

Abdi, Meisam (2015) Evolutionary topology optimization of continuum structures using X-FEM and isovalues of structural performance. PhD thesis, University of Nottingham.

**Access from the University of Nottingham repository:**

<http://eprints.nottingham.ac.uk/31226/1/Meisam%20Abdi%20Thesis.pdf>

**Copyright and reuse:**

The Nottingham ePrints service makes this work by researchers of the University of Nottingham available open access under the following conditions.

- Copyright and all moral rights to the version of the paper presented here belong to the individual author(s) and/or other copyright owners.
- To the extent reasonable and practicable the material made available in Nottingham ePrints has been checked for eligibility before being made available.
- Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way.
- Quotations or similar reproductions must be sufficiently acknowledged.

Please see our full end user licence at:

[http://eprints.nottingham.ac.uk/end\\_user\\_agreement.pdf](http://eprints.nottingham.ac.uk/end_user_agreement.pdf)

**A note on versions:**

The version presented here may differ from the published version or from the version of record. If you wish to cite this item you are advised to consult the publisher's version. Please see the repository url above for details on accessing the published version and note that access may require a subscription.

For more information, please contact [eprints@nottingham.ac.uk](mailto:eprints@nottingham.ac.uk)

# **Evolutionary Topology Optimization of Continuum Structures using X-FEM and Isovalues of Structural Performance**

**Meisam Abdi, BSc, MSc**

**Thesis submitted to the University of Nottingham  
for the degree of Doctor of Philosophy**

**December 2015**

**Dedicated to my parents and my wife**

# Abstract

In the last three decades, advances in modern manufacturing processes, such as additive manufacturing (AM) on one hand and computational power on the other hand, has resulted in a surge of interest in topology optimization as a means of designing high performance components with high degrees of geometrical complexity. Topology optimization seeks to find the best design for a structure by optimally distributing material in a design space. Therefore not only the shape and size of the structure, but also the connectivity of the structure changes during the topology optimization process. As a result, the solution of a topology optimization problem might be represented with a high degree of geometrical complexity as it is not dependent on the initial geometry. The finite element method (FEM) is a powerful numerical analysis technique that was developed to solve complex solid mechanics problems. Many topology optimization approaches use FEM to calculate the response of the structure during the optimization process and some of them, called “element based-methods”, are integrated with FEM to use the properties of finite elements as design variables in the optimization. The solutions of such approaches are usually represented by a uniform finite element mesh that bears no relation to the final geometry and hence they don’t provide an accurate representation of the design boundary. The solution from topology optimization must therefore go through further post processing stages to obtain a manufacturable design. The post processing stages which can include smoothing and shape optimization are costly and time-consuming and may result in the structure becoming less optimal. With traditional manufacturing processes this is acceptable as the manufacturing constraints prevent the optimized design from being manufactured so some re-analysis is necessary. With additive manufacturing, however, this restriction is removed, which means a topology optimization resulting in a manufacturable design is highly desirable.

Evolutionary structural optimization (ESO) is an element based topology optimization approach which operates by systematically removing inefficient

material from the structure until the optimization objective achieves convergence. Due to the intuitive nature of ESO, this method is simple to be programmed and can be easily integrated with FEM or other numerical analysis techniques; thus it is suitable for complex geometries represented with FEM. During the last two decades ESO and its extensions, such as bi-directional ESO (BESO), have been successfully used for many topology optimization problems such as stiffness design, design of compliant mechanisms, heat conduction problems and frequency problems. However, being an element based method, the drawback of poor boundary representation remains.

The extended finite element method (X-FEM) is an extension of the classical FEM that was developed to represent discontinuities, such as cracks and material-void interfaces, inside finite elements. X-FEM can be employed in topology optimization problems to handle the material-void discontinuity introduced by the evolving boundary during the optimization process which potentially enables a sub-element boundary representation. This requires an implicit boundary representation, such as level-set method with the benefits of better computational accuracy through the optimization, more optimized solution and smoother boundaries for direct to manufacture.

In this work a new method of evolutionary structural optimization is proposed in which X-FEM is employed for the more smooth and accurate representation of the design boundary. Linear finite elements are used to discretize the design space. These include 4-node quadrilateral elements in 2D modelling and 8-node hexahedral elements in 3D modelling. To implement the X-FEM, an implicit boundary representation using isoline and isosurface approaches is used. The proposed method which is called “Iso-XFEM” is implemented for various topology optimization problems, including the stiffness design of 2D and 3D structures, stiffness design with additional displacement constraint and topology optimization of geometrically nonlinear problems. The solutions of the Iso-XFEM method are compared with those obtained using BESO, as a representative FE based method. The results confirm a significant improvement in boundary representation of the solutions when compared against BESO, and also demonstrate the feasibility of the application of the proposed method to complex real-life structures and to different objectives. All the programs used to generate

topology optimised solutions using the proposed method and its modifications are developed by the author. These include topology optimization codes, linear and non-linear FEA, and 2D and 3D X-FEM integration schemes.

# Publications

## Journal papers

- Abdi, M., Wildman, R., Ashcroft, I., 2014. Evolutionary topology optimization using extended finite element method and isolines. *Engineering Optimization*, 46 (5), pp. 628-647.
- Abdi, M., Wildman, R., Ashcroft, I., 2015. A generalized evolutionary optimization approach for the design of 2D and 3D structures using isovalues of structural performance and X-FEM. *Optimization and Engineering*. Submitted.

## Book Chapters

- Abdi, M., Ashcroft, I., Wildman, R., 2014. An X-FEM based approach for topology optimization of continuum structures. *Advances in Intelligent Systems and Computing*, 256, pp. 277-289.

## Conference papers

- Abdi, M., Ashcroft, I., Wildman, R., 2015. Topology Optimization of Geometrically Non-Linear Structures Using Iso-XFEM Method. *Key Engineering Materials*, 627, pp. 121-124.
- Abdi, M., Ashcroft, I., Wildman, R., 2014. High resolution topology design with Iso-XFEM. *Proceedings of the 25th Solid Freeform Fabrication Symposium (SFF2014)*, pp. 1288-1303, Texas. Reviewed.

- Abdi, M., Ashcroft, I., Wildman, R., 2013. Application of X-FEM in Isoline/Isosurface Based Topology Optimization. 10th World Congress on Structural and Multidisciplinary Optimization, Florida, USA.
  
- Abdi, M., Ashcroft, I., Wildman, R., 2012. X-FEM Based Topological Optimization Method. SIMULTECH 2012 : 2nd International Conference on Simulation and Modeling Methodologies, Technologies and Applications, Rome, Italy



# Acknowledgements

I am deeply indebted to my supervisors Prof Ian Ashcroft and Prof Ricky Wildman for their patience, support and guidance throughout the entire research. This work would not have been possible without their continued support, motivation and assistance.

I am grateful to Prof Adib Becker, my internal examiner, for his encouragement and guidance.

I would like to thank members of Additive Manufacturing and 3D Printing Research Group (3DPRG), University of Nottingham. These include Prof Richard Hague, Dr Chris Tuck, Dr David Brackett, Dr Ajit Panesar, Dr Adedeji Aremu, Mirela Axinte, Jill Thurman, Mark East, Mark Hardy and Joseph White.

Thanks to staff of Wolfson school of Mechanical and Manufacturing Engineering, Loughborough University, for their help and support during the first year of my PhD at Loughborough University.

I would like to thank my colleagues and friends in University of Nottingham and Loughborough University for their help and assistance.

I would also like to thank my B.Sc. and M.Sc. supervisors Prof Tajbakhsh Navid Chakherlou and Dr Ardeshir Karami Mohammadi, for their encouragement.

# Contents

<b>Abstract</b>	ii
<b>Publications</b>	v
<b>Acknowledgements</b>	vii
<b>List of Figures</b>	xiii
<b>List of Tables</b>	xvi
<b>Nomenclature</b>	xvii
<b>1 Introduction</b>	<b>1</b>
1.1 General	1
1.2 Aims and objectives of the research	4
1.3 Significance and novelty of the research	4
1.4 Layout of the thesis	5
<b>2 Literature Review Part I: Review of Structural Topological Optimization</b>	<b>7</b>
2.1 Structural optimization	7
2.1.1 Mathematical representation of a structural optimization problem	8
2.1.2 Classification of structural optimization problems	9
2.2 Topology optimization	10
2.2.1 History	10
2.2.2 Classification of topology optimization methods	12
2.2.3 Common approaches for topology optimization of continuum structures	14
2.2.3.1 Homogenization	14
2.2.3.2 SIMP	16
2.2.3.3 Evolutionary Structural Optimization	18
2.2.3.4 Genetic Algorithms	21
2.2.3.5 Level set method	23

2.2.4	Numerical instabilities	25
2.2.4.1	Checkerboard	25
2.2.4.2	Mesh dependence	26
2.2.4.3	Local minima	27
2.3	Summary and conclusion	28
<b>3</b>	<b>Literature Review Part II: Numerical Methods for Structural Analysis</b>	<b>30</b>
3.1	Mesh-based and meshfree methods	31
3.2	Finite Element Method	32
3.2.1	History	32
3.2.2	Definition of FEM	33
3.2.3	Common linear elements for structural optimization	34
3.2.4	Application of FEM in topology optimization approaches	35
3.3	Adaptive mesh refinement techniques	36
3.3.1	Review of adaptive mesh refinement techniques	36
3.3.2	Application of mesh refinement techniques in topology optimization	37
3.4	Fixed Grid FEA	38
3.4.1	Review of fixed grid FEA	38
3.4.2	Application of fixed grid FEA in topology optimization	39
3.5	Element Free Galerkin Method	40
3.5.1	Review of element free Galerkin method	40
3.5.2	Application of Meshless Methods to Topology Optimization	41
3.6	X-FEM	43
3.6.1	X-FEM approximation	43
3.6.2	X-FEM for Modelling Holes and Inclusions	45
3.6.3	Application to Topology Optimization	46
3.7	Summary and conclusions	47
<b>4</b>	<b>Two Dimensional Topology Optimization Studies</b>	<b>48</b>
4.1	Introduction to Iso-XFEM approach	48

4.2	Isoline design approach	49
4.3	X-FEM schemes for structural optimization	53
4.4	2D X-FEM integration schemes	54
4.4.1	Triangulation of boundary elements	55
4.4.2	Gauss quadrature integration method	56
4.5	Stiffness design using isovalue-based evolutionary optimization process	58
4.5.1	Stiffness design in evolutionary structural optimization	58
4.5.2	Stiffness design using Iso-XFEM method	59
4.5.3	Steps in the Iso-XFEM algorithm	61
4.6	Numerical experiments	63
4.6.1	Experiment 1: cantilever plate	61
4.6.1.1	Preliminary examination of convergence	65
4.6.1.2	Evaluating Iso-XFEM solutions	68
4.6.1.3	Comparison with BESO solutions	69
4.7	Experiment 2: C clip	75
4.8	Summary and conclusions	78
<b>5</b>	<b>Three Dimensional Topology Optimization Studies</b>	<b>79</b>
5.1	Isosurface design approach	79
5.2	3D X-FEM approach	82
5.3	Evolutionary optimization algorithm	85
5.4	Test cases	88
5.4.1	Test case 3D-1: 3D cantilever beam	89
5.4.2	Test case 3D-2: C clip	92
5.4.2.1	2D and 3D solutions	93
5.4.2.2	Comparison of solutions of the two evolutionary optimization algorithms	93
5.4.2.3	Effect of volume evolution rate	94
5.4.3	Test case 3D-3: an aerospace swing arm	97
5.4.3.1	Experiment 1	98
5.4.3.2	Experiment 2	102
5.4.3.3	Comparison and discussion	103

5.5	Application of Iso-XFEM to problems with alternative objectives or constraints	105
5.5.1	Stiffness design with displacement constraint	106
5.5.1.1	Calculating Lagrangian multiplier	108
5.5.1.2	2D example	109
5.5.1.3	3D example	111
5.6	Summary and conclusions	115
<b>6</b>	<b>Topology Optimization of Geometrically Non-linear Structures</b>	<b>117</b>
6.1	Introduction	117
6.1.1	Types of structural non-linearity	119
6.1.2	Incremental-iterative approach	120
6.1.3	Geometrically non-linear behaviour of a continuum body	121
6.2	Formulation of equation of motion	123
6.2.1	Continuous form of the equilibrium equation	123
6.2.2	Finite element formulation	124
6.2.3	X-FEM for geometrically non-linear behaviour	125
6.3	Stiffness design	126
6.3.1	Objective function and structural performance criteria	126
6.3.2	Filter scheme for Iso-XFEM	127
6.3.1	Iso-XFEM procedure for geometrically non-linear structures	128
6.4	Test cases	128
6.4.1	Validation of the FE Matlab code	129
6.4.2	Test case NL-1: nonlinear cantilever plate	131
6.4.3	Test case NL-2: slender beam	135
6.5	Summary and conclusions	138
<b>7</b>	<b>Discussion</b>	<b>140</b>
7.1	General summary and major findings	141
7.2	Validating Iso-XFEM solutions	145
7.3	Numerical instabilities	148
7.4	Surface roughness of the solutions	150

7.5	Computational efficiency of Iso-XFEM	151
7.6	Final remarks	152
<b>8</b>	<b>Conclusions and Future Work</b>	<b>153</b>
8.1	Achievements	153
8.2	Conclusions	154
8.3	Future work	155
	<b>List of References</b>	<b>157</b>
<b>Appendix A:</b>	<b>Matlab code for 2D version of Iso-XFEM optimization method</b>	<b>168</b>
<b>Appendix B:</b>	<b>Matlab code for 3D version of Iso-XFEM optimization method</b>	<b>176</b>
<b>Appendix C:</b>	<b>Matlab code for 3D version of BESO</b>	<b>191</b>
<b>Appendix D:</b>	<b>Matlab code for Non-linear FEA</b>	<b>196</b>

# List of figures

2.1	Size, shape and topology optimization of a typical structure.....	10
2.2	Rectangular microstructure.....	15
2.3	Flowchart of SIMP.....	18
2.4	Flowchart of BESO.....	20
2.5	(a) A chromosome (b) digital form of chromosome in mesh (c) the ...	22
2.6	(a) Level set function (b) material and void phases. ....	24
2.7	(a) Design domain and its boundary conditions (b) solution with .....	26
2.8	(a) Design domain and its boundary conditions (b) BESO solution ....	27
3.1	(a) Mesh-based approximation and (b) meshfree approximation. ....	32
3.2	Main steps of finite element procedure. ....	33
3.3	Common 2D and 3D linear elements: (a) triangular element (b) .....	35
3.4	Illustrating mesh refinement for a triangular mesh (a) domain initially	36
3.5	A material/void interface problem in a fixed grid mesh. ....	39
3.6	(a) Typical boundary elements for area ratio=0.50. (b) Their density. .	39
3.7	Illustrating numerical model of element free Galerkin method. ....	41
3.8	BESO topology designs for cantilever problems (a) cantilever loaded	42
3.9	X-FEM model of a crack. ....	44
3.10	X-FEM representation of solid/void interfaces. ....	45
4.1	General idea of Iso-XFEM method. ....	49
4.2	Design boundary represented by intersection of a SP with the MLP...	50
4.3	A 2D design domain represented by its relative structural. ....	50
4.4	Application of isoline design approach in topology design of a .....	52
4.5	The elements are classified into 3 groups by superimposing the .....	53
4.6	Solid sub-domain of the boundary elements are partitioned into .....	54
4.7	A boundary element with typical values for relative structural .....	55
4.8	Gauss quadrature in triangles: (a) standard triangle (b) second order ..	57
4.9	X-FEM integration scheme (a) sub-triangles with midline Gauss .....	57

4.10	(a) A solid element represented by classical FEM. (b) The solid .....	58
4.11	Flowchart of optimization algorithm. ....	63
4.12	The two test cases. ....	64
4.13	Test case 2D-1: (a) evolution history of objective function (SE) and ..	66
4.14	Test case 2D-2: (a) evolution history of objective function (SE) and ..	67
4.15	XFEM solution discretized using a converged, fine mesh and .....	68
4.16	Changes in strain energy (SE) by reducing the mesh size (h) in test ...	71
4.17	BESO solutions before/after smoothing. ....	73
4.18	Iso-XFEM solutions before/after smoothing. ....	74
4.19	Design domain, FE mesh, loads and boundary conditions used for the	75
4.20	Evolution of the topology of the C clip. ....	76
4.21	Converged solution for $VF = 0.4$ at iteration 120. ....	77
4.22	Evolution history of the objective function (SE) and volume fraction .	77
5.1	Illustration of isosurfaces of SED for a 3D cantilever beam. ....	80
5.2	Classification of hexahedral elements in isosurface design approach...	81
5.3	16 different hexahedral element topology states according to .....	81
5.4	Decomposing a boundary hexahedral element: (a) a boundary .....	84
5.5	Flowchart of the proposed optimization method. ....	88
5.6	Test case 3D-1: a 3D cantilever plate with unit distributed force. ....	90
5.7	Topology optimization solution of the 3D cantilever plate. ....	90
5.8	Iteration history of strain energy (SE) and volume fraction (VF) of ...	90
5.9	Topology optimization solution of 2D cantilever plate. ....	91
5.10	Comparison of the solutions of the 3D cantilever plate for different ..	92
5.11	Converged solutions of test case 3D-2 for $VF = 0.4$ : (a) 2D design (b)	95
5.12	Evolution history of the objective function (SE) and volume fraction..	96
5.13	Design domain and non-design domain of the aerospace arm, loads ..	97
5.14	The resulting topologies of the aerospace arm for a range of volume ..	98
5.15	Evolution history of the objective function (SE) and volume fraction .	99
5.16	Converged solutions of experiment 1: (a) Iso-XFEM solution and (b)	100
5.17	Converged solutions in experiment 2: (a) Iso-XFEM solution and (b)	100
5.18	Evolution history of the objective function (SE) and volume fraction	101
5.19	Design domain of the 2D optimization problem with horizontal .....	110



5.20	Topology optimized solutions of the simply supported rectangular ....	110
5.21	Evolution histories of constrained displacement (a) and strain energy.	111
5.22	Design domain of the 3D optimization problem with horizontal .....	112
5.23	Topology optimization solutions of the 3D cantilever beam: (a) .....	113
5.24	Evolution histories of constrained displacement (a) and strain energy	114
6.1	Illustration of incremental Newton-Raphson approach. ....	121
6.2	(a) Large displacements and large rotations but small strain (b) large	121
6.3	Objective function $W^c$ for stiffness optimization of nonlinear .....	127
6.4	Illustration of linear and nonlinear large deformation of a cantilever ..	130
6.5	Load deflection plots of the geometrically nonlinear cantilever beam .	130
6.6	Design domain and boundary conditions of the geometrically .....	132
6.7	Evolution histories of objective function and volume fraction of the ...	133
6.8	Iso-XFEM solutions of the large displacement cantilever problem ....	133
6.9	Design domain and boundary conditions of the geometrically .....	134
6.10	Evolution histories of objective function and volume fraction of the ..	135
6.11	(a) Non-linear design of the beam subjected to a downward load (b) ..	136
6.12	(a) Displacement of topology 6.10a (b) displacement of topology .....	137
7.1	Comparison of the solutions obtained from Iso-XFEM with/without ..	143

# List of tables

4.1	Comparison of X-FEM solutions and regenerated NASTRAN .....	69
4.2	Iso-XFEM and BESO solutions of test case 2D-1 for a range of mesh ..	70
4.3	Iso-XFEM and BESO solutions of test case 2D-2 for a range of mesh ..	70
4.4	Comparison of the time cost of BESO and X-FEM for 100 iterations...	71
5.1	Comparison of the strain energies of 2D and 3D solutions with .....	92
5.2	Comparison of the solutions of the 2D C clip stiffness optimization for	96
5.3	Comparison of the solutions in experiment 1 and experiment 2. ....	103
5.4	Comparison of the objective values and surface roughness of the .....	103
6.1	Comparison of linear and non-linear response of the cantilever beam ...	130
6.2	Comparison of the complementary works of linear and non-linear .....	134
6.3	Comparison of the complementary works of non-linear and linear .....	137
7.1	Comparison of Optistruct and Iso-XFEM solutions .....	148

# Nomenclature

## Acronyms

2D	Two dimensional
3D	Three dimensional
4D	Four dimensional
AM	Additive Manufacturing
BESO	Bi-directional Evolutionary Structural Optimization
DEM	Diffuse Element Method
DoF	Degree of Freedom
EFGM	Element Free Galerkin Method
ESO	Evolutionary Structural Optimization
FDM	Finite Difference Method
FEA	Finite Element Analysis
FEM	Finite Element Method
FG-FEA	Fixed Grid Finite element Analysis
FVM	Finite Volume Method
GA	Genetic algorithm
LSM	Level Set Method
MLP	Minimum Level of Performance
MMA	Method of Moving Asymptotes
OCM	Optimally Criteria Method
RBF	Radial Basis Functions
RKPM	Reproducing Kernel Particle Method
SED	Strain Energy Density
SIMP	Solid Isotropic Microstructure with Penalization
SPH	Smoothed Particle Hydrodynamics
SP	Structural Performance
TL	Total Lagrangian
UL	Updated Lagrangian
VF	Volume Fraction

X-FEM      eXtended Finite Element Method

### **Greek Symbols**

$\alpha$	Relative performance function
$\alpha_e$	Element relative structural performance
$\alpha_i$	Nodal values of relative structural performance
$\epsilon_{ij}$	Green-Lagrange strain tensor
$\eta_{ij}$	Non-linear incremental strain
$\lambda$	Lagrangian multiplier
$\nu$	Poisson's ratio
$\rho$	Density
$\xi$	Natural coordinate
$\xi^{(e)}$	Area ratio of the solid part of the element
$\varphi(x)$	Level set function
$\psi(x)$	Enrichment function
$\Omega$	Domain
$\Omega_S$	Solid domain of element

### **Scalars, Tensors and Functions**

$A_S^{(e)}$	Solid area of the element
$A_t^{(e)}$	Total area of the element
$A_v^{(e)}$	Void area of the element
$C$	Compliance/ mean compliance
$C_{ijkl}$	Components of elasticity tensor
$D_S$	Solid phase of the design domain
$D_V$	Void phase of the design domain
$E$	Young's modulus
$E_v$	Young's modulus assigned to solid material
$E_v$	Young's modulus assigned to void part of the element
$ER$	Volume evolution rate
$\tilde{E}_{ijkl}$	Effective Young's modulus
$E_e^n$	Total elemental elastic and plastic strain energy

$e_{ij}$	Linear incremental strain
$f(x)$	Objective function
$g(x)$	Equality constraints
$H(x)$	Heaviside function
$h$	Minimum grid size/ element length
$h(x)$	Inequality constraints
$it$	Iteration number of the evolutionary process
$N_i$	Shape functions
$P$	Penalization power
$PER$	Performance evolution rate
$R_a$	Arithmetic mean surface roughness
$R_q$	Root mean square roughness
$r_{min}$	Filter radius
$S_e$	Sensitivity of element e
$S_j$	Sensitivity number assigned to node j
$S_{ij}$	Cartesian components of second Piola-Kirchhoff stress tensor
$t$	Time
$V$	Volume/ volume of the structure
$V^0$	Volume of the design domain
${}^0V$	Volume at initial configuration
$V^c$	Volume constraint
$v_n(x)$	Normal velocity
$W^C$	Complementary work
$w_i$	Weighting factor
$x$	Design variables
$x_{min}$	Minimum non-zero relative density assigned to void elements
$\partial D_S$	Design boundary
$\Delta t$	Time step/ time increment

### Vectors and Matrices

$B$	Displacement differentiation matrix
$B_{L0}$	Linear strain-displacement transformation matrix used in linear

infinitesimal strain analysis

$B_{L1}$	Linear strain-displacement transformation matrix which depends on the displacement
$B_{NL}$	Denotes the non-linear strain-displacement transformation matrix
C	Element elasticity matrix
F	Global force vector/ resistance force
${}^tF_e$	Element's internal force vector
K	Stiffness matrix/ Global stiffness matrix
$K_B$	Stiffness matrix of a boundary element
$K_S$	Stiffness matrix of a solid element
$K_T$	Tangent stiffness matrix
$K_0$	The usual small displacement stiffness matrix
$K_d$	Large displacement stiffness matrix
$K_\sigma$	The initial stress matrix
$k_e$	Element stiffness matrix
$k_e^0$	Stiffness matrix of the fully solid element
$k_e^s$	Stiffness contribution of the solid part of the boundary element
$k_e^v$	Stiffness contribution of the void part of the boundary element
${}^tk_\sigma$	Element's tangent stiffness matrix
R	Applied load
${}^tS$	Second Piola-Kirchhoff stress matrix
${}^t\bar{S}$	Second Piola-Kirchhoff stress vector
U	Global displacement vector
$U_{ej}$	Virtual displacement vector of element e
$u_e$	Element displacement vector
$v_e$	Element's volume
$\Delta F$	Load unbalance between the external and internal forces
$\Delta R$	Load increment
$\delta u_i$	Virtual displacement field

# Chapter 1

## Introduction

### 1.1. General

In mechanics, a structure can be defined as a combination of parts or materials which are intended to support a set of loads. Structural Design is the process of selecting materials and parts, their size and configuration, such that they provide adequate stability, strength and rigidity for the structure. In modern engineering, there is a need to find the best possible solution for an engineering problem. This is both for commercial reasons and from a government and societal push for designs that are energy efficient and have a small carbon footprint. Consequently, engineers are required to design lightweight, low-cost and high-performance structures. The main task of Structural Optimization is to achieve the best possible configuration of materials that can support service loads. However, in the growing multidisciplinary field of optimization, there is an increasing demand to find the best solution to a structure which can meet all the multidisciplinary requirements imposed by functionality and manufacturing (Wang 2003).

Traditionally, the process of structural design was based on sequential trial and error, gradually improving the existing designs. However, today's competitive industrial market requires high quality products with reduced time and cost of design and manufacturing. This requires the use of scientific methods during the process of design and manufacturing. The availability of high speed computers in recent decades and the development of numerical methods in engineering, have been significant drivers to transfer the mostly academic based field of structural optimization into the current stage of practical implementation in the industry. Currently, many industries benefit from different aspects of structural optimization.

It has been employed in mechanical, aerospace, civil, off-shore and many other fields of engineering (Iyengar & Jagadish 2005).

In a structural optimization problem, the goal could be to optimize the size, shape or topology of a structure. Topology optimization is the most general form of structural optimization in which the topology, as well as the size and shape changes during the optimization process. The aim of topology optimization of continuum structures is to find the best possible layout for a structure within a design domain under a set of loads and boundary conditions. The topology optimization methods generally operate by distributing a limited amount of material in a design space and finding the number of structural members required, and also the connectivity of these members.

The topology optimization process of continuum structures often progresses toward convergence with an increasing geometrical complexity. In order to represent the topology changes and final topological optimized solutions, the optimization algorithms are integrated with a numerical analysis method such as the finite element method (FEM) or a meshfree method. There has been a significant interest in the finite element based methods of topology optimization as many optimization algorithms can simply be integrated with the finite element framework of the structure. In the element based methods, the design domain is discretised using a finite element mesh, and each finite element, or one of its properties, is considered as a design variable. Evolution based algorithms are a group of element based methods of topology optimization that are developed to imitate natural selection. Evolutionary Structural Optimization (ESO), proposed by Xie and Steven (1993), is based on the assumption that the structure evolves to an optimum by gradually removing its inefficient material. As a heuristic method, ESO doesn't have the mathematical complexity of other optimization methods, can be simply programmed and can be applied to many complex structural optimization problems.

Despite the many advantages of element based topology optimization methods, a common drawback is the limitation in representation of the boundary of the design. In the element based methods, the geometry is represented by the finite elements. Thus, the finer elements are near the boundary, the more detailed is the



representation of the boundary. However, it is not always possible to use a fine mesh in topology optimization problems as it can dramatically increase the time cost of the analysis. On the other hand, the solutions from implementing a coarse mesh will require much more post-processing to become a smooth topology. They will also be less able to represent geometrical complexity and may result in a less optimal solution. The post-processing required which generally includes smoothing, reanalysing and shape optimization, is usually very time consuming and can result in a less optimal design.

An alternative approach to reduce the post processing required for the topologically optimized solutions is to combine the structural optimization algorithm with an adaptive mesh refinement strategy, such that it refines the mesh near the boundary where a high resolution is required and coarsens the mesh where less accuracy is required. A significant improvement of the conventional element based methods has been reported by using the adaptive mesh refinement strategies. However, generating several meshes during the optimization process is cumbersome. Also, by refining the mesh, new degrees of freedom are being introduced to the finite element model of the structure which again increases the time-cost of the optimization (Aremu 2013).

Belytschko and Black (1999), followed by Moës et al (1999), proposed a generalized version of finite element method to represent discontinuities inside finite elements without the need to refine the mesh near the discontinuities. The idea was to enrich the classical finite element approximation space with discontinuous functions of the continuous displacement field. This approach which was later known as eXtended Finite Element Method (X-FEM) has been successfully implemented to represent a variety of discontinuities including crack growth, holes and inclusions, and fluid/structure interaction. The main advantage of X-FEM is that in the problems with a progressive change in the topology of the domain during on analysis, such as crack propagation, the FE mesh does not need to be updated to track the discontinuity path; thus, reducing the computational costs and also the projection errors associated with conventional FEM.

Application of X-FEM into structural optimization problems, where the design boundary represented with a discontinuous material/void interface changes during

the optimization process, can improve the quality of optimized solutions. Although a few works have been published in this field, the studies are limited to the use of X-FEM with boundary variation methods and there has been no work addressing the application of X-FEM on material distribution methods such as evolutionary based optimization methods. As the evolutionary topology optimization methods are generally element based approaches suffering from weak boundary representation, this work represents an investigation of the application of X-FEM with an evolutionary based topology optimization algorithm, with the aim of increasing the performance of the solutions and providing high resolution topology designs.

## **1.2. Aim and objectives of the research**

The aim of this research was to develop a reliable and practical algorithm for topology optimization of continuum structures which doesn't have the limitations of the current topology optimization methods in representing the design geometry. Adopting such an algorithm has the potential to minimize the computational time and also the design cost of the engineering products. This aim was achieved by meeting the following objectives:

- Investigating the commonly used topology optimization algorithms.
- Investigating the different numerical techniques used in structural analysis.
- Developing a topology optimization method which benefit from a boundary improvement scheme.
- Extending the method to the topology optimization of 3D real-life structures, multiple objectives and non-linear problems.
- Developing Matlab codes which represent the developed topology optimization method.

## **1.3. Significance and novelty of the research**

Topology optimization as a material distribution problem has been conventionally developed in a finite element based framework. The solutions from topology

optimization which are represented by the finite elements usually require a long post-processing before they become manufacturable topologies. This study is significant in that it provides a simple and applicable method for topology optimization of continuum structures with the solutions which need the least amount of post processing before manufacturing. The advantages of the proposed method are that it is computationally efficient and results in better computational accuracy, more optimal solution and significantly smoother boundary representation than the traditional element based optimization methods. It is also demonstrated that it can be applied to real 3D structures, multiple objectives and non-linear problems. Therefore this method provides the potential engineers to reduce the computational time and the design cost of products.

The novelty of this work is that it proposes a new evolutionary optimization method which uses X-FEM with implicit boundary representation to achieve topology solutions with high resolution. Furthermore, the method is extended to enable the topology optimization of 3D real-life structures, non-linear structures and multiple objectives.

#### **1.4. Layout of the thesis**

This thesis consists of 7 chapters.

Chapter 1 presents an introduction to structural optimization, the general ideas behind the research, the aims and objectives and the significance of the research.

Chapter 2 presents a review of structural optimization by describing the different aspects of structural optimization and its mathematical basis. Topology optimization as the most challenging aspect of structural optimization, is reviewed and different methods applied to the topology optimization of continuum structures is discussed. Finally, some of the main numerical instabilities that come from the current optimization algorithms are described.

Chapter 3 presents an introduction to the various numerical methods used in structural analysis, with emphasis on those used in structural optimization. The

finite element method, adaptive mesh refinement techniques, extended finite element method and meshless methods are reviewed.

Chapter 4 presents an introduction to the proposed topology optimization method. This chapter includes the two dimensional studies of the work. An isoline method used for implicit boundary representation and an X-FEM integration scheme are presented. An evolutionary optimization algorithm combined with the isoline boundary representation and X-FEM integration scheme is proposed to complete the method. Several test cases are presented and a numerical comparison of the results with standard element-based topology optimization solutions is given.

Chapter 5 presents an extension of the method to enable the three dimensional structures to be optimized. This is achieved by implementing an isosurface model for the boundary representation and a 3D X-FEM integration scheme. Several test cases and numerical comparisons, including the application of the proposed method to a real-life engineering structure are given. A discussion section regarding the extension of the method into optimization of alternative objectives is also presented.

Chapter 6 investigates the further application of the proposed method to the topology optimization of geometrically non-linear problems. The changes required in the FE model to achieve this as well as change in the optimization method are presented. The solutions obtained from the linear and non-linear models are compared.

Chapter 7 presents a discussion of the numerical results and the optimization methodologies presented in the previous chapters.

Chapter 8 presents the achievements and general conclusions regarding the effectiveness and efficiency of the proposed method followed by suggestions and recommendations for future work.

Appendixes at the end of the thesis, present samples of Matlab codes developed in this work to represent topology optimization of continuum structures using the proposed method.

# **Chapter 2**

## **Literature Review Part I:**

# **Overview of Structural Topological Optimization**

In this chapter, a review of structural optimization is presented. A mathematical description of the general structural optimization problem is presented. Three aspects of the structural optimization problem, size, shape and topology optimization are then discussed. Finally a review of the various topology optimization strategies and different methods is presented, followed by a discussion of the issue of numerical stabilities in topology optimization methods.

### **2.1. Structural optimization**

A structural optimization problem aims to achieve the best performance for a structure while satisfying all the constraints. Similar to other optimization problems, a structural optimization problem can be described by the objectives of the problem, a set of design constraints and the design variables. The structural optimization problem can be formulated as “minimize (or maximize) the objective function subject to the constraints”. The objective is a scalar response which directs the structural optimization towards the best configuration of the independent design variables. The equation that relates the objective to the design variables is called the performance function. The performance function is continuously evaluated to monitor the system’s response by progressing the structural optimization. A solution to the structural optimization problem would be

a configuration of the design variables in which it gives the best possible value of the performance function. Design variables of a structural optimization problem can be either continuous or discrete. Structural design variables include points defining structural features, dimensional parameters and density of material in a structural region (Aremu 2013). A third component, the constraints, is used to impose certain performance criteria on a structure. The constraints could be behavioural or geometrical (Wang 2003). Below are some examples of objective functions and behavioural constraints:

- Volume or weight of the structure
- A global measure of structural performance such as stiffness (or compliance), natural frequencies, buckling loads, etc.
- Maximum displacement, stress, strain or strain energy density at the whole structure.
- Local responses of the structure such as displacement, stress, or strain at a particular location in the structure.

Geometrical constraints could be manufacturing limitations, maximum/minimum member sizes and physical restrictions. Note that although many of the above may be used as either objective or constraint, in engineering problems, it is usual to have a single objective with most performance measures implemented as constraints in the optimization problem.

### 2.1.1. Mathematical representation of a structural optimization problem

A structural optimization problem in general form can be defined as searching for the minimum (or maximum) value of a function  $f(x)$ , in which  $x = (x_1, x_2, x_3, \dots, x_n)$  is the variables vector. According to (Haftka and Gurdal, 1992), an optimization problem can be expressed as:

$$\begin{aligned}
 &\text{Minimize} && f(x) \\
 &\text{Subject to} && g_j(x) = 0 && j = 1:M \\
 &&& h_k(x) \geq 0 && k = 1:N
 \end{aligned} \tag{2.1}$$

where  $g(x)$  and  $h(x)$  are the equality and inequality constraints, respectively, with  $j$  the number of equality and  $k$  the number of inequality constraints. The

optimization problem is called linear if both objective function and the constraints are linear functions of the design variables. If either the objective function or one of the constraints is a nonlinear function of the design variables, the optimization is called nonlinear.

Such a structural optimization problem can be solved by using calculus methods (Haftka and Gurdal 1992) or using numerical methods such as mathematical programming (Heyman 1951), optimality criteria (Prager and Shield 1968; Prager and Taylor 1968) and genetic algorithms (Holland, 1975).

### **2.1.2. Classification of structural optimization problems**

Structural optimization problems can be classified into three main groups as follows:

- **Size optimization:** Size optimization involves finding the optimal dimensions in a design while its shape is fixed throughout the optimization process. This could include the cross sectional area, moment of inertia, thickness, length and breadth of the design (Aremu 2013). A common size optimization problem is truss structures in which the design variables are the cross sectional area of the trusses.
- **Shape optimization:** In this type of structural optimization, the shape of the boundaries change during the optimization process, however, the number of members in the structure and the manner they are connected remain fixed. In this approach the design variables are often a set of points on the geometry being optimized, and their Cartesian coordinates are iteratively updated until an optimum is reached.
- **Topology optimization:** This is the least constrained and most challenging type of structural optimization as the material distribution within the design domain including the shape, number and location of holes and connectivity of design domain may change throughout the optimization process. Figure

2.1 compares the three types of structural optimization for a typical structure.

Other types of structural optimization include topometry and topography optimization which are very similar to size and shape optimization, respectively. However, these are only applicable to 2D or shell structures where the concept of thickness can be defined (Leiva et al 2007).

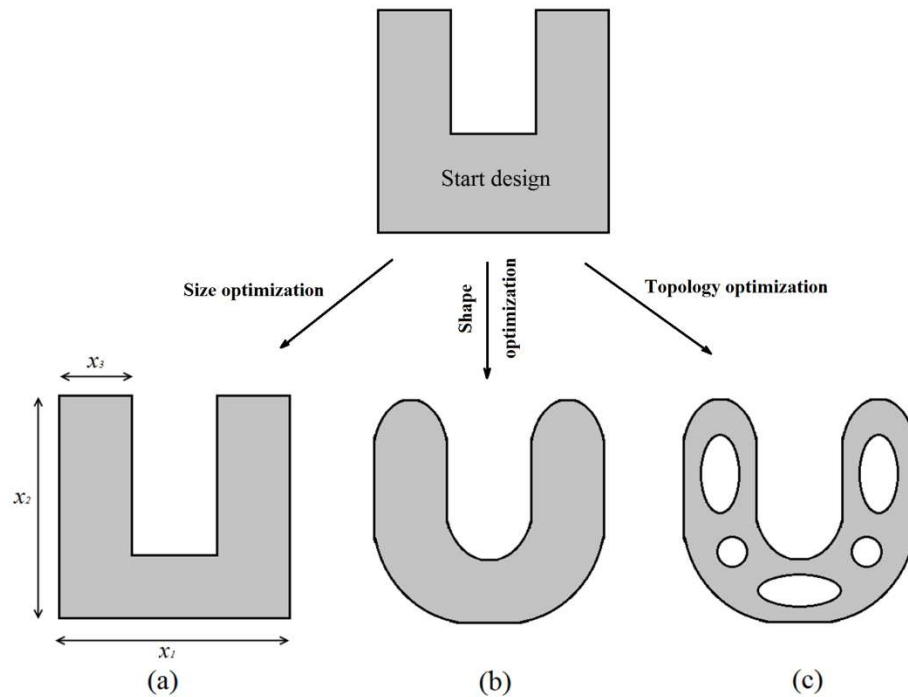


Figure 2.1. (a) Size (b) shape and (c) topology optimization of a typical structure

## 2.2. Topology optimization

### 2.2.1. History

Topology optimization is much younger than size or shape optimization; however it is a rapidly expanding field of structural mechanics. It can be traced back to the early 1900s when the important principles for topology optimization of structures with very low volume fractions (truss-like structures) were established by Michell (1904). Nearly 70 years later Rozvany (1972) extended Michell's approach to grillages (beam systems) paving the way for the optimal layout theory as the first general theory of topology optimization (Prager and Rozvany 1977; Rozvany 1992; Rozvany and Birker 1994). Although this theory was primarily applied to



analytical (exact) optimization of grid-type structures, this would later have significant implications for numerical methods and continuum structures requiring higher volume fractions (called Generalized Shape optimization (Rozvany et al 1992) or variable topology shape optimization (Haber et al 1996)).

The investigation of numerical techniques in topology optimization of continuum structures started in the late 80s with the landmark paper of Bendsoe and Kikuchi (1988), regarding the application of the homogenization method to topology optimization. In this finite element based topology optimization method, the idea was to represent the design domain with periodically distributed perforated microstructures in order to reduce the complex topology optimization problem into a simple size optimization of microstructures. The porous nature of the microstructures associated with the homogenization method's solutions represented with the microstructures made them difficult to be manufactured via conventional manufacturing methods (Sigmund, 2001). A simpler approach called Solid Isotropic Material (or Microstructure) with Penalization (SIMP) was developed by Bendsoe (1989) and Zhou and Rozvany (1991) based on assuming constant material properties within each element of the design domain.

Xie and Steven (1992; 1993; 1997) proposed a simple approach to topology optimization called the evolutionary structural optimization (ESO). This approach was based on an assumption that the structure evolves to an optimum by gradually removing inefficient material (elements with lower stress) from the design domain. An extension of ESO called Bi-directional Evolutionary Structural Optimization (BESO), allowing the elements to be added and removed simultaneously, was proposed by Querin et al (1998; 2000). Although ESO type optimization methods rely on intuition rather than a rigorously derived mathematical proof (Rozvany, 2009; Zhou and Rozvany, 2001), its inherent simplicity has made it an attractive approach for topology optimization. An alternative to the above gradient based approaches is the use of stochastic based methods in topology optimization. A genetic based algorithm for topology optimization was first proposed by Sandgren et al (1990) in order to prevent the topology optimization from converging into a local optima (as can be the case in gradient based algorithms). As the genetic algorithms operate on a population of potential solutions rather than improving one single solution, thus dealing with a much larger number of design variables, the

research in its implementation on structural optimization is limited to relatively small problems (Zuo et al 2008).

The conventional element based methods described above were not capable of accurately representing the shape of the boundaries. Due to the importance of shape representation in shape and topology optimization, a level set based approach for structural optimization was proposed (Sethian and Wiegmann 2000; Wang et al 2003; Allaire et al 2004). The idea was to implicitly represent the boundary using a level set model, and then combine the shape sensitivity with a Hamilton-Jacobi equation in order to track the shape and topology changes.

There are a number of other approaches proposed for the topology optimization of continuum structures which are not as established as those previously mentioned in this chapter. They include ant colony (Kaveh et al., 2008; Luh and Lin, 2009), simulated annealing (Bureerat and Limtragool, 2008), boundary element (Tait and Fenner, 1999), bubble method (Eschenauer et al., 1994) and particle swarm (Luh et al, 2011).

### **2.2.2. Classification of topology optimization methods**

Here, three different classifications are presented for structural optimization methods addressing the topology optimization of continuum structures:

- Mathematical based and heuristic methods

Some topology optimization approaches have a rigorous mathematical proof behind them such as the homogenization and level set methods. Another group called heuristic methods includes those methods which address the topology optimization in a less mathematical but more intuitive way by observation from nature. Although the heuristic methods have proven themselves in providing good solutions, there is no guarantee that a solution will always exist by implementing these methods. Examples of the heuristic methods are ESO type methods, and those which employ genetic based algorithms.

- Gradient based and stochastic based methods

In general, there are two possible ways of finding the optimum set of design variables in an optimization problem, one using a gradient based approach and one by generating random variables. The gradient based methods search for the optimum by using the derivatives of the objective function and the constraints. As these methods heavily rely on aspects of calculus and the starting design, they are sensitive to convergence to local optima. Also these methods assume that the problem is convex and that there is a solution for the optimization problem. However in some structural problems, the optimization problem is not convex as it might be discontinuous or disjointed (Huang and Arora 1997; Haftka and Gürdal 1992; Querin 1997). In this case the use of stochastic methods could be beneficial. The stochastic methods of topology optimization such as genetic algorithm and ant colony operate by distributing random variables and applying survival of the fittest. Due to the large number of design variables defined in these methods they are computationally expensive and not recommended for large optimization problems.

- Material distribution methods and boundary variation methods

Material distribution methods are those which solve the structural optimization problem by optimally distributing the material within the design domain and include the homogenization method, SIMP and ESO. The material distribution methods are mainly element based approaches as the optimization algorithms are integrated with the finite element framework of the structural problem and the elements or a property of the elements are considered as the design variables. The element based methods have been very successful in finding the structurally optimal topologies and they have been the dominant methods of topology optimization. However a drawback to these methods is their weak capability in representing the geometry of the design as it is represented by the finite elements.

In the boundary variation methods, the boundary of the design and the topology are not represented by elements, instead they use geometric methods to represent the boundaries. The boundary variation methods were initially incapable of changing the topologies and were only used for the purpose of shape optimization for a given topology (Haftka and Grandhi 1986). However recent improvements in boundary variation methods, such as the level set method, allow the topology to change during the optimization process (Burger et al 2004; Allaire et al 2005).

### **2.2.3. Common approaches for the topology optimization of continuum structures**

#### 2.2.3.1. Homogenization

The homogenization approach (Babuska, 1976, Cioranescu and Paulin 1979) in general is a mathematical theory which is used to find the effective material properties of the equivalent homogenized domain in a physical problem. For example it can be used to represent a composite material with a homogenous one having the same mechanical characteristics. This approach can be used in topology optimization as the structure to be optimized can be considered as a composite consisting of material and void. The first application of homogenization in topology optimization was by Bendsoe and Kikuchi (1988) in which they derived the effective material properties for porous finite elements. Assuming a rectangular shape for the holes (figure 2.2), the porous elements could be mathematically modelled using three geometrical parameters of the hole's size and orientation,  $a(x)$ ,  $b(x)$  and  $\theta(x)$ , allowing a varying density ( $0 \leq \rho \leq 1$ ) for the porous region (Suzuki and Kikuchi 1991). Following that, both the density and the elasticity tensor can be defined as a function of the geometrical parameters of the microstructures:

$$\begin{aligned}
 \rho(x) &= \rho(a(x), b(x), \theta(x)), \\
 E_{ijkl} &= \tilde{E}_{ijkl}(a(x), b(x), \theta(x)) \\
 Vol &= \int_{\Omega} \rho(x) dx
 \end{aligned} \tag{2.2}$$

where the effective material parameters  $\tilde{E}_{ijkl}$  can be obtained using the homogenization formulas. By implementing the above assumption, the complex topology optimization problem could be simplified into finding the optimal size and orientation of the holes. Implementing the rectangular microvoids in a quadrilateral mesh allows a full range of porous elements, from fully solid to completely void, however implementation of rectangular microvoids in triangular elements is more restrictive. Another drawback of this implementation is the high number of design variables, which is computationally expensive, especially for 3D optimization problems where there exist 3 size variables and 3 orientation variables. Other proposed microstructures include the use of ellipsoidal holes to remove the need for orientation angle (Suzuki and Kikuchi 1991), ranked laminates (Hassani and Hinton 1998), triangular microstructures (Folgado et al 1995) and hexagon microstructure (Hassani and Hinton 1998).

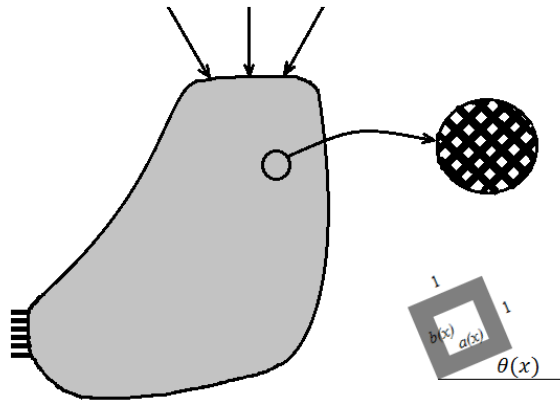


Figure 2.2. Rectangular microstructure

The essential steps in solving topology optimization problems using the homogenization method can be summarized as:

1. Discretizing the design domain with finite elements, each of which consists of a composite material having microvoids covering the density variations between 0 and 1.
2. Finding the optimal size and orientation of the microvoids using a search technique such as optimality criteria method.

### 3. Determining the homogenized material properties of the composite material.

The advantage of the homogenization method is that it has a rigorous theoretical basis which can provide a mathematical bound to the theoretical performance of the structures. Also, it has a good rate of convergence. The disadvantage of this method is that determination and evaluation of the optimal microstructures is cumbersome. Also, the solutions cannot be built directly since no definite length scale is associated with the microstructure (Sigmund 2001).

#### 2.2.3.2. SIMP

The term “SIMP” standing for Solid Isotropic Microstructure with Penalization, was first introduced by Rozvany et al (1992). However, the method was proposed by Bendsøe in 1989 under the term “direct approach”. The idea was to assume continuous elemental densities for the structure as opposed to the discrete form of the homogenization method, thus reducing the number of variables per element into 1. Implementing this assumption into topology optimization, the resulting solution can be represented not only with solid and void regions, but with intermediate densities. However, as these are not generally manufacturable, a power-law approach is used to move the intermediate elemental densities ( $0 < \rho < 1$ ) towards a 0/1 solution by penalizing the intermediate densities, as in equation 2.3:

$$E(\rho) = (\rho)^p \cdot E_0 \quad (2.3)$$

where  $p$  is the penalization power and  $E_0$  is the Young’s modulus of the solid material. The stiffness of the intermediate density elements can be decreased by increasing the penalization power resulting in the solutions moving closer to a discrete 0/1 solution. Although the resulting solution tends to be less optimized.

The topology optimization problem using SIMP method where the objective is to minimize the compliance (maximize the stiffness) subject to a final volume fraction can be shown by (Sigmund 2001):

$$\begin{aligned}
& \text{minimize: } C = U^T K U = \sum_{e=1}^n (x_e)^P u_e^T k_e u_e \\
& \text{subject to: } \frac{\sum_{e=1}^n v_e x_e}{V^0} = V^c \\
& \quad : KU = F \\
& \quad : 0 < x_{min} \leq x \leq 1
\end{aligned} \tag{2.4}$$

where  $U$ ,  $F$  and  $K$  denote the global displacement vector, global force vector and global stiffness matrix, respectively.  $u_e$  and  $k_e$  represent the element displacement vector and the element stiffness matrix, respectively, and  $x_e$  and  $v_e$  are the vectors of design variables and element volumes, respectively.  $n$  is the total number of elements in the design space, and  $V^0$  and  $V^c$  denote the volume of the design domain and the prescribed volume fraction, respectively.  $x_{min}$  is a minimum non-zero relative density usually used for the void elements to avoid singularity. The above optimization problem can be solved using various numerical search techniques, such as the Optimality Criteria Method (OCM) (Bendsoe 1995) and the Method of Moving Asymptotes (MMA) (Svanberg 1987). Figure 2.3 shows a flowchart of the optimization process in SIMP (Bendsoe and Sigmund 2003). The topology optimization starts by an initial guess of the density distribution within the design domain. Then a finite element analysis is performed on the current structure, followed by a sensitivity analysis. A sensitivity filter scheme (Sigmund 1994, 1997) is used to ensure existence of solutions and avoid numerical instabilities, such as checkerboard pattern (Sigmund and Petersson 1998). This finite element based process of sensitivity analysis and design update is then repeated until convergence is achieved.

Since SIMP doesn't need homogenization and it uses only one design variable per element, it uses less mathematics and is therefore easier to be understood than homogenization. Compared to the other types of microstructure, SIMP generates clearer solutions because of the power law approach it uses. Therefore it tends to be easier to implement in practice, hence its use in commercial topology optimization software such as Altair Optistruct and Abaqus-Dassault Systèmes. However the solution depends on the value of penalization power and the final layout of the structure is dependent on the initial density distribution and mesh size. Also there is a need to threshold at an arbitrary density to achieve a discrete solution which may reduce the optimality of the solution.

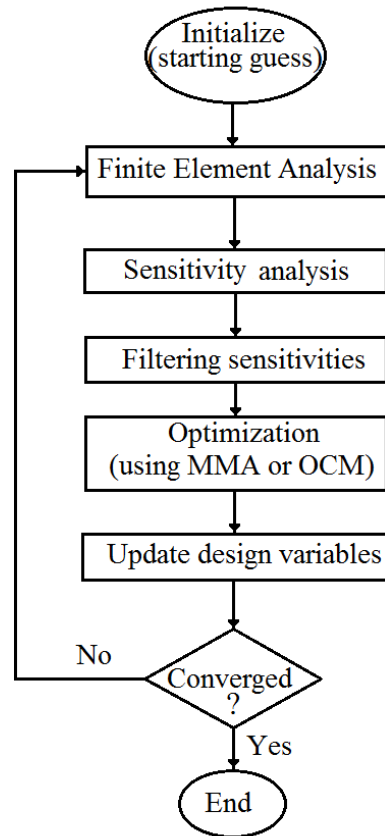


Figure 2.3. Flowchart of SIMP.

### 2.2.3.3. Evolutionary Structural Optimization

Evolutionary structural optimization (ESO) was first proposed by Xie and Steven (1992, 1993) based on the simple concept that the optimal design can be achieved by gradually removing inefficient material (elements) from the design space. This approach can be considered as a combination of heuristic methods and gradient-based approaches (Yang et al 1999). By assuming an elements stress value as its effectiveness criteria, Xie and Steven proposed that by systematically removing the low stress elements the optimal layout of the structure is obtainable. In order to maximize the stiffness of the structure, Chu et al. (1996) replaced the stress criterion with an elemental strain energy criterion. Bi-directional Evolutionary Structural Optimization (BESO) proposed by Querin et al (1998, 2000) is an extension of ESO in which new elements can be added to the structure near the high stress elements, during the evolutionary process of element removal. Yang et al (1999) implemented the BESO method for stiffness optimization problems with an element strain energy criterion by assuming the sensitivity of the void elements



to be obtained from linear extrapolation of the displacement fields. Huang and Xie (2010a) stated the optimization problem for ESO/ BESO methods as follows:

$$\begin{aligned}
\text{minimize: } C &= \frac{1}{2} U^T K U = \frac{1}{2} \sum_{e=1}^n u_e^T k_e u_e & (2.5) \\
\text{subject to: } & \frac{\sum_{e=1}^n v_e x_e}{V^0} = V^c \\
& : KU = F \\
& : x_e = x_{min} \text{ or } 1
\end{aligned}$$

where C is called the mean compliance rather than the compliance ( $U^T K U$ ) as defined in SIMP. The elemental sensitivities are defined as

$$S_e = \frac{\partial C}{\partial x_e} = \frac{1}{2} u_e^T k_e u_e. \quad (2.6)$$

It can be seen that in the above definition of the ESO/BESO approach, the low sensitivity elements are not completely removed from the design domain. Instead, they are assigned a very weak material property,  $x_{min}$ . This scheme is called the soft-kill approach (as opposed to the hard-kill approach in which the element is removed from the design domain). As the complete removal of solid elements during the evolutionary approach can cause theoretical difficulties in the optimization, Huang and Xie (2009) proposed a version of soft-kill BESO with a material interpolation scheme (as in SIMP) in which the hard-kill solutions were obtainable by making the penalty exponent of the element relative densities into infinity:

$$S_e = \frac{x_e^{p-1}}{2} u_e^T k_e u_e. \quad (2.7)$$

ESO/BESO employs a similar sensitivity filtering scheme as SIMP to avoid numerical instabilities in the optimization process (see section 2.3). Also in order to stabilize the evolutionary process, a sensitivity averaging scheme is proposed by Huang and Xie (2009) in which

$$S_j^{it} = \frac{S_j^{it} + S_j^{it-1}}{2} \quad (2.8)$$

where it is the current iteration and j denotes the node number. Once the elemental sensitivities and nodal sensitivity numbers are calculated, the variables can be

updated using the BESO element addition and removal scheme in which the elements having a sensitivity above a threshold sensitivity number are added to the structure (become solid elements) and those having a sensitivity number below a threshold value are removed (become void elements with minimum density). Until the volume constraint is satisfied, the volume of the next iteration can be calculated from

$$V_{it+1} = V_{it}(1 + ER) \quad (2.9)$$

where ER is the volume evolution rate. The flowchart of the BESO method as suggested by Huang and Xie (2010a) is shown in figure 2.4.

