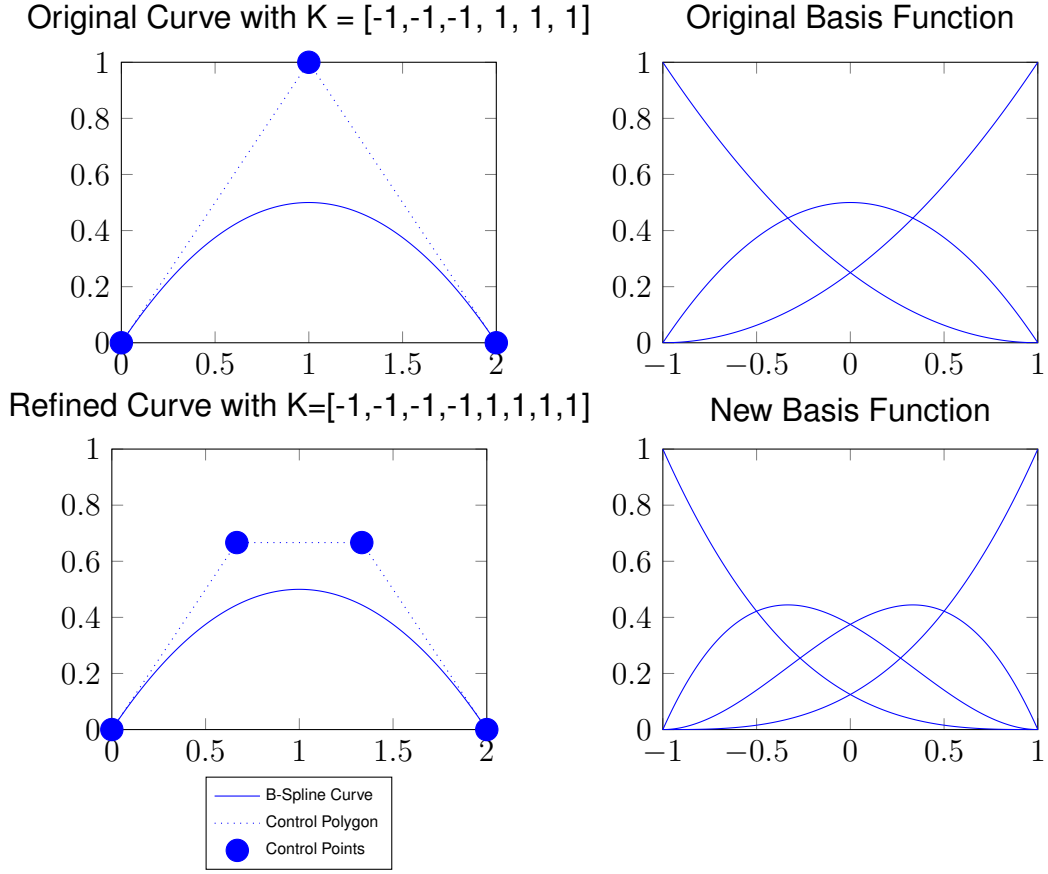


Figure 2.2: B-spline functions: knot insertion

et al., 2005). Despite the flexibility offered by the B-splines, they lack the ability to exactly represent some shapes such as circle and ellipsoids. To improve this, non-uniform rational B-splines (NURBS) are formed through rational functions of B-splines. The NURBS thus form the superset of B-splines. The key ingredients in the construction of NURBS basis functions are: the knot vector (a non decreasing sequence of parameter values,  $\eta_i \leq \eta_{i+1}$ ,  $i = 0, 1, \dots, m - 1$ ), the degree of the curve  $p$  and the weight associated to a control point,  $w$ . A  $p$ th degree NURBS basis function is defined as follows:

$$R(\eta) = \frac{N_{i,p}(\eta)w_i}{W(\eta)} = \frac{N_{i,p}(\eta)w_i}{\sum_{i=0}^n N_{i,p}(\eta)w_i} \quad (2.9)$$

## 2.1. ISOGEOMETRIC ANALYSIS



**Figure 2.3:** B-spline functions: order elevation

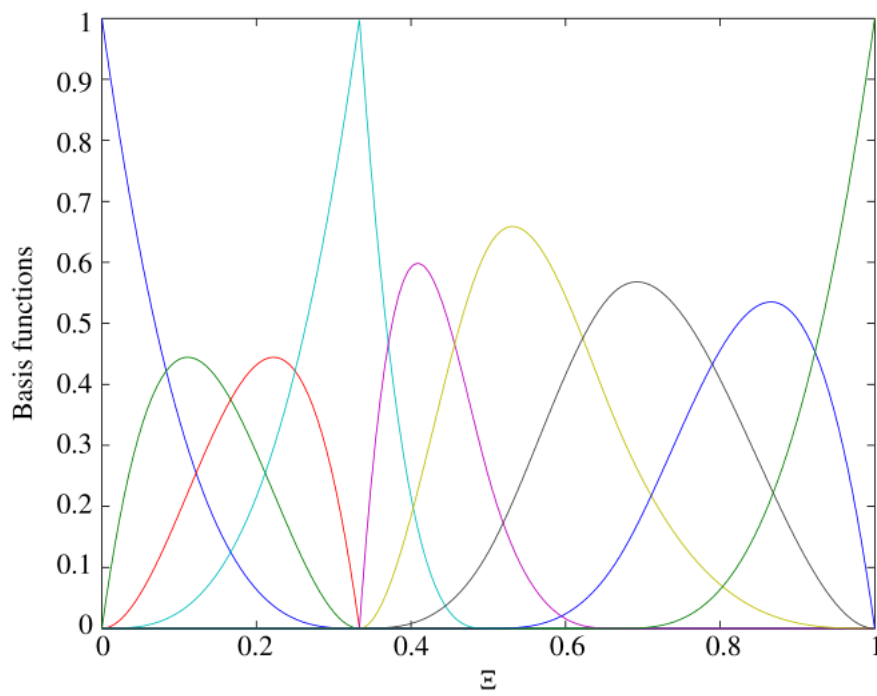
where  $w_i$  are the weights for the  $i$ th basis function  $N_{i,p}(\eta)$ . Fig. 2.4 shows the third order NURBS for an open knot vector.

The first derivative of a NURBS basis function is computed using the quotient rule and is given by:

$$\frac{d}{d\eta} R_{i,p}(\eta) = w_i \frac{N'_{i,p}(\eta)W(\eta) - N_{i,p}(\eta)W'(\eta)}{W(\eta)^2} \quad (2.10)$$

## 2.1. ISOGEOMETRIC ANALYSIS

---



**Figure 2.4:** 3rd NURBS basis function for an open knot vector  $\Xi = \{-1, -1, -1, -1, -1/3, -1/3, -1/3, 0, 1/3, 1, 1, 1, 1\}$ . Note that the functions are only interpolatory at the end points

## 2.2. SCALED BOUNDARY FINITE ELEMENT METHOD

---

### 2.2 Scaled boundary finite element method

The SBFEM developed by Song and Wolf (P. WOLF and Song, 1996) provides a promising semi-analytical method to analyze problem in fracture mechanics and unbounded domain. As a method developed based on the FEM and the BEM, the SBFEM is a fundamental-solution-less boundary element method which keeps the benefits of the both as well as provides some effective solutions to the limitations to the FEM and the BEM (Wolf and Song, 1999). In contrast to the FEM, only the boundary is discretized using the conventional FEM interpolating function which leads to a decline in the number of unknowns. It also allows solving the problem involving bimaterial interfaces and crack faces without the discretization of them. Compared to the BEM, the fundamental solution is no longer required. The infinite boundary can be achieved naturally as the radiation condition at infinity is satisfied in the SBFEM (Wolf, 2003)..

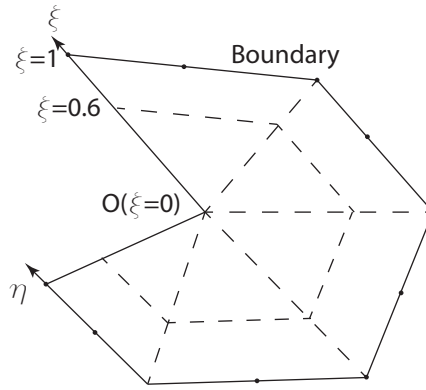
Fig. 2.5 illustrates a basic concept of the SBFEM. A scaling center  $O$  is selected at a point from which the whole boundary of the domain is visible (scaling requirement). This condition is automatically satisfied for all convex polygons and many concave polygons. The scaling requirement is equivalent to the notion of “star convexity” (Bishop, 2014). For the domain that does not meet the scaling requirement, the requirement can always be satisfied by sub-structuring, i.e. dividing the structure into smaller subdomains, for example, scaled boundary polygon formulation (Natarajan et al.,



## 2.2. SCALED BOUNDARY FINITE ELEMENT METHOD

---

2014). The problem domain can be covered by scaling the boundary in the radial direction respect to the scaling center with a ratio  $\xi \in [0, 1]$ . In the domain, only the boundary at  $\xi = 1$  is discretized with the conventional shape functions.



**Figure 2.5:** Two dimensional scaled boundary coordinates, where  $O$  is the scaling center and  $\xi$  is the radial coordinate with  $\xi = 0$  at the scaling center and  $\xi = 1$  on the boundary.

The method proved to be far more versatile and was applied to static problems in bounded domains and extended to take into account prescribed displacements (Deeks, 2004) and concentrated loads (Vu and Deeks, 2014). A simple derivation of the necessary equations based on the virtual work principle is also presented (Deeks and Wolf, 2002a). This spurred the interest among researchers, as the similarity with the virtual work-based FEM derivation was highlighted. The method is further developed by deriving a stress recovery technique that was later adopted in adaptive refinement techniques, such as h- and p-adaptive SBFEM (Deeks and Wolf, 2002c; Vu and Deeks, 2008; Yang et al., 2011). The wave in-

## 2.2. SCALED BOUNDARY FINITE ELEMENT METHOD

---

interaction with a cylindrical structure is investigated (Tao et al., 2007) and the method is also extended to structural dynamics (Song, 2009) where the dynamic stiffness matrix was obtained as a continued fraction solution. The main advantage of this approach is that the inertial effect at high frequencies can be modeled by high-order terms of the continued fraction without introducing an internal mesh. A higher order spectral element was used in computation of the dispersion curves of the elastic wave using the SBFEM (Gravenkamp et al., 2014) and a superior accuracy compared to the conventional approaches is shown.

The conventional FEM is known to be inefficient to deal with internal discontinuities such as material interfaces or singularities. In an effort to overcome the limitations of the FEM, mesh-free methods and enrichment techniques such as the extended finite element method (XFEM) were introduced. Treatment of evolving discontinuities in mesh-free methods (Rabczuk and Belytschko; Rabczuk and Zi, 2007) and enrichment techniques (Areias et al., 2013; Babuška et al., 2003; Bordas et al.; Chau-Dinh et al., 2012) is more straightforward because it does not require conforming mesh or frequent mesh adaptation as the discontinuities evolve. On the other front, by exploiting the unique feature of the scaling center, the method allows the computation of stress intensity factors directly from their definitions (Deeks and Chidgzy, 2005; Song and Wolf, 2002). This has emerged to be an attractive alternate to the already established methods such as the XFEM and the meshless methods to model crack propagation.

## 2.2. SCALED BOUNDARY FINITE ELEMENT METHOD

---

Chidgze et al. (Bird et al., 2010; Chidgze et al., 2008) coupled the SBFEM with the BEM for computations in fracture mechanics. This framework combines the semi-analytical solution accuracy of the SBFEM with the geometric flexibility provided by the BEM. Natarajan and Song (Natarajan and Song) combined the extended FEM and the SBFEM, thus, circumventing the need to know a priori the enrichment functions, required by the former. Recently, Ooi et al. (Ooi et al.) and Natarajan et al. (Natarajan et al., 2014) employed scaled boundary formulation in polygonal elements to study crack propagation and compared the performance with other displacement based formulations, respectively. Li et al. (Li et al., 2013) applied SBFEM to analyze two-dimensional fracture problems in piezoelectric materials. Ooi et al. (Ooi and Yang, 2010; Ooi et al., 2013; Ooi et al.) developed an efficient methodology for automatic crack propagation simulation using the SBFEM.

It can be seen that, since the inception of the method, the SBFEM has been applied to various problems in engineering and science. It should be noted that most of the above studies employed Lagrange interpolants to approximate the unknown fields in the circumferential direction. He et al. (He et al., 2012, 2014) employed moving least square (MLS) shape functions and Fourier series expansion to approximate the displacement fields in the circumferential direction. It should be noted that the MLS and Fourier basis functions do not satisfy Kronecker property and that special care must be employed to enforce the boundary conditions. It was shown

### 2.3. ADAPTIVE MESH REFINEMENT

---

that the SBFEM with MLS and Fourier shape functions yielded more accurate results than the MLS. Very recently, Lin et al. (Lin et al., 2014) employed non-uniform rational B-splines to approximate the unknown field in the circumferential direction. However, their study was limited to 2D elastostatics.

## 2.3 Adaptive mesh refinement

In some situations, the FEM mesh can be so locally coarse that some localized phenomena can not be captured. However, a naive implemented mesh generation algorithm usually produces a uniform mesh where small elements are created even though they are only necessary in limited areas. X-FEM (Moës et al., 1999) or the Generalized FEM (Kim and Duarte; Strouboulis et al., 2001) was proposed to enrich the model when the mesh is so coarse that the local scale phenomena (crack for example) can not be taken into account. Others developed the multigrid algorithms which permits relevant computations while keeping the computational cost acceptable to solve this problem (Passieux et al.; Rannou et al.). However, only ad-hoc softwares support enriched finite element model or multigrid (Duval et al., 2018), which leads to the fact that these methods may not be applicable to all circumstances, especially for the users of the softwares that lack of such features.

## **2.3. ADAPTIVE MESH REFINEMENT**

---

As a consequence, methods using a posteriori error estimator to refine the mesh adaptively was proposed and widely adopted in FEM (Ainsworth and Oden, 1993; Babuška and Rheinboldt; Bauman et al., 2009; Bernardi et al.; Boroomand and Zienkiewicz, 1999; Craig; Duval et al., 2018; Oden and Demkowicz, 1989; Prudhomme et al., 2009; Zienkiewicz and Zhu, a,b,c; Zienkiewicz et al., 1999) and BEM (Guiggiani, 1990; Kamiya and Kawaguchi, 1992; Kita and Kamiya, 1994; Kita et al., 2000; Zhao, 1998; Zhao and Wang, 1999). A posteriori error estimator using stress recovery technique for the SBFEM was also proposed (Deeks and Wolf, 2002c). However, some of these error estimators require extra work such as stress recovery. Besides, it could be difficult to determine the most suitable error indicator to a given problem. Machine learning and deep neural network introduced in Sec. 2.3.1 allow the usage of multiple error indicators was proposed (Saeed Iqbal; Graham .F. Carey, 2005). However, the fact that only the geometric properties were considered and the lack of physical indicators limit the effectiveness of this method.

### **2.3.1 Machine learning**

A machine learning algorithm is such an algorithm that is able to learn from data and find regularities. The “learning” is defined as “A computer program is said to learn from Experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.” (Mitchell, 1997)

### 2.3. ADAPTIVE MESH REFINEMENT

---

The tasks  $T$  refer to the task that are too hard to solve with fixed programs. For example, if a robot is designed to be able to walk, it can be done by a program that help the robot to learn how to walk or by a program written manually to tell the robot how to walk. The second option is rarely chosen because a fixed program can hardly adapt the complex situation in the reality. The tasks usually include classification and regression problems in numerical calculation.

As a way to measure the abilities of a machine learning program, quantitative measurement of its performance must be designed. Accuracy could be one of the most popular performance measurement in classification task. It is simply defined as the rate of examples for which the algorithms gives the same classification as the reality. In real world problem, the ability that a model performs on data that it has not seen before could be more important as it is related to its performance when used in real world problems. Consequently, these performance measures are usually evaluated using a subset of the original data called test set. While another subset of the original data called training set is adopted to train the machine learning system.

A learning algorithm normally is either unsupervised learning or supervised learning. It is dependent on what kind of experience it has during the learning process. Unsupervised learning algorithms experience a dataset that have a series of features. It is expected to learn useful prop-

### 2.3. ADAPTIVE MESH REFINEMENT

---

erties of this dataset. While supervised learning algorithms experience a dataset that not only have many features, but also have a label or target. For example, in Iris Fisher data set (Fisher, 1936), fish species of each iris plant will be included as well as features of the fish.

The machine learning algorithms targeting classification problems used in the proposed method will be introduced in Sec. 2.3.2.

#### 2.3.2 Multilayer perceptron(MLP)

There are plenty of machine learning algorithms such as Support Vector Machine (SVM) (Boser et al., 1996; Cortes and Vapnik, 1995), decision tree (Olshen, 1984), random forest (Ho, 1995), etc. that can be adopted in all kinds of situation. Multilayer Perceptron (MLP), also known as feedforward neural networks or deep feedforward networks will be introduced in detail in this section.

Fig. 2.6 illustrates a typical multilayer perceptron. The reason why it is called feedforward is because that all calculation are conducted all the way from the inputs  $x$  to the outputs  $y$ , passing the intermediate computations. There is no feedback from a deeper layer to a shallower layer otherwise it becomes a recurrent neural network. It is called neural networks is because that they are usually defined by a composition of several distinct functions. These functions can be described as an acyclic graph.

### 2.3. ADAPTIVE MESH REFINEMENT

---

Take a function  $f(x) = f^{(3)}(f^{(2)}(f^{(1)}(x)))$  as an example, it describes a MLP with three layers. In this example,  $x$  will be the input and the function  $f^{(1)}$  is the first layer,  $f^{(2)}$  being the second and the  $f^{(3)}$  is called the output layer. Number of the functions or the length of the chains is called the depth of the neural network. When a neural network with large number of layers, it is consider as a deep neural network and it is where the term “deep learning” comes from. The training purpose of a MLP is to predict the output  $y_p$  that is close to the accurate value  $y_a$  from any given input  $x_p$ . A training set with both inputs  $x_t$  and expected outputs  $y_t$  specify directly the behavior of the output layer upon different inputs. There is no direct relationship between what other layers do and the training data. A suitable learning algorithm is expected to be able to train these intermediate layers to response properly so that the prediction from the output layers is accurate enough. These intermediate layers are called hidden layers as they are not directly related to the desired outputs.

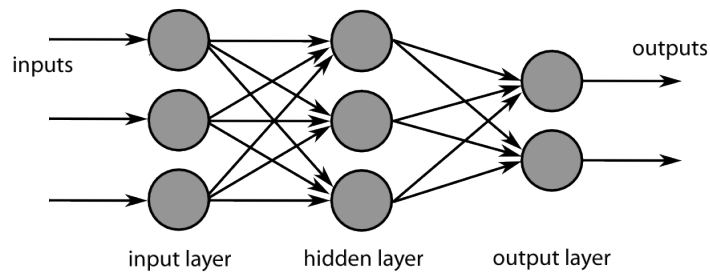
Furthermore, the word neural illustrates that the idea of the topological model comes from neuroscience. In the mathematical model of the neural network, each unit in a layer acts independently like a neuron in human being’s brain. Even though a layer of these ‘neurons’ forms a vector, the model can be more suitably described by each layer contains some neuron that behave analogously (dot production between two vectors) instead of a layer to layer relationship (matrix production between a vector and a matrix). The choice of selecting more than one layers of vector and



### 2.3. ADAPTIVE MESH REFINEMENT

---

each neuron takes all values from the previous layer and then computed its own value to describe the model comes from the neuroscience while the functions used in the model are not guided by the neuroscience but by mathematical and engineering technologies. A neural network so far will never aim to simulate a brain perfectly as well, it is designed to achieve statistical generalization instead.



**Figure 2.6:** A typical topology of a multilayer perceptron

The key question of a MLP is how to find a proper non-linear mapping function  $\phi_i$  that can calculate all units in  $i + 1$ -th layer from the previous one. One of the options is to adopt a generic function, such as the Radial Basis Function(RBF)(Chang et al., 2010) kernel. Even though an RBF kernel is capable to fit the training data with infinite-dimensional function, the performance on the test set usually not as satisfactory as it is on the training set. Some advanced example can not be solved as not enough prior information is encoded and generic feature mappings drawn from the local smoothness is used only. Another widely-accepted approach before the concept of deep learning being popular is to set the function manually. Each used function for separated task typically spends engineers decades of time and it is very unlikely that these functions can be used

## 2.3. ADAPTIVE MESH REFINEMENT

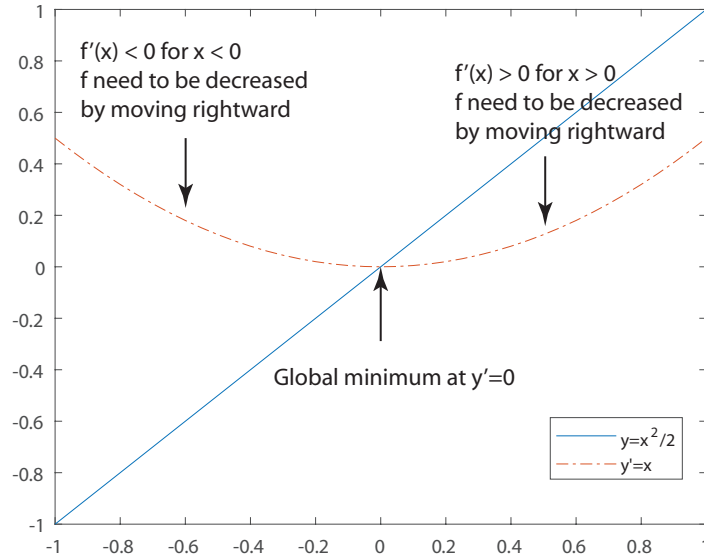
---

in other fields. The strategy of deep learning is to find the function by itself based on the given training set. In this approach, we have a model  $y = f(x; \theta, \omega) = \phi(x; \theta)^T \omega$ . Where  $\theta$  stands for parameters used to learn function  $\phi$  from a large variety of functions. Parameters  $\omega$  then map the intermediate result  $\phi(x)$  to the desired output. Drawback of this method is that the convexity will not be guaranteed during the searching of the global minimum which means finding a local minimum based on gradient no longer gives a global minimum. However, the merits outweigh the shortcomings since the generalization is as high as using an RBF kernel and no human efforts is required.

### 2.3.3 Gradient based optimization

The learning problem of the MLP can be generalized as a optimization problem in mathematics. In other words, the learning algorithm is to find the global minimum of a specific value on a high dimensional surface. The function need to be minimized is called the criterion and it is called cost function or loss function when it is being minimized. Fig. 2.7 illustrates how the derivatives are used to perform a gradient descent algorithm on a parabola. In a two dimensional curve which can be expressed by function  $y = f(x)$  with real numbers  $x$  and  $y$ . The derivative  $f' = \frac{dy}{dx}$  is ultra useful to determine the global minimal value of a function as it stands for the slope and the slope tells whether  $y$  is becoming smaller or larger when  $x$  increase. The global minimal is proven to happen when the derivative

## 2.3. ADAPTIVE MESH REFINEMENT



**Figure 2.7:** An example of gradient descent algorithm on a parabola

equal to zeros or on the boundary if the function is continuous. While a point with zero slope does not necessary stand for a global minimal as it can be a local one or a saddle point.

In training problems of the MLP, chances are that we are end up with a multidimensional function that is not convex which creates extra challenge to determine the global minimal from local ones and saddle points surrounded by very flat regions. As a consequence, in practice, the algorithm will be terminated when a small enough local minimal is found instead of looking for the global one. In the situation where multiple inputs are involved, the concept of partial derivatives  $\frac{\partial}{\partial x_i} f(x)$  is used to demonstrate the change in  $y$  corresponding to the change in the  $i$ -th input  $x_i$ . The gradient  $\nabla_x f(x)$  contains the partial derivatives in all directions and hence

### 2.3. ADAPTIVE MESH REFINEMENT

---

is used in solving the learning problem in MLP. In order to minimize loss function  $f$ , it is expected to determine the direction where  $f$  decreases the fastest and it can be expressed in mathematical as <sup>1</sup>

$$\begin{aligned}
 & \arg \min_{\mathbf{u}} \mathbf{u}^T \nabla_x f(x) \\
 &= \arg \min_{\mathbf{u}} \|\mathbf{u}\|_2 \|\nabla_x f(x)\|_2 \cos \theta \\
 &= \arg \min_{\mathbf{u}} \cos \theta
 \end{aligned} \tag{2.11}$$

where  $\mathbf{u}$  is a directional unit vector ( $\mathbf{u}^T \mathbf{u} = 1$ ) and  $\theta$  is the angle between  $\mathbf{u}$  and the gradient. It can be concluded that  $f$  can be decreased in the fastest way in the direction of the reverse of the gradient which is called gradient descent in Eq. 2.12.

$$x' = x - \epsilon \nabla_x f(x) \tag{2.12}$$

where  $\epsilon > 0$  stands for the learning rate which decides the distance for each step. Several strategies exist to determine learning rate including using a tiny constant, adopting a large number at the beginning and decreasing it during the iteration and trying different learning rates and finding the best (line search).

---

<sup>1</sup>  $\arg \min_x f(x) = \{x | x \in S \wedge \forall y \in S : f(y) \geq f(x)\}$

## 2.3. ADAPTIVE MESH REFINEMENT

---

### 2.3.4 Stochastic gradient descent (SGD)

SGD is an extension of the gradient descent method described above in order to significantly decrease the computational time without lose the accuracy. One commonly accepted way to improve the effectiveness of a MLP model is to increase the training set. However, an increasingly large training set requires higher computational cost. The cost function of a neural network can be written as followed when negative log-likelihood is used

$$\mathbf{J}(\theta) = E_{\mathbf{x}, y \sim \hat{p}_{data}} \mathbf{L}(\mathbf{x}, y, \theta) = \frac{1}{m} \sum_{i=1}^m L(\mathbf{x}^i, y^i, \theta) \quad (2.13)$$

where  $L$  is the per-example loss  $\mathbf{L}(\mathbf{x}, y, \theta) = -\log p(y|\mathbf{x}, \theta)$ . And the calculation of the gradient descent becomes:

$$\nabla_{\theta} \mathbf{J}(\theta) = \frac{1}{m} \sum_{i=1}^m \nabla_{\theta} \mathbf{L}(\mathbf{x}, y, \theta) \quad (2.14)$$

It is clear that the computation of the gradient is an  $O(m)$  operation where  $m$  is the size of the training data. Considering the fact that it takes at least thousands iteration before a model converges and each iteration requires an operation that takes prohibitively long time, some optimization must be taken. The main idea of the SGD is to treat the gradient as an expectation. Based on this assumption, the gradient can be calculated by some sampled data from the training set called minibatch  $\mathbf{B} = \{x^1, x^2, \dots, x^{m'}\}$  drawn from the original set. Size of the minibatch  $m'$  is usually taken as a constant from one to a hundred and is irrelevant to the size of the training set unless its size is extremely small (i.e. smaller than 100). The gradient

### 2.3. ADAPTIVE MESH REFINEMENT

---

then can be calculated based on minibatch with  $O(1)$  operation as

$$g = \frac{1}{m'} \sum_{i=1}^{m'} \mathbf{L}(\mathbf{x}, y, \theta) \quad (2.15)$$

The most outstanding feature of the MLP compared to linear models is that it can capture non-linear features automatically which results in a non-convex loss function. As a consequence, learning algorithms in the MLP adopt the SGD to solve the optimization problem iteratively, instead of finding it directly with a closed form mathematical solution or by convex optimization. This usually leads to a cost function with very low value rather than the global minimum. Furthermore, without a convergence guarantee as non-convex loss function is treated and SGD is adopted, the selection of the initial value may have significant influence on the final result. Hence, it is recommended to initialize all weights and bias to small values.

#### 2.3.5 Cost function

Choosing a suitable cost function can be important to train a MLP. In most of the cases, maximum likelihood or negative log-likelihood performs reasonably satisfactory as the cross-entropy between the training data and the labels remains as for linear models. It can be expressed as:

$$\mathbf{J}(\theta) = -\mathbf{E}_{x,y \sim \hat{p}_{data}} \log p_{model}(y|x) \quad (2.16)$$

## 2.3. ADAPTIVE MESH REFINEMENT

---

One of the outstanding merits to adopt this method is that the design of the cost function for other models is no longer necessary. Setting parameter for a model  $p(x|y)$  and the cost function  $\log p(y|x)$  can be determined automatically. Another advantage using maximum likelihood function as cost function is because that it helps to prevent gradient vanishing. The gradient descent plays an important role in the learning algorithm and the method becomes inefficiency or even fails when the cost function becomes extremely flat (very small gradient). In negative log-likelihood cost function, this kind of circumstance can be prevented because a logarithmic function saturates when the argument is extremely large.

### 2.3.6 Output units

The selection of the output units is highly related to the choice of the loss function. In most of the case the cross-entropy between the distribution of the data and the model is used. In other words, the form of the cross-entropy function decides the presentation of the output. Although all output units can also act as hidden units, the major difference is that the output units must produce the result as expected.

In tasks where binary classification is expected, the sigmoid unit is usually adopted. The probability distribution of a binary classification is a Bernoulli distribution and the maximum-likelihood method is to define this distribution over  $y$  conditioned on  $x$ . The output of the neural net is to

### 2.3. ADAPTIVE MESH REFINEMENT

---

predict the probability  $P(y = 1|x)$  only as it is a binary classification. By enforcing a constrain of  $[0, 1]$  on the probability, it becomes

$$P(y = 1|x) = \max\{0, \min\{1, w^T h + b\}\} \quad (2.17)$$

assuming linear unit is adopted. Even though a valid probability is defined, there will be some issue s during the training as the gradient becomes zero when  $w^T h + b > 1$ . In gradient based learning algorithm, a zero gradient always cause problem because the algorithm may have very little information on how to improve the parameters. In order to guarantee a non-zero gradient, a sigmoid output (Eq. 2.18) unit can be taken.

$$\hat{y} = \sigma(w^T h + b) \quad (2.18)$$

where  $\sigma$  is the logistic sigmoid function

$$\sigma = \frac{1}{1 + e^{-x}} \quad (2.19)$$

A sigmoid unit can be regarded as a combination of linear unit and a sigmoid activation function that convert s the output from the linear component  $z$  into a probability. The definition of the probability distribution over  $y$  using the value  $z$  will be discussed. The sigmoid can be motivated by building an unnormalized probability distribution  $\tilde{P}(y)$ , which does not sum to 1. A valid probability distribution then can be determined by dividing by a specific constant. The unnormalized probabilities can be determined if the unnormalized log probabilities are assumed to be linear in  $y$  and  $z$  at the



### 2.3. ADAPTIVE MESH REFINEMENT

---

beginning. It will be normalized to yield a Bernoulli distribution controlled by a sigmoidal transformation of  $z$ :

$$\begin{aligned} \log \tilde{P}(y) &= yz \\ \tilde{P}(y) &= e^{yz} \\ P(y) &= \frac{e^{yz}}{\sum_{y'=0}^1 e^{y'z}} \\ P(y) &= \sigma((2y - 1)z) \end{aligned} \tag{2.20}$$

Variable  $z$  that defines a distribution with normalization and exponentiation over binary variables is called logit

$$\begin{aligned} J(\theta) &= -\log P(y|x) \\ &= -\log \sigma((2y - 1)z) \\ &= \zeta((1 - 2y)z) \end{aligned} \tag{2.21}$$

where  $\zeta(x)$  is the softplus function

$$\zeta(x) = \log(1 + e^x) \tag{2.22}$$

By rewriting the cost function in terms of softplus, it can be seen that saturation happens only when  $(1 - 2y)z$  approaches negative infinity. In other words, it happens when  $y = 1$  and  $z$  approaches positive infinity which means the prediction is correct, or when  $y = 0$  and  $z$  approaches negative infinity which means the prediction is extremely wrong. In the later case,

## 2.3. ADAPTIVE MESH REFINEMENT

---

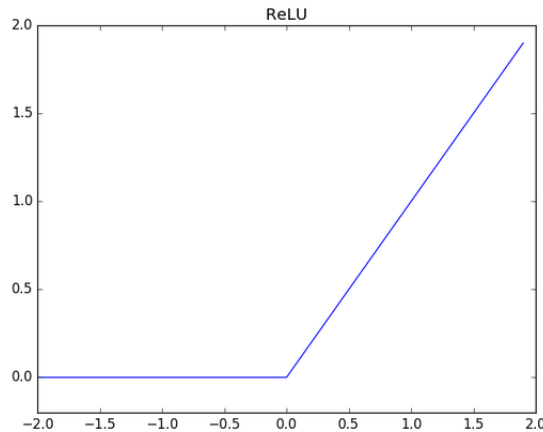
the softplus function can be simplified as

$$\zeta((1 - 2y)z) = \zeta(|z|) \quad (2.23)$$

And its derivative becomes  $\text{sign}(z)$ , which means the softplus function will not have gradient vanishing problem in the later case. It helps the gradient descent learning algorithms to function with stability.

### 2.3.7 Hidden units

The choice of the hidden units lack of definitive principles and is still an active area of research. Nevertheless, the rectified linear unit (RELU) usually acts as the default option. RELU uses an activation function that maps  $g(z)$  to  $\max\{0, z\}$  as shown in Fig. 2.8. Although it is not differentiable at  $z = 0$ ,



**Figure 2.8:** Rectified linear units

the gradient descent still acts as expected in practice. This is because a numerical method rarely finds the local minimal, but ends up with a point

### 2.3. ADAPTIVE MESH REFINEMENT

---

that is close enough to the local minimal. Based on the assumption that a strict zero gradient is not expected, undefined derivatives can be allowed on hidden units.

One of the most outstanding advantage of the RELU is that it is extremely easy to optimize. It is a linear unit expected for the fact that half of its domain yields zero. As a consequence, the derivative of an RELU is significant if the unit is active. RELU is applied on top of the existing affine transformation:

$$\mathbf{h} = g(\mathbf{W}^T \mathbf{x} + \mathbf{b}) \quad (2.24)$$

Initial values for the transformation matrix are suggested to set with a small positive value as they can make most of the RELU active and let the derivatives pass through. Because the output remains constant when it is inactive, the unit can not learn anything when it is not active. Improvements have been proposed recently. Absolute value rectification that maps  $g(z)$  to  $|z|$  has been used in object recognition (Jarrett et al., 2009). Leaky RELU that gives a small positive value for the derivatives (Maas et al.) while parametric RELU regards this derivatives as a learnable parameter (He et al., 2015). Maxout unit (Goodfellow et al., 2013) is a further generalization of the RELU. The maxout unit divides  $z$  into groups of  $k$  values instead of applying an element-wise function  $g(z)$ . The maximum value of the output of all units will be used to represent this group:

$$g(z)_i = \max_{j \in G(i)} z_j \quad (2.25)$$

## 2.3. ADAPTIVE MESH REFINEMENT

---

Table 2.1: Confusion matrix

		Predicted	
		Refined	Not refined
Actual	Refined	True Positive(TP)	False Negative(FN)
	Not refined	False Positive(FP)	True Negative(TN)

where  $G^{(i)}$  is the set of indices into the input for group  $i$ ,  $\{(i-1)k+1, \dots, ik\}$ . This provides a way of learning a piecewise linear function that responds to multiple directions in the input  $x$  space. It can be regarded as learning the activation function as it actually learn a piecewise linear function. High fidelity and any convex function can be achieved by maxout unit if a large  $k$  is used. In practice, maxout unit with  $k = 2$  usually has the same performance compared to the traditional RELU, parametric RELU and Leaky RELU. Since every maxout unit needs to learn its own parameter, the size of the training set must be large enough to support the learning algorithms, or keep the  $k$  value low (Cai et al., 2013).

### 2.3.8 Performance indicators

#### 2.3.8.1 Confusion matrix

A confusion matrix is a matrix used to illustrate the outcome of a classification model on a set of test data whose actual results are known.

## 2.3. ADAPTIVE MESH REFINEMENT

---

### 2.3.8.2 Accuracy

Accuracy may be the most intuitive indicator. It is simply defined as  $\frac{TP+TN}{TP+TN+FP+FN}$ . The importance of the accuracy is dependent on the prior since the model can be highly influenced by the prior probability distribution. For example, a spam detection model is trained from a data set which contains only 10% of the spam e-mails. As a consequence, if the model is extremely conservative and classify almost all incoming e-mails as non-spam, it can easily achieve an accuracy of more than 90% in cross validation which is higher than lots of spam detectors.

### 2.3.8.3 Precision(P)

$$P = TP/(TP + FP)$$

Precision describes the chance that the model gives ‘true’ and it is actually ‘true’. In the spam detector example, a high precision can be expected as a conservative model tends to give non-spam unless it has strong confidence.

### 2.3.8.4 Recall rate(R)

$$R = TP/(TP + FN)$$

## 2.3. ADAPTIVE MESH REFINEMENT

---

Recall rate describes the ratio the model gives 'true' to the total number of 'true's. In the spam detector example, a low recall rate is expected.

### 2.3.8.5 F1 score

$$F1 = 2 * P * R / (P + R)$$

F1 score is an indicator that considers both the precision rate and the recall rate. It is defined as the harmonic mean of them.

### 2.3.8.6 Receiver operating characteristic(ROC)

ROC is a True Positive Rate(TPR) vs False Positive Rate(FPR) curve (Fig. 2.9) where  $TPR = R$  and  $FPR = FP / (FP + TN)$

### 2.3.8.7 Area under the curve(AUC)

As can be seen from the ROC, the larger the area under the curve, the better the classifier is. Consequently, area under the curve (AUC) becomes another important indicator in machine learning.

## 2.4. AUTOMATIC MESH GENERATION IN 3D

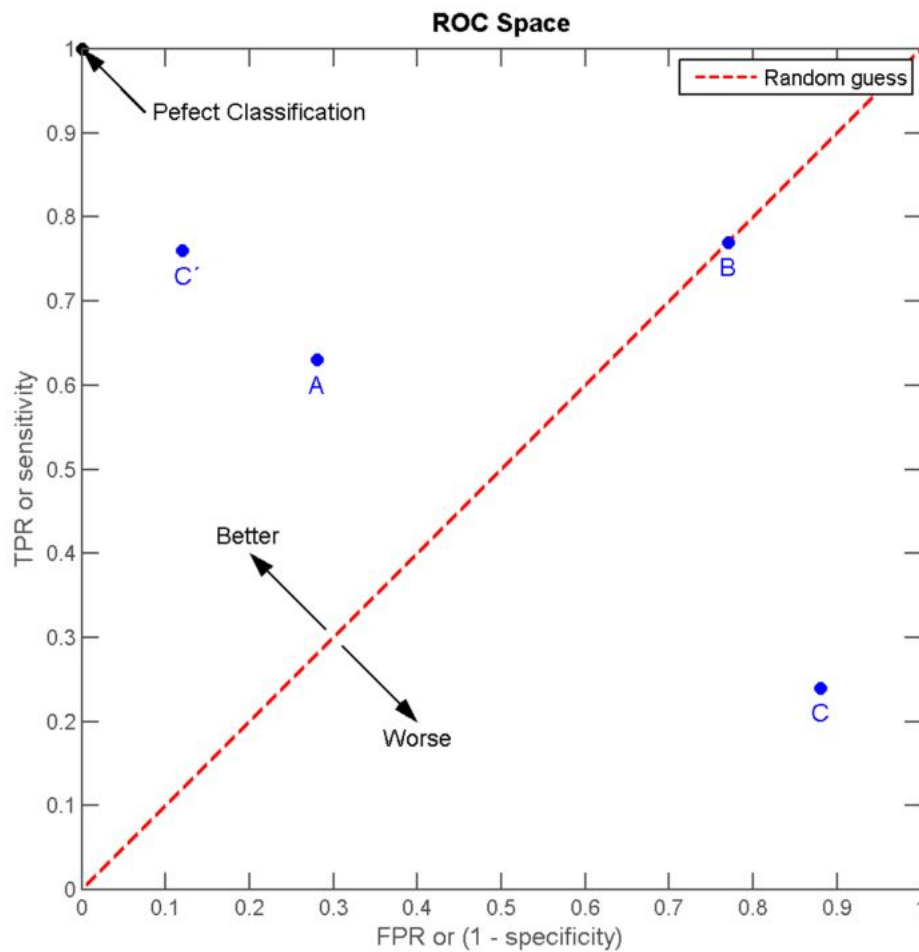


Figure 2.9: Receiver operating characteristic(ROC)

## 2.4 Automatic mesh generation in 3D

The FEM could be one of the most popular numerical methods in engineering. A necessary procedure is to discretize the problem domain into FE mesh of elementary shapes. In 3D problem, the conventional FEM allows hexahedron, tetrahedron, wedge and pyramid only which limits the flexibility of mesh generation. In order to achieve a reasonably accurate result, the mesh of the traditional FEM is required to conform to the bound-

## **2.4. AUTOMATIC MESH GENERATION IN 3D**

---

ary of the problem domain. As a consequence, it could be necessary to develop an automatic mesh generation algorithm using limited types of shapes (Frey and George, 2007). The development of an automatic generation algorithm is reported to be able to save 80% of the overall analysis time (Hughes et al., 2004).

The research toward automatic mesh generation is popular over decades (Blacker and Meyers, 1993; Löhner and Parikh; Owen, 2000; Watson, 1981). Generally speaking, the hexahedral element is favored over the tetrahedral element in terms of accuracy but it could be difficult to generate the mesh with hexahedral elements only automatically. Methods using plastering (Blacker and Meyers, 1993), whisker weaving (TAUTGES et al.), sweeping (Staten et al., 1999) and octree (Yerry and Shephard) were proposed to generate a hexahedral mesh. However, meshing of arbitrary domains using hexahedral elements without losing the exact geometric representation is not achievable up to now. Methods based on tetrahedral elements using Delaunay triangulation (Watson, 1981) and the advancing front technique (AFT) (Löhner and Parikh) were proposed for arbitrary domains. But Delaunay triangulation requires boundary recovery (George et al.; Liu et al., 2014) and AFT need to solve colliding fronts (Shewchuk, 1997). Furthermore, a mix use of all allowed elements (Owen, 2000) was proposed while the accuracy is not as satisfactory as that determined from the all-hexahedron mesh.



## 2.4. AUTOMATIC MESH GENERATION IN 3D

---

As a consequence, new numerical methods that reduce the limitation on element usage including X-FEM (Moës et al., 1999), isogeometric analysis (Hughes et al., 2004) (introduced in Sec. 2.1), finite cell method (FCM) (Parvizian et al., 2007) and SBFEM (Wolf, 2003) (introduced in Sec. 2.2) were proposed.

The X-FEM eases the burden posed on mesh generation as conforming to the geometric boundary is not necessary. Geometric discontinuity within the element is solved by the adoption of partition of unity (Melenk and Babuška, 1996) and enrichment functions. The method is then extended by the help of level-set (Osher and Sethian, 1988) to be able to solve the holes (Sukumar et al., 2001), problems with material interfaces (Fries) and flows (Chessa and Belytschko, 2003). It has been applied in field of fluid-structure-contact interaction problems (Mayer et al., 2010) and crack propagation for elastostatic problems (Gravouil et al., 2002; Moës et al., 2002) in 3D.

In FCM, the mesh is generated by simple unfitted structured mesh of higher-order basis functions and the geometry is represented by averaging the adaptive quadrature points which removes the necessity of boundary conforming. The method has been improved in terms of topology (Parvizian et al., 2012) and applied to voxel model (Schillinger et al.) and problems with material interfaces (Joulaian and Düster, 2013).

## **2.4. AUTOMATIC MESH GENERATION IN 3D**

---

Recently, an automatic mesh generation algorithm developed from the SBFEM and the octree-based algorithm using STL file (introduced in Sec. 2.4.1) was proposed (Liu et al., 2017). Arbitrary element faces are allowed in the scaled boundary finite element which significantly decreases the limitation of element shape. Mesh generated from an octree-based algorithm leads to higher quality elements and higher computational efficiency. However, the geometry can not be retained exactly in the STL file which means the inevitable geometric imperfection may result in considerable accuracy issue (Hughes et al., 2004).

### **2.4.1 STL file**

STL file is another popular format which is used to represent the surfaces in CAD industry other than NURBS introduced in Sec. 2.1.2, especially in 3D printing and rapid prototyping (Gill and Kaplas, 2009; Rengier et al., 2010). One of the most promising advantages of the STL format is its simplicity. Surfaces are divided into unstructured triangles but unexpected behaviors such as ill-shaped, overlapping and self-intersecting are allowed. Although the STL can not represent the geometric information exactly, it has gained more popularity and wider application than that of NURBS.

As a result, several surface re-meshing methods (Béchet et al., 2002; Wang et al., 2007) were proposed in order to conduct the mesh genera-

## 2.5. CONCLUSIONS

---

tion from the STL files. When used as the geometric input of the numerical method, a check of the unexpected behaviors including ill-shaped, overlapping and self-intersecting must be performed. A mesh repairing (Attene et al., 2013) can be adopted when these behaviors are observed. Furthermore, the elements generated will be in tetrahedral and the accuracy could be inferior to that determined from hexahedral elements even though high-quality triangular surface meshes are observed.

## 2.5 Conclusions

This chapter has summarized the linear theory of the Isogeometric analysis, together with a brief introduction on NURBS and its mathematical backgrounds, potentials and limitations. A brief introduction on the IGES file is also presented. Due to the dimensional mismatch, the SBFEM is the technique which can provide a seamless integration with the CAD modeling. As a consequence, the SBFEM is enhanced with the idea of the Isogeometric Analysis, the automatic mesh generation and the adaptive mesh refinement. The concept of the adaptive mesh refinement and its limitations are presented as well. The MLP then is introduced to overcome this limitation by the adoption of multiple error indicators. Finally, the algorithm used to generate octree mesh in 3D from STL file is presented.

# Chapter 3

## Isogeometric enhanced SBFEM in 2D

### 3.1 Introduction

The main objective of this chapter is to combine the concept of isogeometric analysis and the SBFEM. Non-uniform rational B-splines (NURBS) basis functions are employed to approximate the unknown fields in the circumferential direction. This provides a seamless integration with the CAD model. The method is further extended to problems with singularities within the framework of linear elastic fracture mechanics and to dynamic analysis. The proposed method enhances the conventional IGA and the salient features of the method are:

- No tensor-product patches as only the boundary information is required for the stress analysis

### 3.2. FORMULATION OF THE ISOGEOMETRIC SBFEM

---

- Feasible to have a n-sided polygonal domain of mixed type/order of the element on the edges, which leads to high flexibility in meshing and mesh transition
- Model strain/stress singularities without enrichment
- No internal mesh when studying the dynamic response at high frequencies

This chapter organized as follows. Section. 3.2 provides an overview of the SBFEM and important equations pertaining to linear elasticity, followed by extending the formulation to linear elastic fracture mechanics in Section 3.3. After that, methods that perform the interpolation of surface traction and displacement will be presented in Section 3.4 and Section 3.5. Numerical integration with NURBS will be mentioned in Section 3.6 before the accuracy and the convergence properties of the proposed techniques being demonstrated with benchmark problems in the context of linear elasticity and linear elastic fracture mechanics in Section 3.7, followed by concluding remarks in the last section. This chapter is published (Natarajan et al., 2015).

## 3.2 Formulation of the isogeometric SBFEM

The nodal coordinates on the boundary are denoted as  $x_b$ . As in a standard 1D iso-parametric finite element, the geometry of the element

### 3.2. FORMULATION OF THE ISOGEOMETRIC SBFEM

---

described by the coordinates  $\mathbf{x}_b(\eta)$  as shown in Fig. 2.5, is expressed as

$$\mathbf{x}_b(\eta) = \mathbf{N}(\eta)\mathbf{x}_b \quad (3.1)$$

where  $\mathbf{N}(\eta)$  is the shape function matrix. Without loss of generality, the origin of the Cartesian coordinate system is chosen at the scaling center. The geometry of the subdomain, described by  $x$ , is formed by scaling the boundary (Eq. 3.1)

$$\mathbf{x} = \xi \mathbf{x}_b(\eta) \quad (3.2)$$

where  $\xi$  is the normalized radial coordinate running from the scaling center towards the boundary, with  $\xi = 0$  at the scaling center and  $\xi = 1$  on the boundary. The coordinates  $\xi$  and  $\eta$  are the so-called scaled boundary coordinates. They are related to the polar coordinates  $r$  and  $\theta$ . The transformation is expressed as

$$\begin{aligned} r(\xi, \eta) &= \xi r_b(\eta) \\ \theta(\eta) &= \arctan \frac{y(\eta)}{x(\eta)} \end{aligned} \quad (3.3)$$

where  $r_b$  is the distance from the scaling center to a point on the boundary. The transformation between the Cartesian coordinates and the scaled boundary coordinates is similar to the coordinate transformation in the constructing iso-parametric finite elements. The scaling center must be placed at the position so that all boundary of the domain is visible from the point. This limitation can be easily solved by using a mesh generation algorithm

### 3.2. FORMULATION OF THE ISOGEOMETRIC SBFEM

---

described in Sec. 4. The displacement at any point are approximated by

$$\mathbf{u}(\xi, \eta) = \mathbf{N}(\eta)\mathbf{u}(\xi) \quad (3.4)$$

where  $\mathbf{N}(\eta)$  are the shape functions of elements on the boundary and  $\mathbf{u}(\xi)$  is the displacement along the radial lines, represented by a set of  $N$  analytical functions. By substituting Eq. 3.4 into the definition of strain-displacement relations, the strain  $\epsilon(\xi, \eta)$  are expressed as

$$\epsilon(\xi, \eta) = \mathbf{L}\mathbf{u}(\xi, \eta) \quad (3.5)$$

where  $\mathbf{L}$  is a linear operator matrix formulated in the scaled boundary coordinates

$$\mathbf{L} = \mathbf{b}_1(\eta)\frac{\partial}{\partial \xi} + \frac{1}{\xi}\mathbf{b}_2(\eta) \quad (3.6)$$

and

$$\mathbf{b}_1(\eta) = \frac{1}{|J|} \begin{bmatrix} y_b(\eta)_{,\eta} & 0 \\ 0 & -x_b(\eta)_{,\eta} \\ -x_b(\eta)_{,\eta} & y_b(\eta)_{,\eta} \end{bmatrix} \quad (3.7)$$

$$\mathbf{b}_2(\eta) = \frac{1}{|J|} \begin{bmatrix} -y_b & 0 \\ 0 & x_b \\ x_b & y_b \end{bmatrix}$$

The determinant of the Jacobian matrix is

$$|J| = x_b(\eta)y_b(\eta)_{,\eta} - y_b(\eta)x_b(\eta)_{,\eta} \quad (3.8)$$

### 3.2. FORMULATION OF THE ISOGEOMETRIC SBFEM

---

where  $x_b(\eta)$  and  $y_b(\eta)$  are given by Eq. 3.1. The stresses  $\sigma(\xi, \eta)$  are given by:

$$\sigma(\xi, \eta) = \mathbf{D}\mathbf{B}_1(\eta)\mathbf{u}(\xi)_{,\xi} + \frac{1}{\xi}\mathbf{D}\mathbf{B}_2(\eta)\mathbf{u}(\xi) \quad (3.9)$$

where in the above equation, the definition of strain and the linear operator matrix given by Eq. 3.5 and Eq. 3.6 are used with

$$\begin{aligned} \mathbf{B}_1(\eta) &= \mathbf{b}_1(\eta)\mathbf{N}(\eta) \\ \mathbf{B}_2(\eta) &= \mathbf{b}_2(\eta)\mathbf{N}(\eta)_{,\eta} \end{aligned} \quad (3.10)$$

substituting Eq. 3.5 and Eq. 3.9 in the virtual work statement for elastostatics (Deeks and Wolf, 2002b) and following the derivation

$$\begin{aligned} &\delta\mathbf{u}(\xi)^T \left( (\mathbf{E}_0\xi\mathbf{u}(\xi)_{,\xi} + \mathbf{E}_1^T\mathbf{u}(\xi))|_{\xi=1} - \mathbf{F} \right) - \\ &\int_0^1 \delta u(\xi)^T \left( \mathbf{E}_0\xi^2\mathbf{u}(\xi)_{,\xi\xi} + (\mathbf{E}_0 + \mathbf{E}_1^T - \mathbf{E}_1)\xi\mathbf{u}(\xi)_{,\xi} - \mathbf{E}_2\mathbf{u}(\xi) \right) d\xi = 0 \end{aligned} \quad (3.11)$$

where  $\mathbf{u}(\xi)$  is the nodal displacement vector and  $\mathbf{F}$  is the vector of equivalent boundary nodal forces, given by:

$$\mathbf{F} = (\mathbf{E}_0\xi\mathbf{u}(\xi)_{,\xi} + \mathbf{E}_1^T\mathbf{u}(\xi))|_{\xi=1} \quad (3.12)$$

By considering the arbitrariness of  $\delta\mathbf{u}(\xi)$ , the following ODE is obtained:

$$\mathbf{E}_0\xi^2\mathbf{u}(\xi)_{,\xi\xi} + (\mathbf{E}_0 + \mathbf{E}_1^T - \mathbf{E}_1)\xi\mathbf{u}(\xi)_{,\xi} - \mathbf{E}_2\mathbf{u}(\xi) = 0 \quad (3.13)$$



### 3.2. FORMULATION OF THE ISOGEOMETRIC SBFEM

---

where  $\mathbf{E}_0$ ,  $\mathbf{E}_1$  and  $\mathbf{E}_2$  are known as the coefficient matrices and are given by:

$$\begin{aligned}\mathbf{E}_0 &= \int_{\eta} \mathbf{B}_1(\eta)^T \mathbf{D} \mathbf{B}_1(\eta) |J| d\eta \\ \mathbf{E}_1 &= \int_{\eta} \mathbf{B}_2(\eta)^T \mathbf{D} \mathbf{B}_1(\eta) |J| d\eta \\ \mathbf{E}_2 &= \int_{\eta} \mathbf{B}_2(\eta)^T \mathbf{D} \mathbf{B}_2(\eta) |J| d\eta\end{aligned}\tag{3.14}$$

Eq. 3.13 is a homogeneous second-order differential equation. Its solution is obtained by introducing the variable  $\mathbf{\varnothing}(\xi)$

$$\mathbf{\varnothing} = \begin{Bmatrix} \mathbf{u}(\xi) \\ \mathbf{q}(\xi) \end{Bmatrix}\tag{3.15}$$

where  $\mathbf{q}(\xi)$  is the internal force vector

$$\mathbf{q}(\xi) = \mathbf{E}_0 \xi \mathbf{u}(\xi)_{,\xi} + \mathbf{E}_1^T \mathbf{u}(\xi)\tag{3.16}$$

The boundary nodal forces are related to the displacement functions by:

$$\mathbf{F} = \mathbf{q}(\xi = 1) = (\mathbf{E}_0 \xi \mathbf{u}(\xi)_{,\xi} + \mathbf{E}_1^T \mathbf{u}(\xi))|_{\xi=1}\tag{3.17}$$

This allows Eq. 3.13 to be transformed into a first order ordinary differential equation with twice the number of unknowns as:

$$\xi \mathbf{\varnothing}(\xi)_{,\xi} = -\mathbf{Z} \mathbf{\varnothing}(\xi)\tag{3.18}$$

### 3.2. FORMULATION OF THE ISOGEOMETRIC SBFEM

---

where  $\mathbf{Z}$  is a Hamiltonian matrix

$$\mathbf{Z} = \begin{bmatrix} \mathbf{E}_0^{-1} \mathbf{E}_1^T & -\mathbf{E}_0^{-1} \\ \mathbf{E}_1 \mathbf{E}_0^{-1} \mathbf{E}_1^T - \mathbf{E}_2 & -\mathbf{E}_1 \mathbf{E}_0^{-1} \end{bmatrix} \quad (3.19)$$

An eigenvalue decomposition of  $\mathbf{Z}$  is performed and it yields:

$$\mathbf{Z} \begin{bmatrix} \Phi_u \\ \Phi_q \end{bmatrix} = \begin{bmatrix} \Phi_u \\ \Phi_q \end{bmatrix} \Lambda_n \quad (3.20)$$

In Eq. 3.20,  $\Lambda_n = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$  contains only the eigenvalues with negative real part.  $\Phi_u$  and  $\Phi_q$  are the subsets of the eigenvectors corresponding to  $\Lambda_n$ . They represent the modal displacements and forces, respectively. The general solution of Eq. 3.18 is given by:

$$\mathbf{u}(\xi) = \Phi_u \xi^{-\Lambda_n} \mathbf{c} \quad (3.21)$$

$$\mathbf{q}(\xi) = \Phi_q \xi^{-\Lambda_n} \mathbf{c} \quad (3.22)$$

where  $\mathbf{c}$  are integration constants that are obtained from the nodal displacements  $\mathbf{u}_b = \mathbf{u}(\xi = 1)$  as:

$$\mathbf{c} = \Phi_u^{-1} \mathbf{u}_b \quad (3.23)$$

The complete displacement field of a point defined by the sector covered by a line element on the boundary is obtained by substituting Eq. 3.22 into Eq. 3.4 resulting in:

$$\mathbf{u}(\xi, \eta) = \mathbf{R}(\eta) \Phi_u \xi^{-\Lambda_n} \mathbf{c} \quad (3.24)$$

### 3.2. FORMULATION OF THE ISOGEOMETRIC SBFEM

---

Taking the derivative of  $\mathbf{u}(\xi)$  with respect to  $\xi$  and substituting into Eq. 3.9 the stress field  $\sigma(\xi, \eta)$  can be expressed as:

$$\sigma(\xi, \eta) = \Psi_\alpha(\eta) \xi_n^{-\Lambda} - \mathbf{I} \mathbf{c} \quad (3.25)$$

where the stress mode  $\Psi_\sigma(\eta)$  is defined as:

$$\Psi_\alpha(\eta) = \mathbf{D}(-\mathbf{B}_1(\eta) \Phi_u \Lambda_n + \mathbf{B}_2(\eta) \Phi_u) \quad (3.26)$$

The stiffness matrix of an element is obtained by first substituting Eq. 3.23 into Eq. 3.22 at  $\xi = 1$ . This results in:

$$\mathbf{F} = \Phi_q \Phi_u^{-1} \mathbf{u}_b \quad (3.27)$$

From Eq. 3.27, the stiffness matrix  $\mathbf{K}$  can be identified to be given by the expression

$$\mathbf{K} = \Phi_q \Phi_u^{-1} \quad (3.28)$$

The SBFEM has recently been extended to dynamic analysis in bounded domains (Song, 2009). Assuming time-harmonic behavior, the scaled boundary equation in displacement is extended as:

$$\mathbf{E}_0 \xi^2 \mathbf{u}(\xi)_{,\xi\xi} + (\mathbf{E}_0 + \mathbf{E}_1^T - \mathbf{E}_1) \xi \mathbf{u}(\xi)_{,\xi} - \mathbf{E}_2 \mathbf{u}(\xi) + \omega^2 \mathbf{M}_0 \xi^2 \mathbf{u}(\xi) = 0 \quad (3.29)$$

Where  $\mathbf{M}_0$  is a coefficient matrix defined as

$$\mathbf{M}_0 = \int_{\eta} \mathbf{N}^T \rho \mathbf{N} |J| d\eta \quad (3.30)$$

### 3.2. FORMULATION OF THE ISOGEOMETRIC SBFEM

---

Using Eq. 3.16 and Eq. 3.17, Eq. 3.30 can be transformed into an equivalent first-order non-linear differential equation in dynamic stiffness  $S(\omega)$ ,

$$(S(\omega) - E_1)E_0^{-1}(S(\omega) - E_1^T) - E_2 + \omega S(\omega)_{,\omega} + \omega^2 M_0 = 0 \quad (3.31)$$

The dynamic stiffness matrix  $S(\omega)$  relates the nodal forces to the displacements at the boundary as,

$$F = S(\omega)u(\xi = 1) \quad (3.32)$$

Eq. 3.31 is solved by expanding the dynamic stiffness into a series of continued fractions. For this purpose, it is expressed as

$$S(\omega) = K - \omega^2 M + \omega^4 [R^{(1)}]^{-1} \quad (3.33)$$

In Eq. 3.33, the first two terms represent the low-frequency expansion of the dynamic stiffness, whereas the third term corresponds to the residual of the low-frequency approximation. Substituting Eq. 3.33 in Eq. 3.31 and equating terms in increasing order of powers of  $\omega$  to zero yields equations for  $K$ ,  $M$  and  $R^{(1)}$ . Setting the constant terms equal to zero yields an algebraic Riccati equation for the static stiffness matrix  $K$ , which is equivalent to the solution process described earlier. Setting all terms in  $\omega^2$  equal to zero yields a Lyapunov equation for the low-order mass matrix  $M$ . The solution procedure of that Lyapunov equation is described in detail in (Song and Wolf, 1997). The remaining equation for the residual is solved by ex-

### 3.3. APPLICATION OF ISOGEOMETRIC SBFEM TO LINEAR ELASTIC FRACTURE MECHANICS

---

panding  $\mathbf{R}^{(1)}$  as

$$\mathbf{R}^{(1)} = \mathbf{S}_0^{(1)} - \omega^2 \mathbf{S}_1^{(1)} + \omega^4 [\mathbf{R}^{(2)}]^{-1} \quad (3.34)$$

Eq. 3.34 is analogous to the expansion of the dynamic stiffness, Eq. 3.33, where the coefficients  $\mathbf{S}_0^{(1)}$  and  $\mathbf{S}_1^{(1)}$  correspond to the ‘stiffness’ and ‘mass’ term of  $\mathbf{R}^{(1)}$ , respectively. Equations for  $\mathbf{S}_0^{(1)}$  and  $\mathbf{S}_1^{(1)}$  are found by substituting Eq. 3.34 into the equation for  $\mathbf{R}^{(1)}$ . This procedure is continued until the residual  $\mathbf{R}^{M_{cf}+1}$  can be neglected. The symbol of  $M_{cf}$  denotes the order of continued-fraction expansion. Substituting all terms of the expansion back into Eq. 3.33 yields:

$$\begin{aligned} \mathbf{S}(\omega) = \mathbf{K} - \omega^2 \mathbf{M} + \omega^4 \left( \mathbf{S}_0^{(1)} - \omega^2 \mathbf{S}_1^{(1)} \right. \\ \left. + \omega^4 \left( \mathbf{S}_0^{(2)} - \omega^2 \mathbf{S}_1^{(2)} + \dots + \omega^4 \left( \mathbf{S}_0^{(M_{cf})} - \omega^2 \mathbf{S}_1^{(M_{cf})} \right)^{-1} \right)^{-1} \right)^{-1} \end{aligned} \quad (3.35)$$

The coefficients  $\mathbf{S}_0^{(i)}$  and  $\mathbf{S}_1^{(i)}$  in Eq. 3.35 are calculated following a recursive procedure. A more detailed derivation can be found in (Song, 2009).