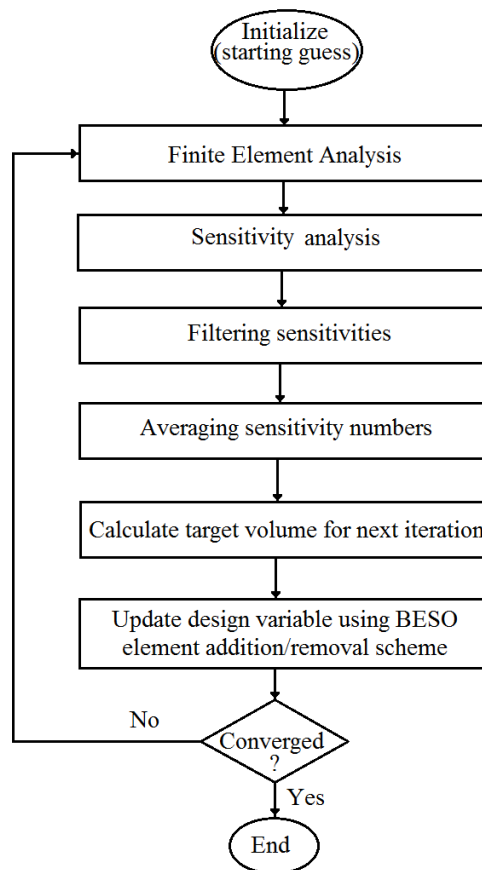


Figure 2.4. Flowchart of BESO.

The ESO/BESO algorithms have the advantage of achieving high quality solutions with a good computational efficiency. Also, the optimization algorithms are easy to

understand and implement (Huang and Xie 2010a). However there have been a few critical comments on these approaches from Zhou and Rozvany (2001) and Rozvany (2009). ESO is heuristic and there is no proof that an optimal solution can be achieved by element elimination and admission. The original ESO based methods are not efficient as one needs to find the best solution by comparing a number of intuitively generated solutions. The evolutionary methods cannot be easily extended to other constraints (displacement for instance) or multi-load/multi-constraint problems. Also, Rozvany (2001) pointed out a non-optimal solution for the optimization of a cantilever tie-beam when ESO was employed. The above comments were addressed by Huang and Xie (2010b). For the case of the cantilever tie-beam, Huang and Xie showed that the BESO solution was a local optima rather than a non-optimal solution, and that even SIMP can fall into the same local optima by changing the initial density distribution and using high values of penalization power.

#### 2.2.3.4. Genetic Algorithms

There have been more papers on the use of genetic algorithms (GAs) in topology optimization than any other stochastic based approach. The idea of a genetic algorithm was first presented by Holland (1975) to mimic biological natural selection. As mentioned earlier, like other stochastic based approaches, genetic algorithms operate by evolving a population of potential solutions toward better solutions, rather than improving a particular solution (as in gradient based approaches). GAs uses an evolutionary survival-of-the-fittest mechanism (Holland 1975; Goldberg 1989) allowing the designs in a population to compete with each other to become parent designs. Then the parent designs create the child generation by swapping portions of their genetic code. After a limited number of random mutations, the child generation which is hoped to be of a higher quality, replaces the parent generation, and the evolutionary process is repeated until it reaches an optimal design (Chapman and Jakiela 1996, Jakiela et al 1999).

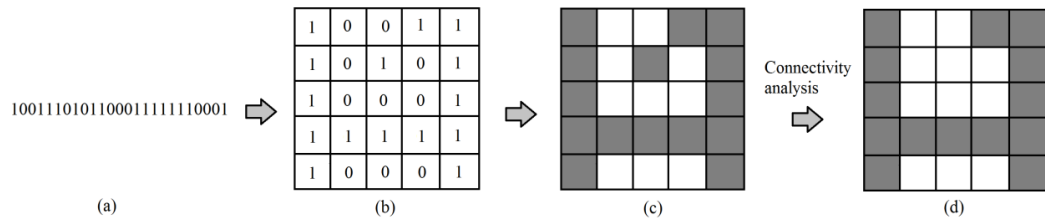


Figure 2.5 (a) A chromosome (b) digital form of chromosome in mesh (c) the equivalent topology of the chromosome (d) the topology after connectivity analysis.

Parent chromosomes

```

100111010110001|1111110001
1111101000010100|100111111
                    Random crossover point

```

Resulting child chromosomes

```

1001110101100010100111111
1111101000010101111110001

```

Figure 2.6. Genetic crossover of chromosomes.

In GAs, the design variables (potential solutions) are expressed with a coded representation, usually a character string, where each character can be thought as an allele positioned in a gene of a chromosome. The new designs are generated by undergoing a genetic crossover which allows the child designs to have traits from both parents. A merit function is used to evaluate the performance of the new designs, giving a higher chance of creating offspring to the designs with higher merit, and surviving into the next generation. Figure 2.5 shows a chromosome and its equivalent 2D topology. Before calculating the value of merit function, a connectivity analysis resulting in voiding all non-connected material elements, might be required to ensure the valid transference of the boundary conditions to all elements of the domain (Kane and Schoenauer, 1996; Wang and Tai, 2005) (figure 2.5d). The performance of the chromosome (topology) can be measured by performing a finite element analysis. Among different types of crossovers, single point crossover is the most basic one in which an allele is randomly selected and the segments of the code after that are swapped (figure 2.6). Other types of crossovers include two point crossover and diagonal crossover (Kane and Schoenauer 1996).

As being a global search technique, there is less chance of falling into local optima for GAs than the gradient based methods. However because of the large number of

design variables and function evaluations, the optimization approach is computationally expensive and is currently limited to systems with a small number of variables. Hence in the structural optimization of continuum structures, a coarse mesh should be employed to avoid excessive computational cost. This can significantly restrict the quality of the solutions. It has been recommended however that the GAs could be of advantage when applying them to problems with little knowledge about the nature of the design domain (Jakiela et al 1999).

#### 2.2.3.5. Level Set Method

The level set method was first introduced by Osher and Sethian (1988) to track moving interfaces. Since then, it has been developed and applied to many physical problems. The application of the level set method to structural and topological optimization problems developed in the last two decades (Sethian and Wiegmann 2000; Osher and Santosa 2001; Wang et al. 2003; Allaire et al. 2004). In the level set method, the boundary is represented implicitly (as opposed to the explicit boundary description in splines), using the contours of a level set function e.g. a signed distance function. This implicit boundary description conveniently allows for the treatment of the topological changes while smoothly representing the design boundaries.

The level set method employs a two-phase material-void model to represent the design domain (figure 2.6). The material phase  $D_S$ , void phase  $D_V$  and the boundary  $\partial D_S$  are defined according to the values of a level set function  $\varphi(x)$  in those regions as:

$$\varphi(x) : \begin{cases} > 0 & \Leftrightarrow & x \in D_S \\ = 0 & \Leftrightarrow & x \in \partial D_S \\ < 0 & \Leftrightarrow & x \in D_V \end{cases} \quad (2.10)$$

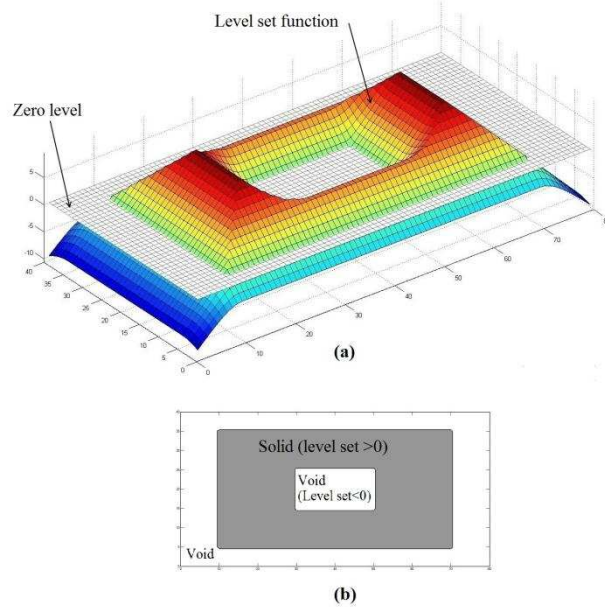


Figure 2.6 (a) Level set function (b) material and void phases.

where  $x$  is a point in space (van Dijk et al 2013). The change of the level set function  $\varphi(x)$  is governed using a Hamilton-Jacobi equation as

$$\frac{\partial \varphi(x)}{\partial t} + v_n(x)|\varphi(x)| = 0 \quad (2.11)$$

where  $v_n(x)$  is the normal velocity which is obtained from the sensitivity of the objective function with respect to the boundary variation. Therefore, solving equation 2.11 allows updating of the level set function by moving the boundary along the normal direction. The level set function is typically initialized as a signed distance function. To address the problem of overshooting, Osher and Sethian (1988) developed an upwind finite difference scheme in order to numerically solve the Hamilton-Jacobi equation. To satisfy the CFL condition, the time step,  $\Delta t$ , needs to be chosen from

$$\Delta t \leq \frac{h}{\max(v_n)} \quad (2.12)$$

where  $h$  is the minimum grid size in the spatial discretization (Sethian 1999, Challis 2010). By evolving the boundary, the level set function does not necessarily remain a signed distance function. Hence, from time to time, the level set function is re-initialized to a signed distance function to insure accuracy in solving the Hamilton-Jacobi equation (Sethian 1999, Challis 2010).

One of the drawbacks of the standard evolution equation of the level set method, equation (2.11), when used for 2D structural optimization problems is that it cannot introduce new void regions into the material region (Allaire et al. 2004). So, one may need to start the optimization problem with an initial design domain having as many as possible holes, in order to account for all possible topologies. Some other techniques such as incorporating the topological derivatives have been proposed to allow topological changes during the shape optimization (Burger et al 2004; Allair et al, 2005). Jia et al (2011) proposed an evolutionary level set method by combining ESO with the level set method algorithm, allowing new holes to be generated in low strain energy regions during the optimization process. Operating on a fixed-grid FE mesh, the level set based topology optimization methods have the advantage of generating solutions with smooth boundaries, which eliminates numerical instabilities such as checkerboard. Also it has shown to benefit from a great numerical efficiency (Allair et al, 2005). However due to the mathematical based nature of this method, its implementation to different types of structural optimization problem is not as simple as the material based approaches.

#### **2.2.4. Numerical instabilities**

Numerical instabilities associated with the topology optimization of continuum structures can be divided into three categories (Sigmund and Petersson 1998): checkerboard, mesh-dependence and local optima.

##### **2.2.4.1. Checkerboard**

A common problem in element-based topology optimization approaches such as homogenization, SIMP and ESO/BESO, is the checkerboard pattern i.e. the formation of alternating solid and void elements in the topology in a manner which looks like a checkerboard (figure 2.7 b). Comparing the stiffness of checkerboard configuration to the uniformly distributed one, Dian and Sigmund (1995) found that the structures with a checkerboard pattern had artificially high stiffness values. Various approaches have been suggested to prevent this problem, including

smoothing (as a post-processing stage), the use of higher order finite elements (Diaz and Sigmund 1995; Jog and Haber 1996) and filtering sensitivities (Sigmund 1994).

The sensitivity filtering scheme was first proposed by Sigmund (1994), based on image processing filtering techniques. The idea was to estimate the design sensitivity of a specific element from the weighted average of the element itself and the neighbouring elements. An extension of this scheme was also proposed to prevent both checkerboard and mesh dependence. In this, the sensitivity of an element was being modified by weighted averaging of the sensitivities of the elements in a fixed neighbourhood of  $r_{min}$ . Figure 2.7 shows converged solutions of the topology optimization of a cantilever plate using a soft-kill BESO method with/without implementing sensitivity filter scheme.

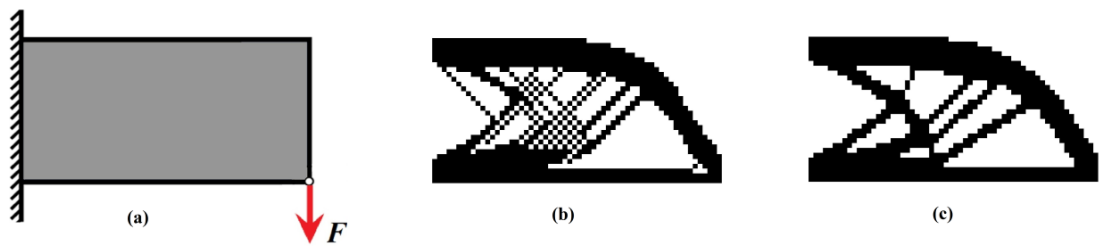


Figure 2.7. (a) Design domain and its boundary conditions (b) solution with checkerboard pattern (using BESO without filter scheme) (c) BESO solution implementing sensitivity filter scheme.

#### 2.2.4.2. Mesh dependence

Mesh dependence refers to the problem of achieving different solutions for an optimization problem by employing a different mesh. Ideally, by using a finer mesh one would expect to obtain the same optimal structure with an improved boundary representation. However implementing a finer mesh to a topology optimization problem can increase the complexity of the solutions i.e. convergence to solutions with more members of smaller sizes (Aremu et al 2013). Some schemes proposed to overcome mesh dependence in topology optimization include perimeter control (Harber et al 1996), local gradient constraint (Pettersson and Sigmund 1998) and sensitivity filter scheme. Figures 2.8b and 2.8c show two different topologies obtained by implementing different mesh and filter radius for



topology optimization problem of the cantilever plate shown in figure 2.8a. It can be seen that the solution obtained from implementing a fine mesh and filter radius of 1 mm (figure 2.8c) has an increased the complexity compared to the solution obtained from implementing a coarse mesh and filter radius of 2mm (figure 2.8b). However, by increasing the filter radius to 2mm, the complexity is reduced (figure 2.8d) and the same topology as that obtained from the coarse mesh is achieved. Aremu et al (2013) found that the higher complex topology optimised solutions tend to have higher structural performance. Therefore, one could argue that these coarsening techniques to overcome mesh dependence can only be of benefit when using traditional manufacturing process where the manufacturing cost is related to the design complexity. However with additive manufacturing offering design freedom, high performance solutions with more complexity is preferred.

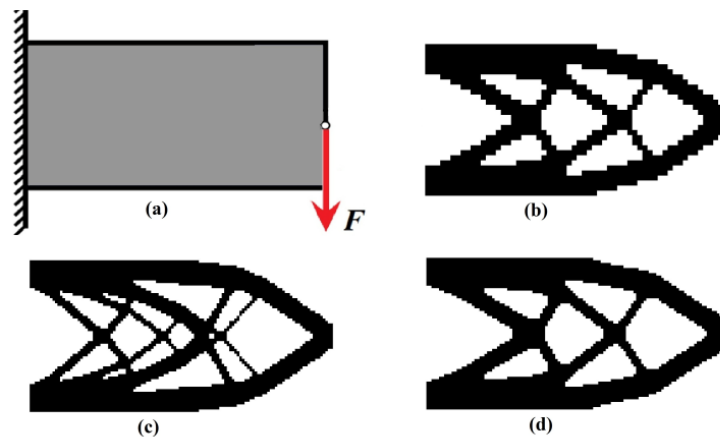


Figure 2.8. (a) Design domain and its boundary conditions (b) BESO solution implementing a 60x30 mesh and  $r_{min} = 2 \text{ mm}$  (c) BESO solution implementing a 120x60 mesh and  $r_{min} = 1 \text{ mm}$  (d) solution implementing a 120x60 mesh and  $r_{min} = 2 \text{ mm}$ .

#### 2.2.4.3. Local minima

As shown in figures 2.7 and 2.8, one may obtain different solutions for the topology optimization of a particular problem, depending on the optimization approach and the initial parameters used for the optimization problem (Aremu et al 2013). Therefore, it can be concluded that there exist many local optima for the topology optimization problem of a continuum structure. Sigmund and Petersson (1998) found that independent of the approach one uses for topology optimization,

single optimization formulations that produce a 0/1 design are nonconvex and subject to converge into a local optima. As the size of a typical topology optimization problem is too large to be handled by global optimization methods, Sigmund and Petersson (1998) suggested the use of continuation methods which also take global information into account.

### **2.3. Summary and conclusions**

In this chapter, structural optimization as a general technique for designing continuum structures was reviewed and the mathematical description of the optimization problem was presented. Topology optimization, as the most challenging topic of structural optimization, was reviewed followed by an overview of the main topology optimization methods that can be found in the literature. The main drawbacks as well as strengths of the methods were mentioned in the text. From this review, it can be concluded that by utilizing different topology optimization approaches for a problem, different solutions can be achieved. In terms of boundary representation, the solutions might be represented with 0/1 density finite elements (BESO solutions for instance), continuous density finite elements (SIMP for instance) or the boundary could be explicitly (as in splines) or implicitly (as in level set functions) defined. Most of the current topology optimization methods suffer from numerical instabilities associated especially with integrating a finite element analysis with the optimization algorithm. It was shown that the converged topology could change by utilizing different parameters for the numerical model of the structure. For example by using a different mesh size, a different solution might be achieved. This fact shows that the numerical simulation technique which is used and integrated with the optimization algorithm can have a significant effect on the quality of the converged solution.

The topology optimization methods presented in this chapter were based on utilizing the classical finite element method as the numerical analysis technique. However, in the last few decades there have been improvements and extensions to the classical FEM, as well as alternative meshfree methods which could be of benefit if combined with structural optimization algorithms. For this reason, the

second part of the literature review which is presented in chapter 3, aims to review different numerical structural analysis techniques appropriate for structural optimization.

# **Chapter 3**

## **Literature Review Part II:**

### **Numerical Methods for Structural Analysis**

Engineering problems are, in general, mathematical models that are used to represent a physical situation. These mathematical models consist of differential equations which come with a set of corresponding initial and boundary conditions. Sometimes for a simple system, it's possible to find the detailed behaviour of the system through an analytical approach. However in the case of practical engineering problems, we usually deal with large and complex mathematical models in which an exact solution cannot be easily obtained. In order to deal with these problems, numerical approaches have been developed. In topology optimization, as a modern structural optimization approach, one needs to tackle the increasing geometrical complexity of the design solutions. The application of numerical techniques in the topology optimization of engineering structures is almost unavoidable. Hence, most topology optimization algorithms are integrated with a numerical structural analysis technique. The finite element method (FEM) is a powerful tool, and indeed most popular numerical method, for solid mechanics design and optimization problems. Many topology optimization algorithms are either directly integrated with FEM (SIMP and ESO/BESO for instance) or use FEM for structural analysis. Despite all the advantages of the use of FEM in structural optimization, it comes with some shortcomings in boundary representation and also in the stability of the optimization problem (see section 2.2.4). Considering the importance of numerical structural analysis methods in

structural optimization, this chapter provides a review of the most important numerical methods appropriate to structural design and optimization.

### **3.1. Mesh-based and meshfree methods**

Numerical structural analysis methods in general can be divided into two categories: mesh-based methods and meshfree (meshless) methods. Mesh-based methods operate on meshes of data points where each point has a fixed number of predefined neighbours (figure 3.1 a). The connectivity of the neighbouring points is used to define the mathematical operators, such as derivative. The mesh-based methods include finite difference method (FDM), finite volume method (FVM) and finite element method (FEM). FVM and FDM have been extensively used in fluid mechanics and heat and mass transfer problems (Eymard et al 2000). FEM has been applied to a wide range of solid mechanics and multiphysics problems.

Meshfree methods don't require an explicit mesh to discretize the specified domain of the problem. In the meshfree and particle methods, the domain of interest is discretized into a set of nodes each having a domain of influence. The overlap of the domains of influence defines the connectivity of the domain of interest (figure 3.1b). Meshfree methods include Smoothed Particle Hydrodynamics (SPH), Diffuse Element Method (DEM), Element Free Galerkin Method (EFGM), Reproducing Kernel Particle Method (RKPM). Meshfree methods have been used to solve a wide range of engineering problems including fluid and solid mechanics, and also large scale problems in astrophysics. However, in solid mechanics applications, EFGM (Belytschko et al 1994) and RKPM (Liu et al 1995) have been more popular. Compared to FEM, meshfree methods have the following advantages (Li and Liu 2002):

- Easier handling of large deformation problems, since the connectivity of nodes can change with time.
- Suitable for problems with complex geometries where generating a mesh is difficult.
- Accuracy can be controlled more easily as one can simply add more nodes where more accuracy is desired.

- Meshfree discretization can provide more accurate representation of the geometry.

However the drawbacks are that meshfree methods are in general slower than FEM and the implementation of boundary conditions are cumbersome (Fernández-Méndez & Huerta, 2004; Chen et al 2007). The methods are also much less refined and developed compared with the long history of FEM development and commercialization.

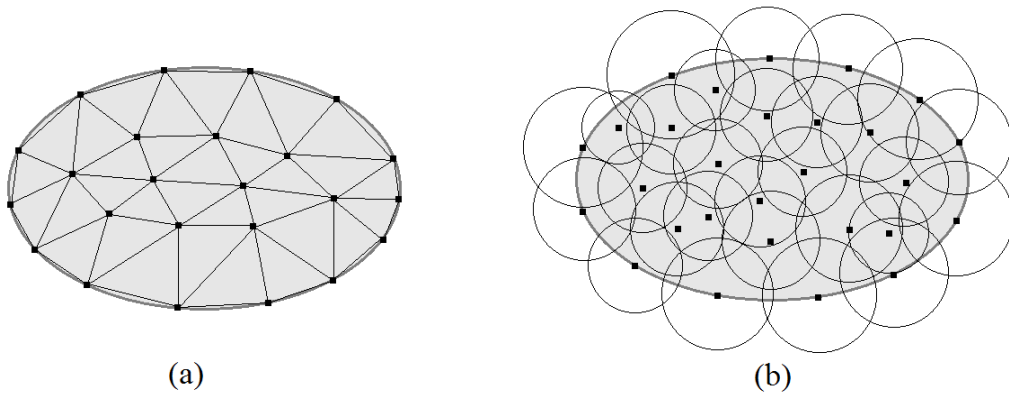


Figure 3.1. (a) Mesh-based approximation and (b) meshfree approximation

## 3.2. Finite Element Method

### 3.2.1. History

It is rather difficult to quote an exact date of invention of the finite element method. In 1909 Ritz proposed an effective approach for solving problems in the mechanics of deformable solids (Ritz 1909). In this method, the energy functional was approximated using known functions with unknown coefficients. Then, a system of equations was obtained by minimizing the functional with respect to each unknown, and the unknown coefficients could be obtained by solving the system of equations. However, in Ritz method, one should only use functions that satisfy the boundary conditions of the problem. Many years later in 1943, in an attempt to increase the possibilities of the Ritz method, Courant introduced special linear functions defined over triangular regions and used the method to solve torsion problems (Courant 1943). In this method, the values of functions at the node points of triangular regions were chosen as unknowns, allowing elimination of the main restriction of the Ritz functions i.e. satisfying the boundary conditions.

The term “finite element method” was first introduced by Clough (1960) in a paper titled “The finite element method in plane stress analysis”. However the method is very similar to the Ritz method together with the Courant modifications. The application of the FEM developed rapidly due largely to the ability of computers to carry out the expensive computations required by FEM. The first book on FEM was published in 1967 by Zienkiewicz and Cheung and is titled “The finite element method in structural and continuum mechanics” (Zienkiewicz and Cheung 1967). The book presents a general interpretation of the finite element method as well as the applicability of FEM to any general field problem. Although the method has been extensively used for solving problems of structural mechanics, it is now extended to use for other types of engineering problem, including heat conduction, fluid dynamics and electric and magnetic fields.

### 3.2.2. Definition of FEM

The finite element method (FEM) is a numerical approach for solving problems which are mathematically represented by partial differential equations that can be formulated as minimizing a functional. In this method, an assembly of finite elements is used to represent the specified domain of interest. FEM defines the approximating functions (shape functions) in terms of nodal values of the physical field which is being studied. The idea is to transform a continuous physical problem into a discretized FE problem with unknown nodal values. In the case of linear problems, a system of linear algebraic equations should be used to represent the physical problem. Once the nodal values are achieved, the values of the physical field inside finite elements can be obtained using shape functions (Nikishkov 2004). The main steps in FEM can be summarized as figure 3.2.

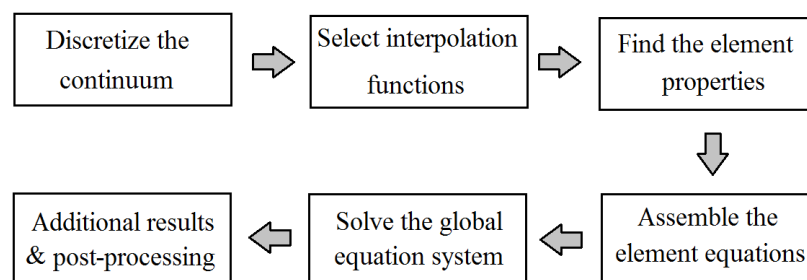


Figure 3.2. Main steps of finite element procedure.

### 3.2.3. Common linear elements for structural optimization

Generally, the element type should be chosen based on the dimension of the problem and the geometry of the domain of interest (Rao, 2011). If the geometry, material properties as well as response, can be expressed with a single spatial coordinate, one-dimensional elements can be used to discretize the domain. Temperature distribution in a rod and displacements in an axially loaded bar are examples of such a problem. Dependent on the design domain and load cases, structural optimization problems should be modelled using two or three dimensional elements with two and three spatial coordinates, respectively. The linear triangular element, consist of three nodes connected with three edges (figure 3.3 a), is the simplest 2D element (Bathe 2006). The linear shape functions used in the linear triangular elements define a constant value of stress or strain throughout the element. One may need to use a greater number of triangular elements in the problems which involve sharp variations in strain/stress across the structure, resulting in a significant increase in the computational cost. The second type of linear elements, which is defined by four nodes, is called the quadrilateral element (figure 3.3 b). More accurate results can be obtained by using this element as stress can vary in a limited way across the element, so one may need to use fewer elements for a problem. If the problem cannot be modelled with 2D elements, three dimensional elements need to be employed. Tetrahedral and hexahedral elements (figures 3.3 c & d) are 3D version of triangular and quadrilateral elements, respectively. Again, because of better accuracy, hexahedral elements are preferred to tetrahedral elements. However, as certain complex geometries are difficult to mesh with hexahedral elements, tetrahedral elements are highly useful. Higher order elements with curved edges can be obtained by inserting mid nodes on the edges of the 2D and 3D linear elements making them useful for modelling the domains with curved boundaries. Higher order elements have higher degree of accuracy than the linear elements, however by using the higher order elements the computational time will be increased (Rao, 2011).

In the isoparametric formulation of finite elements, the same set of shape functions are used to represent both the element geometry and displacement interpolations. The shape functions are defined by natural coordinates such as triangle coordinates for triangles and square coordinates for any quadrilateral. The advantages of



isoparametric elements are the ability to map more complex shapes and also having compatible geometries (Rao, 2011).

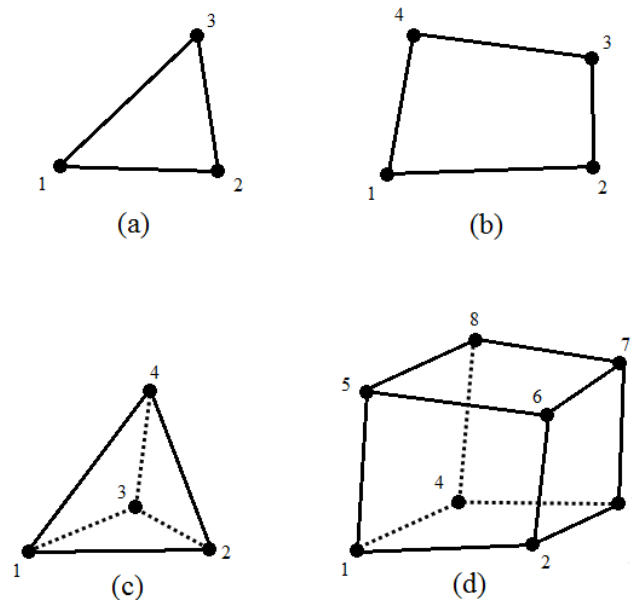


Figure 3.3. Common 2D and 3D linear elements: (a) triangular element (b) quadrilateral element (c) tetrahedral element (d) hexahedral element.

### 3.2.4. Application of FEM in topology optimization approaches

As it has been studied in the previous chapter, there are many topology optimization approaches that are integrated with the standard finite element method. They include homogenization, SIMP, standard ESO/BESO, genetic algorithm and many other approaches which were initially developed within a finite element framework. As mentioned previously, the common drawback of these FE based structural optimization methods are the weak capability of boundary representation. To improve the performance of FEA and provide a more accurate representation of the topology, a number of techniques can be implemented to the structural optimization, including adaptive mesh refinement and fixed-grid FEM, as discussed in the following sections.

### 3.3. Adaptive mesh refinement techniques

#### 3.3.1. Review of mesh refinement techniques

Adaptive mesh improvement is an important technique for increasing the accuracy of FEA by editing the FE meshes. A broad range of adaptive mesh refinement techniques have been established, which can be classified into three categories: h-refinement, p-refinement and r-refinement (Kuo et al., 2006; Aremu 2013). h-refinement enriches a mesh by subdividing elements in regions of interest and coarsens the mesh in other regions (Zienkiewicz et al., 2005). Figure 3.4a shows an arbitrary domain meshed with 8 triangular elements. Assuming the regions inside the broken line as the domain of interest, figure 3.4b shows the mesh obtained after mesh refinement of the domain of interest.

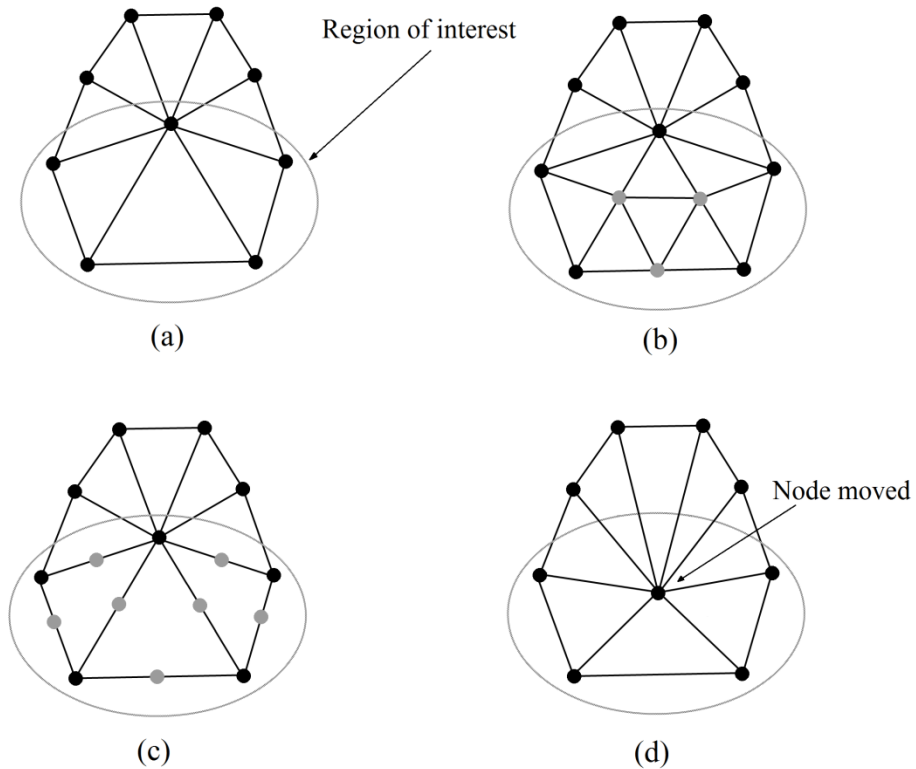


Figure 3.4. Illustrating mesh refinement for a triangular mesh (a) domain initially meshed with 8 triangular elements (b) h-refinement (c) p-refinement (d) r-refinement.

In the second type of mesh refinement, p-refinement, new nodes are introduced in the boundary of the elements resulting to an increase in the order of polynomial constituting the shape function of the element (figure 3.4c). In the third technique, r-refinement, the mesh connectivity is preserved (as opposed to the h-refinement)

and the nodes are moved to a location that optimizes the quality of mesh in their vicinity (figure 3.4d). A common technique for optimizing the mesh quality is the use of Laplacian smoothing (Cannan et al, 1993; Freitag, 1997) in which the nodes are iteratively moved to the mean location of its adjacent vertices.

### **3.3.2. Application of mesh refinement techniques in topology optimization**

There have been a number of papers describing various topology optimization approaches coupled with adaptive mesh refinement techniques. Wang (2007) and Verani et al. (2010) used the SIMP method with adaptive meshing strategies. Kim and Weck (2005) proposed an adaptive topology optimization method using genetic algorithms with variable chromosome length. Guest et al. (2009) also implemented an adaptive mesh refinement scheme with genetic algorithm for topology optimization of two and three dimensional problems. A Heaviside projection scheme was used to isolate design variables from the FE mesh. Ramm et al. (1998) proposed an adaptive density-based topology optimization method for optimizing structures with elastoplastic material. Although these works demonstrated some potential, the computational time and solution complexity were not described in great detail. Also most of these methods couple the adaptive mesh refinement to a density based topology optimization algorithm (Ramm et al. (1998), Guest et al. (2009) and Wang (2007)) which require utilization of penalization schemes to suppress intermediate densities. Aremu (2013) proposed an adaptive BESO method for topology optimization of two and three dimensional structures appropriate to designs for additive manufacturing (AM). By implementing adaptive BESO, Aremu achieved solutions with similar complexity of those obtained using a fixed mesh BESO utilizing a fine mesh, but with fewer degrees of freedom. However, although this was shown to be computationally efficient in 2D analysis problems, with loss of accuracy due to mesh distortion compromised its efficiency for 3D problems.

Generally speaking, by coupling an adaptive mesh refinement technique with a topology optimization algorithm the topology can be represented with more detail however the computational efficiency will be sacrificed when the method is

applied to 3D problems. Another drawback of the remeshing approaches is the projection errors which come from transferring the solution from the old mesh to the new mesh (Lee and bathe 1994).

### 3.4. Fixed Grid FEA

#### 3.4.1. Review of fixed grid FEA

Fixed grid FEA (FG-FEA) is a technique for modelling problems in which the geometry of the object or physical properties of the body change with time without the need for remeshing (Garcia-Ruiz an Steven 1999). Examples of such problems include absorption of liquid from a porous material and phase change problems, as well as structural optimization problems. The idea of FG-FEA is to use a fixed mesh for the analysis of the problem and superimpose the geometry (or the physical property which is changing with time) on the fixed mesh. Therefore the degrees of freedom remain the same and the boundary evolution does not have a significant effect on the computational cost. Figure 3.5 shows a material-void interface problem represented in a fixed grid mesh. It can be seen that by implementing the fixed grid scheme, the elements can be classified into three groups: solid elements, void elements and boundary elements (the elements that lie on the boundary). In the analysis, the solid elements are assigned with full solid material property and the void elements are assigned with a very weak material property (rather than being completely removed). A conventional approach for treating with the boundary elements is to use a density based scheme in which the property (stiffness for instance) of the element is proportional to the area ratio of the solid part of the element (Kim et al 2000):

$$\xi^{(e)} = \frac{A_s^{(e)}}{A_s^{(e)} + A_v^{(e)}} = \frac{A_s^{(e)}}{A_t^{(e)}} \quad (3.1)$$

$$K_B = K_S \xi^{(e)} \quad (3.2)$$

where  $A_s^{(e)}$ ,  $A_v^{(e)}$  and  $A_t^{(e)}$  represent the solid area, void area and the total area of the element, respectively, and  $K_B$  and  $K_S$  are the stiffness matrices of the boundary element and solid element, respectively.

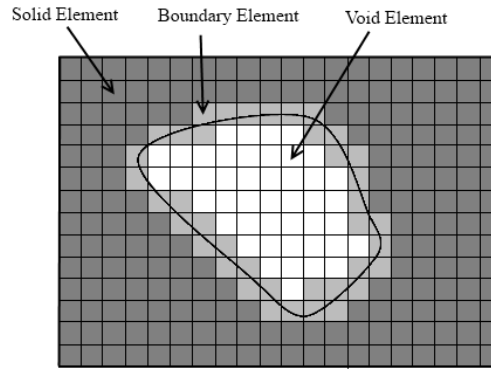


Figure 3.5. A material/void interface problem in a fixed grid mesh

In the density scheme, the material is considered to be uniformly distributed through the whole element and the variations in material distribution in an element are not taken into account in calculating the element stiffness matrix. For example, figure 3.6 shows three different shapes for a boundary element where the area fraction of solid material within the element is 0.50. Using the density method (equation 3.2) the same stiffness will be obtained for all three elements. However, one would expect these elements to have different contributions in the finite element framework. This issue can cause errors near the boundary (Wie et al 2010).

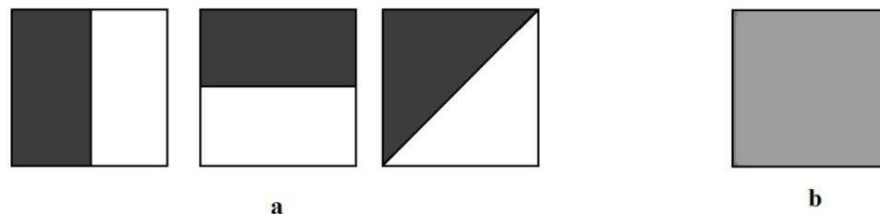


Figure 3.6. (a) Typical boundary elements for area ratio=0.50. (b) Their density scheme equivalent solid element with 50% density.

### 3.4.2. Application of fixed grid FEA in topology optimization

There are many structural optimization algorithms that operate within a Fixed-grid FE framework, including the level set and isoline/isosurface based structural optimization methods. The density based approach for approximating the element properties has been widely used in the level set method (Wang et al 2003; Allaire

et al 2004). Fixed-grid FEA was introduced to ESO by Kim et al (2000). They found the combination of ESO and fixed grid FEA an effective approach which can reduce the computational cost of the optimization. Victoria et al (2009 and 2010) proposed an evolutionary optimization algorithm operating in a fixed grid mesh in which the design boundaries were represented using contours of von-Mises stress. Wie et al (2010) compared the solutions of a density-based level set method and an XFEM-based level set method and found that those obtained from the density-based level set method had coarser boundaries, where the FE results near the boundary were not accurate. By applying topology optimization in a fixed grid FEA, one can improve the computational efficiency as well as the boundary representation of the design. However the density scheme used to calculate element properties may not have adequate accuracy (Kim et al 2000; Wie et al 2010).

### **3.5. Element Free Galerkin Method**

#### **3.5.1. Review of element free Galerkin method**

To overcome the shortcoming of FEM in modelling problems of large deformation, phase change or crack propagation, a number of meshless methods have been developed. The Element Free Galerkin Method (EFGM) (Belytschko et al 1994) is the most popular meshless method for structural optimization. Like many other meshless methods, only a set of nodes and description of the boundaries are required to generate the discrete equations. Each node has a domain of influence which is the subdomain over which it contributes to the approximation. The connectivity between nodes are defined by the overlap of the nodal domains of influence. The shape function is constructed by employing a Moving Least Square (MLS) approximation (Lancaster and Salkauskas 1981) which is a numerical differentiation method for irregularly spaced calculation points (Gossler 2001). Three essential components are required to approximate the function  $u(x)$  with  $u^h(x)$  using MLS approximation (Dolbow and Belytschko 1998): a weight function associated to each node, a basis which is usually a polynomial, for instance a quadratic basis as  $u^h(x) = a_0(x) + a_1(x)x + a_2(x)x^2$ , and a set of coefficients  $a_j(x)$ , which depend on the position.

There is a small subdomain around each node where the weight function is nonzero, called the support. The domain of influence is defined by the support of the weight function. The first application of MLS interpolants in conjunction with a Galerkin method was in the problems of heat conduction (Nayroles et al 1992). Belytschko et al (1994) applied the method to solve problems of elasticity and crack propagation and called it the Element Free Galerkin Method. EFGM is considered as a meshless approach when referring to the shape function construction. However this method still needs a background mesh for solving the partial differential equations by the Galerkin approach (Dolbow and Belytschko 1998) (figure 3.7). The mesh is defined to compute the integral in the weak form using gauss points defined in each cell. A major disadvantage of EFGM is that imposing the essential boundary conditions is cumbersome as the MLS approximation doesn't satisfy the Kronecker delta criterion. Several techniques have been developed to allow the imposition of boundary conditions including the use of Lagrange Multipliers (Belytschko et al 1994), coupling with FEM (Cingoski et al 2000) and the modified variation principle (Lu et al 1994).

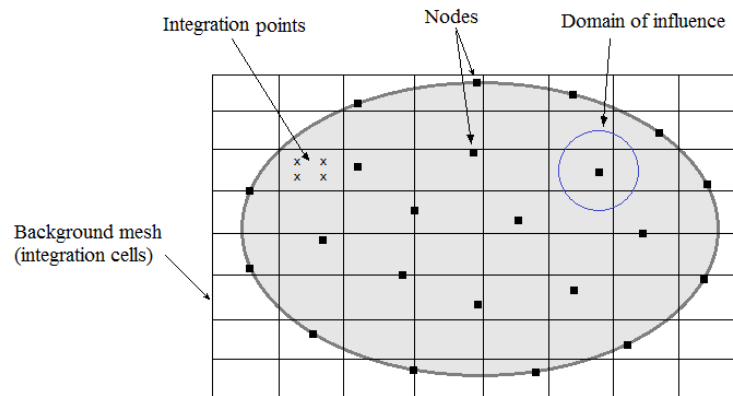


Figure 3.7. Illustrating numerical model of element free galerkin method.

### 3.5.2. Application of Meshless Methods to Topology Optimization

In recent years meshless methods have gained a great popularity in a broad range of areas including structural optimization. The reason for this is that they exhibit a very good numerical stability as well as accuracy (Luo et al 2013). Cho and kwak (2006) and Zhou and Zou (2008) used the Reproducing Kernel Particle Method

(RKPM) for numerical analysis in topology optimization problems. Luo et al (2013) employed SIMP model in a meshless EFGM framework. Luo et al (2012) used the level set method in conjunction with EFGM for shape and topology optimization problems. Juan et al (2010) proposed a meshless topology optimization approach by coupling ESO with EFGM. Du et al (2008) used a SIMP model in an EFGM framework for solving geometrically nonlinear compliant mechanisms.

Most of the research done in the area of application of meshless method to topology optimization of continuum structures is limited to conceptual works proving the validity and feasibility of the methods. Although it has been proved that meshless methods have a good numerical stability in solving benchmark problems, and can provide a more detailed description of the design boundary (figure 3.8), the comparison of computational efficiency of the mesh-based and meshless-based optimization methods has not been properly addressed to date. Due to the difficulties in imposing essential boundary conditions, the application of meshless-based topology optimization methods to real-life problems is challenging.

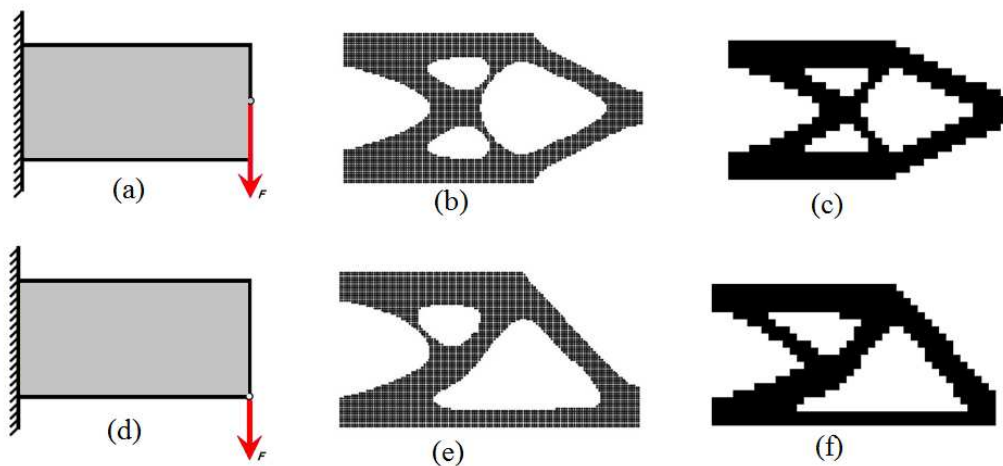


Figure 3.8. BESO topology designs for cantilever problems (a) cantilever loaded on the middle end (b) its meshless BESO solution and (c) its mesh-based BESO solution. (d) Cantilever loaded at the bottom end (e) its meshless BESO solution and (f) its mesh-based BESO solution.



### 3.6. X-FEM

The Extended Finite Element Method (X-FEM) (Belytschko and Black, 1999; Moës et al, 1999) is an alternative fixed mesh approach for modelling problems involving discontinuities, singularities, complex geometries and localized deformations (Belytschko et al 2009). Examples of such problems include propagation of cracks, evolution of phase boundaries, holes and inclusions, modeling grain boundaries, and the evolution of dislocations. The aim of X-FEM is to model these kinds of discontinuities inside finite elements without remeshing the internal boundaries. Hence the finite elements can be defined completely independent of the shape of the discontinuity. Although X-FEM is not a meshfree method, the idea behind it is to retain most of the advantages of meshfree methods while diminishing their drawbacks.

#### 3.6.1. X-FEM Approximation

X-FEM uses the concept of partition of unity (Melenk and Babuska 1996) to extend the FE classical shape functions by adding discontinuous shape functions to the displacement field in order to enrich the FE approximation space near the discontinuity. A partition of unity in a domain  $\Omega$  is a set of  $n$  function  $f_i$  that satisfy the relationship

$$\sum_{i=1}^n f_i(x) = 1 \quad (3.3)$$

or in a more general form

$$\sum_{i=1}^n f_i(x)g(x) = g(x). \quad (3.4)$$

Finite element shape functions (as well as shape functions of many meshfree approximations) also satisfy the partition of unity condition (Belytschko et al 2009):

$$\sum_{i=1}^n N_i(x) = 1 \quad (3.5)$$

where  $N_i$  are the shape functions. Taking advantage of this property, the FE approximation space can be enriched to allow representation of a discontinuity. However it is important to find the appropriate enrichment function for a particular

type of discontinuity, as well as the parts of the approximation which need to be enriched. Several enrichment functions exist for modelling a variety of discontinuities. In the case of a cracked structures (figure 3.9), which is the case X-FEM was initially developed for, the X-FEM approximation space can be represented by

$$u(x) = \sum_i u_i N_i(x) + \sum_j a_j M_j(x). \quad (3.6)$$

In the above equation, the first function on the right hand side shows the conventional finite element approximation of the displacement field in an element where  $N_i(x)$  are the classical shape functions associated to the nodal degrees of freedom,  $u_i$ .  $M_j(x)$ , supported by enriched degrees of freedom,  $a_j$ , are the local enrichment function of node  $j$ , constructed by multiplying a classical  $N_j(x)$  shape function with a global enrichment function  $\psi(x)$ :

$$M_j = N_j(x) \cdot \psi(x). \quad (3.7)$$

As illustrated in figure 3.9, a Heaviside function representing a jump between -1 and +1 on the two sides of the crack, is used to enrich the nodes whose corresponding shape function support is cut by the crack boundary. Also, a tip function is used to enrich the nodes whose corresponding shape function support is cut by the crack tip.

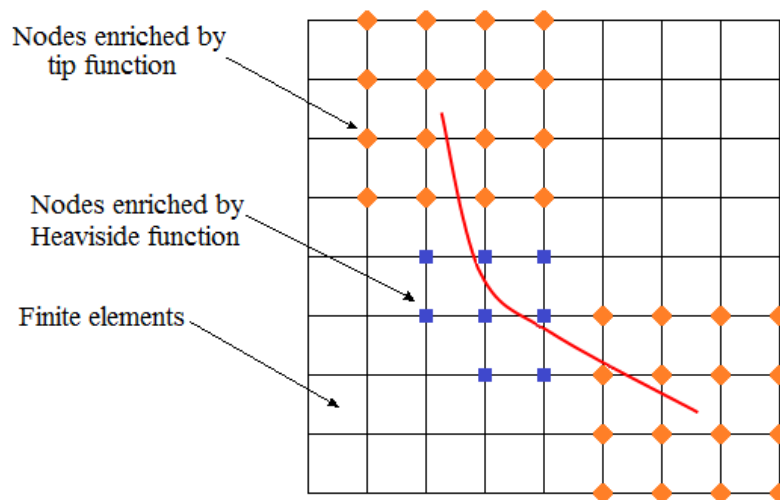


Figure 3.9. X-FEM model of a crack.

### 3.6.2. X-FEM for Modelling Holes and Inclusions

In the case of structural optimization problems, X-FEM can be implemented to represent the evolving boundary of a design on a fixed grid FE mesh with no need for remeshing. In this case the elements which lie on the design boundary experience a material-void discontinuity. In this case, the X-FEM scheme for modelling holes and inclusions proposed by Sukumar et al (2001) can be utilized to represent the material void discontinuity using a Heaviside function:

$$u(x) = \sum_i N_i(x) H(x) u_i \quad (3.8)$$

where the Heaviside function  $H(x)$  has the following properties

$$H(x) = \begin{cases} 1 & \text{if } x \in D_s \\ 0 & \text{if } x \notin D_s \end{cases} \quad (3.9)$$

which means that the value of the Heaviside function is equal to 1 for the nodes and regions in the solid part of the design and switches to 0 for nodes and regions in the void part of the design domain. The above formulation of X-FEM for modelling holes and inclusions doesn't involve enrichment of shape functions; instead the Heaviside function is used to define the discontinuity. Hence, the same degrees of freedom as FEM will be defined for the X-FEM approximation space of a particular problem. To realize this X-FEM scheme, as proposed by Sukumar et al (2001), in the approximation of elements' stiffness matrix, one can remove the integration from the void part of the element and merely perform the integration over the solid sub-domain of the boundary elements (figure 3.10).

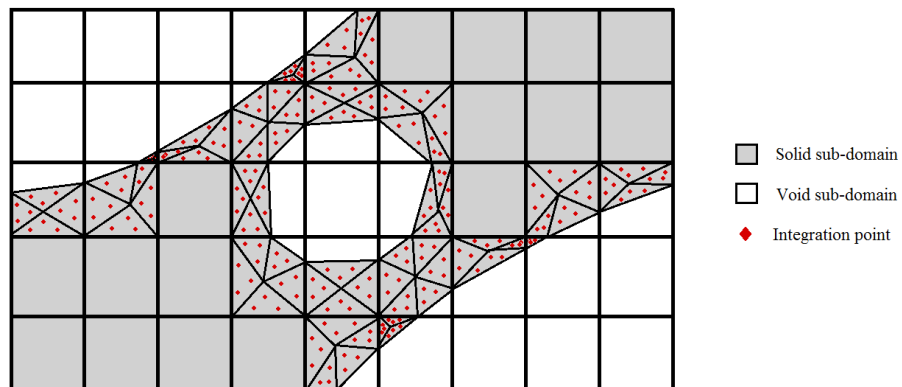


Figure 3.10. X-FEM representation of solid/void interfaces.

### 3.6.3. Application of X-FEM to Topology Optimization

There has been an increasing interest in the use of X-FEM for shape and topology optimization problems when representing the design boundary in a fixed mesh employing an implicit boundary representation approach, such as the level set method (LSM). Currently, most of the topology optimization methods which implement X-FEM approximation are either LSM optimization-based or use the LSM to define the boundary of the design. Miegroet et al (2005) proposed a shape optimization method by the combining level set method and X-FEM approximation and applied the method to the shape optimization of 2D benchmark problems. Miegroet and Duysinx (2009) used a similar LSM-XFEM based approach for the shape optimization of 3D problems. No enrichments were used in the X-FEM approximation of these two works. Lee et al (2009) introduced a topology optimization method by coupling LSM to X-FEM approximation with enrichment. Guo and Zhang (2011) proposed a topology optimization approach using X-FEM and LSM for optimizing problems having stress related objective or constraints. Wei et al (2010) applied X-FEM in a radial basis functions (RBFs) based level set model. A comparison of X-FEM based topology optimization solutions with density based ones showed higher smoothness and accuracy for the X-FEM solutions at the expense of a small increase in computational cost. Also a comparison of the accuracy of X-FEM with/without enrichment showed that X-FEM without enrichment can provide similar accurate results as X-FEM with enrichment but in a less computational time. Li et al (2012) and Villanueva and Maute (2014) extended the X-FEM integration schemes for 2D and 3D level set based topology optimization.

All the above mentioned literature have demonstrated X-FEM to be a powerful and promising approach to be used in the shape and topology optimization of continuum structures, as an alternative to the poor accuracy of density based FEM or the time-consuming remeshing schemes. A common aspect of all these X-FEM based topology optimization methods is the use of level set method to implicitly define the design boundary. Despite the accuracy and reliability of LSM, the use of LSM can increase the mathematical complexity of the optimization problem and restrict the usage of alternative topology optimization approaches.

### 3.7. Summary and conclusions

The wide range of optimization algorithms and numerical methods developed for the topology optimization of continuum structures confirms the significance and complexity of this subject. Although these optimization techniques have been successfully applied to benchmark problems, the application of many of these approaches to real engineering problems can be cumbersome. GAs due to the large number of design variables, have a high computational cost and are thus not suitable for large systems. The FE based Homogenization, SIMP and ESO/BESO methods have a good computational efficiency for large systems of real problems, however the solutions are provided with poor representation of the design geometry. Implementing FE mesh refinement schemes can improve the boundary representation; however this decreases the computational efficiency of the structural optimization. Meshless methods have higher computational cost than FEM, and the imposition of boundary conditions is problematic.

X-FEM is found to be a promising numerical approach to be used as an alternative to FEM in structural optimization problems. X-FEM requires an implicit definition of the boundary within the FE framework, such as a level set method. Because of that, the research in the application of the X-FEM in structural optimization is limited to the level set based structural optimization approaches. This increases the mathematical complexity of the optimization approach. There is currently a gap in the knowledge of the application of X-FEM in material distribution methods, such as evolutionary topology optimization methods. A possible advantage of this implementation could be maintaining the simplicity and computational efficiency of the material distribution algorithms while improving the boundary representation and increasing the accuracy of the solution near the boundary by the use of X-FEM. This is the subject of this thesis.

# Chapter 4

## Two Dimensional Topology Optimization Studies

In this chapter, a new approach for the topology optimization of continuum structures, named Iso-XFEM is introduced. The optimization algorithm is developed and applied to the topology optimization of linear two dimensional problems and the solutions are compared with those obtained from conventional element based methods. The next chapter is the extension of the Iso-XFEM method to the topology optimization of three dimensional and real-life structures. In chapter 6, Iso-XFEM is implemented for the topology optimization of geometrically non-linear structures.

### 4.1. Introduction to the Iso-XFEM approach

Iso-XFEM is an evolutionary based optimization approach, i.e. based on the assumption that the optimized solution can be achieved by gradually removing the inefficient material from the design domain. The idea of the proposed Iso-XFEM method is to represent the design boundary using isolines (or isosurfaces in 3D problems) of structural performance while implementing X-FEM to calculate the properties of elements on the boundary (figure 4.1). The assumption is that the combination of isoline boundary representation and X-FEM allows smooth and accurate representation of the design geometry which is being iteratively optimized using an evolutionary-based algorithm. In the following sections, each of these elements of the proposed Iso-XFEM method is discussed in detail.

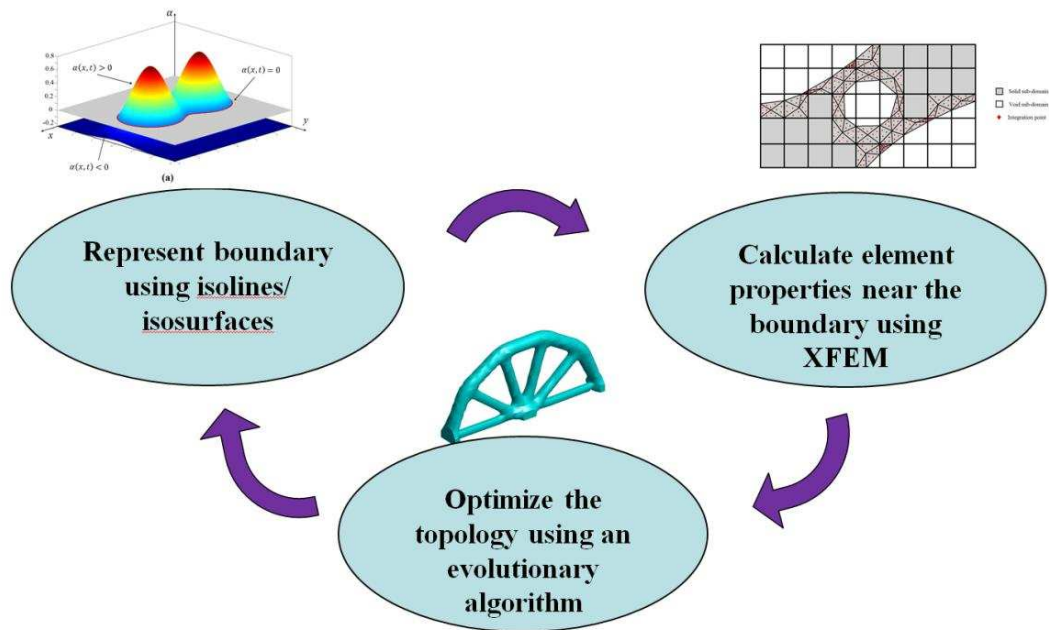


Figure 4.1. General idea of Iso-XFEM method.

## 4.2. Isoline design approach

In general, isolines are the lines that represent points of a constant value, named the isovalue, in a 2D space. The isoline approach is an implicit method of defining the boundaries of a design in a 2D fixed grid design domain. In this approach the boundaries are defined using the contours of a higher dimension (3D) of structural performance, such as Strain Energy Density (SED) or von Mises stress, obtained from finite element analysis of the design space. In structural optimization applications (Victoria et al 2009) the boundaries are defined by the intersection of the structural performance (SP) distribution with a minimum level of performance (MLP), which is typically increasing during the optimization process. Figure 4.2 shows a 2D fixed grid design domain discretized with a 30x30 mesh, where the intersection of the SED distribution as a structural performance criterion with a minimum level of SED gives the design boundary.

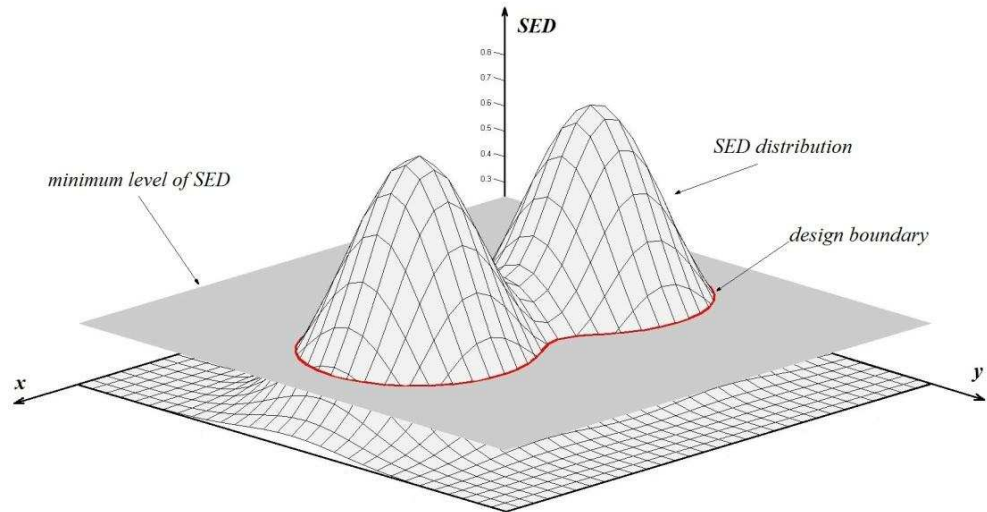


Figure 4.2 Design boundary represented by intersection of a structural performance (SP) with the minimum level of performance (MLP) in which strain energy density is used as the performance criterion.

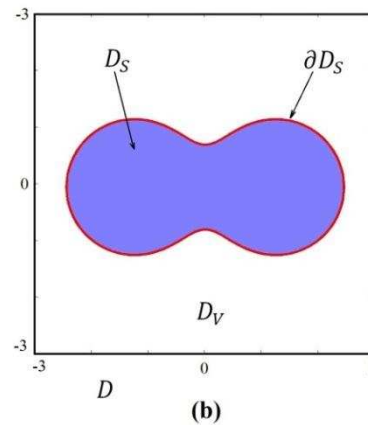
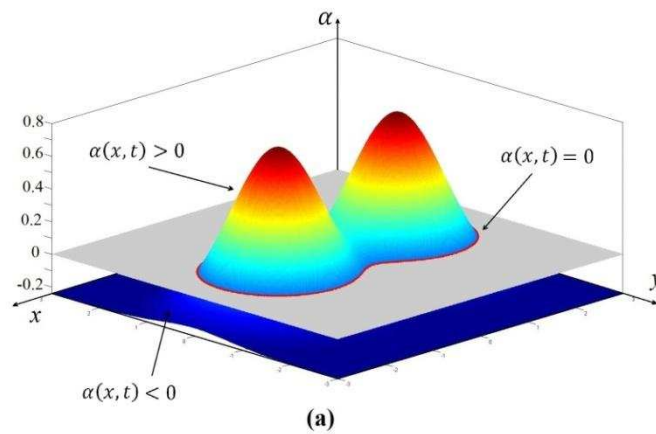


Figure 4.3. A 2D design domain represented by its relative structural performance ( $\alpha$ ). (a) Relative performance model. (b) Design domain ( $D$ ).



The regions of the design domain in which the structural performance is lower than the MLP is defined as the void part of the domain and those which have a higher value than the MLP form the solid part of the design domain. In order to mathematically represent this concept, the relative performance,  $\alpha$ , is defined as:

$$\alpha = SP - MLP \quad (4.1)$$

The values of relative performance in any part of the design domain can be obtained by interpolating the shape functions of the classical FEM:

$$\alpha(x) = \sum N_i(x) \cdot \alpha_i \quad (4.2)$$

where  $N_i$  are the shape functions and  $\alpha_i$  are the nodal relative performances. Following this, the design domain can be partitioned into void phase, boundary and solid phase, with respect to the values of relative performance, as

$$\alpha(x): \begin{cases} > 0 & \text{solid phase } (D_S) \\ = 0 & \text{boundary } (\partial D_S) \\ < 0 & \text{void phase } (D_V) \end{cases} \quad (4.3)$$

Figure 4.3 shows how the design space,  $D$ , from figure 4.2 is partitioned into  $D_S$ ,  $\partial D_S$  and  $D_V$  using the relative performance function  $\alpha(x)$ , distributed over the design space. One may notice that this implicit definition of the design space is very similar to the level set method. In the level set method, a level set function (typically a signed distance function) which is governed by the Hamilton-Jacobi equation is used to represent the design boundary and its evolution (Wang et al., 2003; Allaire et al, 2004). However, in the proposed approach, we have directly employed a structural performance to represent the design boundary. An application of the isoline design approach for topology design of a Mitchell type structure is shown in figure 4.4.

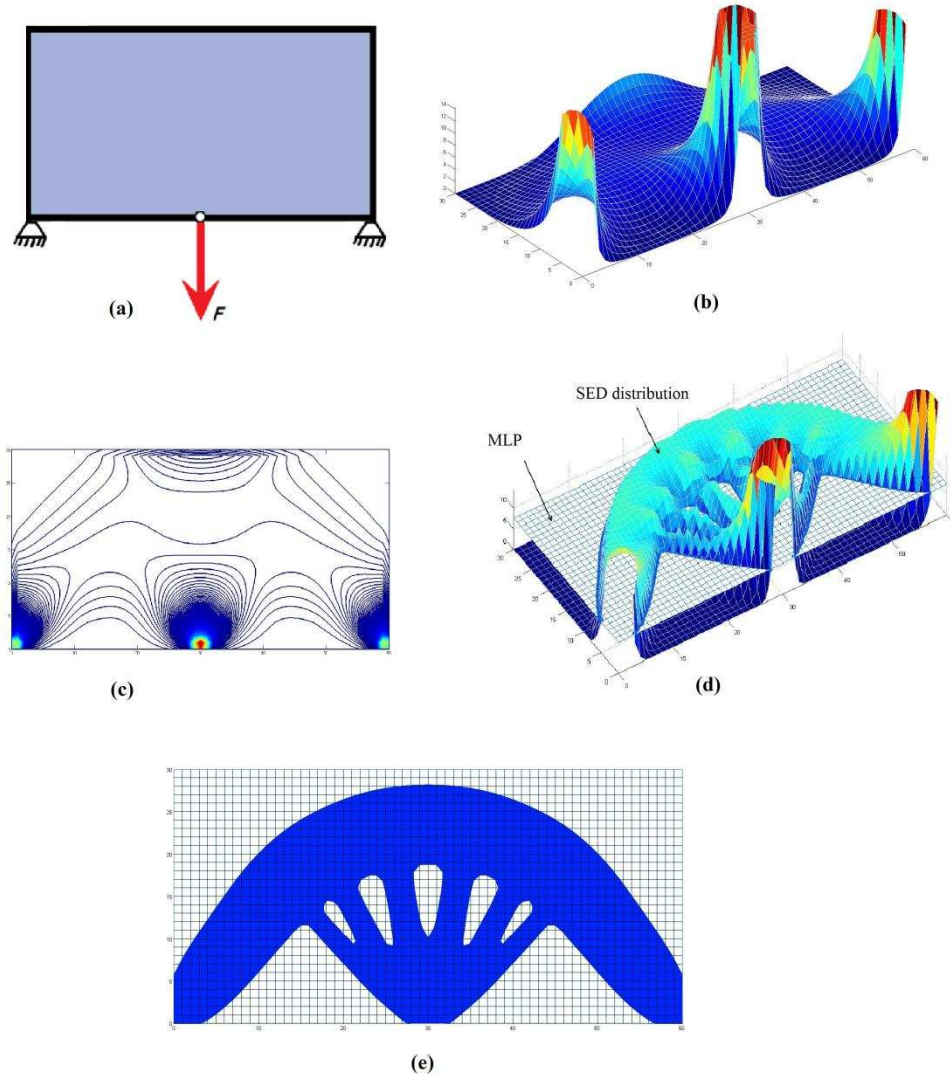


Figure 4.4. Application of isoline design approach in topology design of a Mitchell type structure: (a) Initial design domain with boundary conditions; (b) strain energy density (SED) distribution on the initial design domain; (c) isolines (contours) of SED of the initial design domain; (d) structural boundary represented by intersection of performance criterion (SED) distribution and minimum level of SED at  $VF = 0.5$ ; (e) design geometry at  $VF=0.5$ , represented in a fixed grid design space.

By implementing the above isoline approach, the design boundary is superimposed on the fixed grid finite elements, making three groups of elements in the FE design space: solid elements, void elements, and boundary elements (the elements which lie on the boundary) as demonstrated in figure 4.5. The contribution of solid and void elements to the FE framework could simply be considered using a soft-kill scheme (Huang and Xie, 2010a) in which instead of deleting the elements in the void phase, they are assigned a weak material property. However we would then lose the sub-element resolution of the design boundary definition resulting in a loss of accuracy and less optimal solution. In order to accurately represent the design

boundary whilst avoiding expensive remeshing operations, an X-FEM approach can be employed, as discussed in the next section.

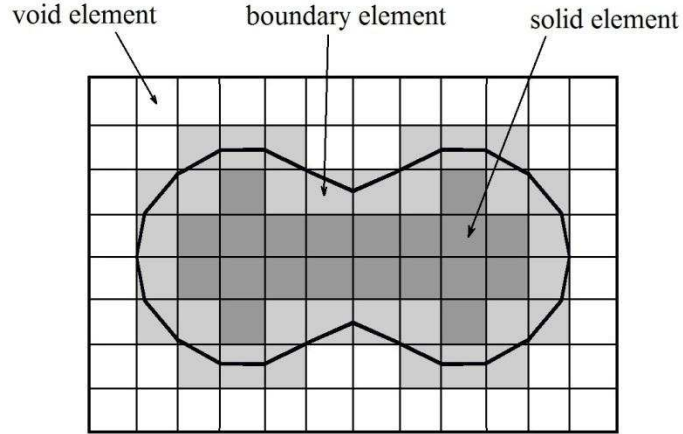


Figure 4.5. The elements are classified into 3 groups by superimposing the design geometry on the fixed grid FE mesh

### 4.3. X-FEM schemes for structural optimization

The material-void discontinuity imposed by the use of the isoline design approach in a fixed grid FE space can be modelled using an extended finite element method (X-FEM). As shown in chapter 3, X-FEM approximation uses a Heaviside function to represent the material-void discontinuity of evolving boundaries in a structural optimization problem:

$$u(x) = \sum_i N_i(x) H(x) u_i \quad (4.4)$$

in which the Heaviside function equals 1 in the solid sub-domain and zero in the void sub-domain of the elements. To realize this X-FEM scheme, we can use the classical FE approximation for the solid and void elements and treat the boundary elements (partially solid, partially void elements) using the special X-FEM integration scheme proposed by Daux et al (2000) and Sukumar et al (2001). In this approach, the stiffness matrix of a 2D boundary element is defined by:

$$k_e = \int_{\Omega} B^T C H(x) B t d\Omega \quad (4.5)$$

where  $\Omega$  is the element domain,  $B$  is the displacement differentiation matrix,  $C$  is the elasticity matrix for the solid material and  $t$  is the thickness of the element. This equation omits void sub-domain of the element and performs integration only on the solid sub-domain of the element, as:

$$k_e = \int_{\Omega_S} B^T C B t d\Omega \quad (4.6)$$

where  $\Omega_S$  denotes the solid sub-domain of the element. Dependent on the element type used in the FE discretization of the design domain, various partitioning schemes can be used to realize this integration approach.

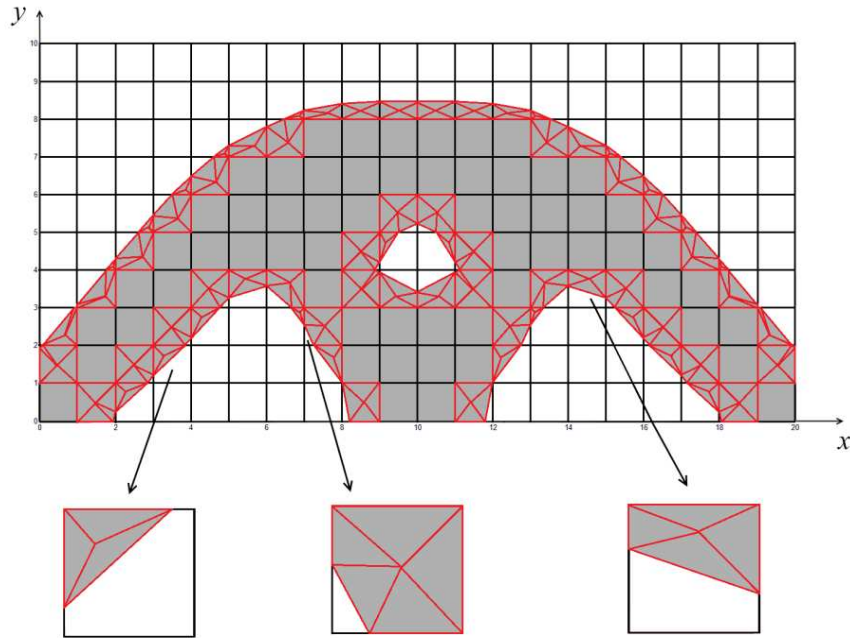


Figure 4.6. Solid sub-domain of the boundary elements are partitioned into several sub-triangles.

#### 4.4. 2D X-FEM integration schemes

In the case of 2D quad elements, the integral given by equation (4.6) can be numerically calculated by dividing the solid sub-domain of the boundary elements into sub-triangles (figure 4.6) and performing the integration using the Gauss quadrature method:

$$k_e = \sum_{i=1}^n \int_{T_i} B^T C B t d\Omega \quad (4.7)$$

where  $n$  is the number of sub-triangles inside the element and  $T$  denotes the triangle domain.

#### 4.4.1. Triangulation of boundary elements

The topology and shape of a boundary element can be found using the nodal values of relative performance which are defined by:

$$\alpha_i = SP_i - MLP \quad (4.8)$$

where  $i$  denotes the node number. Nodes having negative  $\alpha$  belong to the void part of the domain and nodes with positive  $\alpha$  are located in the solid domain. Therefore, an element which has at least one node with negative  $\alpha$  and one with positive  $\alpha$  is a boundary element. The intersection point of the boundary and element edge between two neighbouring nodes  $i$  and  $j$  can be found using a bilinear interpolation of relative structural performances ( $\alpha$ ) and shape functions:

$$x_i = \frac{l_{ij} \alpha_i}{1 - \frac{\alpha_i}{\alpha_j}} \quad (4.9)$$

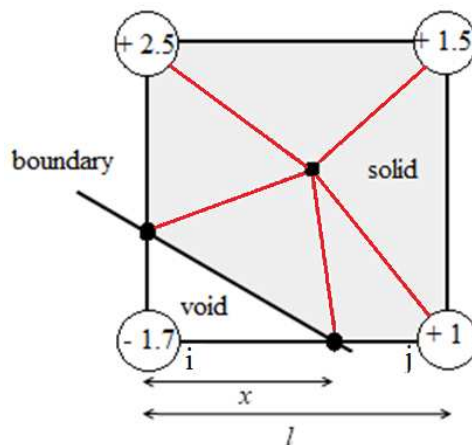


Figure 4.7. A boundary element with typical values for relative structural performance.

where  $x_i$  is the distance between node  $i$  and the intersection point, and  $l_{ij}$  is the element length between the nodes  $i$  and  $j$ . Depending on its topology, a boundary element can have 2 or 4 intersection points. The sub-triangles can be defined by defining an extra point inside the solid sub-domain of the element (typically in the centre of the solid area) and connecting it to the solid nodes as well as the intersection points. Figure 4.7 shows a boundary element with two intersection points and typical values for  $\alpha_i$ .

#### 4.4.2. Gauss quadrature integration method

In order to apply the Gauss quadrature method to calculate the integrals over the sub-triangles, the two dimensional integrals in terms of the physical coordinates are transferred to the triangle's natural coordinates and represented as a series of weighted functions:

$$\iint_{\Omega_e} F(x, y) dx dy = A \sum_{i=1}^m w_i F(\xi_1^i, \xi_2^i, \xi_3^i) \quad (4.10)$$

where  $m$  is the number of gauss points,  $\xi$  the coordinates of the gauss points  $A$  is the area of the triangle and  $w_i$  is weighting factor. Substituting equation (4.10) into equation (4.7), the element stiffness matrix can be obtained by

$$k_e = \sum_{i=1}^n \sum_{j=1}^m A_i w_j f(\xi_1^j, \xi_2^j, \xi_3^j) \quad (4.11)$$

where

$$f = B^T C B t$$

with  $n$  the number of sub-triangles in the solid domain of the element. In the case of linear quad elements with second order shape function, for each sub-triangle three Gauss points are required for calculation of the integral. The two options are midline Gauss points (figure 4.8b) and the gauss points inside the triangles (figure 4.8c). Figure 4.9 shows the proposed X-FEM integration schemes. To validate the proposed X-FEM approach, the stiffness matrix of a fully solid rectangular element having the Young's modulus  $E=1$  and the Poisson's ratio  $\nu=0.3$  was calculated, by two different schemes: first, using the classical finite element approximation;

second, using the X-FEM scheme described above in which the element is divided into sub-triangles and integration is performed using gauss quadrature for triangles, as shown in figure 4.10. Both methods resulted in exactly the same stiffness matrix for the element, thus validating the X-FEM scheme.

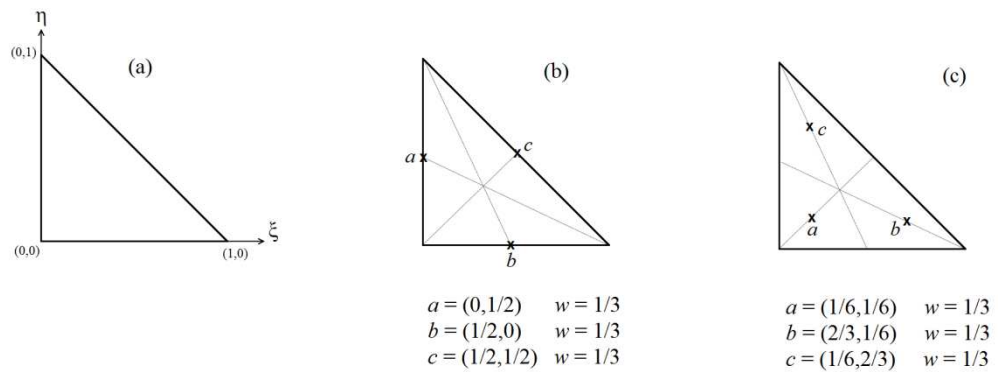


Figure 4.8 Gauss quadrature in triangles: (a) standard triangle (b) second order integration with midline Gauss points (c) second order integration with internal Gauss points .

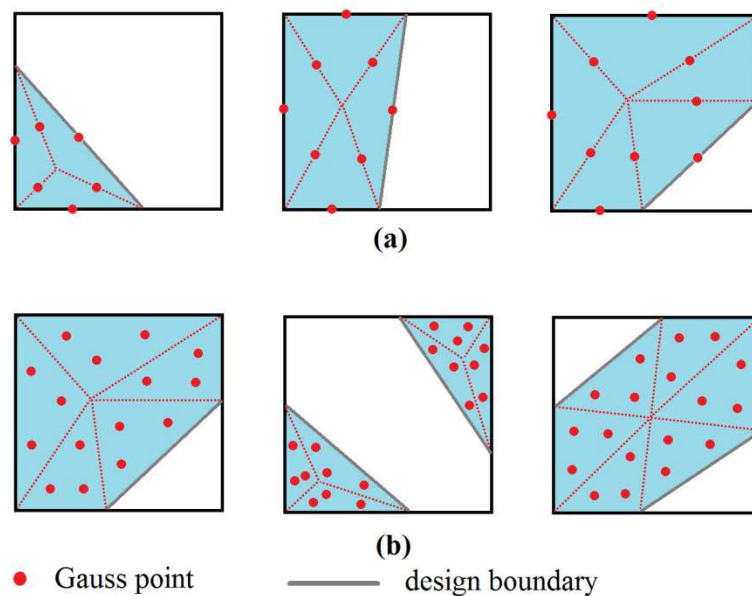


Figure 4.9 X-FEM integration scheme (a) sub-triangles with midline Gauss points (b) sub-triangles with internal Gauss points.

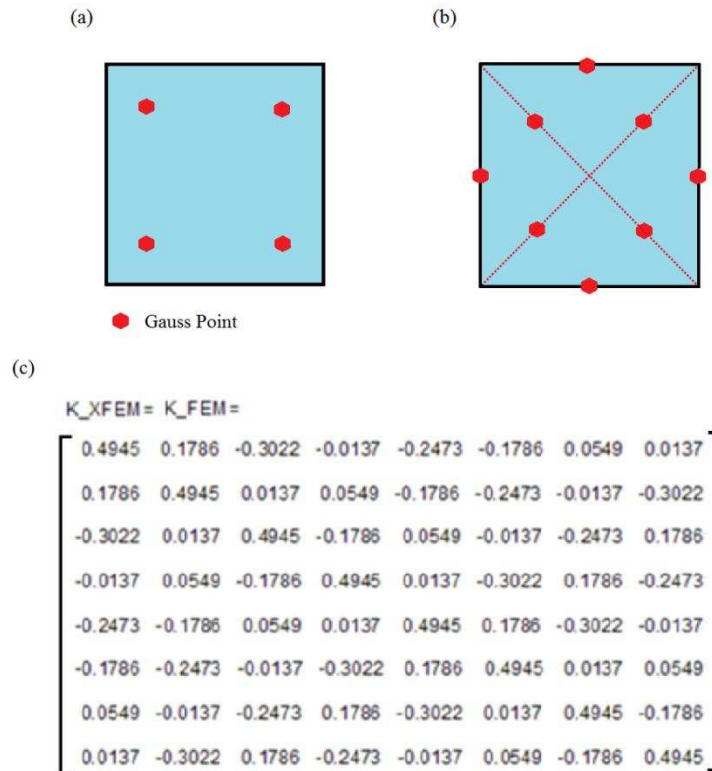


Figure 4.10 (a) A solid element represented by classical FEM. (b) The solid element represented by the proposed X-FEM scheme. (c) The stiffness matrix obtained using both FEM ( $K_{FEM}$ ) and X-FEM ( $K_{XFEM}$ ) approaches.

## 4.5. Stiffness design using isovalue-based evolutionary optimization process

### 4.5.1. Stiffness design in evolutionary structural optimization

As discussed in chapter 2, the basis of evolutionary structural optimization (ESO) is to achieve an optimal design by gradually removing inefficient elements from the design domain. The original idea of ESO was to remove low stress elements from the structure until an optimum was achieved (Xie and Steven 1993). Later it was further extended to problems of frequency optimization by introducing element sensitivity numbers calculated from eigenvalue analysis of the problem (Xie and Steven 1994). ESO was first applied to stiffness optimization by Chu et al (2006). Stiffness design is an important consideration for designing structures like bridges and buildings to keep the maximum deflection within a specified limit. A common measure used for stiffness design is mean compliance which is the



inverse of overall stiffness. In a linear problem, the mean compliance  $C$ , is equal to the total strain energy (SE) of the structure or the work done by the external loads:

$$C = \frac{1}{2}U^T K U = \frac{1}{2}F^T U \quad (4.12)$$

where  $K$ ,  $U$  and  $F$  are global stiffness matrix, displacement vector and force vector, respectively. In ESO/BESO, an element itself (rather than a property of the element such as relative density in SIMP) is considered as a design variable. Considering compliance (the total strain energy) as the objective function to be minimized, the sensitivity of the objective function for adding/removing element would be equal to the contribution of that element to the total strain energy:

$$S_e = \frac{\partial C}{\partial x_e} = \frac{1}{2}u_e^T k_e u_e \quad (4.13)$$

where  $x_e$  denotes element density, and  $u_e$  and  $k_e$  are the elemental displacement vector and stiffness matrix, respectively. If a non-uniform mesh is used for the FE model, the sensitivity numbers can be modified by considering the effect of an element's volume. Hence in this case the element strain energy density (SED) is used:

$$S_e = \frac{1}{2}u_e^T k_e u_e / v_e \quad (4.13)$$

It is evident that minimal increase in the objective function is achieved when elements with low sensitivity numbers which have a lower value of strain energy/strain energy density are removed. Hence in order to minimize the strain energy (maximize the stiffness), low SED elements should be systematically removed. This is the principle of stiffness optimization in ESO/BESO.

#### 4.5.2. Stiffness design using Iso-XFEM method

In the proposed Iso-XFEM method, a similar evolutionary-based concept for structural optimization to that discussed above is considered for stiffness design i.e. the structure evolves to an optimum by removal of low SED material. However, the slight modification is that instead of removing/adding a range of low/high SED elements, the material is removed/ redistributed in low/high SED

regions. That means the whole element or just a part of the element can be affected by the process of material removal/redistribution. In this case, the material distribution inside the element is found by the use of the isoline approach and the properties of the element is calculated using the proposed X-FEM scheme, as discussed in section 4.4. Therefore the same objective function as equation (2.4) is used with a final volume fraction constraint of  $V^c$ :

$$\begin{aligned} \text{minimize: } C &= \frac{1}{2} U^T K U = \frac{1}{2} \sum_{e=1}^n u_e^T k_e u_e & (4.14) \\ \text{subject to: } & \frac{\sum_{e=1}^n v_e x_e}{V_0} = V^c \end{aligned}$$

In order to realize the material removal/redistribution using the isoline approach, a minimum level of performance (MLP) is calculated from the following equation and is gradually increased throughout the optimization process until the volume fraction constraint is met.

$$MLP_{it} = RF_{it} \times SP_{max}/VF_{it} \quad (4.15)$$

where  $SP_{max}$  is the maximum performance over the design domain ( $SED_{max}$  in the case of stiffness design) and  $RF_{it}$  is the redistribution factor for the current iteration. it denotes the iteration number,  $VF_{it}$  is the current volume fraction of the solid material to the whole domain, and is used in the above equation to accelerate the material removal process at lower volume fractions. With the current redistribution factor, the iterative process of the extended finite element analysis and material removal/redistribution takes place until the percentage change in volume fraction is less than a minimum value  $\Delta V$ , which means that a steady state is almost reached. Then, the redistribution factor is increased by adding a performance evolutionary rate, PER:

$$RF_{it} = RF_{it-1} + PER. \quad (4.16)$$

With the new redistribution factor, the extended finite element analysis and material removal/redistribution is repeated until a new steady state is reached. The evolutionary process continues until a desired optimum, such as a prescribed volume fraction ( $V^c$ ) is reached.

### Stabilization of evolutionary process

To stabilize the evolutionary process in BESO, Huang and Xie (2007a) proposed an effective stabilization scheme by averaging the sensitivity number with its historical information. It was found helpful in the proposed Iso-XFEM method to use a similar approach for stabilizing the optimization process. Therefore, the nodal values of relative performance is modified by

$$\alpha_i = \frac{\alpha_i^{it} + \alpha_i^{it-1}}{2} \quad (4.17)$$

in every iteration apart from the first one. In the above equation  $i$  denotes the node number.

### Optimization parameters

The optimization parameter used in the Iso-XFM method is the performance evolution rate, PER. The number of iterations in the evolutionary process in the proposed method is affected by the value of evolution rate as well as the maximum performance  $SP_{max}$ . Selecting a high evolution rate can reduce the computational time. However, very high values of evolution rate may result in local optima or non-convergent solutions. A typical value for the performance evolution rate can be obtained from

$$PER = 0.01 \times \frac{SED_{ave}}{SED_{max}} \quad (4.18)$$

where  $SED_{ave}$  is the average SED for the fully solid design domain.

### **4.5.3. Steps in the Iso-XFEM algorithm**

Figure 4.11 illustrates the topology optimization procedure used, which in general consists of the following steps:

- 1- Initialization: in this step, the dimensions of the design domain, fixed grid mesh and initial material distribution within the design domain are defined; boundary and loading conditions are applied and the parameters of the

optimization algorithm, such as performance evolution rate PER and final volume fraction  $V^c$ , are defined.

- 2- Finite Element Analysis: a classical finite element analysis of the initial design domain is performed.
- 3- Calculate the structural performance criteria over the design domain (SED in the case of stiffness design).
- 4- Calculate the Minimum Level of Performance (MLP) from equation 4.15.
- 5- Calculate relative performance  $\alpha$  (equation 4.1) at nodes and use the stabilization scheme (equation 4.17).
- 6- Calculate the redistribution factor RF.
- 7- Extract the design boundary from  $\alpha(x) = 0$ .
- 8- Check convergence by comparing the convergence criteria with the defined convergence threshold. If the convergence condition is satisfied, jump to step 11, otherwise it progress to the next step.
- 9- Perform an X-FEM structural analysis on the fixed grid design domain. Using the nodal values of relative performance  $\alpha$ , the elements are categorized into three groups: solid, void and boundary elements. Solid and void elements are treated using classical finite element approximation. The stiffness matrix of the boundary elements are calculated by partitioning the solid sub-domain into several sub-triangles and applying the gauss quadrature integration scheme described in section 4.4. The global stiffness matrix is calculated by assembling the element stiffness matrices.
- 10- Go to step 3.
- 11- Stop the optimization process.

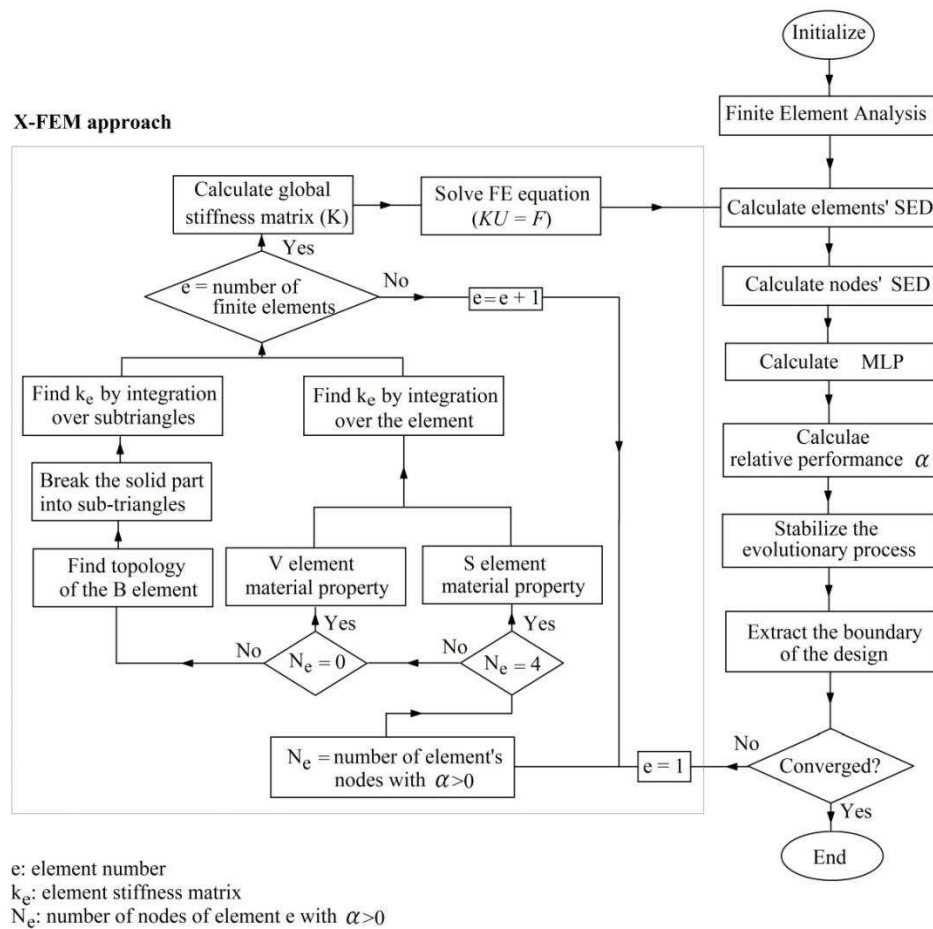


Figure 4.11. Flowchart of optimization algorithm.

## 4.6. Numerical experiments

Two experiments were considered in this chapter to investigate the application of the proposed Iso-XFEM method to the topology optimization of 2D structures. A Matlab code was developed to present the topology optimization of the 2D test cases of the experiments.

In experiment 1, the proposed method was applied to topology optimization of a symmetric and a non-symmetric cantilever plate, frequently used in the literature. This experiment started with an initial examination of convergence, followed by evaluation of Iso-XFEM solutions. Then the Iso-XFEM solutions were compared with the solutions obtained from applying BESO to the same problems, in terms of converged objective, computational time-cost and surface roughness. The effect of initial mesh on final topology optimised solutions was also studied in this experiment.

The second experiment was the application of the proposed method to topology optimization of a ‘C’ clip. This test case was chosen to allow investigating the application of the proposed method to the structures having a more complex design domain than the frequently used benchmark problems, requiring modeling with a non-uniform FE mesh.

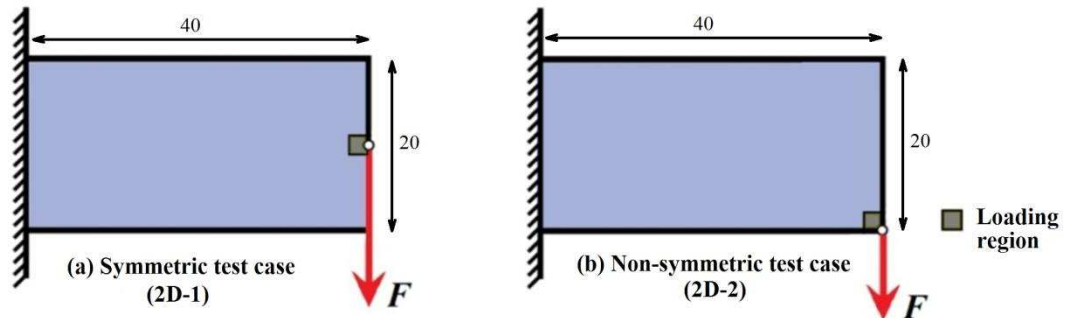


Figure 4.12. The two test cases.

#### 4.6.1. Experiment 1: cantilever plate

The purpose of this experiment was to apply the proposed Iso-XFEM method to the topology optimization of 2D rectangular domains as a first validation stage prior to full 3D implementation. Two test cases were used in the first 2D validation study, i.e. experiment 1, as shown in figure 4.12. These are frequently used benchmark problems in previous studies. A consistent dimensionless set of parameters was used for both test cases. Test case 2D-1 was a short cantilever plate having length 40, height 20 and thickness 1, with a unit concentrated load applied in the middle of the free end (symmetric case). Test case 2D-2 was a cantilever plate having the same dimensions as test case 2D-1 but with the load applied at the bottom of the free end (non-symmetric case). The material properties of the solid material were Young’s modulus  $E=1$  and Poisson’s ratio  $\nu = 0.3$ . The objective was to minimize the total strain energy for a target volume of 50% of the initial design domain. To avoid singularity issues with the concentrated load, the strain energy inside the loading regions shown in figure 4.12 were not used in the calculation of the total strain energy and the tip displacements were measured from outside the loading region along the line of loading. This was done to enable comparison of performance of different solutions obtained from implementing

different mesh densities, since the strain energy near the loading region tends to increase by refining the mesh. However, this scheme does not affect the topology of the solutions.

#### 4.6.1.1. Preliminary examination of convergence

The initial design domain was discretised using a  $60 \times 30$  mesh. The optimization started with a fully solid design domain. The evolution histories of the objective function and volume fraction for test cases 2D-1 and 2D-2 are shown in figures 4.13a and 4.14a. It can be seen that the strain energy increases, as material is gradually removed from the design domain, then reaches a constant value at convergence. The development of the topology in the iterative optimization processes for the two test cases are illustrated in figures 4.13b and 4.14b. It can be seen that initially a number of holes appear as the volume fraction decreases. After a certain number of iterations some of the holes merge to make larger holes, thus reducing the final complexity of the topology. It can be seen in figure 4.13b that the topology for the symmetric test case remains symmetric throughout the optimization which is an indication of good numerical accuracy. It can be seen in the final topologies that despite using a coarse mesh for this optimization problem, the final designs have clearly defined, smooth boundaries that need no further interpretation or smoothing (unlike standard SIMP and ESO/BESO methods). In the next two sections, the accuracy of the results are studied and the method is compared with standard BESO.

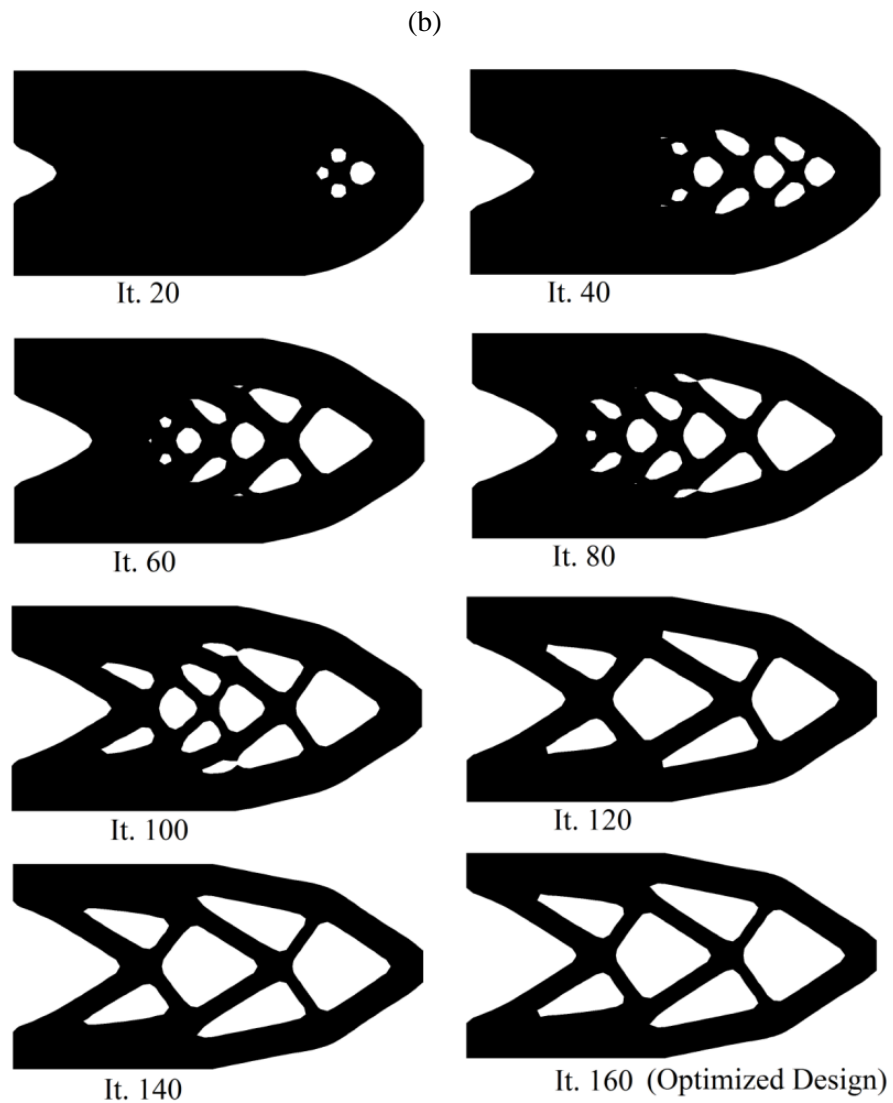
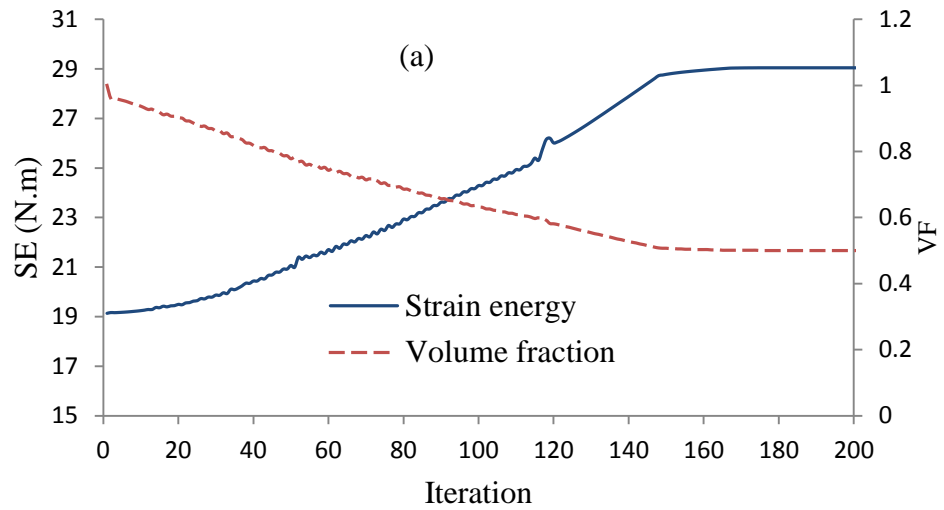


Figure 4.13. Test case 2D-1: (a) evolution history of objective function (SE) and volume fraction evolution of the topology (VF) (b) evolution of the topology.



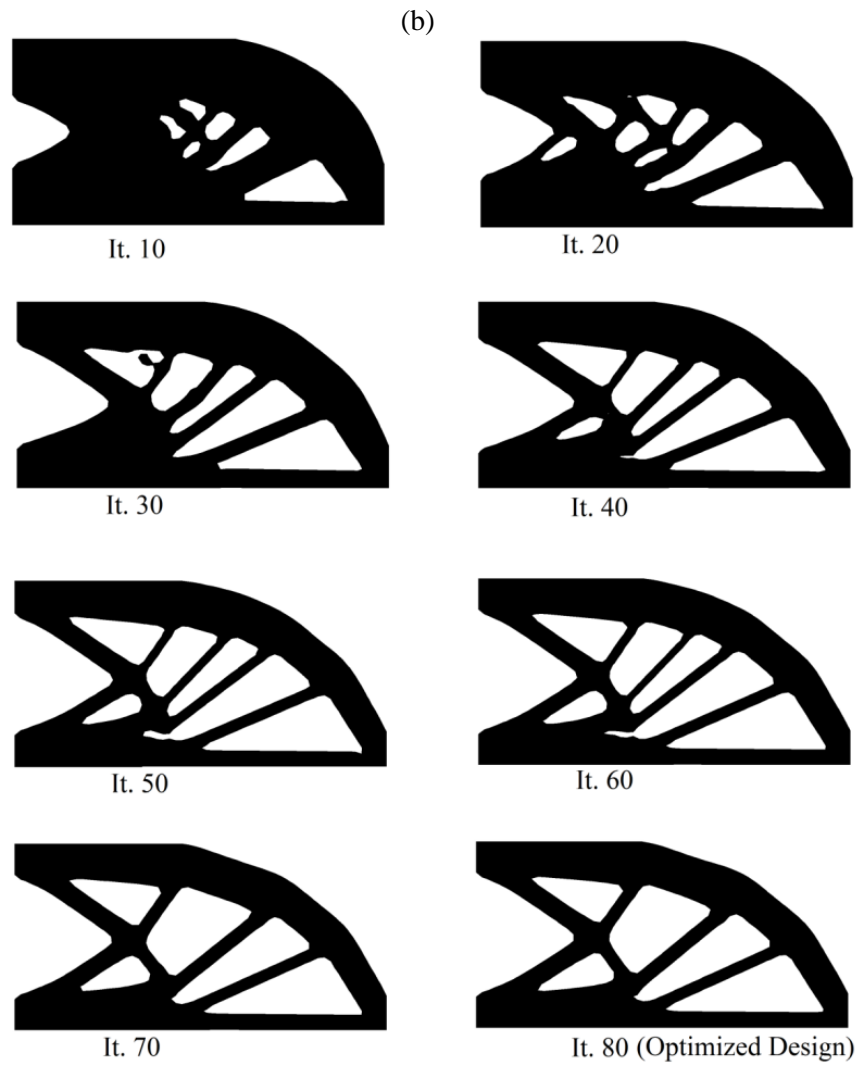
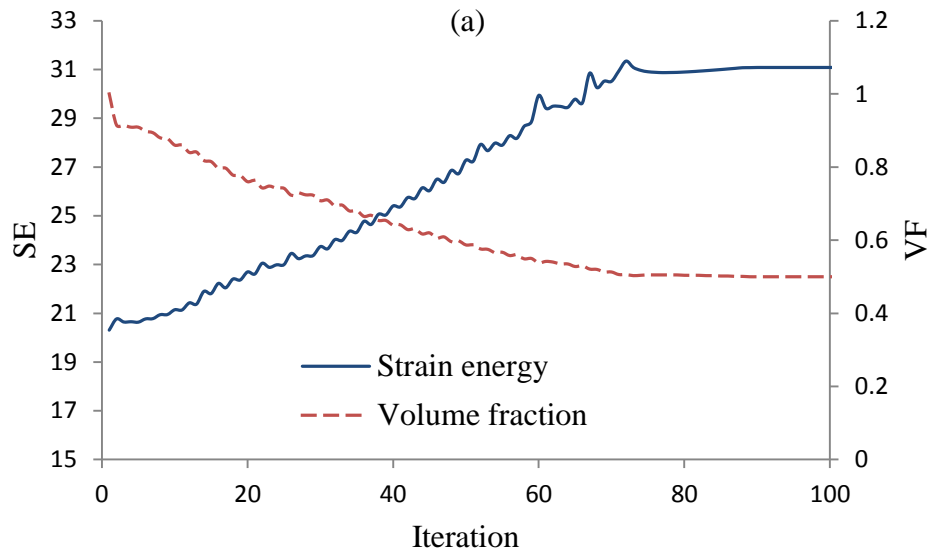


Figure 4.14. Test case 2D-2: (a) evolution history of objective function (SE) and volume fraction evolution of the topology (VF) (b) evolution of the topology.

#### 4.6.1.2. Evaluating Iso-XFEM solutions

In order to accurately evaluate the performance of the final solutions and the accuracy of the proposed method, the obtained solutions were discretized with a converged, fine-structured, finite element mesh and solved using the commercial finite element solver NASTRAN from MSC Software (Santa Ana, California, USA) (figure 4.15).

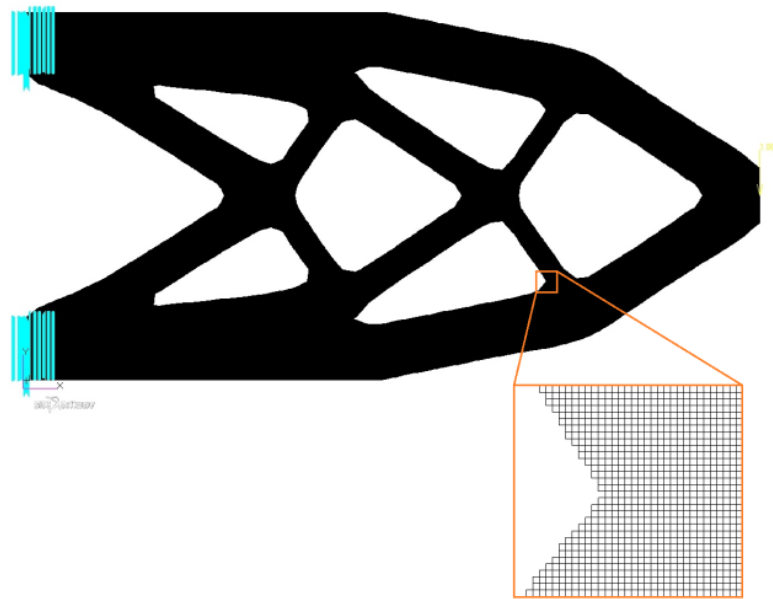


Figure 4.15. Iso-XFEM solution discretized using a converged, fine mesh and imported to NASTRAN for solution.

Table 4.1 compares the Iso-XFEM solutions and the converged NASTRAN solutions in terms of their strain energies and tip displacements. It can be seen that the Iso-XFEM solutions are very close to the regenerated NASTRAN solutions with percentage error less than 0.7. The small difference in the Iso-XFEM and NASTRAN results may be attributed to the different mesh size used in the two approaches and would be expected to decrease by reducing the mesh size of the design space. However it can be argued that the accuracy obtained using the coarse mesh is sufficient for the topology optimization and the added accuracy of a finer mesh will unnecessarily increase computational time. This is an increasingly important consideration when the method is used for the optimization of real-life 3D structures (as will be shown in the next chapter).

Table 4.1. Comparison of Iso-XFEM solutions and regenerated NASTRAN structures.

<b>Test case 1</b>	<b>Strain Energy</b>	<b>Tip Displacement</b>
X-FEM	29.81	57.08
NASTRAN	30.01	57.36
% Error	0.67	0.49
<b>Test case 2</b>	<b>Strain Energy</b>	<b>Tip Displacement</b>
X-FEM	30.82	61.77
NASTRAN	31.04	62.10
% Error	0.70	0.53

#### 4.6.1.3. Comparison with BESO solutions

The solutions obtained from the proposed method are compared with BESO solutions for a range of mesh sizes. BESO generates solutions with defined geometric boundaries through an evolutionary process which is similar to Iso-XFEM algorithm, allowing a fair comparison of the solutions of the two methods. The BESO solutions were obtained using a soft-kill BESO Matlab code (Huang and Xie, 2010a). In order to overcome the checkerboard problem (Jog and Harber, 1996) in the BESO solutions and retain the complexity of the converged solutions, a small filter radius of 1.2 times the element length, was used. The selected volume evolution rate for BESO was 0.004 which was chosen to give approximately the same number of iterations to converge as the Iso-XFEM optimization approach (180-200 iterations).

#### - Comparison of strain energies

Comparing the topologies obtained using the two approaches for test case 2D-1 (the symmetric problem) one may notice that the converged solutions for the same mesh size have similar topologies (table 4.2); however the BESO solutions tend to have higher strain energies than the Iso-XFEM solutions. The reason for this is the poorer edge representation in the BESO method, which has reduced the performance of the converged solutions. At high mesh density the two methods had very similar performance, as would be expected, however, this is obviously much more computationally expensive. To increase the performance of these BESO solutions, additional post-processing is required to smooth the boundaries. In test case 2D-2, which is a non-symmetric problem, the two approaches have

generated different topologies (table 4.3). The strain energies of the Iso-XFEM solutions are again lower than the BESO ones, indicating better performance for the Iso-XFEM solutions.

It can be seen in tables 4.2 and 4.3 that both methods result in final topologies that are mesh dependent, which is generally the case for element based topology optimization methods. Pseudo mesh-independent topologies can be obtained by the use of coarsening actions such as filtering the sensitivities and increasing the filter radius (Huang and Xie, 2007a). However these methods still have their limits in terms of mesh independency and will tend to result in coarser solution that can have lower performance than more refined solutions. Generally speaking, increasing the mesh density increases the complexity of the converged solutions and reduces the strain energy of the solutions. Therefore increasing the mesh density can improve the performance of the final optimized result. This issue has been studied in earlier investigations for BESO (Aremu et al, 2013).

Table 4.2. Iso-XFEM and BESO solutions of test case 2D-1 for a range of mesh sizes.











Mesh	40 x 20	60 x 30	80 x 40	100 x 50	120 x 60
Iso-XFEM					
SE	29.49	29.04	28.91	28.85	28.88
BESO					
SE	31.55	30.34	30.08	30.06	29.80

Table 4.3. Iso-XFEM and BESO solutions of test case 2D-2 for a range of mesh sizes.











Mesh	40 x 20	60 x 30	80 x 40	100 x 50	120 x 60
Iso-XFEM					
SE	31.08	30.82	30.74	30.54	30.64
BESO					
SE	32.91	32.05	31.54	31.72	31.57

Figure 4.16 illustrates the strain energy of the converged solution as a function of mesh density for test case 2D-2. It shows that the Iso-XFEM optimization approach is more robust than the BESO method and that the strain energies of the Iso-XFEM solutions converged earlier than BESO ones. However, as the mesh gets finer the strain energies of the BESO solutions get closer to the Iso-XFEM solutions. It can be seen that increasing the mesh density results in the strain energies produced using Iso-XFEM approach and NASTRAN FEA become close, relative to the fluctuations in the data beyond  $1/h=80$ .

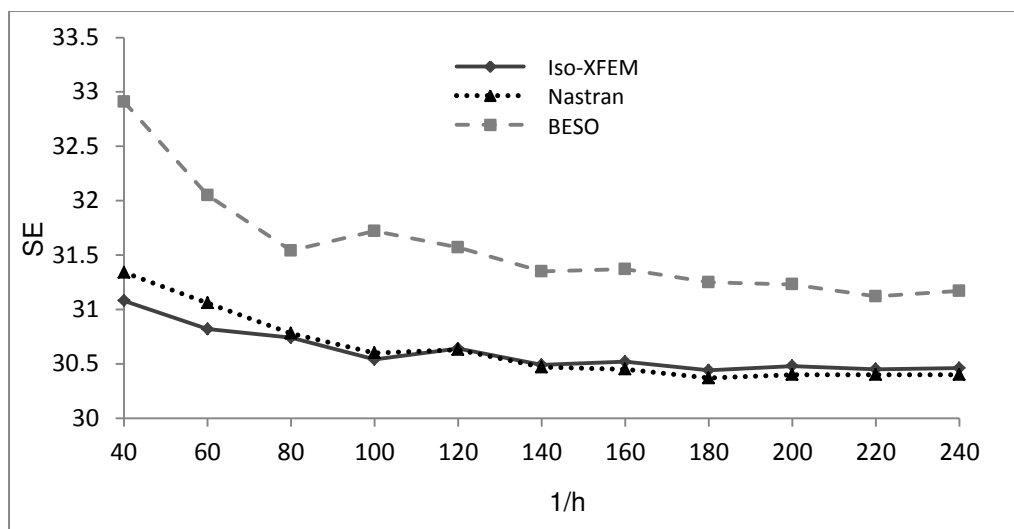


Figure 4.16. Changes in strain energy (SE) by reducing the mesh size (h) in test case 2D-2.

Table 4.4. Comparison of the time cost of BESO and X-FEM for 100 iterations

Approach\Mesh	40x20	60x30	80x40	100x50	120x60
Iso-XFEM	19 s	43 s	92 s	188 s	434 s
BESO	9 s	25 s	65 s	151 s	399 s
Ratio	211%	172%	142%	125%	109%

Table 4.4 compares the computational time for the two optimization methods for test case 1 after 100 iterations. It can be seen that at low mesh densities BESO is much faster than Iso-XFEM, however, the computational time ratio decreases with increasing mesh density. It is shown that BESO solutions require higher mesh

densities and more post-processing to obtain a smooth topology, therefore the total time for design will be decreased by using the Iso-XFEM method.