### 3.3 Application of isogeometric SBFEM to linear elastic fracture mechanics

An attractive feature of the SBFEM is that no a priori knowledge of the asymptotic solution is required to accurately handle the stress singularity at a crack tip as shown in Fig. 2.5. When modeling a cracked structure, a subdomain surrounding the crack tip is selected and the scaling center is placed at the crack tip. The boundary of the subdomain is divided into line

### 3.3. APPLICATION OF ISOGEOMETRIC SBFEM TO LINEAR ELASTIC FRACTURE MECHANICS

---

elements. When the scaling center is placed at the crack tip, the solution for the stress field in Eq.3.25 is expressed, by using Eq.3.3, as

$$\sigma(r, \eta) = \sum_i^n c_i r^{-(\lambda_i+1)} (r_\eta^{\lambda_i+1}(\eta) \psi_{\sigma_i}(\eta)) \quad (3.36)$$

where  $\psi_{\sigma_i}(\eta)$  is the  $i$ th stress mode, i.e. the  $i$ th column of the matrix  $\psi_\sigma(\eta)$ . Like the well-known William expansion (Williams, 1957), Eq. 3.36 is a power series of the radial coordinate  $r$ . The radial variation of each term of the series is expressed analytically by the power function  $r^{-(\lambda_i+1)}$ . At discrete points along the boundary, the angular coordinates (see Eq. 3.3) are arranged as a vector  $\theta(\eta)$  and the stress modes  $\psi_{\sigma_i}(\eta)$  are computed.  $\psi_{\sigma_i}(\eta)$  and  $\theta(\eta)$  from a parametric equation of the angular variation of stresses. The singular stress and the T-stress terms can be easily identified by the value of the exponent  $-(\lambda_i + 1)$ . When the real part of the exponent  $-(\lambda_i + 1)$  of a term is negative, the stresses of this term at the crack tip, i.e.  $\xi = 0$ , tend to infinity. When the exponent  $-(\lambda_i + 1)$  of a term is equal to 0, the stresses of this term are constant and contribute to the T-stress.

In the case of a crack in a homogeneous material or on a material interface, two singular terms exist in the solution. Denoting the singular stress modes as I and II, the singular stress  $\sigma^s(\xi, \eta)$  (superscript s for

### 3.3. APPLICATION OF ISOGEOMETRIC SBFEM TO LINEAR ELASTIC FRACTURE MECHANICS

---

singular stresses) are obtained from Eq. 3.36:

$$\sigma^s(r, \eta) = \sum_{i, \text{I, II}} c_i r^{-(\lambda_i+1)} (r_\eta^{(\lambda_i+1)} \psi_{\sigma_i}(\eta)) \quad (3.37)$$

Note that the singular stress terms are separated from other terms and the stress singularity is represented analytically. This allows the evaluation of the stress intensity factors by directly matching their definition with the singular stress. For convenience, the point on the boundary along the crack from  $\theta = 0$  is considered. The distance from the crack tip on the boundary is denoted as  $L_0 = r_\eta(\theta = 0)$ . From Eq. 3.37, the values of the singular stresses at this point are equal to:

$$\sigma^s(L_0, \theta = 0) = \sum_{i, \text{I, II}} c_i \psi_{\sigma_i}(\theta = 0) \quad (3.38)$$

where  $\psi_{\sigma_i}(\theta = 0)$  is the value of the stress modes at  $\theta = 0$ . It is obtained by interpolating  $\psi_{\sigma_i}(\eta)$  at the discrete points of  $\theta(\eta)$ . The stress intensity factors can be computed directly from their definition using the stresses in Eq. 3.38. For a crack in a homogeneous medium, the classical definition of stress intensity factors  $K_{\text{I}}$  and  $K_{\text{II}}$  for mode I and II are expressed as:

$$\begin{pmatrix} K_{\text{I}} \\ K_{\text{II}} \end{pmatrix} = \sqrt{2\pi r} \begin{pmatrix} \sigma_{\theta\theta}^s(r, \theta = 0) \\ \tau_{r\theta}^s(r, \theta = 0) \end{pmatrix} \quad (3.39)$$

### 3.4. SURFACE TRACTION

---

Formulating Eq. 3.39 at  $r = L_0$  results in

$$\begin{Bmatrix} K_I \\ K_{II} \end{Bmatrix} = \sqrt{2\pi L_0} \begin{Bmatrix} \sigma_{\theta\theta}^s(L_0, \theta = 0) \\ \tau_{r\theta}^s(L_0, \theta = 0) \end{Bmatrix} \quad (3.40)$$

The stress intensity factors are then determined by substituting the stress components  $\sigma_{\theta\theta}^s(L_0, \theta = 0)$  and  $\tau_{r\theta}^s(L_0, \theta = 0)$  obtained from Eq. 3.38 into Eq. 3.40. For a subdomain containing a crack tip, two of the eigenvalues are equal to 1. They represent the T-stress term and the rotational rigid body motion term, which does not contribute to the stresses. They are separated from other terms in Eq. 3.38 and expressed as (superscript T for the T-stress)

$$\sigma^T(\eta) = \sum_{i=T_I, T_{II}} c_i \psi_{\sigma_i}(\eta) \quad (3.41)$$

The T-stress along the crack front( $\theta=0$ ) is determined by interpolating the angular variation of the two stress modes  $(\theta(\eta), \psi_{\sigma_i}(\eta))$ .

## 3.4 Surface traction

In structural analysis, it is common to have boundary conditions such as displacement constraints and applied loads. Due to the property of the NURBS that the control points are not necessarily on the curve, surface tractions can not be applied by the same method used in conventional numerical method like FEM or SBFEM. A surface traction  $\Phi$  can be regarded



### 3.4. SURFACE TRACTION

---

as a Neumann boundary condition which can be expressed as

$$F = - \int_{\Gamma} [N] \Phi_n d\Gamma \quad (3.42)$$

where  $[N]$  describe the shape functions and  $\Phi_n$  is the surface tractions on the nodes.

As mentioned in Sec. 3.6, numerical integration would be preferred over mathematical deduction when the target function is an input. In the flavour of numerical integration, Eq. 3.42 can be expressed as followed.

$$F = - \sum_{i=1}^n a_i [N(\xi_i)] \Phi_n \quad (3.43)$$

where  $\xi_i$  is the integration point and  $a_i$  is the weight,  $n$  is the number of integration points and different quadrature rules require different number of points in order to achieve an optimal accuracy.

It can be found that the term  $[N(\xi_i)] \Phi_n$  is corresponding to  $f(x)$  in Eq. 3.49. In conventional FEM or SBFEM,  $\Phi_n$  can be determined as the real values on the nodes because, geometrically speaking, its shape function is interpolated from the given set of points. In other words, all nodes that determine the shape function in traditional FEM or SBFEM must be on the interpolating function. However, this is not the case in NURBS curves where it is the control points that play the same role as the nodes in existing shape function. In NURBS curves, apart from the first and the last

### 3.4. SURFACE TRACTION

---

points, the control points are not necessarily on the curves as illustrated in Fig. 2.2. This prevents us from adopting physical values on the nodes as  $\Phi_n$  in Eq. 3.43. Instead, a set of “control stress”  $\Phi_c$ , the control points of another NURBS curve that represent the surface traction geometrically, need to be determined as

$$\Phi_c = \arg \min_{\Phi_c} \frac{1}{2} \int_{-1}^1 \|\Phi(\xi) - [N(\xi)] \Phi_c\|^2 d\xi \quad (3.44)$$

It means that “control stress”  $\Phi_c$  describe a minimum mean squared error between surface traction NURBS curve and the real traction  $\Phi$ . One of the simplest mathematical methods to determine  $\Phi_c$  will be the least square method. Given the fact that the shape functions of this NURBS curve will be the same as that describe the geometry,  $[N(\xi)]$  can be considered as known. By selecting  $n$  sample points over the domain of the  $\Phi$ , Eq. 3.44 can be rewrite as

$$\Phi_c = \arg \min_{\Phi_c} \frac{1}{n} \sum_{i=1}^n \|\Phi(\xi_i) - [N(\xi_i)] \Phi_c\|^2 \quad (3.45)$$

Then “control stress”  $\Phi_c$  can be solved by least square as

$$\Phi_c = \left( [N(\xi)]^T [N(\xi)] \right)^{-1} [N(\xi)]^T \Phi(\xi) \quad (3.46)$$

and Eq. 3.43 in the case where NURBS is in use can be rewrite as

$$F = - \sum_{i=1}^n a_i [N(\xi_i)] \Phi_c \quad (3.47)$$

### 3.5. DISPLACEMENT INTERPOLATION

---

## 3.5 Displacement interpolation

Another difference between it with conventional FEM or SBFEM lies in the post processing. After solving the partial differential equation numerically, the displacements on the nodes will be one of the outputs in the traditional method. However, similar to what is discussed in Sec. 3.4, NURBS curves are defined by the control points that are not geometrically located on the curves. As a consequence, not only the inputs such as surface traction  $s$  need to be translated into a NURBS-like representation, the outputs such as the displacements will be the dummy values on the control points as well, or “control displacements”  $\{u_c\}$ . Unlike that in the traditional method, the “control displacements” do not have any physical meaning. It can only be used to interpolate the real displacements within its span.

$$\{u\} = \sum_{i=0}^n R(u) \{u^{(N)}\} \quad (3.48)$$

## 3.6 Numerical integrations

When computing the coefficient matrix (Eq. 3.14) in SBFEM, numerical integrations tend to be overwhelmingly preferred over the mathematical deduction. The reason behind lies in the flexibility of the numerical method and that deduction of exact integrations scheme to any given shape func-

### 3.6. NUMERICAL INTEGRATIONS

---

tions may not be feasible. Due to the fact that the polynomials are adopted as the shape function, the numerical integration methods such as Legendre Quadrature or Gauss Quadrature provide possibility for an exact integration numerically. An integration quadrature is normally defined as followed:

$$\int_{-1}^1 f(x)dx = \sum_{i=1}^n a_i f(x_i) \quad (3.49)$$

The integration of any targeted polynomial function defined on  $[-1, 1]$  can be explicitly expressed as series. A set of integration points  $\{x_1, x_2, \dots, x_n\} \in [-1, 1]$  and the corresponding weight  $s \{a_1, a_2, \dots, a_n\} \in \mathbb{R}$  determined from the integration quadrature can be adopted to perform an exact integration on the given function.

Although shape functions used in NURBS are not polynomials, they can be separated into several spans within which the basis functions are rational polynomials. Based on this property, the numerical integration quadrature on each of these spans can be applied to achieve a reasonably accurate result. In other words, the NURBS curve with a knot vector of  $\underbrace{[-1, -1, \dots, -1]}_{p+1}, u_0, \dots, u_n, \underbrace{[1, 1, \dots, 1]}_{p+1}$  can be integrated as

$$\int_{-1}^1 R(u)du = \int_{-1}^{u_0} R(u)du + \int_{u_0}^{u_1} R(u)du + \dots + \int_{u_n}^1 R(u)du \quad (3.50)$$

Since the rational polynomials instead of the non-rational ones are utilized as the shape functions in the NURBS, the difference between the outputs

### 3.7. NUMERICAL EXAMPLES

---

from Eq. 3.50 and the analytical solution will be so large that can not be regarded as machine error. Based on Eq. 2.9 we can conclude that the basis functions constructed by rational polynomials become non-rational if and only if the weight vector is identical i.e.  $\{w\} = [1, 1, \dots, 1]$  after normalization. That indicates the error of numerical integration will be decreased when the weight vector of the NURBS curve becomes more uniform as the basis functions are more close to the non-rational polynomials. In order to achieve this target, the knot insertion or the order elevation introduced in Section 2.1.2.1 and Section 2.1.2.2 can be used.

## 3.7 Numerical examples

In this section, the accuracy and the convergence properties of the proposed method are demonstrated with a few benchmark problems. First, the proposed method is applied to thick cantilever beam bending, to a plate with a center circular hole and to an L-shaped bracket. In the case of L-shaped bracket both the static and the dynamic responses are studied. The results are compared with analytical solution where available and with the results obtained using quadratic Lagrange shape functions. In the latter part of the section, problems involving strong discontinuities (e.g., plate with an edge crack, angled crack in an orthotropic body) are solved to illustrate the effectiveness of the proposed method. It is noted that, when the proposed method is applied to problems involving strong discontinuities, no special treatment is required, such as, the augmentation of finite

### 3.7. NUMERICAL EXAMPLES

---

element basis or IGA with asymptotic functions (Benson et al., 2010) (De Luycker et al., 2011). For the purpose of error estimation and convergence studies, the relative error, L 2 and H 1 norms are used. The displacement norm is given by:

$$\|u - u^h\|_{L^2(\Omega)} = \sqrt{\int_{\Omega} [(u - u^h) \cdot (u - u^h)]} \quad (3.51)$$

where  $u^h$  is the numerical solution and  $u$  is the analytical solution or a reference solution. The energy norm is given by

$$\|u - u^h\|_{H^1(\Omega)} = \sqrt{\int_{\Omega} [(\epsilon - \epsilon^h)^T D (\epsilon - \epsilon^h)]} \quad (3.52)$$

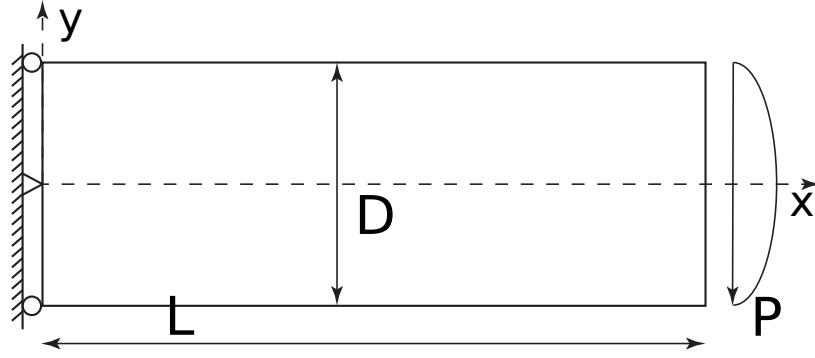
For the purpose of numerical integration of NURBS basis functions, we employ higher order Gaussian quadrature over each span of the NURBS basis functions. Other quadrature rules are possible as outlined in (Hughes et al., 2010). For Lagrange basis functions, conventional GaussLobatto quadrature is employed.

#### 3.7.1 Cantilever beam

A two-dimensional cantilever beam subjected to a parabolic shear load at the free end is examined as shown in Fig. 3.1. The geometry is: length  $L = 8$  m, height  $D = 4$  m. The material properties are: Youngs modulus  $E = 3 \times 10^7$  N m<sup>-2</sup>, Poissons ratio  $\nu = 0.25$ . The parabolic shear force is  $P = 250$  m. The exact solutions for the displacements are given by

### 3.7. NUMERICAL EXAMPLES

---



**Figure 3.1:** Cantilever beam: Geometry and boundary conditions.

(Augarde and Deeks, 2008):

$$\begin{aligned} u(x, y) &= \frac{Py}{6\bar{E}I} \left[ (6L - 3x)x + (2 + \bar{\nu}) \left( y^2 - \frac{D^2}{4} \right) \right] \\ v(x, y) &= -\frac{P}{6\bar{E}I} \left[ 3\bar{\nu}y^2(L - x) + (4 + 5\bar{\nu}) \frac{D^2x}{4} + (3L - x)x^2 \right] \end{aligned} \quad (3.53)$$

where  $I = D^3/12$  is the moment of inertia,  $\bar{E} = E$ ,  $\bar{\nu} = \nu$  and  $\bar{E} = E/(1 - \nu^2)$ ,  $\bar{\nu} = \nu/(1 - \nu)$  for plane stress and plane strain condition respectively.

The stress  $\sigma$  can be expressed as (Augarde and Deeks, 2008)

$$\sigma_{xx} = \frac{P(L - x)y}{I} \quad (3.54a)$$

$$\sigma_{yy} = 0 \quad (3.54b)$$

$$\tau_{xy} = -\frac{P}{2I} \left[ \frac{D^2}{4} - y^2 \right] \quad (3.54c)$$

The strain energy can be derived from Eq. 3.54 and Eq. 3.53 as

$$\epsilon = \frac{1}{2} \left( \frac{D^3 L^3 P^2}{36EI^2} + \frac{D^5 LP^2(1 + \nu)}{60EI^2} \right) \quad (3.55)$$

### 3.7. NUMERICAL EXAMPLES

---

In this example, rigid body motion is constrained by fixing 3 DOF on the left edge of the beam.  $u_x = 0$  for points at  $(0, -D/2)$  and  $(0, D/2)$  and  $u_y = 0$  for point at  $(0, 0)$ . Surface tractions determined from the analytical solution of stress in Eq. 3.54 are applied on the boundary.

From the description in Section 3.4, the expression of the surface tractions must be transformed into NURBS-like representation before they can be applied. The control points that describe a second order function as surface tractions in this example can be solved mathematically. Assume the knot vector is evenly spaced and the shape functions is in second order, i.e. knot vector  $\Xi = [0, 0, 0, 1, 1, 1]$ . Weight vector will be uniform because only the straight line is being interpolated, i.e. weight vector  $w = [1, 1, 1]$ . Three basis functions used in B-Spline will be

$$\begin{aligned} N_1 &= (1 - u)^2 \\ N_2 &= 2u(1 - u) \\ N_3 &= u^2 \end{aligned} \tag{3.56}$$

With the given targeted parabola as  $y = ax^2 + bx + c, x \in [0, 1]$ , the generalized control points for the NURBS curve will be

$$P = \begin{bmatrix} P_x \\ P_y \end{bmatrix} = \begin{bmatrix} 0 & m & 1 \\ c & n & a + b + c \end{bmatrix} \tag{3.57}$$



### 3.7. NUMERICAL EXAMPLES

---

where  $m$  and  $n$  are unknowns for the second control point. B-spline curve  $C = [N][P]$  then can be expressed as in parametric form as

$$\begin{cases} x = 2u(1-u)m + u^2 \\ y = c(1-u)^2 + 2u(1-u)n + (a+b+c)u^2 \end{cases} \quad (3.58)$$

After substituting Eq. 3.58 into  $y = ax^2 + bx + c$ , we then have the system of equations as

$$\begin{bmatrix} 0 \\ 0 \\ 2c - 2n + a + b \\ 2n - 2c \\ c \end{bmatrix} = \begin{bmatrix} 4am^2 - 4am + a \\ -8am^2 + 4am \\ 4am^2 - 2bm + b \\ 2bm \\ c \end{bmatrix} \quad (3.59)$$

$m$  and  $n$  then can be solved as

$$\begin{cases} m = \frac{1}{2} \\ n = \frac{b+2c}{2} \end{cases} \quad (3.60)$$

The numerical convergence of the relative error in the displacement norm and the relative error in the energy norm are shown in Fig. 3.2 for various order of NURBS basis functions with refinement. Fig. 3.2 also shows the error in the displacement norm when quadratic Lagrange shape functions are used along each edge within the scaled boundary formulation. It can be observed that NURBS basis functions yield superior accuracy when compared to Lagrange basis functions of the same order. It is seen that

### 3.7. NUMERICAL EXAMPLES

---

as the order of the shape functions is increased, the error decreases while the convergence rate increases.

#### 3.7.2 Infinite plate with a circular hole

In this example, an infinite plate with a traction free hole under uniaxial tension ( $\sigma = 1 \text{ N m}^{-2}$ ) along x-axis (see Fig. 3.3) is considered. The exact solution of the stresses in polar coordinate  $(r, \theta)$  is given by (Sukumar et al., 2001):

$$\sigma_x(r, \theta) = 1 - \frac{a^2}{r^2} \left( \frac{3}{2} \cos 2\theta + \cos 4\theta \right) + \frac{3a^4}{2r^4} \cos 4\theta \quad (3.61a)$$

$$\sigma_y(r, \theta) = -\frac{a^2}{r^2} \left( \frac{1}{2} \cos 2\theta - \cos 4\theta \right) - \frac{3a^4}{2r^4} \cos 4\theta \quad (3.61b)$$

$$\gamma_{xy}(r, \theta) = -\frac{a^2}{r^2} \left( \frac{1}{2} \sin 2\theta + \sin 4\theta \right) - \frac{3a^4}{2r^4} \sin 4\theta \quad (3.61c)$$

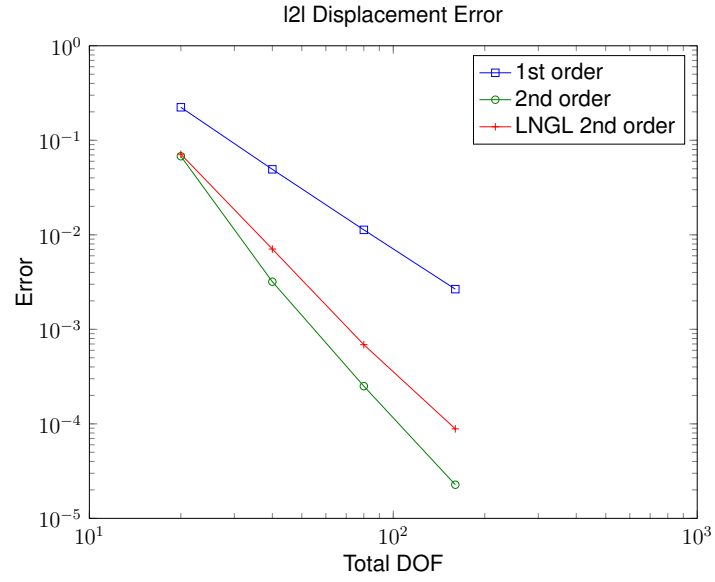
where  $a$  is the radius of the hole. Owing to symmetry, only one quarter of the plate is modeled. The material properties are: Youngs modulus  $E = 100 \text{ N m}^{-2}$  and Poissons ratio  $\nu = 0.3$ . The closed form displacement in Cartesian coordinate is given as

$$u_x(r, \theta) = \frac{R}{8G} \left[ \frac{r}{R}(\kappa + 1) \cos \theta + 2\frac{R}{r}((1 + \kappa) \cos \theta + \cos 3\theta) - 2\frac{R^3}{r^3} \cos 3\theta \right] \quad (3.62a)$$

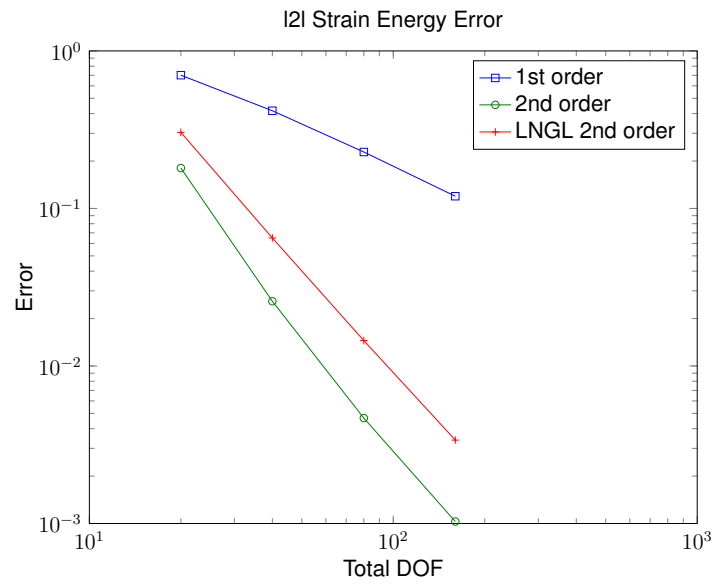
$$u_y(r, \theta) = \frac{R}{8G} \left[ \frac{r}{R}(\kappa - 3) \sin \theta + 2\frac{R}{r}((1 - \kappa) \sin \theta + \sin 3\theta) - 2\frac{R^3}{r^3} \sin 3\theta \right] \quad (3.62b)$$

### 3.7. NUMERICAL EXAMPLES

---



(a) the relative error in displacement norm ( $L^2$ )

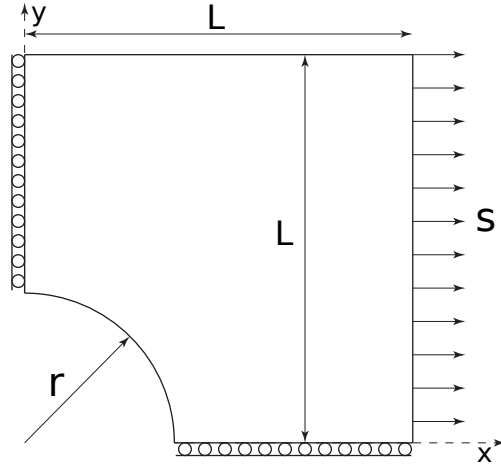


(b) the relative error in the energy norm

**Figure 3.2:** Bending of thick cantilever beam: Convergence results

### 3.7. NUMERICAL EXAMPLES

---



**Figure 3.3:** Infinite plate with a circular hole: geometry and boundary conditions

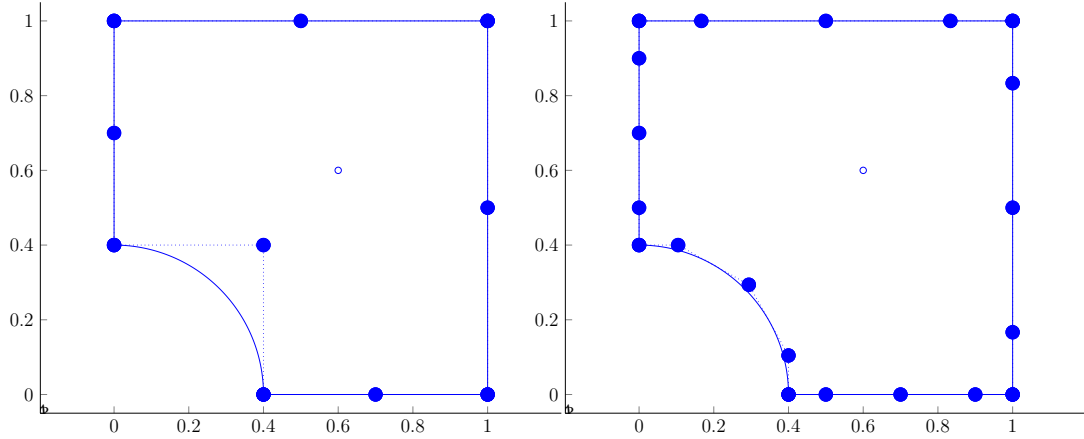
where  $G$  is the shear modulus and  $\kappa$  (Kolosoov constant) is defined as

$$\kappa = \begin{cases} 3 - 4\nu \\ \frac{3 - \nu}{1 + \nu} \end{cases} \quad (3.63)$$

In this example, analytical tractions in Eq. 3.61 are applied on the boundary. The left and bottom boundaries are constrained with a roller boundary condition.  $u_x = 0$  where  $y = 0$  and  $u_y = 0$  where  $x = 0$ .

The convergence rate in terms of the displacement norm is shown in Fig. 3.5. It is observed that the error decreases as the order of the shape functions is increased. Another unique feature of the proposed method is that the method allows different type or order of shape functions to be employed for different segments of the boundary. For example, in Fig. 3.4, the arc AB and the line segment BC can be represented by different basis

### 3.7. NUMERICAL EXAMPLES



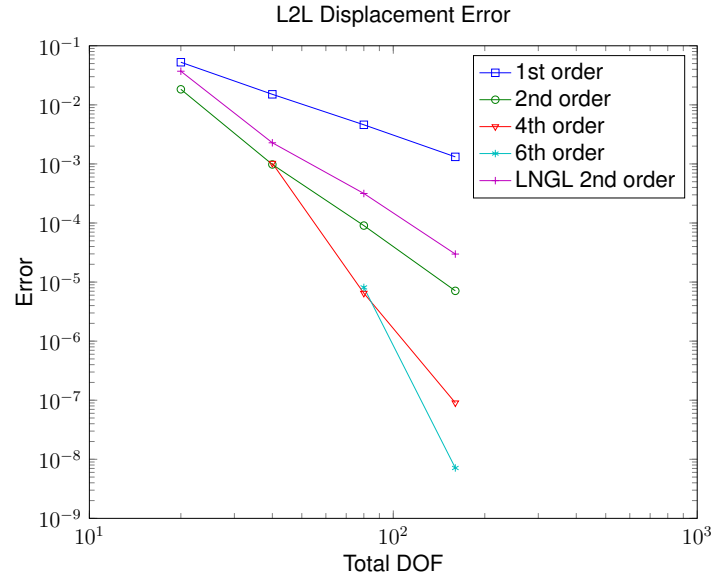
**Figure 3.4:** Plate with circular hole: control net for two different discretizations. Control points ( ● ), Boundary lines ( — ), Control polygon ( - - ● - - ), Scaling center ( ○ ).

functions, i.e., the arc can be represented by NURBS and the line segment can be represented by conventional Lagrange basis functions. Fig. 3.6 shows the plot of the NURBS basis and Lagrange basis functions. It can be seen that at point B, the shape function is continuous. In order to assess the behavior, we represent the arc with quadratic NURBS and the straight lines with conventional Lagrange shape functions. Since the NURBS are interpolatory at the ends, the compatibility requirement is automatically satisfied (see Fig. 3.7). The results from both the approaches converge with mesh refinement

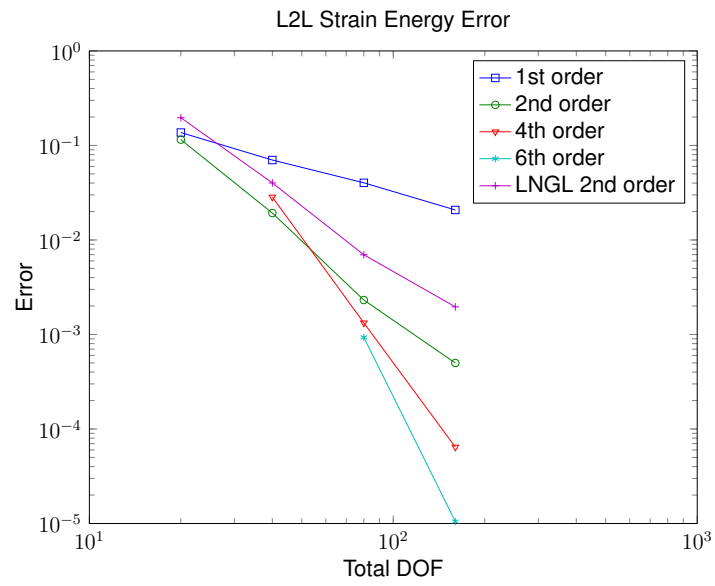
#### 3.7.3 L-shaped bracket

In this example, an L-shaped bracket with isotropic material properties is considered. Fig. 3.8 shows the geometry and the boundary conditions of the problem. The L-shaped bracket is fixed at one end and sub-

### 3.7. NUMERICAL EXAMPLES



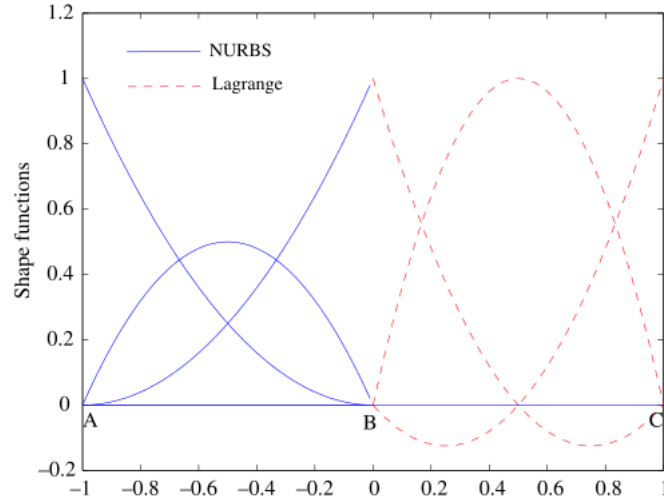
(a) the relative error in displacement norm ( $L^2$ )



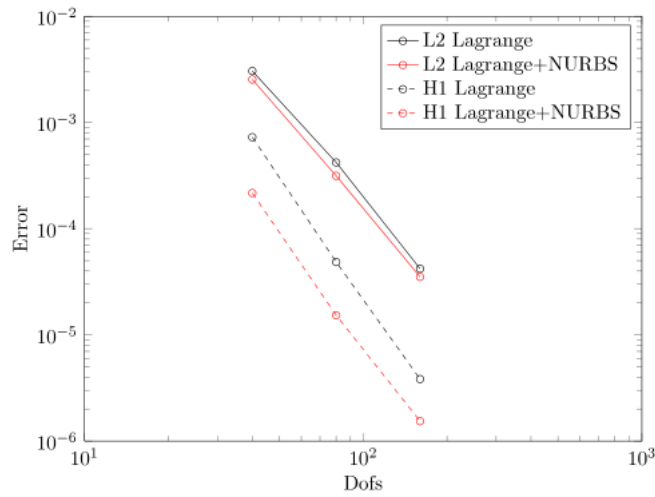
(b) the relative error in the energy norm

**Figure 3.5:** Bending of thick cantilever beam: Convergence results

### 3.7. NUMERICAL EXAMPLES



**Figure 3.6:** NURBS basis functions and Lagrange basis functions. It can be seen that at Point  $B$ , the shape functions are continuous

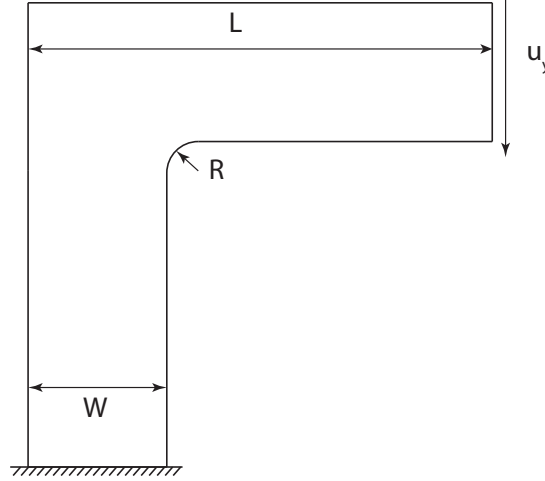


**Figure 3.7:** Infinite plate with a circular hole: Convergence results for the relative error in the displacement norm ( $L^2$ ) and the relative error in the energy norm. In this case, the arc is represented with NURBS and Lagrange shape functions.

### 3.7. NUMERICAL EXAMPLES

---

jected to downward vertical displacement at the other end. Plain strain conditions are assumed. The dimension of the example is:  $L = 14$  m,



**Figure 3.8:** Geometry and boundary conditions

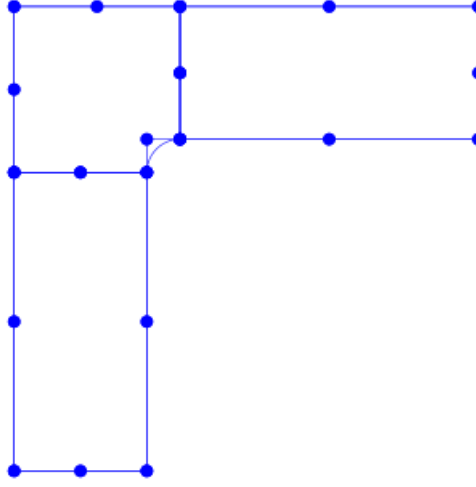
$W = 4$  m and  $R = 1$  m. While the material properties are: Young's modulus  $E = 1 \times 10^3$  MPa and poisson's ratio  $\nu = 0.3$ . This problem was studied in (Lipton et al., 2010) by employing the conventional IGA. In their study, the fillet was modeled as a separate path using biquadratic NURBS with nine control points.

In the present study, the control mesh is directly employed for the stress analysis. However, as the domain does not meet the star convexity, we divide the domain into three subdomains (see Fig. 3.9). We employ NURBS to represent the fillet, whilst for the straight lines, we employ Lagrange basis functions. The results from the present approach are compared with conventional finite element analysis using the commercial software ANSYS®.



### 3.7. NUMERICAL EXAMPLES

---

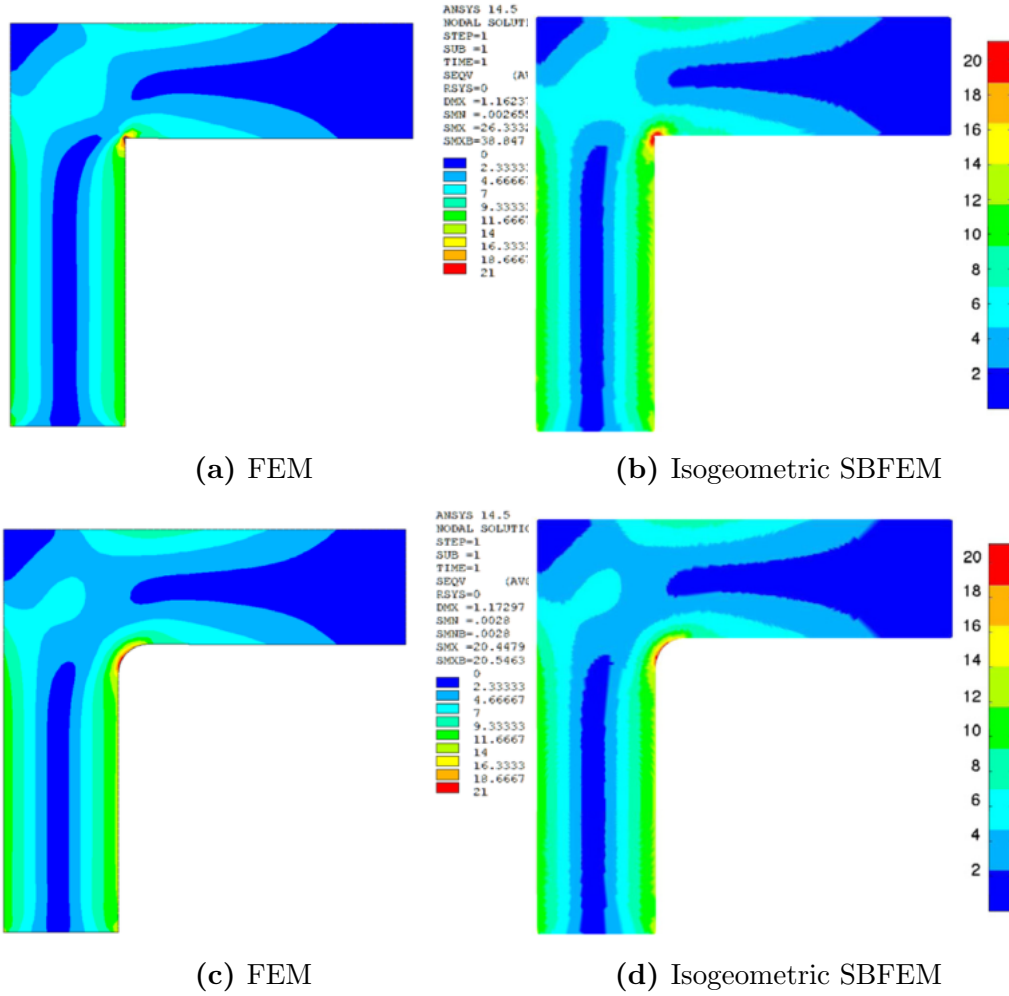


**Figure 3.9:** Control net where ‘filled’ circles represents control points

A total of 2000 8-node quadrilateral elements were used for the finite element analysis. Fig. 3.10 shows the von Mises equivalent stress for the L-shaped bracket with and without the fillet. As expected, the no fillet case shows higher stress when compared to the L-shaped bracket with the fillet. From Fig. 3.10, it can be observed that the results from the present approach qualitatively match with the FE solution. It should be noted that, the proposed method is computationally less intensive than the conventional IGA as it requires only the boundary information.

Next, we extend the present formulation to study the transient response of an L-shaped bracket. The dimensions and the boundary conditions are shown in Fig. 3.11.

### 3.7. NUMERICAL EXAMPLES

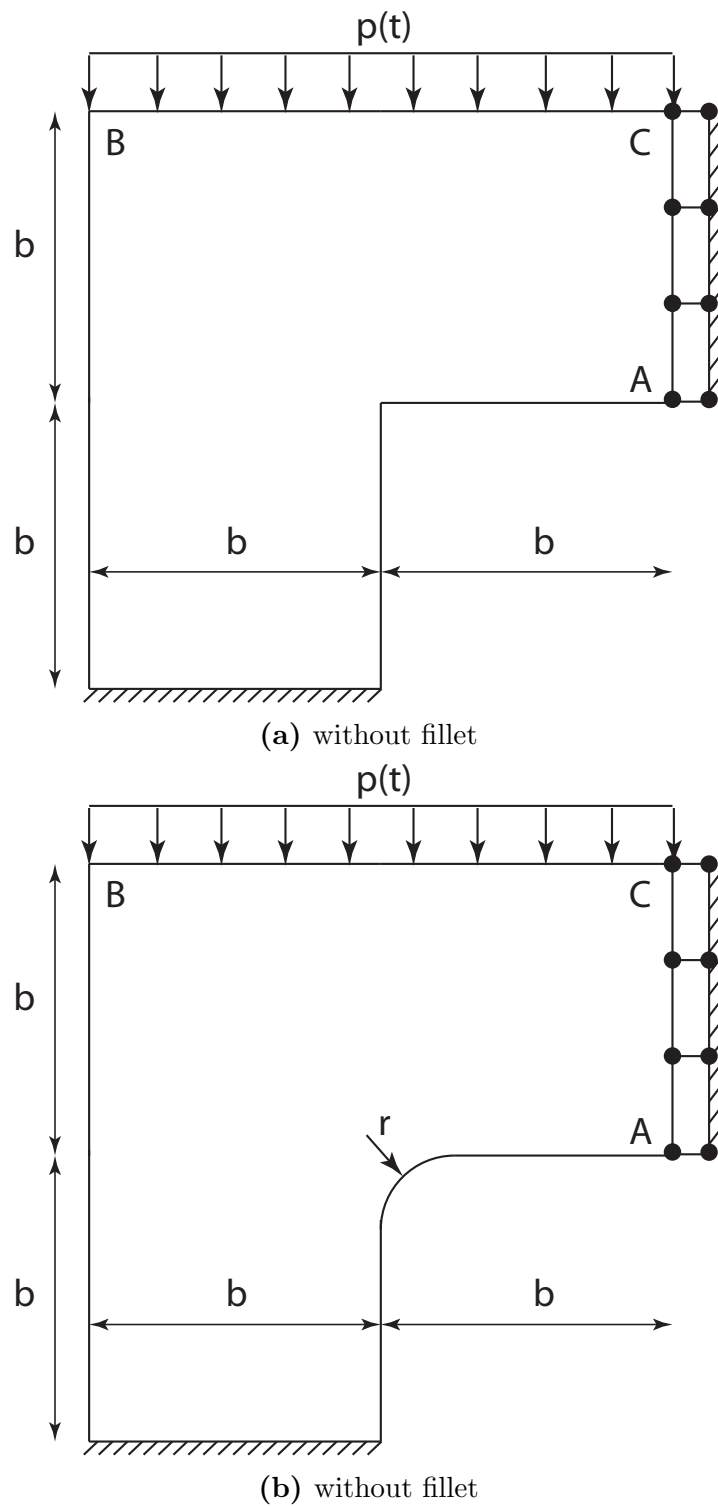


**Figure 3.10:** Von Mises equivalent stress contours for L-shaped bracket without and with fillet. The stress values are in Mpa

In the example,  $b = 1$  m and  $r = 0.2$  m. A state of plane stress is considered and the material properties are: Youngs modulus  $E = 1 \text{ N m}^{-2}$ , poisson's ratio,  $\nu = 1/3$  and mass density,  $\rho = 1 \text{ kg m}^{-3}$ . The shear wave velocity is  $c_s = \sqrt{3/8} \text{ m s}^{-1}$  and the dilatational wave velocity  $c_p = \sqrt{9/8} \text{ m s}^{-1}$ . The order of the continued fraction used in Eq. 3.35 is chosen as  $M_{cf} = 6$ . A uniform pressure  $p(t)$  is applied at the side BC of the

### 3.7. NUMERICAL EXAMPLES

---



**Figure 3.11:** L-shaped bracket: geometry and boundary conditions for transient analysis

### 3.7. NUMERICAL EXAMPLES

---

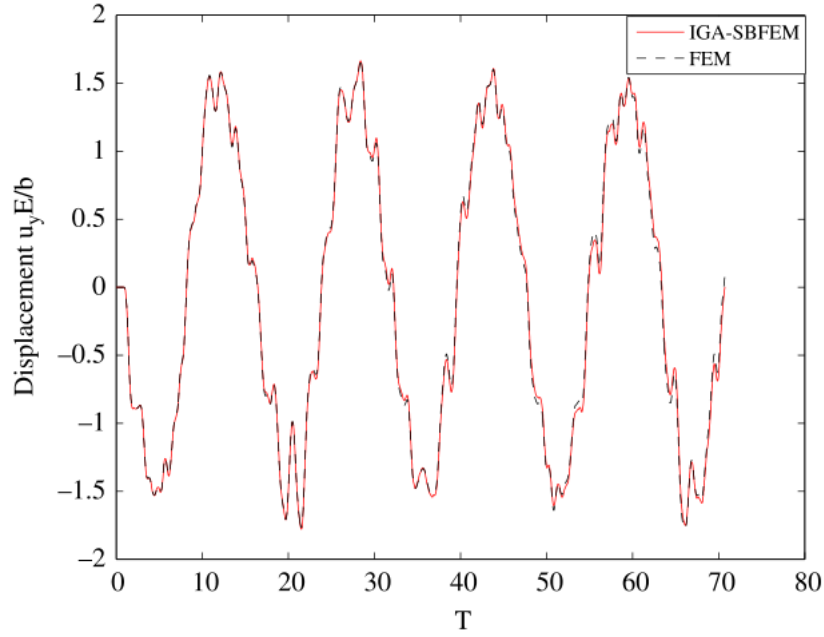
bracket. The pressure varies as a triangular impulse in the time domain. It reaches a peak value  $p$  at time  $t = 0.5b/c_p$ , reduces to 0 at  $t = b/c_p$  and stays at 0 afterwards. The time integration is carried out by using Newmark's method with  $\gamma = 1/2$  and  $\beta = 1/4$ . The time step is chosen as  $\delta t = 0.025b/c_p$ .

The calculation is performed for 3000 time steps. For the Isogeometric-SBFEM, the arc is represented with quadratic NURBS functions and the straight lines are discretized with Lagrange basis functions. As the problem domain does not satisfy the star convexity, the domain is sub-divided into three subdomains as done in the static example. The scaling center for each of the subdomain is placed at the center of the subdomain. To demonstrate the efficacy of the present formulation, the results are compared with those obtained using the conventional finite element method. A FE mesh leading to a similar accuracy is identified from a convergence study. The FE analysis is performed with the commercial software ANSYS® (a total of 2000 8-node quadrilateral elements were employed for this study).

The vertical displacement responses at point A are plotted in Fig. 3.12 as a function of the dimensionless time  $T = c_p t/b$ . It can be seen that the results from the present formulation agree well with the finite element solution.

### 3.7. NUMERICAL EXAMPLES

---



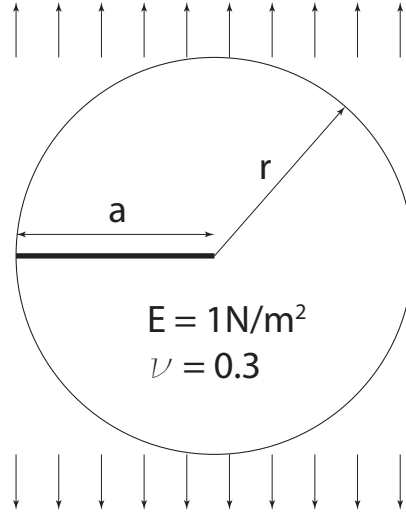
**Figure 3.12:** Vertical displacement of L-shaped bracket at Point A: comparison with conventional FE solution

#### 3.7.4 Circular disk with an edge crack in tension

Next, the present formulation is applied to problems with strong discontinuities and singularities. The unique feature of the proposed framework is that the geometry is exactly represented by using NURBS and the singularities are captured semi-analytically without a priori knowledge of the asymptotic fields. In the first example, consider a circular disk with an edge crack (see Fig. 3.13) with Young's modulus  $E = 1 \text{ N m}^{-2}$  and Poisson's ratio  $\nu = 0.3$ .

### 3.7. NUMERICAL EXAMPLES

---



**Figure 3.13:** Circular disk with an edge crack

The analytical displacements solution for mode I are given by:

$$u_x = \frac{1}{2} \left[ \left( \kappa + \frac{n}{2} + (-1)^n \right) \cos \left( \frac{n\theta}{2} \right) - \frac{n}{2} \cos \left( \left( \frac{n}{2} - 2 \right) \theta \right) \right] \quad (3.64a)$$

$$u_y = \frac{1}{2} \left[ \left( \kappa - \frac{n}{2} - (-1)^n \right) \sin \left( \frac{n\theta}{2} \right) + \frac{n}{2} \sin \left( \left( \frac{n}{2} - 2 \right) \theta \right) \right] \quad (3.64b)$$

$$(3.64c)$$

where  $\kappa$  is the Kolosov constant defined in Eq. 3.63.

### 3.7. NUMERICAL EXAMPLES

---

The analytical stress solution for mode I are given by:

$$\sigma_{xx} = \frac{n}{2} \left[ \left( 2 + \frac{n}{2} + (-1)^n \right) \cos \left( \left( \frac{n}{2} - 1 \right) \theta \right) - \left( \frac{n}{2} - 1 \right) \cos \left( \left( \frac{n}{2} - 3 \right) \theta \right) \right] \quad (3.65a)$$

$$\sigma_{yy} = \frac{n}{2} \left[ \left( 2 - \frac{n}{2} - (-1)^n \right) \cos \left( \left( \frac{n}{2} - 1 \right) \theta \right) + \left( \frac{n}{2} - 1 \right) \cos \left( \left( \frac{n}{2} - 3 \right) \theta \right) \right] \quad (3.65b)$$

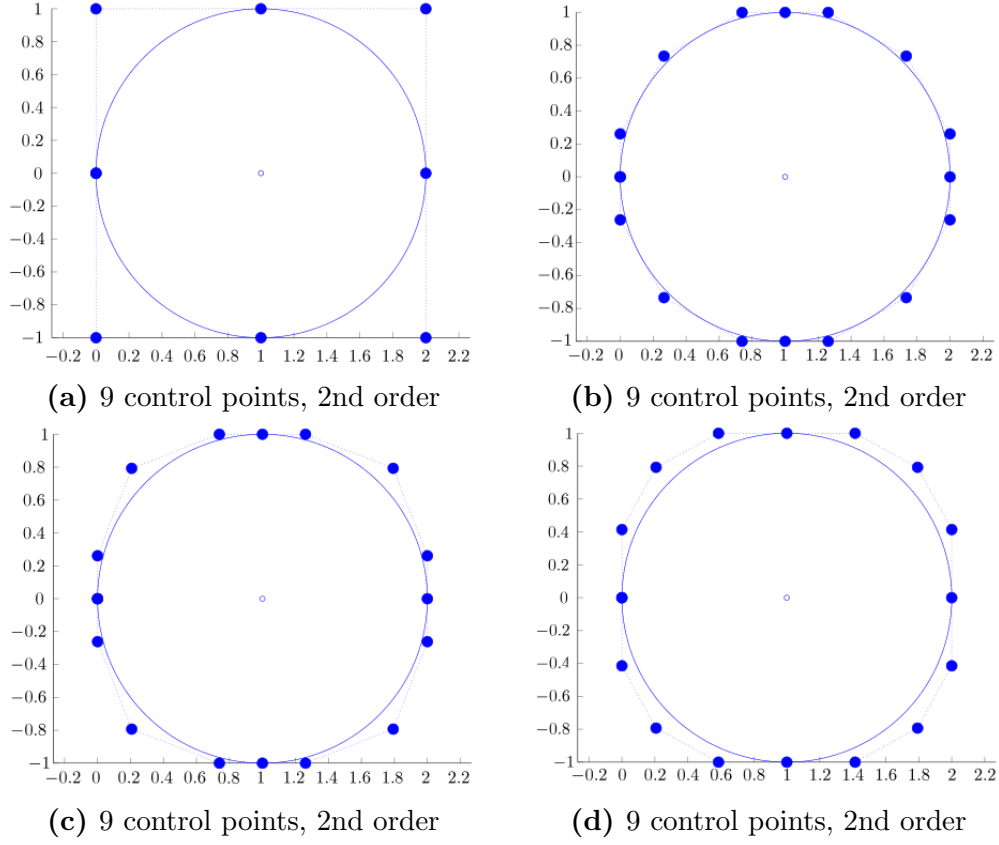
$$\tau_{xy} = \frac{n}{2} \left[ \left( \frac{n}{2} - 1 \right) \sin \left( \left( \frac{n}{2} - 3 \right) \theta \right) - \left( \frac{n}{2} + (-1)^n \right) \sin \left( \left( \frac{n}{2} - 3 \right) \theta \right) \right] \quad (3.65c)$$

In this example, the circular disk is represented by NURBS. The control net and the location of control points are shown in Fig. 3.14 for different NURBS orders.

The circular disk is subjected to a far field tension and the displacement and the stress modes are computed by the proposed isogeometric SBFEM. It is noted that only the boundary of the circular disk is discretized using the NURBS and no tensor product of the corresponding knot vectors is required to represent the unknown fields inside the domain. The convergence of the numerical stress intensity factor (SIF) and the T-stress with mesh refinement is shown in Tab. 3.1.

It can be seen that with mesh refinement the SIF and the T-stress converge. Increasing the order of the NURBS functions increases the con-

### 3.7. NUMERICAL EXAMPLES



**Figure 3.14:** Meshing of the circular disk with an edge notch

**Table 3.1:** T-stress and stress intensity factors for circular disk with an edge crack.

Total DOF	NURBS $p = 2$		NURBS $p = 4$		NURBS $p = 6$	
	SIF	T-stress	SIF	T-stress	SIF	T-stress
18	2.3520	2.9442				
34	2.8693	5.1991	2.8838	5.4050		
74	2.8838	5.3112	2.8840	5.3445	2.8840	5.3447
130	2.8840	5.3318	2.8840	5.3444	2.8840	5.3444



### 3.7. NUMERICAL EXAMPLES

---

vergence behavior. Eq. 3.36 is the parametric equation for the stress field in the polar coordinates  $r$  and  $\theta$ . The terms  $(r_\eta^{\lambda_i+1}(\eta)\psi_{\sigma_i}(\eta))$  in Eq. 3.36 together with  $\theta(\eta)$  in Eq. 3.3 are the stress modes describing the angular distribution at a constant radial coordinate  $r$ . For the converged result, Fig. 3.15 shows the displacement and the stress distribution at a constant radial coordinate  $r$  around the crack tip for mode I fracture. Each of the stress modes is normalized with its value of  $\sigma_{yy} = 0^\circ$ .

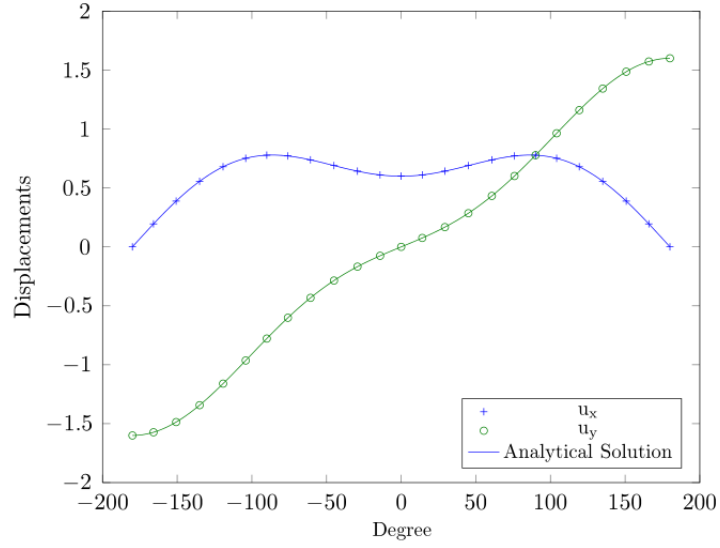
The stress modes from the scaled boundary formulation are compared with the analytical solutions and a very good agreement is observed in Tab. 3.1. The convergence of the displacement and stress modes with  $h$  and  $p$  refinement is shown in Fig. 3.16. It can be seen that with refinement the solution converges monotonically.

#### 3.7.5 Edge crack in tension

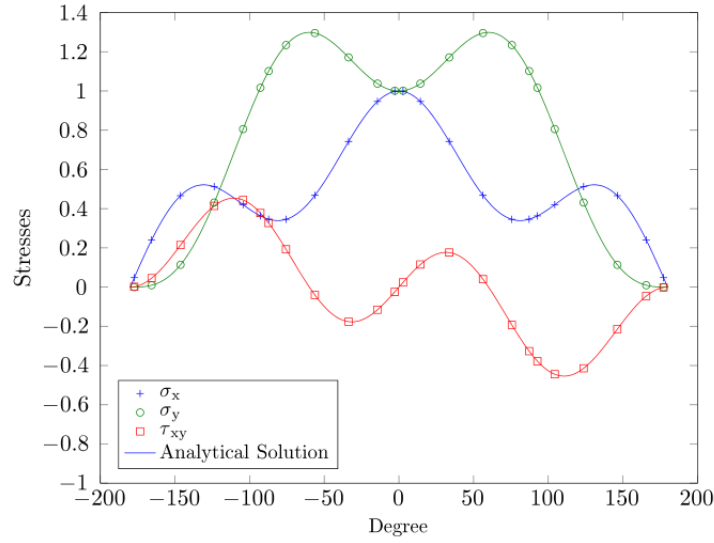
Consider a plate with an edge crack loaded in tension  $\sigma = 1N/m^2$  over the top and the bottom edges. The geometry, loading and boundary conditions are shown in Fig. 3.17. In the figure,  $L = 2m$  and  $H = 1m$ . The reference mode I SIF is given by:

$$K_1 = F \left( \frac{a}{H} \sigma \sqrt{\pi a} \right) \quad (3.66)$$

### 3.7. NUMERICAL EXAMPLES



(a) displacement modes



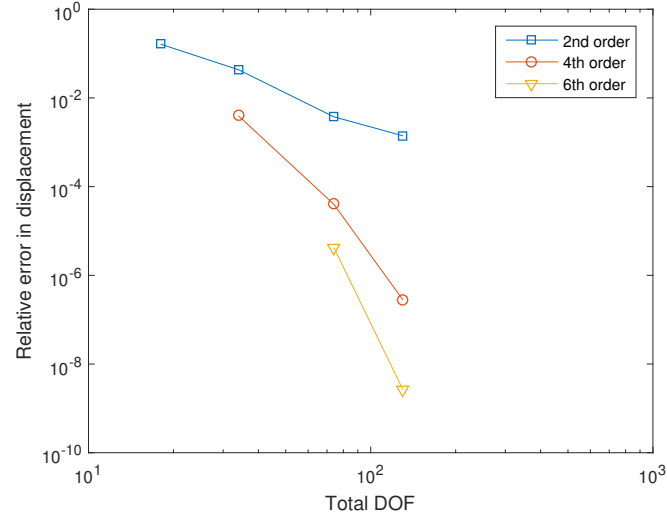
(b) stress modes

**Figure 3.15:** Displacement and stress modes of circular disk with an edge crack using cubic NURBS functions

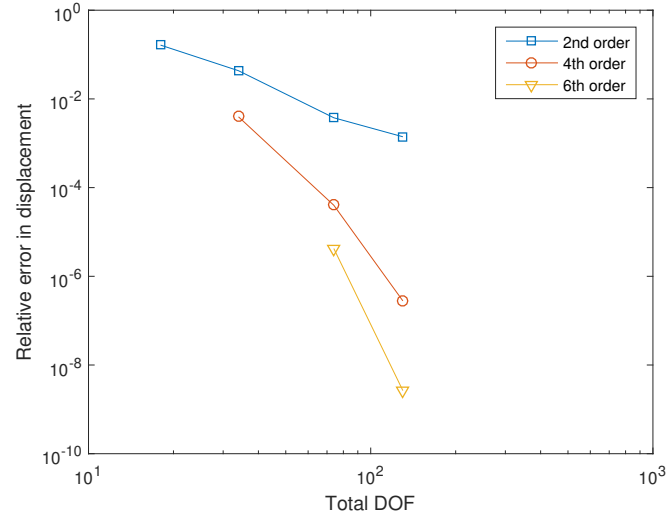
where  $a$  is the crack length,  $H$  is the plate width, and  $F(a/H)$  is an empirical function given by (for  $a/H \leq 0.6$ )

$$F\left(\frac{a}{H}\right) = 1.12 - 0.231\left(\frac{a}{H}\right) + 10.55\left(\frac{a}{H}\right)^2 - 27.72\left(\frac{a}{H}\right)^3 + 30.39\left(\frac{a}{H}\right)^4 \quad (3.67)$$

### 3.7. NUMERICAL EXAMPLES



(a) Displacement mode

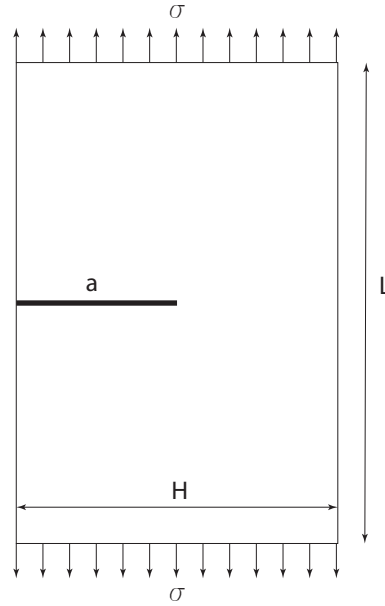


(b) Displacement mode

**Figure 3.16:** Circular disk with an edge crack: convergence of the displacement mode and stress mode

The convergence of the mode I SIF and the T-stress with the mesh size and the order of the NURBS basis function is illustrated in Tab. 3.2. It can be seen that decreasing the mesh size and increasing the order of the NURBS basis function, the numerically obtained SIF and the T-stress

### 3.7. NUMERICAL EXAMPLES



**Figure 3.17:** Plate with an edge crack under tension

converge.

#### 3.7.6 Angled crack in an orthotropic body

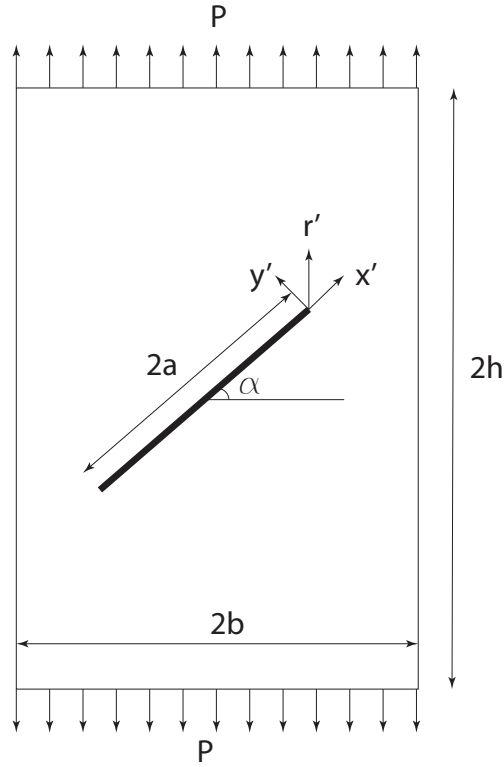
In this example, consider an orthotropic plate ( $b/h = 1$ ) with an angled center-crack of length  $a/h = 0.5$  under uniform far field tension along

**Table 3.2:** Convergence of the mode I SIF and T-stress for an edge crack in tension

Total	NURBS $p = 2$		NURBS $p = 4$		NURBS $p = 6$	
DOF	$K_1$	T-stress	$K_1$	T-stress	$K_1$	T-stress
62	2.7011	-0.4137	2.6647	-0.4170		
98	2.7632	-0.4210	2.8343	-0.4216	2.8335	-0.4217
170	2.8028	-0.4216	2.8252	-0.4217	2.8245	-0.4217
329	2.8264	-0.4217	2.8247	-0.4217	2.8246	-0.4217
Eq. 3.66	2.8264	-				

### 3.7. NUMERICAL EXAMPLES

---



**Figure 3.18:** Angled crack in a rectangular orthotropic body: geometry

the two opposite sides (see Fig. 3.18).

The elastic properties of the material are  $E_{11} = E_{22} = E_{33} = 106.17$  GPa,  $G_{12} = G_{23} = G_{13} = 108.25$  GPa, and  $\nu_{12} = \nu_{23} = \nu_{13} = 0.4009$ . The results are compared with those obtained in (Banks-Sills et al., 2005). The convergence of mode I and mode II SIF with mesh size is shown in Tab. 3.3 for a crack at an angle  $\alpha = \pi/12$ . The influence of the order of the shape functions is also shown. It can be seen that increasing the number of degrees of freedom, the error in the numerical SIF decreases. A very good agreement is observed.

### 3.7. NUMERICAL EXAMPLES

**Table 3.3:** Convergence of the generalized stress intensity factors  $(\overline{K_I}, \overline{K_{II}}) = (K_I, K_{II})/P\sqrt{(\pi a)}$

Order		Number of DOF		
		160	240	320
2	$\overline{K_I}$	1.2770	1.2723	1.2706
	$\overline{K_{II}}$	0.2918	0.2915	0.2913
4	$\overline{K_I}$	1.2700	1.2697	1.2697
	$\overline{K_{II}}$	0.2914	0.2913	0.2913
6	$\overline{K_I}$		1.2668	1.2670
	$\overline{K_{II}}$		0.2913	0.2913

**Table 3.4:** Generalized stress intensity factors  $K_I/P\sqrt{\pi a}$  of angled crack in rectangular orthotropic body

Angle	$K_I$			
$\alpha$	(Banks-Sills et al., 2005)	Order of the curve		
		2	4	6
0	1.3755	1.3603	1.3583	1.3583
$\pi/12$	1.2692	1.2706	1.2697	1.2670
$\pi/6$	1.0268	1.0277	1.0270	1.0270
$\pi/4$	0.6952	0.6941	0.6944	0.6943
$\pi/3$	0.3579	0.3582	0.3580	0.3580
$5\pi/12$	0.1095	0.0996	0.1003	0.1004

Also, increasing the order of the NURBS basis functions, the error decreases. The influence of the crack orientation on the mode I and mode II SIF is shown in Tab. 3.4 and Tab. 3.5. A total of 80 control points are used with different order of the NURBS basis functions. It can be seen that a very good agreement is observed.

### 3.7. NUMERICAL EXAMPLES

**Table 3.5:** Generalized stress intensity factors  $K_{II}/P\sqrt{\pi a}$  of angled crack in rectangular orthotropic body

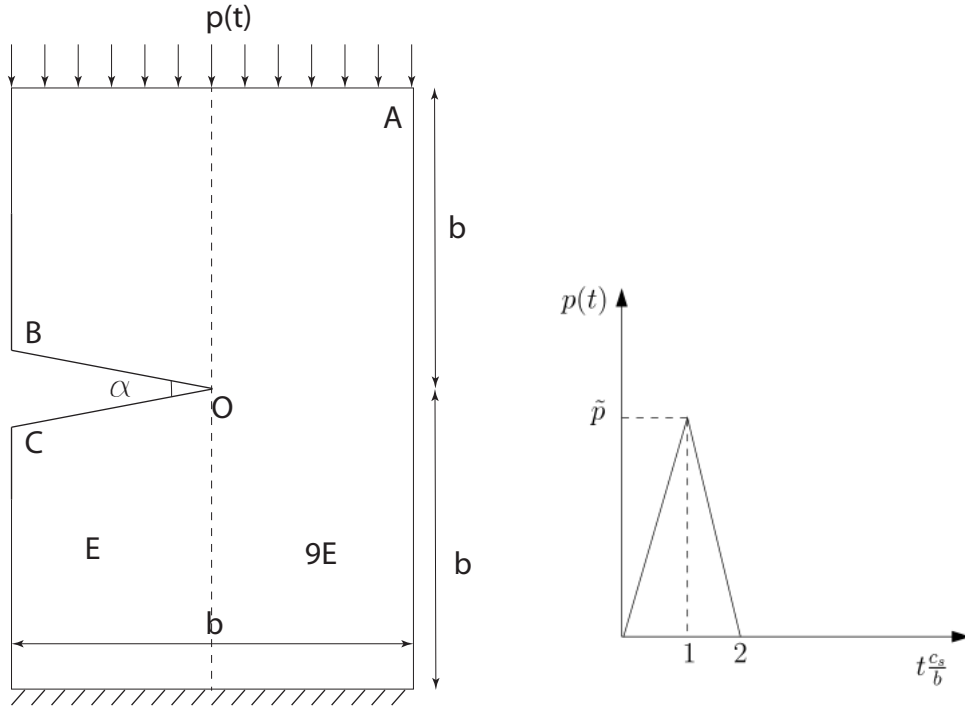
Angle	$K_{II}$			
$\alpha$	(Banks-Sills et al., 2005)	Order of the curve		
		2	4	6
0	0.0000	0.0000	0.0000	0.0000
$\pi/12$	0.2912	0.2914	0.2913	0.2913
$\pi/6$	0.5092	0.5087	0.5093	0.5092
$\pi/4$	0.5807	0.5948	0.5946	0.5946
$\pi/3$	0.5248	0.5247	0.5248	0.5248
$5\pi/12$	0.3108	0.3119	0.3117	0.3117

#### 3.7.7 Transient analysis of bimaterial plate with a notch

In this last example, we study the transient response of a bimaterial plate with a notch terminating at the material interface. The crack is assumed to be in material 1 and terminating at the interface between the two materials (see Fig. 3.19). The length of the crack is assumed to be  $b/2$  and the width of the notch angle is  $\theta = 20^\circ$ . The inputs and the results are non-dimensional. The plate consists of two materials with different Young's modulus and the same mass density and Poisson's ratio  $\nu = 1/3$ . A state of plane stress condition is assumed and the plate is fixed at the bottom.

The main advantage of the proposed method when applied to this example is that, no special treatment is required to represent the weak material discontinuity or the strong discontinuity due to the notch. The scaling center is placed at the point  $O$ , where the notch intersects the ma-

### 3.7. NUMERICAL EXAMPLES

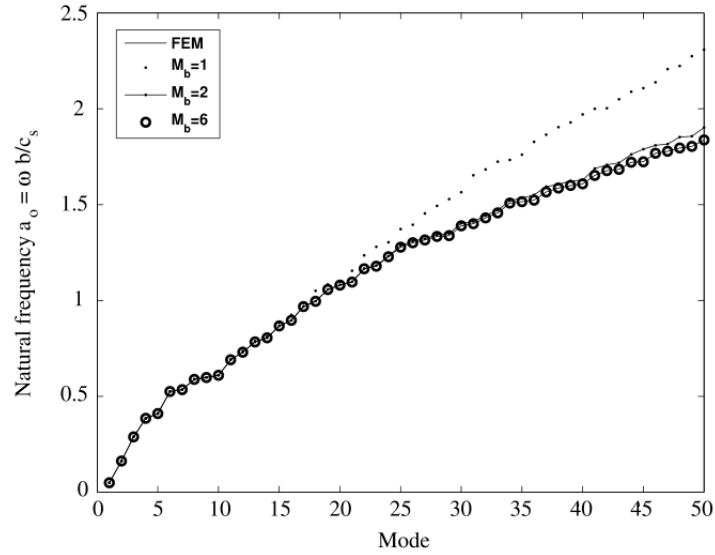


**Figure 3.19:** Bimaterial plate with a notch: geometry, boundary conditions and force history

terial interface. The SBFEM does not require the faces of the notch to be discretized and only the boundary of the domain is discretized. For this study, the vertical boundaries are discretized using four quartic NURBS functions and the horizontal boundaries are discretized using two quartic NURBS functions. Fig. 3.20 shows the non-dimensional natural frequencies computed using the proposed method for different orders of continued fractions. The reference solution is computed by using the commercial software ANSYS® 14.0. It is seen that the results from the present approach are in good agreement with the results from the FE software and for the transient analysis, the order of continued fractions is chosen as  $M_{cf} = 6$ .



### 3.7. NUMERICAL EXAMPLES



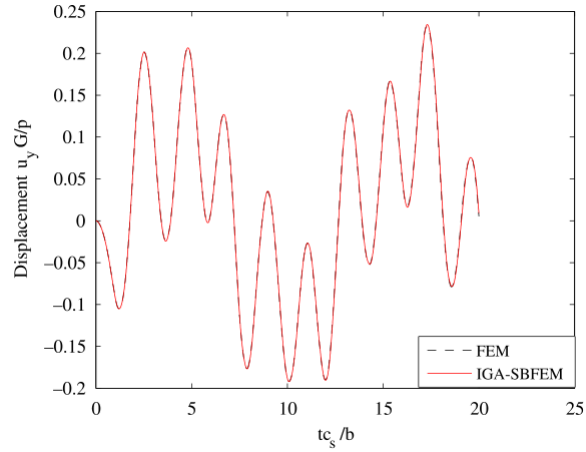
**Figure 3.20:** Non-dimensional natural frequencies of a bimaterial plate with a notch

The transient response of the bimaterial plate with a notch is evaluated considering a uniformly distributed load acting at the top horizontal boundary (see Fig. 3.19). A triangular impulse load with an amplitude  $\tilde{p}$  with a duration of  $2 \times \frac{b}{c_s}$  and peak value  $\tilde{p}$  at time  $\tilde{t} = 1 \frac{c_s}{b}$  is applied. The time integration is carried out by employing Newmark's method with  $\gamma = 1/2$  and  $\beta = 1/4$ . The time step is chosen as  $\Delta t = 0.05 \frac{b}{c_s}$  and a total of 400 time steps are considered. The vertical displacement responses at points  $A$ ,  $B$  and  $C$  are plotted in Fig. 3.21 as functions of the dimensionless time  $\tilde{t} = c_s \frac{t}{b}$ . The corresponding results of a finite element analysis are also shown for comparison and an excellent agreement is observed. The main advantage of the proposed method is that internal discontinuities or material discontinuity does not require special treatment as in other approaches such as the XFEM or the conventional IGA. Moreover, creating a scaled boundary mesh with NURBS or B-spline functions is straightforward as

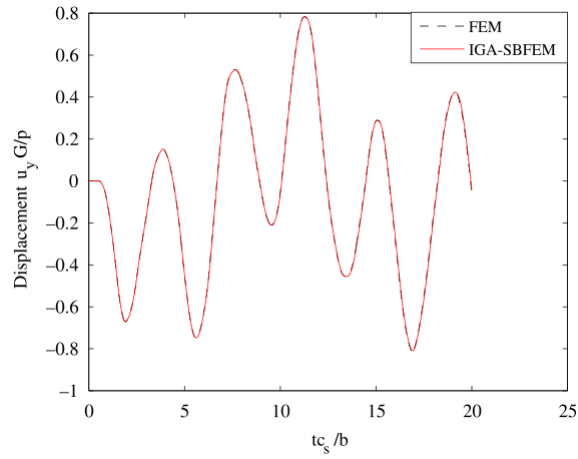
### 3.8. CONCLUSIONS

---

only the boundary information is required when compared to generating finite element mesh around the notch.



(a) Vertical displacement at Point A



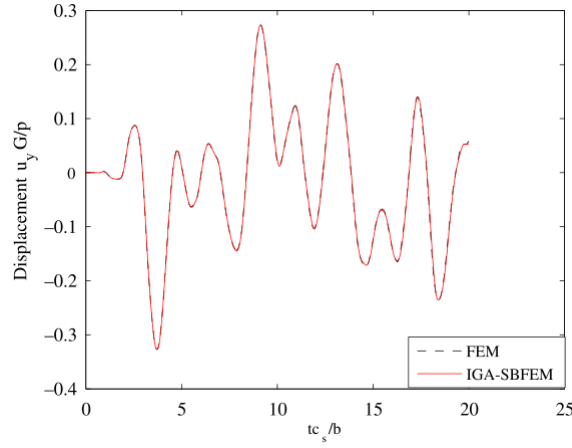
(b) Vertical displacement at Point B

## 3.8 Conclusions

In this chapter, the NURBS basis functions are employed to approximate the unknown fields in the circumferential direction within the

### 3.8. CONCLUSIONS

---



(c) Vertical displacement at point C

**Figure 3.21:** Vertical displacements of bimaterial plate with a notch at (a) point A; (b) point B; (c): point C

framework of the SBFEM. The accuracy, effectiveness and the convergence properties of the proposed method are demonstrated with benchmark problems in linear elasticity and linear elastic fracture mechanics. From the numerical studies, it can be observed that the NURBS basis functions yield superior accuracy when compared to Lagrange basis functions of the same order. The proposed method overcomes the disadvantages of both the isogeometric finite element analysis and the isogeometric boundary element method. Like in the IGAFEM no fundamental solution is required and like in the IGABEM the spatial dimension is reduced by one. However, for complicated geometries, to meet the star convexity, subdivision into smaller sub-domains is required. When applied to problems with singularities, the proposed method does not require additional functions to span the solution space. Moreover, the proposed framework does not require internal discretization to study the dynamic response at high

### **3.8. CONCLUSIONS**

---

frequencies.

# Chapter 4

## Quad-tree mesh in 2D analysis

### 4.1 Introduction

The main objective of this chapter is to implement an adaptor that can parse the geometric information in CAD directly and to develop an automatic mesh generation algorithm based on it. The IGES file introduced in Sec. 2.1.1 is used in the proposed method as the bridge between the CAD and the numerical analysis. A quad-tree mesh generation algorithm described in Sec. 4.3 will be adopted to generate a high quality mesh that meets the scaling requirement discussed in Sec. 2.2. Either the NURBS basis function or the traditional shape function can be used as the shape function of the SBFEM solver. The geometric boundary will be translated into polylines before the meshing and the intersections are calculated based on them. Intersecting points on these polylines are pro-

## 4.2. CAD OUTPUT IN 2D

---

jected back to the NURBS surface after the meshing is generated to retain the exact geometry. The proposed method enhances the conventional SBFEM and the salient features of the method are:

- Direct using design file as geometric input s
- No human effort involvement in mesh generation
- Retained exact geometry
- High quality mesh generated from the quad-tree algorithm

This chapter is organized as follows. The CAD output (iges file) will be introduced first. After that, an overview of the algorithm that can generate quadtree mesh will be provided. Furthermore, a point projection method for NURBS curve utilizing its strong convex hull property is presented. The accuracy and the convergence properties of the proposed techniques are demonstrated with benchmark problems in the context of linear elasticity, followed by concluding remarks in the last section.

## 4.2 CAD output in 2D

The IGES(IGES, 1986) files introduced in Sec. 2.1.1 are used as the bridge between the engineering design and the numerical analysis in the proposed method. As a standard file format in engineering design industry, it is supported by almost all design softwares all over the world.

## **4.2. CAD OUTPUT IN 2D**

---

Consequently, it offers a possibility to minimize the human efforts spent on geometric modeling.

### **4.2.1 Parse geometry in IGES file**

The IGES file provides all information that describe the geometric input. Abstract structure of the IGES file that describes the 2D geometry can be regarded as a simple curve-surface structure. In other words, it defines the geometric input as certain number of surfaces with their boundaries in different colors, which represents different materials in engineering practice. Each surface contains a list of curve indices which are corresponding to the curve information described in IGES file.

When the IGES file is feed into the programme, each line in the directory entry will be parsed entity by entity. Parameters in directory entry describe the type and reference to other useful values for the entity. An entity may refer to one or many entities in directory entry. Detail of the specification of each entity is explained in Nas (2007).

### **4.2.2 Output to mesh generator**

The output file that the mesh generator read in will be a short summary of the geometric input. Boundary representation will be kept as the data structure to describe the geometric input. However, some fields apart

## 4.2. CAD OUTPUT IN 2D

---

from the curves or the surfaces will be added. In the output file, the geometry will be organized in key points, polylines and surfaces. The key points define the coordinates of all points and polylines are used to represent all curves including NURBS curves for simplification. NURBS curves introduced in 2.1.2 can be used directly by the mesh generator as well but it is found that the computational cost in the calculation of the distances between points to NURBS curves surpasses the merits of using it directly. The exact geometry can be retained by projecting the nodes on the boundary back to the origin NURBS curves after the mesh is generated.

Representing a straight line with polyline can be natural, the first and the last points will be enough to achieve an exact representation. However, that is not the case for curves whose curvature is not always zero. Although adding more key points can increase the quality of polyline representation, computational cost in calculating the distance increases at the same time. Yet, having only few key points may result in a bad polyline representation which leads to a poor mesh quality. Elements may be twisted after the nodes on the boundary are projected to the origin curves. As a consequence, a quantified indicator that is able to control the number and the position of the vertices on the polyline can be necessary.



## 4.2. CAD OUTPUT IN 2D

---

### 4.2.3 Discretization of the circular arc

The chord ratio can be a good indicator for circular arc. The arc length to chord length ratio  $\frac{a}{L}$  illustrated in Fig. 4.1 can be expressed in angle  $\alpha$  as:

$$\frac{a}{L} = \frac{\sqrt{2 - 2\cos\alpha}}{\alpha} = \frac{2\sin\frac{\alpha}{2}}{\alpha} \quad (4.1)$$

The maximum angle  $\alpha$  satisfy arc length to chord length ratio  $\frac{a}{L} > 1 - \epsilon$  with  $\sin\frac{\alpha}{2} = 1 - \frac{x^3}{6} + O(x^7)$  can be derived as:

$$\alpha < \sqrt{24\epsilon} \quad (4.2)$$

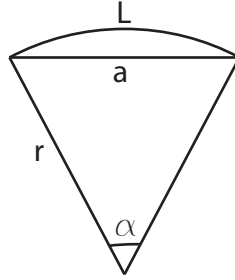
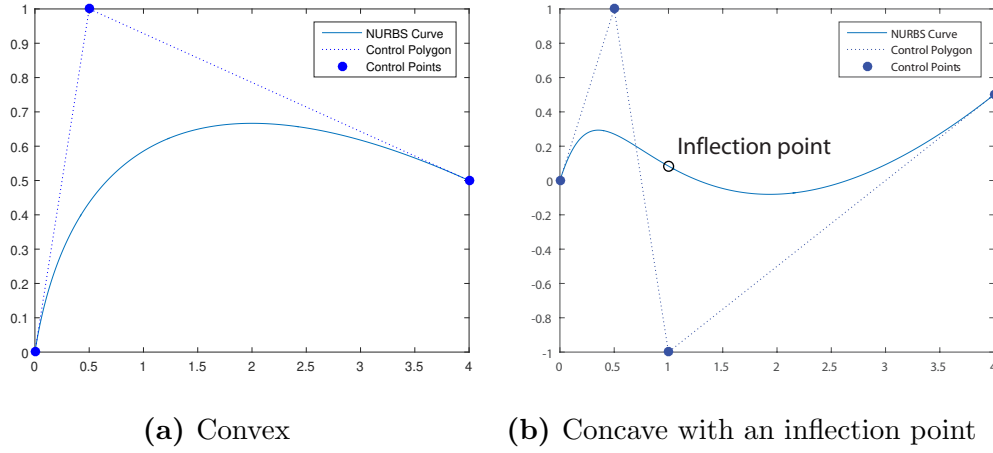


Figure 4.1: Chord length for circular arc

### 4.2.4 Discretization of the NURBS curve

For NURBS curves who have no closed form chord ratio, similar idea can be applied numerically. The NURBS curve is first divided into several smaller ones based on the knot vector as described in Sec. 2.1.2.1. Since the order of each subdivided NURBS curve used in engineering

## 4.2. CAD OUTPUT IN 2D



**Figure 4.2:** Type of sub-divided NURBS curves: convex and concave with an inflection point

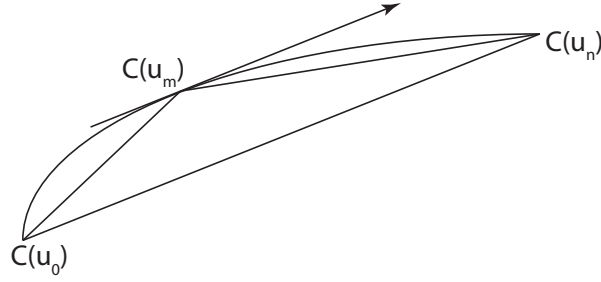
softwares are predominantly lower or equal to three, the sub-curves then can be divided into two classes, convex curves or concave curves with an inflection point as shown in Fig. 4.2. In order to determine whether the target NURBS curve is convex or concave, a cross product will be conducted. By assuming the subdivided NURBS curve is cubic, there will be four control points  $P_1, P_2, P_3$  and  $P_4$ . If the signs of  $cross(\overrightarrow{P_1P_2}, \overrightarrow{P_2P_3})$  and  $cross(\overrightarrow{P_2P_3}, \overrightarrow{P_3P_4})$  are the same, then the curve is convex. Otherwise it will be concave.

### 4.2.4.1 Convex curves

Start with the simple case, in the situation illustrated in Fig. 4.3 where line  $C(u_0)C(u_n)$  and the NURBS curve form a convex set, we are looking for a point  $C(u_m)$  on the curve so that  $C'(u_m)$  is parallel to  $\overrightarrow{C(u_0)C(u_n)}$ . The target is to split one NURBS curve segment into two. The splitting will be

## 4.2. CAD OUTPUT IN 2D

---



**Figure 4.3:** Discretization for convex NURBS curve

processed until the arc length to chord length ratio of any splitted curves are smaller than the tolerance. Based on the property of the convex set, there can be one and only one parameter  $u_m$  that satisfies the condition. As a consequence, numerical methods such as Newton's method can be adopted to determine it. For a given  $u_m$ , the next iteration will be:

$$u_{m_{new}} = u_n - \frac{f(u_m)}{f'(u_m)} \quad (4.3)$$

where

$$f(u) = C'(u) \begin{bmatrix} -C_y(u_n) + C_y(u_0) \\ C_x(u_n) - C_x(u_0) \end{bmatrix} \quad (4.4)$$

The procedure to find  $u_m$  can be concluded in Algorithm. 1 and Algorithm. 2 describes the procedure to find all knots corresponding to the vertexes of the polylines recursively.

### 4.2.4.2 Concave curves

As can be seen in Fig. 4.2, the extracted cubic NURBS curve will have no more than one inflection point. The reason for that is because

## 4.2. CAD OUTPUT IN 2D

---

---

**Algorithm 1:** Split a convex NURBS curve into two

---

```
1 function splitConvexCurve(curve,u_0, u_n)
   Input : curve, the input NURBS curve
           u_0,u_n, two end knots of the NURBS curve
   Output: u_m, in Fig. 4.3
2 u_m = (u_n + u_0) * 0.5
3 pt_0, pt_n = getCurvePts(u_0, u_n)
4 vector_0n = Vector(pt_0,pt_n)
5 angle = vector_0n.atan2()
6 deri1_m = getCurveDeri(u_m,1)
7 angle_m = deri1_m.atan2()
8 while  $abs(angle\_m - angle) < 10^{-4}$  do
9   deri1_m, deri2_m = getCurveDeri(u_m,2)
10  angle_m = atan2(deri1_m.y, deri1_m.x)
11  fu = deri1_m * vector_0n.normalVector()
12  fu_der1 = deri2_m * vector_0n.normalVector()
13  u_m = u_m - fu / fu_der1
14  deri1_m = getCurveDeri(u_m,1)
15  angle_m = deri1_m.atan2()
16 end
```

---

---

**Algorithm 2:** Discrete a convex NURBS curve recursively

---

```
1 function discreteConvexCurve(curve,eps,u_0,u_n,u)
   Input : curve, the input NURBS curve
           eps, the tolerance of the chord to arc-length ratio
           u_0,u_n, two end knots of the NURBS curve
   Output: u, the vector of the knot corresponding to vertexes of the
           polylines
2 arcLength = curve.arcLength(u_0,u_n)
3 chordLength = curve.getPt(u_0).distanceTo(curve.getPt(u_n))
4 if  $1 - chordLength / arcLength < eps$  then
5   | return
6 else
7   | u_m = (splitConvexCurve(curve,u_0,u_n)) u.add(u_m)
   | discreteConvexCurve(curve,eps,u_0,u_m)
   | discreteConvexCurve(curve,eps,u_m,u_n) return
8 end
```

---

## 4.2. CAD OUTPUT IN 2D

---

the target function is cubic and hence its second derivative will be in first order. Consequently, numerical methods such as Newton's method can be used to find this unique point. After that, the curve can be divided into two convex ones and the algorithms introduced in 4.2.4.1 can be used separately. The Newton's iteration can be written as

$$u_{n_{new}} = u_n - \frac{f(u_n)}{f'(u_n)} \quad (4.5)$$

where

$$f(u) = C''(u) \quad (4.6)$$

### 4.2.4.3 Calculation of the arc length

The arc length of the NURBS curve defined on  $u \in [u_0, u_n]$  can be expressed as

$$L = \int_{u_0}^{u_n} \sqrt{C_x^2(u) + C_y^2(u)} du \quad (4.7)$$

The integration can be solved by the help of the numerical integration quadrature described in 3.6 as:

$$L = \sum_{i=0}^n a_i \sqrt{C_x^2(u_i) + C_y^2(u_i)} \quad (4.8)$$

### 4.3. QUAD-TREE STRUCTURE

---

## 4.3 Quad-tree structure

After the geometry information is exported from the IGES file, it can be feed into the quad-tree algorithm to generate mesh of the problem domain. As an algorithm based on computational geometry, it requires great amount of numerical operations and hence the result may be sensitive to the tolerance. An absolute tolerance may not be able to handle problems with very large or small geometric size. As a consequence, the first step is to normalize the geometry into a uniformed space ( $10 \times 10$  is used in this chapter).

### 4.3.1 Background mesh

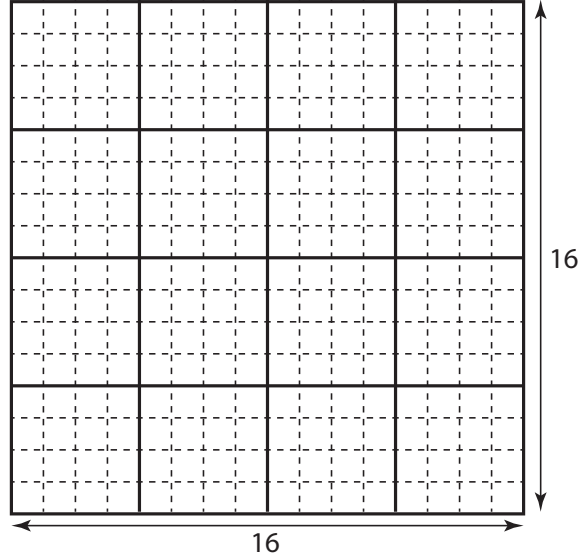
Background mesh describes a mesh in the background. Its density is decided by the geometry. This section will introduce the procedure to generate the background mesh.

First of all, we start with one square which is the root of the tree. The size of it will be slightly larger than the normalized geometric input and it is selected as  $16 \times 16$  in this chapter. An adaptive mesh generation based on numerical error can be found in Sec. 5. After that, the root square will be divided into millions (defined by resolution, defined as  $2^{res} \times 2^{res}$ ) smaller ones like pixels in the image. Then,  $2^{s_{max}} \times 2^{s_{max}}$  “pixels” will be group into the first level of the tree, or initial background mesh as shown in Fig. 4.4. It

### 4.3. QUAD-TREE STRUCTURE

---

is used to control the maximum allowable mesh size globally or separately for different material regions.



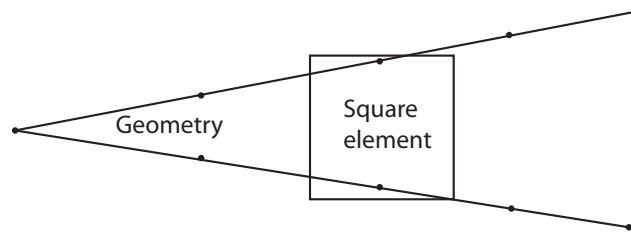
**Figure 4.4:** An example of the background initial mesh:  $16 \times 16$  square are divided into  $2^4 \times 2^4$  pixels (dashed lines,  $res = 4$ ) and  $2^2 \times 2^2$  pixels form the initial mesh (solid lines,  $s_{max} = 2$ )

The criteria to decide whether each individual square in the initial mesh need to be refined or not is based on the concept of seed point  $s$ . The curve will be uniformly discretized into a given number so called seed points uniformly and the mesh will be refined until the number of the seed points within the square is less than the threshold. However, finer mesh is expected at the region where geometry with high curvature appears. It can be solved by treating each segment of the polylines as an individual curve. Due to the fact that algorithm described in Sec . 4.2.2 guarantees the chord length to arc length ratio, polylines ought to have finer segments at the position where the curvature is significant.

### 4.3. QUAD-TREE STRUCTURE

---

Although seed points provide a good guide on the mesh density, situations where high density mesh is required while few seed points appeared may happen as plotted in Fig. 4.5. The geometry limits the seed points due to the lack of curvature. While, it is expected that the square element will be refined at least once as one layer of mesh may not be appropriate to formulate a thin shell structure. As a result, another restriction



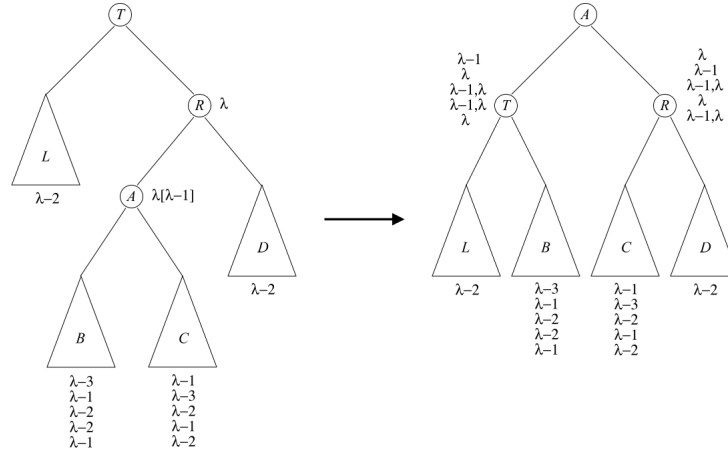
**Figure 4.5:** Limitation of the seed points: few seed points will be generated over a straight line and few seed points will be included in the square element which leads to unexpected behavior. ● stands for the seed points.

will be adopted together with the seed points to prevent this kind of situation from happening. Element with more than two unique intersections will be tagged as “need refinement” no matter how many seed points it contains. In numerical calculation, two points may be regarded as one if they are close enough to each others. Normalization of the geometry described at the beginning of this section helps to define a meaningful tolerance to handle numerical error.

Self-balancing is adopted by most of the tree data structure such as AVL, B/B+, Red-black tree and so on. Fig. 4.6 illustrates a self-balancing operation of the AVL tree. Balancing by rotation is performed because difference in height of the leaf B and L is greater than the threshold. Same

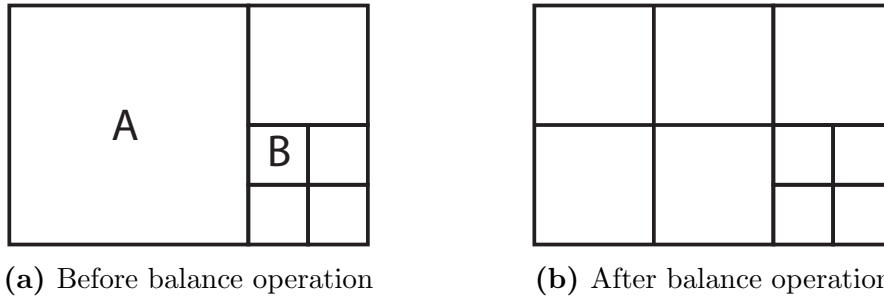


### 4.3. QUAD-TREE STRUCTURE



**Figure 4.6:** Balance of the AVL tree (Roura, 2013)

idea is adopted in the quadtree as well. A refinement will be performed to achieve the balance if the difference in the height of the leaf (Cell A and B in Fig. 4.7 for example) is larger than one. The reason why balanc-



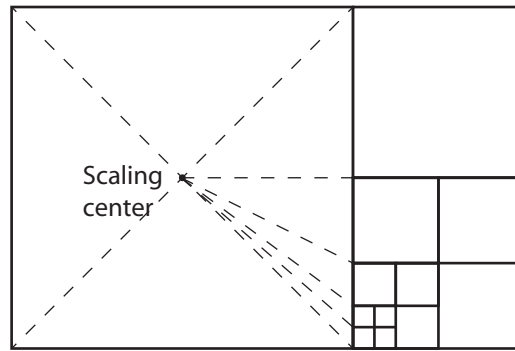
**Figure 4.7:** Balance of quadtree: cell A is refined in order to balance the quadtree.

ing is predominately adopted in tree data structure lies in the guarantee of an  $O(\log(n))$  time complexity for searching in any case. Even though computational cost on searching seems not to be significant during mesh generation using quadtree, a balanced tree provided some other attractive features that can be utilized in numerical analysis. One of the advantages is to improve the mesh quality. Any extremely small or large angle

### 4.3. QUAD-TREE STRUCTURE

---

between the element and the scaling center may result in a bad quality mesh. Chances are that these poor quality mesh may appear without self-balancing, shown in Fig. 4.8. Another reason for that is to kept the pattern

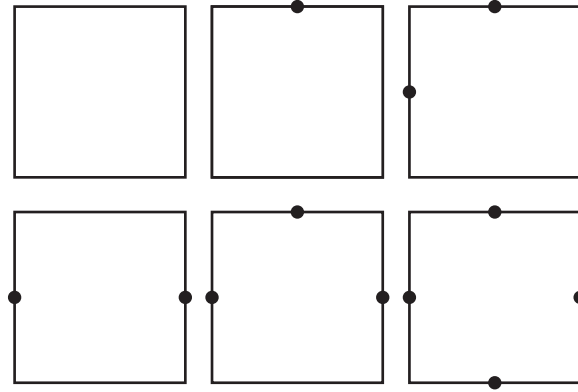


**Figure 4.8:** Small angle between element and scaling center may reduce the mesh quality

of the cells. If the threshold of the self-balancing is set to one (2 : 1 ratio), only six kinds of the cells will appear before the cutting as illustrated in Fig. 4.9. Thanks to the geometric similarity, local stiffness matrix can be calculated and scaled directly when same kind of the cell appears which significantly reduce the computational cost. Hanging nodes in Fig. 4.9 can be a problem for traditional finite element to handle the displacement compatibility (Ebeida et al., 2010; Legrain et al., 2011; Tabarraei and Sukumar, 2008). Solution including triangulation (Bern et al., 1994; Koçyiit et al., 2009; Yerry and Shephard, 1983), using special shape function (Gupta, 1978) and other methods are available but special treatment s is required. As a comparison, SBFEM provides greater flexibility on the element, n-sides polygons with hanging nodes or curved edges can be treated natively.

### 4.3. QUAD-TREE STRUCTURE

---



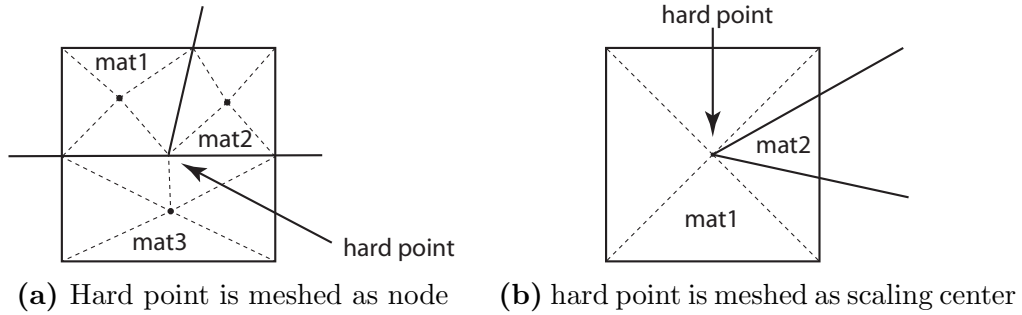
**Figure 4.9:** Types of the cell when 2 : 1 ratio is applied, ● stands for the hanging node

#### 4.3.2 Hard point treatment

A Hard point is a kind of point in the geometry that must be meshed as a node or scaling center. In the proposed method, a concave element may only be observed with the appearance of a hardpoint. Due to the fact that the scaling requirement discussed in Sec. 2.2 can be easily met for all convex polygons simply by adopting their centroids as the scaling center, the hard point treatment could be crucial in order to generate a valid SBFE mesh. When more than two materials are involved, it is common to have some hard points to make sure the point shared by three material can be properly formulated as shown in Fig. 4.10 The difficulty in treating a hard point lies in its position in the background mesh. The further it is away from the geometric center of the background mesh, the poorer the quality of mesh will be generated. If the hard point in Fig. 4.10b is located somewhere that is very close to the left boundary of the background mesh, the mesh for material one after cutting may be quite elongated and

### 4.3. QUAD-TREE STRUCTURE

---



**Figure 4.10:** Example of a hard point, elements around the point shared by three material must be properly divided into three.

twisted. Besides, the requirement of the scaling center in the SBFEM can always be satisfied no matter where it is located in a convex polygon. The opposite is true for a concave polygon, special treatment must be adopted to fulfill this requirement. As a consequence, generally speaking, a convex mesh may be preferred over a concave one and hard point can be the only source that will introduce concave polygons in most of the situations. Although algorithms finding qualified scaling center in a concave polygon exist, the quality of the mesh may not be satisfactory even the scaling center is located on the vertex.

The first step to treat the hard point will be trying to adjust the background mesh so that the hard point is close enough to the cell's geometric center in the background mesh. The ideal size of the background mesh shall ensure that only one hard point is contained and that no points from any other curves should be located in it. No more than one reflex angle is expected if the maximum number of the hard point in the cell is limited to one, under which circumstance the scaling requirement discussed in Sec.

### 4.3. QUAD-TREE STRUCTURE

---

2.2 can be easily met by placing the scaling center either at the hard point or on the bisector of the reflex angle. As a consequence, the size of the containing square  $box\_size$  is set to be one third of the distance to the nearest curves or half of the distance to the nearest hard point, whichever is smaller. These parameter usually result in a valid and large enough background mesh that can treat the hard point easily. When building the background cell containing hard points in the algorithm, it will be implemented by having a considerably fine mesh within the range of the hard point and merge all cells in that range into a larger one to be the background cell. Size of the “considerably fine” mesh  $size\_field$  will be defined with the adjacent vertexes of the hard point derived in Eq. 4.9

$$size\_field = \frac{box\_size}{2^{round(\log(\frac{2\pi}{min(\alpha)})-1)}} \quad (4.9)$$

where  $\alpha$  is the minimal angle  $min(\alpha_1, \alpha_2, \dots, \alpha_n)$  in Fig. 4.11. As is exponentially related to the minimal angle  $\alpha$ , the  $size\_field$  may always be small enough to capture thin shell. After the first step of the hard point treatment is done, the background cell shall have the following properties

1. Distance between hard point to geometric center of the background cell must be smaller than  $\sqrt{2}size\_field$
2. Element after cutting share the hard point as the node or it will be one element with scaling center located at the hard point

With these two properties, the cell can be cut by simply connecting the intersections with the hard point later.