- Comparison of surface roughness

As a post processing stage, a Laplacian smoothing algorithm can be used to create smoother boundaries for the optimised topologies, if this is required for manufacture for instance. Laplacian smoothing is an iterative smoothing technique, commonly used in image processing and improving the quality of finite element meshes (Cannan et al, 1993; Freitag, 1997). In the image processing application, Laplacian smoothing operates by replacing the grey value of a pixel with an average of the grey values of neighbouring pixels. In the FE meshing application, the location of a vertex is modified using the average of the locations of neighbouring vertices (Vollmer et al, 1999). Figure 4.17 shows the boundaries before and after smoothing for the BESO solutions of test case 2D-2 for a range of mesh densities. The average surface roughness of the optimized topologies before smoothing and the number of iterations in the Laplacian smoothing are also included in the figure. The root mean square roughness ( $R_q$ ) was determined by comparison of the topology boundaries before and after the Laplacian smoothing, assuming that  $R_q$  of the topologies after smoothing is zero. It can be seen in Figure 4.17 that the surface roughness of the topologies from the BESO optimization increases as the mesh density decreases. Figure 4.18 shows the boundaries of Iso-XFEM solutions for the same test case before and after smoothing. Compared to the BESO solutions, the surface roughness of the Iso-XFEM solutions are much lower and they need far fewer iteration steps in the Laplacian smoothing. Also it can be seen that unlike the BESO solutions, the surface roughness of the Iso-XFEM solutions have little dependency on the mesh density. It can be seen that smoothing has a little effect on the shape of the Iso-XFEM solutions (figure 4.18), therefore the smoothed Iso-XFEM topologies are expected to have approximately the same performance as the Iso-XFEM solutions before smoothing. However in the case of BESO solutions (figure 4.17), it can be seen that the shape of the solutions are different before and after smoothing which means that the performance may have changed by smoothing, i.e. the BESO solutions may require

an additional stage of post-processing including shape optimization after smoothing.

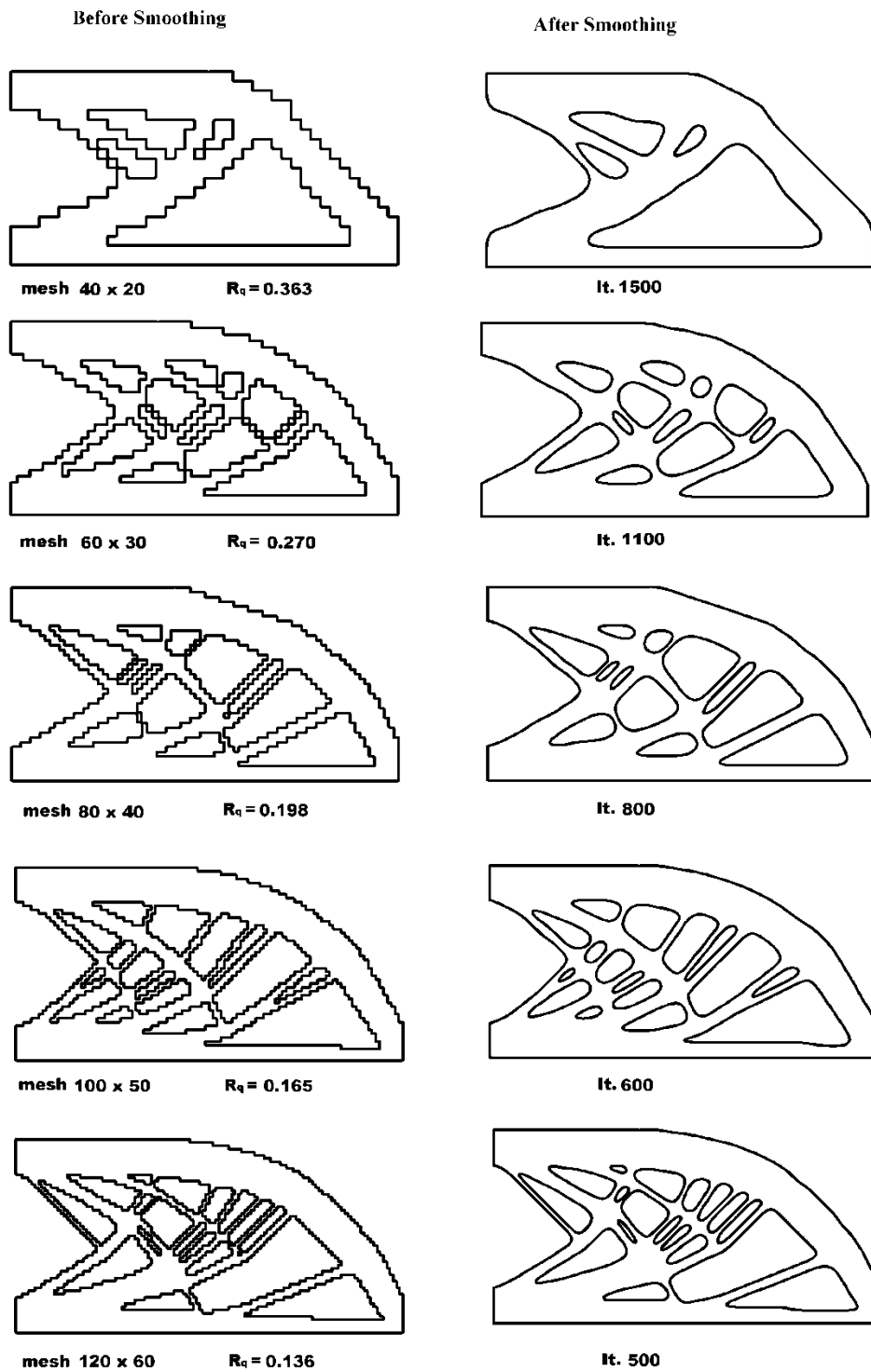


Figure 4.17. BESO solutions before/after smoothing.

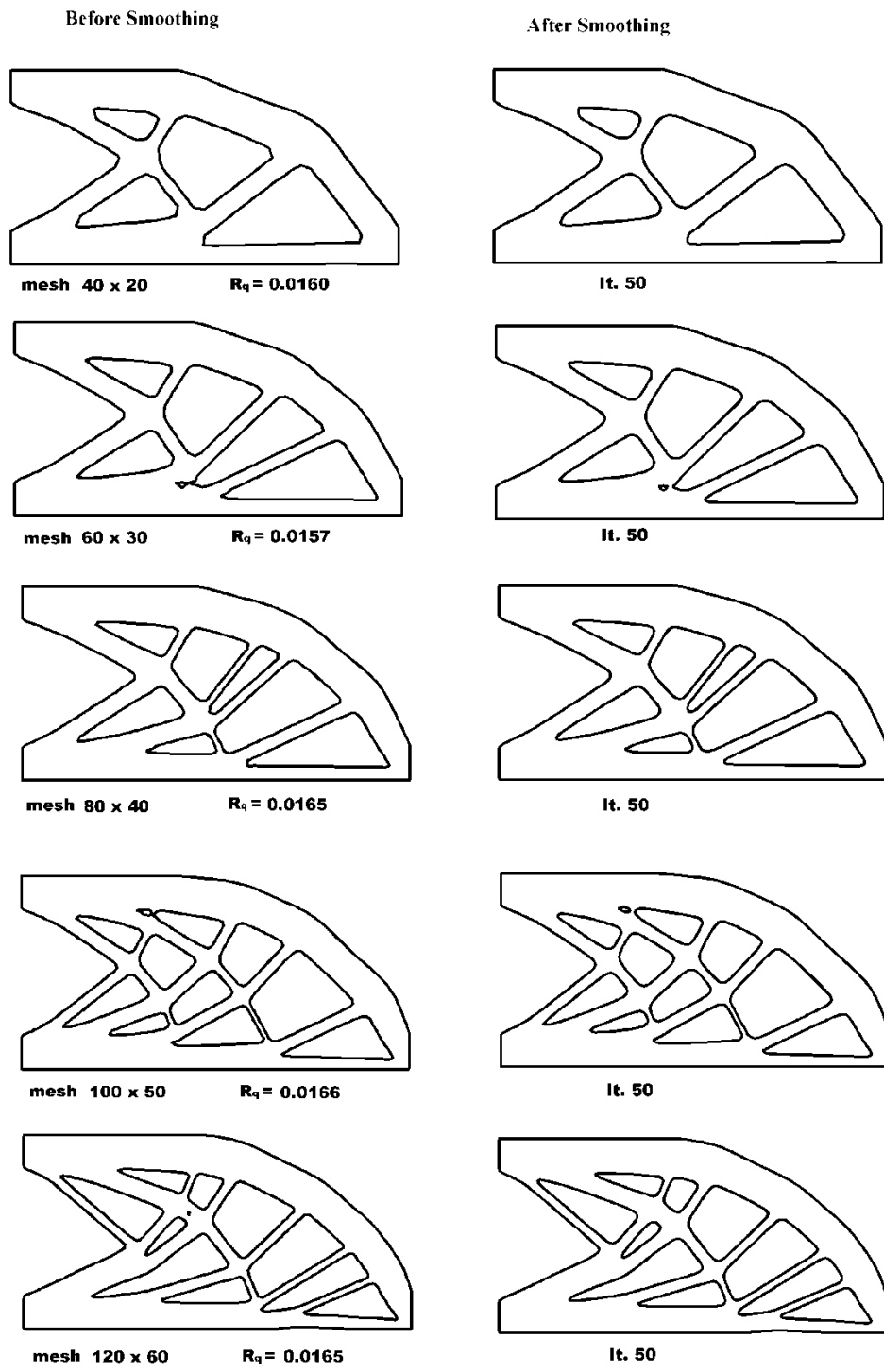


Figure 4.18. Iso-XFEM solutions before/after smoothing.



#### 4.6.2. Experiment 2: C clip

The purpose of this experiment was to apply the Iso-XFEM method to optimize 2D structures with non-rectangular design domain, requiring non-uniform FE mesh. To achieve this, a C clip with the loads and boundary conditions shown in figure 4.19 was considered as the third 2D test case of this chapter, i.e. test case 2D-3. Node A located on the middle of the left side of the clip is fixed in the x direction and node B on the right is fixed in both x and y directions. Due to the complexity of the initial geometry, the FE model of the structure was created in a commercial FE package (Abaqus) and then imported into Matlab in terms of Nodes and Elements matrices. Here, the optimization problem was defined as minimizing the strain energy of the clip subject to a volume fraction constraint of,  $VF^c = 0.4$ , and the strain energy density of the design domain is used as the structural performance criterion. The material used had a Young's modulus of 3 GPa and a Poisson's ratio of 0.3. The optimization parameters used for this test case were  $PER = 0.001$  and  $\Delta V = 0.005$ .

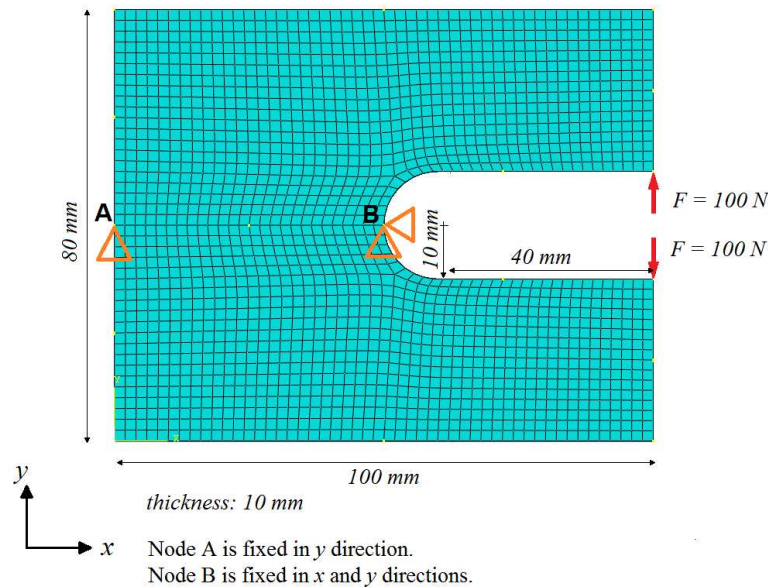


Figure 4.19. Design domain, FE mesh, loads and boundary conditions used for test case 2D-3 (C clip).

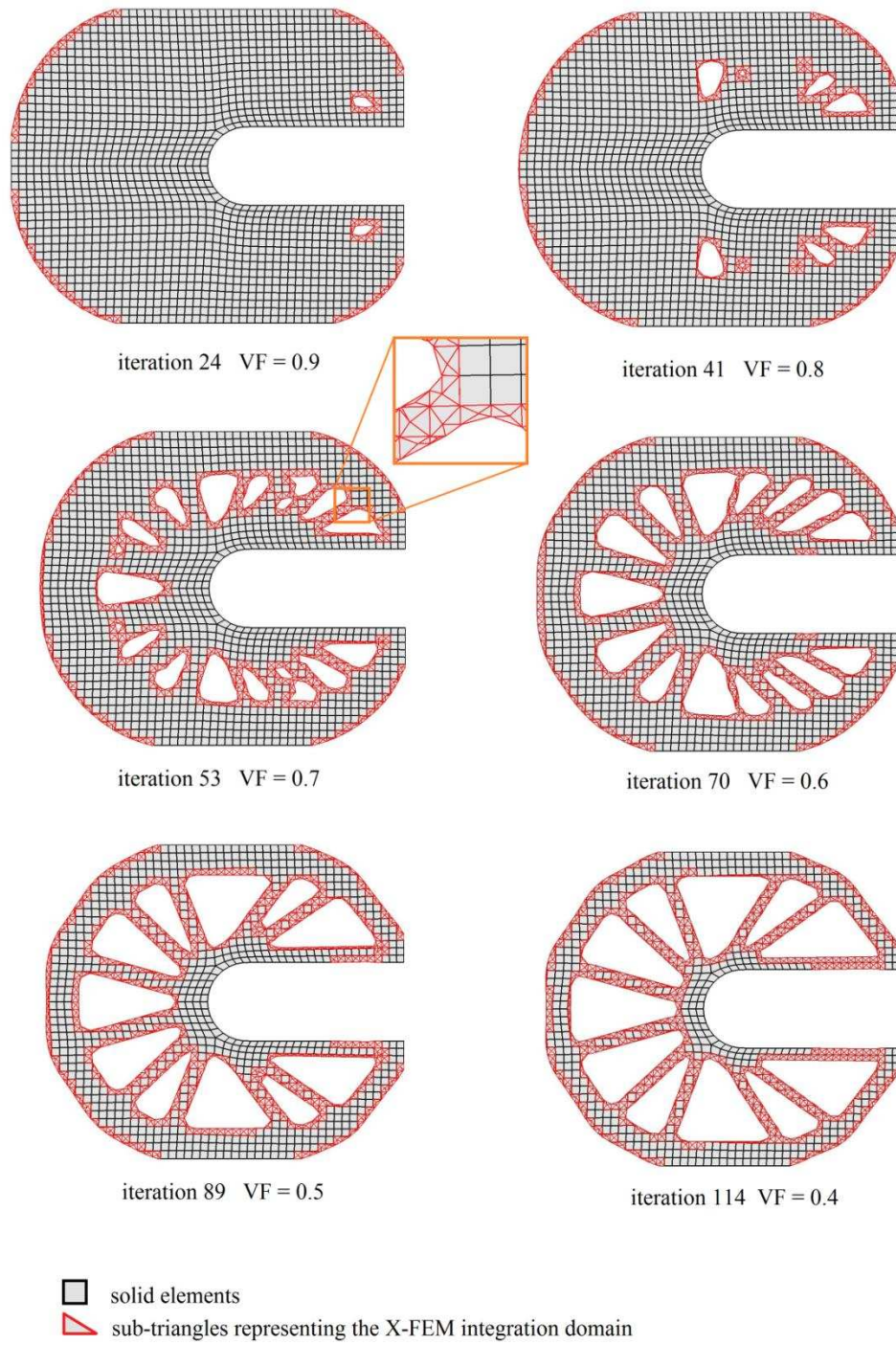


Figure 4.20. Evolution of the topology of the C clip

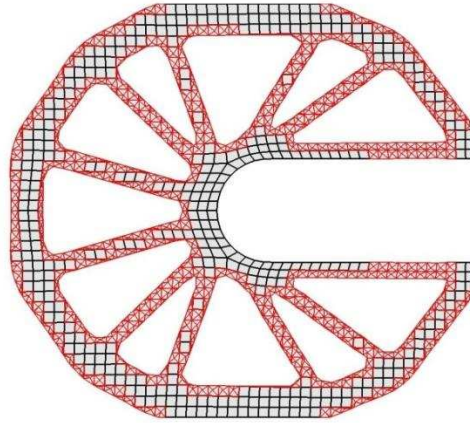


Figure 4.21. Converged solution for  $VF = 0.4$  at iteration 120.

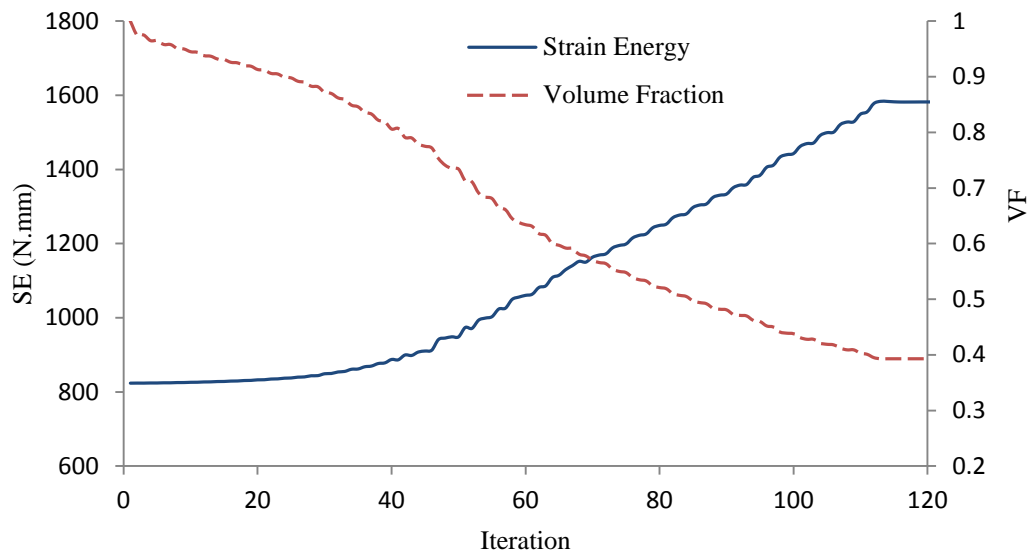


Figure 4.22. Evolution history of the objective function (SE) and volume fraction (VF) of the C clip test case for final volume fraction of  $VF_f = 0.4$ .

Figure 4.20 shows the resulting topologies for a range of volume fractions. It can be seen that the process of material removal was carried out by creating a number of holes in the low SED regions of the design space, followed by the merging and enlarging of holes until the desired volume fraction and convergence conditions were achieved. Note that the triangles near the boundary represent the X-FEM integration domains and are not elements. Therefore as no element is added or removed to/from the design domain, the number of degrees of freedom during the optimization process doesn't change. However, the boundary of the converged solution (figure 4.21) is smoothly defined by implementing the isoline and X-FEM

approaches. The evolution history of the objective function and volume fraction is shown in figure 4.22. It can be seen that by gradually removing material from the design domain, the strain energy has increased. The slight oscillations in the graphs of strain energy and volume fraction come from redistributing the material during the optimization process. The optimization converged after 120 evolutionary iterations which took 170 seconds.

#### **4.7. Summary and conclusions**

In this chapter, the Iso-XFEM method is introduced and applied to the topology optimization of 2D continuum structures. The results suggest that the use of X-FEM combined with isoline boundary representation has significant advantages, not only does it avoid time consuming remeshing techniques, but also generates structures that have smooth boundaries requiring little or no further interpretation or post processing. Using simple test geometries, it has been shown that X-FEM based topology optimization has the potential for greater accuracy and more robust solutions with less dependence on mesh size than BESO. Also it is shown that this approach has the potential to be implemented on more complex geometries though this needs to be established for more realistic, 3D geometries. It is anticipated however that this method is relatively simple to extend. A possible route to achieving this is to use 8-node brick elements where the solid part of the boundary element could be represented by sub-tetrahedrons. This is discussed in the next chapter.

# Chapter 5

## Three Dimensional Topology Optimization Studies

In this chapter the Iso-XFEM is extended to the topology optimization of three dimensions to enable “real-life” structures to be analysed. This involves the introduction of an isosurface design method to capture the design boundary during the optimization process and also extension of the X-FEM integration strategy into 3D. A preliminary investigation of convergence applies the method to the topology optimization of a cantilever plate, as a benchmark test case frequently used in the literature. After this, a more complex test case is considered in order to investigate the possibility of the application of this approach to the topology optimization of real-life structures. A discussion section at the end of the chapter includes suggestions on how to extend the method to treat alternative objective functions or constraints. The stiffness design of a structure with additional displacement constraint is considered as an example of the further implementation of the proposed method.

### 5.1. Isosurface design approach

Similar to the isoline design approach, an isosurface is defined as a surface representing points of a constant value (isovalue) in a 3D space. In this case, the boundaries are defined using contours of 4D structural performance criteria such as SED in stiffness design problems. The intersection of structural performance criteria with a minimum level of performance (zero relative performance) is then used to define the design boundary. Figure 5.1 illustrates isosurfaces of SED for a cantilever beam. It can be seen that the isosurfaces of SED, closer to the loaded

point and the fixed edges, have a higher isovalue which indicates the efficient use of material in those regions. The purpose of isosurface design approach is to systematically identify a threshold isovalue to separate the material and void regions of the design domain using the corresponding isosurface. Similar equations to 4.1-4.3 of the previous chapter can be used to define the relative performance,  $\alpha$ , over the design space.

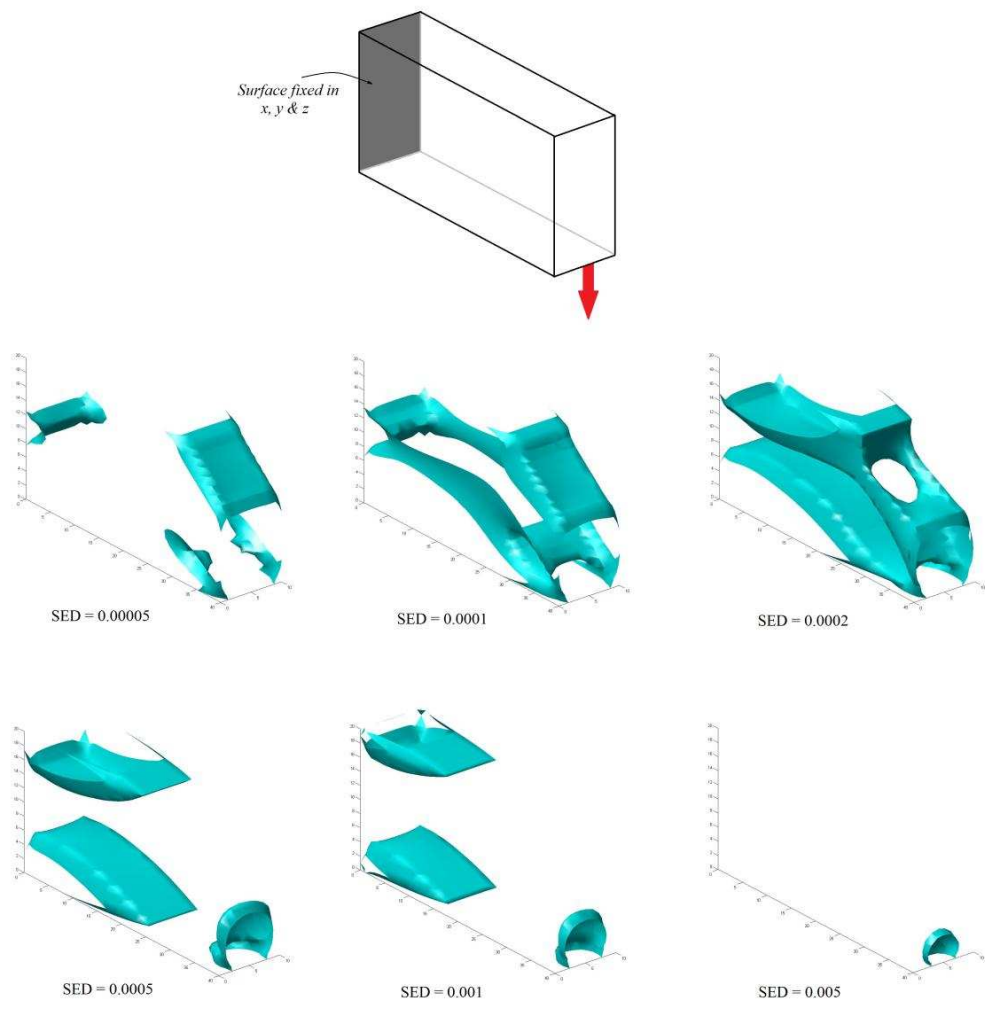


Figure 5.1. Illustration of isosurfaces of SED for a 3D cantilever beam.

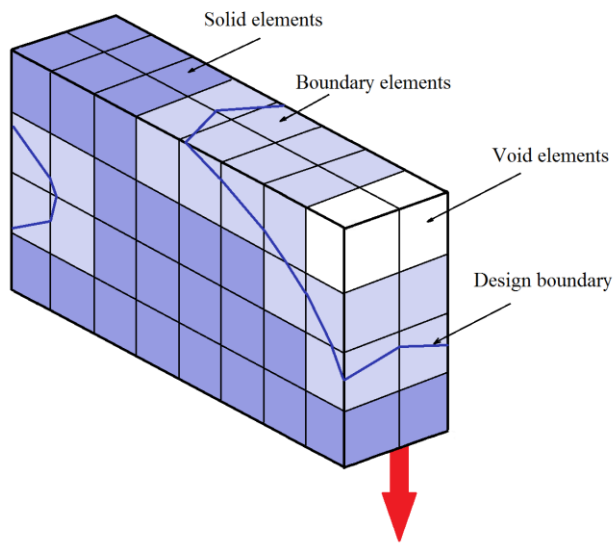


Figure 5.2. Classification of hexahedral elements in isosurface design approach.

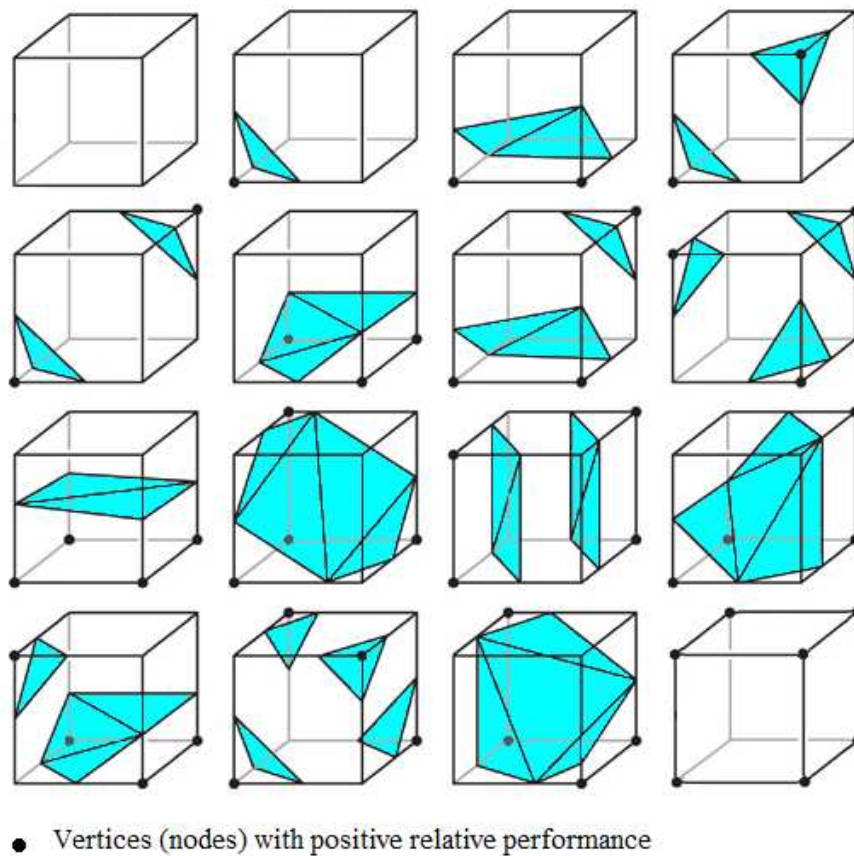


Figure 5.3. 16 different hexahedral element topology states according to marching cubes algorithm.

By superimposing the design geometry onto a fixed grid hexahedral mesh, the finite elements can be classified into solid, boundary and void hexahedral

elements, as illustrated in figure 5.2. If the value of relative performance on all 8 vertices (nodes) of the hexahedral element is greater than zero, the element is classified as a solid element. If the relative performance of all 8 nodes are less than zero the element is classified as a void element. If there is at least one node with positive relative performance and one with negative relative performance, the element is considered to be a boundary element. In this case, the topology of the boundary element is dependent on the sign of relative performance in the 8 vertices of the hexahedron. Assuming linear shape functions for the hexahedral element, the shape of the boundary element (solid sub-domain of the element) can be found by determining the intersection points of the boundary and the hexahedron edges using linear interpolation of the shape functions and relative performance,  $\alpha$  (similar to the process used for the quadrilateral element in the previous chapter).

Since each of the eight nodes of a hexahedral element can have either a positive or a negative value of relative performance ( $+\alpha/-\alpha$ ), there exists a maximum of 256 ( $2^8$ ) scenarios for a hexahedral element topology. Considering reflective and rotational symmetry, as considered in the marching cubes algorithm (Newman & Yi 2006), the total number of hexahedral element topologies can be reduced to 16 (figure 5.3). Once the topologic state and shape of a boundary element is found, the properties of the element can be calculated by performing the integrations over the solid sub-domain of the element using the 3D X-FEM integration scheme presented in the next section.

## 5.2. 3D X-FEM approach

The X-FEM integration scheme presented in the last chapter for representing the material-void discontinuity inside 2D elements can be extended to a 3D design domain. In this case, the stiffness matrix of a 3D boundary element is defined by:

$$k_e = \int_{\Omega} B^T C H(x) B d\Omega \quad (5.1)$$

where the Heaviside function  $H(x)$  is used to allow integration to be performed only on the solid sub-domain of the hexahedron as

$$k_e = \int_{\Omega_S} B^T C B d\Omega \quad (5.2)$$



where  $\Omega_s$  denotes the solid sub-domain of the element. The idea is to partition the solid sub-domain of the boundary elements into sub-tetrahedrons and numerically perform the integration over the solid tetrahedrons. The quadrature rule on tetrahedral has the following form:

$$\iiint_{\Omega_e} F(x, y, z) dx dy dz = V_{th} \sum_{i=1}^m w_i F(\xi_1^i, \xi_2^i, \xi_3^i, \xi_4^i) \quad (5.3)$$

where  $V_{th}$  denotes the volume of the tetrahedron and  $(\xi_1^i, \xi_2^i, \xi_3^i, \xi_4^i)$  are the natural coordinates of the gauss points inside the tetrahedron. In the case of linear third order shape functions, as in 8 node hexahedral elements, at least 4 gauss points are required for accurate integration over a tetrahedron. The element stiffness matrix can be calculated by summing the contributions of all solid sub-tetrahedra

$$k_e = \sum_{i=1}^n V_i \sum_{j=1}^{m_i} w_j f(\xi_1^j, \xi_2^j, \xi_3^j, \xi_4^j) \quad (5.4)$$

with  $V_i$  the volume of tetrahedron  $i$  and  $f = B^T C B$ . In this study, in order to simplify decomposition of the solid region into sub-tetrahedrons and avoid using the marching cubes algorithm, an alternative approach is used. In this approach, the hexahedral boundary element is initially partitioned into 24 sub-tetrahedra by dividing each of the 6 surfaces of the hexahedron into 4 triangles and connecting the vertices of the triangles to the centroid of the hexahedron. The void tetrahedra are then removed from the integration domain. The bi-material (solid-void) tetrahedra are again partitioned into smaller sub-tetrahedra and the void tetrahedra removed from integration domain. The integration is eventually performed over the remaining solid tetrahedra (figure 5.4).

Compared to a normal hexahedron element which can be represented with only 8 gauss points, the proposed X-FEM integration scheme requires a higher number of gauss points. For example, if a fully solid element is divided into sub-tetrahedra using the proposed decomposition scheme, 96 gauss points are required to find the elements properties and, dependent on the topology of a boundary element, more or fewer gauss points might be required. In order to increase the computational efficiency a modified integration scheme is presented, as discussed below.

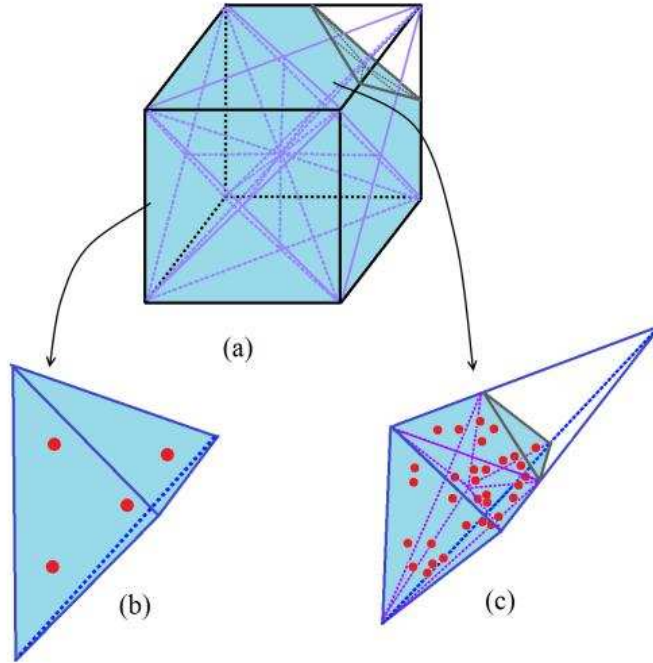


Figure 5.4. Decomposing a boundary hexahedral element: (a) a boundary hexahedral element where the whole element domain is partitioned into 24 sub-tetrahedra; (b) a solid sub-tetrahedron; (c) a boundary sub-tetrahedron where the solid sub-domain is further partitioned into sub-tetrahedrons.

As mentioned earlier, the void region of the design domain is modelled with a very weak material, rather than complete removal from the FE calculations. This is done to avoid singularity. This assumption is also considered in the proposed X-FEM scheme. However, there is no need to perform a separate integration for the void region when the stiffness matrix of a fully solid element as well as the contribution of the solid region have already been obtained, as it can be easily calculated from the other two. Also in the case of the 3D X-FEM scheme, the increase in the number of gauss points can significantly increase the computational cost. To reduce this effect, when the volume ratio of the solid part is more than 50%, we perform the integrations on the void part of the element which has a smaller volume and can thus be represented with fewer sub-tetrahedra and gauss points. The stiffness of the boundary element can then be obtained by subtracting it from the stiffness of the fully solid element. So, if the solid region of the element is smaller than its void region, the stiffness of the element is given by:

$$k_e = k_e^s + (k_e^0 - k_e^s) \times \frac{E_v}{E_s} \quad (5.5)$$

otherwise it can be obtained from

$$k_e = k_e^0 - k_e^v \times \frac{E_s}{E_v} \quad (5.6)$$

where  $k_e^0$  is the stiffness matrix of the fully solid element and  $k_e^s$  and  $k_e^v$  are the stiffness contributions of the solid part and void part of the boundary element, respectively.  $E_s$  and  $E_v$  are the Young modulus of the solid material and void (weak) material respectively.

### 5.3. Evolutionary optimization algorithm

The same evolutionary optimization algorithm presented in the previous chapter for the optimization of 2D structures (section 4.5.3) could be integrated with isosurface design representation and X-FEM in order to extend the Iso-XFEM method into 3D. However in this chapter a new evolutionary optimization algorithm is presented which allows further extension of the method into the optimization of alternative objectives.

Similar to the method presented in the previous chapter, the idea is to identify a criterion for structural performance (SP) and define a minimum level of performance (MLP) where its intersection with the structural performance can be used as the evolving boundary during the optimization process. Then systematically increase the MLP until the desired volume fraction or the convergence conditions are achieved.

As shown in section 4.2, the relative performance can be calculated on any part of the design domain from the nodal values of structural performance (SP) and the minimum level of performance (MLP) using equations 4.1 and 4.2. However, in order to stabilize the evolutionary process, as suggested by Huang and Xie (2007a), we use historical information of the structural performance by averaging the current nodal values of structural performance with those from the previous iteration:

$$SP_i = \frac{SP_i^{it} + SP_i^{it-1}}{2} \quad (5.7)$$

where  $i$  denotes the node number and it is the iteration number. Therefore the MLP in every iteration is calculated for the modified SP of that iteration. By varying the MLP, the volume as well as the shape and topology of the structure can change. The proposed evolutionary optimization algorithm requires the target volume of the design for the current iteration to be calculated before any region is added to or removed from the structure. The target volume of the design for the current iteration is given by

$$V_{it} = \max(V_{it-1}(1 - ER), V^c) \quad (5.8)$$

where ER is the volume evolution rate and  $V^c$  is the specified volume constraint. Then the minimum level of performance which gives target volume of the iteration needs to be identified. An effective way of finding this is to start with an initial guess, for example  $MLP_{it} = ER * \max(SP)$  in the first iteration of the optimization process or  $MLP_{it} = MLP_{it-1}$  for the rest of the iterations. Then the MLP can be modified by

$$MLP_{it}^{k+1} = MLP_{it}^k \times V^k / V_{it} \quad (5.9)$$

until the difference between  $V^k$ , the volume obtained for  $MLP_{it}^k$ , and  $V_{it}$ , the target volume for the current iteration using equation 5.8, is less than a minimum value. Note that since there is no matrix operation in the calculation of  $V^k$  in the above equation, the computational cost of calculating MLP from the given volume is very small. Material removal based on equation 5.8 is carried out until the volume constraint is satisfied. Then the optimization process continues with a constant volume ( $V^c$ ) until other convergence conditions are satisfied.

The presented method of evolutionary topology optimization using X-FEM and isovalues of structural performance can be summarized into the following steps:

- 1- Define the design space, non-design domain, material properties, a fixed grid FE mesh, loads and boundary conditions.
- 2- Define the parameters of the optimization algorithm (ER,  $V^c$ ) and choose an appropriate performance criteria (such as SED for stiffness optimization).

- 3- Carry out the extended finite element analysis and calculate the structural performance over the design domain.
- 4- Average the nodal values of structural performance (equation 5.7) to stabilize the evolutionary process.
- 5- Calculate the target volume for the current iteration using equation 5.8.
- 6- Calculate the Minimum Level of Performance (MLP) in which its intersection with the structural performance distribution gives the target volume.
- 7- Calculate the relative performance ( $\alpha$ ) by subtracting the MLP from structural performance (eq. 4.1).
- 8- Extract the boundary of the design from  $\alpha(x, t) = 0$ . Assign solid material properties to the regions with  $\alpha(x, t) > 0$  and void material properties to the regions with  $\alpha(x, t) < 0$
- 9- If the convergence condition is reached, stop the design process, else, go to step 3.

A flowchart showing the proposed optimization method is illustrated in figure 5.5.

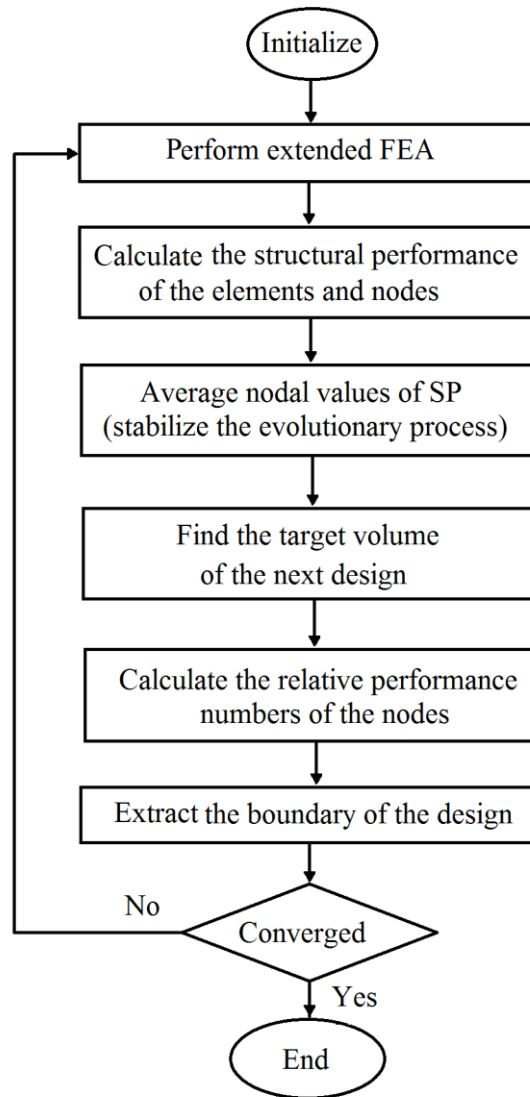


Figure 5.5. Flowchart of the proposed optimization method.

#### 5.4. Test cases

A Matlab code was developed to represent the 3D Iso-XFEM approach. The code includes all the numerical modelling associated with the proposed method, including FEM, X-FEM and isosurface model. Three test cases were considered in this part of the work. Test case 3D-1 was a 3D cantilever plate frequently used in the previous studies (it was the 3D version of test case 2D-2 of the previous chapter). The examination of convergence was studied in this case and the need for using a full 3D analysis was investigated. The second test-case, 3D-2, was the C clip previously studied in chapter 4. A comparison of the 2D and 3D solutions

obtained using the proposed algorithm of this chapter is presented. The third test-case, 3D-3, was an aerospace swing arm representing a real-life structural optimization problem. The initial design domain of this test case was more complex than the previous benchmark problems. Also, a non-design domain was included, which is usually the case in the topology optimization of real structures. Therefore, due to the complexity of the initial geometries, the FE models of the structures were created in commercial FE package, Abaqus, and then imported into Matlab in terms of Nodes and Elements matrices. All the results were generated using a desktop computer with an Intel Xeon 2 processor of 2.4 GHz speed and 24 GB RAM.

#### **5.4.1. Test case 3D-1: 3D cantilever plate**

Test case 3D-1, was the 3D version of test case 2D-2 of the previous chapter. The purpose of this experiment was to apply the new optimization algorithm, proposed in this chapter, to a 3D benchmark problem, in order to investigate the convergence of the 3D Iso-XFEM algorithm, and also, to enable comparing 2D and 3D solutions. The 3D cantilever plate was subjected to a unit distributed force applied to the bottom of the free end. Again, a consistent dimensionless set of parameters was used for this test case. The size of the beam was  $40 \times 20 \times 2$  and a mesh of  $40 \times 20 \times 2$  hexahedral elements was used for the FE model of the beam. The material properties of the solid material were Young's modulus  $E=1$  and Poisson's ratio  $\nu = 0.3$ . The objective was to minimize the total strain energy for a target volume of 50% of the initial design domain. The optimization parameter used was a volume evolution rate of  $ER = 0.02$ .

Figure (5.7) shows the converged solution of the stiffness optimization problem of the cantilever plate, represented by hexahedral elements inside the structure and tetrahedral X-FEM integration domains near the boundary. Figure (5.8) shows the evolution history of the objective function and volume fraction. It can be seen that the method has good convergence and numerical stability.

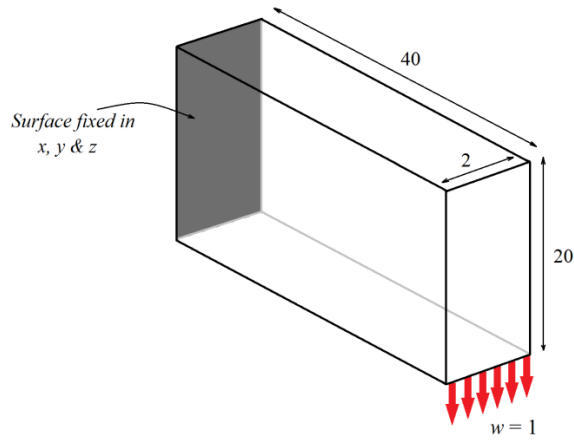


Figure 5.6. Test case 3D-1: a 3D cantilever plate with unit distributed force.

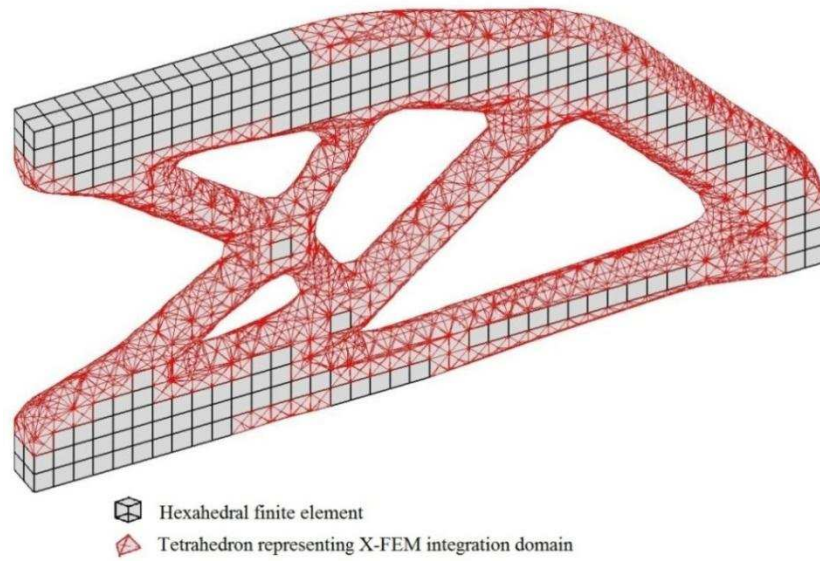


Figure 5.7. Topology optimization solution of the 3D cantilever plate.

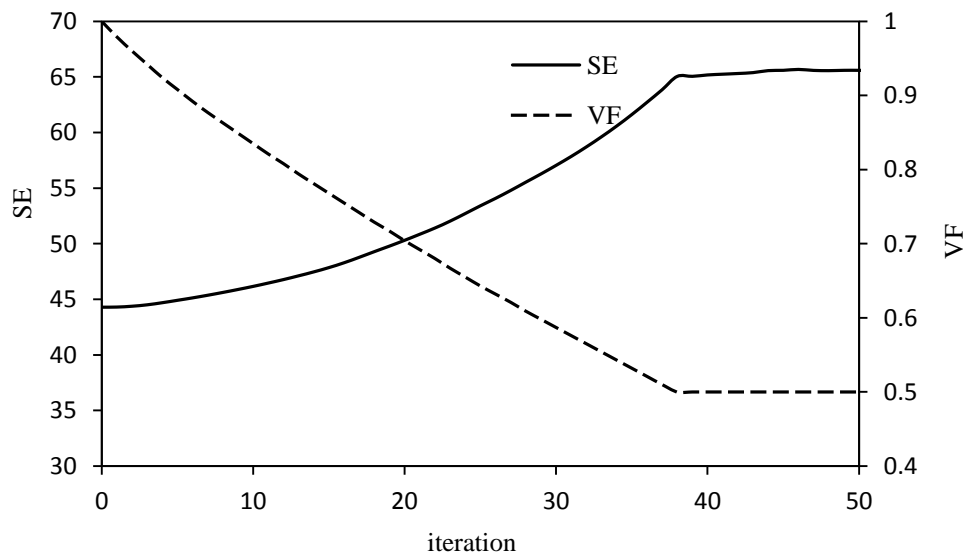


Figure 5.8. Iteration history of strain energy (SE) and volume fraction (VF) of the 3D cantilever plate.



A simple experiment was performed to determine the effect of extending the cantilever plate analysis to 3D. For this purpose, the same structure as test case 3D-1 was considered and only the thickness, as well as the number of elements in thickness direction was changed. The design domain was discretized with unit size hexahedral elements. The 3D optimized design topologies as a function of thickness were then compared to the design identified using a 2D analysis (figure 5.9). Figure 5.10 shows the 3D topology optimization solutions for the cantilever plate problem with thickness of 1, 2, 4 and 6. The equivalent result for 2D would be the result shown in figure 5.9 extended to the appropriate thickness. It can be seen in figure 5.10, however that, beyond a thickness of 2, the final topologies are dependent on the thickness of the structure. When the thickness of the structure is small compared to its other two dimensions, i.e. a plane stress condition, the converged solution is close to that obtained by 2D optimization approach. However solutions change significantly by increasing the thickness. Table 5.1 shows the strain energy to volume ratio of the converged solutions of 2D and 3D models. It can be seen that the strain energy to volume ratio of the 3D solution for  $t=1$  is very close to that of the 2D solution, however this decreases (i.e. performance increases) with increasing thickness in the 3D model. This shows that a 3D analysis enables a more optimum solution to be obtained when thickness is beyond the plane stress condition.

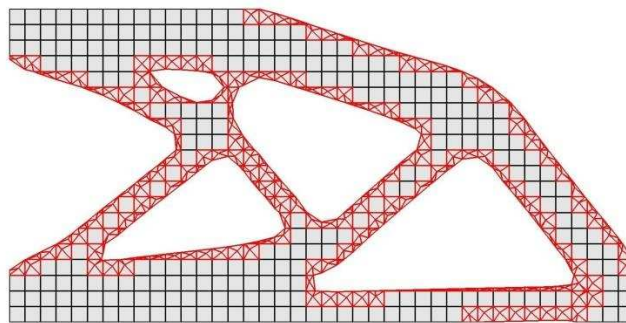


Figure 5.9. Topology optimization solution of 2D cantilever plate.

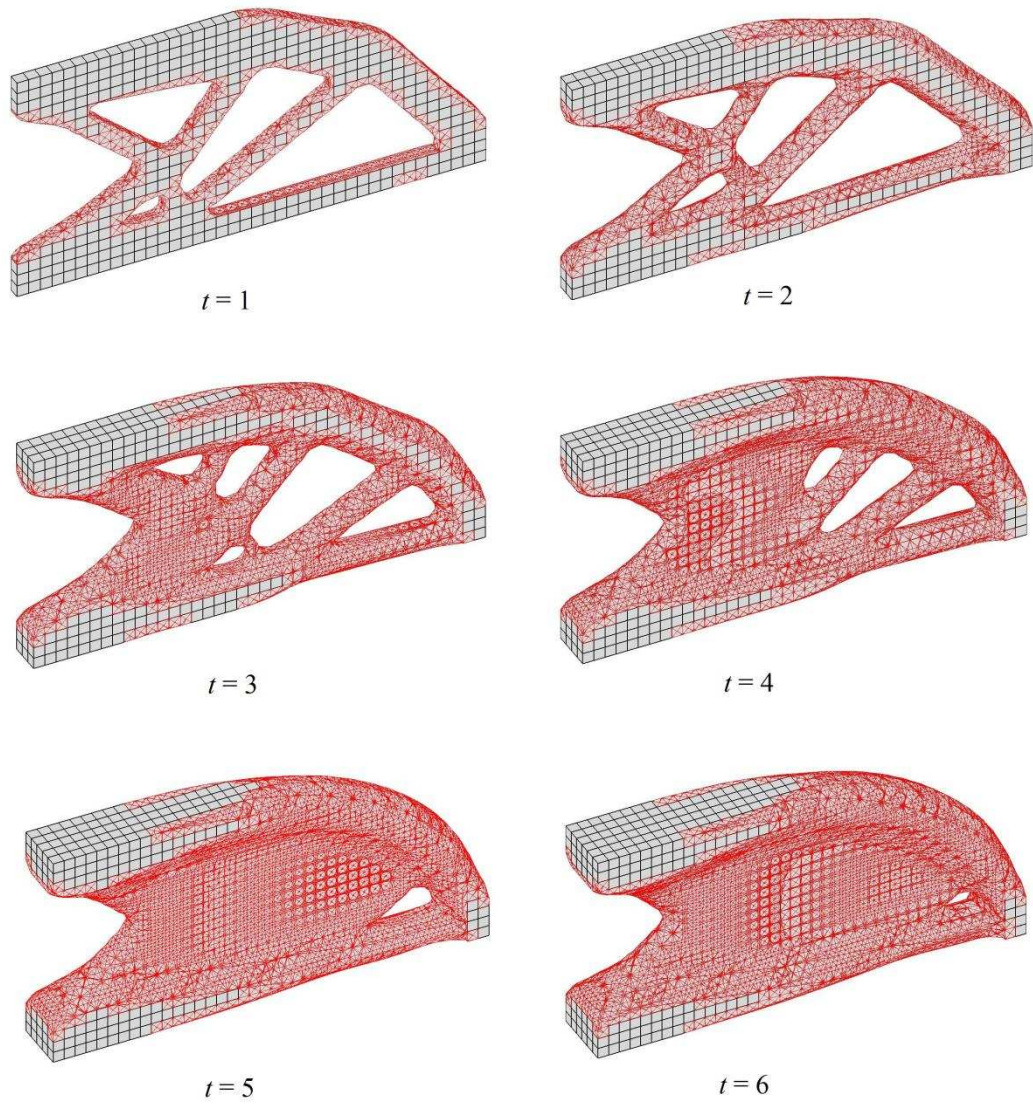


Figure 5.10. Comparison of the solutions of the 3D cantilever plate for different thickness.

Table 5.1. Comparison of the strain energies of 2D and 3D solutions with different thickness.

Thickness	2D	1	2	3	4	5	6
SE/V E-2	8.28	8.25	8.21	8.03	7.85	7.72	7.70

#### 5.4.2. Test case 3D-2: C clip

Test case 3D-2 was the 3D version of the C clip problem, test case 2D-3 presented in the previous chapter (section 4.6.2). The purpose of this set of experiments was to compare the two different evolutionary optimization algorithms proposed in chapters 4 and 5. Also, an experiment was performed to find an appropriate range

for volume evolution rate,  $ER$ , used in the new Iso-XFEM algorithm of this chapter.

The optimization problem was defined as minimizing the strain energy of the clip subject to a volume constraint of 40% of the initial design. The thickness of the structure was 20 mm. The material used had a Young's modulus of 3 GPa and a Poisson's ratio of 0.3. The volume evolution rate used was  $ER = 0.02$ . Both 2D and 3D FE models of the C clip were generated in Abacus and were imported into Matlab in terms of node coordinates and element connectivity matrices.

#### 5.4.2.1. 2D and 3D solutions

The solution obtained using 2D modelling and employing the proposed evolutionary optimization algorithm of chapter 5 is shown in figure 5.11a. It can be seen that the 2D solution is very similar to the one obtained in section 4.6.2 using the proposed algorithm of chapter 4.

The next experiment was to apply the 3D Iso-XFEM method to the same problem. 17540 hexahedral elements were used to generate the FE model of the 3D C clip. All loads and boundary conditions were assumed to be distributed uniformly in the  $z$  direction. Figure 5.11b shows the converged solution obtained from the 3D Iso-XFEM optimization approach. It can be seen that a less complex topology than the optimized 2D structure is obtained by implementing the 3D optimization method. However comparing the total strain energy of the converged solutions which are 1590 N.mm for the 2D design and 1549 N.mm for the 3D design, it can be seen that a higher performance can be achieved by implementing the 3D optimization approach.