### 4.3. QUAD-TREE STRUCTURE

---

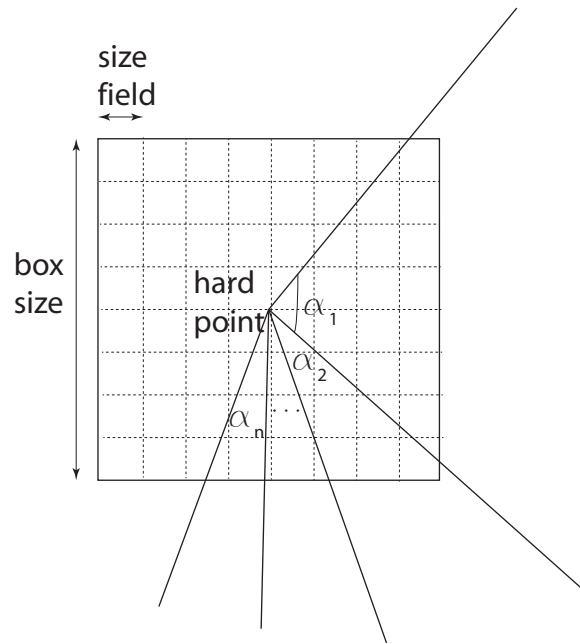


Figure 4.11: Hard point treatment step1: find *box\_size* and *size\_field*

#### 4.3.3 Bucket sort algorithm

When cutting the boundary, it is necessary to find the intersections between the polylines and the background mesh. Finding geometric relationship between each cell and each curve may be necessary. However, it could be computational expansive when quite a few curves are involved and a fine mesh is targeted. As a result, an algorithm inspired by buck sort is introduced in order to reduce the computational cost of this process.

Bucket sort algorithm is a sorting algorithm that put an array of objects into different buckets and then sort the objects in each bucket. Objects in each bucket then are sorted by other sorting algorithm or by applying the buckets sort recursively. Although the sort algorithm can not

### 4.3. QUAD-TREE STRUCTURE

---

be adopted directly in finding the intersection, the idea “divide and conquer” can be utilized in order to reduce the computational cost in finding the intersections.

First step of this method would be very similar to the first step in generating the background mesh. The entire background square or the root of the quadtree will be divided evenly into  $2^n \times 2^n$  smaller ones. A larger  $n$  will result in an increasing number of buckets and will detect more irrelevant curves but take more time. A trade off normally is not necessary and  $n = 8$  usually provides a good enough result so it can be fixed as 8 unless special need is posed on. Fig. 4.12 illustrates an example of the algorithm ( $n$  is taken as 4 in this example in order to have a clear figure, it should usually be taken as no less than 6). The end points of the curve are located in grid D2 and I5, as a consequence, the curve will be put in bucket “D2-I5”. In other words, cells that are not in this region shall not have any intersections with the curve.

#### 4.3.4 Cutting with boundary

Generally, there are three situations where cutting is involved with one curve and they are sorted by the number of the intersections as shown in Fig. 4.13. Due to the fact that the background cell is refined properly and hence the curves can be regarded as straight lines. Case 3 in Fig. 4.13c is an exception as it shall not happen unless even the background cell

### 4.3. QUAD-TREE STRUCTURE

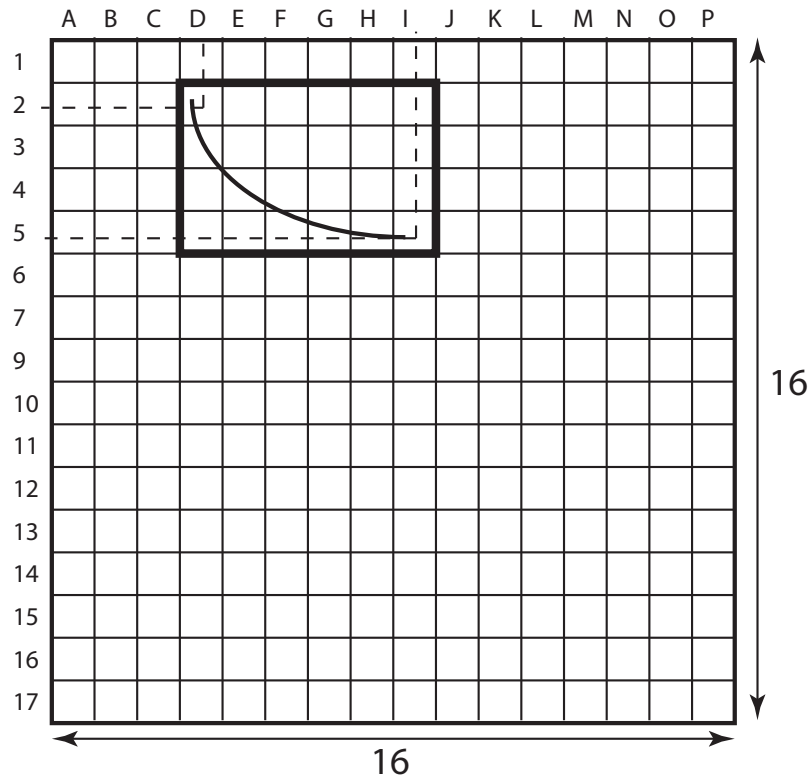


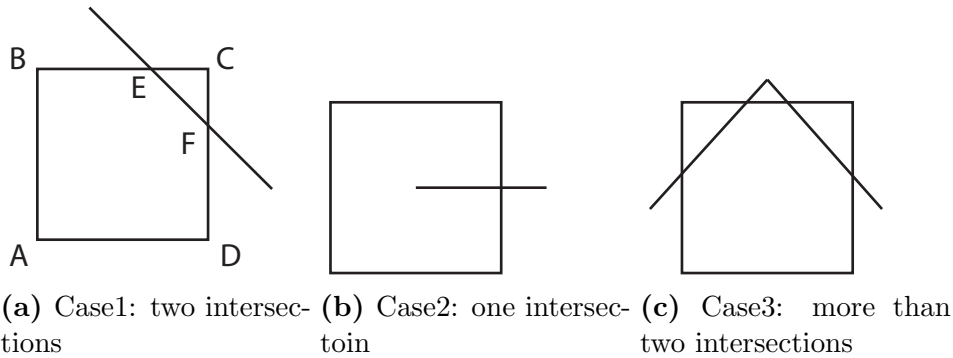
Figure 4.12: Bucket sort in quadtree

with the minimal allowable size may not be able to catch the feature. Unfortunately, result from case 3 may not be as expected but increasing the resolution or decreasing the minimal allowable cell size can help to prevent case 3 from happening. In situation a (Fig. 4.13a) where two intersections are observed, cell can be split by connecting points  $EF$ . To start with, any vertex among  $A$ ,  $B$ ,  $C$  or  $D$  can be selected as the starting point. After the point is selected, point  $A$  for example, another point from any of the line connected it will be chosen, point  $B$  for line  $AB$  for example. This procedure will be repeated recursively until the intersection points  $E$  or  $F$  is found and next point will be  $E$  or  $F$  whichever is not selected at the moment. Recursion will be continued until the starting point is found again. A

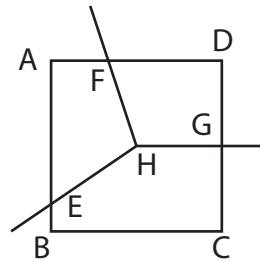


### 4.3. QUAD-TREE STRUCTURE

---



**Figure 4.13:** Situations in cutting: one, two and more than two intersections



**Figure 4.14:** Hard point treatment in cutting

closed region then will be formed by all selected points in sequence. Another cell can also be easily found by finding the implement set. It should be noted that all cells split in this situation must be convex.

In case 2 where only one intersection will be detected, multiple curves with the intersections should be observed as in Fig. 4.14 and the hard points must be involved (point  $H$  in Fig. 4.14). Handling case 2 would be similar but slightly tricky compared to that in case 1. At the beginning, the hard point is selected (point  $H$  in the example). After that, any intersection point is selected ( $F$  in this example). Then, the other point of any line segment that has  $F$  will be selected ( $D$  in this example). The former

### 4.3. QUAD-TREE STRUCTURE

---

step will be performed recursively until another intersection point is found. Once the hard point is detected again, a closed region is found. Repeating the procedure can help to find all closed regions cut by the hard point and multiple curves.

In case 3 where more than two intersections are found, split cells may not be as expected as this situation shall not happen normally. If case 3 does happen, the first and the last intersection points will be selected and all other intersection points will be ignored. After that, method in case one can be applied directly.

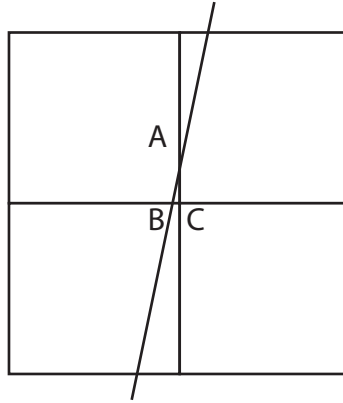
#### 4.3.5 Points moving

Although all types of cells with intersections are able to be treated by the algorithm described above, quality of the mesh sometimes may become out of control as example shown in Fig. 4.15.

In order to improve the mesh quality, the point moving is adopted. Generally there are two ways of moving, one is moving the closest points to the line (move point  $C$  to line  $AB$  in Fig. 4.15) and another is moving points on the line to the closest point, e.g. point  $B$  will be moved to  $C$  in Fig. 4.15. Both methods has its own advantages and disadvantages and they will be discussed later.

### 4.3. QUAD-TREE STRUCTURE

---

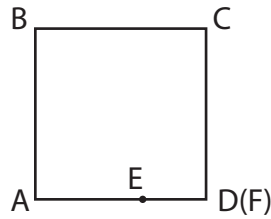


**Figure 4.15:** Poor quality of mesh: triangle element ABC is too small compared to its neighbours

The recommended method is to mark the nodes that are close enough to the curves as intersections and project them back onto the curves at the end. One significant advantage is that only convex cells may appear during cutting but special treatment needs to be added in generating the background mesh. The algorithm for cutting a convex cell into two will always yield valid results in existing algorithms. For example, point *C* in Fig. 4.15 will be marked as the intersection and points *A* and *B* will be ignored. Point *C* will be projected back using the algorithm introduced in Sec. 4.4 to the curve when the cutting is finished. The threshold usually is taken as a proportion to the length of the background cell size. When finding the intersection, another detection will be posed after. If the distance of the intersection to a node is smaller than  $\epsilon a$  where  $\epsilon$  is tolerance and  $a$  is the size of the background cell, the intersection will be ignored and the nearest node will be marked as intersection instead. Fig. 4.16 illustrates an example when points are moved toward one of the nodes. If both intersections are moved then cutting may not be necessary. When one is moved and another is

### 4.3. QUAD-TREE STRUCTURE

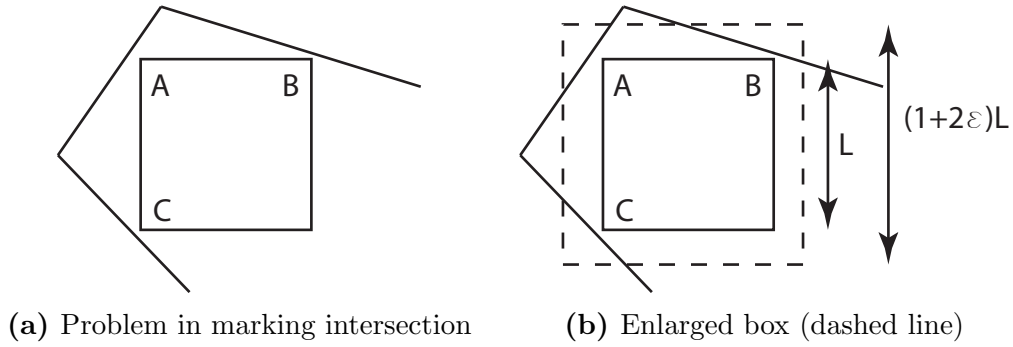
---



**Figure 4.16:** Treatment of cell cutting when point is moved

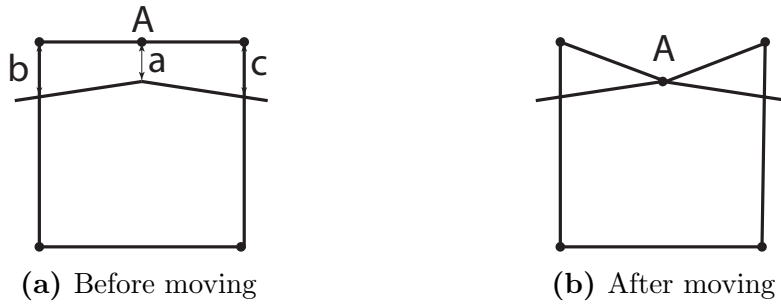
retained, algorithm described in previous section can be adopted directly as the topology remain the same. The disadvantage is that situation in Fig. 4.17a when generating the background mesh need to be taken into consideration. Points  $A$ ,  $B$  and  $C$  will be marked as intersections and it becomes situation 3 in Fig. 4.13c. However, background cells like it shall be refined otherwise unexpected behavior resulted from cutting algorithm may appear and such unexpected behavior shall be only allowed to happen when minimal allowable cell size in setting is too large to capture the local feature. As a consequence, when refining background mesh in section. 4.3.1, not only the number of intersections between curves with the origin cell need to be counted, but also that from the enlarged box need to be counted as shown in Fig. 4.17. Either more than two intersections are found in origin cell or the enlarged one in Fig. 4.17b will result in the refinement of the cell. Another approach is projecting the nodes in background mesh on to the curves before the cutting using algorithm introduced in Sec. 4.4. Special treatments in generating the background mesh is no longer necessary but cutting concave cells may be hard to treat. Fig. 4.18 illustrates an example of a concave cell resulted from point moving. An intersecting curve that is not a straight line (even with a very small curva-

### 4.3. QUAD-TREE STRUCTURE



**Figure 4.17:** Adopting enlarged box to ensure the background mesh is refined properly

ture) leads to a possibility that the distance  $a$  from the hanging node  $A$  to the curve is slightly smaller than the tolerance while  $b$  and  $c$  are slightly larger than tolerance. As a result, point  $A$  will be moved on to the curve



**Figure 4.18:** Concave cell resulted from point moving

but the other two nodes will not and hence cutting a concave cell need to be treated. This method is not recommended because special treatment on cutting concave cells need to be implemented and it could be much more complicated than that in previous method. First of all, more than two intersections will be observed in Fig. 4.18b and it may be mixed with case 3 in Fig. 4.13c. Moreover, the concave cell may be cut into three instead of two which creates extra difficulty. As a consequence, the sequence of all

### 4.3. QUAD-TREE STRUCTURE

---

intersections may need to be recorded in order to distinguish it from case 3 in Fig. 4.13c and special treatment in cutting concave into more than three parts need to be implemented as well. Generally speaking, this approach may be much more complex than solving background mesh issue when moving points before cutting.

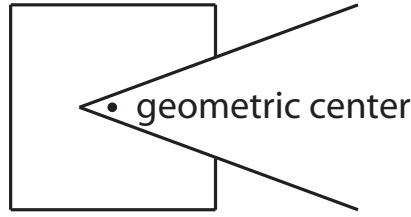
#### 4.3.6 Color the region

Multiple material properties and holes may appear in the input and the programme is expected to assign each mesh with correct material id, or color. Each different material will be related to one color and the hole can be regarded as void material. There are two ways to achieve this goal: method based on geometry and that based on topology. Using geometric location to find the color is easy to implement and not sensitive to the correctness of the mesh but may introduce errors in coloring. While using topological property requires more computational cost and is sensitive to the validity of the mesh but usually provide correct coloring if former steps are correct. Considering the possibility of coloring becoming the bottle neck of the whole programme is low and the fact that the mesh is expected to be valid, topological property may be preferred in most of the cases.

Geometric locations provide a good reference of the color in most of the case but may fail in some situation. Calculating geometric center s

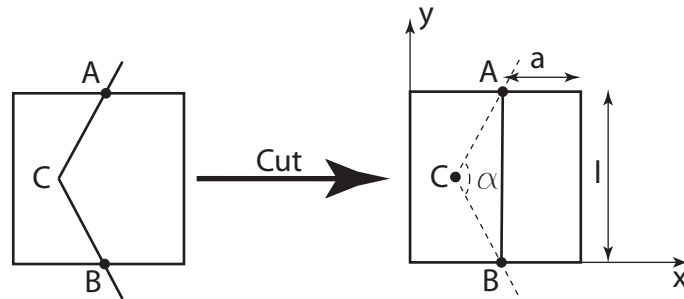
### 4.3. QUAD-TREE STRUCTURE

and finding which bounded area contains it is a computational cheap work and can be parallelized to utilize multiple threads or map-reducer using powerful clouding computing. However it can fail in some cases especially for a concave polygon whose geometric center is outside of itself as shown in Fig. 4.19. Using the scaling center which is always inside of the polygon



**Figure 4.19:** Coloring problem using location of geometric center in concave polygon

instead of geometric center can not prevent it from happening. Fig. 4.20 discusses the possibility of coloring failure in convex cell where the geometric center is coincide with scaling center. The coordinate of the scaling center of the rectangle in the left is  $(\frac{l-a}{2}, \frac{l}{2})$ . If the x coordinate of point  $C$  is less than  $\frac{l-a}{2}$ , angle  $\alpha$  is so significant that the point  $C$  is not recognized as the hard point and  $a$  is not so large compared to  $l$  that the point  $A$  and  $B$  will not be merged to the node of the rectangle. In other words, coloring



**Figure 4.20:** Coloring problem using location of scaling center

### 4.3. QUAD-TREE STRUCTURE

---

failure may happen if

$$\begin{aligned} C_x &\leq \frac{l-a}{2} \\ \alpha &> \epsilon_1 \\ \frac{l-a}{l} &> \epsilon_2 \end{aligned} \tag{4.10}$$

where  $C_x$  refer to the  $x$  coordinate of the point  $C$ ,  $\epsilon_1$  is the tolerance used in hard point definition and  $\epsilon_2$  is the tolerance used for point moving in Sec . 4.3.5. Assuming the point  $C$  is located on the geometric center of the rectangle in the left, or  $C_x = \frac{l-a}{2}$ , it is concluded that

$$\begin{aligned} 2 \arctan \frac{l}{l-a} &> \epsilon_1 \\ \frac{l-a}{l} &> \epsilon_2 \end{aligned} \tag{4.11}$$

Result of minimal  $\epsilon_1$  with different  $\epsilon_2$  is summarized in table. 4.1. Setting  $\epsilon_1$  greater than  $150^\circ$  may result in large amount of unexpected hard points and a  $\epsilon_2 > 0.35$  could lead to significantly increased points moving. A combination of  $\epsilon_1 = 150^\circ$  and  $\epsilon_2 = 0.3$  seems to be the only possible solution. However, sticking on the parameters that produces some unexpected hard points and points moving may not surpass the advantage of this method.

**Table 4.1:** Minimal  $\epsilon_2$  with different  $\epsilon_1$

$\epsilon_1$ (Deg)	100	110	120	130	140	150	160	170
$\epsilon_2$	0.84	0.70	0.58	0.47	0.36	0.27	0.18	0.09

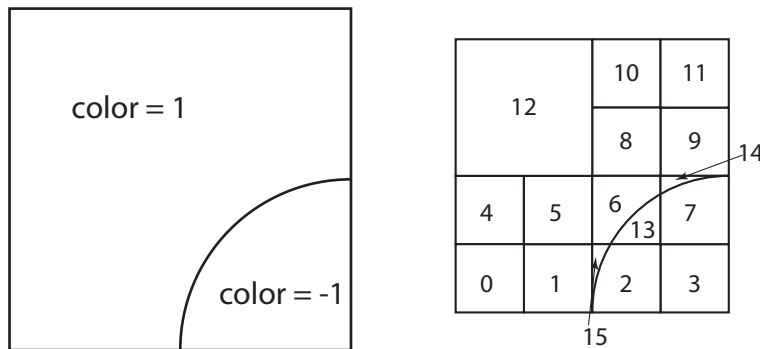
A method based on the topology, on the other hand, provides correct coloring in theory if the mesh generated is valid. In other words, the mesh satisfies the following requirements:



### 4.3. QUAD-TREE STRUCTURE

- The background mesh shall be refined properly so that situation in Fig. 4.13c will not appear
- The Hard points are set properly
- The input has been properly scaled so that there is a few outer margin

The coloring algorithm will utilize a breadth-first search algorithm to find color for each of the cell. Any polygon can be picked as the root element of the search tree. For convenience, the most bottom-left element is defined as the root of the search tree. All neighbor polygons will be added as child nodes recursively until a cell whose boundary is shared by another color is found. Once such a cell is found, the algorithm then consider the whole branch share the same color. The loops stops when there are not any unvisited polygons found under all available parent nodes. Any uncolored polygons will have the same color as its youngest parent's. Take Fig. 4.21 as an example, The root element will be the polygon 0 and its neighbor



(a) Geometry of an example for coloring (b) Mesh of an example for coloring

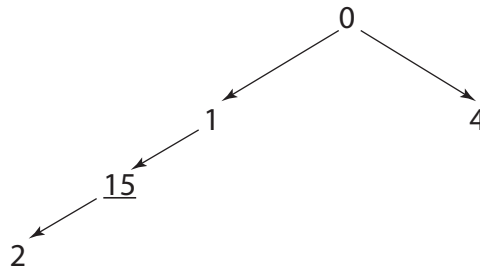
**Figure 4.21:** Example for coloring

polygons 1 and 4 will be appended as child node. Since no boundary of polygon 1 and 4 is the border between two materials, the recursion goes

### 4.3. QUAD-TREE STRUCTURE

---

on. The polygon 15 then will be found as the child node of the polygon 1, the recursion will pause before polygon 5 being append to polygon 1. The hypotenuse of the polygon 15 is the border between the color 1 and  $-1$ , considering the fact that the known color of its youngest parent (the polygon 0) is 1, whole branch (the polygon 0, 1 and 15) will be marked as color 1. By knowing the hypotenuse of the polygon 15 can only be contained in one other polygon, next child node, polygon 2, will be marked as color  $-1$  in next recursion. The recursion resumes for the polygon 1 and



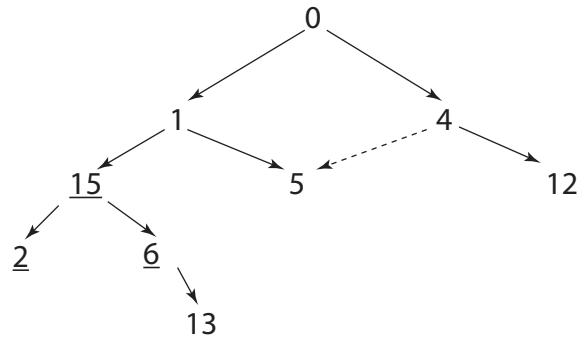
**Figure 4.22:** Example for coloring: step 1

the polygon 5 shall be appended and the polygon 4 find 5 and 12. Due to the fact that the polygon 5 is visited by the polygon 1, it will not be added to the polygon 4's child. After that, the polygon 15 find the polygon 2, whose bottom boundary is the border between the color 1 and  $-1$  (background color is set as 1 and hence it is assumed color outside of the bonding box is 1). The oldest node with known color is itself, the polygon 2 and as a result the branch with single element will be marked as color  $-1$ . Bottom edge of the polygon 2 is contained in itself only, which implies the termination of the branch. As another child node of polygon 15, the polygon 6, whose hypotenuse is border between two colors and the youngest parent with known color will be polygon 15 which is marked as color 1. The polygon

### 4.3. QUAD-TREE STRUCTURE

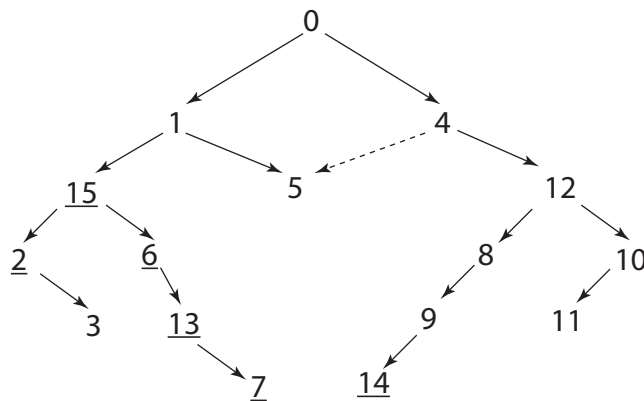
---

13 will be marked as color  $-1$  in next recursion. Fig. 4.24 illustrates the



**Figure 4.23:** Example for coloring: step 2

final search tree after a few more recursion. End node polygon 11 and 3 do not trigger the termination process normally will be marked with color in the end with the youngest parent rule, which gives branch 11, 10, 12, 4 and branch 3 the same color as polygon 0 and 2.



**Figure 4.24:** Example for coloring: final tree

## 4.4 Points projection

### 4.4.1 Projection algorithm

A point projection algorithm is to find the nearest point in parameter  $(u)$  on the NURBS curve of the test point. In the proposed method, all points on the NURBS curve is generated based on the approximated polylines and hence are not exactly on the boundaries. Although there exists a closed form solution for point projection, it require the order of the NURBS curve must be less than 4 (Piegl and Tiller, 1997). As a consequence, a projection algorithm (Ma and Hewitt, 2003) using Newton-Raphson method is introduced to tackle this problem.

For a given point  $P = (x, y)$ , its projection on the curve  $C(u)$  so that the distance  $|P - C(u)|$  is minimum is targeted. However, in the proposed method, the existence of the large number of the possible curves increases the computational cost significantly. The projection point for the test point  $P$  need to be determined for every existing curve and the one with the smallest minimum distance will be selected. One possible improvement could be limit the possible curves to only a few by utilizing the fact that the NURBS curves has been divided into multiple sub-curves without interior knot by knot insertion introduced in Sec. 2.1.2.1. Another property that can be utilized is that most of the test point  $P$  is expected to be extremely close to its projection on the curve  $C(u)$ .

#### 4.4. POINTS PROJECTION

---

As a consequence, the strong convex hull property can be adopted to limit the number of possible curves to be less than 2. The building of the convex hull is explained in detail in Sec. 4.4.2. The signed distance  $s$  of the test point to all curves' convex hulls are calculated and only the curves with negative signed distance  $s$  which indicate that the point is in the convex hull will be selected as candidates. If no negative distance is detected, a few number (taken as 3 in the proposed method) of curves with minimum signed distance will be selected.

In order to find the projection of the test point  $P$  on the curve  $C$ , target function  $f$  can be expressed as

$$f(u) = \mathbf{C}'(u) \cdot (\mathbf{C}(u) - \mathbf{P}) \quad (4.12)$$

When  $f(u)$  gives 0, the point either located on the curve or the distance  $|\mathbf{C}(u) - \mathbf{P}|$  is minimal. and two scalars  $f$  and  $g$  are defined as The iteration can be concluded as

$$u_{i+1} = u_i - \frac{f(u_i)}{f'(u_i)} \quad (4.13)$$

After one iteration is finished, the following criteria are checked in sequence.

- 1 Is the point coincide with  $C(u_i)$

$$|\mathbf{C}(u_i) - \mathbf{P}| \leq \epsilon_1$$

#### 4.4. POINTS PROJECTION

---

where  $\epsilon_1$  stands for the tolerance for distance in Euclidean space.

**2** Is the cosine zero

$$\frac{|\mathbf{C}'(u) \cdot (\mathbf{C}(u) - \mathbf{P})|}{|\mathbf{C}'(u)| |\mathbf{C}(u) - \mathbf{P}|} \leq \epsilon_2$$

where  $\epsilon_2$  stands for the tolerance for cosine. If either of these conditions are met, the iteration is terminated. Otherwise Eq. 4.13 is performed to find the parameter  $u_{i+1}$  for next iteration.

**3** Make sure  $u$  and  $v$  are within there domains

$$u_{i+1} \in [a, b]$$

where  $a$  and  $b$  are the lower and upper bounds for the knot vector of curve  $C$ . If the curve is open

$$\begin{cases} u_{i+1} = a & u_{i+1} < a \\ u_{i+1} = b & u_{i+1} > b \end{cases} \quad (4.14)$$

If the curve is closed

$$\begin{cases} u_{i+1} = b - (a - u_{i+1}) & u_{i+1} < a \\ u_{i+1} = a + (u_{i+1} - b) & u_{i+1} > b \end{cases} \quad (4.15)$$

#### 4.4. POINTS PROJECTION

---

4 The difference between the new parameter  $u_{i+1}$  and the old one  $u_i$  is insignificant

$$|(u_{i+1} - u_i)\mathbf{C}'(u_i)| \leq \epsilon_1$$

The iteration will be terminated if this condition is meet.

##### 4.4.2 Convex hull in 2D

The convex hull property of the NURBS curve indicates that all points on the curve must be contained within the convex hull constructed by its control points (Selimovic, 2009). There are great number of algorithm that can be used including gift wrapping (Cormen et al., 2009), graham scan (Anderson, 1978), quick hull (Barber et al., 1996), Chan's algorithm (Chan, 1996) and so on (Andrew, 1979; Kirkpatrick and Seidel, 1986). The quick hull is adopted in the proposed as it provides a computationally efficient and stable algorithm. The algorithm utilize the idea of “divide and conquer” to build the convex hull with an expected time complexity of  $O(n \log(n))$  and  $O(n^2)$  for the worst case. Generally speaking, it works as expected in most of the situation except for the case of high symmetry or most of the points located at the circumference of a circle. The algorithm can be implemented with following steps:

1. Find the most left and right points (points with minimal and maximum  $x$ ) since they are proved to be part of the convex hull.
2. Connect these two points and use the line to separate other points

## 4.5. NUMERICAL EXAMPLES

---

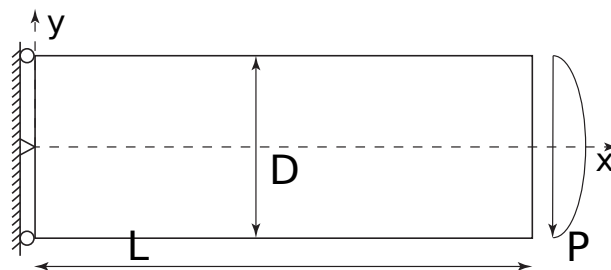
into two group.

3. Find the point with maximum distance to the line in step 2 in any group.
4. Construct a triangle with two points in step 2 and the point in step 3.
5. Eliminate all points contained by these two subsets in step 4.
6. Repeat the previous three steps and the distance calculated in step 2 is determined as the point to the triangle instead of the line in step 1.
7. Terminate the iteration when no points are left

## 4.5 Numerical Examples

### 4.5.1 Cantilever beam

A two-dimensional cantilever beam subjected to a parabolic shear load at the free end is examined as shown in Fig. 4.25. The geometry



**Figure 4.25:** Cantilever beam: Geometry and boundary conditions.



#### 4.5. NUMERICAL EXAMPLES

---

is: length  $L = 8$  m, height  $D = 4$  m. The material properties are: Young's modulus  $E = 3 \times 10^7 \text{ N m}^{-2}$ , Poisson's ratio  $\nu = 0.25$ . The parabolic shear force is  $P = 250$  N. The exact solutions for the displacements are given by (Augarde and Deeks, 2008) as Eq. 3.53. where  $I = D^3/12$  is the moment of inertia,  $\bar{E} = E$ ,  $\bar{\nu} = \nu$  and  $\bar{E} = E/(1 - \nu^2)$ ,  $\bar{\nu} = \nu/(1 - \nu^2)$  for plane stress and plane strain condition respectively. The stress  $\sigma$  can be expressed as (Augarde and Deeks, 2008) as Eq. 3.54. The strain energy can be derived from Eq. 3.54 and Eq. 3.53 as Eq. 3.55.

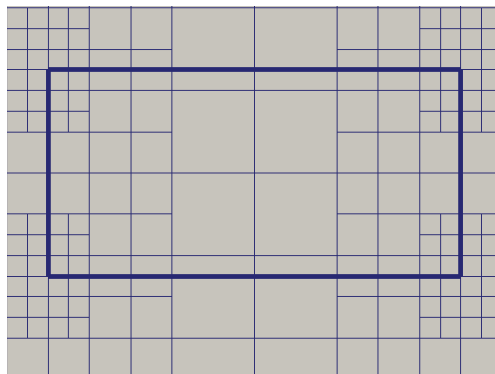
In this example, rigid body motion is constrained by fixing 3 DOF on the left edge of the beam.  $u_x = 0$  for points at  $(0, -D/2)$  and  $(0, D/2)$  and  $u_y = 0$  for point at  $(0, 0)$ . Stress from analytical solution in Eq. 3.54 are applied on the boundaries.

Due to the fact that the geometry of the cantilever beam can be described by four points and four straight lines, drawing in AutoCAD may not be necessary. As a result, the input geometry is defined manually. Generated background mesh, coloring and the final result with  $res = 32$ ,  $s_{max} = 16$  and  $s_{min} = 1$  are shown in Fig. 4.26, Fig. 4.27 and Fig. 4.28.

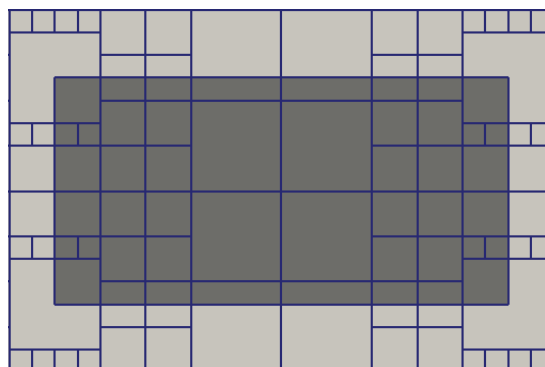
Mesh with different parameters are plotted in Fig. 4.29 and the convergence study is plotted in Fig. 4.30

## 4.5. NUMERICAL EXAMPLES

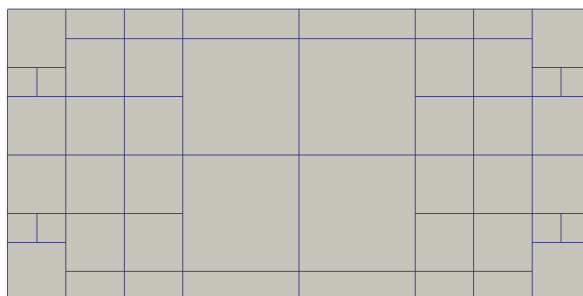
---



**Figure 4.26:** Background mesh of cantilever beam : Bold lines represents the input geometry



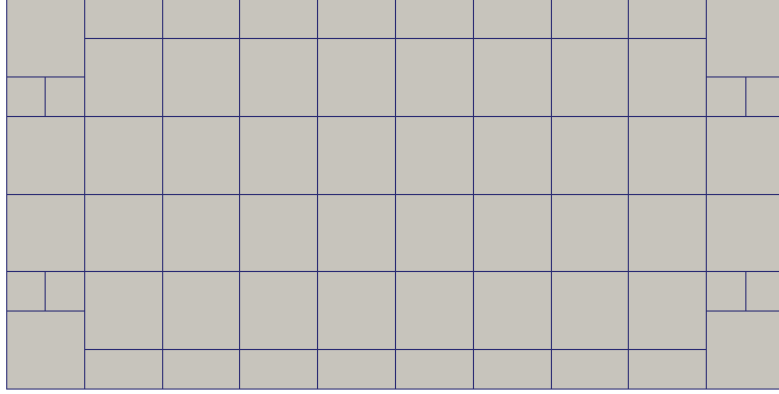
**Figure 4.27:** Mesh coloring of cantilever beam : Grey area represents the cantilever beam



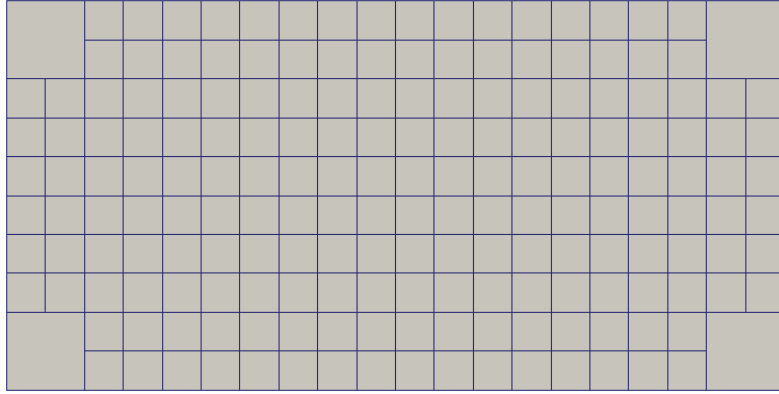
**Figure 4.28:** Final mesh of cantilever beam

## 4.5. NUMERICAL EXAMPLES

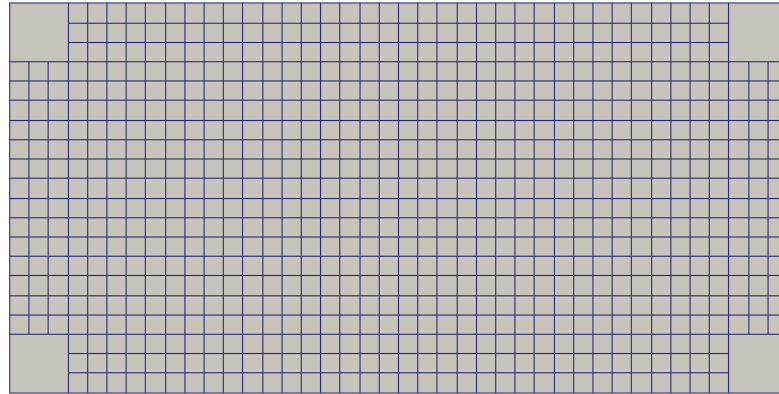
---



(a) Mesh with  $res = 32$ ,  $s_{max} = 2$ , 178 DOFs



(b) Mesh with  $res = 64$ ,  $s_{max} = 2$ , 438 DOFs



(c) Mesh with  $res = 128$ ,  $s_{max} = 2$ , 1658 DOFs

**Figure 4.29:** Mesh of the cantilever beam

## 4.5. NUMERICAL EXAMPLES

---

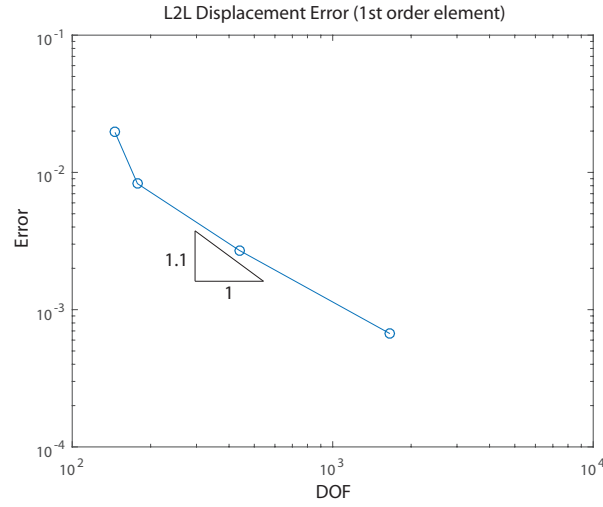


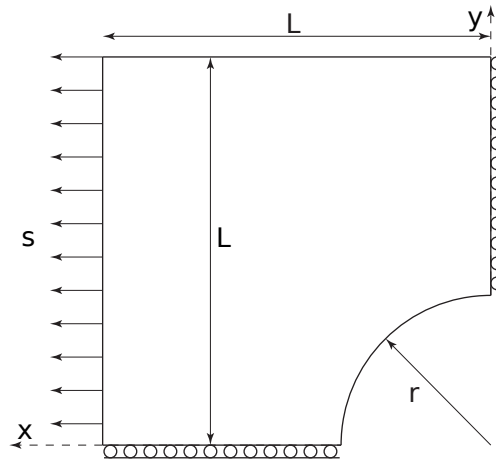
Figure 4.30: Convergence of the cantilever beam

### 4.5.2 Infinite plate with a circular hole

In this example, an infinite plate with a traction free hole under uni-axial tension ( $\sigma = 1 \text{ N m}^{-2}$ ) along x-axis (see Fig. 4.31) is considered.  $L$  is taken as 20 and  $r$  is 5. The material properties are: Youngs modulus  $E = 100 \text{ N m}^{-2}$  and Poissons ratio  $\nu = 0.3$ .

## 4.5. NUMERICAL EXAMPLES

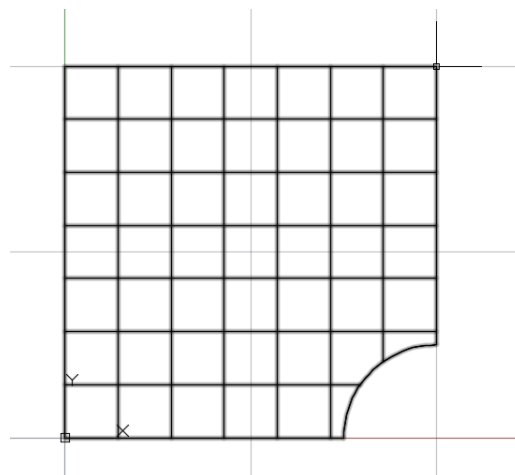
---



**Figure 4.31:** Infinite plate with a circular hole: geometry and boundary conditions

The exact solution of the principal stresses in Cartesian coordinate  $(r, \theta)$  is given by (Sukumar et al., 2001) in Eq. 3.61. The closed form displacement in Cartesian coordinate is given in Eq. 3.62.

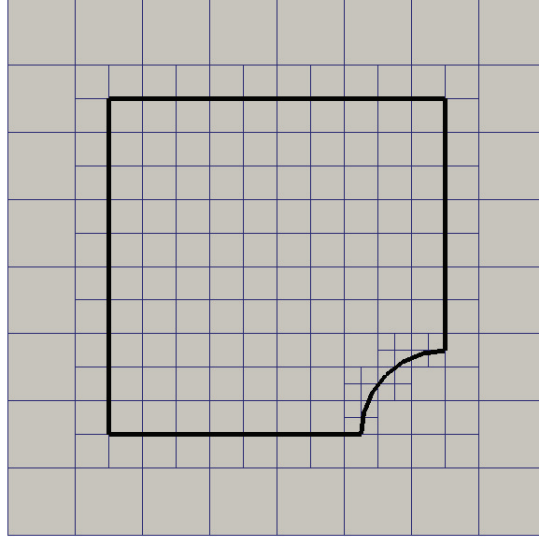
Geometry of the example will be extracted from the iges file drawn in AutoCAD (Fig. 4.32).



**Figure 4.32:** Infinite plate with a circular hole: CAD drawing

## 4.5. NUMERICAL EXAMPLES

---



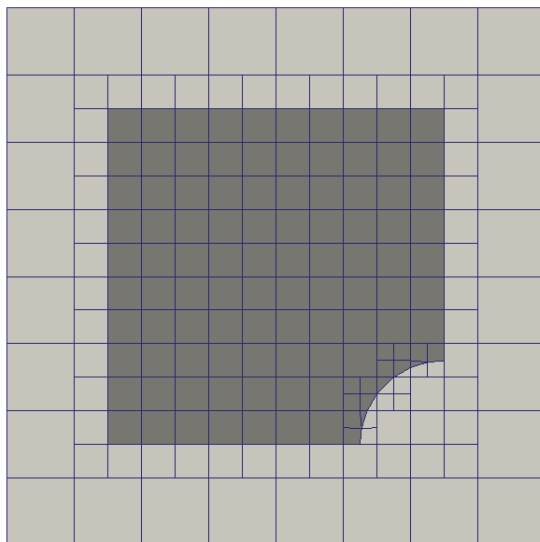
**Figure 4.33:** Background mesh of infinite plate with a circular hole : Bold lines represents the input geometry

Generated background mesh, coloring and the final result with  $res = 32$ ,  $s_{max} = 4$  and  $s_{min} = 1$  are shown in Fig. 4.41, Fig. 4.42 and Fig. 4.43.

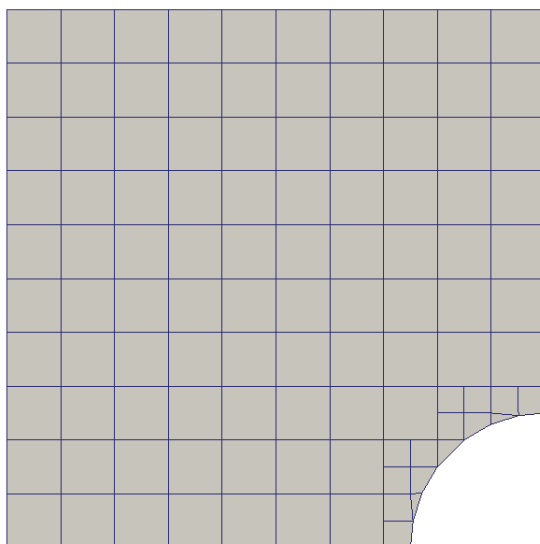
Mesh with different parameters are plotted in Fig. 4.36 and the convergence study is plotted in Fig. 4.37

## 4.5. NUMERICAL EXAMPLES

---



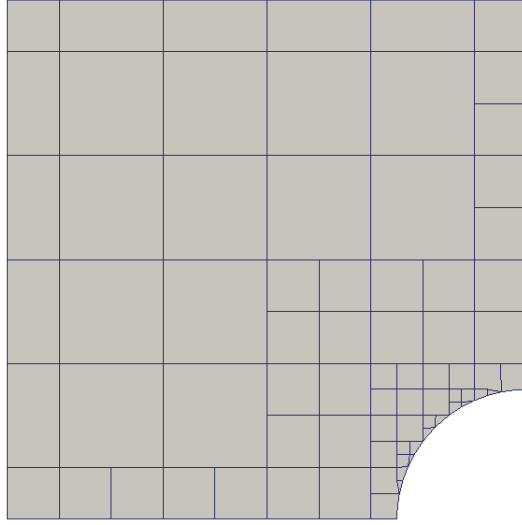
**Figure 4.34:** Mesh coloring of infinite plate with a circular hole : Grey area represents the plate



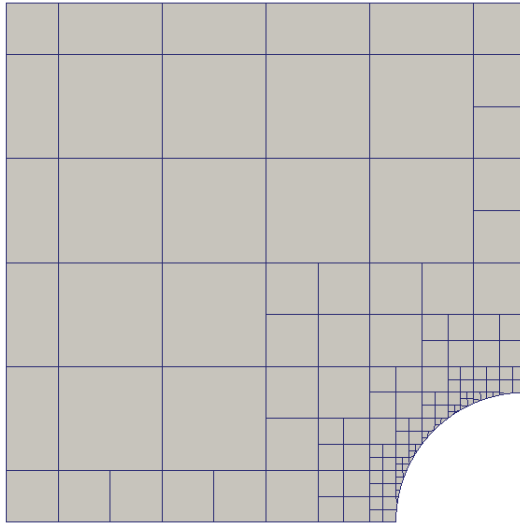
**Figure 4.35:** Final mesh of infinite plate with a circular hole

## 4.5. NUMERICAL EXAMPLES

---



(a) Mesh with  $res = 64$ ,  $s_{max} = 8$ , 152 DOFs

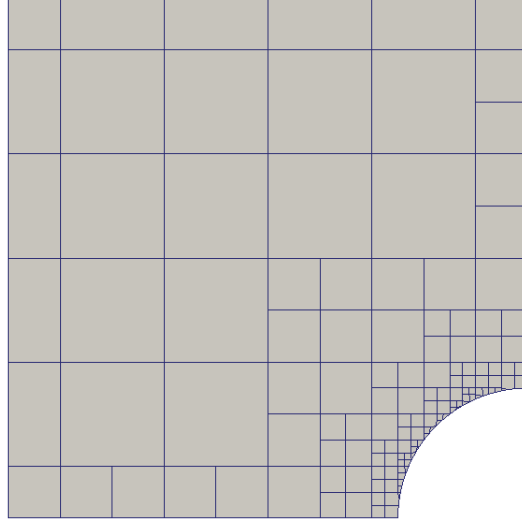


(b) Mesh with  $res = 128$ ,  $s_{max} = 4$ , 272 DOFs



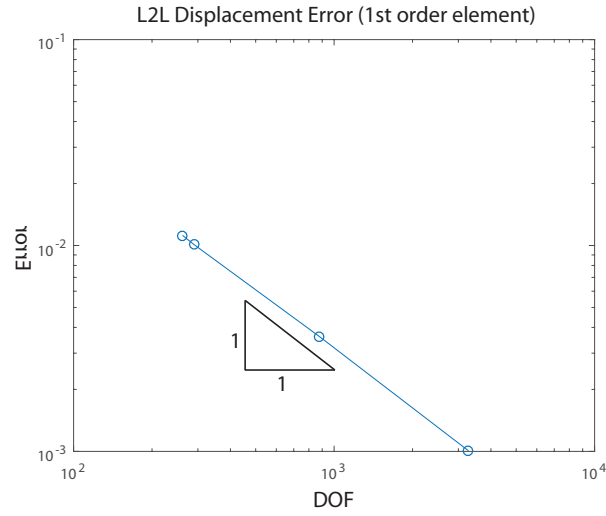
## 4.5. NUMERICAL EXAMPLES

---



(c) Mesh with  $res = 128$ ,  $s_{max} = 16$ , 488 DOFs

**Figure 4.36:** Mesh of infinite plate with a circular hole



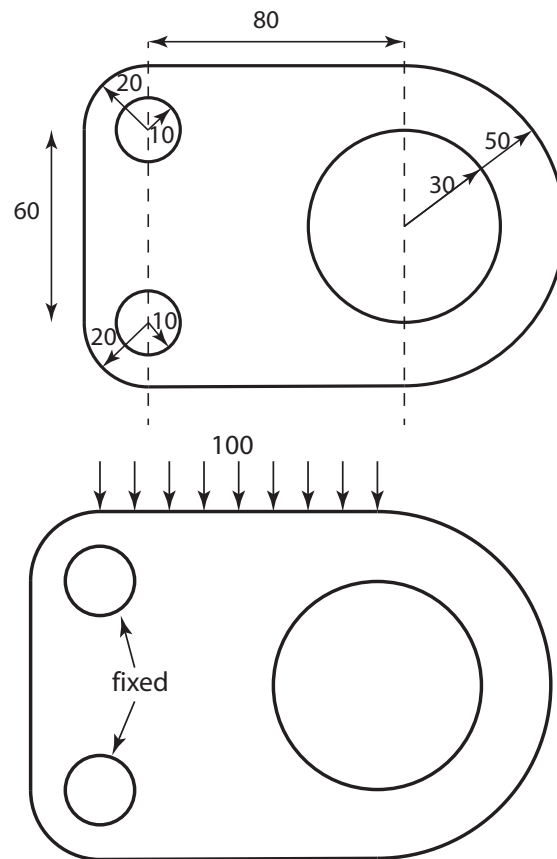
**Figure 4.37:** Convergence of the infinite plate with a circular hole

## 4.5. NUMERICAL EXAMPLES

---

### 4.5.3 Plane strain bracket

In this example, a plane strain bracket with a downward uniform distributed load on the top is considered (see Fig. 4.38). The material properties are: Youngs modulus  $E = 2 \times 10^5 \text{ N m}^{-2}$  and Poissons ratio  $\nu = 0.3$ .

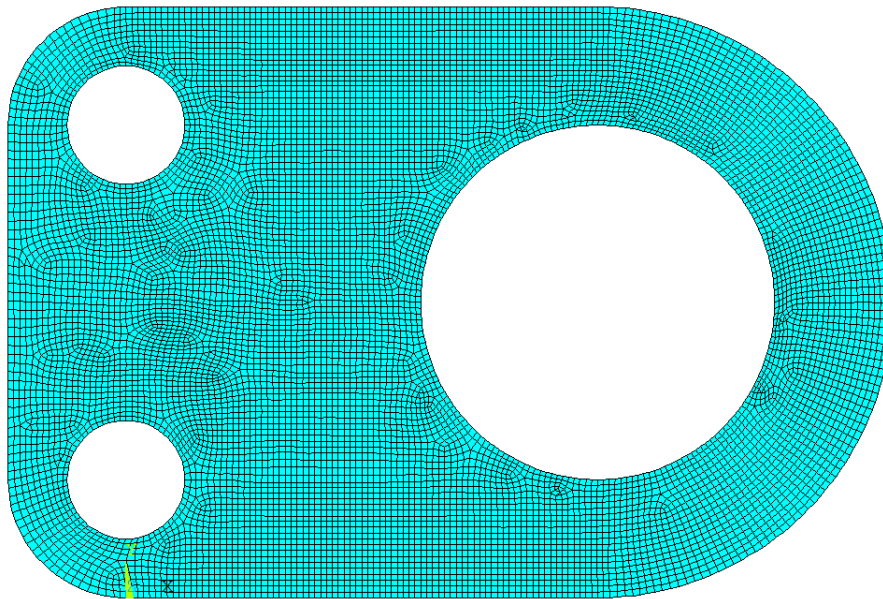


**Figure 4.38:** Plane strain bracket: geometry and boundary conditions

A total strain energy of 282.927 J is determined by ANSYS with the mesh shown in Fig. 4.39

## 4.5. NUMERICAL EXAMPLES

---

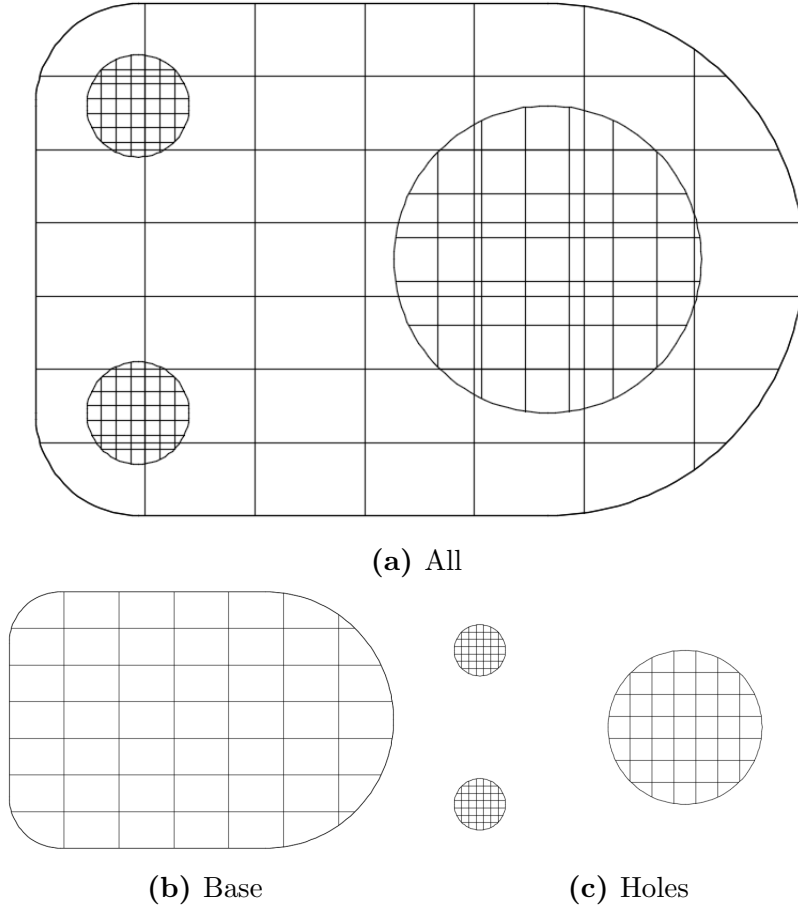


**Figure 4.39:** ANSYS mesh for plane strain bracket (62004 DOF)

Drawing in AutoCAD will be divided into two parts: base and the holes as shown in Fig. 4.40

## 4.5. NUMERICAL EXAMPLES

---



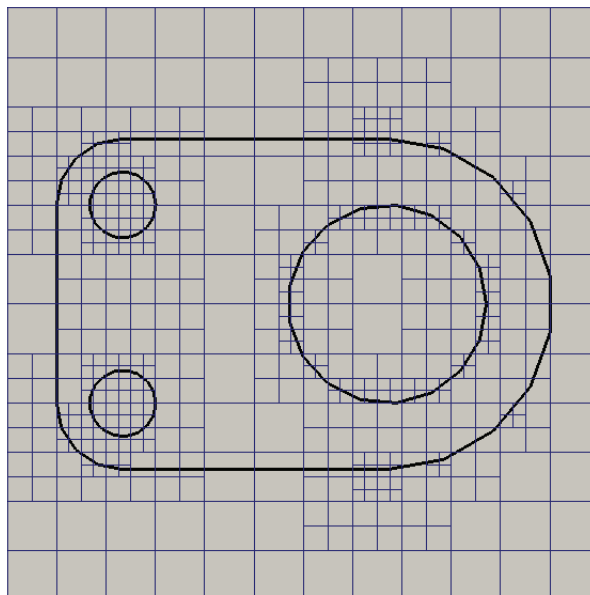
**Figure 4.40:** CAD drawing of plane strain bracket

Generated background mesh, coloring and the final result with  $res = 32$ ,  $s_{max} = 4$  and  $s_{min} = 1$  are shown in Fig. 4.41, Fig. 4.42 and Fig. 4.43.

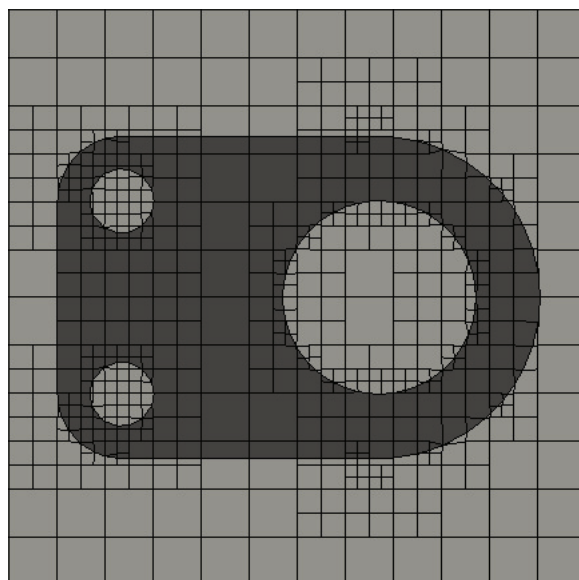
Mesh with different parameters are plotted in Fig. 4.44 and the convergence study is plotted in Fig. 4.45

## 4.5. NUMERICAL EXAMPLES

---



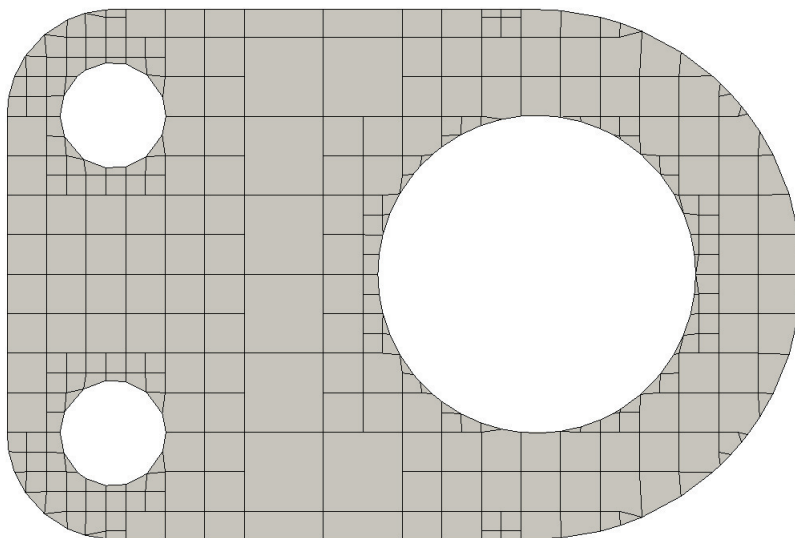
**Figure 4.41:** Background mesh of the bracket : Bold lines represents the input geometry



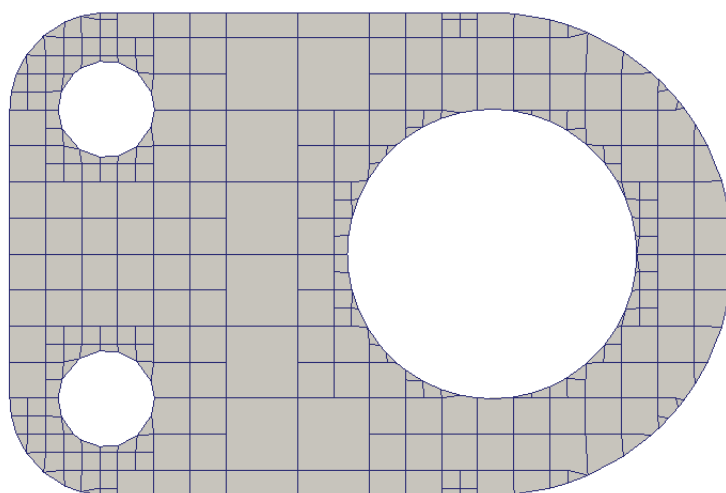
**Figure 4.42:** Mesh coloring of the bracket : Grey area represents the bracket

## 4.5. NUMERICAL EXAMPLES

---



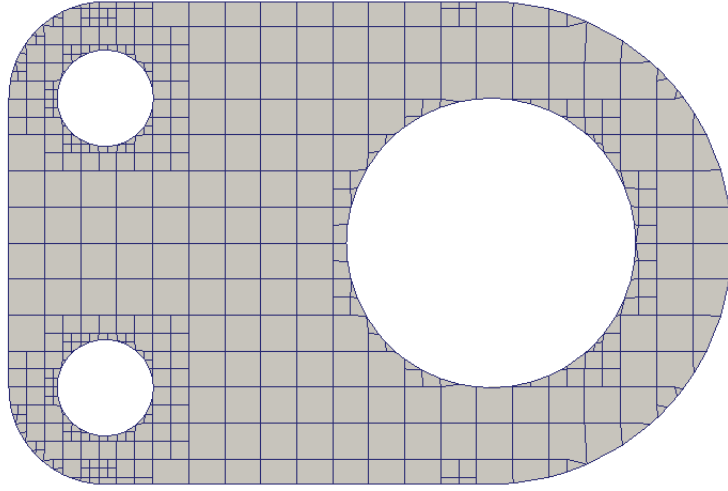
**Figure 4.43:** Final mesh of the bracket



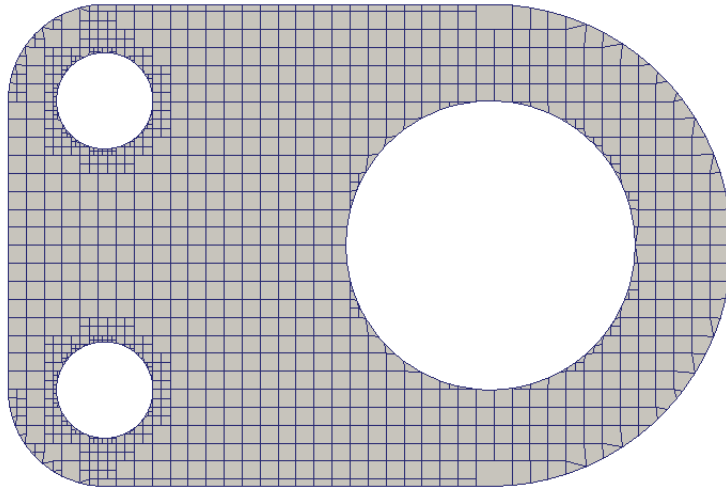
**(a)** Mesh with  $res = 64$ ,  $s_{max} = 4$ , 1656 DOFs

## 4.5. NUMERICAL EXAMPLES

---



(b) Mesh with  $res = 128$ ,  $s_{max} = 4$ , 2548 DOFs

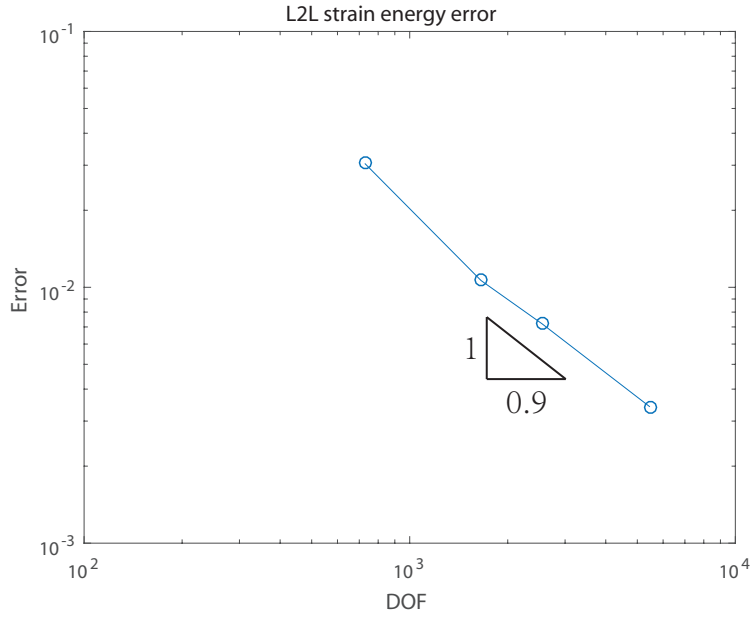


(c) Mesh with  $res = 256$ ,  $s_{max} = 4$ , 5464 DOFs

**Figure 4.44:** Mesh of the plane strain bracket

## 4.5. NUMERICAL EXAMPLES

---



**Figure 4.45:** Convergence of the plane strain bracket

Fig. 4.46 shows the von Mises equivalent stress for the plane strain bracket. From Fig. 4.46, it can be observed that the results from the present approach qualitatively match with the FE solution.

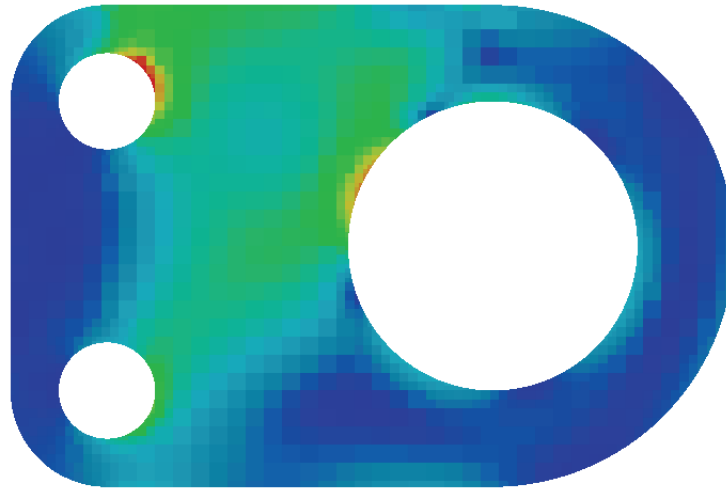
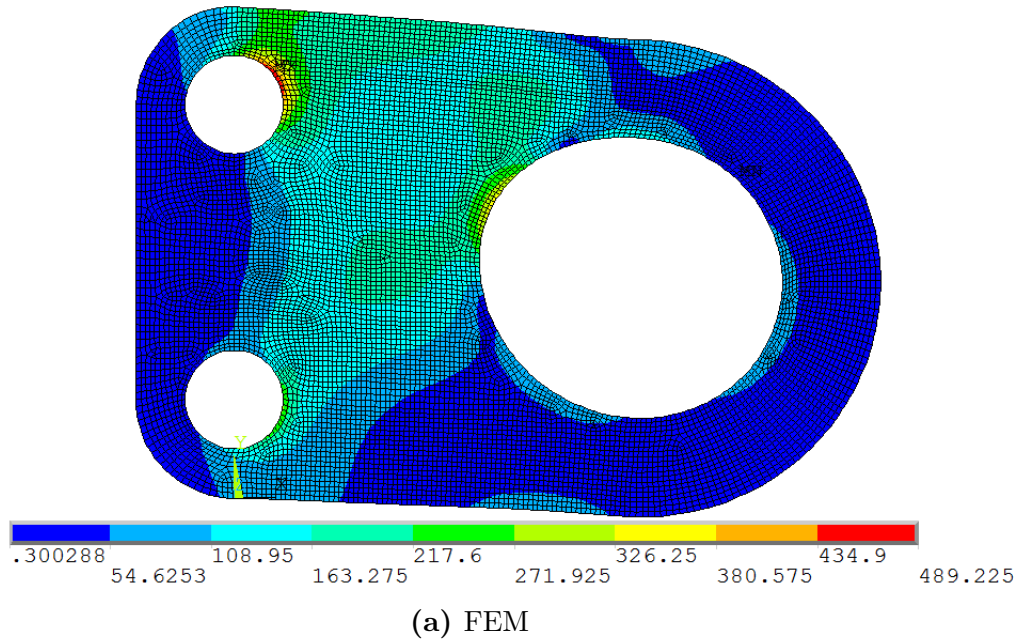
### 4.5.4 Buildings on the ground

In this example, a few buildings on different layers under gravity of soils are considered. Fig. 4.47 illustrates the geometry and boundary condition of the problem and Fig. 4.48 shows the material property of the problem. Fig. 4.49 represents the design file in AutoCAD. Fig. 4.50 and Fig. 4.51 plot the mesh and the von-mises stress correspondingly.



## 4.5. NUMERICAL EXAMPLES

---



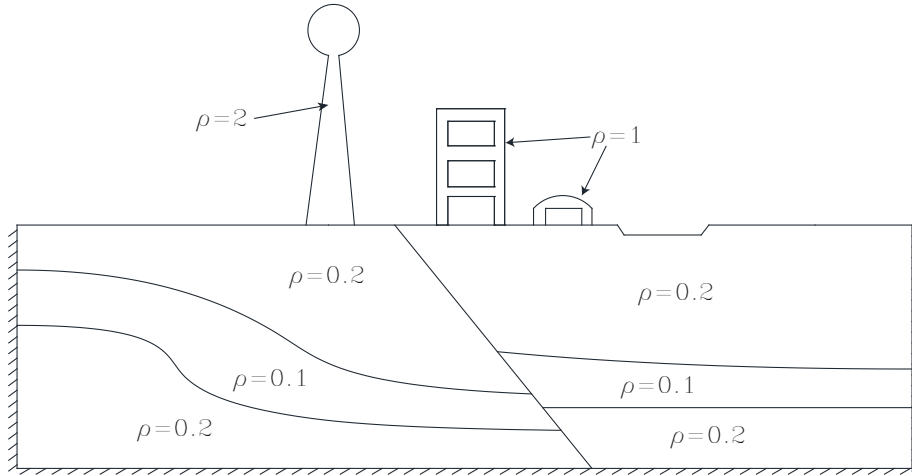
**Figure 4.46:** Von Mises equivalent stress contours for plane strain bracket

### 4.5.5 Other meshes

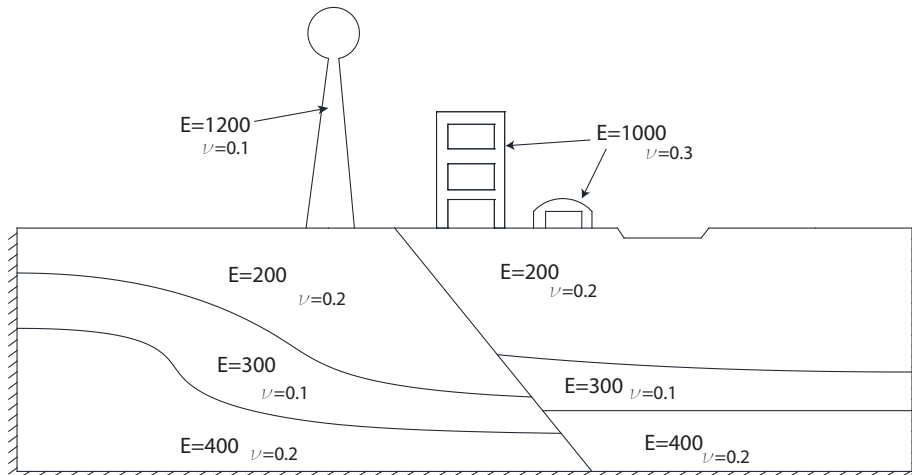
In this section, some other mesh examples with irregular geometric boundaries are considered. Fig. 4.52 and Fig. 4.53 show the mesh generated

## 4.5. NUMERICAL EXAMPLES

---



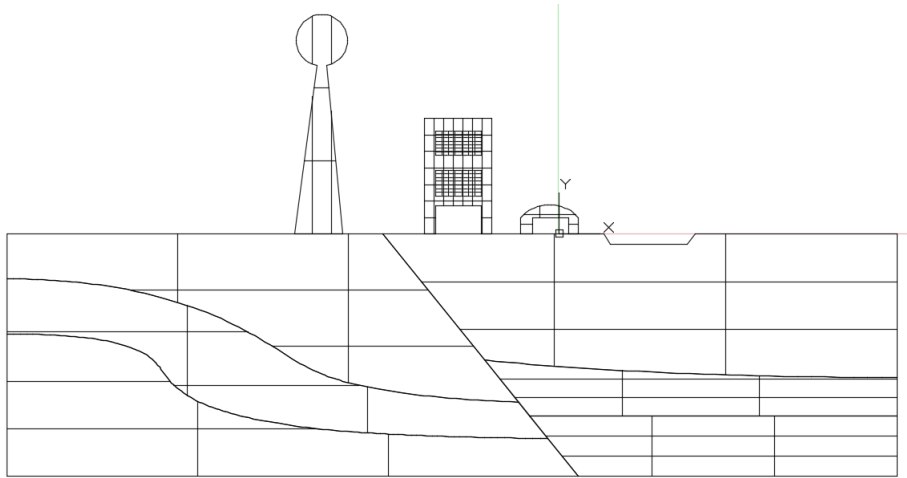
**Figure 4.47:** Geometry and boundary condition of the problem



**Figure 4.48:** Geometry and boundary condition of the problem

## 4.6. CONCLUSIONS

---



**Figure 4.49:** Design in AutoCAD

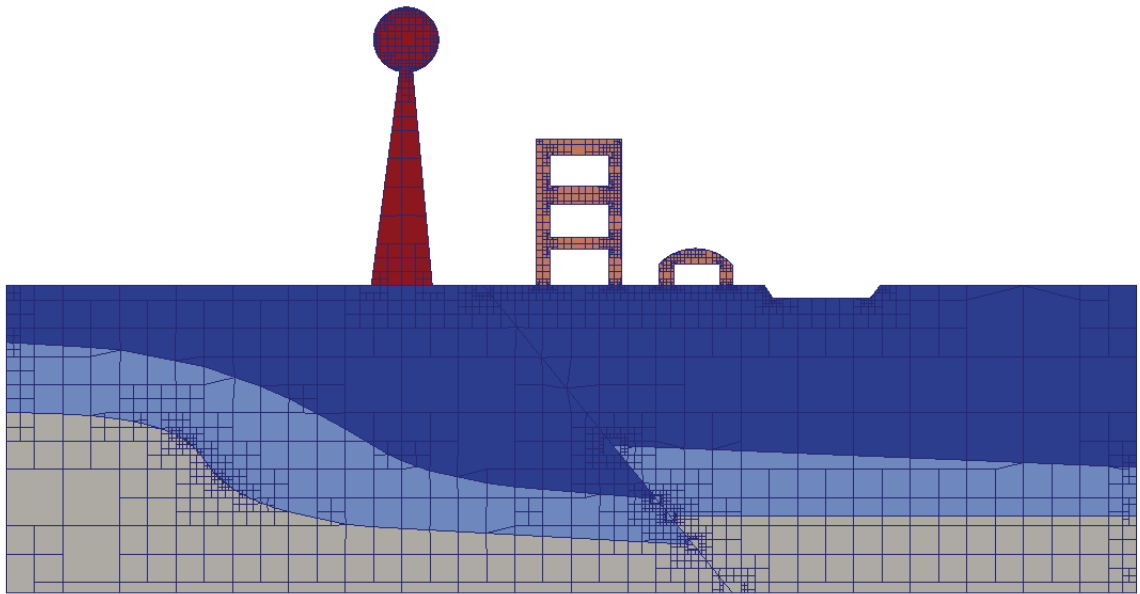
for a flower input. Fig. 4.52b and Fig. 4.53b highlight the sharp corner treatment of the algorithm.

## 4.6 Conclusions

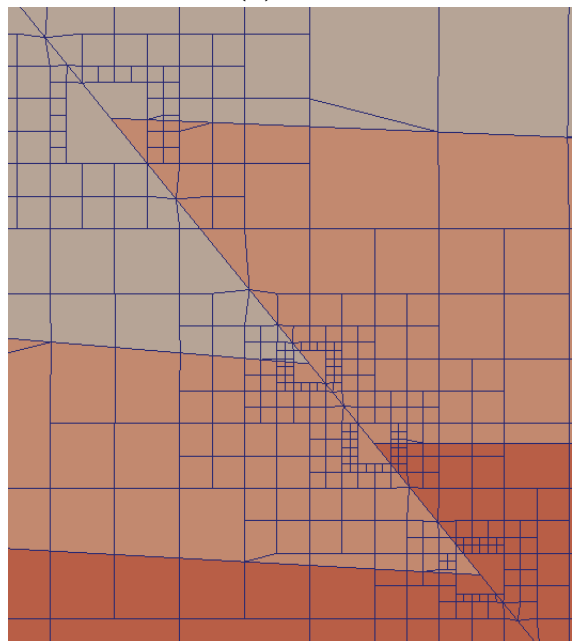
In this chapter, the IGES file is employed directly from the CAD output to represent the geometry during the preprocessing. The proposed methods provides a systematic and automatic mesh generation algorithm where a high quality mesh that meets the scaling requirement discussed in Sec. 2.2 is produced efficiently. The CAD design file can be used directly and the exact geometry can also be retained, which largely reduce the human efforts involved in numerical analysis. Moreover, it helps to reduce the analytical error as the difference in the geometric representation is minimized. Computational efficiency is improved via utilizing the pattern

#### 4.6. CONCLUSIONS

---



(a) Mesh

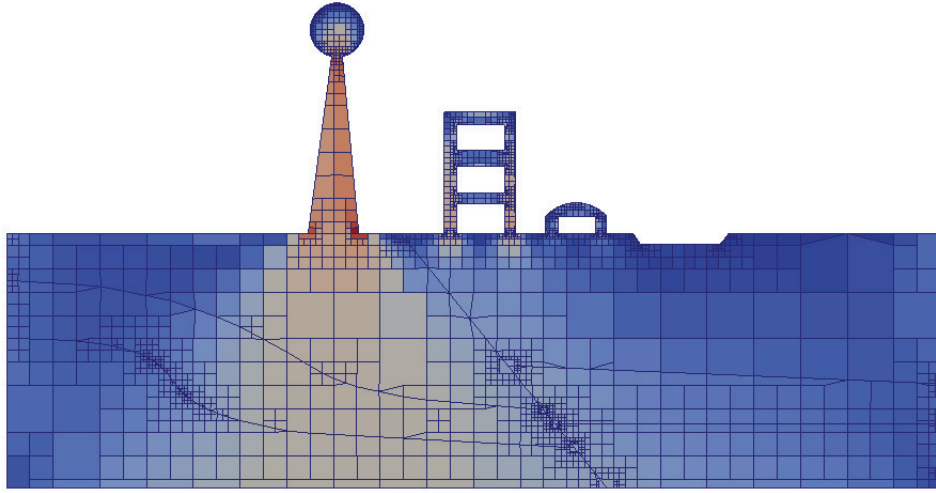


(b) Mesh at the sharp corner

**Figure 4.50:** Meshing of the buildings

## 4.6. CONCLUSIONS

---

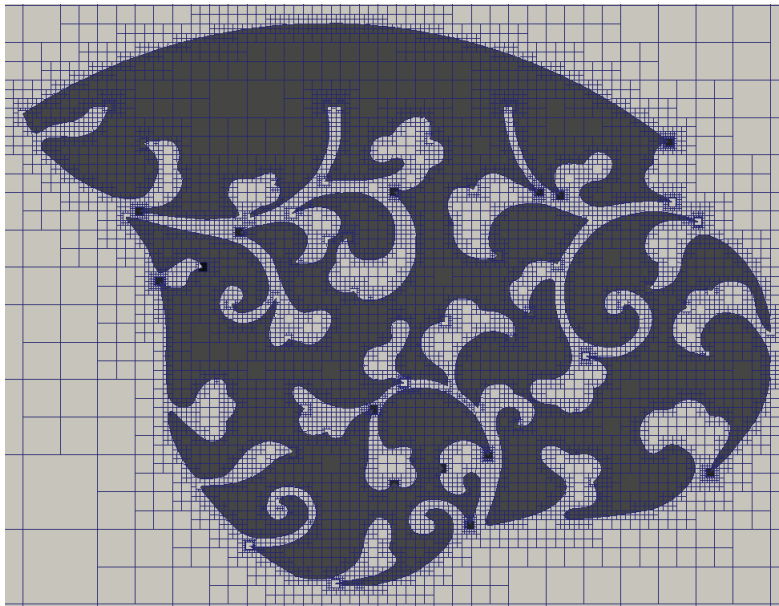


**Figure 4.51:** Von-mises stress contour for the buildings on the ground

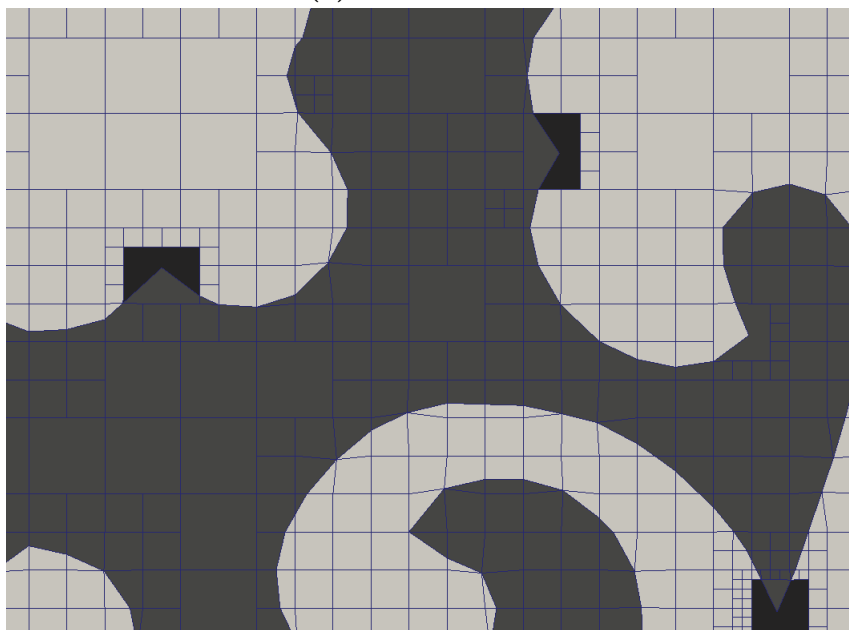
of the quadtree to prevent repeated calculation. Points projection algorithm is accelerated using the strong convex hull property of the NURBS curve. The accuracy, effectiveness and the convergence properties of the proposed method are demonstrated with benchmark problems in linear elasticity mechanics. From the numerical studies, it can be observed that the quadtree mesh yields better accuracy compared to uniform mesh with same degree of freedoms.

#### 4.6. CONCLUSIONS

---



(a) Mesh for flower



(b) Mesh for flower : Sharp corners treatment

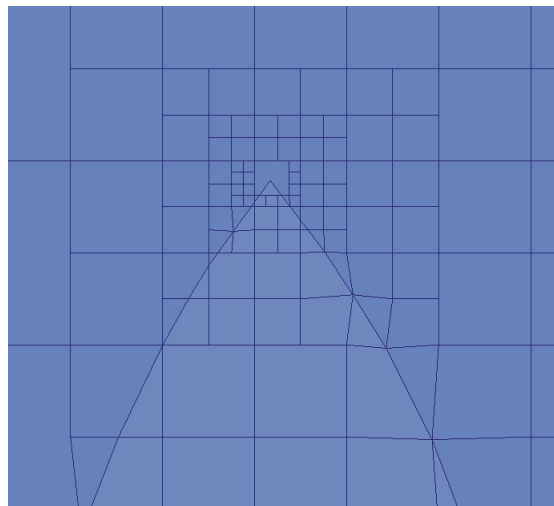
**Figure 4.52:** Mesh of flower

#### 4.6. CONCLUSIONS

---



(a) Mesh for woolworth logo : Different colors for different materials



(b) Mesh for woolworth logo : Sharp corners treatment

**Figure 4.53:** Mesh of woolli logo

# Chapter 5

## Adaptivity

### 5.1 Introduction

The main objective of this chapter is to introduce a way that can develop an adaptive mesh automatically based on the posteriori error estimator. Most of the unnecessary refinement in the region which contributes few to the improvement of the accuracy will be prevented. The expressions related to the eigenvalues of the SBFEM formulation representing the quantity of the error in the interpolation are adopted as one of the error indicators, together with the area and other geometric properties of the Scaled Boundary Finite Element. A machine learning model using the Multilayer Perceptron (MLP) is trained to determine whether a Scaled Boundary Finite Element needs refinement or not based on all these information.



## 5.2. ERROR INDICATOR

---

The proposed method enhances the SBFEM with quad-tree mesh introduced in Sec. 4 and the outstanding features of the method are:

- No human effort involvement in mesh refinement
- Smart refinement detection
- Highly extensible criteria, any other error indicators can be added to the existing framework

This chapter is organized as follows. The error indicators used in the proposed method will be introduced first. After that, a machine learning algorithm that can be trained to determine the necessity of the refinement of a cell based on these indicators is presented. Furthermore, a triangle merging algorithm is developed to bypass the lack of eigenvalue error indicators in first order triangular elements in the SBFEM. Finally, the matrix representation of NURBS curves is introduced to improve the computational efficiency and the robustness of the NURBS related calculation. The accuracy and the convergence properties of the proposed techniques are demonstrated with benchmark problems in the context of linear elasticity, followed by concluding remarks in the last section.

## 5.2 Error indicator

This section details the proposed error indicators for adaptive scaled boundary finite element analysis. In comparison with the works in (Deeks

## 5.2. ERROR INDICATOR

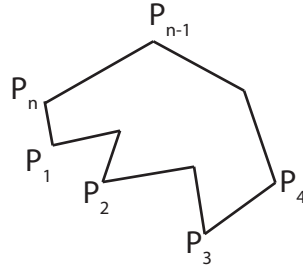
---

and Wolf, 2002c), no stress recovery is required. This reduces the computational efforts and leads to an efficient adaptive analysis. The error estimation is directly invoked from the scaled boundary finite element solutions.

### 5.2.1 Mesh Size

The area of a subdomain will apparently influence the accuracy of the result. Generally speaking, a larger subdomain tends to lead to a higher error. The area of any polygon in Fig. 5.1 can be calculated as

$$S = \frac{1}{2} \sum_{k=1}^n (x_k y_{k+1} - x_{k+1} y_k) \quad (5.1)$$

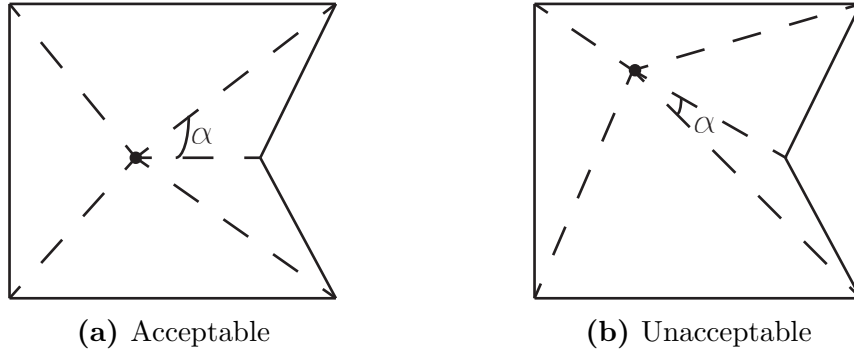


**Figure 5.1:** A polygon with  $n$  vertices

## 5.2. ERROR INDICATOR

### 5.2.2 Mesh Quality

Mesh quality is another important factor that influences the accuracy. In SBFEM, the mesh quality is highly related to the minimal angle formed by the intersecting lines connected to the scaling center and the adjacent polygon vertexes. An extremely small angle as shown in Fig. 5.2 may raise numerical stability issues and hence decrease the accuracy of the result.



**Figure 5.2:** Scaling center ( ● ), minimal angle  $\alpha$

### 5.2.3 Eigenvalue in SBFEM

From Eq. 3.24, the displacement in the SBFEM can be calculated as

$$u(\xi) = c_1 \xi^{-\lambda_1} \phi_u^1 + c_2 \xi^{-\lambda_2} \phi_u^2 + \dots \quad (5.2)$$

where  $\phi_u^i$  stands for the eigenvector corresponding to the  $i$ th eigenvalue in the eigenvalue matrix  $\Lambda^n$  in Eq. 3.20. The contribution of each mode is

## 5.2. ERROR INDICATOR

---

represented by every individual term in Eq. 5.2 (Deeks and Wolf, 2002b). On the other hand, after the displacement solution is calculated on the boundary nodes by Eq. 3.21 with  $\xi = 1$ , displacement solution in circumferential direction then is interpolated by the help of the  $p$ -th order shape function  $R(\eta)$  in Eq. 3.24. As a result, terms corresponding to the eigenvalue  $\lambda_i \leq p$  can be exactly interpolated by the shape function. These terms can be regarded as exact. However, terms corresponding to the eigenvalue  $\lambda_i > p$  indicate the shape functions are not capable to interpolate the displacement exactly and thus shall be taken as error. Consequently, displacement on the boundary  $u(\xi = 1)$  can be expressed with exact terms and approximation terms as followed

$$u_b = u_e(\xi = 1) + u_a(\xi = 1) \quad (5.3)$$

where the exact terms for the displacement on the boundary can be expressed as

$$u_e(\xi = 1) = \sum c_i \phi_u^i \text{ for all } \lambda_i \leq p \quad (5.4)$$

and similarly, the approximation terms can be written as

$$u_a(\xi = 1) = \sum c_i \phi_u^i \text{ for all } \lambda_i > p \quad (5.5)$$

### 5.3. MACHINE LEARNING

---

Follow the same logic, nodal forces on the boundary can be expressed As

$$\begin{aligned} q_b &= q_e(\xi = 1) + q_a(\xi = 1) \\ q_e(\xi = 1) &= \sum c_i \phi_q^i \text{ for all } \lambda_i \leq p \\ q_a(\xi = 1) &= \sum c_j \phi_q^j \text{ for all } \lambda_j > p \end{aligned} \quad (5.6)$$

The energy can be calculated by the

$$\begin{aligned} U &= \frac{1}{2} u_b^T q_b = U_e + U_a \\ U_e &= u_e q_e \\ U_a &= u_e q_a + u_a q_e + u_e q_e \end{aligned} \quad (5.7)$$

The relative error for energy can be calculated as  $\frac{U_a}{U}$  and the same logic for displacement and stress.

In order to obtain satisfactory solutions within certain accuracy, the contribution of  $U_a$  towards  $U$  should be minimized and  $U_a$  should also be distributed evenly to all cells. This is, in fact, similar to the relative energy norm error used in the literature (Zienkiewicz et al., 1989).

## 5.3 Machine learning

Due to the fact that there are a few error indicators, it could be hard to determine the relationship between them and whether a subdomain needs

### 5.3. MACHINE LEARNING

---

refinement or not. As a result, specific technique is required in order to search for the patterns in the data. After that, the final decision which is a binary classification in this case as each subdomain will be labeled as “refine” or “not refine” can be made based on the input of the error indicators explained in Sec. 5.2. Hence, the discovery of regularities plays a key role in the adaptive analysis and an automatic discovery of regularities is usually associated with pattern recognition by the help of algorithms.

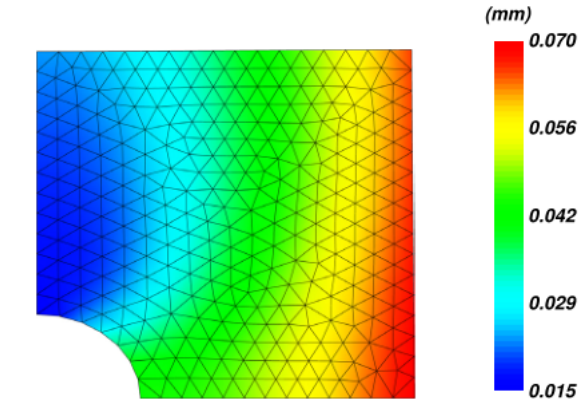
#### 5.3.1 Training Set

The training set is determined by numerical examples in Fig. 5.3 and Fig. 5.4. The training data is annotated with whether a subdomain is refined or not and the model can study the data and learn to classify each subdomain based on all four features explained in 5.2. With the same example, uniform mesh of quadtree is conducted and the same region are marked as refined as shown in Fig. 5.5. Criteria taken into consideration are:

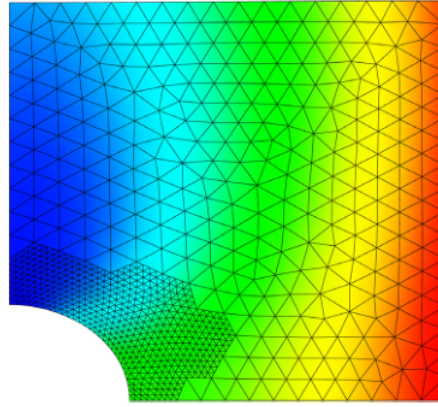
1. Ratio of the area of the cell to the total area
2. Minimal angle formed by the intersecting lines connected by scaling center and adjacent polygon vertexes
3. Eigenvalue error indicator for displacement
4. Eigenvalue error indicator for stress

### 5.3. MACHINE LEARNING

---



(a) Mesh and displacement field before refinement



(b) Mesh and displacement field after refinement

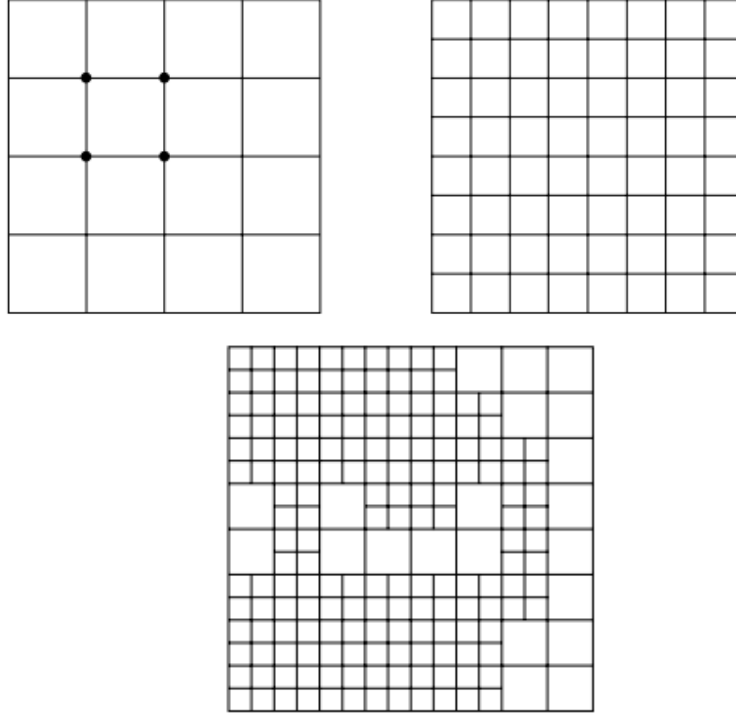
**Figure 5.3:** Mesh refinement for square plate with circular hole (Duval et al., 2018)

#### 5.3.2 Regularization for MLP

**Bagging** Bagging is a technique that utilizes multiple models in order to increase the accuracy of the prediction (Breiman, 1996). The principle idea of that is simple: training multiple models separately and let them vote for the prediction. It is adopted vastly in machine learning. The idea behind the optimization technique is that different models often make dif-

### 5.3. MACHINE LEARNING

---



**Figure 5.4:** Mesh refinement of a short cantilever beam (Zienkiewicz et al., 2005)

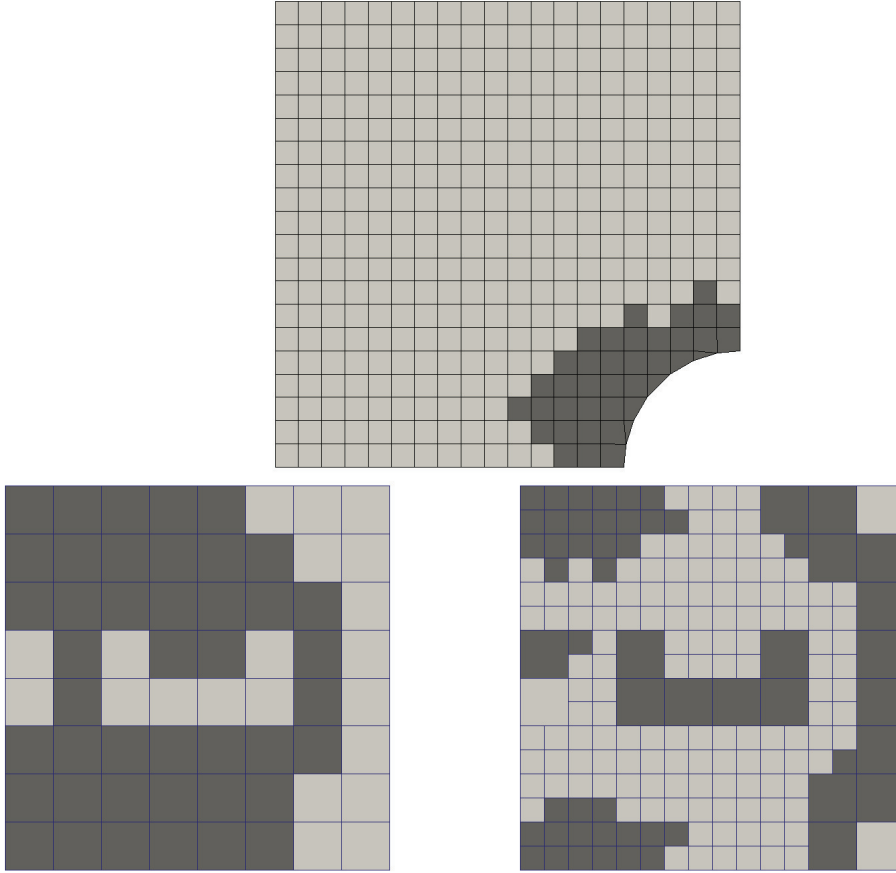
ferent errors on the same test set. Take a set of  $k$  regression models for an example, error  $\epsilon_i$  is made by each model on each example. The errors are drawn from a zero-mean multivariate normal distribution with variances  $E[\epsilon_i^2] = \nu$  and covariances  $E[\epsilon_i \epsilon_j] = c$ . As a result, the overall error determined by averaging of all models is  $\frac{1}{k} \sum_i \epsilon_i$ . The expected squared error of the ensemble predictor is

$$\begin{aligned}
 E\left[\left(\frac{1}{k} \sum_i \epsilon_i\right)^2\right] &= \frac{1}{k^2} \left[ \sum_i (\epsilon_i^2) + \sum_{j \neq i} \epsilon_i \epsilon_j \right] \\
 &= \frac{1}{k} \nu + \frac{k-1}{k} c
 \end{aligned} \tag{5.8}$$



### 5.3. MACHINE LEARNING

---



**Figure 5.5:** Training data for SVM: Cells in black is marked as refined.

The bagging technique does not help if the errors are highly correlated and  $c = \nu$  as the mean squared error can reduce to  $\nu$ . On the other hand, the effect of this technique may be significant if the errors are highly uncorrelated and  $c = 0$ , in which situation the overall mean squared error becomes  $\frac{1}{k}\nu$ . In other words, the overall mean squared error decreases linearly with the model size. To conclude, the performance of this technique must not be worse than any of its individual model and a significant improvement can be expected if its members make independent errors.

### 5.3. MACHINE LEARNING

---

There are several different methods to train ensemble. For instance, each member of the ensemble can be trained with same training set but different algorithms or different training set with same algorithm. The reuse of the same kind of models and algorithms is allowed thanks to the bagging. The bagging technique used in the proposed method is constructing  $k$  different training set. Each training is built by sampling the original data set randomly. In other words, each individual model may have some dataset which is not appear in any other models. There is also some dataset are shared by multiple models.

Due to the large number of initialization parameter for a neural network, bagging technique can be extremely effective because the random selection of parameters such as initialized weight, mini-batches, hyper-parameters and so on leads to a different outcome even with the same training data. The method is proved to be reliable and powerful for minimizing the generalization error. Averaging dozens of models is widely adopted in those who won the machine learning contests and a recent example is Netflix Grand Price (Koren et al., 2009).

**Dropout** Dropout (Srivastava et al., 2014) is another optimization approach used in the proposed method. An effective but computationally cheap regularizing technique is provided. Roughly speaking, the dropout can be regarded as one method to create different models in bagging described in Sec. 5.3.2 by training multiple models. One problem in bagging

### 5.3. MACHINE LEARNING

---

is that it could be impractical when each model is so large that the training can be computationally expensive in terms of time and space complexity. Hence, the number of models used in ensembles tends to be in the range of 5 and 10 and the winner of ILSVRC (Szegedy et al., 2014) used 6 models. As a comparison, dropout provides an cheap algorithm that can train and evaluate a large number of bagged ensemble.

In dropout, random hidden units are disabled during the training, which can be easily achieved by manually setting the weight to zero. By doing so, the contribution of this hidden unit is ignored based on the fact that matrix production is used. It can also be implemented by removing the unit completely from the network. In bagging technique,  $k$  different models with different training data are trained. This is shared by the dropout as well, while dropout supports larger number of neural networks. During the training process using dropout, a minibatch-based learning algorithm introduced in Sec. 2.3.4 is used to make insignificant steps. Whenever a sample is loaded into the minibatch, a random binary mask is applied to all of the inputs and hidden units in the network. The probability a unit be disabled is independent from others and shall be determined from the user input as one of the hyperparameter. In the proposed method, an input units has 20% chance of being disabled and the number is 50% for the hidden units.

### 5.3. MACHINE LEARNING

---

More formally, assume that a mask vector  $\mu$  specifies which units to include and  $J(\theta, \mu)$  defines the cost of the model defined by parameters  $\theta$  and mask  $\mu$  (Goodfellow et al., 2016). After that, the training of the dropout becomes minimizing  $E_{\nu} J(\theta, \mu)$ . The expectation may includes great number of terms while it could be possible to determine an unbiased estimation of its gradient by sampling values of  $\mu$ .

The training of the dropout is quite different to that of bagging. Models use same parameters but different training data in bagging as shared parameters help to represent numerous number of models without occupying too many memory. During the training using bagging, individual model is trained until convergence while during the training using dropout, most of the models are not explicitly trained. That is because the number of the possible sub-network in dropout is too significant to finish the training within the lifetime of the universe. Instead, only minor proportion are trained for a single step. Parameter sharing guarantee that the remaining sub-networks are able to arrive at good settings. The bagging algorithms are followed after that. For instance, the training data used by individual model actually is sampled from the original training set. Simple majority of the votes from all members are adopted to determine the prediction of the ensemble. Although bagging and dropout does not require a explicitly probabilistic model, it is assumed that a probability distribution  $p^{(i)}(y|x)$  will be the output. The prediction of the ensemble is given by averaging all

### 5.3. MACHINE LEARNING

---

these distributions,

$$\frac{1}{k} \sum_{i=1}^k (y|x) \quad (5.9)$$

For dropout, individual model with a mask vector  $\mu$  defines a probability distribution  $p(y|x, \mu)$ . The arithmetic mean over all masks is given by

$$\sum_{\mu} p(\mu) p(y|x, \mu) \quad (5.10)$$

where  $p(\mu)$  is the probability distribution used to sample  $\mu$  during the training. Since numerous number of terms are involved in the summation, it could be difficult to calculate Eq. 5.10 without simplification. However, a deep neural network do not allow such simplification. Instead, it is possible to approximate the simplification with sampling by calculate the mean value from many masks. It is suggested that around 15 masks are able to provide satisfactory performance (Goodfellow et al., 2016)

Yet, the existence of a better approach allows the determination of a satisfactory approximation to the predictions of the ensemble at the cost of one forward propagation. It can be achieved by adopting geometric mean instead of the arithmetic mean of the ensemble member's predicted distributions (Warde-Farley et al., 2014). However, using the geometric mean instead of the arithmetic mean leads to the result which is not guaranteed to be a probability distribution. In order to enforce a probability distribution as the result, the requirement that all sub-models must assign a non-zero probability to all events is imposed. After that, the resulted distribution is

### 5.3. MACHINE LEARNING

---

normalized again. The unnormalized probability distribution from geometric mean is:

$$\tilde{p}_{ensemble}(y|x) = 2^d \sqrt{\prod_{\mu} p(y|x, \mu)} \quad (5.11)$$

where  $d$  is the number of units that may be dropped. To simplify the presentation, a uniform distribution over  $\mu$  is adopted. But it should be noted that non-uniform distributions can be treated as well. Eq. 5.12 need to be performed on the ensemble in order to determine the prediction

$$p_{ensemble}(y|x) = \frac{\tilde{p}_{ensemble}(y|x)}{\sum_{y'} \tilde{p}_{ensemble}(y'|x)} \quad (5.12)$$

The key idea behind the dropout is to approximate  $p_{ensemble}$  by calculating  $p(y|x)$  in one model (Hinton et al., 2012). The aim of this improvement is to capture the correct output from that unit. It is called weight scaling inference rule which shows outstanding performance empirically.

Since an inclusion probability of  $\frac{1}{2}$  is widely adopted, the weight scaling rule tends to half the weights after training. The model is used as usual after that. It can also be achieved by multiplying the states of the units by 2 during training. Both of these two methods are to narrow the difference between the expected total input to a unit at test time and that at the training when approximately half of the hidden units are dropped. For those models without nonlinear hidden units, the weight scaling inference rule gives exact result. For instance, a softmax regression classifier with  $n$

### 5.3. MACHINE LEARNING

---

input variables represented by the vector  $v$  is considered:

$$P(y = y|v) = \text{softmax}(W^T v + b)_y \quad (5.13)$$

They can be indexed into the family of sub-models by element-wise multiplication of the input with a binary vector  $d$ :

$$P(y = y|v; d) = \text{softmax}(W^T (d \odot v) + b)_y \quad (5.14)$$

#### 5.3.2.1 Indicator used in adaptivity

All terms used in the performance indicators in machine learning are introduced in Sec. 2.3.8. The balance of the indicators is highly dependent on classifiers' objective. Take the spam detector for an example, a false positive can be more dangerous than a false negative is as an important e-mail being marked as spam can be a disaster. While in adaptivity, there may be no favour over either of them. It is because the refinement of a cell with lower error or leaving a cell with higher error unrefined may not produce significant influence on the final result. As a result, F1 score can be the most important indicator as it takes both recall rate and precision into consideration.

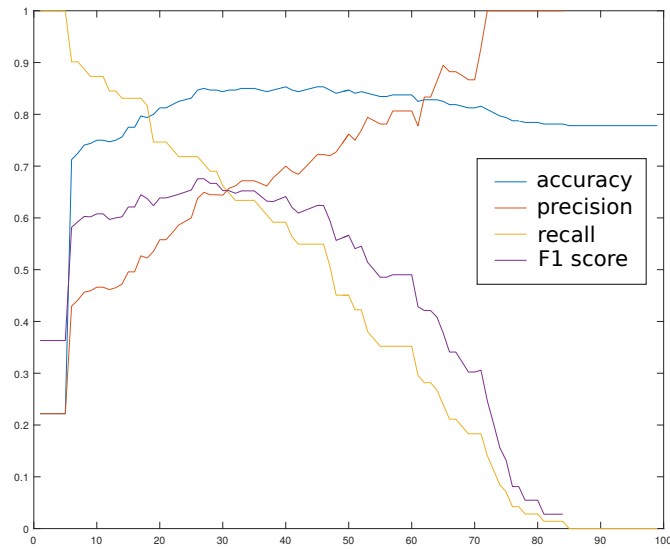
## 5.3. MACHINE LEARNING

### 5.3.3 Result

All training data are standardized by Eq. 5.15 where  $\bar{x}$  and  $\sigma$  is the mean and standard deviation of the data. It makes features in training data have zero means and unit variance. Cells that need to be refined are labeled as 1 and the rest are labeled as 0. Radial basis function with  $\sigma = 0.7624$  is adopted as kernel function in SVM.

$$x' = \frac{x - \bar{x}}{\sigma} \quad (5.15)$$

Half of the training data (320 out of 640) are used to train the model and the rest are used for cross validation. Different class weights are set for testing different performance in regard to all indicators in Fig. 5.6. A class



**Figure 5.6:** Accuracy, precision, recall rate and F1 score vs different class weight



## 5.4. MERGING TRIANGLE

---

weight is a vector that influence s the predication directly. The model will calculate the probability for each classification based on the input and the one with higher probability will be chosen as the result in default situation. Change class weight to 1 : 2 will force the model to choose first class when its probability is higher than 66.67% instead of 50%. A class weight of 3 : 7 was chosen from Fig. 5.6 to guarantee a balance between precision and recall rate. The corresponding result is listed in Tab. 5.1.

Table 5.1: Result of cross validation

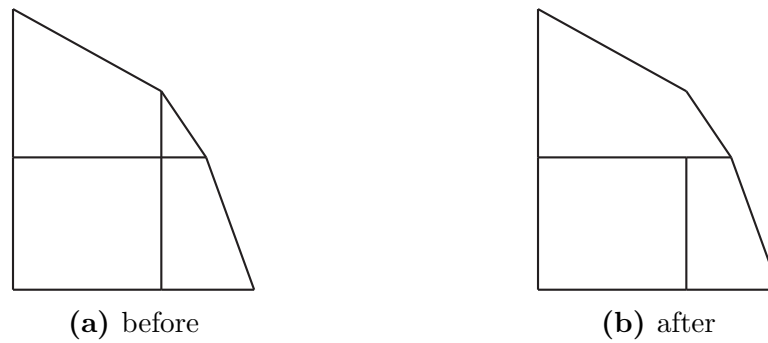
Accuracy	84.38%
Precision	64.38%
Recall rate	66.20%
F1 score	65.28%

## 5.4 Merging triangle

Due to the property of the SBFEM, first order triangle elements will result identical eigenvalues of one. Under this circumstance, the displacement, nodal force and energy using eigenvalues failed as they are always equal to the order of the shape function, one. As a consequence, the triangular elements shall be merged with their neighbours to form a larger polygon. Besides, due to the fact that triangular elements are formed by cutting with the geometric boundary, triangular elements have a higher chance to be obtuse triangles, which lead to poor quality mesh. As a result, merging triangle can help to improve the mesh quality as well.