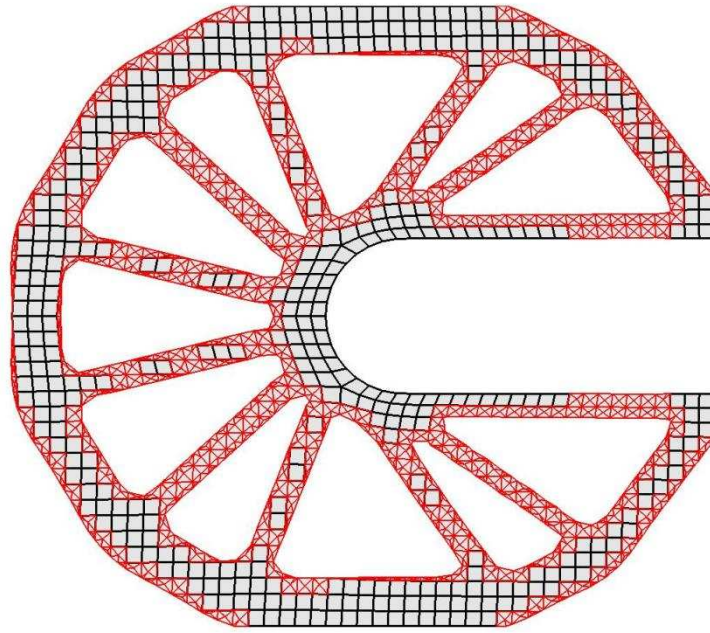
#### 5.4.2.2. Comparison of solutions of the two evolutionary optimization algorithms

The evolutionary optimization algorithm developed in this chapter was different to the one proposed in chapter 4, since in the new algorithm, the material removal is based on the defined volume evolution rate (rather than a performance evolution

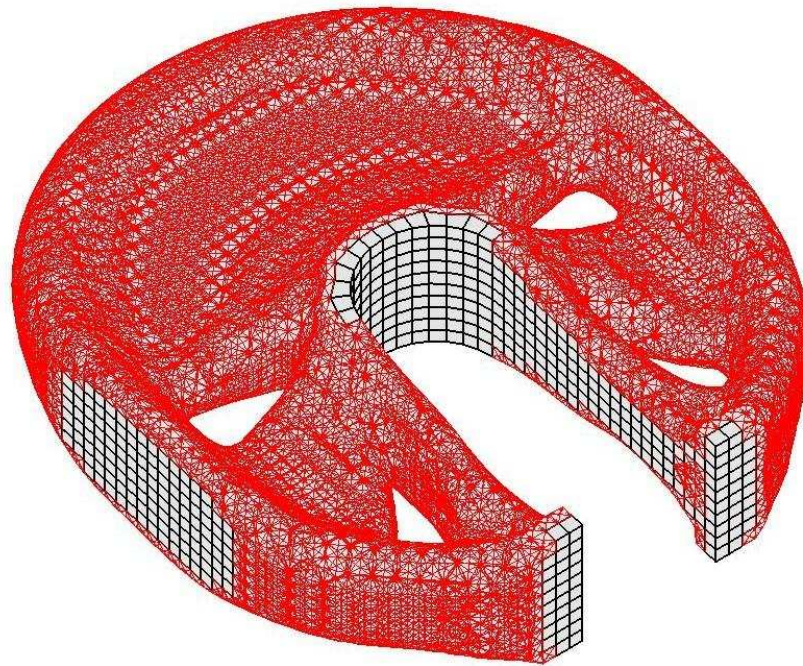
rate as used in the optimization algorithm of chapter 4). Therefore, this allows a better control of material removal than the previously presented evolutionary optimization algorithm. The evolution history of the objective function and volume fraction of the 2D and 3D Iso-XFEM method with optimization algorithm of chapter 5 are shown in figures 5.12a and 5.12b, respectively. It can be seen that in both 2D and 3D models, by gradually removing material from the design domain according to the defined volume evolution rate, the strain energy has smoothly increased. It shows more stability compared to the objective plot in the previous chapter (figure 4.22), although the final 2D optimised topologies are almost the same (figures 5.11a and 4.21). The design was converged after about 55 evolutionary iterations, which took 85 seconds for the 2D optimization approach and 71900 seconds for the 3D version.

#### 5.4.2.3. Effect of volume evolution rate

As discussed in section 5.3, the only optimization parameter that needs to be defined for the proposed Iso-XFEM method with the evolutionary optimization algorithm of chapter 5 is the volume evolution rate, ER. In order to evaluate the sensitivity of the optimization to this parameter, solutions of the 2D C clip topology optimization problem for a range of evolution rate were obtained and the results are shown in Table 1. It can be seen that for  $ER < 0.03$ , almost the same topology (the same number of holes and the same value of SE) is obtained, and within this range, increasing the evolution rate results in convergence at a lower number of iterations. However when  $ER > 0.04$  is applied, different topologies are obtained and the number of iterations to convergence is not dependent on the evolution rate. It is suggested, therefore, that an evolution rate within the range of 0.01-0.02 is a good option for the proposed optimization method and problem. More information regarding the effect of volume evolution rate for BESO can be found in a previous study (Aremu et al 2013).



(a)



(b)

Figure 5.11. Converged solutions of test case 3D-2 for  $VF = 0.4$ : (a) 2D design (b) 3D design.

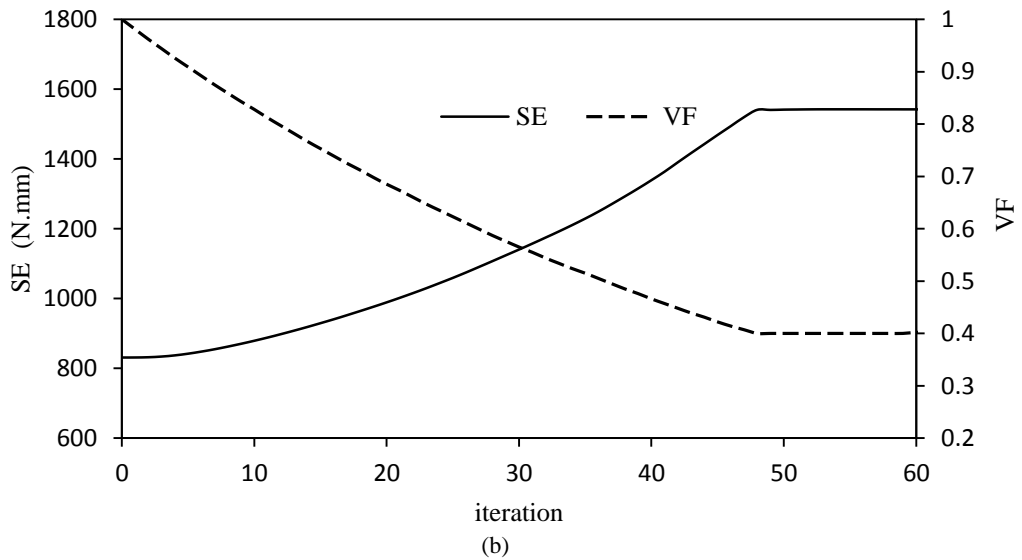
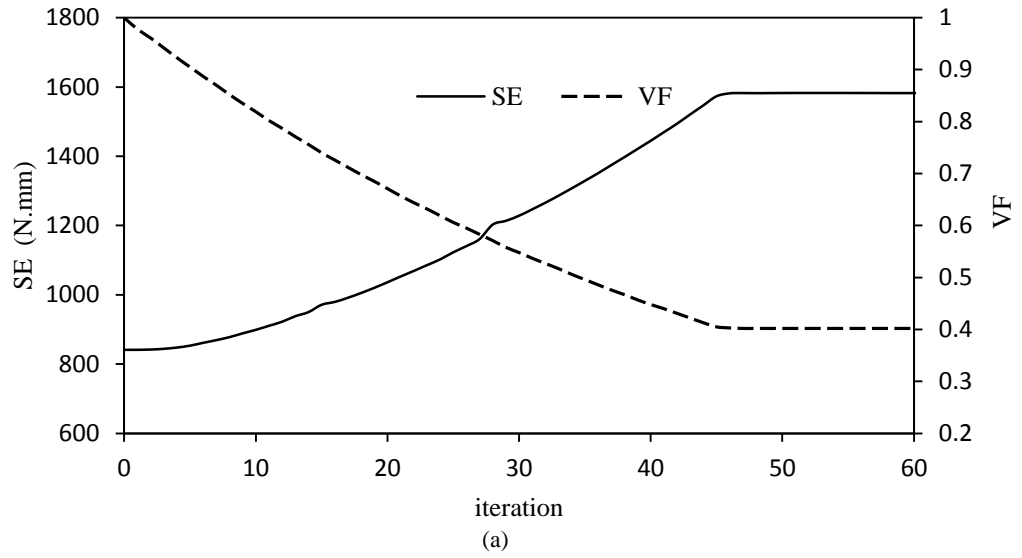


Figure 5.12. Evolution history of the objective function (SE) and volume fraction (VF) of test case 3D-2 for final volume fraction of  $VF^c = 0.4$ . (a) 2D Iso-XFEM (b) 3D Iso-XFEM.

Table 5.2. Comparison of the solutions of the 2D C clip stiffness optimization for a range of volume evolution rate.

ER	0.003	0.006	0.01	0.02	0.03	0.04	0.05	0.06	0.08	0.10	0.12
No. of holes	11	11	11	11	11	9	8	9	9	11	9
No. of iterations	310	160	100	55	42	30	80	60	160	100	64
SE	1592	1590	1590	1590	1590	1605	1624	1676	1614	1590	1615

### 5.4.3. Test case 3D-3: an aerospace arm

In order to implement the Iso-XFEM topology optimization method on a practical problem, the aerospace swing arm shown in figure 5.13, which was previously studied using BESO (Aremu et al, 2013), was considered. This part was used in the first class cabin of an airplane to support and control a monitor. Since there were a number of these parts fitted into the airplane, optimizing these parts would contribute in minimizing the fuel consumption of the airplane.

A load of 667.23 N was uniformly distributed on the lower edge of surface A, and all degrees of freedom on surface B were fixed, forming a cantilever beam. Two levels of difficulty exist in this problem compared to the 2D/3D cantilever problems often used in the literature; an increased geometrical complexity and the existence of a non-design domain. The dark grey cylindrical regions in figure 5.13 were set as non-design domain and the rest of the structure was the design domain. The objective was to minimize the total strain energy subject to a volume constraint of 15% of the design domain. The material properties of the arm were a Young's modulus  $E = 74$  GPa and Poisson's ratio  $\nu = 0.33$ . Two experiments employing different mesh sizes were considered for this test case. In each of the experiments, the optimized solution was obtained using both the proposed Iso-XFEM approach and the element-based BESO method. This was to enable the effectiveness and efficiency of the two approaches to be compared.

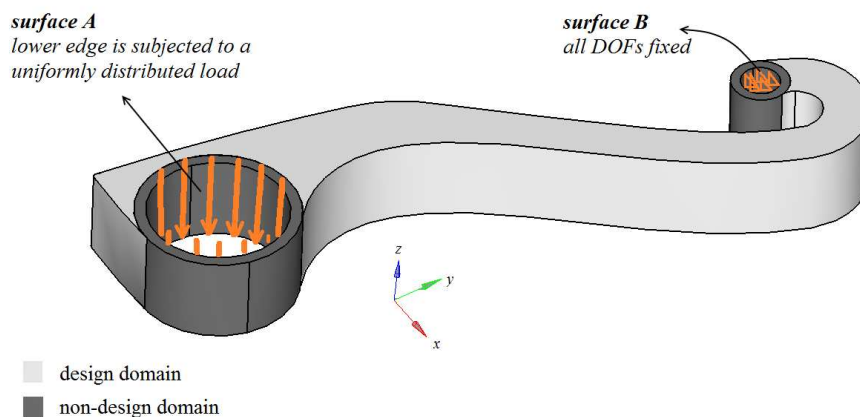


Figure 5.13. Design domain and non-design domain of the aerospace swing arm, loads and boundary conditions.

### 5.4.3.1. Experiment 1

The whole domain was meshed using approximately 22000 hexahedral elements. The volume evolution rate was  $ER = 0.02$ . The resulting topologies during the evolution with Iso-XFEM method for a range of volume fractions up to 0.15 are shown in figure 5.14. It can be seen that by the evolution of the topology up to  $VF=0.2$ , the number of members of the design has increased and then by further material removal up to the volume constraint, some members of the topology have disappeared from the design.

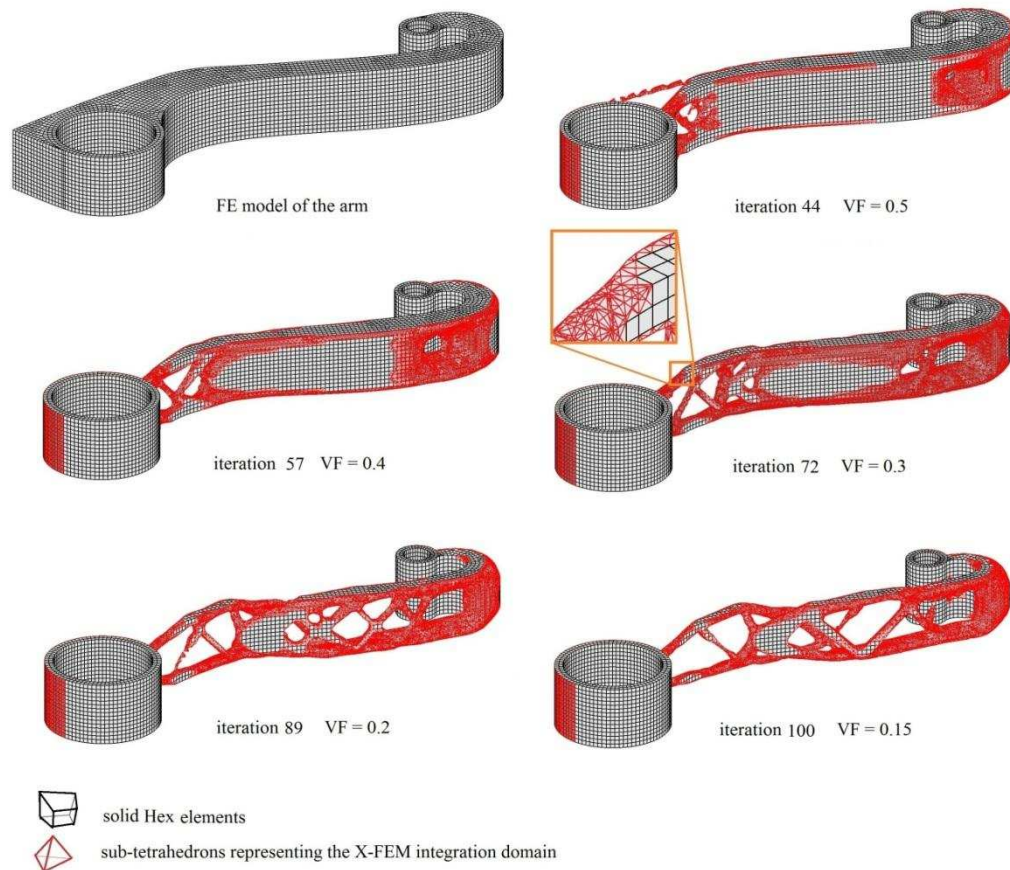


Figure 5.14. The resulting topologies of the aerospace arm for a range of volume fractions obtained using the Iso-XFEM approach.



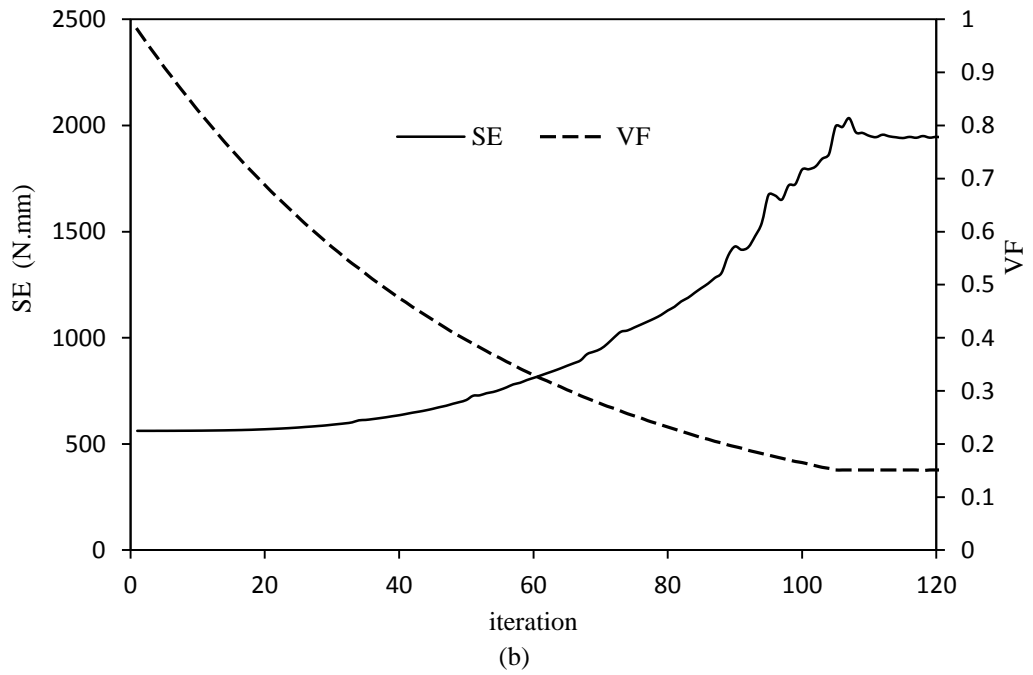
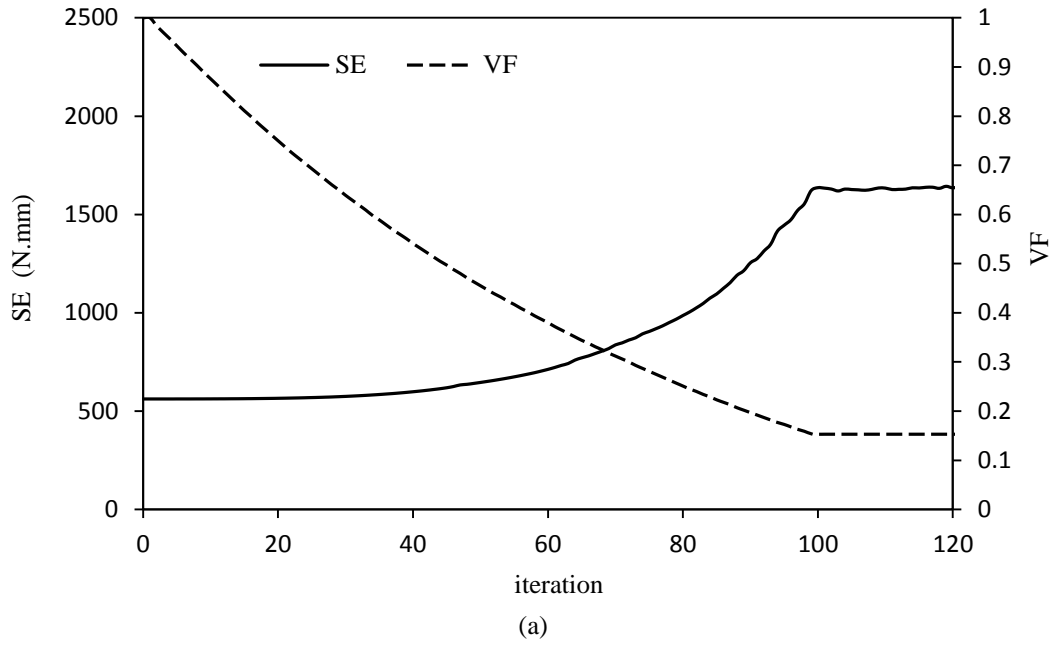


Figure 5.15. Evolution history of the objective function (SE) and volume fraction (VF) of the aerospace arm for the final volume fraction of  $VF^c = 0.15$ , utilizing coarse mesh (experiment 1) (a) applying the Iso-XFEM method and (b) applying the BESO method.

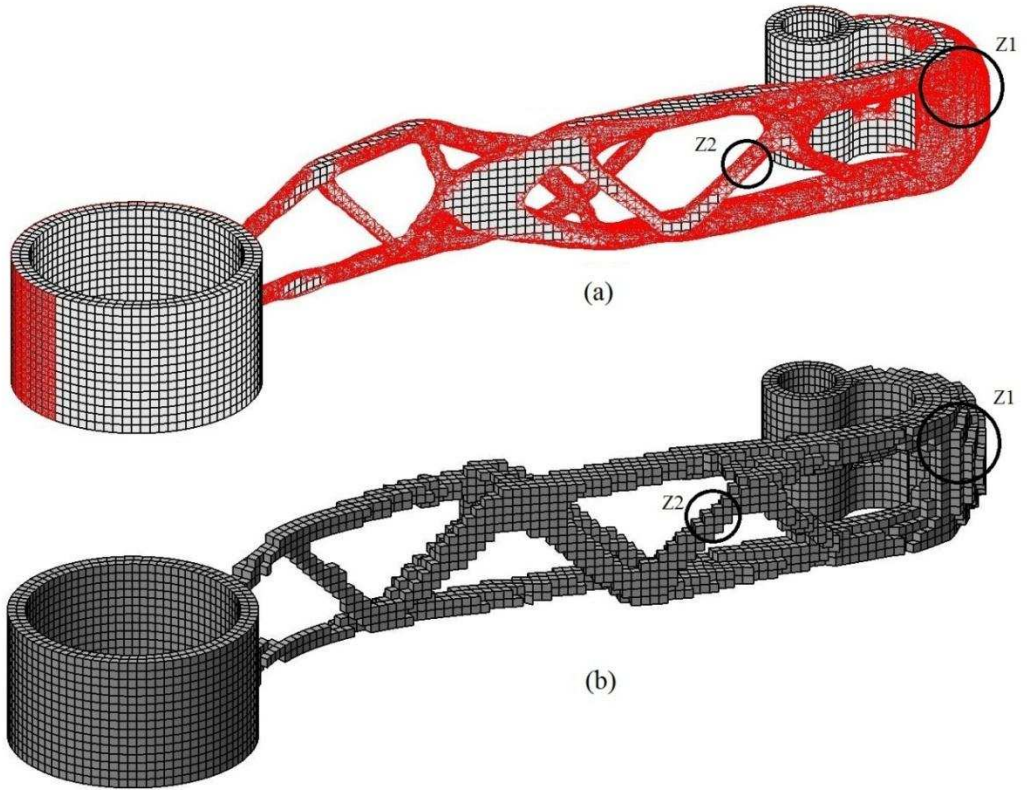


Figure 5.16. Converged solutions of experiment 1: (a) Iso-XFEM solution and (b) BESO solution for  $VF = 0.15$ . Surface roughness was measured in regions Z1 and Z2.

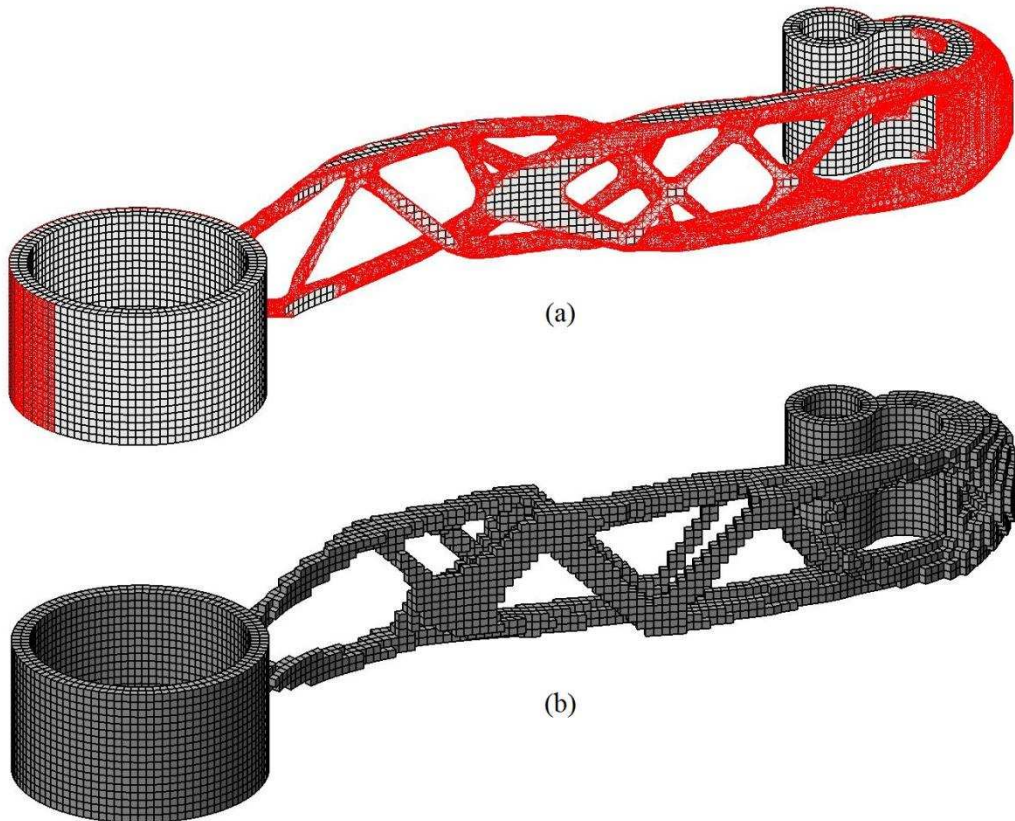


Figure 5.17. Converged solutions in experiment 2: (a) Iso-XFEM solution and (b) BESO solution for  $VF = 0.15$ .

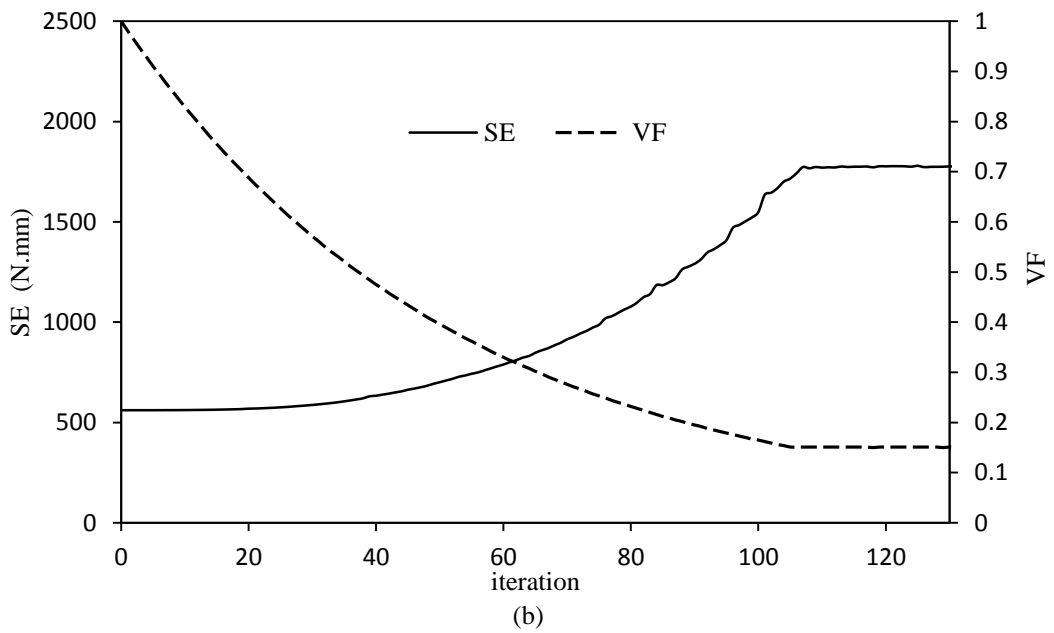
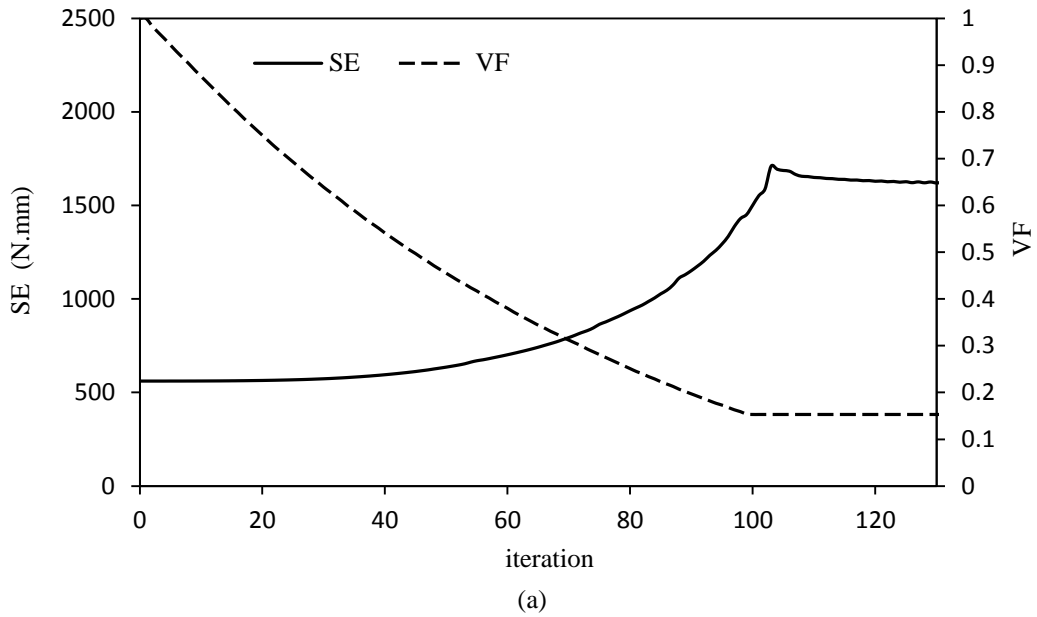


Figure 5.18. Evolution history of the objective function (SE) and volume fraction (VF) of the aerospace arm for the final volume fraction of  $VF^c = 0.15$ , utilizing fine mesh (experiment 2) (a) applying the Iso-XFEM method and (b) applying the BESO method.

Figure 5.15(a) shows the evolution history of the objective function and volume fraction for the case of a final volume fraction of 0.15. It can be seen that the plot of strain energy is very smooth before the volume reaches the volume constraint which is when the material is generally just being removed from the design domain. However slight oscillations can be noticed in the last iterations when the

volume reaches the volume constraint. This can be attributed the fact that at this stage i.e. often the target volume fraction has been reached, both material addition and removal occurs. Similar plots were seen with the other target volume fractions.

Employing the same mesh, the topology optimization problem of the arm was solved using a soft-kill BESO code which was based on Huang and Xie's study (2007a). The BESO parameters used were an evolution rate,  $ER = 0.02$ , and a sensitivity filter radius of 1.2 times the average element size. Figure 5.15(b) illustrates the evolution history of the BESO objective function and volume fraction for  $VF^c = 0.15$ . The converged solutions of the two approaches are shown in figure 5.16. It can be seen that the topologies obtained from the two methods are similar; however the Iso-XFEM approach has converged to a finer solution represented with sub-element resolution. A more detailed comparison is presented in section 5.4.3.3.

#### 5.4.3.2. Experiment 2

In this experiment, the whole domain of the aerospace arm was discretised using approximately 32000 hexahedral elements. The topology optimization of the arm was considered using both the BESO and Iso-XFEM optimization approaches with the same volume evolution rate as were used in experiment 1,  $ER = 0.02$ . Figure 5.17 shows the converged solutions obtained using the two different approaches. It can be seen that the converged solutions in experiment 2 have similar topologies to those seen in experiment 1 (figure 5.16), apart from the fact that topologies with more members are derived when using the finer mesh in experiment 2 (table 5.3). Figure 5.18 shows the plots of evolution of strain energy and volume fraction of the aerospace arm utilizing fine mesh. It can be seen that the plot of strain energy for the Iso-XFEM solution (figure 5.18a) is similar to the previous experiment utilizing coarse mesh (figure 5.15a). However, utilizing fine mesh has resulted the plot of strain energy for BESO solution (figure 5.18b) to become smoother than the one obtained from utilizing coarse mesh in the previous experiment (figure 5.15b).

Table 5.3. Comparison of the solutions in experiment 1 and experiment 2.

		Exp. I	Exp. II	$ \Delta $ %
Number of elements		22000	32000	45%
Time cost of the first 100 iterations (s)	Iso-XFEM	155190	258361	66%
	BESO	78636	155280	97%
Objective of the Solution (Nmm)	Iso-XFEM	1662	1598	3.8%
	BESO	1945	1770	9%
Number of members	Iso-XFEM	21	26	25%
	BESO	15	19	27%
Surface roughness $R_a$ of Z1 (mm)	Iso-XFEM	0.037	0.019	49%
	BESO	0.62	0.42	32%
Surface roughness $R_a$ of Z2 (mm)	Iso-XFEM	0.15	0.12	20%
	BESO	0.65	0.61	6%

Table 5.4. Comparison of the objective values and surface roughness of the solutions of the two approaches at the same time cost of 155 ks per 100 iterations.

Time cost = 155 ks/(100 it)	Iso-XFEM (Exp. I)	BESO (Exp. II)	$ \Delta $ %
No. of elements	22000	32000	45%
Objective (Nmm)	1662	1770	6.5%
$R_a$ of Z1 (mm)	0.037	0.422	1140%
$R_a$ of Z2 (mm)	0.15	0.61	306%

#### 5.4.3.3. Comparison and discussion

Considering experiment 1, it can be seen from figure 5.15 that both X-FEM based and BESO approaches successfully obtained converged solutions for the topology optimization problem of the aerospace arm as an example of a real structure. It can be seen in figure 5.16 that the overall topologies of the converged solutions look similar; however the Iso-XFEM solution has a much smoother surface than the BESO solution. Measuring the surface roughness of the solutions is not very easy as it varies depending on the geometrical features forming the topology. Assuming that spherical and cylindrical features have higher values of surface roughness than flat features, two regions Z1 (hemispherical) and Z2 (cylindrical) in figure 5.16 for experiment 1 (and similar regions in figure 5.17 for experiment 2) were considered and their arithmetic mean surface roughness,  $R_a$ , was calculated, as shown in table 5.3. It can be seen that in both regions, much higher values of surface roughness were obtained with the BESO solution. The jagged edges of the coarse finite elements on the boundary of the BESO solution (figure 5.16b) resulted in an unfeasible design boundary which needs further interpretation of the topology and

post processing such as smoothing, reanalysing and shape optimization before manufacture.

Comparing the strain energies of the BESO and Iso-XFEM solutions in table 5.3, it can be seen that the Iso-XFEM approach has resulted in a performance for the converged solution approximately 18% higher than the BESO solution when compared with the same mesh size. It can also be seen that the BESO solution has fewer members (less complexity) than the Iso-XFEM solution (table 5.3). A reason for this could be the sensitivity filtering scheme (Sigmond and Petersson, 1998) that is employed with the BESO method (and many other element-based methods). This scheme is used to eliminate checkerboard pattern problems, however this filtration scheme is not required for the proposed Iso-XFEM method, although it can be used if solutions with less complexity are desired, e.g. to simplify manufacturing by traditional methods. Comparing the evolution history of both approaches (figures 5.15 and 5.18), we can see that the general trend is the same. However, the plots of strain energy for Iso-XFEM (figures 5.15a & 5.18a) are smoother than for BESO (figures 5.15b & 5.18b), demonstrating a higher stability for the global isovalue based material removal than the element based material removal in BESO.

Comparing the time cost of the two optimization approaches in experiment 1 (table 5.3), one may notice that the Iso-XFEM optimization approach operates slower than the BESO method for the same initial mesh as it takes more time to calculate the properties of boundary elements using the X-FEM integration scheme. However the time spent to solve the finite element linear system of equations will be approximately the same, as the same number of degrees of freedom exists in both methods. Therefore it is expected that by increasing the mesh density which results in an increase in the number of degrees of freedom, the percentage difference between the solution time of the two approaches decreases. This issue was investigated by performing experiment 2 in which the design domain was discretised with a finer mesh than experiment 1.

It can be seen that by using a finer mesh in experiment 2, the time cost of both optimization approaches is increased and the objective values decrease (table 5.3). Comparing the effect of mesh refinement on the optimization's time cost and the

solutions' objective of the two approaches, one can see that the Iso-XFEM optimization approach is less dependent on the mesh refinement as it has lower percentage differences of time cost and objective in the two experiments. Also the comparison shows that when a finer mesh is employed, the time cost of the Iso-XFEM optimization approach becomes closer to the BESO optimization approach.

It can be seen that the time cost of BESO in experiment 2 is almost equal to the time cost of Iso-XFEM in experiment 1 (approximately 155 ks), thus enabling a comparison of BESO and Iso-XFEM performance at a normalised computational time cost. Using the values obtained for this time cost, table 5.4 compares the objectives and surface roughness of Iso-XFEM and BESO solutions for the same solution time. From table 5.4 it can be seen that even when employing the lower number of elements for the Iso-XFEM approach, the objective is still lower than that achieved with BESO demonstrating the higher performance of the Iso-XFEM solutions at the same time cost. Again for the same solution time, it can be seen that the Iso-XFEM solutions are represented with significantly smoother boundaries than the BESO solutions.

## **5.5. Application of Iso-XFEM to problems with alternative objectives or constraints**

Application of the evolutionary based topology optimization methods is not just limited to minimizing compliance and dealing with linear structures. Over the past few years, there have been many studies regarding the application of ESO/BESO methods to a variety of problems and objective functions including (but not limited to) minimizing volume subject to a displacement constraint (Huang and Xie 2010a), topology design of compliant mechanisms (Ansola et al 2007), maximizing natural frequencies (Yang et al 1999; Huang et al 2010), topology optimization of heat conduction problems (Li et al 1999; Gao et al 2008), topology optimization of structures with design dependent loads (Yang et al 2005) and topology optimization of problems involving material and/or geometrical nonlinearities (Huang and Xie 2007b).

Similar to other optimization methods, the evolutionary based methods require an objective function appropriate to the physics of the problem to be defined. In the ESO/ BESO methods, the elemental sensitivity numbers need to be determined by calculating the sensitivity of the objective function to the removal/addition of elements. Then the element removal/addition takes place according to the element sensitivity numbers. A similar procedure can be used in the Iso-XFEM method in order to extend it into optimization of alternative objective functions. In this case, the elemental sensitivity numbers needs to be calculated and the structural performance function can be obtained from elemental sensitivity numbers. The steps required for this extension is summarized below:

- 1- Calculate the element sensitivity numbers and set the elemental values of structural performance equal to them.
- 2- Calculate the nodal values of structural performance from the elemental ones.
- 3- Form the structural performance function using the FE shape functions and nodal values of structural performance.
- 4- Calculate the relative performance by subtracting the structural performance function from the minimum level of performance specified in the evolutionary iterations.

Below is an example of this extension for stiffness design subject to a displacement constraint.

### **5.5.1. Stiffness design with displacement constraint**

Stiffness design problems may include local displacement constraints. If the displacement constraint is not directly related to the overall structural performance i.e. not imposed at the loading DoFs, the stiffness optimization problem could be different to the one defined in section 4.5. Kočvara (1997) studied the topology optimization of truss structures having an additional displacement constraint. Huang and Xie (2010c) extended BESO to stiffness design of continuum structures



with an additional displacement constraint. Considering an additional displacement constraint, the stiffness optimization problem can be defined as:

$$\begin{aligned} \text{minimize: } C &= \frac{1}{2} U^T K U = \frac{1}{2} \sum_{e=1}^n u_e^T k_e u_e & (5.10) \\ \text{subject to: } & \sum_{e=1}^n v_e^s = V^c \\ & u_j \leq u_j^c \end{aligned}$$

where  $u_j$  denotes the  $j^{\text{th}}$  displacement and  $u_j^c$  is its constraint. In order to solve the above multiple constraint optimization problem using an evolutionary optimization algorithm, the displacement constraint can be added to the objective function using a Lagrangian multiplier  $\lambda$  (Huang and Xie 2010c):

$$f(x) = \sum_{e=1}^n u_e^T k_e u_e + \lambda(u_j - u_j^c) \quad (5.11)$$

The Lagrangian multiplier is used to compromise between the objective function and the displacement constraint. If the displacement is equal to the constraint, the modified objective function is equivalent to the original one. Otherwise if the displacement is smaller than the constraint,  $\lambda$  takes the value of zero (because the displacement constraint is already satisfied). When the displacement is larger than the constraint,  $\lambda$  tends to infinity to allow the minimization of the displacement, so the constraint can be satisfied in the later iterations. Using the adjoint method, Huang and Xie (2010c) obtained the elemental sensitivities for BESO employing a uniform mesh as

$$SP_e = \frac{1}{2} U_e^T K_e U_e + \lambda U_{ej}^T K_e U_e \quad (5.12)$$

where  $K_e$  and  $U_e$  are the contributions of the element stiffness matrix  $k_e$  and the element displacement vector  $u_e$  in the global stiffness matrix and displacement vector, respectively.  $U_{ej}$  is the virtual displacement vector of element  $e$  resulting from a unit virtual load applied to the degree of freedom  $j$ . In order to apply the Iso-XFEM method to the optimization problem defined with equation 5.10, we can define the elemental structural performance using the values of the elemental sensitivities defined in the above equation as:

$$S_e = (\frac{1}{2} U_e^T K_e U_e + \lambda U_{ej}^T K_e U_e) / v_e \quad (5.13)$$

and find the structural performance function using the elemental values. It can be seen that the structural performance function is dependent on the value of  $\lambda$ , which needs to be calculated for each evolutionary iteration. An effective way of finding an appropriate value for  $\lambda$  is to use an iterative process until all constraints are satisfied, as shown below.

#### 5.5.1.1. Calculating Lagrangian multiplier

We assume that  $\lambda$  is defined by

$$\lambda = \frac{1-q}{q} \quad (5.14)$$

where  $q$  is a constant which is  $0 < q_{min} \leq q \leq 1$ . This allows  $\lambda$  to vary from zero to infinity. Then the following steps can be used to find the appropriate value for  $q$ :

- 1- Set the initial band values of  $q$  as  $q_{lower} = q_{min}$  and  $q_{upper} = 1$  and start with an initial guess of  $q = q_{upper}$ .
- 2- Calculate  $\lambda$ .
- 3- Calculate the structural performance function using equation 5.13 and find an appropriate value of MLP for the target volume using the method discussed in section 5.3.
- 4- Estimate  $j^{th}$  displacement in the next iteration  $u_j^{k+1}$  from  $u_j^{k+1} = u_j^k + \sum_e (U_{ej}^T K_e U_e) \Delta v_e$ .
- 5- If  $u_j^{k+1} < u_j^c$ ,  $q$  is updated as  $q_{new} = \frac{q_{old} + q_{lower}}{2}$  and then the upper band of  $q$  is updated as  $q_{upper} = q_{old}$ . Otherwise  $q$  is updated as  $q_{new} = \frac{q_{old} + q_{upper}}{2}$  and then  $q_{lower} = q_{old}$ .
- 6- Repeat steps 2-5 until the difference between  $q_{lower}$  and  $q_{upper}$  is less than a minimum value.

Assuming  $(q_{upper} - q_{lower}) < 10^{-5}$  as convergence condition for the above iterative process of finding  $\lambda$ , a total number of 17 iterations is required. Assuming that in the estimation of displacement (step 4) the stiffness matrix of a boundary

element is proportional to the volume ratio of the solid part of the element (density based approach), the cost of finding  $\lambda$  can be negligible.

#### 5.5.1.2. 2D example

The 2D simply supported rectangular domain shown in figure 5.19 is considered to demonstrate the procedure described above, i.e. the application of the proposed Iso-XFEM method to the topology optimization of problems with an additional displacement constraint. The reason for choosing this test case was that this structure experiences horizontal displacement at right bottom edge of the structure having roller joint (node A), and the horizontal displacement is not directly related to the overall performance of the structure. Therefore the additional displacement constraint can be defined for node A. A load of 100 N is applied to the middle of the lower edge. The thickness of the structure is assumed to be 1 mm. The objective is to minimize the total strain energy of the structure subject to a volume constraint of 50% of the design domain as well as a maximum horizontal displacement of 1 mm for node A. The material used has a Young's modulus of 1 GPa and Poisson's ratio of 0.3. A mesh of 100x50 was used for the FE model of the structure and a volume evolution rate of 0.01 was considered for the Iso-XFEM optimization approach. Figure 5.20 shows the resultant topologies of the problem without imposing the displacement constraint (17a) compared to the solution obtained after imposing the displacement constraint (17b). Evolution histories of strain energy and constrained displacement (node A) are shown in figure 5.21. It can be seen that in the structure with no horizontal displacement constraint, the displacement of node A gradually increases by removing material from the design domain until convergence. However when the displacement constraint is applied on node A, when removing material from the design domain the displacement increases until it reaches the maximum allowed value. It then remains almost constant as further material is removed. It can be seen from the plots of strain energy that there is an increase in the total strain energy of the structure with the addition of the horizontal displacement constraint, which would be expected.

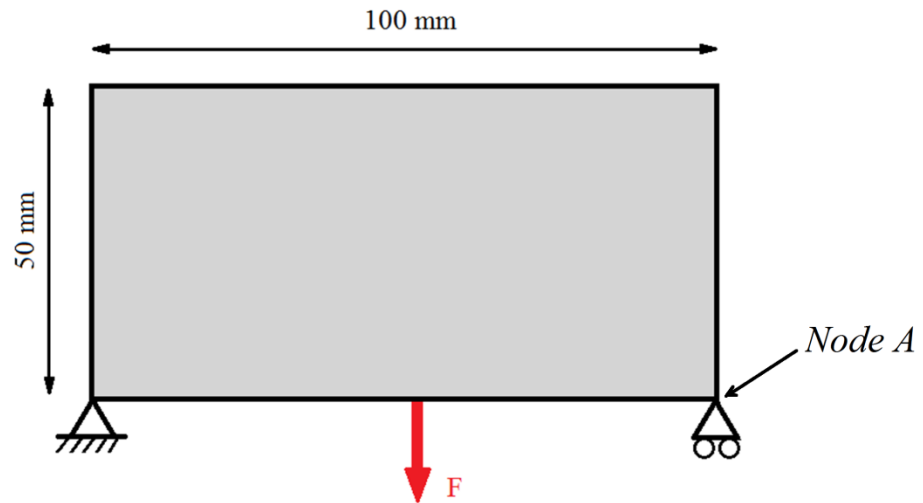
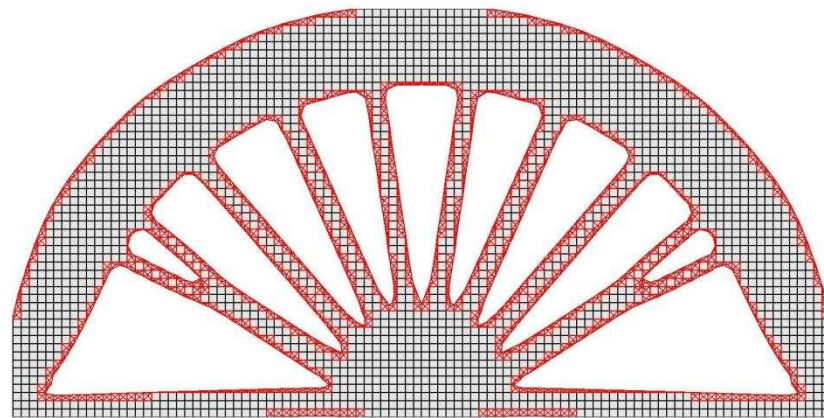
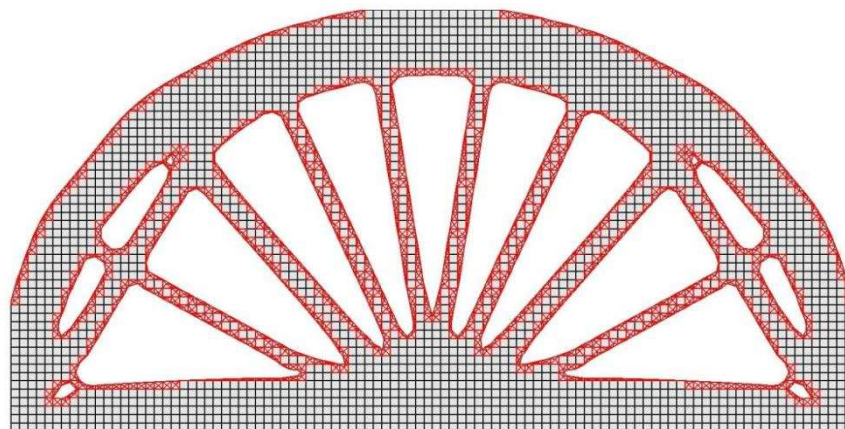


Figure 5.19. Design domain of the 2D optimization problem with horizontal displacement constraint.

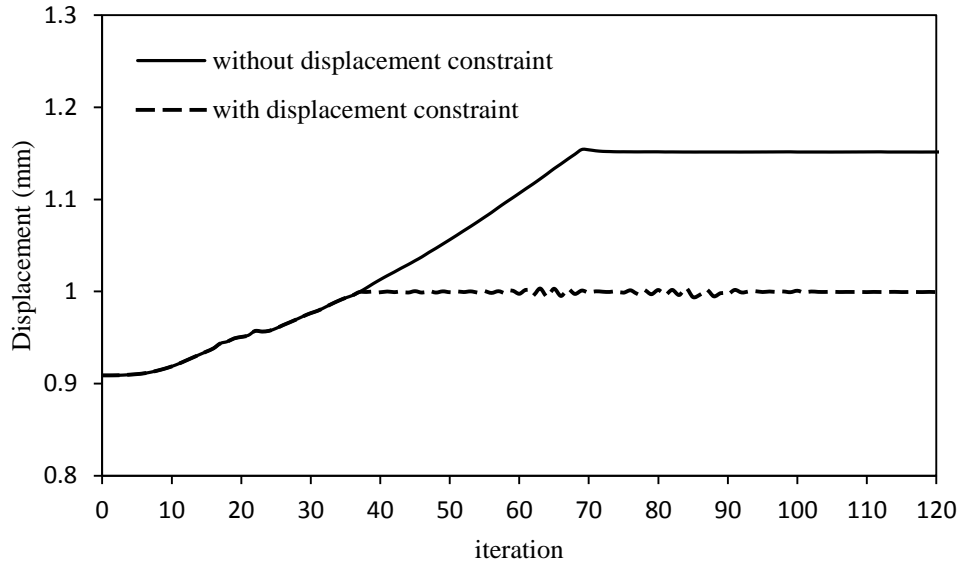


(a)

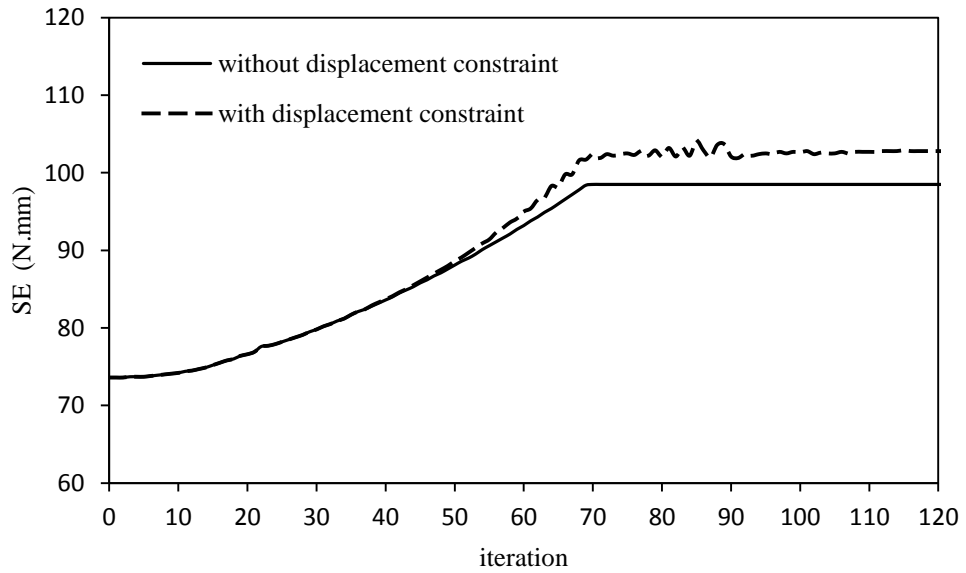


(b)

Figure 5.20. Topology optimized solutions of the simply supported rectangular design domain: (a) stiffness optimization subject to volume constraint only (b) stiffness optimization subject to volume and displacement constraints.



(a)



(b)

Figure 5.21. Evolution histories of (a) and strain energy (b) of the simply supported plate problem, with and without a displacement constraint.

### 5.5.1.3. 3D example

The purpose of this experiment was to investigate the application of the procedure described above to stiffness design of 3D structures with an additional displacement constraint. A new 3D test case shown in figure 5.2.2 was used rather than a 3D version of the previous test case. This was to enable the illustration of

the difference in the optimised topologies with and without a displacement constraint, as they would have been very similar if the previous test case was used (note that the 3D solutions tend to be less complex than the equivalent 2D ones when the thickness is beyond the plane stress condition, as was observed in sections 5.4.1 & 5.4.2). The beam has a size of 100x25x5 mm and a concentrated load of 100 N is applied to the bottom of the free end. The objective is to minimize the total strain energy of the structure subject to a volume constraint of 40% of the design domain as well as maximum horizontal displacement of 1.4 mm for node A. Note that the vertical deflection of the tip is directly related to the objective (strain energy or compliance), but the horizontal displacement of the tip is not related to the objective. Therefore it can be used as an additional constraint of the stiffness optimization problem. The material used has a Young's modulus of 1 GPa and Poisson's ratio of 0.3. A mesh of 100x25x6 elements was used to generate the FE model of the structure and a volume evolution rate of 0.01 was considered for the Iso-XFEM optimization approach.

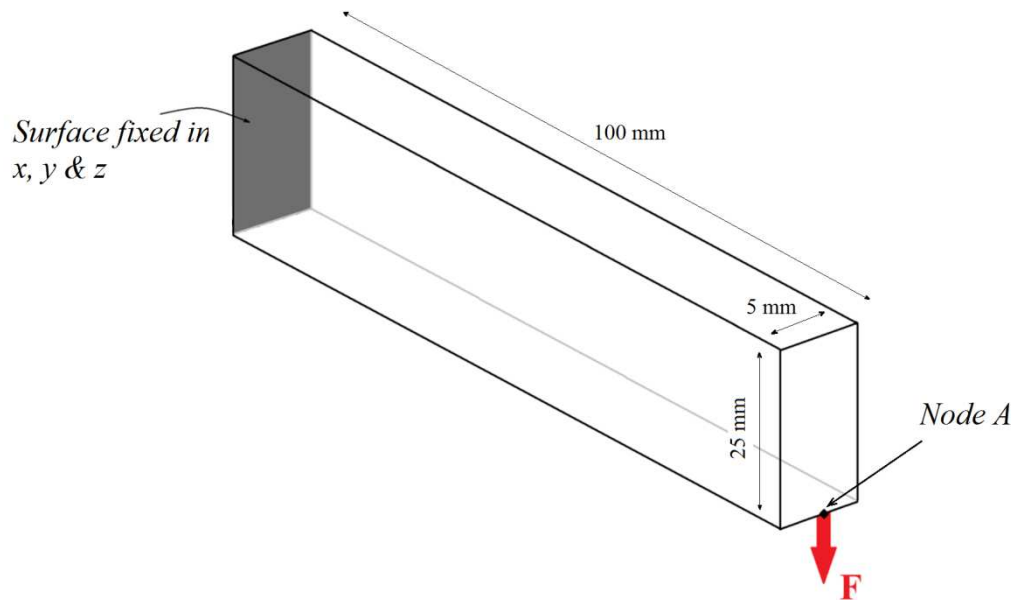
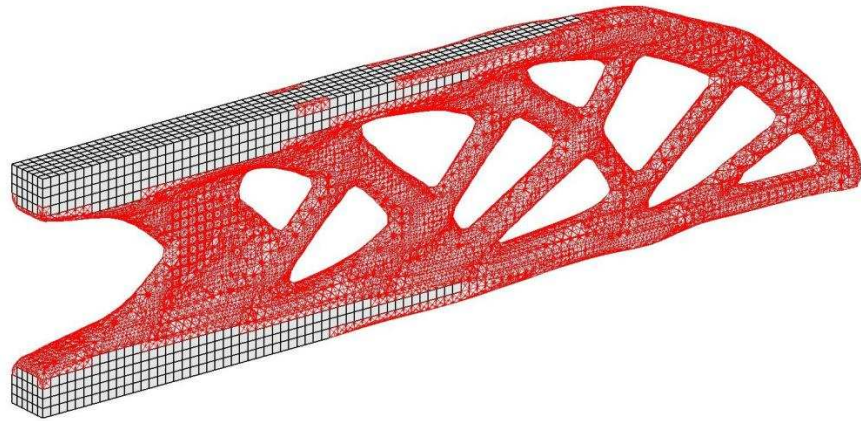
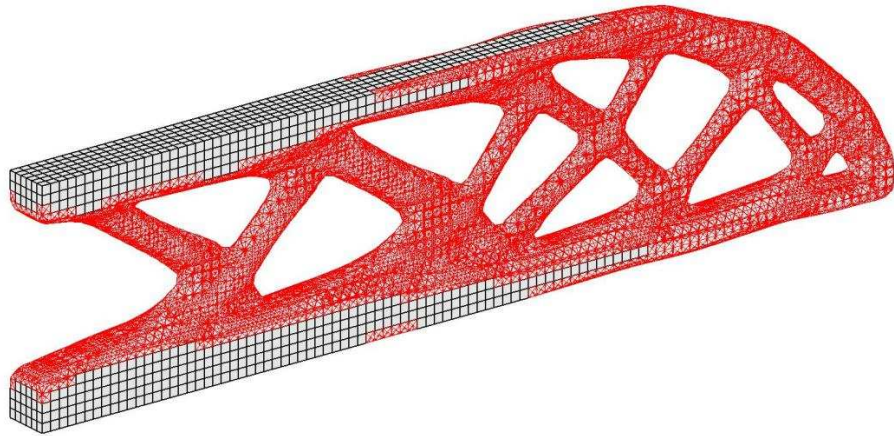


Figure 5.22. Design domain of the 3D optimization problem with horizontal displacement constraint.

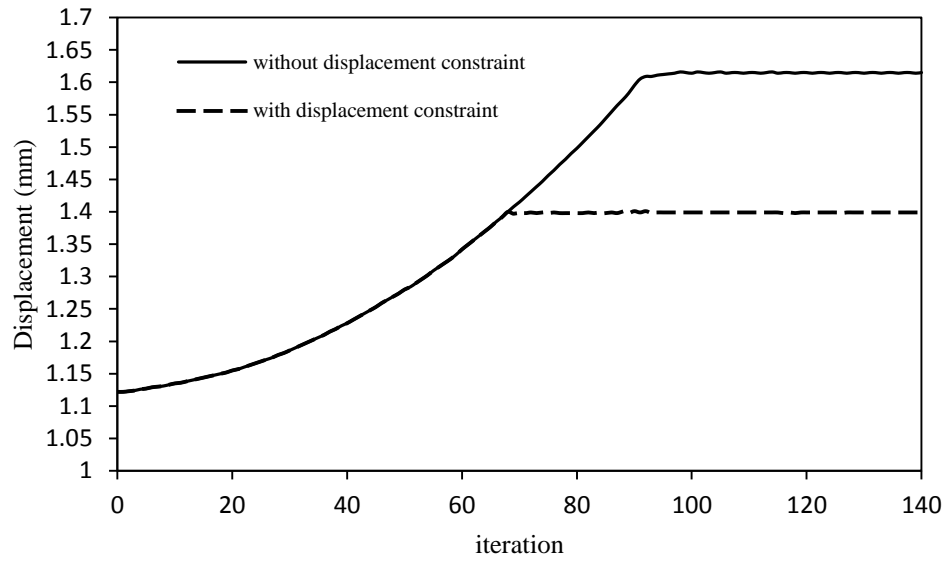


(a)

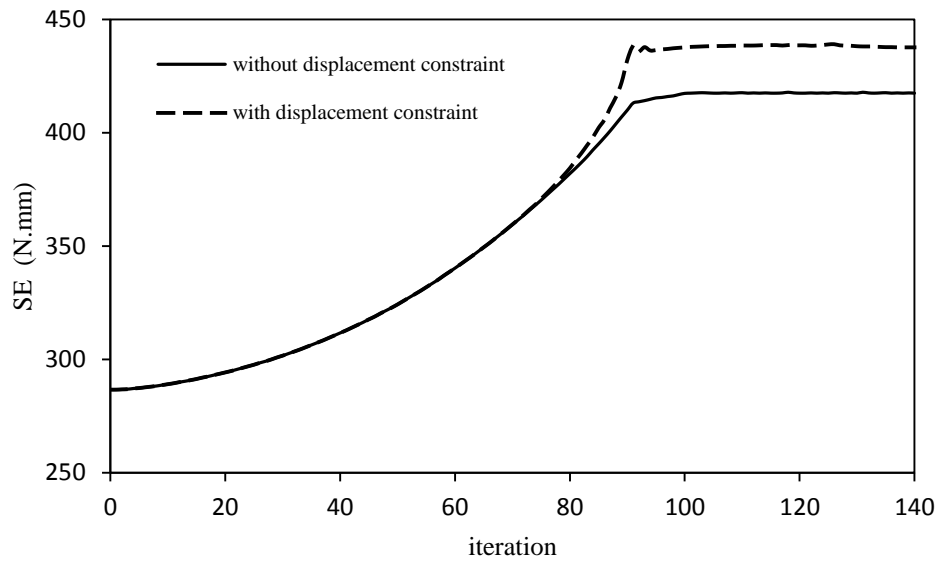


(b)

Figure 5.23. Topology optimization solutions of the 3D cantilever beam: (a) solution of the problem with volume constraint only (b) solution of the problem with both volume and displacement constraint.



(a)



(b)

Figure 5.24. Evolution histories of (a) and strain energy (b) of the 3D cantilever beam problem, with and without displacement constraint.

Figure 5.23 compares the solutions before and after imposing the displacement constraint. It can be seen that a slightly different solution is obtained after imposing the displacement constraint. Figure 5.24 shows the evolution histories of strain energy and constrained displacement. It can be seen that, similar to the previous test case, when the constrained displacement reaches the maximum allowed value, it remains almost constant throughout the rest of the material removal process and that the strain energy of the structure with constrained



displacement is higher than that without displacement constraint. Again, it is to be expected that additional constraints would result in a less optimal solution.

## **5.6. Summary and conclusion**

In this chapter, Iso-XFEM was successfully extended to the topology optimization of 3D design domains. A new evolutionary optimization algorithm was developed and this was shown to have increased stability compared to the one presented in the previous chapter. The new method also allows extension of the method to alternative optimization problems.

It was shown that by enabling optimization in 3D rather than 2D, a different solution with higher performance could be achieved. This indicates the benefit of a full 3D analysis when the thickness to length ratio is large enough that plane stress conditions cannot be assumed.

The efficiency of the Iso-XFEM method was then investigated by applying the method to optimize an aerospace arm, as a real-life problem, and comparing the solutions with BESO solutions of the same problem. It was shown that solutions with much higher resolution than BESO ones can be achieved utilizing the Iso-XFEM method. Although the use of X-FEM in structural optimization of continuum structures can increase the solution time, it is shown that this is more than offset by the benefit of being able to employ a coarse mesh to generate a smooth topology. Therefore, not only are the structural performances of the final solutions higher than the solutions obtained from an evolutionary element-based method, such as BESO, the total time that is required to present a solution with clearly defined boundaries can be greatly reduced.

The possibility of the application of the method to alternative optimization problems was also discussed in this chapter and examples of the application of the method to the stiffness optimization of structures with an additional displacement constraint were presented. It is, therefore, shown that the method has the potential to be implemented on alternative objectives and constraints if an appropriate structural performance criterion can be defined for the problem.

As the focus of this work in the last two chapters was mainly the stiffness optimization of linear 2D and 3D continuum structures using the proposed Iso-XFEM approach, the question may arise whether this method can also be implemented for the topology optimization of nonlinear structures. This will be investigated in the next chapter of the thesis.

# Chapter 6

## Topology Optimization of Geometrically Non-linear Structures

In this chapter the Iso-XFEM is further developed to enable the topology optimization of geometrically non-linear structures undergoing large deformations. This involves extension of the previous linear finite element analysis into non-linear FEA. This is achieved using a total Lagrangian FE formulation and an incremental-iterative Newton-Raphson procedure to determine the equilibrium solution at every evolutionary iteration. A structural performance criterion appropriate for the objective function of the optimization problem is defined and implemented in the Iso-XFEM approach. The Iso-XFEM solutions for geometrically nonlinear test-cases implementing linear and non-linear modelling are compared, and the necessity of the use of non-linear modelling for the topology optimization of geometrically non-linear structures is investigated.

### 6.1. Introduction

The majority of work regarding topology optimization of structures is based on linear modelling of the problems, assuming the structure contains only linear elastic materials and undergoes small displacements. Although this assumption can be effectively applied to a large range of structural design problems, there are still many cases that require non-linear modelling in order to obtain valid solution. Large deformation is a significant source of non-linearity that can be found in many non-linear problems. Examples of such problems include energy absorption structures, compliant mechanisms and any other structure which undergoes large deformation, which can be classified as a “geometrically non-linear structure”.

There have been a number of previous works which considered geometrical non-linearity in topology optimization problems. Jog (1996) used a perimeter method for topology design problems of non-linear thermoelasticity. Bruns and Tortorelli (1998) introduced a Gaussian weighted density measure for solving topology optimization problem of geometrically nonlinear structures and compliant mechanisms. According to Buhl et al (2000), the examples provided in the above mentioned works were not able to clearly show a significant difference in the converged topologies or values of the objective function between linear and non-linear modelling. Buhl et al (2000) coupled SIMP with a non-linear FE formulation to address the topology optimization of geometrically nonlinear problems. With the examples provided, Buhl showed that in many cases, the solutions from the nonlinear modelling are only slightly different from the linear ones. However if snap-through effects are involved in the problems, the difference could be significant. Gea and Luo (2001) proposed a microstructure-based design approach with a nonlinear FE formulation for the topology optimization of structures with geometrical non-linearity. Pedersen et al (2001) considered topology optimization of non-linear compliant mechanisms represented with frame elements. Bruns and Tortorelli (2003) proposed an element removal and reintroduction strategy for topology optimization problems with geometrical nonlinearity. Ha and Cho (2008) and Luo and Tong (2008) developed a level set based topology optimization method for large deformation problems. Huang and Xie (2007c&2008) applied BESO for topology optimization of geometrically nonlinear structures under both force loading and displacement loading.

An important consideration when applying topology optimization techniques to nonlinear structures should be the computational efficiency of the method as the analysis requires much more computation than that of a linear structure. This becomes even more important when applying the method to 3D structures. The other issue which may arise in density based topology optimization approaches such as SIMP is the existence of intermediate densities in the solutions. Because of the large displacements, the tangent stiffness matrix of low density elements may become indefinite or even negatively definite during the optimization process (Buhl et al 2000; Bruns and Tortorelli 2003). To overcome this issue, Bruns and Tortorelli (2003) proposed totally removing low-density elements. Huang and Xie

(2007c&2008) suggested using hard-kill BESO to increase the computational efficiency and avoid issues regarding the existence of intermediate density elements.

The application of the Iso-XFEM to the topology optimization of geometrically non-linear structures could be of significant benefit because of its high computational efficiency and lack of intermediate density elements in the solutions. In the next sections of the chapter, a non-linear modelling strategy for geometrically non-linear structures based on an incremental-iterative Newton-Raphson approach is presented. An appropriate structural performance criterion for stiffness design is derived and the Iso-XFEM method is applied to several test cases.

### **6.1.1. Types of structural non-linearity**

In general, three sources of non-linearity can be specified in structural analysis:

- Geometrical non-linearity: as mentioned earlier, the effect of large deformations on the overall geometric configuration of the structure can cause the structure to respond non-linearly. An example of such a non-linear behaviour is the deformation of a fishing rod.
- Material non-linearity: the material shows nonlinear behaviour when the stress-strain relationships are non-linear. Examples of such material models include non-linear elastic and elastoplastic.
- Boundary conditions: if the boundary conditions are displacement dependant, the response is non-linear. Contact problems are the most frequent type of non-linearity in boundary conditions.

The non-linearity can have several consequences in the analysis of the structures. The structural behaviour becomes non-proportional to the applied load. The principle of superposition cannot be applied, e.g., combining the results of several load cases, and only one load case can be handled at a time. Also, the sequence of the application of load may become important in this case. For example plastic deformation depends on the manner of loading. So it's advised to divide loads into small increments in non-linear structural analysis.

### 6.1.2. Incremental-iterative approach

As discussed earlier, one of the characterizations of a non-linear analysis is that the load deformation behaviour is non-proportional. In this case, the response of the structure to an incremental loading is affected by the loading level and the deformed geometry of the structure. As a result, the stiffness matrix of the structure becomes a function of element force as well as the deflection of the structure. Therefore for large systems of equations, like the FE models of continuum structures, it is necessary to use numerical incremental and/or iterative procedures in order to allow for the geometrical change of the structure. The Newton-Raphson is an iterative approach which can be used in conjunction with an incremental model of loading to find the equilibrium state at each loading increment. In the incremental Newton-Raphson approach, the applied load ( $R$ ) is first divided into a set of smaller load increments. Then starting from the first load increment, using the tangent stiffness matrix ( $K_T$ ), the displacement caused by that force increment is computed. Using the accumulated displacement, the resistant force ( $F$ ) is obtained and the unbalanced force ( ${}^tR - {}^tF$ ), which is the difference between the applied and the resistant forces, is determined. The iterative process at this load increment continues by calculating a new tangent stiffness matrix, finding the displacement and the unbalanced force (figure 6.1). The equations used in the Newton-Raphson method can be stated as (Bathe 1996):

$${}^{t+\Delta t}K_T^{(it-1)}\Delta u^{(it)} = {}^{t+\Delta t}R - {}^{t+\Delta t}F^{(it-1)} \quad (6.1)$$

$${}^{t+\Delta t}u^{(it)} = {}^{t+\Delta t}u^{(it-1)} + \Delta u^{(it)}$$

where  $\Delta t$  is a suitably chosen time increment and it denotes the iteration number of the Newton-Raphson procedure in each time increment. The initial conditions at the start of each time increment are:

$${}^{t+\Delta t}u^{(0)} = {}^t u; \quad {}^{t+\Delta t}K_T^{(0)} = {}^t K_T; \quad {}^{t+\Delta t}F^{(0)} = {}^t F \quad (6.2)$$

Convergence is achieved when both the errors, measured as the Euclidean norms of the unbalanced forces and of the residual displacements, are less than a minimum value. The complete equilibrium path can be traced by finding the subsequent solution points at higher load levels using the same approach.

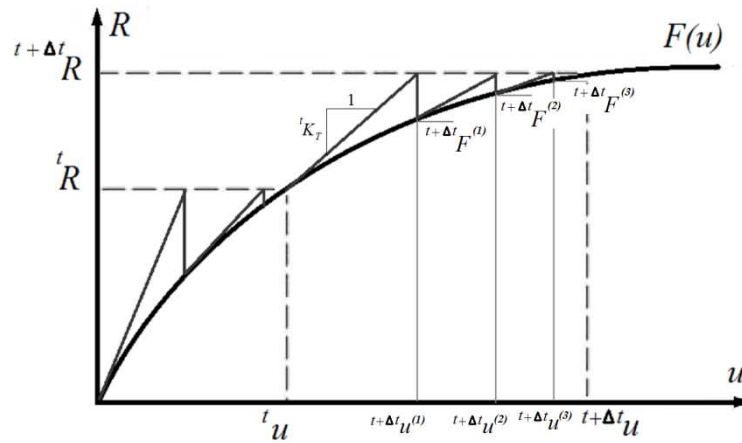


Figure 6.1. Illustration of incremental Newton-Raphson approach.

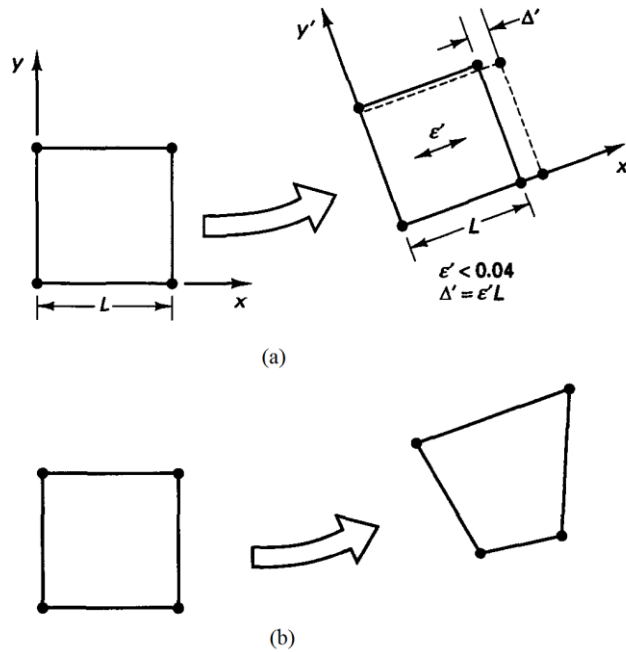


Figure 6.2. (a) Large displacements and large rotations but small strain (b) large displacements, large rotations and large strains (Bathe 2006).

### 6.1.3. Geometrically non-linear behaviour of a continuum body

In general, two classes of geometrically non-linear behaviour exist. First, large displacements, large rotations, but small strains. Second, large displacement, large rotations and large strains. Unlike the first class of geometrically nonlinear behaviour where the strains are small, in the second class the element extensions and angle changes within the element are large (figure 6.2). Typical formulations that are used to model these non-linear behaviours include the Total Lagrangian

(TL) formulation and Updated Lagrangian (UL) formulation. In the TL formulation all static and kinematic variables are referred to the initial undeformed configuration of the structure and the integrals are calculated with respect to that configuration. Due to the transformations, a new measure for stress, the second Piola-Kirchhoff stress tensor, has to be introduced with the Green-Lagrange strain tensor. Considering TL formulation for a general body subjected to applied body forces  $f^B$  and surface tractions  $f^S$  on the surface  $S$  and displacement field  $\delta u_i$ , the equation of motion is given by (Gea and Luo 2001):

$$\int_{0V} S_{ij} \delta \epsilon_{ij} d^0V = \int_{0V} f_i^B \delta u_i d^0V + \int_{0S_f} f_i^S \delta u_i^S d^0S \quad (6.3)$$

where  $S_{ij}$  denote the Cartesian components of the second Piola-Kirchhoff stress tensor,  $\delta \epsilon_{ij}$  are the components of the Green-Lagrange strain tensor corresponding to the virtual displacement field  $\delta u_i$  and  ${}^0V$  denotes the body volume at initial configuration. The Green-Lagrange strain tensor which is defined with respect to the initial configuration of the body is given by (Gea and Luo 2001):

$$\epsilon_{ij} = \frac{1}{2} \left( \frac{\partial u_i}{\partial {}^0x_j} + \frac{\partial u_j}{\partial {}^0x_i} + \frac{\partial u_k}{\partial {}^0x_i} \frac{\partial u_k}{\partial {}^0x_j} \right) \quad (6.4)$$

Considering reasonably small strains, the general elastic constitutive equation can still be used:

$$S_{ij} = C_{ijkl} \epsilon_{ij} \quad (6.5)$$

where  $C$  is the elasticity tensor. Equations (6.3)-(6.5) are the basic equations for calculating the response of a continuum body using the TL formulation. However in order to solve these equations for strongly non-linear problems, one may need to use an incremental-iterative approach, such as Newton-Raphson, as discussed in section 6.1.2.

If the UL formulation is implemented, all static and dynamic variables are referred to the last calculated configuration and the integrals are calculated with respect to the last configuration. In this case a different measure of stress and strain must be used. Both TL and UL formulations can be used to find the response of a geometrically non-linear continuum body and the only advantage of using one rather than the other could be a greater numerical efficiency. In this study, the



assumption is that the structures undergo large displacements and rotations with small strain, and the TL formulation in conjunction with the Newton-Raphson approach is employed to determine the non-linear response of the structures.

## 6.2. Formulation of equation of motion

### 6.2.1. Continuous form of the equilibrium equation

Introducing the incremental approach to find the structural responses in non-linear structures, one can decompose the displacements, strains and stresses at time  $t + \Delta t$  as

$${}^{t+\Delta t}u_i = {}^t u_i + \Delta u_i; \quad {}^{t+\Delta t}\epsilon_{ij} = {}^t \epsilon_{ij} + \Delta \epsilon_{ij}; \quad {}^{t+\Delta t}S_{ij} = {}^t S_{ij} + \Delta S_{ij} \quad (6.6)$$

where  $\Delta u_i$ ,  $\Delta \epsilon_{ij}$  and  $\Delta S_{ij}$  denote the displacements, strains and stresses increments, respectively, to be determined. Implementing Eq. (6.6) into the displacement-strain relation given in Eq. (6.4), the strain increments can be defined as the sum of linear and non-linear terms as

$$\Delta \epsilon_{ij} = e_{ij} + \eta_{ij} \quad (6.7)$$

where the linear incremental strain,  $e_{ij}$  is given by

$$e_{ij} = \frac{1}{2} \left( \frac{\partial \Delta u_i}{\partial {}^0 x_j} + \frac{\partial \Delta u_j}{\partial {}^0 x_i} + \frac{\partial \Delta u_k}{\partial {}^0 x_j} \frac{\partial {}^t u_k}{\partial {}^0 x_i} + \frac{\partial \Delta u_k}{\partial {}^0 x_i} \frac{\partial {}^t u_k}{\partial {}^0 x_j} \right) \quad (6.8)$$

and the non-linear incremental strain,  $\eta_{ij}$  is defined by

$$\eta_{ij} = \frac{1}{2} \frac{\partial \Delta u_k}{\partial {}^0 x_i} \frac{\partial \Delta u_k}{\partial {}^0 x_j} \quad (6.9)$$

Implementing Eq. (6.6) into the equilibrium equation (6.3) and assuming  $\Delta S_{ij} = C_{ijkl} e_{ij}$  and  $\delta \Delta \epsilon_{ij} = \delta e_{ij}$ , the linearized incremental equation of motion is obtained as

$$\begin{aligned} & \int_{0_V} c_{ijkl} e_{ij} \delta e_{kl} d^0 V + \int_{0_V} {}^t S_{ij} \delta \eta_{ij} d^0 V = \\ & \int_{0_V} {}^{t+\Delta t} f_i^B \delta \Delta u_i d^0 V + \int_{0_{S_f}} {}^{t+\Delta t} f_i^S \delta \Delta u_i^S d^0 S - \int_{0_V} {}^t S_{ij} \delta e_{ij} d^0 V \end{aligned} \quad (6.10)$$

The left hand side of Eq. (6.10), which is dependent on the displacements and stress field defines the so-called tangent structure. The right hand side of this equation represents the out of balance virtual work of the body. One may need to use iterative methods for solving this equation until the out of balance force vanishes.

### 6.2.2. Finite element formulation

Transforming the continuous form of the equation of motion represented by Eq. (6.10) to a finite element formulation, the equilibrium equation is obtained as (Gea and Luo 2001; Bathe 1996)

$${}^tK_T\Delta U = (K_0 + K_d + K_\sigma)\Delta U = \Delta F \quad (6.11)$$

where  ${}^tK_T$  is the tangent stiffness matrix and  $\Delta F$  is the load unbalance between the external forces  ${}^{t+\Delta t}R$  and the internal forces  ${}^tF$ .  $K_0$  is the usual small displacement stiffness matrix represented by

$$K_0 = \int_{V_0} B_{L0}^T C B_{L0} d^0V \quad (6.12)$$

where  $B_{L0}$  is a linear strain-displacement transformation matrix used in linear infinitesimal strain analysis. The stiffness matrix  $K_d$  in Eq. (6.11) represents the large displacement stiffness matrix and is defined by

$$K_d = \int_{V_0} (B_{L0}^T C B_{L1} + B_{L1}^T C B_{L0} + B_{L1}^T C B_{L1}) d^0V \quad (6.13)$$

where  $B_{L1}$  is a linear strain-displacement transformation matrix which depends on the displacement.  $K_\sigma$  in Eq. (6.11) is the initial stress matrix dependent on the stress level, and is given by

$$K_\sigma = \int_{V_0} B_{NL}^T {}^tS B_{NL} d^0V \quad (6.14)$$

where  $B_{NL}^T$  denotes the non-linear strain-displacement transformation matrix and  ${}^tS$  denotes the second Piola-Kirchhoff stress matrix, which in a 2D formulation is defined by

$$\begin{bmatrix} {}^tS_{11} & {}^tS_{12} & 0 & 0 \\ {}^tS_{21} & {}^tS_{22} & 0 & 0 \\ 0 & 0 & {}^tS_{11} & {}^tS_{12} \\ 0 & 0 & {}^tS_{21} & {}^tS_{22} \end{bmatrix} \quad (6.15)$$

The correct calculation of the internal forces,  ${}^tF$  in Eq. (6.11) is very crucial as any error in this calculation will result in an inaccurate response prediction. The internal forces can be found from

$${}^tF = \int_{V_0} (B_{L0} + B_{L1})^T {}^t\bar{S} d^0V \quad (6.16)$$

where  ${}^t\bar{S}$  is the second Piola-Kirchhoff stress vector. Eq. (6.11) is used to find the displacement increment corresponding to the state  $t + \Delta t$  which is then added to the displacement at state  $t$  to obtain displacement at state  $t + \Delta t$ . The strain-displacement relation in Eq. (6.4) allows the strain to be determined from the displacements and using the constitutive relation in Eq. (6.5), one can then calculate the corresponding stresses.

### 6.2.3. X-FEM for geometrically non-linear behaviour

In order to find the properties of the elements on the evolving boundary during the optimization process, a similar X-FEM scheme as the linear case can be used. In the case of geometrically non-linear problems, the contribution of the solid parts of the boundary elements into the elements' tangent stiffness matrix as well as the elements' internal forces needs to be identified. For 2D 4-node quadrilateral elements this can be done by dividing the solid part of the boundary elements into sub-triangles and performing Gauss quadrature, similar to the method presented in chapter 4, section 4.4. Following that, the element's tangent stiffness matrix  ${}^t k_\sigma$  and internal force vector  ${}^t F_e$  can be obtained from

$${}^t k_\sigma = \sum_{i=1}^n \sum_{j=1}^m A_T w_j f_1(\xi_1^j, \xi_2^j, \xi_3^j) \quad (6.17)$$

where

$$f_1 = B^T C B t + B_{L0}^T C B_{L1} + B_{L1}^T C B_{L0} + B_{L1}^T C B_{L1} + B_{NL}^T {}^t S B_{NL} \quad (6.18)$$

and

$${}^tF_e = \sum_{i=1}^n \sum_{j=1}^m A_T W_j f_2(\xi_1^j, \xi_2^j, \xi_3^j) \quad (6.19)$$

where

$$f_2 = (B_{L0} + B_{L1})^T {}^t\bar{S} \quad (6.20)$$

The elements' tangent stiffness matrices and internal force vectors can then be assembled to obtain the global tangent stiffness matrix  ${}^tK_\sigma$  and global internal force vector  ${}^tF$  of the structure.

### 6.3. Stiffness design

#### 6.3.1. Objective function and structural performance criteria

In order to find the stiffest design, the natural choice is to minimize the deflection or compliance. However, as shown by Buhl et al (2000), the drawback of this objective function is that it may result in structures that can only support the maximum load they are designed for and may break down for lower loads. In order to avoid this, when the non-linear structure is loaded under force control, the complementary work  $W^C$  can be chosen as the objective function (figure 6.3). In this case the optimization problem can be defined as:

$$\text{Minimize: } f(x) = W^C = \lim_{n \rightarrow \infty} \left[ \frac{1}{2} \sum_{i=1}^n \Delta R^T (U_i - U_{i-1}) \right] \quad (6.21)$$

$$\text{subject to: } \sum_{e=1}^n v_e^s = V^c$$

where  $\Delta R$  is the load increment,  $i$  denotes the increment number and  $n$  is the total number of load increments.

The sensitivity of the objective functions with respect to design variable  $x_e$  is:

$$S_e = \frac{\partial f(x)}{\partial x_e} = \lim_{n \rightarrow \infty} \left[ \frac{1}{2} \sum_{i=1}^n (R_i^T - R_{i-1}^T) \left( \frac{\partial U_i}{\partial x_e} - \frac{\partial U_{i-1}}{\partial x_e} \right) \right] \quad (6.22)$$

To find the elemental sensitivities, an adjoint equation is introduced to the above equation by adding a series of Lagrangian multipliers to the objective function

(Buhl et al 2000). Solving the above equation, Huang and Xie (2010a) found that for nonlinear structures under force control, the elemental sensitivity numbers are equal to the total elemental elastic and plastic strain energy,  $E_e^n$ . This can be used in BESO as the criterion for element removal and addition to find the solution for stiffness optimization of nonlinear structures. Similar to the formulation of the linear Iso-XFEM optimization method, the elemental sensitivity numbers can be used to find the structural performance:

$$SP_e = \frac{E_e^n}{V_e} \quad (6.23)$$

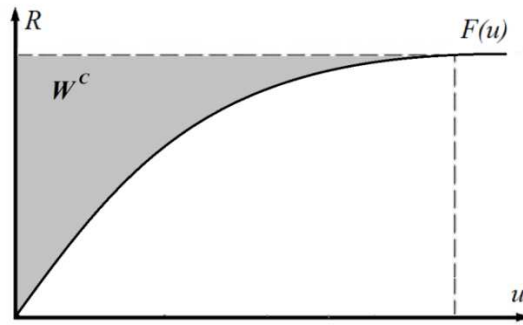


Figure 6.3. Objective function  $W^C$  for stiffness optimization of nonlinear structures under force control.

### 6.3.2. Filter scheme for Iso-XFEM

In order to increase the stability of the Iso-XFEM method applied to geometrically non-linear problems, a similar filter scheme to the one used for BESO (Huang and Xie 2010a) and SIMP (Sigmund 2001) can be employed. This is used to smooth the structural performance distribution over the design domain by averaging the nodal values of structural performance with those of neighbouring nodes. The modified values of structural performance can then be defined by

$$SP_i = \frac{\sum_{j=1}^k w_{ij} SP_j}{\sum_{j=1}^k w_{ij}} \quad (6.24)$$

where  $k$  is the number of nodes inside a domain centred at node  $i$  having a filter radius of  $r_{min}$ .  $w_{ij}$  are the weighting factors defined by

$$w_{ij} = r_{min} - r_{ij} \quad (6.25)$$

where  $r_{ij}$  is the distance between node  $i$  and the neighbouring node  $j$ .

### **6.3.3. Iso-XFEM procedure for geometrically non-linear structures**

The Iso-XFEM procedure for stiffness design of geometrically non-linear structures is similar to the linear ones except that non-linear finite element analysis must be used. Thus, it can be summarized as the following steps:

- 1- Initialize: define the design space, non-design domain, material properties, a fixed grid FE mesh, loads and boundary conditions, and the optimization parameters.
- 2- Perform non-linear FEA: divide the applied load into a suitable number of load increments and find the equilibrium path using the Newton-Raphson approach. Find the properties of the boundary elements using the X-FEM scheme.
- 3- Calculate the elements total strain energy and find the structural performance distribution over the design domain.
- 4- Filter structural performance numbers.
- 5- Average the structural performance numbers with those of the previous iteration.
- 6- Calculate the target volume of the current iteration and find a minimum level of performance to meet the target volume.
- 7- Find the relative performance,  $\alpha$ , over the design domain and extract the design boundary. Assign solid material properties to regions having  $\alpha > 0$  and void material properties to the regions having  $\alpha < 0$ .
- 8- If the convergence condition is reached, stop the design process, else, go to step 2.

## **6.4. Test cases**

A Matlab code was developed to represent the FEA model for analysing geometrically non-linear 2D structures. It uses plane stress 4-node quadrilateral elements to represent the FE model of the structure. An incremental loading

scheme is used in the code and the Newton-Raphson method is used to find the equilibrium at each load increment. The assumption is that the test cases used in the study experience large deformation but with small strain. The non-linear response of the structure is then used to obtain the structural performance which is required for the Iso-XFEM optimization method. This section of the chapter include validation of the Matlab code for analysing geometrically-nonlinear 2D structures, followed by two 2D test cases to demonstrate the application of the Iso-XFEM method to the topology optimization of geometrically non-linear structures.

#### **6.4.1. Validation of the FE Matlab code**

A long cantilever beam was used as a test case to validate the developed non-linear Matlab code. This was selected from MSC Patran exercise work book, enabling comparison of the results of the Matlab code with MSC Nastran results. The beam has length of 100"(2540 mm), width of 2" (50.8 mm) and thickness of 1" (25.4). The material properties of the beam include Young's modulus of  $30 \times 10^6 \text{ lb/in}^2$  (207 GPa) and Poisson's ratio of 0.3. A load of 6000 lbs (26.69 kN) is applied to the free end of the beam. The beam has a high length to cross section ratio which allows large deformations when subjected to a reasonably high amount of vertical load on the tip. In this experiment, the deflection of the beam is obtained from linear and nonlinear FE analysis using the developed Matlab code and the results are compared with those obtained from analysing the same problem using the commercial non-linear FEA code, MSC Nastran, implementing the same element type and the same mesh. Figure 6.4 illustrates the cantilever deformation results from linear and non-linear modelling, with load deflection plots shown in figure 6.5. Table 6.1 compares the tip deflection results of the linear and non-linear modelling using the Matlab code and Nastran FE modelling. It can be seen that the tip displacement obtained from the non-linear analysis is about 23% lower than that from the linear solution in both the Matlab and Nastran FE models. Also, it can be seen that in the linear FEA, both models have resulted in the same tip displacement for the cantilever beam. In the case of the non-linear FEA, there is a difference of less than 0.8% between the solutions obtained from the Nastran and Matlab FEA codes.

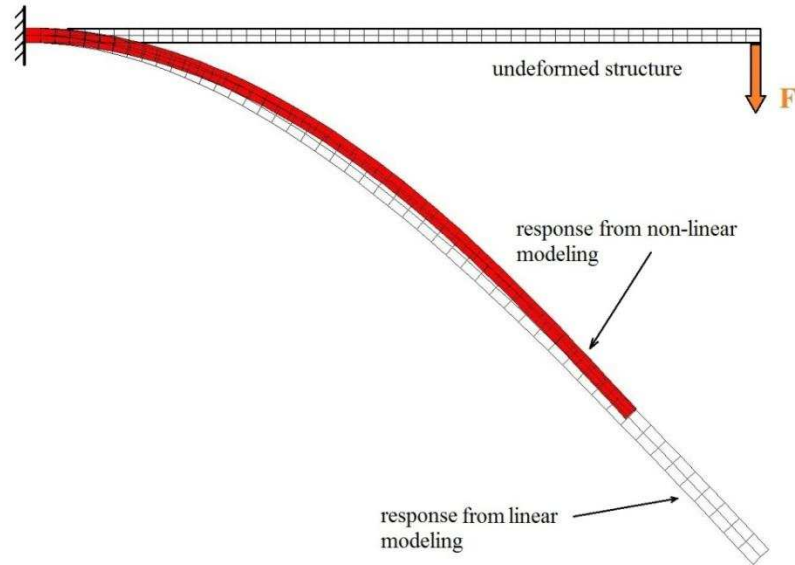


Figure 6.4. Illustration of linear and nonlinear large deformation of a cantilever beam calculated using the Matlab code. Deformations are to scale.

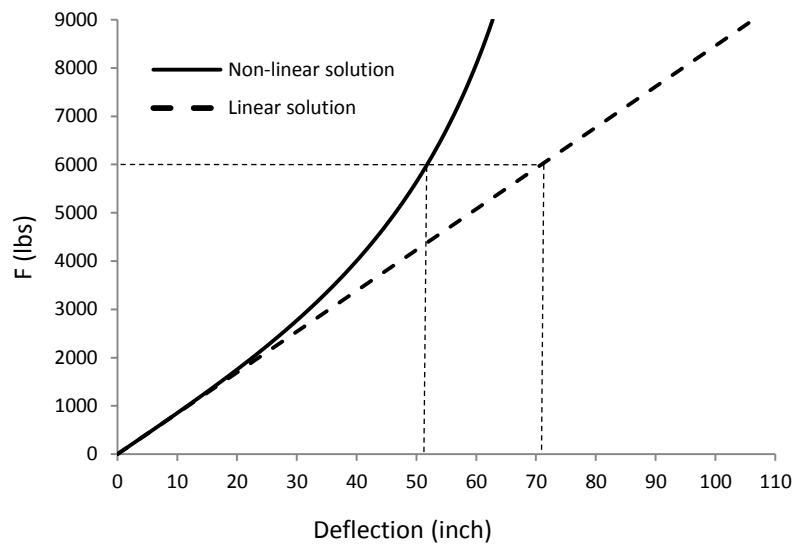


Figure 6.5 Load deflection plots of the geometrically nonlinear cantilever beam.

Table 6.1. Comparison of linear and non-linear response of the cantilever beam obtained from the Matlab FE code and Nastran FE modelling.

	Vertical tip displacement (linear FEA)	Vertical tip displacement (non-linear FEA)
Matlab FE code solution	70.91" (1801 mm)	51.78" (1315.2)
Nastran FE solution	70.91" (1801 mm)	51.38" (1305.1)



#### 6.4.2. Test case NL-1: non-linear cantilever plate

The cantilever plate shown in figure 6.6 was considered as the first non-linear test case of this thesis. Length to cross section ratio of this cantilever was greater than those considered in the 2D linear studies of the thesis, to increase the chance of large deformation in the optimised topologies. Also, this beam has been used as a test case in previous studies, implementing SIMP (Buhl et al 2000) and BESO (Huang and Xie 2010a), allowing the comparison of the Iso-XFEM solutions with the other two methods. The cantilever plate was 1 m in length, 0.25 m in width and 0.1 m in thickness, and was subjected to a concentrated load at the middle of the free end. The material used was nylon which has a low Young's modulus of  $E = 3$  GPa and Poisson's ratio of  $\nu = 0.4$ . Non-linear, stiffness optimised designs of the plate with a volume constraint of 50% of the design domain under two point loads 60 and 144 kN, were investigated and compared. Note that in the case of the non-linear analysis, simple scaling can't be used to determine the results of the FEA at different loads. A mesh of 200x50 quadrilateral elements was used for the FE model of the structure. The volume evolution rate used for this experiment was  $ER = 0.005$ , which was smaller than the volume evolution rate used in the linear study. The reason for this was to increase the stability of the non-linear Iso-XFEM method by performing the material removal within a higher number of evolutionary iterations, i.e. applying less change to the topology at each iteration. Also, a filter radius of  $r_{min} = 1.2$  times the element size was considered. The reason for using a small filter radius was to stabilize the evolutionary process without significantly changing the complexity of the solutions.

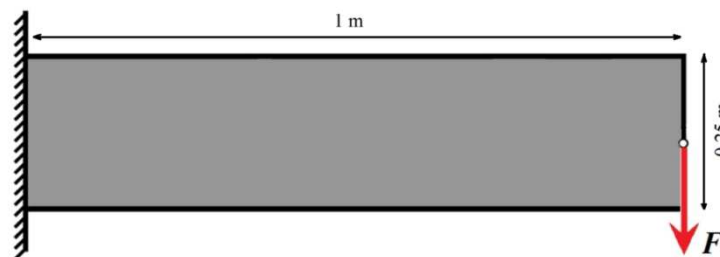


Figure 6.6. Design domain and boundary conditions of the geometrically nonlinear cantilever of test case NL-1.

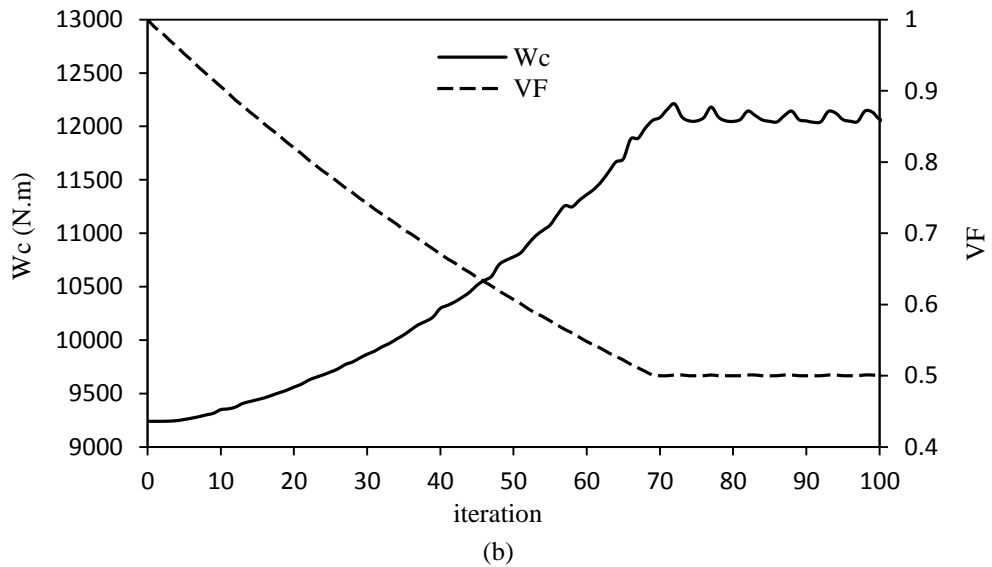
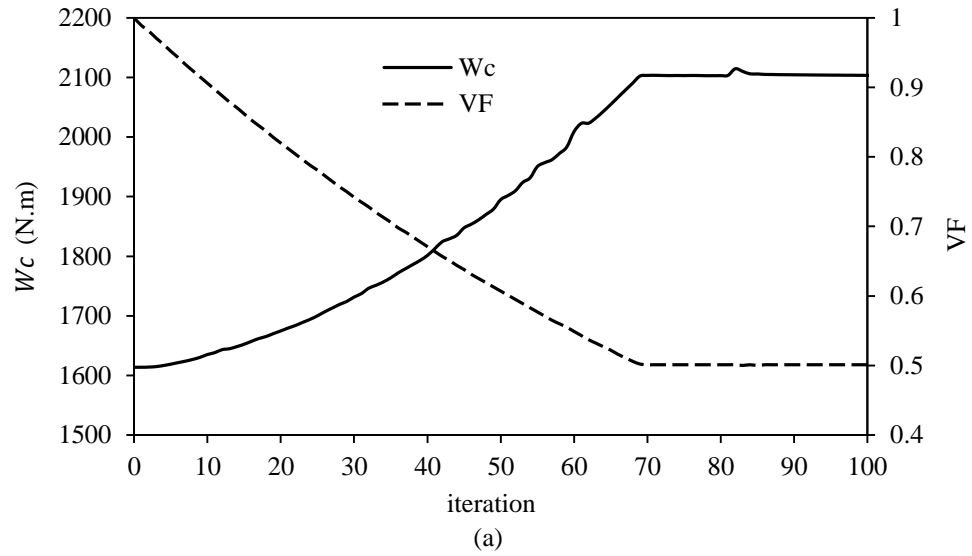


Figure 6.7. Evolution histories of objective function and volume fraction of the non-linear cantilever subjected to (a) a point load of 60kN, and (b) a point load of 144 kN.

Figure (6.7) shows the evolution histories of the objective function ( $W^C$ ) and volume fraction for the both load cases of 60 kN and 144 kN. It can be seen that the evolutionary optimization process of the non-linear structure subjected to the point load of 60 kN has good stability. However, by increasing the load to 144 kN (figure 6.7b), i.e. increasing the degree of non-linearity, some instability was observed in the plot of complementary work (iteration 70 afterward). Figure (6.8) shows the solutions obtained from the linear and non-linear based optimization for the two different load values. Note that linear Iso-XFEM solution for both load cases are the same when same target volume fraction is used. As expected, it can

be seen that the linear Iso-XFEM has converged to a symmetric solution. This is expected as the design is optimised with respect to the equilibrium geometry of the undeformed beam. However, different designs are obtained by implementing the non-linear topology design, showing that the optimal topologies depend on the magnitude of the applied load. These are now non-symmetric as the design is optimised for the deformed beam under load, which is not symmetric. The large deformation of the Iso-XFEM solutions is illustrated in Figure (6.9). This test case has been previously studied using SIMP (Buhl et al 2000) and BESO (Huang and Xie 2010a). Table (6.2) compares the objective values of the linear and non-linear Iso-XFEM solutions with those of SIMP and BESO. It can be seen that the non-linear designs obtained from both Iso-XFEM designs have lower magnitudes of complementary work than their linear designs, indicating a better performance for the load they are designed for. Also comparing the Iso-XFEM with BESO and SIMP solutions in terms of their complimentary work, it can be seen that the Iso-XFEM solutions have lower magnitudes of complementary work than BESO and SIMP solutions, showing better performance of the Iso-XFEM solutions due to their smooth boundary representation. The slightly lower complimentary work of the SIMP solutions compared to the BESO solutions has been attributed to the effect of intermediate density elements in SIMP topology where their strain energy might have been overestimated (Huang and Xie 2010a).

This test case showed that by using non-linear FE modelling in the Iso-XFEM method, a different solution with a higher performance than the linear design can be achieved. However, it could be argued that difference in the overall topology of the linear and non-linear solutions of this test case was insufficient to justify the extra effort of the non-linear analysis. As will be shown in the next example, in some cases the difference can be extremely large and can make the use of non-linear modelling essential.

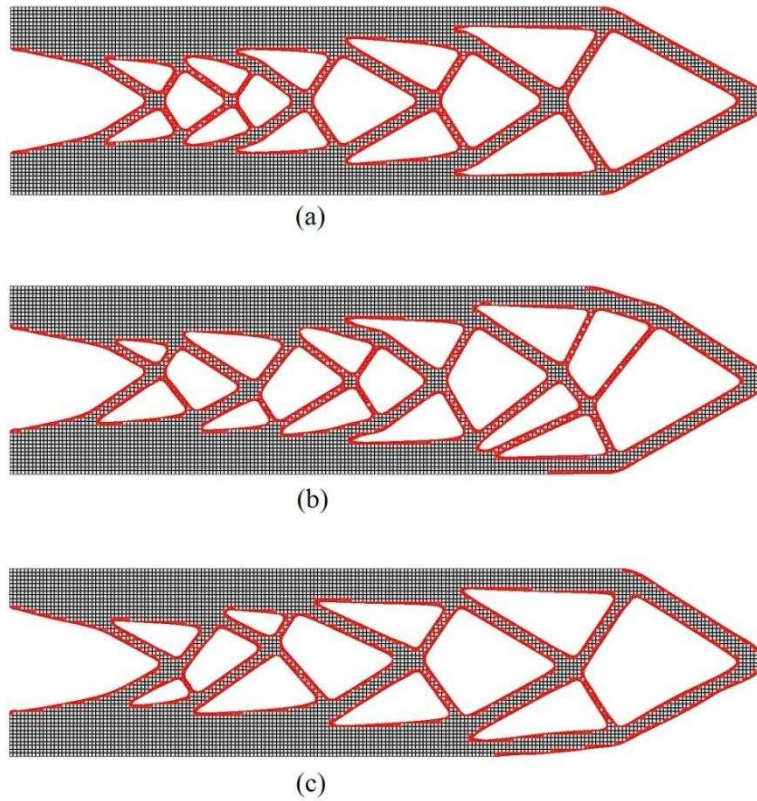


Figure 6.8. Iso-XFEM solutions of the large displacement cantilever problem: (a) linear design (for both load cases of 60 kN and 144 kN) (b) non-linear design for point load of 60 kN (c) non-linear design for point load of 144 kN.

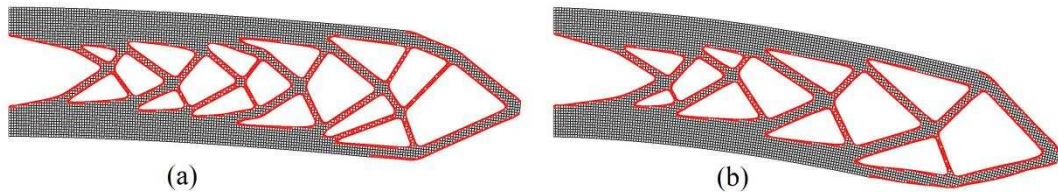


Figure 6.9. Illustration of the large deformation of the cantilever of test-case NL-1 (a) cantilever subjected to point load of 60 kN (b) Cantilever subjected to point load of 144 kN. The deformations are to scale.

Table 6.2. Comparison of the complementary works of linear and non-linear designs for test case NL-1.

	Design for F=60 kN	Design for F=140 kN
Linear design from Iso-XFEM	2.107 kJ	12.072 kJ
Non-linear design from Iso-XFEM	2.101 kJ	12.063 kJ
Non-linear design from BESO (Huang and Xie (2010a))	2.171 kJ	12.38 kJ
Non-linear design from SIMP (Buhl et al 2000)	2.331 kJ	13.29 kJ

### 6.4.3. Test case NL-2: slender beam

The purpose of this experiment was to apply the Iso-XFEM method to topology optimization of a structure having snap-through buckling effects, i.e. the transition between two stable states in a structure. In this case, different topologies could be obtained by using linear and non-linear modelling in the structural optimization problem. As an example of a structure involving snap-through effects, topology optimization of a slender beam with the design domain and boundary conditions shown in figure (6.10) was considered. The beam was 8 m long, 1 m deep and 100 cm thick. A load of 400 kN was applied to the centre of the top edge. The material properties of the beam were Young's modulus of  $E = 3$  GPa and Poisson's ratio of  $\nu = 0.4$ . Non-linear and linear stiffness optimised designs of the beam for a volume constraint of 20% of the design domain for downward and upward loads were investigated. A mesh of 320x40 quadrilateral elements was used for the FE model of the structure in all the experiments, and a volume evolution rate of  $ER = 0.01$  and a filter radius of  $r_{min} = 1.2$  times the element size were used as optimization parameters.

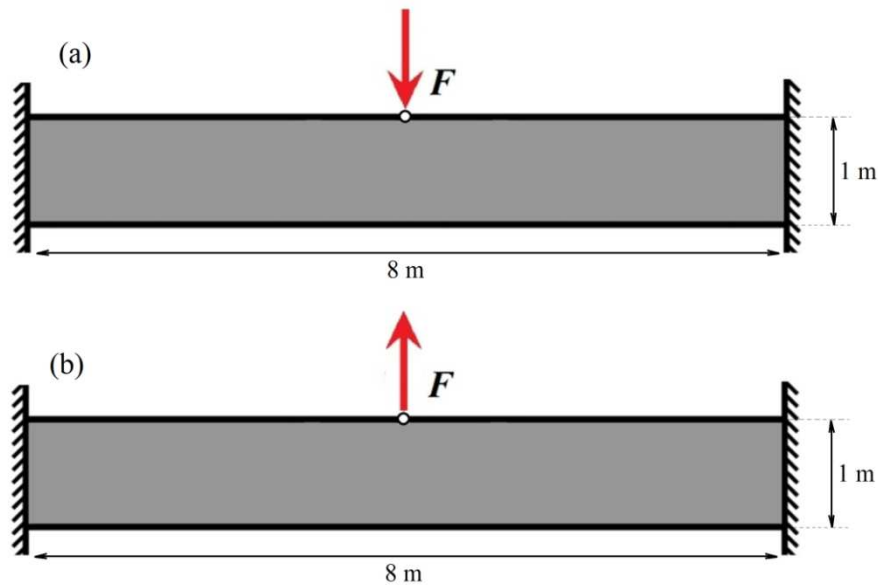


Figure 6.10. Design domain and boundary conditions of the geometrically nonlinear slender beam of test case NL-2: (a) beam subjected to downward load (b) beam subjected to upward load.

Figure 6.11 compares the non-linear Iso-XFEM solutions of the beam subjected to downward (6.11a) and upward (6.11b) loads with the linear Iso-XFEM solution (6.11c). It can be seen that with the linear modelling the same solution is obtained for the structure subjected to either an upward or downward load, i.e. the magnitude of the load doesn't change the solution in a linear topology optimization implementation. However, it can be seen that by using the non-linear topology design approach, very different solutions are obtained for upward and downward loads. It can also be seen in figure 6.11 that in this test case, the solution for the upward load case is very similar to the linear solution. This can be explained by looking at the deformations of the various designs under load.

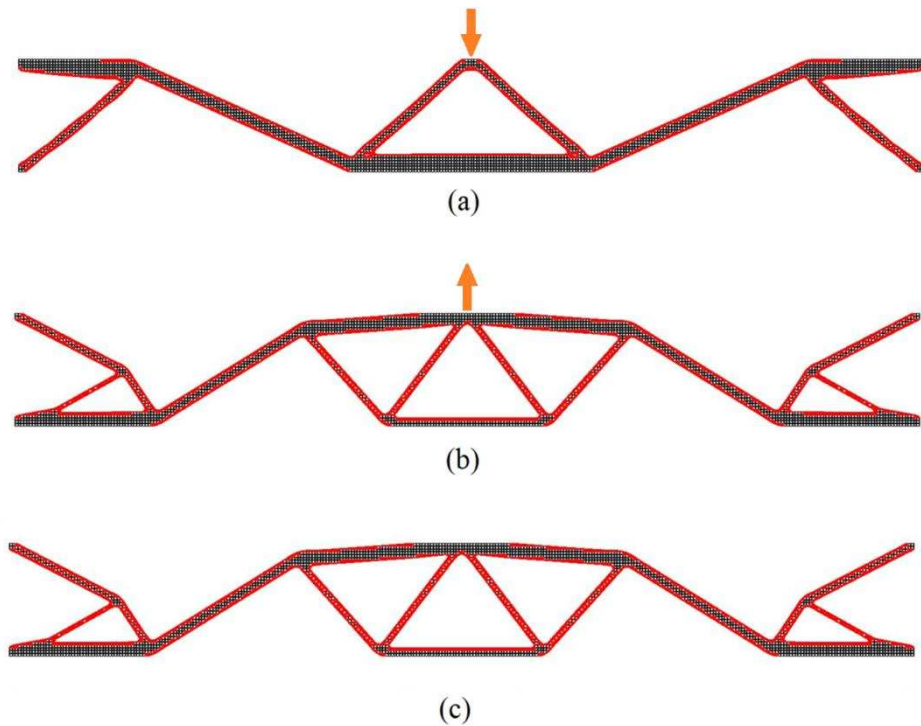
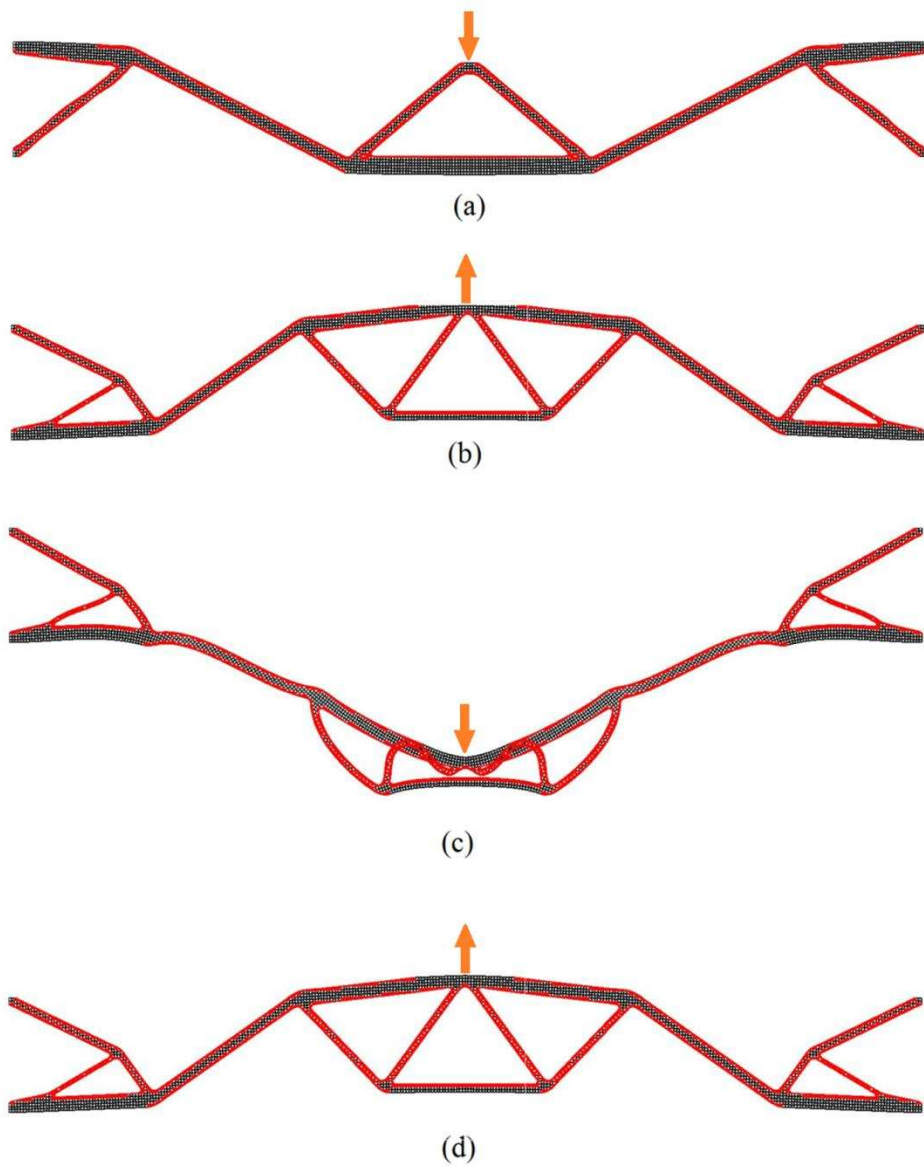


Figure 6.11. (a) Non-linear design of the beam subjected to a downward load (b) non-linear design of the beam subjected to an upward load (c) linear design of the beam (for both downward load and upward load cases).