## 5.4. MERGING TRIANGLE

---



**Figure 5.7:** Merging triangle

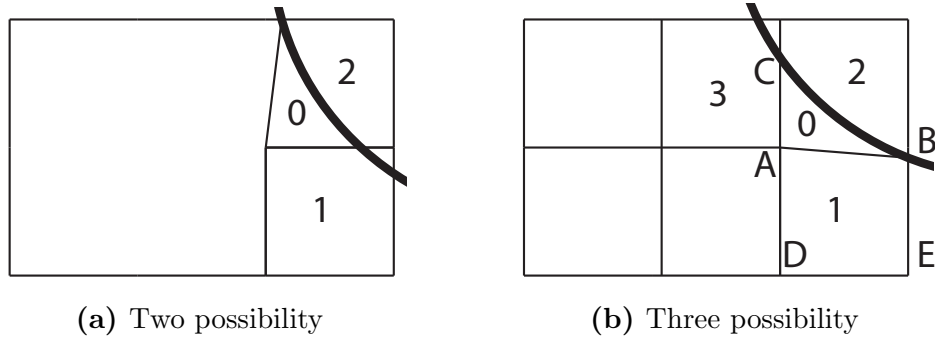
There are three steps to perform a triangle merging.

1. Finding the polygon to merge with
2. Merging triangle with the polygon
3. Adjusting scaling center

The first step of merging triangle is to find the appropriate cell to merge with. Generally speaking, there will be two or three candidate as shown in Fig. 5.8. Since triangular elements are always generated by cut by the boundary, one of these possibility must contain the other part of the origin cell (cell 2 in Fig. 5.8). However, as the cut by the boundary is necessary, undo the cut could be meaningless. As a result, this possibility can be ignored. If there are two remaining possibility as in Fig. 5.8b, the triangle will be merged with the cell that share the longer edge with it. Triangle 0 in Fig. 5.8b will be merged with cell 1 instead of 3 because  $AB > AC$ . It is because of the following reason. Take Fig. 5.8b as an example, edge  $AD$  will be extended to  $DC$  if triangle 0 is merged with polygon 1. So merging

## 5.4. MERGING TRIANGLE

triangle with the polygon that share a longer edge indicates the polygon will be less elongated and leads to a better mesh quality. Furthermore, the polygon to merge with the triangle must have the same material property with the triangular element, otherwise they shall not be combined together. In some extreme case, the merging triangle may not be possible as the triangular element itself fills the whole region with a specific material property. A reasonable selection of hyperparameter in generating mesh can prevent it from happening.



**Figure 5.8:** Choose the cell to merge with

The second step which is merging the triangular element with the polygon is straightforward. It should be noted that one point (point  $A$  in Fig. 5.8b) is possible to be removed. A calculation of the distance from point  $A$  to line  $CD$  can always help to determine whether the point can be removed or not.

The last step is to find the scaling center for the new generated polygon. Due to the fact that the merged polygon is very unlikely to be a poly-

#### 5.4. MERGING TRIANGLE

---

gon that looks like a square, taking the mean of all vertexes as the scaling center may not be a good idea. Calculating the geometric center of the polygon and take it as the scaling center may be an optimized solution. The centroid  $C$  of a polygon with  $n$  points  $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$  can be calculated as

$$\begin{aligned} C_x &= \frac{1}{6A} \sum_{i=0}^{n-1} (x_i + x_{i+1}) (x_i y_{i+1} - x_{i+1} y_i) \\ C_y &= \frac{1}{6A} \sum_{i=0}^{n-1} (y_i + y_{i+1}) (x_i y_{i+1} - x_{i+1} y_i) \end{aligned} \tag{5.16}$$

where  $A$  stands for the area of the polygon.

However, centroid of a concave polygon may be located at the outside of the polygon, result in an invalid scaling center. Consequently, special treatment is necessary to adjust the scaling center. If the mesh in the previous steps is correct, there should not be more than one reflexive angle in the polygon. So the vertex of that reflexive angle must be found first. It can be easily done by check the cross product between any two adjacent edges as vectors and find the only one whose sign is different with others. An angle bisector then is drawn and the intersection between it with the edge of the polygon will be recorded (point  $B$  in Fig. 5.9). The mid point of line  $AB$  will be select as the scaling center of that concave polygon.

## 5.5. MATRIX REPRESENTATION OF NURBS CURVES

---

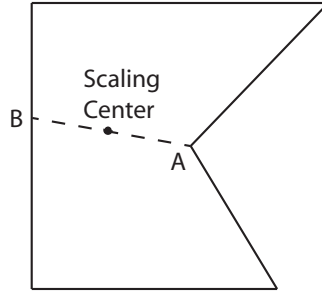


Figure 5.9: Scaling center for concave polygon

## 5.5 Matrix representation of NURBS Curves

### 5.5.1 Implicit Matrix Representation

Matrix representation of a parameterized algebraic curve allows an easy way to calculate the intersection between it with another curve and find the corresponding parameter based on the given point on the curve. For an algebraic curve  $t \in \mathbf{R}^1 \xrightarrow{\phi} \left( \frac{f_1(t)}{f_0(t)}, \frac{f_2(t)}{f_0(t)}, \frac{f_3(t)}{f_0(t)} \right) \in \mathbf{R}^3$ ,  $f_0, f_1, f_2$  and  $f_3$  are polynomials functions in parameter  $t$  with degree  $\leq p$ . The procedure of constructing the matrix representation for NURBS curves are explained detail in (Laurent, 2014).

The aim of this method is to find 4-tuples of polynomials  $(g_0(t), g_1(t), g_2(t), g_3(t))$  with order  $v$  so that

$$\sum_{i=0}^3 g_i(t) f_i(t) \equiv 0 \quad (5.17)$$

## 5.5. MATRIX REPRESENTATION OF NURBS CURVES

---

who is a vector space and one of its bases can be:

$$\mathbf{L}_j(t, X, Y, Z) = g_0(t) + Xg_1(t) + Yg_2(t) + Zg_3(t) \quad (5.18)$$

Since  $g$  is also a polynomial based function, it can be expressed in the vector space with a set of bases of  $\{\psi_1(t), \psi_2(t), \dots, \psi_{m_v}(t)\}$  and the bases  $\mathbf{L}_j$  can be expressed as

$$\begin{aligned} \mathbf{L}_j &= \sum_{i=1}^{m_v} \left( \lambda_{0,i}^{(j)} + \lambda_{1,i}^{(j)}X + \lambda_{2,i}^{(j)}Y + \lambda_{3,i}^{(j)}Z \right) \psi_i(t) \\ &= \sum_{i=1}^{m_v} \Lambda_{i,j}(X, Y, Z) \psi_i(t) \end{aligned} \quad (5.19)$$

Finally, a matrix which represents the mapping of  $\phi$  in a  $m_v \times r_v$ -matrix  $\mathbf{M}_v$  with order  $v$

$$\mathbf{M}_v(\phi) = \begin{bmatrix} \Lambda_{1,1} & \Lambda_{1,2} & \dots & \Lambda_{1,r_v} \\ \Lambda_{2,1} & \Lambda_{2,2} & \dots & \Lambda_{2,r_v} \\ \vdots & \vdots & & \vdots \\ \Lambda_{m_v,1} & \Lambda_{m_v,2} & \dots & \Lambda_{m_v,r_v} \end{bmatrix} \quad (5.20)$$

### 5.5.2 Matrix Representation for Rational Bzier Curves

An rational bzier curves can be defined by

$$\phi : t \in \mathbf{R} \rightarrow \frac{\sum_{i=0}^p w_i \mathbf{P}_i B_i^p(t)}{\sum_{i=0}^p w_i B_i^p(t)} \quad (5.21)$$

where

$$B_i^p(t) = \mathbf{C}_i^d t^i (1-t)^{d-i} \quad (5.22)$$

## 5.5. MATRIX REPRESENTATION OF NURBS CURVES

---

The aim is to find a matrix whose vector is in the form of

$$[\alpha] = \begin{bmatrix} \alpha_{0,0} & \alpha_{0,1} & \dots & \alpha_{0,v} & \alpha_{1,0} & \dots & \alpha_{3,v} \end{bmatrix}^T \quad (5.23)$$

where  $g_j(t)$  in Eq. 5.18 can be expressed as

$$g_j(t) = \sum_{i=0}^v \alpha_{j,i} B_i^v(t) \quad (5.24)$$

Based on Eq. (5.17), it can be concluded that  $\mathbf{R} \times [\alpha] = 0$

$$\mathbf{R} = \begin{bmatrix} B_0^v(t)f_0(t) & \dots & B_v^v(t)f_0(t) & B_0^v(t)f_1(t) & \dots & B_v^v(t)f_3(t) \end{bmatrix} \quad (5.25)$$

By having another set of basis  $\mathbf{L}_v$  and the transformation matrix  $\mathbf{S}$  so that  $\mathbf{L}_v \mathbf{S} = \mathbf{R}$  where

$$\mathbf{L}_v = \begin{bmatrix} B_0^{v+d}(t) & B_1^{v+d}(t) & \dots & B_{v+d}^{v+d}(t) \end{bmatrix} \quad (5.26)$$

This leads to

$$\begin{bmatrix} B_0^{v+d}(t) & B_1^{v+d}(t) & \dots & B_{v+d}^{v+d}(t) \end{bmatrix} \times \mathbf{S} \times [\alpha] = \mathbf{R} \times [\alpha] = 0 \quad (5.27)$$

which indicates  $[\alpha]$  is in the null space of  $\mathbf{S}$

After substituting  $f(t) = \sum_{i=0}^d c_i B_i^d(t)$  into  $\mathbf{R}$ , the following can be deduced

$$B_j^v(t)f(t) = \sum_{i=0}^d c_i B_i^d(t) B_j^v(t) = \sum_{i=0}^d \frac{\mathbf{C}_j^v \mathbf{C}_i^d}{\mathbf{C}_{i+j}^{d+v}} c_i B_{i+j}^{d+v}(t) \quad (5.28)$$

## 5.5. MATRIX REPRESENTATION OF NURBS CURVES

---

which indicates that

$$S_{i+j,j} = \frac{C_j^v C_i^d}{C_{i+j}^{d+v}} C_i \quad (5.29)$$

Finally, the null space of  $S_v$ ,  $M_v$  is the matrix representation of the rational bezier curve.

### 5.5.3 Intersection

The calculation of the intersection is described in detail in (Ba et al., 2009; Busé and Luu Ba, 2010). All intersections can be calculated at once by using matrix representation of the algebraic curve.

Given a rational curve/surface  $C1$

$$\mathbf{P}^1 \xrightarrow{\phi_1} \mathbf{P}^n : (u, v) \rightarrow (f_0, f_1, f_2, f_3)(u, v) \quad (5.30)$$

the aim is to find the intersection between it with another rational curve  $C2$  via matrix representation

$$\mathbf{P}^1 \xrightarrow{\phi_2} \mathbf{P}^n : (t) \rightarrow (g_0, g_1, g_2, g_3)(t) \quad (5.31)$$

is to find

$$\mathbf{M}_{v1}(\phi_2(t)) = 0 \quad (5.32)$$

which leads to

$$\mathbf{M}_0 g_0 + \mathbf{M}_1 g_1 + \mathbf{M}_2 g_2 + \mathbf{M}_3 g_3 = 0 \quad (5.33)$$



## 5.5. MATRIX REPRESENTATION OF NURBS CURVES

---

By knowing  $g_n$  is a polynomial function with order  $p$ , Eq. (6.20) can be rearranged as

$$\mathbf{M}(t) = \sum_{i=0}^p \mathbf{M}_i t^i \quad (5.34)$$

After that, the generalized companion  $q \times p$ -matrices  $A, B$  with rank  $\rho$  are introduced

$$A = \begin{bmatrix} 0 & I & \dots & \dots & 0 \\ 0 & 0 & I & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & \dots & I \\ M_0^t & M_1^t & \dots & \dots & M_{d-1}^t \end{bmatrix} \quad (5.35)$$

$$B = \begin{bmatrix} I & 0 & \dots & \dots & 0 \\ 0 & I & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & I & 0 \\ 0 & 0 & \dots & \dots & -M_d^t \end{bmatrix} \quad (5.36)$$

Before the eigenvalues are calculated, the regular part of a non-square pencil of the matrices shall be extracted first which is done by the following step

## 5.5. MATRIX REPRESENTATION OF NURBS CURVES

---

**Step 1** Transform  $B$  into its column echelon form: SVD-decomposition is adopted to perform the task.

$$\begin{aligned} B_1 = BV_0 &= [\underbrace{B_{1,1}}_{\rho} \mid \underbrace{0}_{q-\rho}] \\ A_1 = AV_0 &= [\underbrace{A_{1,1}}_{\rho} \mid \underbrace{A_{1,2}}_{q-\rho}] \end{aligned} \quad (5.37)$$

**Step 2** Transform  $A_{1,2}$  into its row echelon form:

$$U_1 A_{1,2} = \begin{bmatrix} A'_{1,2} \\ 0 \end{bmatrix} \quad (5.38)$$

where  $A'_{1,2}$  is in full row rank.

At the end of step 2, matrix  $A$  and  $B$  can be represented as

$$\begin{aligned} A'_1 &= \begin{bmatrix} A'_{1,1} & A'_{1,2} \\ A_2 & 0 \end{bmatrix} \\ B'_1 &= \begin{bmatrix} B'_{1,1} & 0 \\ B_2 & 0 \end{bmatrix} \end{aligned} \quad (5.39)$$

where  $A'_{1,2}$  has full row rank

$$\begin{bmatrix} B'_{1,1} \\ B_2 \end{bmatrix} \text{ has full column rank}$$

$$\begin{bmatrix} B'_{1,1} \\ B_2 \end{bmatrix} \text{ and } B_2 \text{ are in echelon form}$$

## 5.5. MATRIX REPRESENTATION OF NURBS CURVES

---

$A_2$  and  $B_2$  will be the new  $A$  and  $B$  matrices for next iteration until  $B$  has full rank. If  $B$  has full row rank but not full rank,  $A = A^T$  and  $B = B^T$  are conducted.

After these processes,  $A$  and  $B$  become two square matrices and  $B$  is invertible so that the solution for the intersection parameter  $t$  can be determined from the eigenvalue of the matrix  $AB^{-1}$

However, the method may fail when the intersection is under the case where nearly tangential geometric conditions happens and return two empty matrices. It is addressed by adding another step after extracting the real part of the  $A$  and  $B$  if the results are empty matrices(Shen et al., 2016). If the input matrix  $A$  and  $B$  are not in full row rank or full column rank, it is considered that  $C1 \cap C2 = C2$ . If the input matrix  $A$  and  $B$  are in full row rank or full column rank, a rank  $m$  square sub-pencil is extracted assuming  $A$  and  $B$  have a rank of  $m$ . Then the eigenvalues yield the intersections.

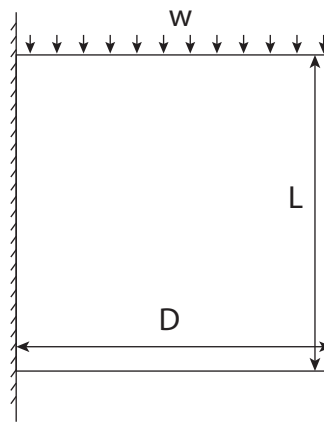
## 5.6. NUMERICAL EXAMPLES

---

### 5.6 Numerical examples

#### 5.6.1 Short cantilever beam

A two-dimensional short cantilever beam subjected to a uniformly distributed load at the top is examined as shown in Fig. 5.10.



**Figure 5.10:** Short cantilever beam: Geometry and boundary conditions.

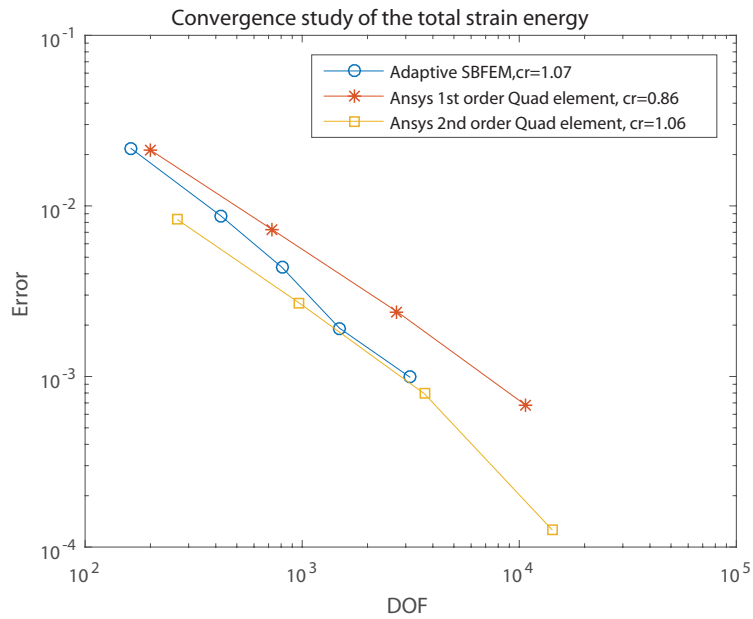
The geometry is: length  $L = 1$  m, height  $D = 1$  m. The material properties are: Young's modulus  $E = 20 \text{ N/m}^2$ , Poisson's ratio  $\nu = 0.3$ . The uniformly distributed load is  $w = 10 \text{ N m}^{-1}$ . Plane stress condition is assumed.

The reference strain energy of 4.020 79 J is determined by the help of the ANSYS. In the ANSYS, a mesh with 17930 DOF using 2<sup>nd</sup> order plane element 183 is used to calculate the result.

Due to the fact that the geometry of the cantilever beam can be described by four points and four straight lines, drawing in AutoCAD may not be necessary. As a result, the input geometry is defined manually.

## 5.6. NUMERICAL EXAMPLES

The numerical convergence of the the relative error in the energy norm is shown in Fig. 5.11. It can be observed that data mining based adaptive SBFEM yields superior convergence rate when compared to the result calculated in ANSYS.



**Figure 5.11:** the relative error in the energy norm

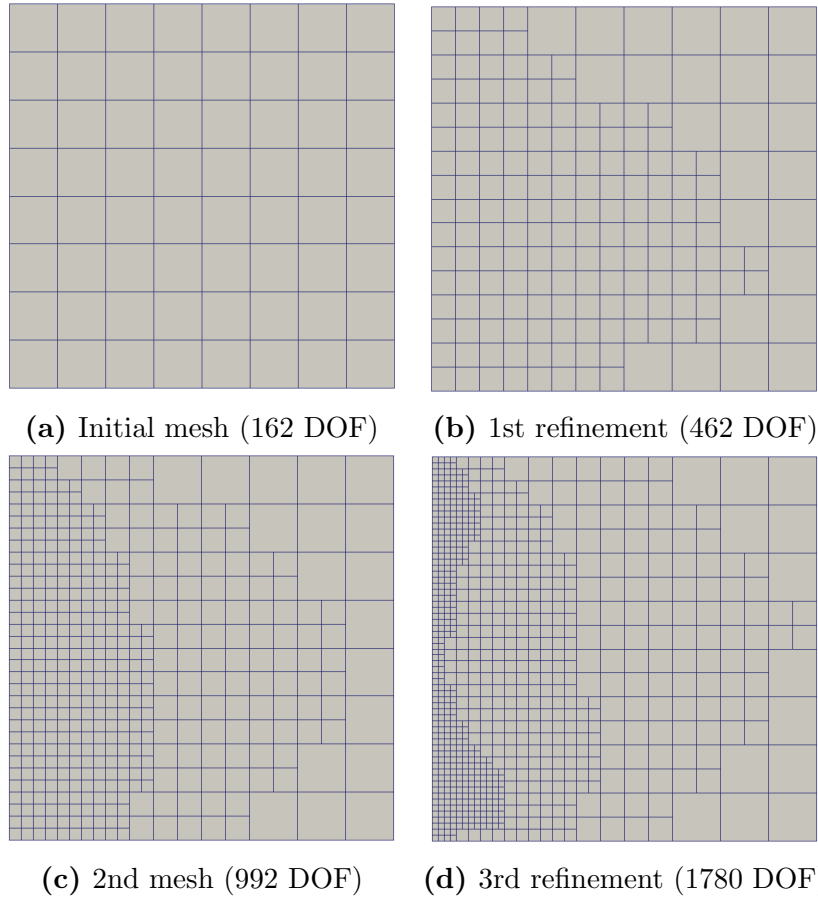
Corresponding mesh development are plotted in Fig. 5.12 (SBFEM 1st order element) and Fig. 5.13 (ANSYS 9-node quadrilateral element). Stress contour plotted in ANSYS using 9-node quadrilateral elements (17930 DOFs) are shown in Fig. 5.19

### 5.6.2 Infinite plate with a circular hole

In this example, an infinite plate with a traction free hole under uni-axial tension ( $\sigma = 1 \text{ N/m}^2$ ) along y-axis (see Fig. 5.15) is considered.

## 5.6. NUMERICAL EXAMPLES

---



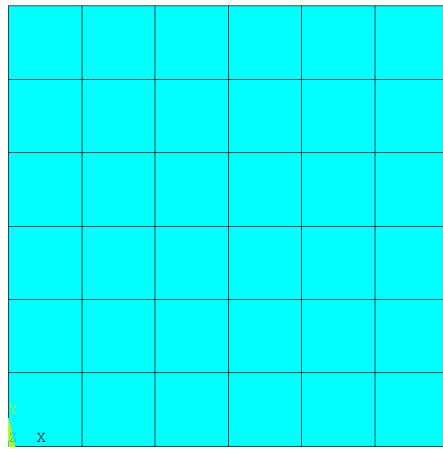
**Figure 5.12:** Short cantilever beam: mesh development (SBFEM)

Owing to symmetry, only one quarter of the plate is modeled. In this example, uniform distributed loads are applied on the top and plane stress condition is assumed. The bottom and right boundaries are enforced with roller boundary conditions.  $u_y = 0$  where  $y = 0$  and  $u_x = 0$  where  $x = 0$ .

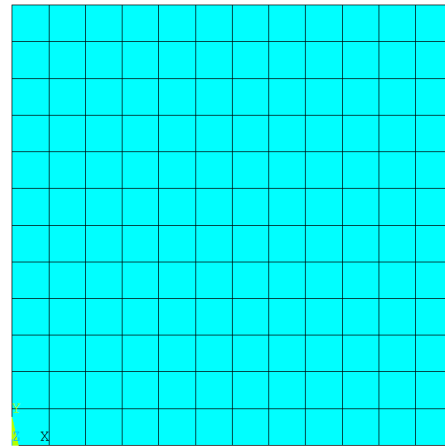
The rate of convergence in terms of the total strain energy is shown in Fig. 5.16 and corresponding mesh development are presented in Fig. 5.17 (SBFEM) and Fig. 5.18 ( Ansys 4-node quadrilateral element ). Higher

## 5.6. NUMERICAL EXAMPLES

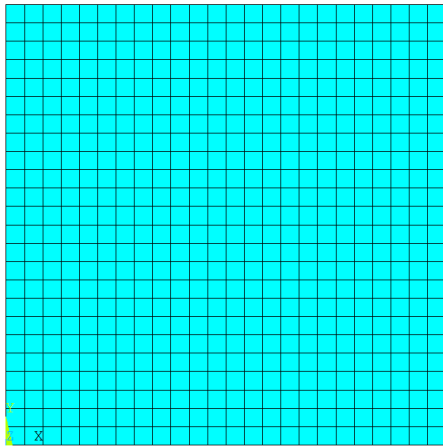
---



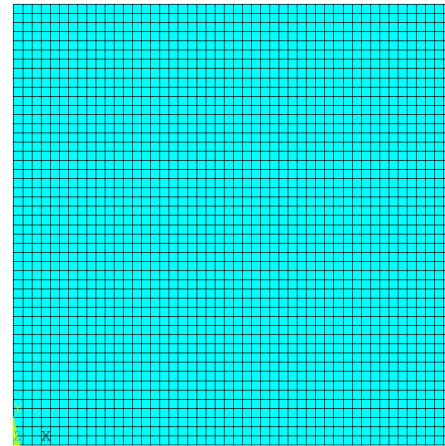
(a) Initial mesh, 266 DOFs



(b) 1st refinement, 962 DOFs



(c) 2nd refinement, 3650 DOFs



(d) 3rd refinement, 14210 DOFs

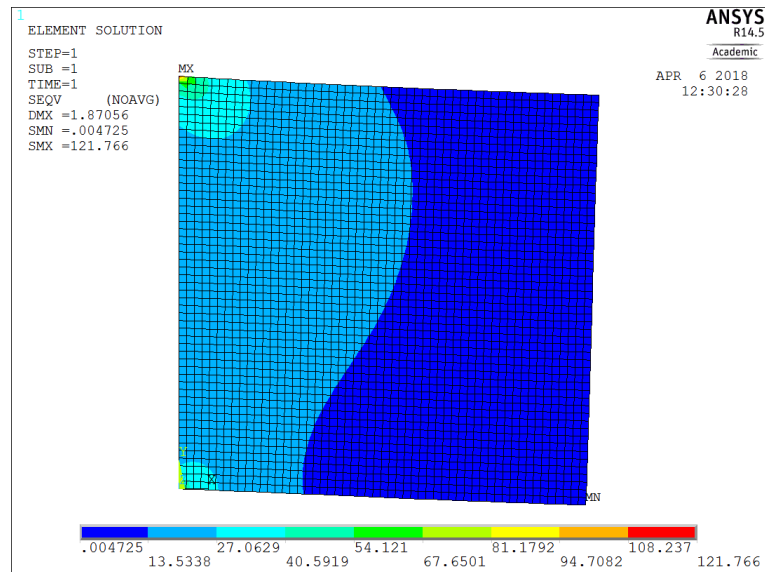
**Figure 5.13:** Short cantilever beam: mesh development (Ansys)

convergence rate compared to the result determined in ANSYS is observed.

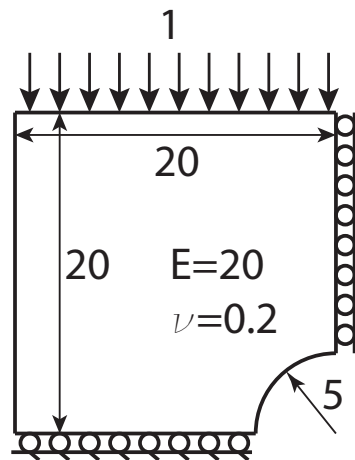
### 5.6.3 Plane strain bracket

In this example, a plane strain bracket with a downward uniform distributed load on the top is considered (see Fig. 5.20). The material prop-

## 5.6. NUMERICAL EXAMPLES



**Figure 5.14:** Von-mises stress contour using 9-node quadrilateral element in ANSYS (17930 DOFs)



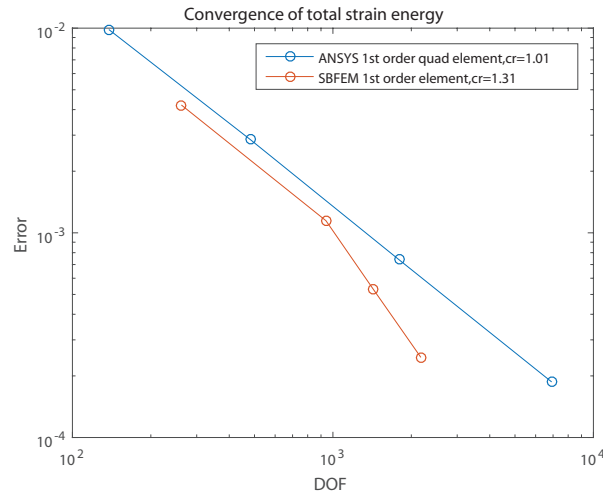
**Figure 5.15:** Infinite plate with a circular hole: geometry and boundary conditions

erties are: Youngs modulus  $E = 2 \times 10^5 \text{ N m}^{-2}$  and Poissons ratio  $\nu = 0.3$ .



## 5.7. CONCLUSIONS

---



**Figure 5.16:** Infinite plate with a circular hole: Convergence study

A total strain energy of 282.927 J is determined by ANSYS with the mesh shown in Fig. 4.39

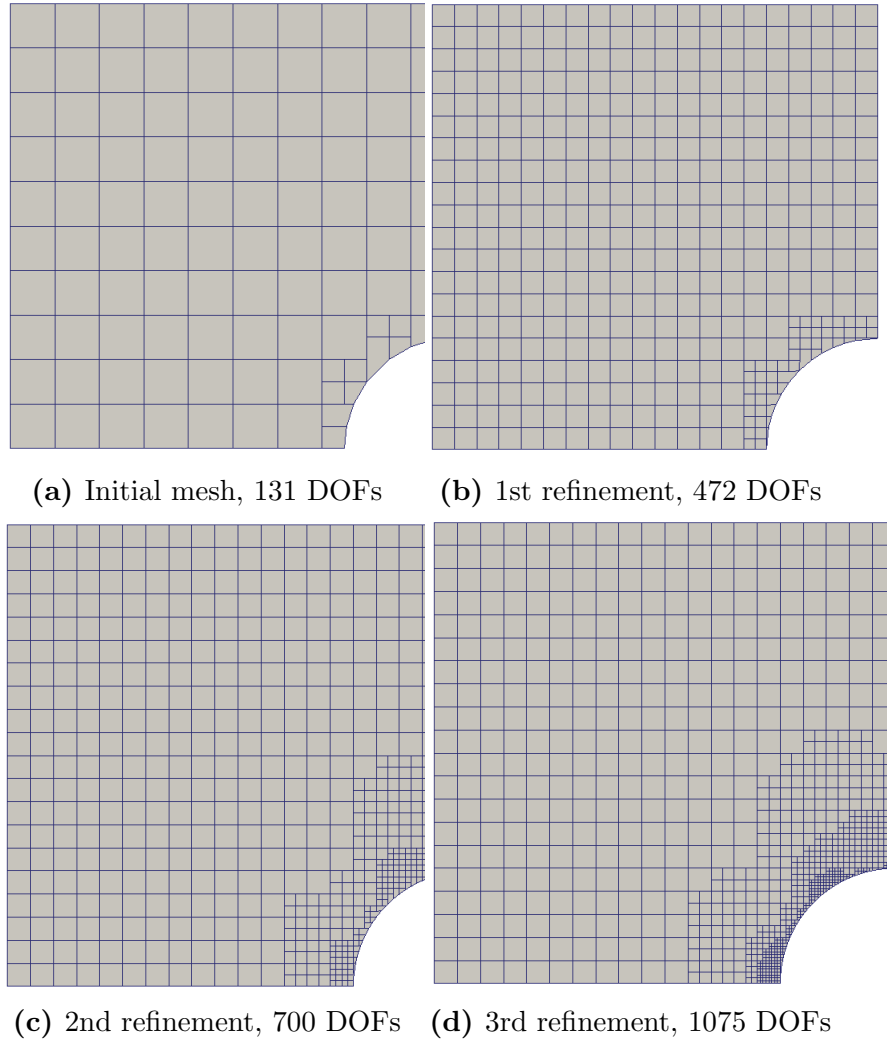
The rate of convergence in terms of the total strain energy is shown in Fig. 5.21 and corresponding mesh development are presented in Fig. 5.22 (SBFEM) and Fig. 5.23 ( Ansys 4-node quadrilateral element ). Higher convergence rate compared to the result determined in ANSYS is observed.

## 5.7 Conclusions

In this chapter, the machine learning algorithm is adopted to develop an extensible and flexible error indicator. Any other error estimators can be added to the existing framework and their effects can be detected

## 5.7. CONCLUSIONS

---

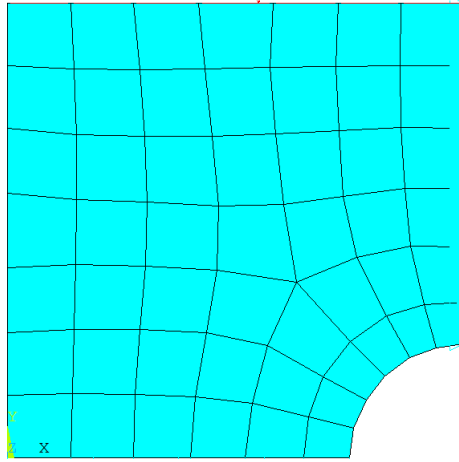


**Figure 5.17:** Infinite plate with a circular hole: Mesh development (SBFEM)

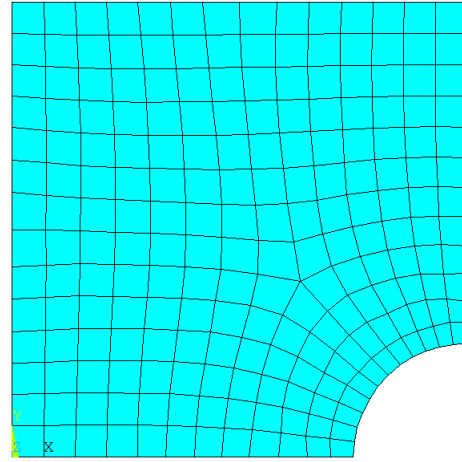
based on the performance indicators in machine learning. A MLP trained error estimator that concludes expressions related to the eigenvalues of the SBFEM formulation and some key geometric properties of the scaled boundary finite element give a higher convergence rate compared to the uniform refinement. In order to improve the learning effectiveness of the MLP, regularization methods including bagging and dropout are utilized.

## 5.7. CONCLUSIONS

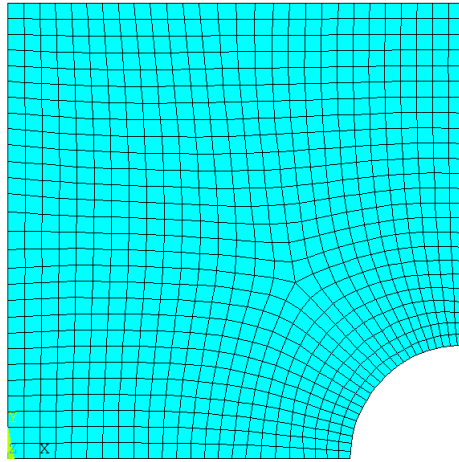
---



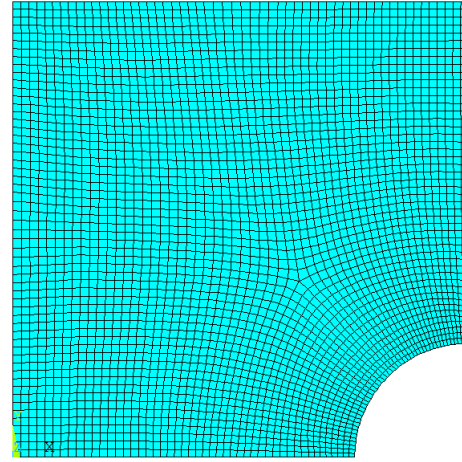
(a) Initial mesh, 138 DOFs



(b) 1st refinement, 482 DOFs



(c) 2nd refinement, 1794 DOFs

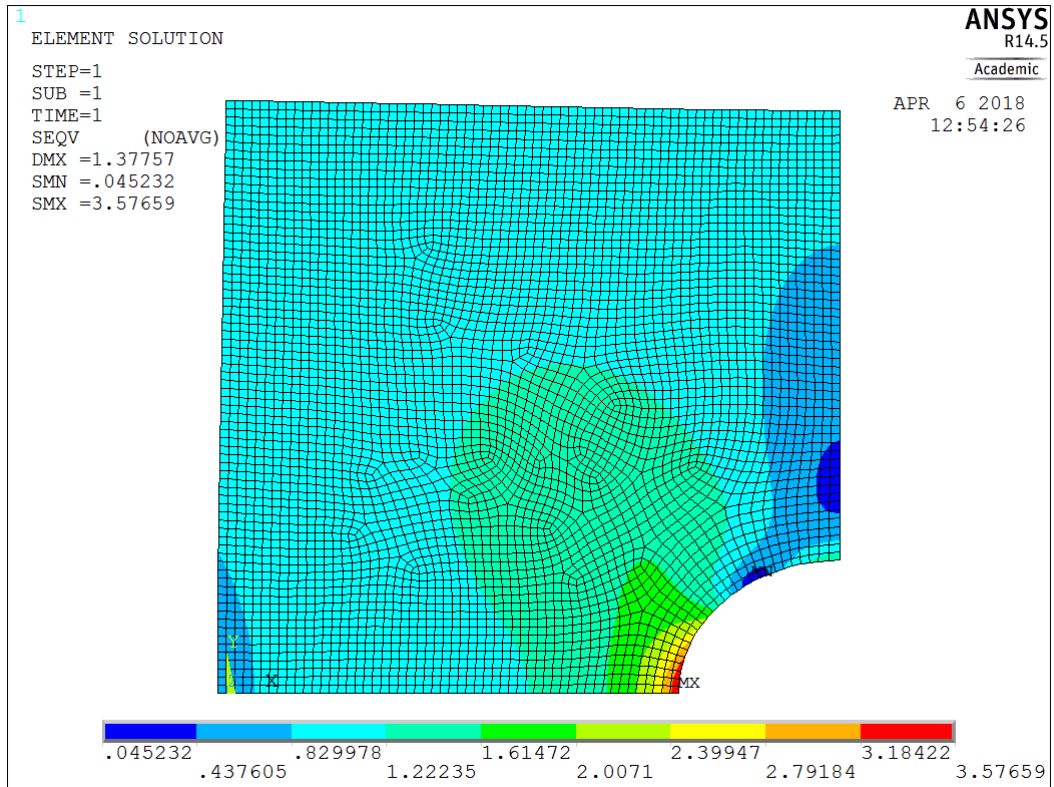


(d) 3rd refinement, 6914 DOFs

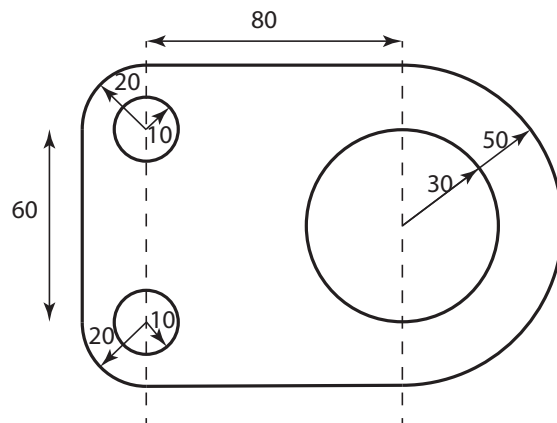
**Figure 5.18:** Infinite plate with a circular hole: Mesh development (Ansys)

Due to the lack of the eigenvalue error indicator in the first order triangular element, method that eliminates these situation is developed. A matrix representation of the NURBS curves is presented to achieve a higher efficiency and stability in calculating the intersections between edge  $s$  of the element and the NURBS curve. Stress analysis is conducted on 2D linear elasticity problems.

## 5.7. CONCLUSIONS

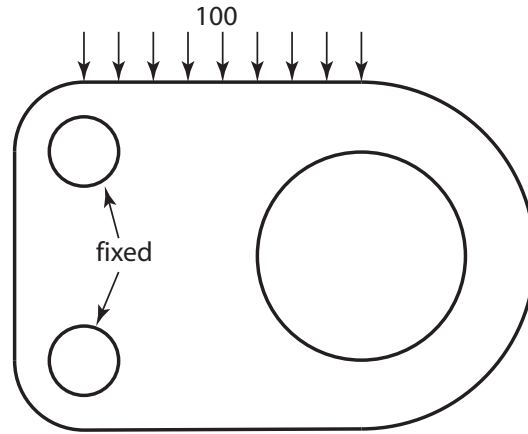


**Figure 5.19:** Von-mises stress contour using 9-node quadrilateral element in ANSYS (28450 DOFs)

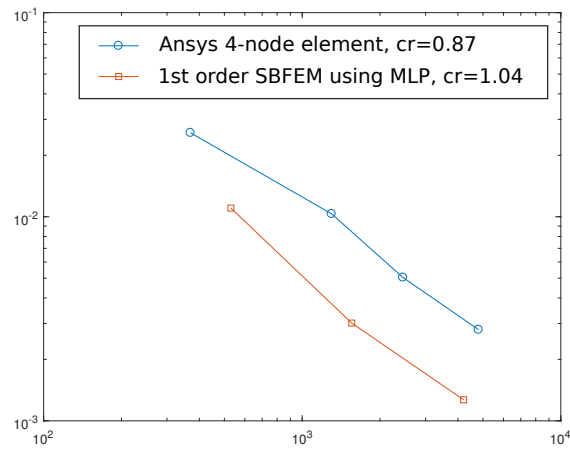


## 5.7. CONCLUSIONS

---



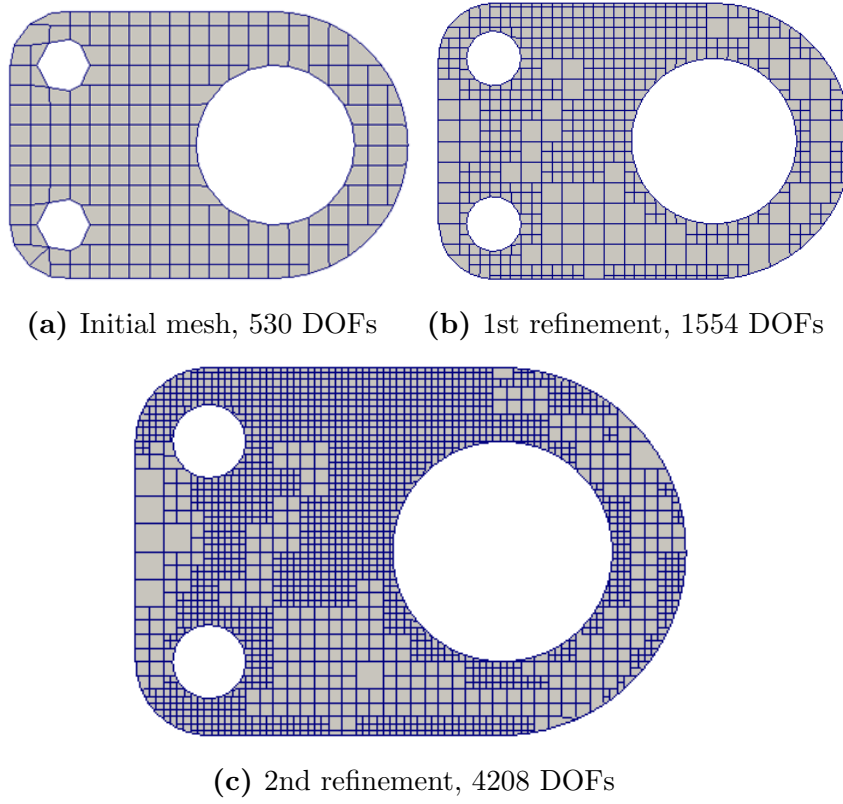
**Figure 5.20:** Plane strain bracket: geometry and boundary conditions



**Figure 5.21:** Plane strain bracket: Convergence study

## 5.7. CONCLUSIONS

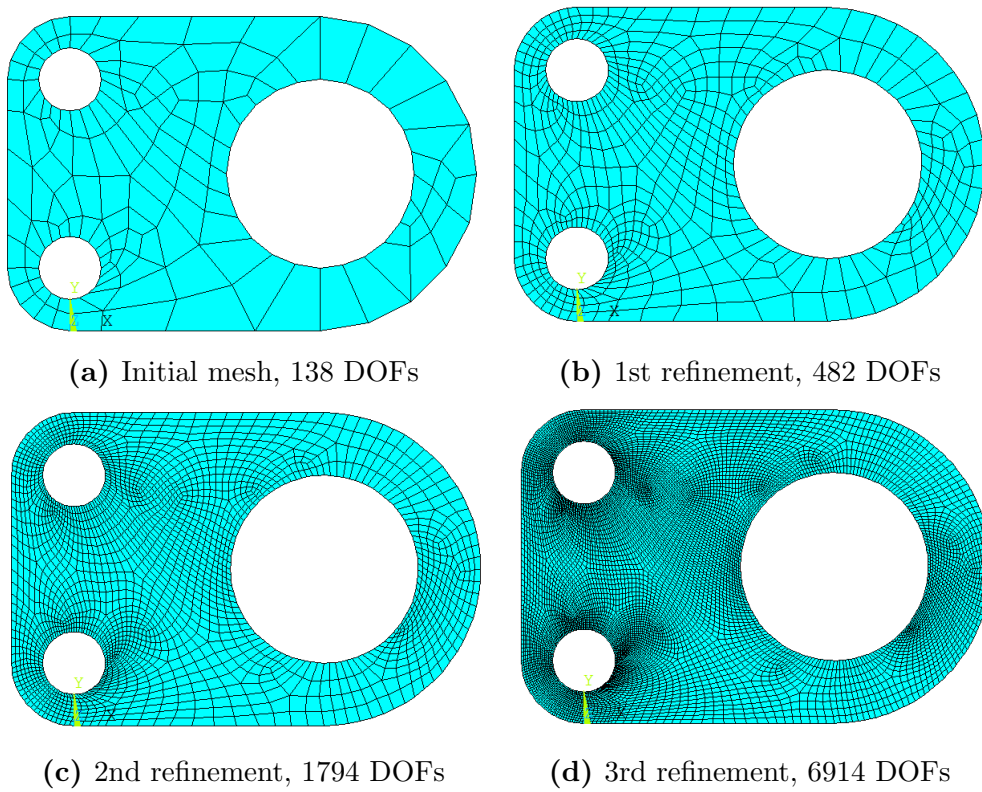
---



**Figure 5.22:** Plane strain bracket: Mesh development (SBFEM)

## 5.7. CONCLUSIONS

---



**Figure 5.23:** Plane strain bracket: Mesh development (Ansys)

## **Chapter 6**

# **Isogeometric enhanced SBFEM in 3D**

### **6.1 Introduction**

This chapter starts with a mesh generated by the help of the octree based algorithm using STL file directly (Liu et al., 2017). In this algorithm, the intersection is calculated between the edge of the element and the triangular surface which is an approximation of the exact geometry. In order to achieve the geometric precision, a point projection method for 3D NURBS surface is presented. Its computational efficiency is significantly improved by implementing a NURBS surface splitting algorithm and by utilizing the strong convex hull property of the NURBS. The quick hull algorithm is also introduced to construct the convex hull from the control points in 3D. Alternative method to retain the exact geometry is targeted as well.



## **6.2. PROJECTION**

---

The calculation of the intersection in Liu et al. (2017) is replaced by finding that between the edge of the element and the NURBS surface directly.

The advantage of the proposed method is that the exact geometry can be retained and hence improves the accuracy of the result.

This chapter will be organized as followed: points projection of the NURBS surface is introduced at the beginning, together with the surface splitting and the convex hull construction. After that, the calculation of the intersection of the straight line with the NURBS surface is developed. Furthermore, a brief introduction on SBFEM formulation in 3D elasticity is presented. The accuracy and the convergence properties of the proposed method are demonstrated with benchmark problems in the context of linear elasticity. Some other mesh examples from complex geometric input are also plotted at the end of this chapter.

## **6.2 Projection**

### **6.2.1 Surfaces division**

Mapping points back to NURBS surfaces in 3D can be extremely time consuming as there is no known closed form mathematical solution. Every point takes about ten to hundreds iterations before the nearest pro-

## 6.2. PROJECTION

---

jection point on the NURBS surface is found, depending on the size of the projection surface. However, in the problem that the proposed method is targeting, reasonably complex geometry is expected. As a result, points projection back to such kind of NURBS surfaces may take much more computational time than any others do and it may be necessary to find a more complicated but computational efficient algorithm other than the naive implementation.

One concept that can be utilized to improve the efficiency here is the “divided and conquer”. As the time complexity of the naive algorithm is  $O(n^3)$  where  $n$  is directly correlated to the order of the basis function and the number of control points used to describe the NURBS surfaces, dividing a surface into two generally will make the projection algorithm four times faster. Consequently, breaking the origin NURBS surfaces into multiple smaller ones could be one of the practical practices.

The surface division can be performed by the help of knot insertion (Sec. 2.1.2.1). Assuming a NURBS surface defined by two knot vectors

$$\Xi_1 = [-1, -1, -1, a_1, a_2, \dots, a_n, 1, 1, 1]$$

$$\text{and } \Xi_2 = [-1, -1, -1, b_1, b_2, \dots, b_m, 1, 1, 1].$$

Several knots will be inserted into these two vectors so that all interior knots will repeat  $p + 1$  times where  $p$  stands for the order of the NURBS basis function in that direction. After the knot insertion, the same NURBS surface will now be described by two new vectors

## 6.2. PROJECTION

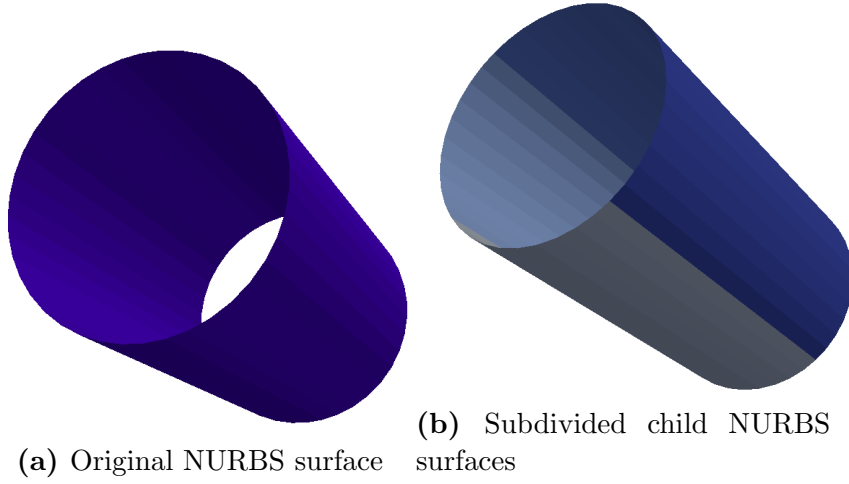
---

$$\Xi'_1 = [-1, -1, -1, a_1, a_1, a_1, a_2, a_2, a_2, \dots, a_n, 1, 1, 1]$$

$$\text{and } \Xi'_2 = [-1, -1, -1, b_1, b_1, b_1, b_2, b_2, b_2, \dots, b_m, 1, 1, 1].$$

Extraction then can be conducted by taking the sub-matrix from the generated control point matrix  $P'$  and weight matrix  $w'$ .

Fig. 6.1 shows a sub-division of breaking a cylinder surface into four smaller ones.



**Figure 6.1:** NURBS surface subdivision

### 6.2.2 Projection algorithm

In the proposed method, the mesh is generated by the method introduced in Liu et al. (2017) where intersections are calculated between the edge of the element and the triangular surface. These triangular surfaces extracted from the STL files are an approximation of the exact geometry. In order to retain the exact geometry, a point projection algorithm is required

## 6.2. PROJECTION

---

to find the nearest point in parameter  $(u, v)$  on the NURBS surface of the intersection point. This is achieved by finding a few possible sub-surfaces generated in Sec. 6.2.1 and finding the projection points  $P'$  of the test point  $P$  on these surfaces. The projection point  $P'_i$  with minimum  $|P'_i P|$  is adopted as the solution.

After all NURBS surfaces are divided into smaller ones, a filtering algorithm will be conducted for the test point  $P$  in order to limited the candidate sub-surfaces to only a few. The convex hull property of the NURBS surface introduced in Sec. 2.1.2 indicates that all points on the surface must be contained within the convex hull constructed by its control points. Considering the fact that the test point  $P$  is on the triangular surface of the STL file and the STL file is an approximation of the exact geometry, it can be concluded that the point  $P$  shall be also contained by the convex hull or be extremely close to it. As a consequence, convex hulls will be constructed for all sub-surfaces and those who do not contain the point  $P$  will be filtered. If the point  $P$  is not in any convex hull, distances between the point to them are calculated and 3 sub-surfaces whose convex hulls have the smallest distances to the point are selected as the candidates.

After that, the projection points of the test point  $P$  to each individual candidate sub-surface are calculated. Some  $O(\log N)$  algorithms were developed (Chin and Wang, 1983; Edelsbrunner, 1985) while presumably large number of tests on polygons is imposed. Furthermore, the accuracy

## 6.2. PROJECTION

---

is out of satisfactory for computer graphics and CAD communities. As a consequence, a projection algorithm (Ma and Hewitt, 2003) using Newton-Raphson method is introduced to tackle this problem.

For a given point  $P = (x, y, z)$ , its projection on the surface  $S(u, v)$  so that the distance  $|P - S(u, v)|$  is minimum is targeted. In order to find the projection, the vector  $r$  is defined as

$$\mathbf{r}(u, v) = \mathbf{S}(u, v) - \mathbf{P} \quad (6.1)$$

and two scalars  $f$  and  $g$  are defined as

$$\begin{cases} f(u, v) = \mathbf{r}(u, v) \cdot \mathbf{S}_u(u, v) = 0 \\ g(u, v) = \mathbf{r}(u, v) \cdot \mathbf{S}_v(u, v) = 0 \end{cases} \quad (6.2)$$

In order to solve Eq. 6.2, several notations are introduced

$$\begin{aligned} \delta_i &= \begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix} = \begin{bmatrix} u_{i+1} - u_i \\ v_{i+1} - v_i \end{bmatrix} \\ J_i &= \begin{bmatrix} f_u & f_v \\ g_u & g_v \end{bmatrix} = \begin{bmatrix} |S_u|^2 + r \cdot S_{uu} & S_u \cdot S_v + r \cdot S_{uv} \\ S_u \cdot S_v + r \cdot S_{uv} & |S_v|^2 + r \cdot S_{vv} \end{bmatrix} \\ \kappa_i &= - \begin{bmatrix} f(u_i, v_i) \\ g(u_i, v_i) \end{bmatrix} \end{aligned}$$

## 6.2. PROJECTION

---

Where all values in matrix  $J_i$  can be evaluated at  $(u_i, v_i)$ . 2 by 2 matrix  $\delta_i$  can be determined at step  $i$  as

$$J_i \delta_i = \kappa_i \quad (6.3)$$

It can be derived by utilizing  $\delta_i$  so that

$$u_{i+1} = u_i + \Delta u \quad (6.4a)$$

$$v_{i+1} = v_i + \Delta v \quad (6.4b)$$

The iteration can be concluded as

**1** Is the point coincide with  $S(u_i, v_i)$  ?

$$|S(u_i, v_i) - P| \leq \epsilon_1$$

where  $\epsilon_1$  stands for the tolerance for distance in Euclidean space.

**2** Is the cosine zero ?

$$\frac{|S_u(u_i, v_i) \cdot (S(u_i, v_i) - P)|}{|S_u(u_i, v_i)| |S(u_i, v_i) - P|} \leq \epsilon_2$$
$$\frac{|S_v(u_i, v_i) \cdot (S(u_i, v_i) - P)|}{|S_v(u_i, v_i)| |S(u_i, v_i) - P|} \leq \epsilon_2$$

where  $\epsilon_2$  stands for the tolerance for cosine. If either of these conditions are met, the iteration is terminated. Otherwise Eq. 6.4 is performed to find

## 6.2. PROJECTION

---

the parameters  $u_{i+1}$  and  $v_{i+1}$  for next iteration.

**3** Make sure  $u$  and  $v$  are within there domains

$$u_{i+1} \in [a, b]$$

$$v_{i+1} \in [c, d]$$

where  $a, b, c$  and  $d$  are the lower and upper bounds for the knot vectors of surface  $S$ . If the surface is open in  $u$  direction

$$\begin{cases} u_{i+1} = a & u_{i+1} < a \\ u_{i+1} = b & u_{i+1} > b \end{cases} \quad (6.5)$$

If the surface is open in  $v$  direction

$$\begin{cases} v_{i+1} = c & v_{i+1} < c \\ v_{i+1} = d & v_{i+1} > d \end{cases} \quad (6.6)$$

If the surface is closed in  $u$  direction

$$\begin{cases} u_{i+1} = b - (a - u_{i+1}) & u_{i+1} < a \\ u_{i+1} = a + (u_{i+1} - b) & u_{i+1} > b \end{cases} \quad (6.7)$$

If the surface is closed in  $v$  direction

$$\begin{cases} v_{i+1} = d - (c - v_{i+1}) & v_{i+1} < c \\ v_{i+1} = c + (v_{i+1} - d) & v_{i+1} > d \end{cases} \quad (6.8)$$

## 6.2. PROJECTION

---

4 The difference between the new parameters  $u_{i+1}$  and  $v_{i+1}$  and the old ones  $u_i$  and  $v_i$  is insignificant ?

$$|u(i+1) - u_i| \mathbf{S}_u(u_i, v_i) + (v_{i+1} - v_i) \mathbf{S}_v(u_i, v_i)| \leq \epsilon_1$$

The iteration will be terminated if this condition is met.

### 6.2.3 Convex hull in 3D

The convex hull property of the NURBS surface indicates that all points on the surface must be contained within the convex hull constructed by its control points (Selimovic, 2009) The algorithm in 3D can be very similar to that in 2D as introduced in Sec. 4.4.2.

1. Find the most left and right points (points with minimal and maximum  $x$ ) since they are proved to be part of the convex hull.
2. Connect these two points and use the line to separate other points into two group.
3. Find the point with maximum distance to the line in step 2 in any group.
4. Construct a triangle with two points in step 2 and the point in step 3.
5. Find the point with maximum distance to the triangle in step 4.
6. Construct a tetrahedron with the triangle in step 4 and the point in step 5.



### 6.3. INTERSECTION

---

7. Follow the same procedures in Sec. 4.4.2

## 6.3 Intersection

Instead of projecting all points located on the boundary surfaces to the NURBS surfaces, the exact geometry can be retained by calculating the intersection points directly. By utilizing the work from Sec. 6.2.1, the number of the surfaces which may contain the intersection can be limited to a few by determining the relationship between their convex hulls and the line. Only surfaces whose convex hull has intersection with the line will be selected as candidates. The number of the candidates increase when the mesh becomes coarser as a longer line segment tends to intersect more convex hulls. On the other hand, when the mesh is refined and the line segment becomes significantly shorter, the number of the candidates decreases at the same time. A considerable improvement in computational cost can be expected as the time complexity is only meaningful when numerous intersections are calculated. Only a few intersections need to be determined when the mesh is coarse hence having quite a few candidates at this circumstance may not significantly increase the computational time.

### 6.3. INTERSECTION

#### 6.3.1 Matrix Representation for Rational Bzier Surface

The intersection between a straight line and a NURBS surface is calculated using matrix representation method. The method in 2D (NURBS curve) has been introduced in Sec. 5.5.1. A tensor-product rational bzier surface of degree  $(p_1, p_2)$  can be expressed as

$$\phi(u, v) \in \mathbf{R}^2 \rightarrow \frac{\sum_{i=0}^{p_1} \sum_{j=0}^{p_2} w_{i,j} \mathbf{P}_{i,j} B_i^{p_1}(t) B_j^{p_2}(t)}{\sum_{i=0}^{p_1} \sum_{j=0}^{p_2} w_{i,j} B_i^{p_1}(t) B_j^{p_2}(t)} \quad (6.9)$$

For the surface,  $\mathbf{L}$  and  $\mathbf{R}$  with order  $(v_1, v_2)$

$$\mathbf{L} = \begin{bmatrix} B_0^{v_1+p_1}(u) B_0^{v_2+p_2}(v) & B_1^{v_1+p_1}(u) B_0^{v_2+p_2}(v) & \dots & B_{v_1+p_1}^{v_1+p_1}(u) B_{v_2+p_2}^{v_2+p_2}(v) \end{bmatrix} \quad (6.10a)$$

$$\mathbf{R} = \begin{bmatrix} B_0^{v_1}(u) B_0^{v_2}(v) f_0(u, v) & B_0^{v_1}(u) B_1^{v_2}(v) f_0(u, v) & \dots & B_{v_1}^{v_1}(u) B_{v_2}^{v_2}(v) f_3(u, v) \end{bmatrix} \quad (6.10b)$$

Following the same manner in Sec. 5.5.1, it can be derived that

$$S_{((i+k)(v_2+p_2+1)+j+l, l(v_1+1)+k)} = \frac{C_k^{v_1} C_l^{v_2} C_i^{p_1} C_j^{p_2}}{C_{i+k}^{v_1+d_2} C_{j+l}^{v_2+d_2}} C_{(i,j)} \quad (6.11)$$

#### 6.3.2 Properties of the $\mathbf{M}_v$ Matrix

As described in the previous sections, the  $\mathbf{M}_v$  matrix is defined so that

$$\begin{bmatrix} \psi_1(t_0) \dots \psi_{m_v}(t_0) \end{bmatrix} \times \mathbf{M}_v(\mathbf{P}) = \vec{0} \quad (6.12)$$

### 6.3. INTERSECTION

---

where  $\mathbf{P}$  is a point on the rational bzier curve/surface. The order  $v$  shall be no less than a critical value and it is proofed to be

- $v > \max(p - 1, 1)$  for rational bzier curve
- $(v_1, v_2) > (2p_1 - 1, p_2 - 1)$  or  $(v_1, v_2) > (p_1 - 1, 2p_2 - 1)$

The following properties are proofed in (Laurent, 2014)

1. For all degrees  $geq$  critical degree and all point  $\in \mathbf{R}^3$ ,  $\text{rank}(\mathbf{M}_v(\mathbf{P})) < m_v$  if and only if  $\mathbf{P} \in \overline{\text{Im}}(\phi)$ .
2. If  $\mathbf{P} \in \mathbf{R}^3$  is a point with a unique pre-image by  $\phi$ , the dimension of the null space of  $\mathbf{M}_v(\mathbf{P})^T$  is one.
3.  $\delta\mathbf{M}_v(\mathbf{P}) = 0$  if  $\mathbf{P} \in \overline{\text{Im}}(\phi)$

where

$$\delta\mathbf{M}_v(\mathbf{P}) = \prod_{i=1}^{m_v} \sigma_i(\mathbf{M}_v(\mathbf{P})) \quad (6.13)$$

and  $\sigma_i$  is the diagonal of  $\Sigma$  in the SVD decomposition of  $\mathbf{M}_v(\mathbf{P}) = U\Sigma V^T$

4.  $\forall \mathbf{P} \in \mathbf{R}^3, d(\mathbf{P}, \overline{\text{Im}}(\phi))^{n_1} \leq c_1 \delta\mathbf{M}_v(\mathbf{P})$
5.  $\forall \mathbf{P} \in \mathbf{R}^3, \delta\mathbf{M}_v(\mathbf{P})^{n_2} \leq c_2 d(\mathbf{P}, \overline{\text{Im}}(\phi))^{n_2}$

where  $c_1, c_2, n_1, n_2$  are constant.

These two properties give a distance function like function of the  $M_v$  matrix. When the point get away to the surface,  $\delta\mathbf{M}_v$  is getting larger and vice versa.

### 6.3. INTERSECTION

---

If the point is on the curve/surface, in which case  $\delta\mathbf{M}_v(\mathbf{P}) = 0$ , the corresponding parametric value on the curve/surface can be easily found by a SVD numerically.

The computation of the null space of  $\mathbf{M}_v(\mathbf{P})$  will give a single vector  $V = [v_1, v_2, \dots, v_{m_v}]$  based on 2 and  $V$  will be proportional to

$$\begin{bmatrix} \psi_1(t_0) \dots \psi_{m_v}(t_0) \end{bmatrix} \quad (6.14)$$

More specifically, it will be proportional to

$$\begin{bmatrix} B_0^v & B_1^v & \dots \end{bmatrix} \quad (6.15)$$

for the rational bzier curves and

$$\begin{bmatrix} B_0^{v_1} B_0^{v_2} & B_0^{v_1} B_1^{v_2} & \dots \end{bmatrix} \quad (6.16)$$

for the rational bzier surfaces. The calculation of the intersection is described in detail in (Busé and Luu Ba, 2010) and (Ba et al., 2009). All intersections can be calculated at once by using matrix representation of the algebraic curve/surface.

Given a rational curve/surface  $C1$

$$\mathbf{P}^1 \xrightarrow{\phi_1} \mathbf{P}^n : (u, v) \rightarrow (f_0, f_1, f_2, f_3)(u, v) \quad (6.17)$$

### 6.3. INTERSECTION

---

the aim is to find the intersection via matrix representation method between it with another ration curve  $C^2$

$$\mathbf{P}^1 \xrightarrow{\phi_2} \mathbf{P}^n : (t) \rightarrow (g_0, g_1, g_2, g_3)(t) \quad (6.18)$$

is to find

$$\mathbf{M}_{v1}(\phi_2(t)) = 0 \quad (6.19)$$

which leads to

$$\mathbf{M}_0 g_0 + \mathbf{M}_1 g_1 + \mathbf{M}_2 g_2 + \mathbf{M}_3 g_3 = 0 \quad (6.20)$$

By knowing  $g_n$  is a polynomial function with order  $p$ , Eq. (6.20) can be rearranged as

$$\mathbf{M}(t) = \sum_{i=0}^p \mathbf{M}_i t^i \quad (6.21)$$

After that, the generalized companion  $q \times p$ -matrices  $A, B$  with rank  $\rho$  are introduced

$$A = \begin{bmatrix} 0 & I & \dots & \dots & 0 \\ 0 & 0 & I & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & \dots & I \\ M_0^t & M_1^t & \dots & \dots & M_{d-1}^t \end{bmatrix} \quad (6.22)$$

### 6.3. INTERSECTION

---

$$B = \begin{bmatrix} I & 0 & \dots & \dots & 0 \\ 0 & I & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & I & 0 \\ 0 & 0 & \dots & \dots & -M_d^t \end{bmatrix} \quad (6.23)$$

Before the eigenvalues are calculated, the regular part of a non-square pencil of the matrices shall be extracted first which is done by the following step

**Step 1** Transform B into its column echelon form by SVD-decomposition.

$$\begin{aligned} B_1 = BV_0 &= \left[ \underbrace{B_{1,1}}_{\rho} \mid \underbrace{0}_{q-\rho} \right] \\ A_1 = AV_0 &= \left[ \underbrace{A_{1,1}}_{\rho} \mid \underbrace{A_{1,2}}_{q-\rho} \right] \end{aligned} \quad (6.24)$$

**Step 2** Transform  $A_{1,2}$  into its row echelon form

$$U_1 A_{1,2} = \begin{bmatrix} A'_{1,2} \\ 0 \end{bmatrix} \quad (6.25)$$

### 6.3. INTERSECTION

---

where  $A'_{1,2}$  is in full row rank.

At the end of step 2, matrix  $A$  and  $B$  can be represented as

$$\begin{aligned} A'_1 &= \begin{bmatrix} A'_{1,1} & A'_{1,2} \\ A_2 & 0 \end{bmatrix} \\ B'_1 &= \begin{bmatrix} B'_{1,1} & 0 \\ B_2 & 0 \end{bmatrix} \end{aligned} \quad (6.26)$$

where  $A'_{1,2}$  has full row rank

$$\begin{bmatrix} B'_{1,1} \\ B_2 \end{bmatrix} \text{ has full column rank}$$

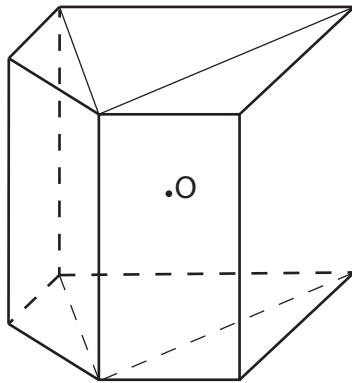
$$\begin{bmatrix} B'_{1,1} \\ B_2 \end{bmatrix} \text{ and } B_2 \text{ are in echelon form}$$

$A_2$  and  $B_2$  will be the new  $A$  and  $B$  matrices for next iteration until  $B$  has full rank. If  $B$  has full row rank but not full rank,  $A = A^T$ ,  $B = B^T$ .

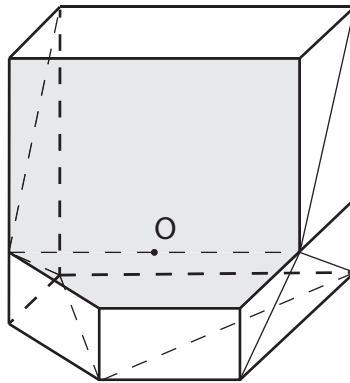
After these process,  $A$  and  $B$  become two square matrices and  $B$  is invertible so that the solution for the intersection parameter  $t$  can be determined from the eigenvalue of the matrix  $AB^{-1}$

## 6.4. INTRODUCTION OF SBFEM IN 3D

---



(a) Scaling center located inside of the subdomain



(b) Scaling center located on one of the edges

Figure 6.2: Polyhedral subdomain in SBFEM

## 6.4 Introduction of SBFEM in 3D

A polyhedral cell can be regarded as one of the subdomain  $s$  in the SBFEM as plotted in Fig. 6.2. Similar to SBFEM in 2D which has been introduced in Sec. 3.2, only the boundary surface requires discretization and a scaling center  $O$  located at the place where all boundary surfaces of the subdomain are visible is necessary. A scaling center on the edge as shown in Fig. 6.2b is also valid which increases the flexibility and allows the concave subdomain. In this situation, the triangulated faces containing the scaling center will not be discretized.



#### 6.4. INTRODUCTION OF SBFEM IN 3D

---

The transformation between SBFEM coordinate  $(\xi, \eta, \zeta)$  and Cartesian coordinate  $(x, y, z)$  can be derived from Eq. 3.1 and Eq. 3.2 as:

$$x(\xi, \eta, \zeta) = \xi \hat{x}(\eta, \zeta) = \xi \mathbf{N}(\eta, \zeta) \hat{\mathbf{x}} \quad (6.27a)$$

$$y(\xi, \eta, \zeta) = \xi \hat{y}(\eta, \zeta) = \xi \mathbf{N}(\eta, \zeta) \hat{\mathbf{y}} \quad (6.27b)$$

$$z(\xi, \eta, \zeta) = \xi \hat{z}(\eta, \zeta) = \xi \mathbf{N}(\eta, \zeta) \hat{\mathbf{z}} \quad (6.27c)$$

where  $(\hat{x}, \hat{y}, \hat{z})$  stands for the point in Cartesian coordinate on the boundary,  $(\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{z}})$  represents the nodal coordinate vectors and  $\mathbf{N}$  is the shape function. The corresponding Jacobian matrix in 3D on the boundary ( $\xi = 1$ ) can be expressed as:

$$\mathbf{J}(\eta, \zeta) = \begin{bmatrix} x(\eta, \zeta) & y(\eta, \zeta) & z(\eta, \zeta) \\ x(\eta, \zeta)_{,\eta} & y(\eta, \zeta)_{,\eta} & z(\eta, \zeta)_{,\eta} \\ x(\eta, \zeta)_{,\zeta} & y(\eta, \zeta)_{,\zeta} & z(\eta, \zeta)_{,\zeta} \end{bmatrix} \quad (6.28)$$

and the displacement interpolation in Eq. 3.4 becomes

$$\mathbf{u}(\xi, \eta, \zeta) = \mathbf{N}(\eta, \zeta) \mathbf{u}(\xi) \quad (6.29)$$

The linear operator in Eq. 3.6 becomes

$$\mathbf{L} = \mathbf{b}_1(\eta, \zeta) \frac{\partial}{\partial \xi} + \frac{1}{\xi} \left( \mathbf{b}_2(\eta, \zeta) \frac{\partial}{\partial \eta} + \mathbf{b}_3(\eta, \zeta) \frac{\partial}{\partial \zeta} \right) \quad (6.30)$$

#### 6.4. INTRODUCTION OF SBFEM IN 3D

---

Little  $b$  matrix in Eq. 3.7 becomes

$$\mathbf{b}_1(\eta, \zeta) = \frac{1}{|\mathbf{J}|} \begin{bmatrix} y_{,\eta} z_{,\zeta} - z_{,\eta} y_{,\zeta} & 0 & 0 \\ 0 & z_{,\eta} x_{,\zeta} - x_{,\eta} z_{,\zeta} & 0 \\ 0 & 0 & x_{,\eta} y_{,\zeta} - y_{,\eta} x_{,\zeta} \\ 0 & x_{,\eta} y_{,\zeta} - y_{,\eta} x_{,\zeta} & z_{,\eta} x_{,\zeta} - x_{,\eta} z_{,\zeta} \\ x_{,\eta} y_{,\zeta} - y_{,\eta} x_{,\zeta} & 0 & y_{,\eta} z_{,\zeta} - z_{,\eta} y_{,\zeta} \\ z_{,\eta} x_{,\zeta} - x_{,\eta} z_{,\zeta} & y_{,\eta} z_{,\zeta} - z_{,\eta} y_{,\zeta} & 0 \end{bmatrix} \quad (6.31a)$$

$$\mathbf{b}_2(\eta, \zeta) = \frac{1}{|\mathbf{J}|} \begin{bmatrix} y_{,\zeta} z - z_{,\zeta} y & 0 & 0 \\ 0 & z_{,\zeta} x - x_{,\zeta} z & 0 \\ 0 & 0 & x_{,\zeta} y - y_{,\zeta} x \\ 0 & x_{,\zeta} y - y_{,\zeta} x & z_{,\zeta} x - x_{,\zeta} z \\ x_{,\zeta} y - y_{,\zeta} x & 0 & y_{,\zeta} z - z_{,\zeta} y \\ z_{,\zeta} x - x_{,\zeta} z & y_{,\zeta} z - z_{,\zeta} y & 0 \end{bmatrix} \quad (6.31b)$$

$$\mathbf{b}_3(\eta, \zeta) = \frac{1}{|\mathbf{J}|} \begin{bmatrix} yz_{,\eta} - zy_{,\eta} & 0 & 0 \\ 0 & zx_{,\eta} - xz_{,\eta} & 0 \\ 0 & 0 & xy_{,\eta} - yx_{,\eta} \\ 0 & xy_{,\eta} - yx_{,\eta} & zx_{,\eta} - xz_{,\eta} \\ xy_{,\eta} - yx_{,\eta} & 0 & yz_{,\eta} - zy_{,\eta} \\ zx_{,\eta} - xz_{,\eta} & yz_{,\eta} - zy_{,\eta} & 0 \end{bmatrix} \quad (6.31c)$$

Follow the same manner in Sec. 3.2, the strains can be expressed as

$$\epsilon(\xi, \eta, \zeta) = \mathbf{B}_1(\eta, \zeta) \mathbf{u}(\xi)_{,\xi} + \frac{1}{\xi} \mathbf{B}_2(\eta, \zeta) \mathbf{u}(\xi) \quad (6.32)$$

#### 6.4. INTRODUCTION OF SBFEM IN 3D

---

where

$$\mathbf{B}_1(\eta, \zeta) = \mathbf{b}_1(\eta, \zeta)\mathbf{N}(\eta, \zeta) \quad (6.33a)$$

$$\mathbf{B}_2(\eta, \zeta) = \mathbf{b}_2(\eta, \zeta)\mathbf{N}(\eta, \zeta)_{,\eta} + \mathbf{b}_3(\eta, \zeta)\mathbf{N}(\eta, \zeta)_{,\xi} \quad (6.33b)$$

The stress in Eq. 3.9 becomes

$$\sigma(\xi, \eta, \zeta) = \mathbf{D} \left( \mathbf{B}_1(\eta, \zeta)\mathbf{u}(\xi)_{,\xi} + \frac{1}{\xi}\mathbf{B}_2(\eta, \zeta)\mathbf{u}(\xi) \right) \quad (6.34)$$

and the ODE equation in Eq. 3.13 becomes

$$\mathbf{E}_0\xi^2\mathbf{u}(\xi)_{,\xi\xi} + (2\mathbf{E}_0 + \mathbf{E}_1^T - \mathbf{E}_1)\xi\mathbf{u}(\xi)_{,\xi} - (\mathbf{E}_1^T - \mathbf{E}_2)\mathbf{u}(\xi) = 0 \quad (6.35)$$

The coefficient matrices  $\mathbf{E}_0$ ,  $\mathbf{E}_1$  and  $\mathbf{E}_2$  for element  $e$  are expressed as

$$\mathbf{E}_0 = \int_e \mathbf{B}_1^T \mathbf{D} \mathbf{B}_1 |\mathbf{J}| d\eta d\zeta \quad (6.36a)$$

$$\mathbf{E}_1 = \int_e \mathbf{B}_2^T \mathbf{D} \mathbf{B}_1 |\mathbf{J}| d\eta d\zeta \quad (6.36b)$$

$$\mathbf{E}_2 = \int_e \mathbf{B}_2^T \mathbf{D} \mathbf{B}_2 |\mathbf{J}| d\eta d\zeta \quad (6.36c)$$

The internal nodal forces  $\mathbf{q}(\xi)$  on the boundary can be determined as (Song, 2004)

$$\mathbf{q}(\xi) = \xi(\mathbf{E}_0\xi\mathbf{u}(\xi)_{,\xi} + \mathbf{E}_1^T\mathbf{u}(\xi)) \quad (6.37)$$

#### 6.4. INTRODUCTION OF SBFEM IN 3D

---

$X(\xi)$  is introduced to reduce the order of the ODE in Eq. 6.35 as

$$\mathbf{X}(\xi) = \begin{bmatrix} \xi^{0.5} \mathbf{u}(\xi) \\ \xi^{-0.5} \mathbf{q}(\xi) \end{bmatrix} \quad (6.38)$$

substituting Eq. 6.38 and Eq. 6.37 into Eq. 6.35 yields

$$\xi \mathbf{X}(\xi)_{,\xi} = -\mathbf{Z} \mathbf{X}(\xi) \quad (6.39)$$

where  $\mathbf{Z}$  is the hamiltonian matrix

$$\mathbf{Z} = \begin{bmatrix} \mathbf{E}_0^{-1} \mathbf{E}_1^T - 0.5 \mathbf{I} & -\mathbf{E}_0^{-1} \\ -\mathbf{E}_2 + \mathbf{E}_1 \mathbf{E}_0^{-1} \mathbf{E}_1^T & -(\mathbf{E}_1 \mathbf{E}_0^{-1} - 0.5 \mathbf{I}) \end{bmatrix} \quad (6.40)$$

An eigenvalue decomposition of Eq. 6.40 is performed and it yields

$$\mathbf{X}(\xi) = \begin{bmatrix} \Phi_{u1} & \Phi_{u2} \\ \Phi_{q1} & \Phi_{q2} \end{bmatrix} \begin{bmatrix} \xi^{-\lambda} & 0 \\ 0 & \xi^{\lambda} \end{bmatrix} \begin{bmatrix} \mathbf{c}_1 & 0 \\ 0 & \mathbf{c}_2 \end{bmatrix} \quad (6.41)$$

where  $\pm\lambda$  is the eigenvalue and  $-\lambda$  is corelate to the real parts.  $\Phi$  is part of the matrix of the eigenvector matrix.  $\mathbf{c}_1$  and  $\mathbf{c}_2$  stand for the integration constant. Modal displacements and forces in Eq. 3.22 becomes

$$\mathbf{u}(\xi) = \Phi_{u1} \xi^{-\Lambda-0.5\mathbf{I}} \mathbf{c}_1 \quad (6.42a)$$

$$\mathbf{q}(\xi) = \Phi_{q1} \xi^{-\Lambda+0.5\mathbf{I}} \mathbf{c}_2 \quad (6.42b)$$

## 6.5. NUMERICAL EXAMPLES

---

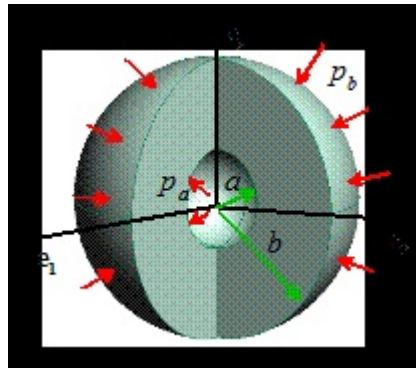
The stiffness matrix then can be derived as

$$\mathbf{K} = \Phi_{q1} \Phi_{u1}^{-1} \quad (6.43)$$

## 6.5 Numerical examples

### 6.5.1 Pressurized hollow sphere

The problem is a pressurized hollow sphere subjected to the internal pressure. The geometry of the problem is described in Fig.6.3.



**Figure 6.3:** Pressurized hollow sphere

In the example, the external pressure is set to be zero so that only the internal one is considered. The geometric properties are:  $a = 10$  m and  $b = 50$  m. The boundary conditions are:  $p_a = 10 \text{ N m}^{-2}$ ,  $p_b = 0$  and the rigid body motion is prevented. The material properties are:  $E = 200 \text{ N m}^{-2}$  and  $\nu = 0.3$ . For simplification, only a quarter of the sphere is analysed

## 6.5. NUMERICAL EXAMPLES

---

as shown in Fig. 6.5 Analytical surface traction is applied on all of the boundary surfaces. First order tetrahedral element is adopted to calculate the displacement and the stress and they are compared to the exact solution as in Eq. 6.44d in spherical coordinate.

$$u = \frac{1}{2E(b^3 - a^3)R^2} \{2(p_a a^3 - p_b b^3)(1 - 2\nu)R^3 + (p_a - p_b)(1 + \nu)b^3 a^3\} \quad (6.44a)$$

$$\sigma_{RR} = \frac{p_a a^3 - p_b b^3}{b^3 - a^3} - \frac{(p_a - p_b)b^3 a^3}{(b^3 - a^3)R^3} \quad (6.44b)$$

$$\sigma_{\theta\theta} = \frac{p_a a^3 - p_b b^3}{b^3 - a^3} + \frac{(p_a - p_b)b^3 a^3}{2(b^3 - a^3)R^3} \quad (6.44c)$$

$$\sigma_{\phi\phi} = \sigma_{\theta\theta} \quad (6.44d)$$

The tensor transformation from spherical coordinate to cartesian coordinate can be written as Eq. 6.45 according to Fig. 6.4.

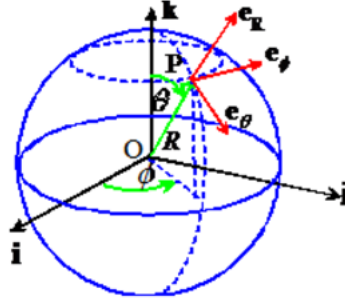
$$\begin{bmatrix} S_{xx} & S_{xy} & S_{xz} \\ S_{xy} & S_{yy} & S_{yz} \\ S_{xz} & S_{yz} & S_{zz} \end{bmatrix} = T \begin{bmatrix} S_{RR} & S_{R\theta} & S_{R\phi} \\ S_{R\theta} & S_{\theta\theta} & S_{\theta\phi} \\ S_{R\phi} & S_{\theta\phi} & S_{\phi\phi} \end{bmatrix} T^T \quad (6.45a)$$

$$T = \begin{bmatrix} \sin \theta \cos \phi & \cos \theta \cos \phi & -\sin \phi \\ \sin \theta \sin \phi & \cos \theta \sin \phi & \cos \phi \\ \cos \theta & -\sin \theta & 0 \end{bmatrix} \quad (6.45b)$$

Stress boundary condition in Eq. 6.46 is applied on two spherical

## 6.5. NUMERICAL EXAMPLES

---



**Figure 6.4:** Coordinate transformation

surfaces.

$$\sigma_{RR}(R = a, \phi, \theta) = \frac{p_a a^3 - p_b b^3}{b^3 - a^3} - \frac{(p_a - p_b) b^3 a^3}{(b^3 - a^3) R^3} \quad (6.46a)$$

$$u_z(x, y, 0) = 0 \quad (6.46b)$$

$$u_y(x, 0, z) = 0 \quad (6.46c)$$

$$u_x(0, y, z) = 0 \quad (6.46d)$$

The convergence study is plotted in Fig. 6.8 and the mesh with corresponding DOFs are plotted in Fig. 6.5, Fig. 6.6 and Fig. 6.7

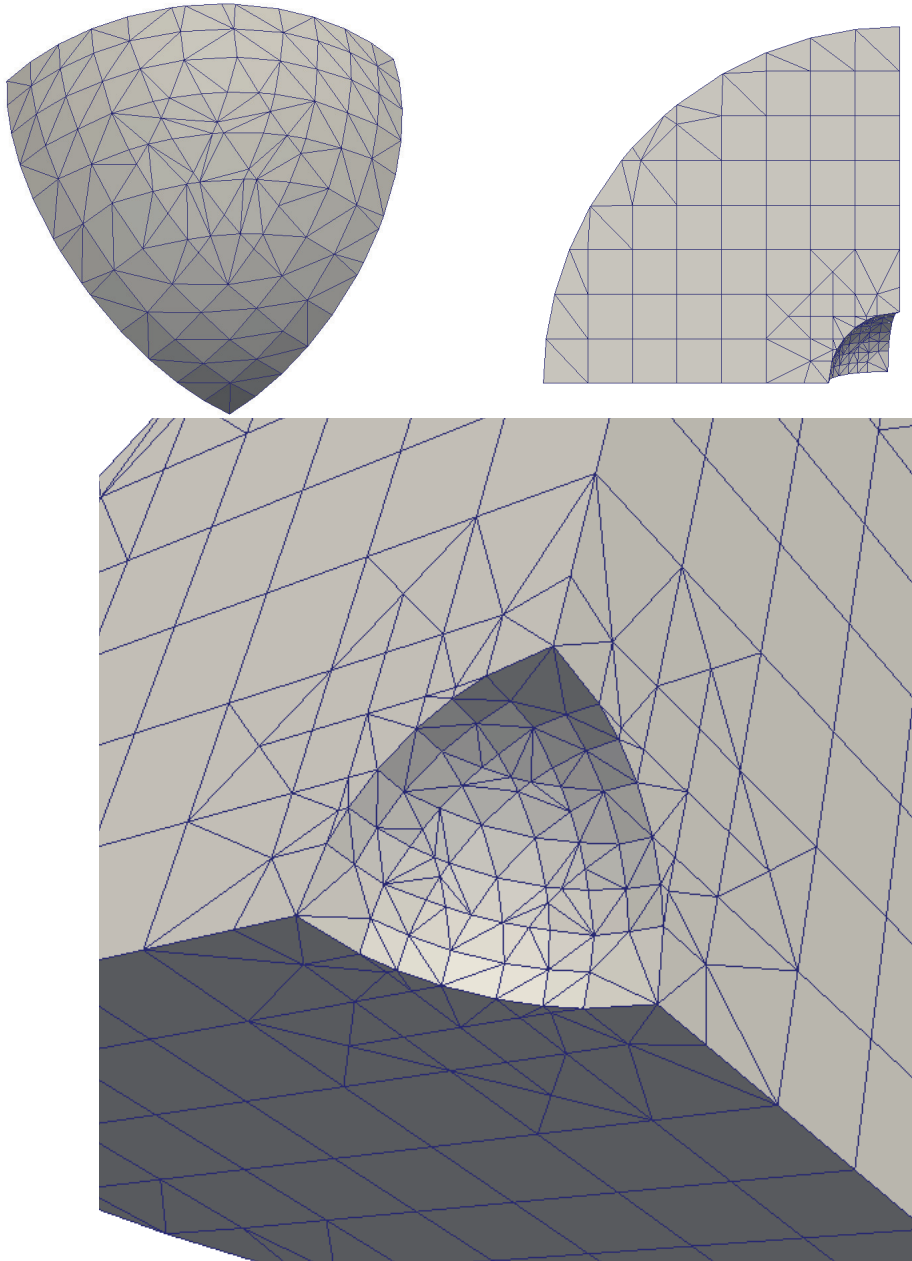
### 6.5.2 Capsule Cutting From the Cuboid with Bending

The example is a capsule section cutting from a cuboid with pure bending illustrated in Fig. 6.9 and the generated mesh is plotted in Fig. 6.11.

The displacement analytical solution (Timoshenko and Goodier, 1951) is applied on the outer surface of the capsule as the boundary condition and the displacement and the stress (Eq. 6.48) inside is compared to the

## 6.5. NUMERICAL EXAMPLES

---

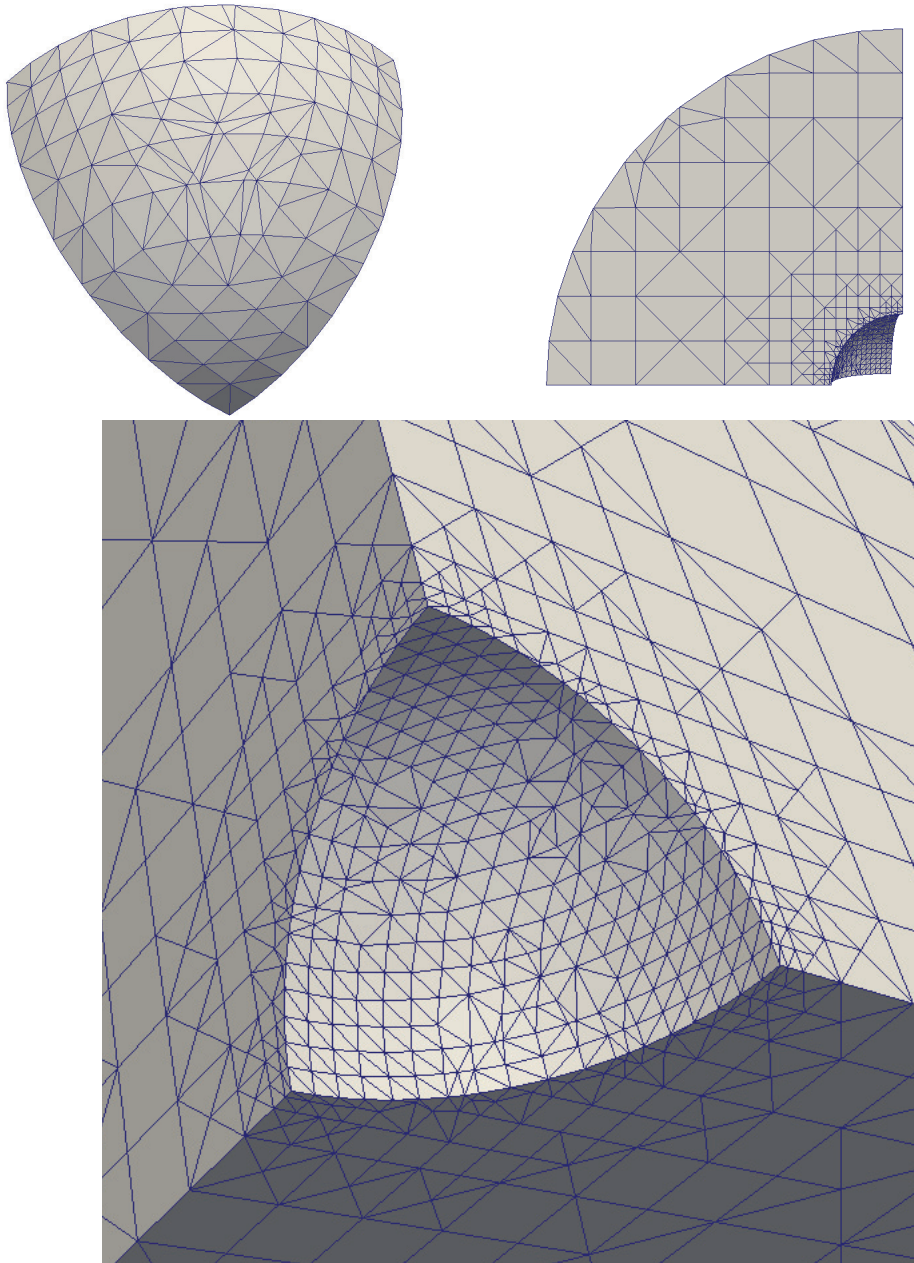


**Figure 6.5:** Mesh of the hollow sphere with 1716 DOFs



## 6.5. NUMERICAL EXAMPLES

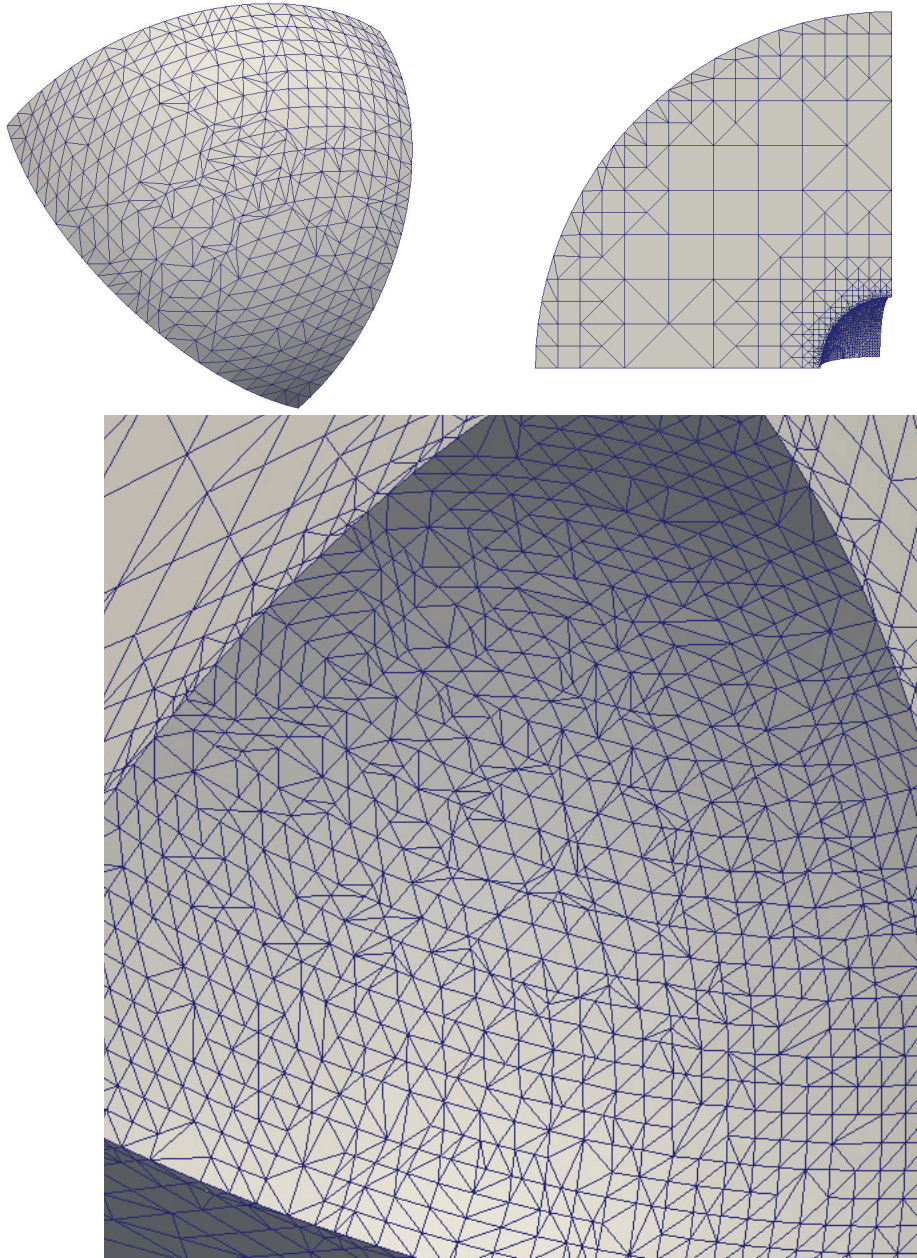
---



**Figure 6.6:** Mesh of the hollow sphere with 3896 DOFs

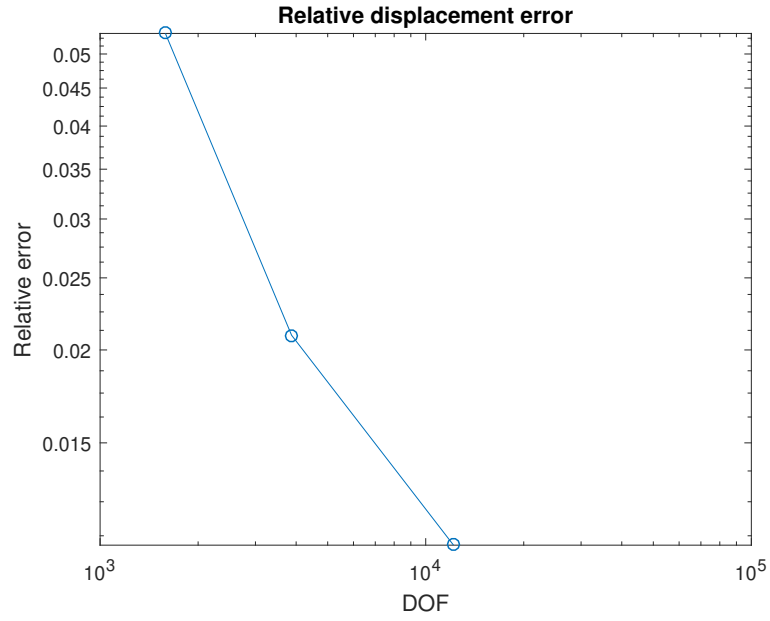
## 6.5. NUMERICAL EXAMPLES

---

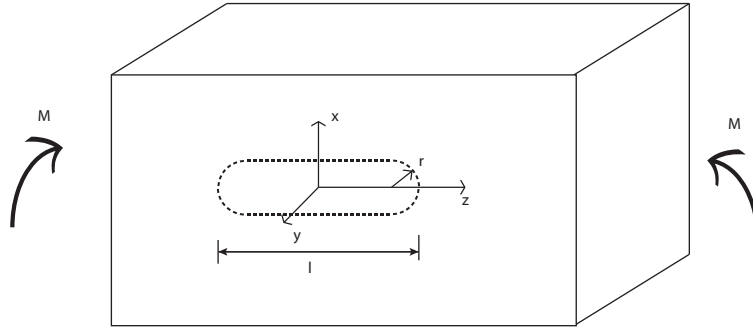


**Figure 6.7:** Mesh of the hollow sphere with 12078 DOFs

## 6.5. NUMERICAL EXAMPLES



**Figure 6.8:** Convergence of displacement error



**Figure 6.9:** Problem layout

analytical solution. All stress component except  $\sigma_z$  is zero.

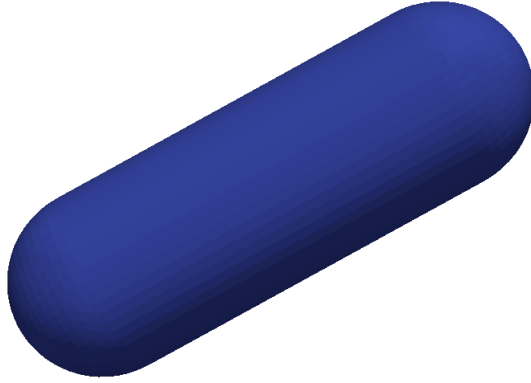
$$u_x = -\frac{1}{2R} [z^2 + \nu (x^2 - y^2)] \quad (6.47a)$$

$$u_y = -\frac{\nu xy}{R} \quad (6.47b)$$

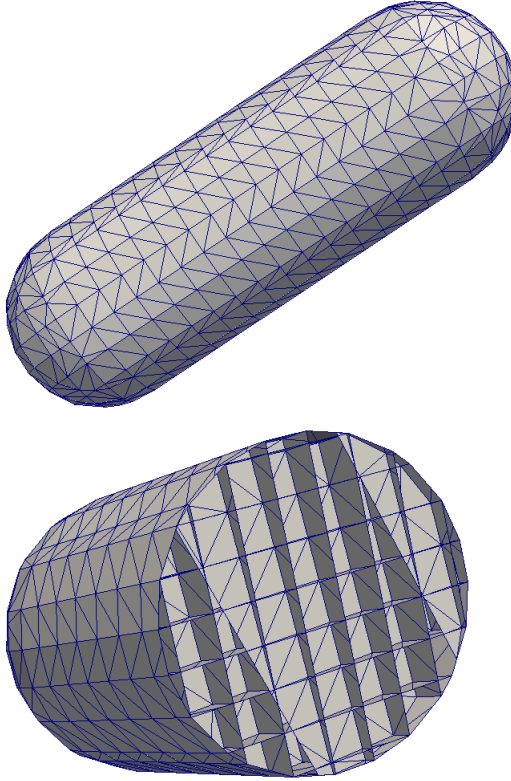
$$u_z = \frac{xz}{R} \quad (6.47c)$$

## 6.5. NUMERICAL EXAMPLES

---



**Figure 6.10:** Geometry of the capsule



**Figure 6.11:** Mesh of the Capsule

$$\sigma_x = \frac{Ex}{R} \tag{6.48}$$

## 6.5. NUMERICAL EXAMPLES

---

In the numerical calculation, The dimension of the outer cuboid will not affect the result because of the independence of the analytical solution (Eq.6.47c). 6-nodes triangular elements are used to achieve an exact solution. The geometric properties are:  $l = 100$  m and  $r = 17.5$  m The material properties are:  $\nu = 0.2$  and  $E = 30 \text{ N m}^{-2}$ . The error of the displacement is calculated as followed.

$$e_u = \frac{||u_{ex} - u||}{||u_{ex}||} \quad (6.49a)$$

$$e_s = \frac{||\sigma_{ex} - \sigma||}{||\sigma_{ex}||} \quad (6.49b)$$

The error of the displacement norm is  $1.7563 \times 10^{-14}$  and the error of the stress norm is  $1.3184 \times 10^{-9}$ .

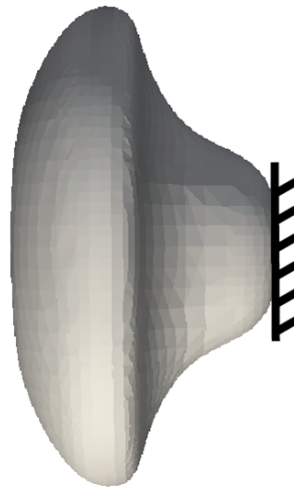
### 6.5.3 Wall lamp

In this example, a wall lamp illustrated in Fig. 6.12 with gravity only is considered. The material properties are: Youngs modulus  $E = 20 \text{ N m}^{-2}$ , Poissons ratio  $\nu = 0.3$  and density  $\rho = 2 \text{ kg m}^{-3}$ .

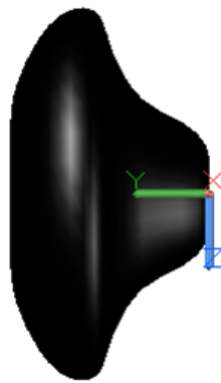
The geometric model is extracted from the AutoCAD as illustrated in Fig. 6.13. The background mesh before cutting is shown in Fig. 6.14. The generated mesh are plotted in Fig. 6.15. The displacement is plotted as the deformed shape in Fig. 6.16.

## 6.5. NUMERICAL EXAMPLES

---



**Figure 6.12:** Boundary condition of the wall lamp



**Figure 6.13:** CAD drawing of the wall lamp

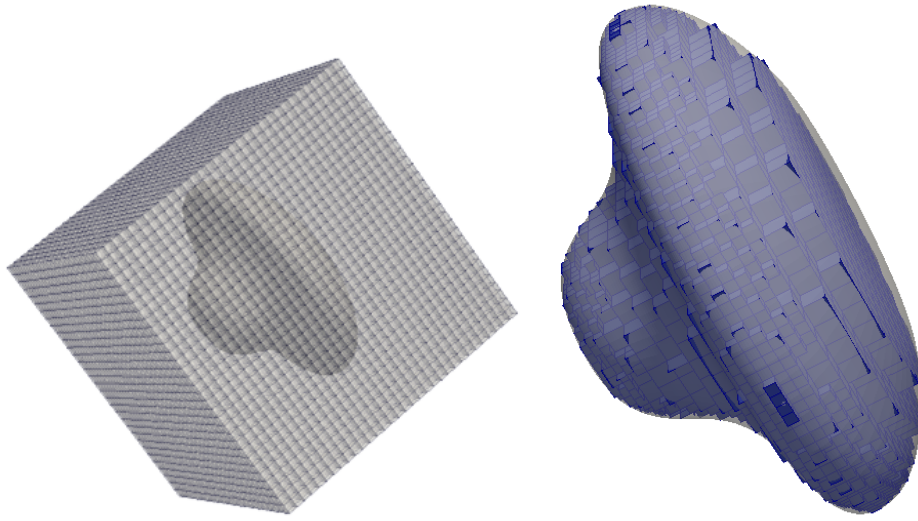
### 6.5.4 Meshing examples

In this section, some other mesh examples with irregular geometric boundaries are considered. Fig. 6.18 shows the mesh generated for a spinner with a CAD input illustrated in Fig. 6.17. Fig. 6.20 shows the mesh generated for an Egypt Sphinx with a CAD input plotted in Fig. 6.19



## 6.6. CONCLUSIONS

---



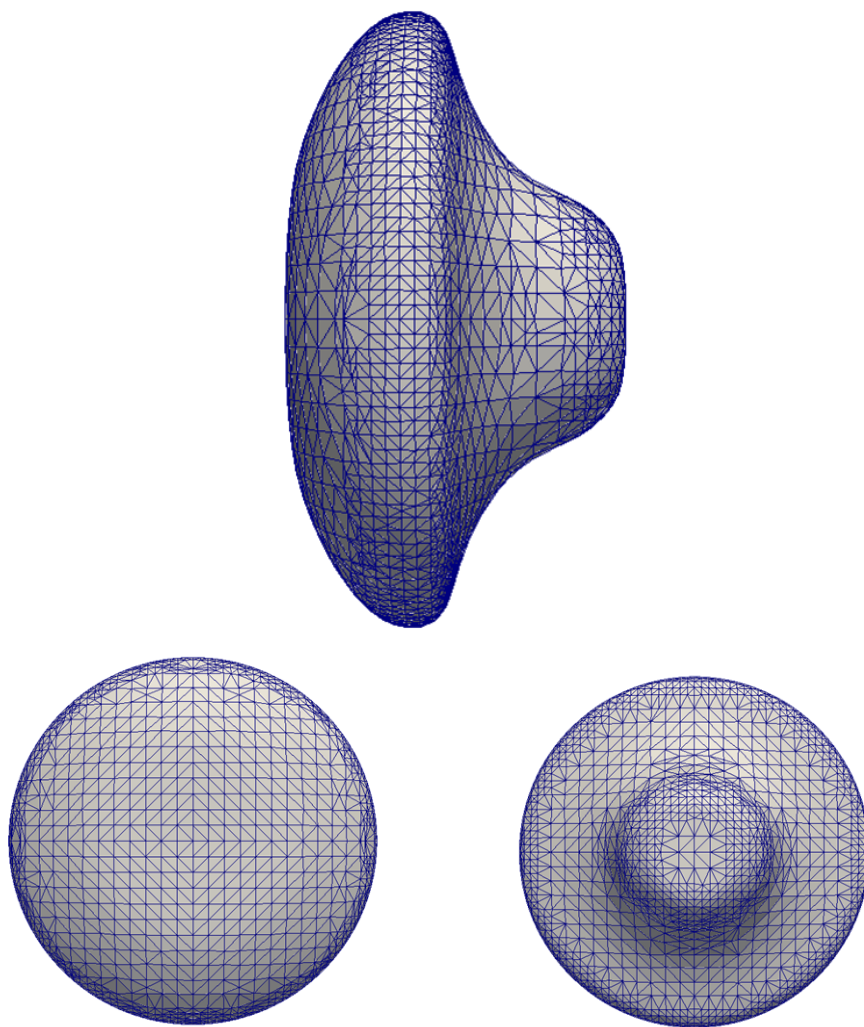
**Figure 6.14:** Background mesh of the wall lamp; left: background mesh; right: background mesh around the boundary

## 6.6 Conclusions

In this chapter, a projection is conducted after the octree mesh is generated from the STL file. Geometric exact is retained by projecting intersections on the triangular surfaces back to original NURBS surfaces or by replacing the intersection calculation with finding that between the edge of the element and the NURBS surface directly. Both of the projecting and the intersection calculating are accelerated by utilizing the strong convex hull property of the NURBS. The proposed method is able to generate the mesh from an arbitrary geometric input and shows higher accuracy compared to conventional method.