6.12. (a) Displacement of topology 6.11a (b) displacement of topology 6.11b (c) displacement of topology 6.11c subjected to downward load (d) displacement of topology 6.11c subjected to upward load. The deformations are to scale.

Table 6.3. Comparison of the complementary works of non-linear and linear designs for test-case 2.

Complementary work	Design for downward load	Design for upward load
Non-linear design from Iso-XFEM	38.700 kJ	36.492 kJ
Linear design from Iso-XFEM	55.548 kJ	36.494 kJ

Figure 6.12 shows the deflection of the nonlinear and linear topology optimization solutions of the beam, subjected to both upward and downward loads. The deflection of all non-linear and linear solutions has been determined using geometrically non-linear FEA. It can be seen that the solution of the non-linear design subjected to the downward load remains stable after applying the specified load it is designed for (figure 6.12a). However the linear design has become distorted under the prescribed downward load (figure 6.12c), which can be attributed to the buckling effects (Buhl et al 2000). This is because the linear solution of figure 6.11c has two thin members in the middle which are put under compression with the downward load. Although this is not an issue when linear modelling is used, with non-linear modelling the thin compressed beams buckle and the whole structure experiences snap-through as seen in figure 6.12c. The snap-through effect was not an issue for the upward loading as the thin struts were not put under compression, hence the similarity of the linear and nonlinear designs for upward loading (figures 6.12b and 6.12d). Table 6.3 compares the complementary works of the solutions subjected to downward and upward loads. As anticipated, the difference between the complementary works of non-linear and linear solutions for the upward load case is not significant. However in the case of the downward load case, the complementary work of the linear design involving buckling and snap-through effects is much higher than the non-linear one, showing the importance of implementing a non-linear topology optimization approach for large displacement problems such as those involving snap-through effects.

## **6.5. Summary and conclusions**

In this chapter, the topology optimization of geometrically nonlinear structures was investigated, assuming the structures undergo large displacement with small strain. A total Lagrangian FE formulation was used to model the geometrically non-linear behaviour of continuum structures and a Newton-Raphson iterative method was used to find the equilibrium solution at each load increment. The non-linear FE code developed for 2D structures was then integrated into the Iso-XFEM method to enable the topology optimization of structures undergoing large deformation. A



filter scheme was used in the method to increase the stability of the evolutionary optimization approach applied to non-linear structures.

The topology optimization results achieved implementing linear and non-linear modelling showed that, for the presented test cases, a non-linear based optimization returns solutions that are dependent on the magnitude of the load. Also, the solutions achieved from the optimization using non-linear modelling have a higher performance than those with linear modelling. Although in the first test-case of this study, there is not a significant difference between the solutions achieved from linear and non-linear modelling, the results from the second test case, which involves snap-through effects, showed the importance of implementing non-linear modelling in large displacement problems.

# Chapter 7

## Discussion

The aim of this research was to develop a reliable and efficient topology optimization algorithm which doesn't have the limitations of conventional FE-based methods in representing design boundary. The following objectives have been achieved:

- Commonly used topology optimization methods were investigated.

The weaknesses and strengths of each method were investigated in chapter 2. It was found that evolutionary based optimization methods are simple to programme and apply to 3D geometrically complex structures; also, they have the advantage of representing the solutions with clearly defined boundaries.

- Different numerical analysis techniques used in structural analysis were investigated.

In chapter 3, different numerical techniques appropriate for structural analysis were investigated. The idea was to find a numerical analysis technique to use in conjunction with one of the topology optimization algorithms investigated in chapter 2, to develop a new efficient and reliable topology optimization method, benefiting from improved boundary representation. It was found that X-FEM can be used as an efficient method to improve the quality of topology optimised solutions near the boundary, without the need for remeshing.

- A topology optimization method which is benefiting from a boundary improvement scheme was developed.

In chapter 4, a topology optimization method, called ‘Iso-XFEM’, was developed by combining X-FEM and isoline/isosurface boundary representation with an evolutionary based optimization algorithm.

- The proposed method was extended to the topology optimization of 3D real-life structures, multiple objectives and non-linear problems.

In chapter 5, the proposed method was extended to topology optimization of 3D geometrically complex structures, and the required modifications for extension of the method to optimization of different objectives were presented. As an example the method was applied to stiffness optimization of structures with an additional displacement constraint. In chapter 6, the method was extended to topology optimization of geometrically non-linear problems.

- Matlab codes presenting the proposed topology optimization method and its modifications and extensions were developed.

Matlab was used to present the algorithms used in the proposed method and its extensions. Samples of these codes are shown in appendices A:D.

## **7.1. General summary and major findings**

Topology optimization is the most challenging aspect of structural optimization, an approach that has been developed to enable design of high performance lightweight structures. A common drawback of the dominant topology optimization methods is the need for additional post-processing of the topology optimised solutions, as the quality of the optimised topology is mainly dependent on the use of finite element method in the topology optimization approach. The aim of this work was to develop an alternative topology optimization method which allows the generation of reliable and high quality topology optimised solutions, in order to avoid additional costly and time consuming post-processing. It was found that there could be two main factors that influence the quality of the

topology optimised solution. First, the optimization algorithm used, and second, the numerical structural analysis technique integrated with the optimization algorithm. These two were investigated in the two literature review chapters of the thesis (chapters 2&3).

In terms of the topology optimization algorithm, the author's interest was in the material distribution methods as these methods allow better handling of topological changes during the optimization process, compared to boundary variation methods. These methods include homogenization, SIMP and evolutionary based topology optimization methods (ESO & BESO). Evolutionary based topology optimization methods was found promising for the objectives of this work, because these are well integrated with FE framework of the structures, making the methods easy to be programmed and be applied to complex 3D structures. Furthermore, these methods generate discrete (0/1) solutions with clear FE based boundaries as opposed to SIMP with variable density solutions.

To find an appropriate numerical analysis method to be integrated with an evolutionary optimization algorithm, both meshfree and mesh-based methods were investigated in chapter 3. It was found that in general, FEM is faster than meshless methods, and because it is an established, well developed technique, its application to different structural and multiphysics problems is more convenient. Different modifications of FEM to improve quality of solutions near boundary were investigated, including adaptive mesh refinement schemes and extended finite element method (X-FEM). The conclusion was that mesh refinement schemes could increase the computational cost of the optimization, especially in 3D applications, however, the X-FEM scheme proposed for modeling holes and inclusions (which is the type of discontinuity in material distribution problems) benefits from a high numerical efficiency and is a promising technique for structural optimization applications.

In order to use X-FEM in conjunction with an evolutionary based topology optimization algorithm, there was a need to use a boundary representation approach to define the boundary of the material/void discontinuity within finite elements. This was achieved in chapters 4 and 5 by introducing isoline and isosurface approaches as implicit boundary representation schemes for 2D and 3D

structures. The idea was to implicitly define the design boundary (material-void interface) using isolines (or isosurfaces in 3D) of a measure of structural performance. The X-FEM scheme was used to find the properties of the elements on the boundary, and an evolutionary optimization algorithm was used to drive the solution toward optimum. The proposed method, called Iso-XFEM, has been found to be very efficient in generating optimal topologies with smooth and accurate representation of the design boundary. It was found that this method benefits from the simplicity of the conventional evolutionary based optimization methods in the application of the method to practical problems, while some drawbacks of the conventional ESO/BESO methods, such as checkerboarding and poor FE based boundary representation, are removed. The other challenges existed during development of this work were the application of the method to 3D real-life problems, different objectives, as well as problems involving nonlinearity. These were investigated in chapters 5 and 6.

Two different evolutionary optimization algorithms were developed in this study. In the first algorithm, which was presented in chapter 4, a performance evolution rate was used to drive the evolutionary optimization algorithm. Although this method was successfully used for the stiffness design of structures, a drawback was that by using the performance evolution rate, there was little control of the amount of material removed in every evolutionary iteration. As a result, the plots of objective function and volume fraction histories showed some instabilities.

In order to increase the stability of the evolutionary optimization process, and to enable application to alternative optimization problems, a new optimization algorithm was presented in chapter 5. In this method, a volume evolution rate was used, thus allowing more control over the amount of material removed during the evolution of the optimised topology. As a result, smooth plots of objective function and volume fraction histories were obtained using the new algorithm, showing the increased stability of the method compared to the one presented in chapter 4.

The Iso-XFEM method presented in chapter 5 was successfully applied to the stiffness optimization of 2D and 3D structures. The required modifications of the method to enable extension to alternative optimization problems with different objectives or constraints were also discussed in chapter 5. As an example of such

an extension of the method, the stiffness optimization of structures with an additional displacement constraint were considered. 2D and 3D test cases were presented to validate the applicability of the method in meeting the design constraints.

In chapter 6, the method was extended to geometrical non-linear problems. This was achieved by using a total Lagrangian FE formulation and an incremental-iterative Newton-Raphson procedure to determine the equilibrium solution at every evolutionary iteration. A filter scheme was also used to increase the stability of Iso-XFEM method dealing with non-linear problems. A non-linear FE Matlab code was developed to enable the large deformation response of the structure during the optimization process to be determined. The comparison of the solutions obtained from linear and non-linear modelling showed that the solutions obtained from non-linear modelling have lower values of objective function (higher performance) than those obtained from linear modelling. It was also observed that for the structures involving snap-through effects, the use of non-linear modelling is of significant benefit in topology optimization.

Throughout the thesis, Iso-XFEM solutions have been compared with BESO solutions for many 2D and 3D test-cases, in terms of performance, surface quality, and computational efficiency of the methods. The reason for choosing BESO for comparing with Iso-XFEM solutions was that they are both evolutionary based optimization approaches, with defined geometric boundaries, unlike the variable density solutions seen with SIMP. This and the different optimization parameters make an objective comparison with SIMP difficult. Especially, the Iso-XFEM algorithm presented in chapter 5 behaves very similar to BESO in terms of material removal, by considering a volume evolution rate in the algorithm. Therefore it allows doing a fair comparison of the solutions by using the same mesh and the same optimization parameters in both optimization approaches. In the following sections, a summary of these comparisons and major findings is presented.

## 7.2. Validating the Iso-XEM solutions

During the work, validation of the method has been done for the 2D, 3D and non-linear versions of Iso-XFEM method. In the case of 2D problems, a cantilever beam frequently used as a benchmark problem in the literature was used as initial validation of the Iso-XFEM method presented in chapter 4. Two different loading conditions were considered for the cantilever and the solutions were found to be very similar to those seen in the literature as well as the results obtained from BESO method. Moreover, in order to evaluate the accuracy of the objectives obtained using X-FEM scheme, the Iso-XFEM solutions of the cantilever beams were reanalysed implementing a fine converged mesh, using standard FEM, and the results were found to be in close agreement with those found using the X-FEM scheme.

Validation of the 3D version of the Iso-XFEM method has been investigated in two sections of chapter 5. First, in section 5.4.1, 3D Iso-XFEM solutions of a cantilever plate test case with varying thickness were compared with 2D Iso-XFEM solution. The results showed that the 3D Iso-XFEM solutions of the cantilever plate are very similar to the 2D solution when the structure is in plane stress condition. However, when the thickness increases beyond the plane stress condition, the performance of the 3D solutions is higher than the 2D solution. The next validation of 3D Iso-XFEM solutions was in section 5.4.3 where the Iso-XFEM solutions of an aerospace arm were compared with BESO solutions. The results showed that Iso-XFEM solutions had similar topologies as BESO solutions, implementing the same starting mesh, however, of higher performance.

Regarding the non-linear topology optimization studies of this work in chapter 6, the non-linear FEA Matlab code was first validated by comparing the deflection of a geometrically non-linear cantilever beam with Nastran non-linear FEA solutions (section 6.4.1). Then Iso-XFEM solutions of a geometrically non-linear cantilever plate problem were compared with those obtained using SIMP and BESO. Again, similar topologies, however of higher performance have been observed with Iso-XFEM method.

It was observed from 2D, 3D and non-linear studies that by using the Iso-XFEM method, similar topologies as BESO can be achieved. However the performance of Iso-XFEM solutions tends to be higher than BESO. This can be attributed to two facts. First, the poor boundary of BESO solutions is a reason for the solutions to be less optimal than the Iso-XFEM solutions which are represented with smooth boundary. Also, the use of checkerboard filter in BESO reduces the complexity and performance of the solutions. However, with Iso-XFEM, more complex solutions can be obtained, as this method doesn't need checkerboard filter.

It was observed that the combination of X-FEM and isoline/isosurface modelling with an evolutionary optimization algorithm enabled the design of topology optimised structures represented with smooth and clearly defined boundaries. However, the question that may arise is whether Iso-XFEM solutions are more optimal than solutions from conventional element-based methods after thresholding the densities, for instance, a SIMP solution thresholded in an arbitrary density value. To address this, Altair Optistruct software was used to solve topology optimization problem for test case 3D-1 (section 5.4.1) using the SIMP method, with the same starting mesh as the one used with Iso-XFEM method, i.e.  $40 \times 20 \times 2$ . Figure 7.1 compares the topologies obtained from Optistruct with the Iso-XFEM solution. Figure 7.1a is Optistruct solution (optimal density distribution) without implementing minimum member size constraint. It can be seen that design boundary is not clear as the solution is represented with intermediate densities. Figure 7.1b shows the densities thresholded in relative density of 0.5 which results to a solution with approximately the same volume as the prescribed volume constraint. However, it can be seen that thresholding the densities resulted in an unfeasible design as the structure has lost its member connectivity. It is possible to preserve member connectivity by choosing a lower isovalue of density. However, this would result in a solution of higher volume than the volume constraint, which requires further optimization and post-processing.

In order to have a near 0/1 solution and prevent the loss of member connectivity after thresholding the densities, a minimum member thickness of  $2 \times$  *element size* was applied, resulting in the solution shown in figure 7.1c. It can be seen that this is a near 0/1 solution and the members connectivity is well defined, however, with rough boundaries. Figure 7.1d shows the solution obtained by



thresholding the densities of the solution shown in figure 7.1c. The densities were thresholded in isovalue of 0.5, resulting in the same volume as the prescribed volume constraint, thus, enabling the comparison of all solutions with the Iso-XFEM solution, shown in figure 7.1e.

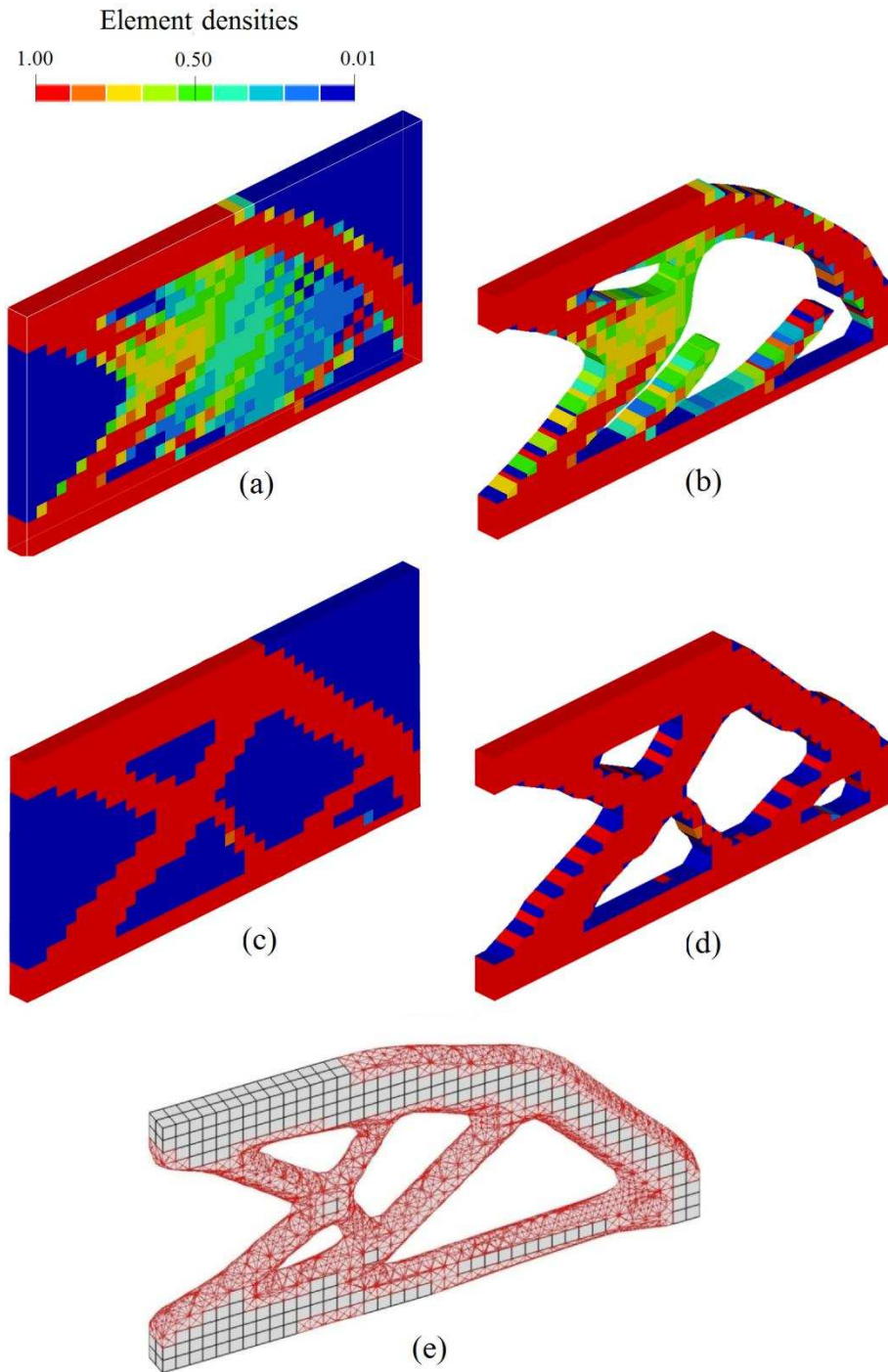


Figure 7.1. Comparison of Optistruct and Iso-XFEM solutions for test case 3D-1: (a) Optistruct solution without applying minimum member size constraint, and (b) densities of this solution thresholded in isovalue of 0.5. (c) Optistruct solution with minimum member size constraint, and (d) densities of this solution thresholded in isovalue of 0.5. (e) Iso-XFEM solution.

To quantify the optimality of the Optistruct and Iso-XFEM solutions shown in figure 7.1, the strain energy of the solutions has been measured and presented in table 7.1. It can be seen that the design obtained by thresholding densities of optistruct solution has a higher value of objective than Optistruct solutions with/without minimum member size constraint. This is because the design obtained by thresholding the densities of a topology optimised solution, is not an optimised solution anymore, and will require additional shape optimization and further post-processing until it becomes an optimised manufacturable design. However, it can be seen that the strain energy of the X-FEM solution is lower than the other three designs, indicating that the Iso-XFEM solution is a more optimal solution compared to the Optistruct solutions before/after thresholding the densities. The results of this experiment clearly show that although thresholding the densities can improve the design boundary, it can reduce the optimality of the solutions.

Table 7.1. Comparison of Optistruct and Iso-XFEM solutions.

	SE
Optistruct solution without minimum member size constraint (figure 7.1a)	7.34
Optistruct solution with minimum member size constraint (figure 7.1c)	7.17
Design obtained by thresholding densities of the Optistruct solution with minimum member size constraint (figure 7.1d)	7.63
Iso-XFEM solution (figure 7.1e)	6.56

### 7.3. Numerical instabilities

The examples presented in this thesis showed that Iso-XFEM can produce checkerboard free solutions without the need to implement any sensitivity filtration scheme. However element-based methods such as BESO and SIMP require sensitivity filtration scheme or similar approaches to suppress the checkerboard pattern formation in the solutions of topology optimization problem. This allows Iso-XFEM to generate solutions with higher complexity (which have higher performance) than those can be obtained from BESO when implementing the same FE mesh.

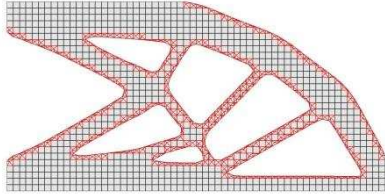
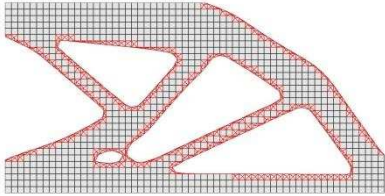
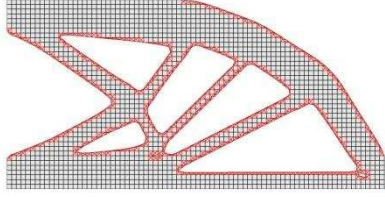
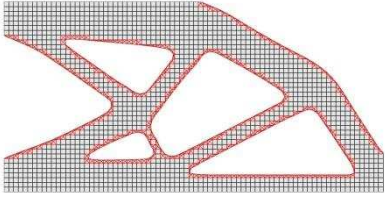
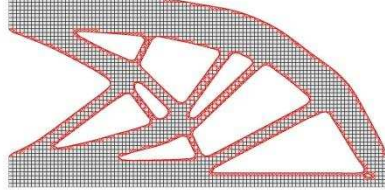
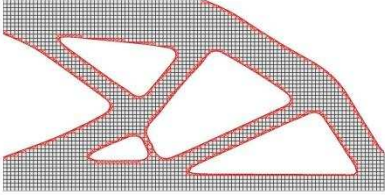
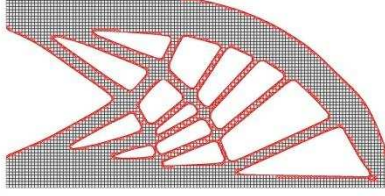
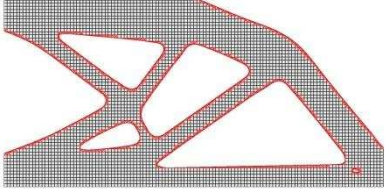
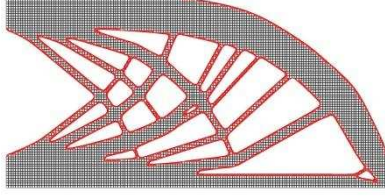
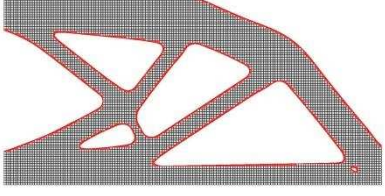
Mesh	Topology implementing no filtration scheme	Topology implementing filter radius of $r = 3$
60x30	 SE = 30.81	 SE = 31.39
80x40	 SE = 30.96	 SE = 31.13
100x50	 SE = 30.75	 SE = 31.19
120x60	 SE = 30.56	 SE = 31.31
160x80	 SE = 30.55	 SE = 31.35

Figure 7.2. Comparison of the solutions obtained from Iso-XFEM with/without implementing sensitivity filtration scheme.

Applying Iso-XFEM to the structures represented with different mesh sizes (cantilever beam in chapter 4 and the aerospace arm in chapter 5) showed that by increasing the mesh density, solutions of a higher complexity, having higher performance can be achieved. This also means that Iso-XFEM is a mesh-dependent approach like BESO or other element based methods of topology optimization. A

degree of mesh dependency is unavoidable in element based topology optimization methods. A pseudo mesh independent solution can be obtained by actions such as filtering the sensitivities and by increasing the filter radius, but often they result in coarse topologies. A similar filtration scheme can also be employed in the Iso-XFEM method by substituting the structural performance number of an element to an average value of the structural performance number of neighbouring elements. Figure (7.2) shows the Iso-XFEM solutions of a cantilever beam with/without performance filtration for the beam represented with different FE mesh. Similar to the study in section 4.7, in order to avoid singularity issues with concentrated loading, the element's strain energies near the loading region is not considered in the calculation of the objective function (total strain energy). Figure 7.2 shows that by applying the filtration scheme, almost the same topologies have been obtained for the structures with different starting mesh, thus achieving mesh independent solutions. However these solutions tend to have a lower performance than those obtained without using the filtration scheme (and have higher geometrical complexity). Therefore if geometrical complexity is not an issue in fabricating the parts (additive manufacturing for instance) the use of Iso-XFEM without implementing filtration scheme is preferable.

#### **7.4. Surface roughness of the solutions**

The experiments performed in this thesis showed that the Iso-XFEM solutions have much smoother boundaries than BESO solutions as their solutions are represented with sub-element resolution. This fact has been quantified for 2D and 3D solutions in chapters 4 and 5, respectively. In chapter 4, a Laplacian smoothing process as a post-processing stage was used to smooth both Iso-XFEM and BESO solutions, and then the deviation of the smooth boundary from its original state was calculated in terms of root mean square roughness ( $R_q$ ). The comparison of the values of  $R_q$  between Iso-XFEM and BESO solutions showed that Iso-XFEM solutions require only a little post-processing and the FE mesh that is used for the optimization doesn't have a significant effect on the value of surface roughness. A similar experiment was performed in chapter 5 to compare the surface roughness of 3D solutions on the different regions of the structures in terms of their

arithmetic mean surface roughness,  $R_a$ . The Iso-XFEM solutions had much lower values of surface roughness than the BESO solutions.

## **7.5. Computational efficiency of Iso-XFEM**

Comparison of computational efficiency of different topology optimization methods is not very straightforward as the optimization parameters selected for an optimization process can affect the number of iterations to converge. However, the Iso-XFEM method developed in this work is similar to BESO in terms of the optimization algorithm. Therefore the method was compared with BESO in terms of the computational cost of the first 100 iteration of optimization process.

In general, because Iso-XFEM method employs a consistent FE mesh throughout the optimization process, i.e. no remeshing, and the number of degrees of freedom of FE system doesn't change, one would not expect a significantly different computational cost for Iso-XFEM method than BESO. Here, the only source for the additional computational cost of Iso-XFEM method compared to BESO, could come from the need for calculating the stiffness matrix of boundary elements in the Iso-XFEM method using the proposed X-FEM scheme. However this could be less important for the large systems of finite elements associated with higher number of degrees of freedom.

Comparing the computational cost of Iso-XFEM and BESO in optimization of 2D structures, it was observed that BESO performs faster when a coarse mesh is implemented and by increasing the mesh density, the difference in the computational cost of the two approaches decreases. However as both Iso-XFEM and BESO are quit fast in 2D, the use of Iso-XFEM can be preferable as it generates solutions with sub-element resolution, requiring less post-processing.

Compared to the topology optimization of 2D structures, application of the topology optimization approaches to the design of 3D structures requires much more computational effort because of the increase in the number of FE DoFs. However 3D implementation is necessary when the geometry and boundary conditions cannot be represented in 2D, or when the thickness to length ratio is

relatively high. As shown in chapter 5, the computational cost of Iso-XFEM is higher than BESO when implementing the same FE mesh. However BESO would require a much finer mesh to generate structures with similar resolution as those generated with Iso-XFEM. Otherwise it requires a time consuming post-processing to obtain structures with a smooth boundary. Therefore the higher computational cost of Iso-XFEM is offset by the benefit of being able to employ a coarse mesh to generate a smooth topology.

## **7.6. Final remarks**

The proposed method, Iso-XFEM, has numerous advantages to the conventional element based evolutionary structural optimization methods. The method benefits from simplicity and potential applicability of evolutionary based methods to complex problems, however, many issues regarding the element-based nature of the resulting topologies from conventional methods have been resolved. No checkerboard is observed in the Iso-XFEM solutions. High resolution topologies with clearly defined smooth boundaries can be obtained with the proposed method. The Iso-XFEM solutions are less sensitive to start mesh than BESO solutions, for instance, and a smaller change in the value of converged objectives were seen in the Iso-XFEM solutions by refining the start mesh. Similar to conventional evolutionary based methods, as well as SIMP, the issue of mesh dependency exists. However, similar strategies than those used in the conventional methods, such as sensitivity filter scheme, can be used to prevent increasing the complexity by refining the start mesh. The Iso-XFEM method has a good computational efficiency which is lower than BESO. However, this can be offset by the benefit of its capability in generating high resolution smooth topologies employing a coarse mesh. Extension of the Iso-XFEM method to different practical problems, including those dealing with non-linearity effects and alternative objectives/constraints could be straightforward, as some examples of such extensions are presented in this thesis.

The next chapter presents the achievements and conclusions of this work, followed by recommendations for the future work.

# Chapter 8

## Conclusions and Future Work

### 8.1. Achievements

The subject of topology optimization was introduced and reviewed in the first three chapters of this thesis. Other chapters include the original work of this author in developing the Iso-XFEM method as a new evolutionary topology optimization method, as well as modifications and extensions of the method to various topology optimization problems. The key achievements of this work are summarized below:

- The Iso-XFEM method was developed and introduced as an alternative to the conventional evolutionary based topology optimization methods.
- Compliance minimization (stiffness optimization) problem for the Iso-XFEM method was formulated.
- The Iso-XFEM method was extended to topology optimization of 3D and real-life structures.
- The Iso-XFEM method was extended to optimization of problems with alternative objectives/constraints, including stiffness optimization with an additional displacement constraint.
- The Iso-XFEM method was formulated for stiffness optimization of geometrically non-linear problems.
- Matlab codes presenting the Iso-XFEM method and its modifications/extensions were developed.

## 8.2. Conclusions

Based on the achievements, the following conclusions can be drawn from this study:

- Iso-XFEM method is a reliable and efficient evolutionary optimization method for topology optimization of continuum structures.
- Compared to the conventional element based methods of topology optimization, Iso-XFEM solutions require less post-processing, since they are represented with sub-element resolution.
- No checkerboard was observed in Iso-XFEM solutions, thus no need to apply checkerboard filter scheme. However, in some cases, a similar filter scheme can be implemented to increase the stability (in nonlinear problems) or to reduce design complexity (when using traditional manufacturing process).
- Iso-XFEM solutions have high accuracy and the objectives are less sensitive to mesh refinement compared to BESO.
- Two different evolutionary optimization algorithms were presented in chapters 4 and 5, employing performance evolution rate and volume evolution rate, respectively. The evolutionary optimization algorithm with volume evolution rate has a better stability than the one with performance evolution rate, and can be used to extend the method to optimization of objectives other than compliance.
- Iso-XFEM method works well with 3D complex geometries and structures represented with non-uniform FE mesh.
- Little changes are required to extend the Iso-XFEM method to optimization of alternative problems. For instance, Iso-XFEM can be used for stiffness optimization of structures with an additional displacement constraint.
- Iso-XFEM method can be used to optimise nonlinear structures.



- The non-linear Iso-XFEM approach returns solutions that may be different from linear ones. For instance, non-linear Iso-XFEM solution for a symmetric structure could be non-symmetric. It is vital to use the non-linear modelling for optimization of structures with snap-through effects.

### **8.3. Future work**

In this study, Iso-XFEM was proposed as an alternative to current element-based evolutionary structural optimization methods. Although extensive work has been conducted by introducing the proposed method, much more work is still required to improve and extend the method to different applications. This should include:

- **Optimizing the Iso-XFEM algorithm and Matlab code:** This can significantly simplify the further extensions and improvements of the method and increase the computational efficiency of the structural optimization. Another idea could be to write a simple, short code for Iso-XFEM intended for educational purposes, similar to the 99 line SIMP code (Sigmund 2001).
- **Optimizing X-FEM integration scheme:** The X-FEM integration scheme should be optimized to increase the computational efficiency of the Iso-XFEM method, especially in the case of 3D modelling. This can be achieved by modifying the decomposition scheme in such a way that by dividing the solid domain into fewer tetrahedra to give the same accuracy and smoothness as the proposed decomposition scheme at lower computational cost.
- **Utilizing different elements for Iso-XFEM:** Although the linear quadrilateral and hexahedral elements used in this study are of a relatively high accuracy, meshing geometrically complex 3D design domains with Hex elements is sometimes cumbersome and requires pre-processing. To mesh complex geometries, higher order tetrahedral elements are preferable. Also, the use of higher order finite elements can increase the accuracy of the FE analysis during the optimization process. By using different element types one can extend the method to the topology optimization of alternative structures such as plate and

shell structures. Implementing Iso-XFEM with alternative element types may require modifying the X-FEM integration scheme.

- Applying Iso-XFEM to alternative optimization problems: As discussed in section 5.5, the application of Iso-XFEM and other evolutionary optimization methods is not just limited to stiffness design. They have the potential to be implemented on many alternative optimization problems such as maximizing natural frequencies and the optimization of heat conduction problems. Also, to include more realistic applications, multiple load cases should be considered.
- Considering different sources of non-linearity: Continuum mechanics problems may include material non-linearity, geometrical nonlinearity or both. In this study, only the topology optimization of structures with geometrical non-linearity was studied. An application of this is in the topology optimization of large-displacement, compliant mechanisms. To further extend this research, one can apply the method to topology optimization of the structures with material non-linearity and structures with both material and geometrical non-linearity. This will allow further application of the method to alternative non-linear topology optimization problems, such as the topology optimization of energy absorption structures.
- Utilizing manufacturing constraints: A major focus of this research was to improve the design boundary of the topology optimization solutions. This was to reduce the post-processing required before the design becomes manufacturable. Another significant consideration to bridge the gap between structural optimization and manufacturing is to introduce manufacturing constraints into the optimization problem. An example of this is to minimize the support structure requirements for topology optimization solutions which are seen in additive manufacturing processes such as selective laser melting and fused deposition modeling.

# List of References

- Allaire, G., Gournay, F.D., Jouve, F., & Toader, A.M. (2005). Structural optimization using topological and shape sensitivity via a level set method. *Control and cybernetics*, 34, 59-80.
- Allaire, G., Jouve, F., Toader, A. M. (2004). Structural optimisation using sensitivity analysis and a level set method, *J. Comp. Phys.*, 194, 363-393.
- Ansola, R., Veguería, E., Canales, J., & Tárrago, J. A. (2007). A simple evolutionary topology optimization procedure for compliant mechanism design. *Finite Elements in Analysis and Design*, 44(1), 53-62.
- Aremu, A. (2013). Topology optimization for additive manufacture. (Doctoral dissertation, Loughborough University).
- Aremu, A., Ashcroft, I., Wildman, R., Hague, R., Tuck, C., Brackett, D. (2013). The effects of bidirectional evolutionary structural optimization parameters on an industrial designed component for additive manufacture. *Proc IMechE Part B: J Engineering Manufacture*, 227(6), 794-807.
- Babuška, I. (1976). Solution of interface problems by homogenization. I. *SIAM Journal on Mathematical Analysis*, 7(5), 603-634.
- Bathe, K. J. (2006). *Finite element procedures*. Prentice-Hal.
- Bendsoe, M. P., & Sigmund, O. (2003). *Topology optimization: theory, methods and applications*. Springer.
- Bendsoe, M. P. (1995). *Optimization of structural topology, shape, and material*. Berlin etc: Springer.
- Bendsøe, M.P. (1989). Optimal shape design as a material distribution problem. *Struct. Optim.* 1, 193-202.
- Bendsøe, M.P. and Kikuchi, N. (1988). Generating optimal topologies in structural design using a homogenization method. *Computer Methods in Applied Mechanics and Engineering*, 71, 197-224.
- Belytschko, T., Gracie, R., & Ventura, G. (2009). A review of extended/generalized finite element methods for material modeling. *Modelling and Simulation in Materials Science and Engineering*, 17(4), 043001.
- Belytschko, T., S. P. Xiao, and C. Parimi (2003). Topology optimization with implicit functions and regularization. *International Journal for Numerical Methods in Engineering*, 57(8), 1177-1196.

- Belytschko, T. and Black, T. (1999). Elastic crack growth in finite elements with minimal remeshing. *International Journal for Numerical Methods in Engineering*, 45 (5), 601-620.
- Belytschko, T., Lu, Y. Y., & Gu, L. (1994). Element-free Galerkin methods. *International journal for numerical methods in engineering*, 37(2), 229-256.
- Bruns, T. E., & Tortorelli, D. A. (2003). An element removal and reintroduction strategy for the topology optimization of structures and compliant mechanisms. *International journal for numerical methods in engineering*, 57(10), 1413-1430
- Bruns, T. E., & Tortorelli, D. A. (1998, September). Topology optimization of geometrically nonlinear structures and compliant mechanisms. In *Proceedings of the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, 1874-1882.
- Buhl, T., Pedersen, C. B., & Sigmund, O. (2000). Stiffness design of geometrically nonlinear structures using topology optimization. *Structural and Multidisciplinary Optimization*, 19(2), 93-104.
- Bureerat, S., Limtragool, J., (2008). Structural topology optimization using simulated annealing with multiresolution design variables. *Finite Elements in Analysis and Design*, 44, 738-747.
- Burger, M., Hackl, B., & Ring, W. (2004). Incorporating topological derivatives into level set methods. *Journal of Computational Physics*, 194(1), 344-362.
- Canann, S., Stephenson, M. and Blacker, T., 1993. Optismoothing: An optimization-driven approach to mesh Smoothing. *Finite Elements in Analysis and Design*, 13, 185–190.
- Challis, V. J. (2010). A discrete level-set topology optimization code written in Matlab. *Structural and multidisciplinary optimization*, 41(3), 453-464.
- Chen, J. S., Hu, W., Puso, M. A., Wu, Y., & Zhang, X. (2007). Strain smoothing for stabilization and regularization of Galerkin meshfree methods. *Meshfree Methods for Partial Differential Equations III* (pp. 57-75). Springer Berlin Heidelberg.
- Chapman, C.D., & Jakiela, M.J. (1996). Genetic algorithm-based structural topology design with compliance and topology simplification considerations. *Journal of Mechanical Design*, 118(1), 89-98.
- Cho, S., & Kwak, J. (2006). Topology design optimization of geometrically nonlinear structures using meshfree method. *Computer methods in applied mechanics and engineering*, 195(44), 5909-5925.
- Chu, D.N., Xie, Y.M., Hira, A. and Steven, G.P.,1996. Evolutionary structural optimisation for problems with stiffness constraints. *Finite Elements in Analysis and Design*, 21, 239–51.

- Cingoski, V., Miyamoto, N., & Yamashita, H. (2000). Hybrid element-free Galerkin-finite element method for electromagnetic field computations. *Magnetics, IEEE Transactions on*, 36(4), 1543-1547.
- Cioranescu, D., & Paulin, J. S. J. (1979). Homogenization in open sets with holes. *Journal of mathematical analysis and applications*, 71(2), 590-607.
- Clough, R.W., (1960). The finite element method in plane stress analysis.
- Courant, R. (1943). Variational methods for the solution of problems of equilibrium and vibrations. *Bull. Amer. Math. Soc*, 49(1), 1-23.
- Daux, C., Moës, N., Dolbow, J., Sukumar, N., Belytschko, T. (2000). Arbitrary cracks and holes with the extended finite element method. *Int. J. Numer. Methods Engrg.*, 48 (12), 1741-1760.
- Diaz, A., & Sigmund, O. (1995). Checkerboard patterns in layout optimization. *Structural optimization*, 10(1), 40-45.
- Dolbow, J., & Belytschko, T. (1998). An introduction to programming the meshless Element Free Galerkin method. *Archives of Computational Methods in Engineering*, 5(3), 207-241.
- Du, Y., Chen, L., & Luo, Z. (2008). Topology synthesis of geometrically nonlinear compliant mechanisms using meshless methods. *Acta Mechanica Solida Sinica*, 21(1), 51-61.
- Daux, C., Moës, N., Dolbow, J., Sukumar, N., Belytschko, T., (2000). Arbitrary branched and intersecting cracks with the extended finite element method. *International Journal for Numerical Methods in Engineering*; 48, 1741–1760.
- Dunning, P., Kim, H. A. and Mullineux, G., 2008. Error analysis of fixed grid formulation for boundary based structural optimisation. In: 7th ASMO UK / ISSMO conference on Engineering Design Optimisation, 7-8 July 2008, Bath, UK.
- Duysinx, P., Van Miegroet, L., Jacobs, T., & Fleury, C. (2006). Generalized shape optimization using x-fem and level set methods. In *IUTAM Symposium on Topological Design Optimization of Structures, Machines and Materials* (pp. 23-32). Springer Netherlands.
- Eschenauer, H. A., Kobelev, V. V., & Schumacher, A. (1994). Bubble method for topology and shape optimization of structures. *Structural optimization*, 8(1), 42-51.
- Eymard, R., Gallouët, T., & Herbin, R. (2000). Finite volume methods. *Handbook of numerical analysis*, 7, 713-1018.
- Fernández-Méndez, S., & Huerta, A. (2004). Imposing essential boundary conditions in mesh-free methods. *Computer methods in applied mechanics and engineering*, 193(12), 1257-1275.
- Folgado, J., Rodrigues, H., & Guedes, J. M. (1995). Layout design of plate reinforcements with a buckling load criterion. *Structural and Multidisciplinary Optimization*, ISSMO, Goslar, Germany, Oxford: Pergamon, 659-666.

- Freitag, L., 1997. On combining Laplacian and optimization based mesh smoothing techniques. Proceedings of the 1997 Joint Summer Meeting of American Society of Mechanical Engineers (ASME) American Society of Civil Engineers (ASCE) and Society of Engineers Science (SES), 37–44.
- Gao, T., Zhang, W. H., Zhu, J. H., Xu, Y. J., & Bassir, D. H. (2008). Topology optimization of heat conduction problem involving design-dependent heat load effect. *Finite Elements in Analysis and Design*, 44(14), 805-813.
- Garcia-Ruiz, M. J., & Steven, G. P. (1999). Fixed grid finite elements in elasticity problems. *Engineering Computations*, 16(2), 145-164.
- Gea, H. C., & Luo, J. (2001). Topology optimization of structures with geometrical nonlinearities. *Computers & Structures*, 79(20), 1977-1985.
- Gerstenberger, A., Wall, W. A., 2008. An eXtended finite element method/Lagrange multiplier based approach for fluid-structure interaction. *Computer Methods in Applied Mechanics and Engineering*, 197, pp.1699-714.
- Goldberg, D. E. (1990). E.(1989). Genetic algorithms in search, optimization and machine learning. Reading: Addison-Wesley.
- Gossler, A. (2001). Moving Least-Squares: a numerical differentiation method for irregularly spaced calculation points. SANDIA Report, SAND2001-1669.
- Guest, J. K., & Smith Genut, L. C. (2010). Reducing dimensionality in topology optimization using adaptive design variable fields. *International journal for numerical methods in engineering*, 81(8), 1019-1045.
- Guo, X., Zhang, W. S., Wang, M. Y., & Wei, P. (2011). Stress-related topology optimization via level set approach. *Computer Methods in Applied Mechanics and Engineering*, 200(47), 3439-3452.
- Ha, S. H., & Cho, S. (2008). Level set based topological shape optimization of geometrically nonlinear structures using unstructured mesh. *Computers & Structures*, 86(13), 1447-1455.
- Haber, R. B., Jog, C. S., & Bendsøe, M. P. (1996). A new approach to variable-topology shape design using a constraint on perimeter. *Structural Optimization*, 11(1-2), 1-12.
- Haftka, R. T., & Gurdal, Z. (1992). Elements of structural optimization. Kluwer Academic Publishers, Netherlands.
- Haftka, R. T., & Grandhi, R. V. (1986). Structural shape optimization —a survey. *Computer Methods in Applied Mechanics and Engineering*, 57(1), 91-106.
- Hassani, B., & Hinton, E. (1998). A review of homogenization and topology optimization I—homogenization theory for media with periodic structure. *Computers & Structures*, 69(6), 707-717.
- Heyman, J. (1951). Plastic design of beams and frames for minimum material consumption. *Quarterly of Applied Mathematics*, 8, 373-381.

- Holland, J.H. (1975). *Adaptation in natural and artificial systems*. University of Michigan Press, Ann Arbor, USA.
- Huang, M.W., & Arora, J.S. (1997). Optimal design with discrete variables: some numerical experiments. *International Journal for Numerical Methods in Engineering*, 40(1), 165-188.
- Huang, X., & Xie, M. (2010a). *Evolutionary topology optimization of continuum structures: methods and applications*. John Wiley & Sons.
- Huang, X., & Xie, Y. M. (2010b). A further review of ESO type methods for topology optimization. *Structural and Multidisciplinary Optimization*, 41(5), 671-683.
- Huang, X., & Xie, Y.M. (2010c). Evolutionary topology optimization of continuum structures with an additional displacement constraint. *Structural and Multidisciplinary Optimization*, 40(1-6), 409-416.
- Huang, X., & Xie, Y. M. (2009). Bi-directional evolutionary topology optimization of continuum structures with one or multiple materials. *Computational Mechanics*, 43(3), 393-401.
- Huang, X., & Xie, Y. M. (2008). Topology optimization of nonlinear structures under displacement loading. *Engineering structures*, 30(7), 2057-2068.
- Huang, X. and Xie, Y.M. (2007a). Convergent and mesh-independent solutions for bi-directional evolutionary structural optimization method. *Finite Elem Anal Des*, 43, 1039-1049.
- Huang, X. H., & Xie, Y. (2007b). Bidirectional evolutionary topology optimization for structures with geometrical and material nonlinearities. *AIAA journal*, 45(1), 308-313.
- Huang, X. H., & Xie, Y. (2007c). Bidirectional evolutionary topology optimization for structures with geometrical and material nonlinearities. *AIAA journal*, 45(1), 308-313.
- Huang, X., Zuo, Z. H., & Xie, Y. M. (2010). Evolutionary topological optimization of vibrating continuum structures for natural frequencies. *Computers & structures*, 88(5), 357-364.
- Iyengar, R.N., & Jagadish, K.S. (2005). *Recent Advances in Structural Engineering*. Universities Press.
- Jakiela, M., Chapman, C., Duda, J., Adewuya, A., Saitou, K. (1999). Continuum structural topology design with genetic algorithms. *Comput Methods Appl Mech Eng*, 186(2), 339-356.
- Jia, H., Beom, H. G., Wang, Y., Lin, S., & Liu, B. (2011). Evolutionary level set method for structural topology optimization. *Computers & Structures*, 89(5), 445-454.

- Jog, C. (1996). Distributed-parameter optimization and topology design for non-linear thermoelasticity. *Computer Methods in Applied Mechanics and Engineering*, 132(1), 117-134.
- Jog, C.S. and Harber, R.B., 1996. Stability of finite element models for distributed-parameter optimization and topology design. *Comput. Meth. Appl. Mech. Engng.* 130: 1951–65.
- Juan, Z., Shuyao, L., & Guangyao, L. (2010). The topology optimization design for continuum structures based on the element free Galerkin method. *Engineering Analysis with Boundary Elements*, 34(7), 666-672.
- Kane, C., & Schoenauer, M. (1996). Topological optimum design using genetic algorithms. *Control and Cybernetics*, 25, 1059-1088.
- Kaveh, A., Hassani, B., Shojaee, S., Tavakkoli, S.M., (2008). Structural topology optimization using ant colony methodology. *Engineering Structures*, 30, 2559-2566, 2008.
- Kim, I.Y., & De Weck, O.L. (2005). Variable chromosome length genetic algorithm for progressive refinement in topology optimization. *Structural and Multidisciplinary Optimization*, 29(6), 445-456.
- Kim, H., Garcia, M. J., Querin, O. M., Steven, G. P., & Xie, Y. M. (2000). Introduction of fixed grid in evolutionary structural optimisation. *Engineering Computations*, 17(4), 427-439.
- Kočvara, M. (1997). Topology optimization with displacement constraints: a bilevel programming approach. *Structural optimization*, 14(4), 256-263.
- Kuo, Y.L., Cleghorn, W.L., Behdinan, K., Fenton, R.G., (2006). The h-p-r-refinement finite element analysis of a planar high-speed four-bar mechanism. *Mechanism and Machine Theory*, 41, 505-524.
- Lancaster, P., & Salkauskas, K. (1981). Surfaces generated by moving least squares methods. *Mathematics of computation*, 37(155), 141-158.
- Lee, D. K., Starossek, U., & Shin, S. M. (2009). Topological Shape Optimum Design of Structures via X-FEM and Level Set Method. *Journal of Solid Mechanics and Materials Engineering*, 3(6), 887-897.
- Lee, D., Park, S., Shin, S., (2007). Node-wise topological shape optimum design for structural reinforced modeling of Michell-type concrete deep beams. *J Solid Mech Mater Eng*, 1(9), 1085-96.
- Lee, N. S., & Bathe, K. J. (1994). Error indicators and adaptive remeshing in large deformation finite element analysis. *Finite Elements in Analysis and Design*, 16(2), 99-139.
- Leiva, J. P., Watson, B. C., & Kosaka, I. (2007). A comparative study of topology and topometry structural optimization methods within the genesis software. In 7th World Congress of Structural and Multidisciplinary Optimization.



- Li, L., Wang, M.Y., Wei, P. (2012). XFEM schemes for level set based structural optimization. *Front. Mech. Eng.*, 7(4), 335–356.
- Li, S., & Liu, W. K. (2002). Meshfree and particle methods and their applications. *Applied Mechanics Reviews*, 55(1), 1-34.
- Li, Q., Steven, G. P., Querin, O. M., & Xie, Y. M. (1999). Shape and topology design for heat conduction by evolutionary structural optimization. *International Journal of Heat and Mass Transfer*, 42(17), 3361-3371.
- Liu, W. K., Jun, S., & Zhang, Y. F. (1995). Reproducing kernel particle methods. *International journal for numerical methods in fluids*, 20(8-9), 1081-1106.
- Lu, Y. Y., Belytschko, T., & Gu, L. (1994). A new implementation of the element free Galerkin method. *Computer methods in applied mechanics and engineering*, 113(3), 397-414.
- Luh, G.C., Lin, C.Y., & Lin, Y.S. (2011). A binary particle swarm optimization for continuum structural topology optimization. *Applied Soft Computing*, 11(2), 2833-2844.
- Luh, G., Lin, C., (2009). Structural topology optimization using ant colony optimization algorithm, *Applied Soft Computing*. 9, 1343-1353.
- Luo, Z., Zhang, N., Wang, Y., & Gao, W. (2013). Topology optimization of structures using meshless density variable approximants. *International Journal for Numerical Methods in Engineering*, 93(4), 443-464.
- Luo, Z., Zhang, N., Gao, W., & Ma, H. (2012). Structural shape and topology optimization using a meshless Galerkin level set method. *International Journal for Numerical Methods in Engineering*, 90(3), 369-389.
- Luo, Z., & Tong, L. (2008). A level set method for shape and topology optimization of large-displacement compliant mechanisms. *International Journal for Numerical Methods in Engineering*, 76(6), 862-892.
- Maute, K., Ramm, E. (1995). Adaptive topology optimisation. *Struct Optim*, 10, 100-12.
- Melenk, J.M. and *Babuška*, I., 1996. The partition of unity finite element method: Basic theory and applications. *Computer Methods in Applied Mechanics and Engineering*, 139, pp. 289-314.
- Michell, A.G.M. (1904). The limit of economy of material in frame-structures. *Phil. Mag.*, 8, 589-597.
- Van Miegroet, L., & Duysinx, P. (2009). 3D shape optimization with X-FEM and a level set constructive geometry approach. *Proceeding of the 8th World Congress on Structural and Multidisciplinary Optimization*.
- Miegroet, L.V., Duysinx, P., 2007. Stress concentration minimization of 2D Filets using X-FEM and level set description. *Structural and Multidisciplinary Optimisation*, 33, pp. 425-38.

Van Miegroet, L., Moës, N., Fleury, C., & Duysinx, P. (2005). Generalized shape optimization based on the level set method. Proceedings of the 6th World Congress of Structural and Multidisciplinary Optimization (WCSMO6).

Moës, N., Dolbow, J. and Belytschko, T. (1999). A finite element method for crack growth without remeshing, *International Journal for Numerical Methods in Engineering*, 46, 131-150.

Nayroles, B., Touzot, G., & Villon, P. (1992). Generalizing the finite element method: diffuse approximation and diffuse elements. *Computational mechanics*, 10(5), 307-318.

Newman, T. S., & Yi, H. (2006). A survey of the marching cubes algorithm. *Computers & Graphics*, 30(5), 854-879.

Nikishkov, G. P. (2004). Introduction to the finite element method. University of Aizu.

Norato, J.A., Bendsøe, M.P., Haber, R.B., & Tortorelli, D.A. (2007). A topological derivative method for topology optimization. *Structural and Multidisciplinary Optimization*, 33(4-5), 375-386.

Osher, S. J., & Santosa, F. (2001). Level set methods for optimization problems involving geometry and constraints: I. Frequencies of a two-density inhomogeneous drum. *Journal of Computational Physics*, 171(1), 272-288.

Osher, S., & Sethian, J. A. (1988). Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *Journal of computational physics*, 79(1), 12-49.

Pedersen, C. B., Buhl, T., & Sigmund, O. (2001). Topology synthesis of large-displacement compliant mechanisms. *International Journal for numerical methods in engineering*, 50(12), 2683-2705.

Petersson, J., & Sigmund, O. (1998). Slope constrained topology optimization. *International Journal for Numerical Methods in Engineering*, 41(8), 1417-1434.

Prager, W. and Rozvany, G.I.N. (1977). Optimization of structural geometry. *Dynamic Systems, Academic Press, New York*, 265-293.

Prager, W. and Shield, R.T. (1968), Optimal design of multi-purpose structures, *Int. J. Solids and Structures*, 4, 469-475.

Prager, W. and Taylor J.E. (1968). Problems of optimal structural design. *Journal of Applied Mechanics*, 35, 102-106.

Querin, O.M., Steven, G.P. and Xie, Y.M., (2000). Evolutionary Structural optimization using an additive algorithm. *Finite Element in Analysis and Design*, 34, 291-308.

Querin, O.M., Steven, G.P. and Xie, Y.M., 1998. Evolutionary structural optimisation (ESO) using a bidirectional algorithm. *Engineering Computations*, 15 (8), pp. 1031-1048.

- Querin, O. M. (1997). Evolutionary structural optimisation: stress based formulation and implementation (Doctoral dissertation, Department of Aeronautical Engineering, University of Sydney, Australia).
- Rao, S. S., (2011). The finite element method in engineering, 5ed., Elsevier, Boston.
- Ramm, E., Maute, K., & Schwarz, S. (1998). Adaptive topology and shape optimization. Computational Mechanics, New Trends and Applications, Barcelona, Spain.
- Ritz, W. (1909). Über eine neue Methode zur Lösung gewisser Variationsprobleme der mathematischen Physik. Journal für die reine und angewandte Mathematik, 135, 1-61.
- Rozvany, G. I. (2009). A critical review of established methods of structural topology optimization. Structural and Multidisciplinary Optimization, 37(3), 217-237.
- Rozvany, G. I. N. (1992). Optimal layout theory: analytical solutions for elastic structures with several deflection constraints and load conditions. Structural optimization, 4(3-4), 247-249.
- Rozvany, G.I.N. & Birker, T. (1994). On singular topologies in exact layout optimization. Struct. Optim. 8, 228–235.
- Rozvany, G. I. N., Zhou, M., & Birker, T. (1992). Generalized shape optimization without homogenization. Structural Optimization, 4(3-4), 250-252.
- Rozvany, G. I. N. (1972). Grillages of maximum strength and maximum stiffness. International Journal of Mechanical Sciences, 14(10), 651-666.
- Ryoo, J., Hajela, P. (2004). Handling variable string lengths in GA-based structural topology optimization. Struct Multidisc Optim, 26, 318-325.
- Sandgren, E., Jensen, E., & Welton, J. (1990). Topological design of structural components using genetic optimization methods. Sensitivity analysis and optimization with numerical methods, 115, 31-43.
- Sethian, J. A., & Wiegmann, A. (2000). Structural boundary design via level set and immersed interface methods. Journal of computational physics, 163(2), 489-528.
- Sethian, J. A. (1999). Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science (Vol. 3). Cambridge university press.
- Sigmund, O., 2001. A 99 line topology optimisation code written in Matlab. Struct Multidiscipl Optim, 21, pp. 120–127.
- Sigmund, O. and Petersson, J. (1998). Numerical instabilities in topology optimization: A survey on procedures dealing with checkerboards, mesh-dependencies and local minima. Struct. Optim. 16, 68-75.