## 6.6. CONCLUSIONS

---

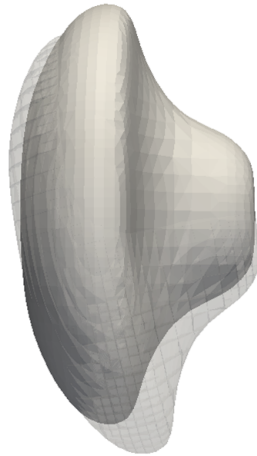


**Figure 6.15:** Mesh of the wall lamp

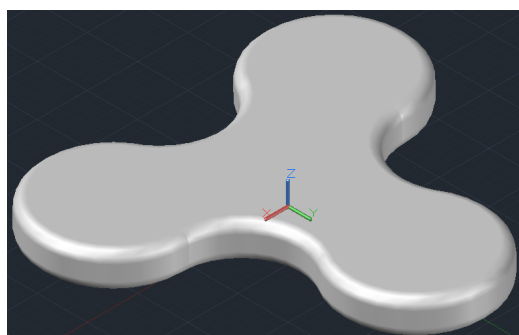


## 6.6. CONCLUSIONS

---



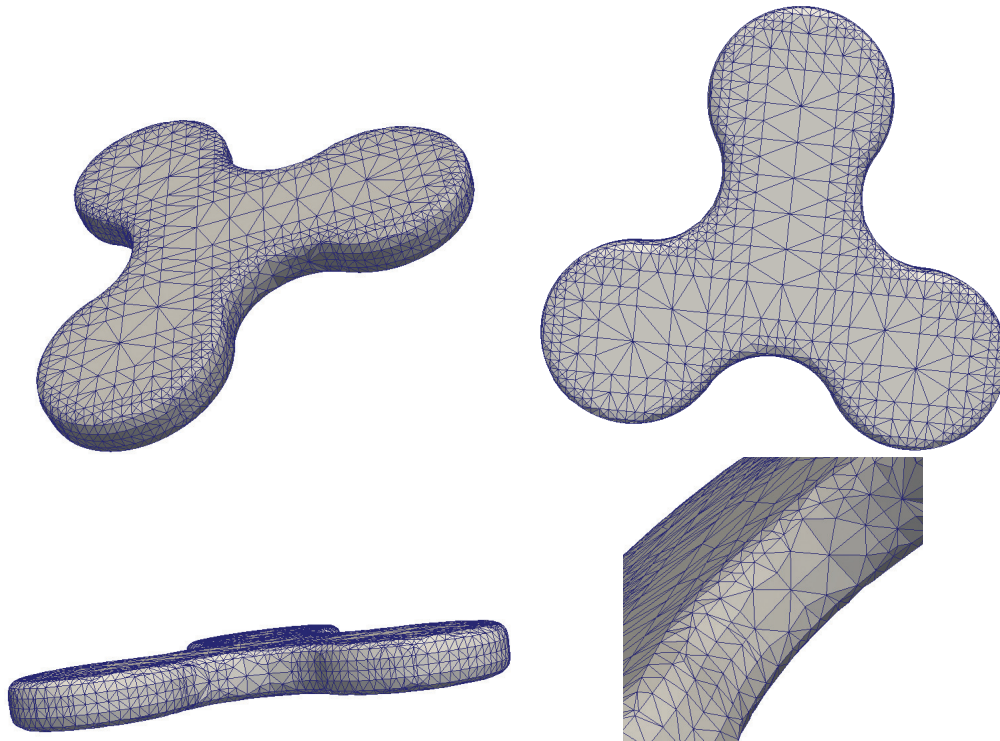
**Figure 6.16:** Deformed shape of the wall lamp



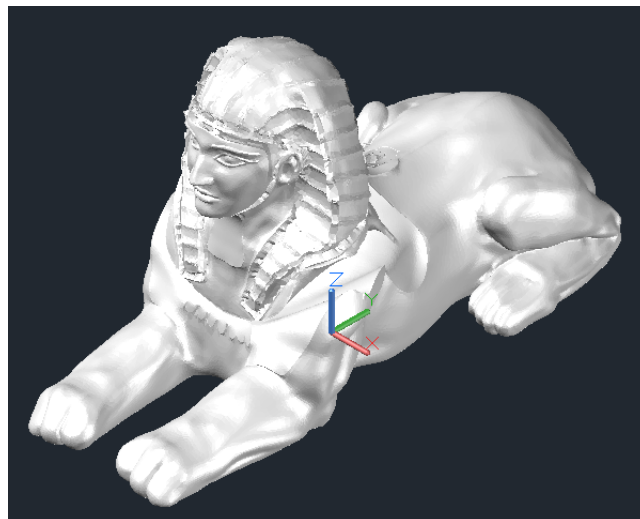
**Figure 6.17:** CAD design for the spinner

## 6.6. CONCLUSIONS

---



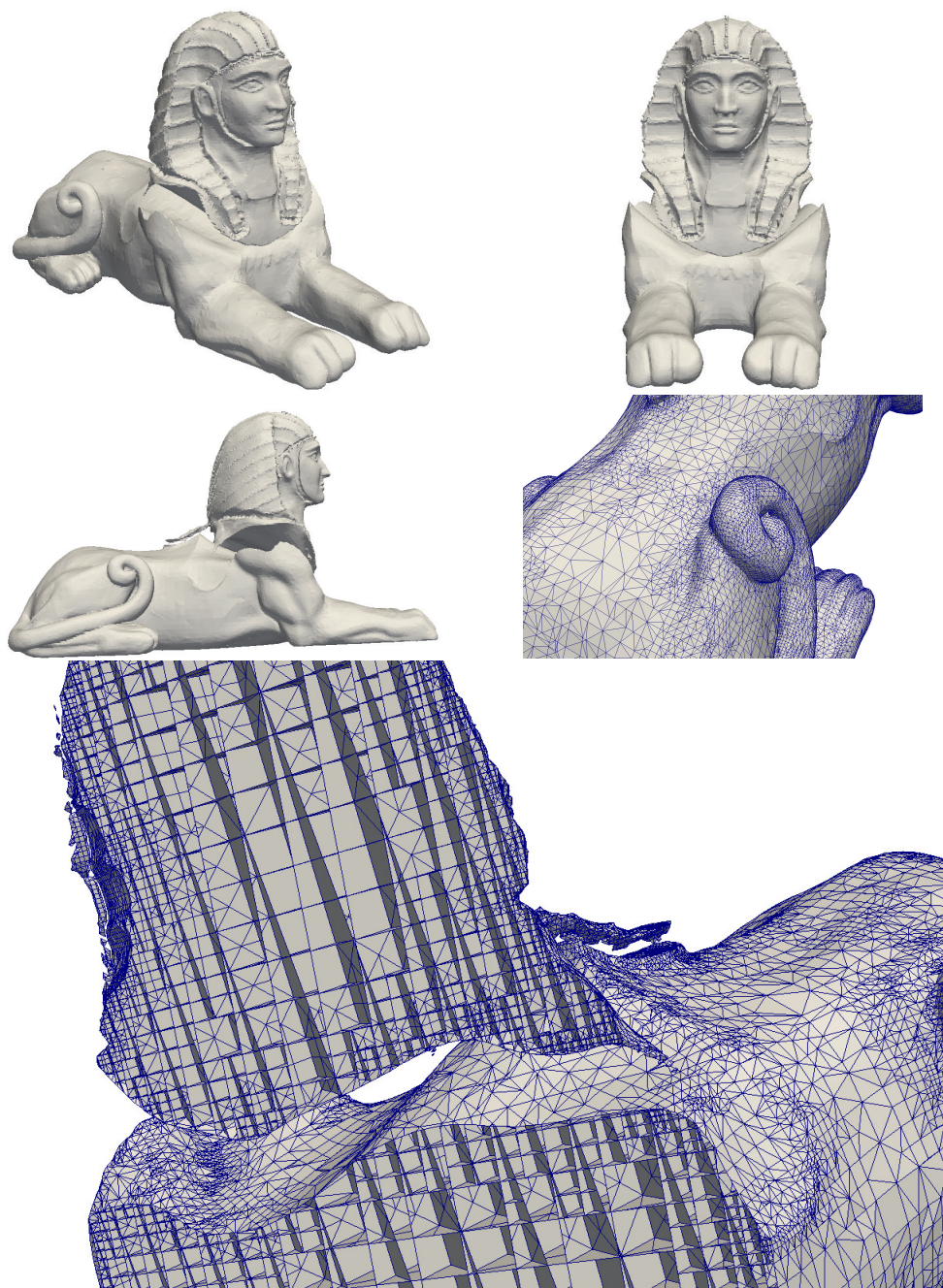
**Figure 6.18:** Mesh for the spinner



**Figure 6.19:** CAD design for the Egypt Sphinx

## 6.6. CONCLUSIONS

---



**Figure 6.20:** Mesh for the Egypt Sphinx

# **Chapter 7**

## **Conclusions and recommendation**

### **7.1 Summary**

In this thesis, a systematic numerical method where all procedures are conducted without human involvement for an arbitrary geometric input in both 2D and 3D situations has been developed based on the SBFEM. The SBFEM is a semi-analytical method which combines the main advantages of the finite element method and the boundary element method but also has unique features of its own. In contrast to the FEM, only the boundary is discretized using the conventional FEM interpolating function which leads to a decline in the number of unknowns. It also allows solving the problem involving bimaterial interfaces and crack faces without the discretization of them. Compared to the BEM, the fundamental solution

## 7.1. SUMMARY

---

is no longer required. The infinite boundary can be achieved naturally as the radiation condition at infinity is satisfied in the SBFEM . As only the boundary information is required in SBFEM and hence provides a seamless integration with the CAD modeling compared to conventional Isogeometric Method. The discretization of the boundary can be based on the standard finite element interpolation functions for computational efficiency or NURBS basis function for exact geometry.

The preprocessing of the proposed method is conducted automatically without losing exact geometric representation. In 2D problems, the geometric information from the IGES file exported from CAD design is extracted and a quad-tree based mesh will be generated automatically. High mesh quality and reduced computational cost in the calculation of the element stiffness matrix can be expected. For 3D cases, both of the STL and the IGES files are exported from the CAD design and an octree based mesh will be generated based on STL file. Since the STL file represents the triangulation of the geometric surfaces, intersections calculated during cutting stage are supposed to be located on the triangular plane  $s$  instead of the original surface  $s$  . As a consequence, a point projection algorithm is adopted to move the points back to the NURBS surfaces after the mesh has been generated. Then, an arbitrary geometric shape in 2D and 3D can be meshed automatically with high quality mesh and solved by the SBFEM. The geometric exact can also be achieved by replacing the distance calculation with finding that between the edge of the element and

## 7.1. SUMMARY

---

the NURBS surface directly.

After the solution is determined from the SBFEM, a mesh refinement may be necessary to improve the accuracy and check the convergence. In order to find out the scaled boundary finite element which improves the accuracy significantly after refinement, an adaptive mesh refinement algorithm is required. Expressions related to the eigenvalues of the SBFEM formulation are adopted as the physical error indicator to prevent extra work such as stress recovery. Some key geometric error estimator including area of the subdomain, minimal angle and so on are also included. The decision based on numerous error estimators are conducted by the help of a machine learning algorithm. The proposed adaptive mesh refinement exhibits higher rate of convergence compared to an uniform mesh refinement using SBFEM.

The thesis commences with an introduction to the research topic in Chapter 1. A background and motivation for the research are presented followed by the objective and outline of the research. In Chapter 2, the linear theory of the Isogeometric analysis, together with a brief introduction on the NURBS and its mathematical backgrounds, potentials and limitations are summarized. A brief introduction on the IGES file is also presented. Due to the dimensional mismatch between the FEM and the CAD, the SBFEM is the technique which can provide a seamless integration with the CAD modeling. The concept of adaptive mesh refinement and its lim-

## 7.1. SUMMARY

---

itation is presented as well. The MLP then is introduced to overcome this limitation by the adoption of multiple error indicators. Finally, algorithms used to generate octree mesh in 3D from STL file is presented.

In Chapter 3, the NURBS basis functions are employed to approximate the unknown fields in the circumferential direction within the framework of the SBFEM. The accuracy, effectiveness and the convergence properties of the proposed method are demonstrated with benchmark problems in linear elasticity and linear elastic fracture mechanics. From the numerical studies, it can be observed that the NURBS basis functions yield superior accuracy when compared to Lagrange basis functions of the same order. The proposed method overcomes the disadvantages of both the isogeometric finite element analysis and the isogeometric boundary element method. When applied to problems with singularities, the proposed method does not require additional functions to span the solution space. Moreover, the proposed framework does not require internal discretization to study the dynamic response at high frequencies. However, for complicated geometries, to meet the star convexity, subdivision into smaller sub-domains is required.

In Chapter 4, the IGES file is employed directly from the CAD output to represent the geometry during the preprocessing. The proposed methods provides a systematic and automatic mesh generation algorithm where high quality mesh is produced efficiently. The CAD design file

## 7.1. SUMMARY

---

can be used directly and the exact geometry can also be retained, which largely reduce the human efforts involved in numerical analysis. Moreover, it helps to reduce the analytical error as the difference in the geometric representation is minimized. Computational efficiency is improved via utilizing the pattern of the quadtree to prevent repeated calculation. Points projection algorithm is accelerated using the strong convex hull property of the NURBS curve. The accuracy, effectiveness and the convergence properties of the proposed method are demonstrated with benchmark problems in linear elasticity mechanics. From the numerical studies, it can be observed that the quadtree mesh yield better accuracy when compared to uniform mesh with same degree of freedoms.

Chapter 5 adopts a machine learning algorithm to develop an extensible and flexible error indicator. Any other error estimators can be added to the existing framework and their effects can be detected based on the performance indicators in machine learning. A MLP trained error estimator that concludes expressions related to the eigenvalues of the SBFEM formulation and some key geometric properties of the scaled boundary finite element gives a higher convergence rate compared to the uniform refinement. In order to improve the learning effectiveness of the MLP, regularization methods including bagging and dropout are utilized. Due to the lack of the eigenvalue error indicator in the first order triangular element, method that eliminates these situation is developed. A matrix representation of the NURBS curves is presented to achieve a higher efficiency and



## 7.2. FUTURE WORK

---

stability in calculating the intersections between edge of the element and the NURBS curve.

In Chapter 6, a projection is conducted after the mesh is generated from STL file using octree or calculating the intersection between the edge of the element and the original NURBS surface to retain the exact geometry. In order to utilize the strong hull property of the NURBS surface, a quick hull algorithm in 3D is introduced. As a consequence, the computational efficiency of the points projection and the intersection calculation are improved significantly. The proposed method is able to generate the mesh from an arbitrary geometric input and shows higher accuracy compared to conventional method.

## 7.2 Future work

As discussed in Chapter 3, there is no exact numerical integration for the NURBS basis function at the moment due to its rational property. Furthermore, the computational cost of calculating a value of NURBS basis function can be 10 times higher than that of the traditional Legendre polynomials. In chapters after Chapter 3, the conventional shape functions are adopted instead of the NURBS basis functions to avoid too many expensive computational operations when the number of the elements grow drastically. Future work can be performed on deriving an exact and

## 7.2. FUTURE WORK

---

efficient numerical integration quadrature to prevent extra error. The algorithm that evaluates the NURBS basis function can also be optimized so that the whole programme can be much more efficient as it is involved in almost all NURBS operations.

As discussed in Chapter 4, one of the most outstanding advantages of using machine learning algorithm to train the model responsible for making decision whether an element needs to be refined lies in the allowance of unlimited number of error indicators. The effectiveness of the deep neural network is closely related to the quantity of the training data. In Chapter 4, less than a thousand of the training data is used, which limits the depth of the neural network and the number of the error indicators. Future work can concentrate on building a platform so the cloud computing can be utilized and the service can be available to the whole community. By having a platform, number of the training data can be solved naturally as any request of calculation contributes to one extra training data. A large training set then motivates the scholars adding any error indicator.

# Bibliography

*Initial Graphics Exchange Specifications (IGES)*, pages 125–130. Springer US, Boston, MA, 2007. ISBN 978-0-387-23324-6.

Aigner, M; Heinrich, C; Juttler, B; Pilgerstorfer, E; Simeon, B, and Vuong, A. Swept volume parameterization for isogeometric analysis, 2009.

Ainsworth, Mark and Oden, J Tinsley. A unified approach to a posteriori error estimation using element residual methods. *Numerische Mathematik*, 65(1):23–50, dec 1993. ISSN 0945-3245.

Anderson, Kenneth R. A reevaluation of an efficient algorithm for determining the convex hull of a finite planar set. *Information Processing Letters*, 7(1):53–55, 1978. ISSN 0020-0190.

Andrew, A M. Another efficient algorithm for convex hulls in two dimensions. *Information Processing Letters*, 9(5):216–219, 1979. ISSN 0020-0190.

Areias, P; Rabczuk, T, and Dias-da Costa, D. Element-wise fracture algorithm based on rotation of edges. *Engineering Fracture Mechanics*, 110: 113–137, 2013. ISSN 0013-7944.

## BIBLIOGRAPHY

---

- Attene, Marco; Campen, Marcel, and Kobbelt, Leif. Polygon Mesh Repairing: An Application Perspective. *ACM Comput. Surv.*, 45(2):15:1—15:33, mar 2013. ISSN 0360-0300.
- Augarde, C E and Deeks, A J. The use of Timoshenko's exact solution for a cantilever beam in adaptive analysis. *Finite Elements in Analysis and Design*, 44(9-10):595–601, 2008.
- Auricchio, F; Calabrò, F; Hughes, T J R; Reali, A, and Sangalli, G. A simple algorithm for obtaining nearly optimal quadrature rules for NURBS-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 249252(0):15–27, 2012.
- Ba, Thang Luu; Bus, Laurent, and Mourrain, Bernard. Curve / Surface Intersection Problem by means of Matrix Representations. pages 71–78, 2009.
- Babuška, I and Rheinboldt, W C. A-posteriori error estimates for the finite element method. *International Journal for Numerical Methods in Engineering*, 12(10):1597–1615.
- Babuška, I; Banerjee, U, and Osborn, J E. Survey of meshless and generalized finite element methods: A unified approach. *Acta Numerica*, 12: 1–125, 2003.
- Banks-Sills, L; Hershkowitz, I, and Wawrzynek, P A. Methods for calculating stress intensity factors in anisotropic materials: Part I -  $z = 0$  is a symmetric plane. *Engineering Fracture Mechanics*, 72(15):2328–2358, 2005.

## BIBLIOGRAPHY

---

- Barber, C Bradford; Dobkin, David P; Dobkin, David P, and Huhdanpaa, Hannu. The Quickhull Algorithm for Convex Hulls. *ACM Trans. Math. Softw.*, 22(4):469–483, dec 1996. ISSN 0098-3500.
- Bauman, Paul T; Oden, J Tinsley, and Prudhomme, Serge. Adaptive multi-scale modeling of polymeric materials with Arlequin coupling and Goals algorithms. *Computer Methods in Applied Mechanics and Engineering*, 198(5):799–818, 2009. ISSN 0045-7825.
- Bazilevs, Y; Calo, V M; Zhang, Y, and Hughes, T J R. Isogeometric fluid-structure interaction analysis with applications to arterial blood flow. *Computer Methods in Applied Mechanics and Engineering*, 38:310–322, 2006a.
- Bazilevs, Y; de Veiga, L; Cottrell, J A; Hughes, T J R, and Sangalli, G. Isogeometric analysis: approximation, stability and error estimates for h-refined meshes. *Mathematical Models and Methods in Applied Sciences*, 16:1031–1090, 2006b.
- Bazilevs, Y; Hsu, M.-C., and Scott, M A. Isogeometric fluidstructure interaction analysis with emphasis on non-matching discretizations, and with application to wind turbines. *Computer Methods in Applied Mechanics and Engineering*, 249-252:28–41, 2012. ISSN 0045-7825.
- Béchet, E; Cuilliere, J.-C., and Trochu, F. Generation of a finite element MESH from stereolithography (STL) files. *Computer-Aided Design*, 34 (1):1–17, 2002. ISSN 0010-4485.

## BIBLIOGRAPHY

---

- Benson, D J; Bazilevs, Y; de Luycker, E; Hsu, M.-C.; Scott, M; Hughes, T J R, and Belytschko, T. A generalized finite element formulation for arbitrary basis functions: From isogeometric analysis to XFEM. *International Journal for Numerical Methods in Engineering*, 83(6):765–785, 2010.
- Berkeley, U C. Introduction to the finite element method, 2010.
- Bern, Marshall; Eppstein, David, and Gilbert, John. Provably good mesh generation. *Journal of Computer and System Sciences*, 48(3):384–409, 1994. ISSN 0022-0000.
- Bernardi, Christine; Maday, Yvon, and Rapetti, Francesca. Basics and some applications of the mortar element method. *GAMM-Mitteilungen*, 28(2):97–123.
- Bird, G E; Trevelyan, J, and Augarde, C E. A coupled BEM/scaled boundary FEM formulation for accurate computations in linear elastic fracture mechanics. *Engineering Analysis with Boundary Elements*, 34(6):599–610, 2010. ISSN 0955-7997.
- Bishop, J E. A displacementbased finite element formulation for general polyhedra using harmonic shape functions. *International Journal for Numerical Methods in Engineering*, 97(1):1–31, 2014.
- Blacker, Ted D and Meyers, Ray J. Seams and wedges in plastering: A 3-D hexahedral mesh generation algorithm. *Engineering with Computers*, 9(2):83–93, jun 1993. ISSN 1435-5663.

## BIBLIOGRAPHY

---

- Boor De, C. On calculating with B-spline. *Journal of Approximation Theory*, 6:50–62, 1972.
- Bordas, Stéphane; Nguyen, Phu Vinh; Dunant, Cyrille; Guidoum, Amor, and Nguyen-Dang, Hung. An extended finite element library. *International Journal for Numerical Methods in Engineering*, 71(6):703–732.
- Boroomand, B and Zienkiewicz, O C. Recovery procedures in error estimation and adaptivity. Part II: Adaptivity in nonlinear problems of elastoplasticity behaviour. *Computer Methods in Applied Mechanics and Engineering*, 176(1):127–146, 1999. ISSN 0045-7825.
- Boser, Bernhard; Guyon, Isabelle, and N. Vapnik, Vladimir. A Training Algorithm for Optimal Margin Classifier. *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory*, 5, 1996.
- Breiman, Leo. Bagging Predictors. *Machine Learning*, 24(2):123–140, aug 1996. ISSN 1573-0565.
- Buffa, A; de Falco, C, and Sangalli, G. IsoGeometric Analysis: Stable elements for the 2D Stokes equation. *International Journal for Numerical Methods in Fluids*, 65(11-12):1407–1422, 2011.
- Busé, Laurent and Luu Ba, Thang. Matrix-based implicit representations of rational algebraic curves and applications. *Computer Aided Geometric Design*, 27(9):681–699, 2010. ISSN 01678396.
- Cai, M; Shi, Y, and Liu, J. Deep maxout neural networks for speech recog-

## BIBLIOGRAPHY

---

- dition. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 291–296, dec 2013.
- Chan, T M. Optimal output-sensitive convex hull algorithms in two and three dimensions. *Discrete & Computational Geometry*, 16(4):361–368, apr 1996. ISSN 1432-0444.
- Chang, Yin-Wen; Hsieh, Cho-Jui; Chang, Kai-Wei; Ringgaard, Michael, and Lin, Chih-Jen. Training and Testing Low-degree Polynomial Data Mappings via Linear SVM. *J. Mach. Learn. Res.*, 11:1471–1490, aug 2010. ISSN 1532-4435.
- Chau-Dinh, Thanh; Zi, Goangseup; Lee, Phill-Seung; Rabczuk, Timon, and Song, Jeong-Hoon. Phantom-node method for shell models with arbitrary cracks. *Computers & Structures*, 92-93:242–256, 2012. ISSN 0045-7949.
- Chessa, J and Belytschko, T. An Extended Finite Element Method for Two-Phase Fluids. *Journal of Applied Mechanics*, 70(1):10–17, jan 2003. ISSN 0021-8936.
- Chidgzey, S R; Trevelyan, J, and Deeks, A J. Coupling of the boundary element method and the scaled boundary finite element method for computations in fracture mechanics. *Computers & Structures*, 86(11): 1198–1203, 2008. ISSN 0045-7949.
- Chin, Francis Y L and Wang, Cao An. Optimal Algorithms for the Intersection and the Minimum Distance Problems Between Planar Polygons. *IEEE Transactions on Computers*, C-32:1203–1207, 1983.



## BIBLIOGRAPHY

---

- Choi, B K; Yoo, W S, and Lee, C S. Matrix representation for NURBS curves and surfaces. *CAD*, 22(4):235–240, 1990.
- Clough, R W. The finite element method after twenty-five years: A personal view. *Computers & Structures*, 12:361–370, 1980.
- Cohen, E; Martin, T; Kirby, R M; Lyche, T, and Riesenfeld, R F. Analysis-aware modeling: understanding quality considerations in modeling for isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 199(5-8):334–356, 2010.
- Cormen, Thomas H; Leiserson, Charles E; Rivest, Ronald L, and Stein, Clifford. *Introduction to Algorithms, Third Edition*. The MIT Press, 3rd edition, 2009. ISBN 0262033844, 9780262033848.
- Cortes, Corinna and Vapnik, Vladimir. Support-vector networks. *Machine Learning*, 20(3):273–297, sep 1995. ISSN 1573-0565.
- Cottrell, J A; Reali, A; Bazilevs, Y, and Hughes, T J R. Isogeometric analysis of structural vibrations. *Computer Methods in Applied Mechanics and Engineering*, 195(41-43):5257–5296, 2006.
- Cottrell, J A; Hughes, T J R, and Reali, A. Studies of refinement and continuity in isogeometric structural analysis. *Computer Methods in Applied Mechanics and Engineering*, 196(41-44):4160–4183, 2007.
- Cottrell, J A; Hughes, T J R, and Bazilevs, Y. *Isogeometric Analysis Toward Intergration of CAD and FEA*. WILEY, 1st editio edition, 2009.

## BIBLIOGRAPHY

---

- Craig, Alan. Accuracy estimates and adaptive refinements in finite element computations. *International Journal for Numerical Methods in Engineering*, 24(6):1228.
- De Luycker, Emmanuel; Benson, David; Belytschko, T; Bazilevs, Yuri, and Hsu, Ming-Chen. X-FEM in isogeometric analysis for linear fracture mechanics. *International Journal for Numerical Methods in Engineering*, 87:541–565, 2011.
- Deeks, A J and Chidgze, S R. Determination of coefficients of crack tip asymptotic fields using the scaled boundary finite element method. *Engineering Fracture Mechanics*, 72:2019–2036, 2005.
- Deeks, A J and Wolf, J P. A virtual work derivation of the scaled boundary finite-element method for elastostatics. *Computational Mechanics*, 28 (6):489–504, 2002a.
- Deeks, A. J. and Wolf, J. P. A virtual work derivation of the scaled boundary finite-element method for elastostatics. *Computational Mechanics*, 28 (6):489–504, 2002b. ISSN 01787675.
- Deeks, Andrew J. Prescribed side-face displacements in the scaled boundary finite-element method. *Computers & Structures*, 82(15):1153–1165, 2004. ISSN 0045-7949.
- Deeks, Andrew J and Wolf, John P. An h-hierarchical adaptive procedure for the scaled boundary finite-element method. *International Journal for Numerical Methods in Engineering*, 54(4):585–605.

## BIBLIOGRAPHY

---

- Deeks, Andrew J and Wolf, John P. Stress recovery and error estimation for the scaled boundary finite-element method. *International Journal for Numerical Methods in Engineering*, 54(4):557–583, 2002c. ISSN 1097-0207.
- Duval, M.; Lozinski, A.; Passieux, J. C., and Salaün, M. Residual error based adaptive mesh refinement with the non-intrusive patch algorithm. *Computer Methods in Applied Mechanics and Engineering*, 329:118–143, 2018. ISSN 00457825.
- Ebeida, Mohamed S; Davis, Roger L, and Freund, Roland W. A new fast hybrid adaptive grid generation technique for arbitrary two-dimensional domains. *International Journal for Numerical Methods in Engineering*, 84(3):305–329, 2010. ISSN 1097-0207.
- Edelsbrunner, H. Computing the Extreme Distances Between Two Convex Polygons. *J. Algorithms*, 6(2):213–224, jun 1985. ISSN 0196-6774.
- Felippa, C A. Variational methods for the solution of problems of equilibrium and vibrations. *International Journal for Numerical Methods in Engineering*, 37:2159–2187, 1994.
- Fisher, R A. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, 1936.
- Frey, Pascal Jean and George, Paul-Louis. *Mesh Generation: Application to Finite Elements*. ISTE, 2007. ISBN 1903398002.
- Fries, Thomas-Peter. A corrected XFEM approximation without problems

## BIBLIOGRAPHY

---

- in blending elements. *International Journal for Numerical Methods in Engineering*, 75(5):503–532.
- George, P L; Borouchaki, H, and Saltel, E. Ultimate' robustness in meshing an arbitrary polyhedron. *International Journal for Numerical Methods in Engineering*, 58(7):1061–1089.
- Ghasemi, Hamid; Brighenti, Roberto; Zhuang, Xiaoying; Muthu, Jacob, and Rabczuk, Timon. Optimization of fiber distribution in fiber reinforced composite by using NURBS functions. *Computational Materials Science*, 83:463–473, 2014. ISSN 0927-0256.
- Gill, Simranpreet Singh and Kaplas, Munish. Comparative Study of 3D Printing Technologies for Rapid Casting of Aluminium Alloy. *Materials and Manufacturing Processes*, 24(12):1405–1411, 2009.
- Goodfellow, Ian; Bengio, Yoshua, and Courville, Aaron. *Deep Learning*. MIT Press, 2016.
- Goodfellow, Ian J; Warde-Farley, David; Mirza, Mehdi; Courville, Aaron, and Bengio, Yoshua. Maxout Networks. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML'13, pages III–1319—III–1327. JMLR.org, 2013.
- Grabowski, H and Li, X. Coefficient formula and matrix of nonuniform B-spline functions. *CAD*, 24(12):637–642, 1992.
- Gravenkamp, Hauke; Bause, Fabian, and Song, Chongmin. On the computation of dispersion curves for axisymmetric elastic waveguides using

## BIBLIOGRAPHY

---

- the Scaled Boundary Finite Element Method. *Computers & Structures*, 131:46–55, 2014. ISSN 0045-7949.
- Gravouil, A; Moës, N, and Belytschko, T. Non-planar 3D crack growth by the extended finite element and level setsPart II: Level set update. *International Journal for Numerical Methods in Engineering*, 53(11):2569–2586, 2002.
- Guiggiani, Massimo. Error indicators for adaptive mesh refinement in the boundary element method. A new approach. *International Journal for Numerical Methods in Engineering*, 29:1247–1269, 1990.
- Gupta, A. K. A finite element for transition from a fine to a coarse grid. *International Journal for Numerical Methods in Engineering*, 12(1):35–45, 1978.
- He, Kaiming; Zhang, Xiangyu; Ren, Shaoqing, and Sun, Jian. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, pages 1026–1034, Washington, DC, USA, 2015. IEEE Computer Society. ISBN 978-1-4673-8391-2.
- He, Yiqian; Yang, Haitian, and Deeks, Andrew J. An Element-free Galerkin (EFG) scaled boundary method. *Finite Elements in Analysis and Design*, 62:28–36, 2012. ISSN 0168-874X.
- He, Yiqian; Yang, Haitian, and Deeks, Andrew J. Use of Fourier shape

## BIBLIOGRAPHY

---

- functions in the scaled boundary method. *Engineering Analysis with Boundary Elements*, 41:152–159, 2014. ISSN 0955-7997.
- Hinton, Geoffrey E; Srivastava, Nitish; Krizhevsky, Alex; Sutskever, Ilya, and Salakhutdinov, Ruslan. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0, 2012.
- Ho, Tin Kam. Random Decision Forests. In *Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 1) - Volume 1*, ICDAR '95, pages 278—, Washington, DC, USA, 1995. IEEE Computer Society. ISBN 0-8186-7128-9.
- Hosseini, Saman; Remmers, Joris J C; Verhoosel, Clemens V, and de Borst, René. An isogeometric continuum shell element for non-linear analysis. *Computer Methods in Applied Mechanics and Engineering*, 271:1–22, 2014. ISSN 0045-7825.
- Hughes, T J R; Cottrell, J A, and Bazilevs, Y. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194(39-41): 4135–4195, 2004.
- Hughes, T J R; Cottrell, J A, and Bazilevs, Y. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry, and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194:4135–4195, 2005.
- Hughes, T J R; Reali, A, and Sangalli, G. Efficient quadrature for NURBS-

## BIBLIOGRAPHY

---

- based Isogeometric Analysis. *Computer Methods in Applied Mechanics and Engineering*, 199:301–313, 2010.
- Hughes, Thomas J. R. The Finite Element Method: Linear Static and Dynamic Finite Element Analysis: Thomas J. R. Hughes. *Computer-Aided Civil and Infrastructure Engineering*, 4(3):245–246.
- IGES, . *Initial graphics exchange specification, version 3.0*. National Bureau of Standards,, Gaithersburg, MD, USA, 1986.
- Jarrett, Kevin; Kavukcuoglu, Koray; Ranzato, Marc’Aurelio, and Lecun, Yann. What is the Best Multi-Stage Architecture for Object Recognition?, 2009.
- Jia, Yue; Zhang, Yongjie; Xu, Gang; Zhuang, Xiaoying, and Rabczuk, Timon. Reproducing kernel triangular B-spline-based FEM for solving PDEs. *Computer Methods in Applied Mechanics and Engineering*, 267: 342–358, 2013. ISSN 0045-7825.
- Jia, Yue; Anitescu, Cosmin; Ghorashi, Seyed Shahram, and Rabczuk, Timon. Extended isogeometric analysis for material interface problems. *IMA Journal of Applied Mathematics*, 80(3):608–633, 2015.
- Joulaian, Meysam and Düster, Alexander. Local enrichment of the finite cell method for problems with material interfaces. *Computational Mechanics*, 52(4):741–762, oct 2013. ISSN 1432-0924.
- Kamiya, N and Kawaguchi, K. Error analysis and adaptive refinement of

## BIBLIOGRAPHY

---

- boundary elements in elastic problem. *Advances in Engineering Software*, 15(3):223–230, 1992. ISSN 0965-9978.
- Kim, J and Duarte, C A. A new generalized finite element method for two-scale simulations of propagating cohesive fractures in 3-D. *International Journal for Numerical Methods in Engineering*, 104(13):1139–1172.
- Kirkpatrick, D and Seidel, R. The Ultimate Planar Convex Hull Algorithm? *SIAM Journal on Computing*, 15(1):287–299, 1986.
- Kita, E and Kamiya, N. Recent studies on adaptive boundary element methods. *Advances in Engineering Software*, 19(1):21–32, 1994. ISSN 0965-9978.
- Kita, E; Higuchi, K, and Kamiya, N. Application of r- and hr-adaptive BEM to two-dimensional elastic problem. *Engineering Analysis with Boundary Elements*, 24(4):317–324, 2000. ISSN 0955-7997.
- Koçyiit, Ö; Gültop, T, and Alyavuz, B. Numerical Solution of Seepage Problem Using Quad-Tree Based Triangular Finite Elements, 2009.
- Koo, Bonyong; Yoon, Minho, and Cho, Seonho. Isogeometric shape design sensitivity analysis using transformed basis functions for Kronecker delta property. *Computer Methods in Applied Mechanics and Engineering*, 253(0):505–516, 2013.
- Koren, Y; Bell, R, and Volinsky, C. Matrix Factorization Techniques for Recommender Systems. *Computer*, 42(8):30–37, aug 2009. ISSN 0018-9162.



## BIBLIOGRAPHY

---

- Laurent, Busé. Implicit matrix representations of rational Bézier curves and surfaces. *CAD Computer Aided Design*, 46(1):14–24, 2014. ISSN 00104485.
- Legrain, G; Allais, R, and Cartraud, P. On the use of the extended finite element method with quadtree/octree meshes. *International Journal for Numerical Methods in Engineering*, 86(6):717–743, 2011. ISSN 1097-0207.
- Li, Chao; Man, Hou; Song, Chongmin, and Gao, Wei. Fracture analysis of piezoelectric materials using the scaled boundary finite element method. *Engineering Fracture Mechanics*, 97:52–71, 2013. ISSN 0013-7944.
- Li, Kang and Qian, Xiaoping. Isogeometric analysis and shape optimization via boundary integral. *CAD*, 43(11):1427–1437, 2011.
- Lian, Haojie;Simpson, Robert;Bordas, Stéphane;. Sensitivity Analysis and Shape Optimisation through a T-spline Isogeometric Boundary Element Method. In *International Conference on Computational Mechanics*, 2013.
- Lin, Gao; Zhang, Yong; Hu, ZhiQiang, and Zhong, Hong. Scaled boundary isogeometric analysis for 2D elastostatics. *Science China Physics, Mechanics and Astronomy*, 57(2):286–300, feb 2014. ISSN 1869-1927.
- Lipton, S; Evans, J A; Bazilevs, Y; Elguedj, T, and Hughes, T J R. Robustness of isogeometric structural discretizations under severe mesh distortion. *Computer Methods in Applied Mechanics and Engineering*, 199(5):357–373, 2010. ISSN 0045-7825.

## BIBLIOGRAPHY

---

- Liu, G R. Mesh Free Methods Moving Beyond Finite Element Method. *Boca Raton*, 2003.
- Liu, Yan; Lo, S H; Guan, Zhen-Qun, and Zhang, Hong-Wu. Boundary recovery for 3D Delaunay triangulation. *Finite Elements in Analysis and Design*, 84:32–43, 2014. ISSN 0168-874X.
- Liu, Yan; Saputra, Albert A.; Wang, Junchao; Tin-Loi, Francis, and Song, Chongmin. Automatic polyhedral mesh generation and scaled boundary finite element analysis of STL models. *Computer Methods in Applied Mechanics and Engineering*, 313:106–132, 2017. ISSN 00457825.
- Löhner, Rainald and Parikh, Paresh. Generation of three-dimensional unstructured grids by the advancing-front method. *International Journal for Numerical Methods in Fluids*, 8(10):1135–1149.
- Ma, Ying Liang and Hewitt, W T. Point inversion and projection for NURBS curve and surface: Control polygon approach. *Computer Aided Geometric Design*, 20(2):79–99, 2003. ISSN 0167-8396.
- Maas, Andrew L; Hannun, Awni Y, and Ng, Andrew Y. Rectifier Nonlinearities Improve Neural Network Acoustic Models.
- Martin, T; Cohen, E, and Kirby, R M. Volumetric parameterization and trivariate B-spline fitting using harmonic functions. Computer Aided Geometric Design. *Computer Aided Geometric Design*, 26(6):648–664, 2009.

## BIBLIOGRAPHY

---

- Mayer, Ursula M; Popp, Alexander; Gerstenberger, Axel, and Wall, Wolfgang A. 3D fluid–structure-contact interaction based on a combined XFEM FSI and dual mortar contact approach. *Computational Mechanics*, 46(1):53–67, jun 2010. ISSN 1432-0924.
- Melenk, J M and Babuška, I. The partition of unity finite element method: Basic theory and applications. *Computer Methods in Applied Mechanics and Engineering*, 139(1):289–314, 1996. ISSN 0045-7825.
- Mitchell, Thomas M. *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition, 1997. ISBN 0070428077, 9780070428072.
- Moës, N; Gravouil, A, and Belytschko, T. Non-planar 3D crack growth by the extended finite element and level setsPart I: Mechanical model. *International Journal for Numerical Methods in Engineering*, 53(11):2549–2568, 2002.
- Moës, Nicolas; Dolbow, John, and Belytschko, Ted. A finite element method for crack growth without remeshing. *International Journal for Numerical Methods in Engineering*, 46(1):131–150, 1999.
- Nasri, Ahmad H. NURBS from the Pioneers: An Introduction to NURBS with Historical Perspective; David F. Rogers (Ed.); Morgan Kaufmann, San Francisco, 2001, 300 pages, ISBN 1-55860-669-6, \$33.95. *CAD*, 35(1):121–122, 2003.
- Natarajan, Sundararajan and Song, Chongmin. Representation of singular fields without asymptotic enrichment in the extended finite element

## BIBLIOGRAPHY

---

- method. *International Journal for Numerical Methods in Engineering*, 96 (13):813–841.
- Natarajan, Sundararajan; Ooi, Ean Tat; Chiong, Irene, and Song, Chongmin. Convergence and accuracy of displacement based finite element formulations over arbitrary polygons: Laplace interpolants, strain smoothing and scaled boundary polygon formulation. *Finite Elements in Analysis and Design*, 85:101–122, 2014. ISSN 0168-874X.
- Natarajan, Sundararajan; Wang, JunChao; Song, Chongmin, and Birk, Carolin. Isogeometric analysis enhanced by the scaled boundary finite element method. *Computer Methods in Applied Mechanics and Engineering*, 283:733–762, 2015. ISSN 0045-7825.
- Nguyen, Vinh Phu; Kerfriden, Pierre, and Bordas, Stéphane P A. Two- and three-dimensional isogeometric cohesive elements for composite delamination analysis. *Composites Part B: Engineering*, 60:193–212, 2014. ISSN 1359-8368.
- Nguyen, Vinh Phu; Anitescu, Cosmin; Bordas, Stéphane P A, and Rabczuk, Timon. Isogeometric analysis: An overview and computer implementation aspects. *Mathematics and Computers in Simulation*, 117: 89–116, 2015. ISSN 0378-4754.
- Nguyen-Thanh, N; Kiendl, J; Nguyen-Xuan, H; Wüchner, R; Bletzinger, K U; Bazilevs, Y, and Rabczuk, T. Rotation free isogeometric thin shell analysis using PHT-splines. *Computer Methods in Applied Mechanics and Engineering*, 200(47):3410–3424, 2011a. ISSN 0045-7825.

## BIBLIOGRAPHY

---

- Nguyen-Thanh, N; Nguyen-Xuan, H; Bordas, S P A, and Rabczuk, T. Iso-geometric analysis using polynomial splines over hierarchical T-meshes for two-dimensional elastic solids. *Computer Methods in Applied Mechanics and Engineering*, 200(21):1892–1908, 2011b. ISSN 0045-7825.
- Nguyen-Xuan, H; Tran, Loc V; Thai, Chien H; Kulasegaram, S, and Bordas, S P A. Isogeometric analysis of functionally graded plates using a refined plate theory. *Composites Part B: Engineering*, 64:222–234, 2014. ISSN 1359-8368.
- Oden, J and Demkowicz, Leszek. Advances in adaptive improvements: Survey of adaptive finite element methods in computational mechanics. *Accuracy Estimates and Adaptive Refinements in Finite Element Computations*, pages 1–43, 1989.
- Oh, Hae-Soo; Kim, Hyunju, and Jeong, Jae Woo. Enriched isogeometric analysis of elliptic boundary value problems in domains with cracks and/or corners. *International Journal for Numerical Methods in Engineering*, 97(3):149–180.
- Olshen, Leo Breiman; Jerome Friedman; Charles J. Stone; R.A. *Classification and Regression Trees*. Taylor & Francis, 1984.
- Ooi, E T and Yang, Z J. A hybrid finite element-scaled boundary finite element method for crack propagation modelling. *Computer Methods in Applied Mechanics and Engineering*, 199(17):1178–1192, 2010. ISSN 0045-7825.

## BIBLIOGRAPHY

---

- Ooi, E T; Shi, M; Song, C; Tin-Loi, F, and Yang, Z J. Dynamic crack propagation simulation with scaled boundary polygon elements and automatic remeshing technique. *Engineering Fracture Mechanics*, 106: 1–21, 2013. ISSN 0013-7944.
- Ooi, Ean Tat; Song, Chongmin; Tin-Loi, Francis, and Yang, Zhenjun. Polygon scaled boundary finite elements for crack propagation modelling. *International Journal for Numerical Methods in Engineering*, 91(3):319–342.
- Osher, Stanley and Sethian, James A. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79(1):12–49, 1988. ISSN 0021-9991.
- Owen, Steven. A Survey of Unstructured Mesh Generation Technology. *7th International Meshing Roundtable*, 3, 2000.
- P. WOLF, John and Song, Chongmin. Finite-Element Modelling of Unbounded Media. 1996.
- Pan, R J. Explicit matrix representation for NURBS curves and surfaces and its algorithm. *Chinese Journal of Computers*, 24(4):358–366, 2001.
- Parvizian, J; Düster, A, and Rank, E. Topology optimization using the finite cell method. *Optimization and Engineering*, 13(1):57–78, mar 2012. ISSN 1573-2924.
- Parvizian, Jamshid; Düster, Alexander, and Rank, Ernst. Finite cell

## BIBLIOGRAPHY

---

- method. *Computational Mechanics*, 41(1):121–133, dec 2007. ISSN 1432-0924.
- Passieux, J C; Gravouil, A; Réthoré, J, and Baietto, M C. Direct estimation of generalized stress intensity factors using a three-scale concurrent multigrid X-FEM. *International Journal for Numerical Methods in Engineering*, 85(13):1648–1666.
- Peng, Xuan; Atroshchenko, Elena; Bordas, Stéphane;. Damage tolerance assessment directly from CAD: (extended) isogeometric boundary element methods (XIGABEM). In *6th International Conference on Advanced Computational Methods in ENgineering*, 2014.
- Piegl, L and Tiller, W. Curve and surface constructions using rational B-splines. *CAD*, 19(9):485–498, 1987.
- Piegl, L and Tiller, W. *The NURBS book*. Springer, 2nd editio edition, 1997.
- Prudhomme, Serge; Chamoin, Ludovic; Dhia, Hachmi Ben, and Bauman, Paul T. An adaptive strategy for the control of modeling error in two-dimensional atomic-to-continuum coupling simulations. *Computer Methods in Applied Mechanics and Engineering*, 198(21):1887–1901, 2009. ISSN 0045-7825.
- Qian, Xiaoping and O., Sigmund. Isogeometric shape optimization of photonic crystals via Coons patche. *computer Methods in Applied Mechanics and Engineering*, 200:2237–2255, 2011.

## BIBLIOGRAPHY

---

- Qin, K H. Matrix formulae for NURBS curves/surfaces and their applications. *Chinese Journal of Computers*, 19(12):941–947, 1996.
- Rabczuk, T and Belytschko, T. Cracking particles: a simplified meshfree method for arbitrary evolving cracks. *International Journal for Numerical Methods in Engineering*, 61(13):2316–2343.
- Rabczuk, Timon and Zi, Goangseup. A Meshfree Method based on the Local Partition of Unity for Cohesive Cracks. *Computational Mechanics*, 39(6):743–760, may 2007. ISSN 1432-0924.
- Rannou, J; Gravouil, A, and Baïetto-Dubourg, M C. A local multigrid X-FEM strategy for 3-D crack propagation. *International Journal for Numerical Methods in Engineering*, 77(4):581–600.
- Rengier, F; Mehndiratta, A; von Tengg-Kobligh, H; Zechmann, C M; Unterhinninghofen, R; Kauczor, H.-U., and Giesel, F L. 3D printing based on imaging data: review of medical applications. *International Journal of Computer Assisted Radiology and Surgery*, 5(4):335–341, jul 2010. ISSN 1861-6429.
- Roura, Salvador. Fibonacci BSTs: A new balancing method for binary search trees. *Theoretical Computer Science*, 482:48–59, 2013. ISSN 03043975.
- Saeed Iqbal; Graham .F. Carey, . Neural nets for mesh assessment. Technical report, DOD High Performance Computing Modernization Program Ofc, Programming Environment & Training (PET), 1010 North Glebee Road Suite 510, Arlington, VA, 22201, 2005.



## BIBLIOGRAPHY

---

- Schillinger, D; Düster, A, and Rank, E. The hp-d-adaptive finite cell method for geometrically nonlinear problems of solid mechanics. *International Journal for Numerical Methods in Engineering*, 89(9):1171–1202.
- Scott, M A; Simpson, R N; Evans, J A; Lipton, S; Bordas, S P A; Hughes, T J R, and Sederberg, T W. Isogeometric boundary element analysis using unstructured T-splines. *Computer Methods in Applied Mechanics and Engineering*, 254(0):197–221, 2013.
- Sederberg, Thomas W; Zheng, Jianmin; Bakenov, Almaz, and Nasri, Ahmad. T-splines and T-NURCCs. *ACM Trans. Graph.*, 22(3):477–484, jul 2003. ISSN 0730-0301.
- Selimovic, Ilijas. On NURBS algorithms using tangent cones. *Computer Aided Geometric Design*, 26(7):772–778, 2009. ISSN 0167-8396.
- Sevilla, Ruben and Fernández-Méndez, Sonia. Numerical integration over 2D NURBS-shaped domains with applications to NURBS-enhanced FEM. *Finite Elements in Analysis and Design*, 47(10):1209–1220, 2011.
- Shen, Jingjing; Busé, Laurent; Alliez, Pierre, and Dodgson, Neil. A Line/Trimmed NURBS Surface Intersection Algorithm Using Matrix Representations. *Computer Aided Geometric Design*, 48:1–16, 2016. ISSN 01678396.
- Shewchuk, Jonathan R. *Delaunay Refinement Mesh Generation*. PhD thesis, Carnegie-Mellon University Pittsburgh, 1997.

## BIBLIOGRAPHY

---

- Simpson, R N; Bordas, S P A; Trevelyan, J, and Rabczuk, T. A two-dimensional Isogeometric Boundary Element Method for elastostatic analysis. *Computer Methods in Applied Mechanics and Engineering*, 209-212:87–100, 2012. ISSN 0045-7825.
- Simpson, R N; Bordas, S P A; Lian, H, and Trevelyan, J. An isogeometric boundary element method for elastostatic analysis: 2D implementation aspects. *Computers & Structures*, 118(0):2–12, 2013.
- Simpson, R N; Scott, M A; Taus, M; Thomas, D C, and Lian, H. Acoustic isogeometric boundary element analysis. *Computer Methods in Applied Mechanics and Engineering*, 269:265–290, 2014. ISSN 0045-7825.
- Song, C. The scaled boundary finite element method in structural dynamics. *International Journal for Numerical Methods in Engineering*, 77: 1139–1171, 2009.
- Song, Chongmin. A matrix function solution for the scaled boundary finite-element equation in statics. *Computer Methods in Applied Mechanics and Engineering*, 193(2326):2325–2356, 2004.
- Song, Chongmin and Wolf, John P. The scaled boundary finite-element method as consistent infinitesimal finite-element cell method for elastodynamics. *Computer Methods in Applied Mechanics and Engineering*, 147(34):329–355, 1997.
- Song, Chongmin and Wolf, John P. Semi-analytical representation of stress singularities as occurring in cracks in anisotropic multi-materials

## BIBLIOGRAPHY

---

- with the scaled boundary finite-element method. *Computers & Structures*, 80(2):183–197, 2002. ISSN 0045-7949.
- Srivastava, Nitish; Hinton, Geoffrey; Krizhevsky, Alex; Sutskever, Ilya, and Salakhutdinov, Ruslan. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15: 1929–1958, 2014.
- Staten, M L; Canann, S A, and Owen, S J. BMSweep: Locating Interior Nodes During Sweeping. *Engineering with Computers*, 15(3):212–218, sep 1999. ISSN 1435-5663.
- Stefano, T and Politecnico, D T. The Future and the Evolution of CAD, 2010.
- Strouboulis, T; Copps, K, and Babuška, I. The generalized finite element method. *Computer Methods in Applied Mechanics and Engineering*, 190 (32):4081–4193, 2001. ISSN 0045-7825.
- Sukumar, N; Chopp, D L; Moës, N, and Belytschko, T. Modeling holes and inclusions by level sets in the extended finite-element method. *Computer Methods in Applied Mechanics and Engineering*, 190(46):6183–6200, 2001. ISSN 0045-7825.
- Sutradhar, Alok; H. Paulino, Glaucio, and Gray, Leonard. *The Symmetric Galerkin Boundary Element Method*. 2008.
- Szabo, B; Duster.A, , and Rank, E. *The p-version of the finite element method*, volume 1. Wiley, New York, 2004.

## BIBLIOGRAPHY

---

Szegedy, Christian; Liu, Wei; Jia, Yangqing; Sermanet, Pierre; Reed, Scott E; Anguelov, Dragomir; Erhan, Dumitru; Vanhoucke, Vincent, and Rabinovich, Andrew. Going Deeper with Convolutions. *CoRR*, abs/1409.4, 2014.

Tabarraei, A and Sukumar, N. Extended finite element method on polygonal and quadtree meshes. 197(5):425–438, 2008.

Tao, Longbin; Song, Hao, and Chakrabarti, Subrata. Scaled boundary FEM solution of short-crested wave diffraction by a vertical cylinder. *Computer Methods in Applied Mechanics and Engineering*, 197(1):232–242, 2007. ISSN 0045-7825.

TAUTGES, T J; BLACKER, T, and MITCHELL, S A. THE WHISKER WEAVING ALGORITHM: A CONNECTIVITY-BASED METHOD FOR CONSTRUCTING ALL-HEXAHEDRAL FINITE ELEMENT MESHES. *International Journal for Numerical Methods in Engineering*, 39(19): 3327–3349.

Timoshenko, S and Goodier, J N. *Theory of Elasticity*. McGraw-Hill Book Company, New York, 2nd edition, 1951.

U.S. Product Data Association;, . Initial Graphics Exchange Specification 5.3, 2006.

Vu, T H and Deeks, A J. A p-adaptive scaled boundary finite element method based on maximization of the error decrease rate. *Computational Mechanics*, 41(3):441–455, 2008.

## BIBLIOGRAPHY

---

- Vu, Thu Hang and Deeks, Andrew J. Using fundamental solutions in the scaled boundary finite element method to solve problems with concentrated loads. *Computational Mechanics*, 53(4):641–657, apr 2014. ISSN 1432-0924.
- Wall, Wolfgang A; Frenzel, Moritz A, and Cyron, Christian. Isogeometric structural shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 197(33):2976–2988, 2008. ISSN 0045-7825.
- Wang, D; Hassan, O; Morgan, K, and Weatherill, N P. Enhanced remeshing from STL files with applications to surface grid generation. *Communications in Numerical Methods in Engineering*, 23(3):227–239, 2007.
- Wang, Dongdong and Xuan, Junchang. An improved NURBS-based isogeometric analysis with enhanced treatment of essential boundary conditions. *Computer Methods in Applied Mechanics and Engineering*, 199(3740):2425–2436, 2010.
- Wang, G X; Shu, Q L; Wang, J, and Wang, W S. Algorithm of Fast Evaluation and Derivation for the Technique of NURBS Direct Interpolation. *Journal of Northeastern University (Natural Science)*, 33(7):1021–1024, 2012.
- Warde-Farley, David; Rabinovich, Andrew, and Anguelov, Dragomir. Self-informed neural network structure learning. *CoRR*, abs/1412.6, 2014.
- Wardle, Leigh. An Introduction to the Boundary Element Method. In Noye, John, editor, *Computational Techniques for Differential Equations*, vol-

## BIBLIOGRAPHY

---

- ume 83 of *North-Holland Mathematics Studies*, pages 525–551. North-Holland, 1984.
- Watson, D F. Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes\*. *The Computer Journal*, 24(2):167–172, 1981.
- Weisberg, David E. *The Engineering Design Revolution, The People, Companies and Computer Systems That Changed Forever the Practice of Engineering*, 2008.
- Williams, M L. On the stress distribution at the base of a stationary crack. *J. appl. Mech., Trans. Am. Soc. mech. Engrs*, 79:109, 1957.
- Wolf, J P. *The Scaled Boundary Finite Element Method*. WILEY, England, 1st editio edition, 2003.
- Wolf, J P and Song, C. The scaled boundary fnite-element method - a primer: derivations. *Computer & structures*, 78:191–210, 1999.
- Wolfgang, Dornishch and Sven, Klinkel. Boundary Conditions and Multi-Patch Connections in Isogeometric Analysis. *Proceedings in Applied Mathematics and Mechanics*, 11:207–208, 2011.
- Yang, P and Qian, Xiaoping. A spline-based approach to heterogeneous objects design and analysis. *CAD*, 39(2):95–111, 2007.
- Yang, Z J; Zhang, Z H; Liu, G H, and Ooi, E T. An h-hierarchical adaptive scaled boundary finite element method for elastodynamics. *Computers & Structures*, 89(13):1417–1429, 2011. ISSN 0045-7949.

## BIBLIOGRAPHY

---

- Yerry, M A and Shephard, M S. A Modified Quadtree Approach To Finite Element Mesh Generation. *IEEE Computer Graphics and Applications*, 3(1):39–46, jan 1983. ISSN 0272-1716.
- Yerry, Mark A and Shephard, Mark S. Automatic three-dimensional mesh generation by the modified-octree technique. *International Journal for Numerical Methods in Engineering*, 20(11):1965–1990.
- Zhang, Y; Bazilevs, Y; Goswami, S; Bajaj, C, and Hughes, T J R. Patient-specific vascular NURBS modeling for isogeometric analysis of blood flow. *Computer Methods in Applied Mechanics and Engineering*, 196: 2943–2959, 2007.
- Zhao, Zhiye. Error estimation in adaptive BEM by postprocessing interpolation. *Communications in Numerical Methods in Engineering*, 14(7): 633–645, 1998.
- Zhao, Zhiye and Wang, Xin. Error estimation and h adaptive boundary elements. *Engineering Analysis with Boundary Elements*, 23(10):793–803, 1999. ISSN 0955-7997.
- Zienkiewicz, O C and Zhu, J Z. A simple error estimator and adaptive procedure for practical engineering analysis. *International Journal for Numerical Methods in Engineering*, 24(2):337–357, a.
- Zienkiewicz, O C and Zhu, J Z. The superconvergent patch recovery and a posteriori error estimates. Part 1: The recovery technique. *International Journal for Numerical Methods in Engineering*, 33(7):1331–1364, b.

## BIBLIOGRAPHY

---

- Zienkiewicz, O C and Zhu, J Z. The superconvergent patch recovery and a posteriori error estimates. Part 2: Error estimates and adaptivity. *International Journal for Numerical Methods in Engineering*, 33(7):1365–1382, c.
- Zienkiewicz, O C; Zhu, J Z, and Gong, N G. Effective and practical hp-version adaptive analysis procedures for the finite element method. *International Journal for Numerical Methods in Engineering*, 28(4):879–891, 1989. ISSN 1097-0207.
- Zienkiewicz, O C; Boroomand, B, and Zhu, J Z. Recovery procedures in error estimation and adaptivity Part I: Adaptivity in linear problems. *Computer Methods in Applied Mechanics and Engineering*, 176(1):111–125, 1999. ISSN 0045-7825.
- Zienkiewicz, O C; Taylor, R L, and Zhu, J Z. 14 - Adaptive finite element refinement. In Zienkiewicz, O C; Taylor, R L, and Zhu, J Z, editors, *The Finite Element Method Set (Sixth Edition)*, pages 500–524. Butterworth-Heinemann, Oxford, sixth edit edition, 2005. ISBN 978-0-7506-6431-8.