- Sigmund, O. (2007). Morphology-based black and white filters for topology optimization. *Structural and Multidisciplinary Optimization*, 33(4-5), 401-424.
- Sigmund, O. (1994). Design of material structures using topology optimization (Doctoral dissertation, Technical University of Denmark).
- Sukumar, N., Chopp, D. L., Moës, N. and Belytschko, T. (2001). Modeling Holes and Inclusions by Level Sets in the Extended Finite Element Method. *Computer Methods in Applied Mechanics and Engineering*, 190, 6183-6200.
- Suzuki, K., & Kikuchi, N. (1991). A homogenization method for shape and topology optimization. *Computer methods in applied mechanics and engineering*, 93(3), 291-318.
- Svanberg, K. (1987). The method of moving asymptotes—a new method for structural optimization. *International journal for numerical methods in engineering*, 24(2), 359-373.
- Tait, K., Fenner, R.T., (1999). Optimum shape and topology design using the boundary element method. *International Journal of Solids and Structures*, 36(114), 2021-2040.
- van Dijk, N. P., Maute, K., Langelaar, M., & Van Keulen, F. (2013). Level-set methods for structural topology optimization: a review. *Structural and Multidisciplinary Optimization*, 48(3), 437-472.
- Verani, M., Bruggi, M., Cini, C., & Venini, P. (2010). Topology optimization and mesh adaptivity. In XVIII GIMC Conference.
- Van Miegroet, L., & Duysinx, P. (2009). 3D shape optimization with X-FEM and a level set constructive geometry approach. *Proceeding of the 8th World Congress on Structural and Multidisciplinary Optimization*.
- Victoria, M., Martí, P., Querin, O.M. (2009). Topology design of two-dimensional continuum structures using isolines. *Computer and Structures*, 87, 101-109.
- Victoria, M., Querin, O.M., Martí, P. (2010). Topology design for multiple loading conditions of continuum structures using isolines and isosurfaces. *Finite Elements in Analysis and Design*, 46, 229-237.
- Villanueva, C. H., & Maute, K. (2014). Density and level set-XFEM schemes for topology optimization of 3-D structures. *arXiv preprint arXiv:1401.6475*.
- Vollmer, J., Mencl, R. and Müller, H., 1999. Improved Laplacian Smoothing of Noisy Surface Meshes. *Computer Graphic Forum*, 18(3), pp. 131-138.
- Wang, S. (2007). Krylov subspace methods for topology optimization on adaptive meshes. PhD Thesis, University of Illinois, Urbana-Champaign.
- Wang, S. Y., & Tai, K. (2005). Structural topology design optimization using genetic algorithms with a bit-array representation. *Computer methods in applied mechanics and engineering*, 194(36), 3749-3770.

- Wang, Y. (2003). A Study on Microstructures of Homogenization for Topology Optimization (Doctoral dissertation, Victoria University).
- Wang, M.Y., Wang, X., Guo, D. (2003). A level set method for structural topology optimisation. *Comput. Meth .Appl. Eng.*, 192, 227-46.
- Wei, P., Wang, M.Y., Xing, X. (2010). A study on X-FEM in continuum structural optimization using level set method. *Computer-Aided Design*, 42, 708-719.
- Xie, Y. M. and Steven, G.P. (1997). *Evolutionary structural optimization*, London.
- Xie, Y. M., & Steven, G. P. (1994). A simple approach to structural frequency optimization. *Computers & structures*, 53(6), 1487-1491.
- Xie, Y. M. and Steven, G.P. (1993). A simple evolutionary procedure for structural optimization. *Computers & Structures*, 49, 885-896.
- Xie, Y.M. and Steven, G.P. (1992). Shape and layout optimisation via an evolutionary procedure. *Proceedings of International Conference on Computational Engineering Science*, Hong Kong.
- Yang, X.Y., Xie, Y.M., & Steven, G.P. (2005). Evolutionary methods for topology optimisation of continuous structures with design dependent loads. *Computers & structures*, 83(12), 956-963.
- Yang, X. Y., Xie, Y. M., Steven, G. P., and Querin, O. M., (1999). Bidirectional evolutionary method for stiffness optimisation. *AIAA J.*, 37(11), pp.1483–1488.
- Yang, X.Y., Xie, Y.M., Steven, G.P., & Querin, O.M. (1999). Topology optimization for frequencies using an evolutionary method. *Journal of Structural Engineering*, 125(12), 1432-1438.
- Zhou, J. X., & Zou, W. (2008). Meshless approximation combined with implicit topology description for optimization of continua. *Structural and Multidisciplinary Optimization*, 36(4), 347-353.
- Zhou, M., & Rozvany, G.I.N. (2001). On the validity of ESO type methods in topology optimization. *Structural and Multidisciplinary Optimization*, 21(1), 80-83.
- Zhou, M., Rozvany, G.I.N. (1991). The COG algorithm, Part II: Topological, geometrical and general shape optimisation. *Comp. Meth. Appl. Mech. Eng.*, 89, 309-336.
- Zienkiewicz, O.C., Taylor, R.L., Zhu, J.Z., (2005). *The Finite Element Method: Its Basis and Fundamentals*, Butterworth-Heinemann, 6th ed.
- Zienkiewicz, O. C., & Cheung, Y. K. (1967). *The finite element method in structural and continuum mechanics: numerical solution of problems in structural and continuum mechanics (Vol. 1)*. London, New York: McGraw-Hill.
- Zuo, Z. H., Xie, Y. M., & Huang, X. (2009). Combining genetic algorithms with BESO for topology optimization. *Structural and Multidisciplinary Optimization*, 38(5), 511-523.

# Appendix A:

## Matlab Code for 2D Version of Iso-XFEM Method

The Matlab code presented in this appendix is the 2D version of the proposed Iso-XFEM method with the modifications presented in chapter 5, implemented for topology optimization of a rectangular design domain. The code accepts the geometric properties of the design domain including length (L), height (h) and thickness (t), number of grids in x and y (nelx & nely), material properties including Young's modulus (E) and Poisson's ratio (nu), applied force (f), as well as the optimization parameters including volume evolution rate (er) and volume constraint factor (volfrac).

```
##### 2D Iso-XFEM topology optimization code by Meisam Abdi,
University of Nottingham #####
% This code generates topology optimization solutions for 2D
structures using Iso-XFEM method.
function IsoXFEM2D(L,h,t,nelx,nely,er,volfrac,E,nu,f)

% L: length, h: height, t: thickness
% nelx: number of elements in x direction
% nely: number of elements in y direction
% er: volume evolution rate
% volfrac: volume constraint
% E: Young's modulus, nu: Poisson's ratio
% f: magnitude of load
% IsoXFEM2D(100,50,1,60,30,0.01,0.5,1,0.3,1) % example

[con,coord]=concoord(nelx,nely,L,h); % Element connectivity and
Nodes' coordinate matrix
nel=size(con,1);nnd=size(coord,1); %number of nodes and elements
D=E/(1-nu^2)*[1,nu,0;nu,1,0;0,0,(1-nu)/2]; % Element elasticity
matrix

% Force and Boundary Conditions
F = sparse(2*nnd,1);
```

```

bottomend=2*(nelx+1)*(nely+1); %DoF number of the bottom of the
free end
middleend=2*(nelx+1)*(nely+1) -nely; %DoF number of the middle of
the free end
fdof= bottomend;
F(fdof)=f;

U=sparse(2*nnd,1); %Displacement DoFs
fixeddofs=[1:2*(nely+1)];
alldofs    = [1:2*nnd];
freedofs   = setdiff(alldofs,fixeddofs);

%elements connected to each node
ncon=zeros(nnd,6);
for i=1:nnd
    [row,col] = find(con==i);
    d=size(row,1);
    ncon(i,1:d)=row;
end

%area and stiffness of the elements
Ael=zeros(nel,1); % initial elements' area
Kel=zeros(8,8,nel); % initial elements' stiffness matrix
for el=1:nel
    elcoord=coord(con(el,:)',' );
    Ael(el,1)=polyarea(elcoord(:,1),elcoord(:,2));
    Kel(:, :, el)=stiffnessmat(D,t,el,coord,con);
end

A=sum(sum(Ael)); % area of the whole structure
Ke=Kel;Ae=Ael; % updated elements' stiffness matrix and area

for it=1:200

if it>1; ndcold=ndc; end

[K]=stiffness(nel,nnd,con,Ke);
U(freedofs,1)=K(freedofs,freedofs)\F(freedofs,1);

ese=zeros(nel,1);dc=ese; % vector of element strain energies and
element structural performances
for el = 1:nel
    edof = [2*con(el,1)-1; 2*con(el,1); 2*con(el,2)-1;
2*con(el,2); 2*con(el,3)-1; 2*con(el,3); 2*con(el,4)-1;
2*con(el,4)];
    Ue=U(edof,1);
    ese(el,1)=0.5*Ue'*Ke(:, :, el)*Ue;
    dc(el,1) = ese(el,1)/Ael(el,1);
end
SE=sum(sum(ese)); % objective (total strain enrgy)

ndc=zeros(nnd,1); % nodal values of structural performance
for i=1:nnd
    a=find(ncon(i,:)>0);
    b=dc(ncon(i,a),1);
    ndc(i,1)=sum(b)/max(4,numel(a));
end

```

```

if it>1; ndc=(ndc+ndcold)/2; end % stabilizing the evolutionary
process

vf(it)=sum(sum(Ae))/A; % volume fraction of the design for the
current iteration

% finding a minimum level of performance (MLP) which
% results in the target volume of the current iteration
if it == 1; smax=max(max(ndc)); vfi=1; er=0.01; MLP=0.99*smax; end
vfi=max(volfrac, vfi*(1-er)); vfk=0; % target volume fraction of the
current iteration
while abs(vfi-vfk)/vfi>0.001
    rcr=ndc-MLP*ones(size(ndc)); % relative performance
    Ve=solidarea(Ael, con, rcr, nel);
    vfk=sum(Ve)/A;
    MLP=MLP*vfk/vfi; % updating MPL
end

% updating elements' stiffness matrix (KE) using X-FEM
xcoord=coord; xcon=con;
voidel=[]; solel=[]; bel=[];
for el=1:nel
    [KE, AE, elshape, addcoord]=
elm_prop(con, coord, rcr, el, Kel, Ael, D, t);
    Ke(:, :, el)=KE; Ae(el, 1)=AE;
    if elshape==1
        solel=[solel; el]; %solid elements
    elseif elshape==0
        voidel=[voidel; el]; % void elements
    else
        bel=[bel; el]; % boundary elements
        xcon=[xcon; elshape+size(xcoord, 1)];
        xcoord=[xcoord; addcoord];
    end
end

solcon=xcon(solel, :); boncon=xcon(nel+1:end, :);
xsol=zeros(size(solcon)); ysol=zeros(size(solcon));
xb=zeros(size(solcon)); yb=zeros(size(solcon));
for i=1:size(solcon, 1)
    for j=1:4
        xsol(j, i)=xcoord(solcon(i, j), 1);
        ysol(j, i)=xcoord(solcon(i, j), 2);
    end
end
for i=1:size(boncon, 1)
    for j=1:4
        xb(j, i)=xcoord(boncon(i, j), 1);
        yb(j, i)=xcoord(boncon(i, j), 2);
    end
end

% visualization of the topology
figure(it)
h1=patch(xsol, ysol, 'w'); axis equal; axis([min(coord(:, 1))
max(coord(:, 1)) min(coord(:, 2)) max(coord(:, 2))]); axis off;
h2=patch(xb, yb, 'w'); axis equal; axis([min(coord(:, 1))
max(coord(:, 1)) min(coord(:, 2)) max(coord(:, 2))]); axis off;
set(h1, 'LineWidth', 1)
set(h2, 'EdgeColor', 'r')

```



```

set(h2,'LineWidth',1)
set(h1,'FaceColor',[.9 .9 .9])
set(h2,'FaceColor',[.9 .9 .9])
hgsave(num2str(it))

disp([' It.: ' sprintf('%4i',it) ' Obj.: ' sprintf('%10.4f',SE) '
Vol.: ' sprintf('%6.3f',vf(it) )]);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% connectivity and coordinate
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [connectivity,coord]=concoord(nelx,nely,L,h)
connectivity=zeros(nelx*nely,4);
for ii=1:nelx*nely
    rw=mod(ii,nely);
    cl=fix((ii-1)/nely)+1;
    connectivity(ii,1)=cl-1+ii;
    connectivity(ii,4)=connectivity(ii,1)+1;
    connectivity(ii,2)=connectivity(ii,1)+nely+1;
    connectivity(ii,3)=connectivity(ii,2)+1;
end

coord=zeros((nelx+1)*(nely+1),2);
for ii=1:(nelx+1)*(nely+1)
    coord(ii,1)=fix((ii-1)/(nely+1))*L/nelx;
    coord(ii,2)=mod((ii-1),(nely+1))*h/nely;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Element Stiffness Matrix %%%%%%%%%%
function [ke]=stiffnessmat(D,t,en,coord,con)

GP=[-1/sqrt(3),-1/sqrt(3);1/sqrt(3),-
1/sqrt(3);1/sqrt(3),1/sqrt(3);-1/sqrt(3),1/sqrt(3)];

i=con(en,1);j=con(en,2);k=con(en,3);l=con(en,4);
x01=coord(i,1);y01=coord(i,2);
x02=coord(j,1);y02=coord(j,2);
x03=coord(k,1);y03=coord(k,2);
x04=coord(l,1);y04=coord(l,2);

ke=zeros(8,8);
for i=1:4

    r=GP(i,1);s=GP(i,2);

    dN1r=-1/4*(1-s);dN1s=-1/4*(1-r);
    dN2r=1/4*(1-s);dN2s=-1/4*(1+r);
    dN3r=1/4*(1+s);dN3s=1/4*(1+r);
    dN4r=-1/4*(1+s);dN4s=1/4*(1-r);

    j11=x01*dN1r+x02*dN2r+x03*dN3r+x04*dN4r;
    j12=y01*dN1r+y02*dN2r+y03*dN3r+y04*dN4r;
    j21=x01*dN1s+x02*dN2s+x03*dN3s+x04*dN4s;
    j22=y01*dN1s+y02*dN2s+y03*dN3s+y04*dN4s;
    J=[j11 j12;j21 j22];detj=det(J);

    dNxy=inv(J)*[dN1r dN2r dN3r dN4r; dN1s dN2s dN3s dN4s];
    dN1x=dNxy(1,1);dN2x=dNxy(1,2);dN3x=dNxy(1,3);dN4x=dNxy(1,4);

```

```

dN1y=dNxy(2,1);dN2y=dNxy(2,2);dN3y=dNxy(2,3);dN4y=dNxy(2,4);

BL0=[dN1x 0 dN2x 0 dN3x 0 dN4x 0
      0 dN1y 0 dN2y 0 dN3y 0 dN4y
      dN1y dN1x dN2y dN2x dN3y dN3x dN4y dN4x];

ke=ke+t*BL0'*D*BL0*detj;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [K]=stiffness(nel,nnd,con,Ke)
K = sparse(2*nnd, 2*nnd);
for el = 1:nel
    edof = [2*con(el,1)-1; 2*con(el,1); 2*con(el,2)-1;
            2*con(el,2); 2*con(el,3)-1; 2*con(el,3); 2*con(el,4)-1;
            2*con(el,4)];
    K(edof,edof) = K(edof,edof) + Ke(:, :, el);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% estimation of the element areas using nodes' relative
performance numbers
function [Ae]= solidarea(Ae1,con,rcr,nel)
for en=1:nel
if min(rcr(con(en,:),1)) > 0 % the case that the element is inside
the boudary
    AreaRatio=1;
elseif max(rcr(con(en,:),1)) < 0
    AreaRatio=0;% the case that the element is outside the boudary
else% the case that the element is cut by the boudary
    [ s , t ] = meshgrid([-1 : 0.1 : 1],[-1 : 0.1 : 1]);
    tmpPhi = (1 - s(:)).*(1 - t(:))/4 * rcr(con(en,1)) + (1 +
s(:)).*(1 - t(:))/4 * rcr(con(en,2)) ...
    + (1 + s(:)).*(1 + t(:))/4 * rcr(con(en,3)) + (1-s(:)).*(1 +
t(:))/4 * rcr(con(en,4));
    AreaRatio = length(find( tmpPhi >= 0 ))/length(s(:));
end;
Ae(en,1)=AreaRatio*Ae1(en,1);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Dividing the boundary elements into sub-triangles
function [KE,AE,elshape,addcoord]=
elm_prop(con,coord,rcr,en,Kel,Ae1,D,t)
KE1=Kel(:, :, en);AE1=Ae1(en,1);
if min(rcr(con(en,:),1)) >= 0 % the case that the element is
inside the boudary
    KE = KE1;AE=AE1;elshape=1;addcoord=[];
elseif max(rcr(con(en,:),1)) <= 0 % the case that the element is
outside the boudary
    KE = KE1*0.0001;AE=0;elshape=0;addcoord=[];
else% the case that the element is cut by the boudary
    elrcr=rcr(con(en,:),1); %relative criteria level of nodes of
the element
    elcoord=coord(con(en,:), :);
    NND=find(elrcr<0); %nodes with negative rcr
    PND=find(elrcr>0);%nodes with positive rcr
    elm=[1 2 3 4 1];

```

```

    nbnodes=[4 2;1 3;2 4;3 1]; % neighboring nodes, row no is equal
to negative node no.

    relrcr=[elrcr;elrcr(1,1)]; % rcr for [1 2 3 4 1]
    ni=0; % no of intersection points
    for i=1:4
        if relrcr(i,1)*relrcr(i+1,1)<0
            ni=ni+1;
            intp(ni,1:2)=[elm(i),elm(i+1)]; % nodes which have a
boundary intersection in between
        end
    end

    for i=1:size(intp,1) %finding the location of intersection
points
        x1=elcoord(intp(i,1),:);x2=elcoord(intp(i,2),:);rel=abs(elrcr(intp
(i,1),1)/elrcr(intp(i,2),1));
        intl(i,:)=x1+(x2-x1)*rel/(1+rel);
    end

    nelcoord=[elcoord;intl];
    nelrcr=[elrcr;zeros(ni,1)];
    nelm=elm;elrcr=[elrcr;elrcr(end,1)];
    ts=0;
    for i=1:4
        if relrcr(i,1)*relrcr(i+1,1)<0

aa=find(intp(:,1)==elm(1,i));bb=find(intp(:,2)==elm(1,i+1));

cc=find(intp(:,1)==elm(1,i+1));dd=find(intp(:,2)==elm(1,i));
        if aa==bb;ip=aa+4;else ip=cc+4;end
        nelm=[nelm(1,1:i+ts)';ip;nelm(i+ts+1:end)'];
        ts=ts+1;
    end
end

    a=find(nelrcr(nelm(1,:),1)>=0);
    pelnd=nelm(1,a); % positive nodes + intersection nodes of the
element
    if pelnd(1,1)~= pelnd(1,end);pelnd=[pelnd,pelnd(1,1)]; end %
loop positive nodes
    nelcoord=[nelcoord;mean(nelcoord(pelnd,:))]; %defining a
centre point for the element to construct the triangles
    triang=zeros(size(pelnd,2)-1,3);
    for i=1:size(pelnd,2)-1
        triang(i,:)=[pelnd(1,i),pelnd(1,i+1),size(nelcoord,1)];

triarea(i,1)=polyarea(nelcoord(triang(i,:)',1),nelcoord(triang(i,:
) ',2));
    end

    AE=sum(sum(triarea));
    AreaRatio=AE/AE1;

    KE =K_XFEM(D,t,en,coord,con,triang,nelcoord); % X-FEM

    elshape=triang;elshape(:,4)=triang(:,3);
    addcoord=nelcoord;
end

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% X-FEM scheme: calculate element stiffness by integration over
% subtriangles
function [k0]=K_XFEM(D,t,en,coord,con,triang,tricoord)

i=con(en,1);j=con(en,2);k=con(en,3);l=con(en,4);
x01=coord(i,1);y01=coord(i,2);
x02=coord(j,1);y02=coord(j,2);
x03=coord(k,1);y03=coord(k,2);
x04=coord(l,1);y04=coord(l,2);

% calculate the transformation matrix
A=[x01 y01 x01*y01 1; x02 y02 x02*y02 1; x03 y03 x03*y03 1; x04
y04 x04*y04 1];
B=[-1 -1 1 1; 1 -1 -1 1; 1 1 1 1; -1 1 -1 1];
TM=A\B;

% calculate natural coordinates of triangle nodes
gcoord=[tricoord tricoord(:,1).*tricoord(:,2)
ones(size(tricoord,1),1)];
natcoord=gcoord*TM(:,1:2);

% calculate gauss points of the triangles in natural coordinates
tcoor = zeros(3,2,size(triang,1));
for ij=1:size(triang,1)
    tcoor(:, :, ij)=natcoord(triang(ij, :)', :);
    tarea(ij,1)=polyarea(tcoor(:,1,ij),tcoor(:,2,ij));
end

GP = zeros(3,2,size(triang,1));% Gauss Points
GP(1, :, :)=(tcoor(1, :, :)+tcoor(2, :, :))/2;
GP(2, :, :)=(tcoor(2, :, :)+tcoor(3, :, :))/2;
GP(3, :, :)=(tcoor(3, :, :)+tcoor(1, :, :))/2;

k0=zeros(8,8);
for i=1: numel(tarea)
    for j=1:3
        r= GP(j,1,i);s=GP(j,2,i) ;

        N1=1/4*(1-r)*(1-s);
        N2=1/4*(1+r)*(1-s);
        N3=1/4*(1+r)*(1+s);
        N4=1/4*(1-r)*(1+s);

        dN1r=-1/4*(1-s);dN1s=-1/4*(1-r);
        dN2r=1/4*(1-s);dN2s=-1/4*(1+r);
        dN3r=1/4*(1+s);dN3s=1/4*(1+r);
        dN4r=-1/4*(1+s);dN4s=1/4*(1-r);

        j11=x01*dN1r+x02*dN2r+x03*dN3r+x04*dN4r;
        j12=y01*dN1r+y02*dN2r+y03*dN3r+y04*dN4r;
        j21=x01*dN1s+x02*dN2s+x03*dN3s+x04*dN4s;
        j22=y01*dN1s+y02*dN2s+y03*dN3s+y04*dN4s;
        J=[j11 j12;j21 j22];detj=det(J);

        dNxy=inv(J)*[dN1r dN2r dN3r dN4r; dN1s dN2s dN3s dN4s];

dN1x=dNxy(1,1);dN2x=dNxy(1,2);dN3x=dNxy(1,3);dN4x=dNxy(1,4);

```

```
dN1y=dNxy(2,1);dN2y=dNxy(2,2);dN3y=dNxy(2,3);dN4y=dNxy(2,4);

BL0=[dN1x 0 dN2x 0 dN3x 0 dN4x 0
      0 dN1y 0 dN2y 0 dN3y 0 dN4y
      dN1y dN1x dN2y dN2x dN3y dN3x dN4y dN4x];

k0=k0+t*BL0'*D*BL0*detj/3*tarea(i,1);
end
end
```

# Appendix B:

## Matlab Code for 3D version of Iso-XFEM Method

The Matlab code presented in this appendix is the 3D version of the proposed Iso-XFEM method presented in chapter 5, implemented for topology optimization of a cuboid design domain. The code accepts the geometric properties of the design domain including length (L), width (W) and height (h), number of grids in x, y and z (nelx, nely & nelz), material properties including Young's modulus (E) and Poisson's ratio (nu), applied force (f), as well as the optimization parameters including volume evolution rate (er) and volume constraint factor (volfrac).

```

%%% 3D Iso-XFEM topology optimization code by Meisam Abdi,
University of Nottingham %%%
% This code generates topology optimization solutions for 3D
structures
% usin Iso-XFEM method.

function IsoXFEM3D(L,W,h,nelx,nely,nelz,volfrac,E,NU,er,f)

% L: length, W: width, h: height
% nelx: number of elements in x direction
% nely: number of elements in y direction
% nelz: number of elements in z direction
% er: volume evolution rate
% volfrac: volume constraint
% E: Young's modulus, nu: Poisson's ratio
% f: magnitude of load
% IsoXFEM3D(40,2,20,20,2,10,0.5,1,0.3,0.01,1) % example

[con,coord]=concoord3D(L,W,h,nelx,nely,nelz); % Element
connectivity and Nodes' coordinate matrix
nel=size(con,1);nnd=size(coord,1); %number of nodes and elements

% Element elasticity matrix
D = (E/((1+NU)*(1-2*NU)))*[1-NU NU NU 0 0 0 ; NU 1-NU NU 0 0 0 ;
NU NU ...
1- NU 0 0 0 ;
0 0 0 (1-2*NU)/2 0 0 ; 0 0 0 0 (1- 2*NU)/2 0 ; 0 0 0 0 0 (1-
2*NU)/2];

```

```

% Display and save the start mesh (full solid design domain)
surfcon=[];
for i=1:nel
    surfcon(end+1:end+6,1:4)=[con(i,1:4);con(i,5:8);con(i,[1
4 8 5])
    con(i,[2 3 7 6]);con(i,[4 3 7 8]);con(i,[1 2 6 5])];
end
figure (1)
for i=1:size(surfcon,1)
    for j=1:4
        xcoord(j,i)=coord(surfcon(i,j),1);
        ycoord(j,i)=coord(surfcon(i,j),2);
        zcoord(j,i)=coord(surfcon(i,j),3);
    end
end
h=patch(xcoord,ycoord,zcoord,'w');axis equal;
axis([min(coord(:,1)) max(coord(:,1)) min(coord(:,2))
max(coord(:,2)) min(coord(:,3)) max(coord(:,3))]);axis off;
set(h,'LineWidth',1.5)
set(h,'FaceColor',[.85 .85 .85])
hgsave('0')

% Force and Boundary Conditions
U=sparse(3*nnd,1);
F = sparse(3*nnd,1);
fixeddofs=[1:3*(nely+1)*(nelz+1)]; % Fixed DoF
fdof=3*(nelx*(nely+1)*(nelz+1) + (nely/2+1)*(nelz+1)); % DoF of
the buttom of the free end
F(fdof,1) = f; % apply force
alldofs = [1:3*nnd];
freedofs = setdiff(alldofs,fixeddofs);

% elements connected to each node
ncon=zeros(nnd,8);
for i=1:nnd
    row=[];col=[];
    [row,col] = find(con==i);
    d=size(row,1);
    ncon(i,1:d)=row;
end

%volume and stiffness of the elements
for el=1:nel
    elcoord=coord(con(el,:),'');
    [~,Ael(el,1)] =
convhulln([elcoord(1,:);elcoord(2,:);elcoord(3,:);...
elcoord(4,:);elcoord(5,:);elcoord(6,:);elcoord(7,:);elcoord(8,:)])
;
    Kel(:,:,el)=Brick_Stiffness(D,el,coord,con); % element
stiffness at start mesh
end

A=sum(sum(Ael)); % area of the whole structure
Ke=Kel;Ae=Ael; % set the current stifnees and volume as those of
the initial design

for it=1:200

```

```

if it>1; ndcold=ndc; end

[K]=stiffness(nel,nnd,con,Ke); % Global stiffness matrix
U(freedofs,1)=K(freedofs,freedofs)\F(freedofs,1); % FEA for
initial structure

for el = 1:nel
    edof = [3*con(el,1)-2; 3*con(el,1)-1; 3*con(el,1)
            3*con(el,2)-2; 3*con(el,2)-1; 3*con(el,2)
            3*con(el,3)-2; 3*con(el,3)-1; 3*con(el,3)
            3*con(el,4)-2; 3*con(el,4)-1; 3*con(el,4)
            3*con(el,5)-2; 3*con(el,5)-1; 3*con(el,5)
            3*con(el,6)-2; 3*con(el,6)-1; 3*con(el,6)
            3*con(el,7)-2; 3*con(el,7)-1; 3*con(el,7)
            3*con(el,8)-2; 3*con(el,8)-1; 3*con(el,8)];
    Ue=U(edof,1);
    ese(el,1)=0.5*Ue'*Ke(:, :, el)*Ue; % element strain energy
    dc(el,1) = ese(el,1)/Ae1(el,1); % element strain energy
density (structural performance)
end
SE=sum(sum(ese));

for i=1:nnd
    a=find(ncon(i, :)>0);
    b=dc(ncon(i, a), 1);
    ndc(i,1)=sum(b)/max(8, numel(a)); % structural performance
value at nodes
end

if it>1; ndc=(ndc+ndcold)/2; end % stabilization of evolutionary
process

vf(it)=sum(sum(Ae))/A; % volume fraction of current design

%----- evolutionary optimization-----
% finding a minimum level of performance (MLP) which
% results in the target volume of the current iteration
if it == 1; smax=max(max(ndc)); vfi=1; er=0.01; MLP=0.99*smax; end
vfi=max(volfrac, vfi*(1-er)); vfk=0; % target volume fraction of the
current iteration
while abs(vfi-vfk)/vfi>0.001
    rcr=ndc-MLP*ones(size(ndc)); % relative performance
    Ve=solidvolume(Ae1, con, rcr, nel);
    vfk=sum(Ve)/A;
    MLP=MLP*vfk/vfi; % updating MPL
end
%-----

% updating elements' stiffness matrix (KE) using X-FEM
xcoord=coord;xcon=con;
voidel=[];solel=[];bel=[];
for el=1:nel
    [KE, AE, elshape, addcoord]=
brick_prop(con, coord, rcr, el, Ke1, Ae1, D); % element properties
    Ke(:, :, el)=KE; Ae(el,1)=AE;
    if elshape==1
        solel=[solel;el]; %solid elements
    elseif elshape==0
        voidel=[voidel;el]; % void elements
    end
end

```



```

else
    bel=[bel;el]; % boundary elements
    ELshape{el}=elshape;ADDcoord{el}=addcoord;
end
end
% -----

%----- visualization of the topology
for i=1:numel(bel)
    el=bel(i);
    xcon=[xcon;ELshape{el}+size(xcoord,1)];
    xcoord=[xcoord;ADDcoord{el}];
end

xcon([voidel;bel],:)=[];
solsurf=[];
for i=1:numel(solel)

solsurf(end+1:end+6,1:4)=[xcon(i,1:4);xcon(i,5:8);xcon(i,[1 4 8
5])
    xcon(i,[2 3 7 6]);xcon(i,[4 3 7 8]);xcon(i,[1 2 6 5])]];
end

bonsurf=[];
for i=1+numel(solel):size(xcon,1)

bonsurf(end+1:end+6,1:4)=[xcon(i,1:4);xcon(i,5:8);xcon(i,[1 4 8
5])
    xcon(i,[2 3 7 6]);xcon(i,[4 3 7 8]);xcon(i,[1 2 6 5])]];
end

figure (it+1)
xs=[];ys=[];zs=[];
for i=1:size(solsurf,1)
    for j=1:4
        xs(j,i)=xcoord(solsurf(i,j),1);
        ys(j,i)=xcoord(solsurf(i,j),2);
        zs(j,i)=xcoord(solsurf(i,j),3);
    end
end
hsl=patch(xs,ys,zs,'w');
axis([min(coord(:,1)) max(coord(:,1)) min(coord(:,2))
max(coord(:,2)) min(coord(:,3)) max(coord(:,3))]);axis off;
set(hsl,'LineWidth',1.5)
xb=[];yb=[];zb=[];
for i=1:size(bonsurf,1)
    for j=1:4
        xb(j,i)=xcoord(bonsurf(i,j),1);
        yb(j,i)=xcoord(bonsurf(i,j),2);
        zb(j,i)=xcoord(bonsurf(i,j),3);
    end
end
hbl=patch(xb,yb,zb,'w');
axis equal; axis([min(coord(:,1)) max(coord(:,1)) min(coord(:,2))
max(coord(:,2)) min(coord(:,3)) max(coord(:,3))]);axis off;
set(hbl,'EdgeColor','r')
set(hsl,'FaceColor',[.9 .9 .9])
set(hbl,'FaceColor',[.9 .9 .9])
colormap(gray(50))
hgsave(num2str(it))

```

```

close(it)

disp([' It.: ' sprintf('%4i',it) ' Obj.: ' sprintf('%10.4f',SE) '
Vol.: ' sprintf('%6.3f',vf(it) ) ]);

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [K]=stiffness(nel,nnd,con,Ke)
K = sparse(3*nnd, 3*nnd);
for el = 1:nel
    edof = [3*con(el,1)-2; 3*con(el,1)-1; 3*con(el,1)
            3*con(el,2)-2; 3*con(el,2)-1; 3*con(el,2)
            3*con(el,3)-2; 3*con(el,3)-1; 3*con(el,3)
            3*con(el,4)-2; 3*con(el,4)-1; 3*con(el,4)
            3*con(el,5)-2; 3*con(el,5)-1; 3*con(el,5)
            3*con(el,6)-2; 3*con(el,6)-1; 3*con(el,6)
            3*con(el,7)-2; 3*con(el,7)-1; 3*con(el,7)
            3*con(el,8)-2; 3*con(el,8)-1; 3*con(el,8)];
    K(edof,edof) = K(edof,edof) + Ke(:, :, el);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% estimation of the element volumes using nodes' relative
performance numbers
function [Ae]= solidvolume(Ae1,con,rcr,nel)
for en=1:nel
if min(rcr(con(en,:),1)) > 0 % the case that the element is inside
the boudary
    AreaRatio=1;
elseif max(rcr(con(en,:),1)) < 0
    AreaRatio=0;% the case that the element is outside the boudary
else% the case that the element is cut by the boudary
    [ r, s , t ] = meshgrid([-1 : 0.1 : 1],[-1 : 0.1 : 1],[-1 : 0.1 :
1]);
    tmpPhi = (1 - r(:)).*(1 - s(:)).*(1 - t(:))/8 * rcr(con(en,1)) +
(1 + r(:)).*(1 - s(:)).*(1 - t(:))/8 * rcr(con(en,2))...
+ (1 + r(:)).*(1 + s(:)).*(1 - t(:))/8 * rcr(con(en,3)) + (1 -
r(:)).*(1+s(:)).*(1 - t(:))/8 * rcr(con(en,4))...
+ (1 - r(:)).*(1 - s(:)).*(1 + t(:))/8 * rcr(con(en,5)) + (1 +
r(:)).*(1 - s(:)).*(1 + t(:))/8 * rcr(con(en,6))...
+ (1 + r(:)).*(1 + s(:)).*(1 + t(:))/8 * rcr(con(en,7)) + (1 -
r(:)).*(1+s(:)).*(1 + t(:))/8 * rcr(con(en,8));

    AreaRatio = length(find( tmpPhi >= 0 ))/length(s(:));
end;
Ae(en,1)=AreaRatio*Ae1(en,1);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function finds the boundary elements and represent solid
part of
% boundary elements with sub-tetrahedra
% by Meisam Abdi, University of Nottingham
function [KE,AE,elshape,addcoord]=
brick_prop(con,coord,rcr,en,Ke1,Ae1,D)

```

```

KE1=Kel(:, :, en); AE1=Ael(en, 1);
if min(rcr(con(en, :), 1)) >= 0 % the case that the element is
inside the boudary
    KE = KE1; AE=AE1; elshape=1; addcoord=[];
elseif max(rcr(con(en, :), 1)) <= 0 % the case that the element is
outside the boudary
    KE = KE1*0.0001; AE=0; elshape=0; addcoord=[];
else % the case that the element is cut by the boudary

    elrcr=rcr(con(en, :), 1); %relative criteria level of nodes of
the element
    elcoord=coord(con(en, :), :); %coordinates of the nodes of the
brick

    quadsurf=[1 2 3 4; 5 6 7 8; 1 5 8 4; 2 6 7 3; 1 5 6 2; 4 8 7
3]; % define the nodes on each surface of the xehahedra making a
quad
    natcoord=[-1 -1 -1; 1 -1 -1; 1 1 -1; -1 1 -1; -1 -1 1; 1 -1 1;
1 1 1; -1 1 1];
    midquads=[0 0 -1; 0 0 1; -1 0 0; 1 0 0; 0 -1 0; 0 1 0];
%centre of the quads
    midbrick=[0 0 0]; % centre of the brick
    midp=[midquads; midbrick];
    nelcoord=[natcoord; midquads; midbrick];
    subth=[1 2 9 15; 2 3 9 15; 3 4 9 15; 4 1 9 15
           5 6 10 15; 6 7 10 15; 7 8 10 15; 8 5 10 15
           1 5 11 15; 5 8 11 15; 8 4 11 15; 4 1 11 15
           2 6 12 15; 6 7 12 15; 7 3 12 15; 3 2 12 15
           1 5 13 15; 5 6 13 15; 6 2 13 15; 2 1 13 15
           4 8 14 15; 8 7 14 15; 7 3 14 15; 3 4 14 15];
%    subth=[5 1 2 3; 5 1 3 4; 5 2 6 3; 5 3 4 8; 5 3 6 7; 5 3 7 8];
% the brick is divided into 6 subtetrahedra
    edgth=[1 2; 1 3; 1 4; 2 3; 2 4; 3 4]; % edges of a
subtetrahedron

    %finding the value of rcr for the midpoints using the shape
functions
    s=midp(:, 1); t=midp(:, 2); u=midp(:, 3);
    N1 = 1/8*(1-s).* (1-t).* (1-u);    N2 = 1/8*(1+s).* (1-t).* (1-u);
    N3 = 1/8*(1+s).* (1+t).* (1-u);    N4 = 1/8*(1-s).* (1+t).* (1-u);
    N5 = 1/8*(1-s).* (1-t).* (1+u);    N6 = 1/8*(1+s).* (1-t).* (1+u);
    N7 = 1/8*(1+s).* (1+t).* (1+u);    N8 = 1/8*(1-s).* (1+t).* (1+u);
    nr-cr = N1*elrcr(1, 1) + N2*elrcr(2, 1) + N3*elrcr(3, 1) +
N4*elrcr(4, 1) ...
           + N5*elrcr(5, 1) + N6*elrcr(6, 1) + N7*elrcr(7, 1) +
N8*elrcr(8, 1);
    nr-cr=[elrcr; nr-cr];
    elrcr=nr-cr;
    ond=find(elrcr==0); %nodes with 0 rcr
    elrcr(ond, 1)= max(elrcr)*.00001;

    solth=[]; vth=[]; bth=[];
    for i=1:size(subth, 1)
        if min(elrcr(subth(i, :)))>0
            solth=[solth; i]; % solid subtetrahedra
        elseif max(elrcr(subth(i, :)))<0
            vth=[vth; i]; % void subtetrahedra
        else

```

```

        bth=[bth;i]; % boundary subtetrahedra
    end
end

nib=0;sth=subth(solth,:); % nib: global number of intersection
point, sth: subtetrahedra obtained from decomposition of solid
part of tetrahedra
for i=1:size(bth,1)
    ni=0;intcoord=[];intp=[];
    pnd=find(elrcr(subth(bth(i,1),:))>0); %nodes with +rcr
    for j=1:6 % size(edgth,1)
        if
elrcr(subth(bth(i,1),edgth(j,1)))*elrcr(subth(bth(i,1),edgth(j,2))
)<0
            ni=ni+1;nib=nib+1; %ni: no. of int points of the
subtetrahedron, nib: no. of int points of the brick
            if elrcr(subth(bth(i,1),edgth(j,1)))>0
                intp(ni,1:2)=subth(bth(i,1),edgth(j,:)); %
nodes which have a boundary intersection in between
            else %
first column shows the positive nodes, second col, negative nodes
intp(ni,1:2)=[subth(bth(i,1),edgth(j,2)),subth(bth(i,1),edgth(j,1)
)];
            end

x1=nelcoord(intp(ni,1),:);x2=nelcoord(intp(ni,2),:);
            rel=abs(elrcr(intp(ni,1),1)/elrcr(intp(ni,2),1));
            intl=x1+(x2-x1)*rel/(1+rel); % coordinates of
intersection points

            s=intl(1,1);t=intl(1,2);u=intl(1,3);
            N1 = 1/8*(1-s).*(1-t).*(1-u);    N2 =
1/8*(1+s).*(1-t).*(1-u);
            N3 = 1/8*(1+s).*(1+t).*(1-u);    N4 = 1/8*(1-
s).*(1+t).*(1-u);
            N5 = 1/8*(1-s).*(1-t).*(1+u);    N6 =
1/8*(1+s).*(1-t).*(1+u);
            N7 = 1/8*(1+s).*(1+t).*(1+u);    N8 = 1/8*(1-
s).*(1+t).*(1+u);
            orcr = N1*elrcr(1,1) + N2*elrcr(2,1) +
N3*elrcr(3,1) + N4*elrcr(4,1) ...
            + N5*elrcr(5,1) + N6*elrcr(6,1) + N7*elrcr(7,1) +
N8*elrcr(8,1);
            rcrl=elrcr(intp(ni,1),1);rcr2=elrcr(intp(ni,2),1);
            while abs(orcr/mean(abs([rcrl rcr2])))>0.0001
%numerical calculation of the intersection point
                if orcr<0
                    x2=intl;rcr2=orcr;
                else
                    x1=intl;rcr1=orcr;
                end
                rel=abs(rcr1/rcr2);
                intl=x1+(x2-x1)*rel/(1+rel); % coordinates of
intersection points

            s=intl(1,1);t=intl(1,2);u=intl(1,3);
            N1 = 1/8*(1-s).*(1-t).*(1-u);    N2 =
1/8*(1+s).*(1-t).*(1-u);
            N3 = 1/8*(1+s).*(1+t).*(1-u);    N4 = 1/8*(1-
s).*(1+t).*(1-u);

```

```

        N5 = 1/8*(1-s).* (1-t) .* (1+u);    N6 =
1/8*(1+s) .* (1-t) .* (1+u);
        N7 = 1/8*(1+s) .* (1+t) .* (1+u);    N8 = 1/8*(1-
s) .* (1+t) .* (1+u);
        orcr = N1*elrcr(1,1) + N2*elrcr(2,1) +
N3*elrcr(3,1) + N4*elrcr(4,1) ...
        + N5*elrcr(5,1) + N6*elrcr(6,1) +
N7*elrcr(7,1) + N8*elrcr(8,1);
        end
        intcoord(ni,:) = int1;
    end
end
nelcoord = [nelcoord; intcoord]; sc = size(nelcoord, 1);
tri = []; qdl = [];
if ni == 3 && numel(pnd) == 1; % in this case the solid part
of the subtetrahedra is only 1 tetrahedron
    sth = [sth; [sc-2 sc-1 sc subth(bth(i,1), pnd)]];
elseif ni == 3 && numel(pnd) == 3; % in this case, the solid
part of the subtetrahedron has 5 surfaces
    qdl = zeros(3, 4); tri = zeros(5, 3);
    tri(1, :) = subth(bth(i,1), pnd); % which is 2 triangles
and 3 quadrilaterals
    tri(2, :) = [sc-2 sc-1 sc];
    qdl(1, :) = [sc-2 sc-1 intp(ni-1,1) intp(ni-2,1)];
    qdl(2, :) = [sc-1 sc intp(ni,1) intp(ni-1,1)];
    qdl(3, :) = [sc sc-2 intp(ni-2,1) intp(ni,1)];
    qdl = [qdl'; qdl(:,1)']; % we are making triangles from
quadrilaterals
    tri(3:5, :) = qdl(:, 1:3); tri(6:8, :) = qdl(:, 3:5); %
surfaces of the solid part of the subtetrahedron represented by
subtriangles

cntr = mean([nelcoord(subth(bth(i,1), pnd), :); nelcoord(sc-2:sc, :)]);
% centre coordinates of the solid part
nelcoord = [nelcoord; cntr]; % new nelcoord including the
centre

nsth = [tri'; (sc+1)*ones(1, 8)'];
sth = [sth; nsth]; % sub-tetrahedra

elseif ni == 4; % 2 triangles 3 quads
    qdl = zeros(3, 4); tri = zeros(5, 3);
    a1 = find(intp(:, 1) == intp(1, 1)); b1 = setdiff(a1, [1]);
% qdl(1, 2) = sc - ni + b1;
    a2 = find(intp(:, 2) == intp(b1, 2)); b2 = setdiff(a2, [b1]);
% qdl(1, 3) = sc - ni + b2;
    b3 = setdiff([1:4], [1, b1, b2]); % qdl(1, 4) = sc - ni + b3;
    qdl(1, :) = [sc-3, sc-ni+b1, sc-ni+b2, sc-ni+b3]; % first
quad consist of the intersection points: 1(+)b1(-)b2(+)b3(-)1
    qdl(2, :) = [sc-ni+b1, sc-ni+b2, intp(b3, 1), intp(1, 1)];
% second quad
    qdl(3, :) = [sc-ni+b3, sc-3, intp(1, 1), intp(b3, 1)]; % third
quad
    tri(1, :) = [sc-3, sc-ni+b1, intp(1, 1)];
    tri(2, :) = [sc-ni+b2, sc-ni+b3, intp(b2, 1)];
    qdl = [qdl'; qdl(:, 1)']; % we are making triangles from
quadrilaterals
    tri(3:5, :) = qdl(:, 1:3); tri(6:8, :) = qdl(:, 3:5); % quads
decomposed into triangles

cntr = mean([nelcoord(subth(bth(i,1), pnd), :); nelcoord(sc-3:sc, :)]); %
centre coordinates of the solid part

```

```

        nelcoord=[nelcoord;cntr]; %new nelcoord including the
centre
        nth=[tri';(sc+1)*ones(1,8)]';
        sth=[sth;nsth]; %subtetrahedra
    end
end

vst=zeros(size(sth,1),1);
for i=1:size(sth,1)
    vst(i) = 1/6*abs(dot(nelcoord(sth(i,2),:)-
nelcoord(sth(i,1),:),cross(nelcoord(sth(i,3),:)-
nelcoord(sth(i,1),:),nelcoord(sth(i,4),:)-nelcoord(sth(i,1),:))));
end
v=sum(sum(vst));

% finding the physical coordinates of the subtetrahedrals
s=nelcoord(:,1);t=nelcoord(:,2);u=nelcoord(:,3);
N1 = 1/8*(1-s).*(1-t).*(1-u);    N2 = 1/8*(1+s).*(1-t).*(1-u);
N3 = 1/8*(1+s).*(1+t).*(1-u);    N4 = 1/8*(1-s).*(1+t).*(1-u);
N5 = 1/8*(1-s).*(1-t).*(1+u);    N6 = 1/8*(1+s).*(1-t).*(1+u);
N7 = 1/8*(1+s).*(1+t).*(1+u);    N8 = 1/8*(1-s).*(1+t).*(1+u);
physcoord(:,1:3) = N1*elcoord(1,:) + N2*elcoord(2,:) +
N3*elcoord(3,:) + N4*elcoord(4,:) ...
                + N5*elcoord(5,:) + N6*elcoord(6,:) +
N7*elcoord(7,:) + N8*elcoord(8,:);

V=zeros(size(sth,1),1); %volume of the distorted element
for i=1:size(sth,1)
    V(i) = 1/6*abs(dot(physcoord(sth(i,2),:)-
physcoord(sth(i,1),:),cross(physcoord(sth(i,3),:)-
physcoord(sth(i,1),:),physcoord(sth(i,4),:)-
physcoord(sth(i,1),:))));
end
AE=sum(sum(V));

KE =Kxfem3D(elcoord,nelcoord,sth,vst,D); % stiffness matrix
from X-FEM scheme

elshape=zeros(size(sth,1),8);
elshape(:,1:3)=sth(:,1:3);elshape(:,4)=sth(:,3);

elshape(:,5)=sth(:,4);elshape(:,6)=sth(:,4);elshape(:,7)=sth(:,4);
elshape(:,8)=sth(:,4);

addcoord=physcoord;

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% this function calculates the coordinate and connectivity matrix
of 3D
% structures represented with 8-node Hex elements
Function
[connectivity,coordinate]=concoord3D(L,W,h,nelx,nely,nelz)

connectivity=zeros(nelx*nely*nelz,8);coordinate=zeros((nelx+1)*(ne
ly+1)*(nelz+1),3);

```

```

for ii=1:nelx*nely*nelz
    elx=fix((ii-1)/(nely*nelz))+1;
    ely=fix(((ii-(elx-1)*nely*nelz)-1)/nelz)+1;
    elz=mod(ii,nelz);if elz==0;elz=nelz;end;
    connectivity(ii,1)=(nely+1)*(nelz+1)*(elx-1)+(ely-
1)*(nelz+1)+elz;
    connectivity(ii,2)=connectivity(ii,1)+(nely+1)*(nelz+1);
    connectivity(ii,3)=connectivity(ii,2)+(nelz+1);
    connectivity(ii,4)=connectivity(ii,1)+(nelz+1);
    connectivity(ii,5)=connectivity(ii,1)+1;
    connectivity(ii,6)=connectivity(ii,2)+1;
    connectivity(ii,7)=connectivity(ii,3)+1;
    connectivity(ii,8)=connectivity(ii,4)+1;
end

dx=L/nelx;dy=W/nely;dz=h/nelz;
for ii=1:(nelx+1)*(nely+1)*(nelz+1)
    nx=fix((ii-1)/((nely+1)*(nelz+1)))+1;
    ny=fix(((ii-(nx-1)*(nely+1)*(nelz+1))-1)/(nelz+1))+1;
    nz=mod(ii,(nelz+1));if nz==0;nz=nelz+1;end;
    coordinate(ii,:)=[(nx-1)*dx (ny-1)*dy (nz-1)*dz];
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function calculates element stiffness matrix for 8-node Hex
elements.
% by Meisam Abdi, university of Nottinham
function [ke] = Brick_Stiffness(D,en,coord,con)

g=1/sqrt(3);
GP=[-g -g -g;g -g -g;g g -g;-g g -g;-g -g g;g -g g;g g g;-g g g];
% integration points

i=con(en,1);j=con(en,2);k=con(en,3);l=con(en,4);
m=con(en,5);n=con(en,6);o=con(en,7);p=con(en,8);
x1=coord(i,1);y1=coord(i,2);z1=coord(i,3);
x2=coord(j,1);y2=coord(j,2);z2=coord(j,3);
x3=coord(k,1);y3=coord(k,2);z3=coord(k,3);
x4=coord(l,1);y4=coord(l,2);z4=coord(l,3);
x5=coord(m,1);y5=coord(m,2);z5=coord(m,3);
x6=coord(n,1);y6=coord(n,2);z6=coord(n,3);
x7=coord(o,1);y7=coord(o,2);z7=coord(o,3);
x8=coord(p,1);y8=coord(p,2);z8=coord(p,3);

% shape functions
% N1 = 1/8*(1-s)*(1-t)*(1-u);    N2 = 1/8*(1+s)*(1-t)*(1-u);
% N3 = 1/8*(1+s)*(1+t)*(1-u);    N4 = 1/8*(1-s)*(1+t)*(1-u);
% N5 = 1/8*(1-s)*(1-t)*(1+u);    N6 = 1/8*(1+s)*(1-t)*(1+u);
% N7 = 1/8*(1+s)*(1+t)*(1+u);    N8 = 1/8*(1-s)*(1+t)*(1+u);

% interpolation of shape functions
% x = N1*x1 + N2*x2 + N3*x3 + N4*x4 + N5*x5 + N6*x6 + N7*x7 +
N8*x8;
% y = N1*y1 + N2*y2 + N3*y3 + N4*y4 + N5*y5 + N6*y6 + N7*y7 +
N8*y8;
% z = N1*z1 + N2*z2 + N3*z3 + N4*z4 + N5*z5 + N6*z6 + N7*z7 +
N8*z8;

ke=zeros(24,24);
for ii=1:8

```

```

s=GP(ii,1);t=GP(ii,2);u=GP(ii,3);

xs = (x2*(t - 1)*(u - 1))/8 - (x1*(t - 1)*(u - 1))/8 - (x3*(t +
1)*(u - 1))/8 + (x4*(t + 1)*(u - 1))/8 + (x5*(t - 1)*(u + 1))/8 -
(x6*(t - 1)*(u + 1))/8 + (x7*(t + 1)*(u + 1))/8 - (x8*(t + 1)*(u +
1))/8; % diff(x,t);

xt = x2*(s/8 + 1/8)*(u - 1) - x1*(s/8 - 1/8)*(u - 1) - x3*(s/8 +
1/8)*(u - 1) + x4*(s/8 - 1/8)*(u - 1) + x5*(s/8 - 1/8)*(u + 1) -
x6*(s/8 + 1/8)*(u + 1) + x7*(s/8 + 1/8)*(u + 1) - x8*(s/8 -
1/8)*(u + 1); % diff(x,t);

xu = x2*(s/8 + 1/8)*(t - 1) - x1*(s/8 - 1/8)*(t - 1) - x3*(s/8 +
1/8)*(t + 1) + x4*(s/8 - 1/8)*(t + 1) + x5*(s/8 - 1/8)*(t - 1) -
x6*(s/8 + 1/8)*(t - 1) + x7*(s/8 + 1/8)*(t + 1) - x8*(s/8 -
1/8)*(t + 1); % diff(x,u);

ys = (y2*(t - 1)*(u - 1))/8 - (y1*(t - 1)*(u - 1))/8 - (y3*(t +
1)*(u - 1))/8 + (y4*(t + 1)*(u - 1))/8 + (y5*(t - 1)*(u + 1))/8 -
(y6*(t - 1)*(u + 1))/8 + (y7*(t + 1)*(u + 1))/8 - (y8*(t + 1)*(u +
1))/8; % diff(y,s);

yt = y2*(s/8 + 1/8)*(u - 1) - y1*(s/8 - 1/8)*(u - 1) - y3*(s/8 +
1/8)*(u - 1) + y4*(s/8 - 1/8)*(u - 1) + y5*(s/8 - 1/8)*(u + 1) -
y6*(s/8 + 1/8)*(u + 1) + y7*(s/8 + 1/8)*(u + 1) - y8*(s/8 -
1/8)*(u + 1); % diff(y,t);

yu = y2*(s/8 + 1/8)*(t - 1) - y1*(s/8 - 1/8)*(t - 1) - y3*(s/8 +
1/8)*(t + 1) + y4*(s/8 - 1/8)*(t + 1) + y5*(s/8 - 1/8)*(t - 1) -
y6*(s/8 + 1/8)*(t - 1) + y7*(s/8 + 1/8)*(t + 1) - y8*(s/8 -
1/8)*(t + 1); % diff(y,u);

zs = (z2*(t - 1)*(u - 1))/8 - (z1*(t - 1)*(u - 1))/8 - (z3*(t +
1)*(u - 1))/8 + (z4*(t + 1)*(u - 1))/8 + (z5*(t - 1)*(u + 1))/8 -
(z6*(t - 1)*(u + 1))/8 + (z7*(t + 1)*(u + 1))/8 - (z8*(t + 1)*(u +
1))/8; % diff(z,s);

zt = z2*(s/8 + 1/8)*(u - 1) - z1*(s/8 - 1/8)*(u - 1) - z3*(s/8 +
1/8)*(u - 1) + z4*(s/8 - 1/8)*(u - 1) + z5*(s/8 - 1/8)*(u + 1) -
z6*(s/8 + 1/8)*(u + 1) + z7*(s/8 + 1/8)*(u + 1) - z8*(s/8 -
1/8)*(u + 1); % diff(z,t);

zu = z2*(s/8 + 1/8)*(t - 1) - z1*(s/8 - 1/8)*(t - 1) - z3*(s/8 +
1/8)*(t + 1) + z4*(s/8 - 1/8)*(t + 1) + z5*(s/8 - 1/8)*(t - 1) -
z6*(s/8 + 1/8)*(t - 1) + z7*(s/8 + 1/8)*(t + 1) - z8*(s/8 -
1/8)*(t + 1); % diff(z,u);

J = [xs ys zs; xt yt zt; xu yu zu];
detJ = xs*(yt*z u - zt*yu) - ys*(xt*z u - zt*xu) + zs*(xt*yu -
yt*xu);

N1s = -((t - 1)*(u - 1))/8; % diff(N1,s);
N2s = ((t - 1)*(u - 1))/8; % diff(N2,s);
N3s = -((t + 1)*(u - 1))/8; % diff(N3,s);
N4s = ((t + 1)*(u - 1))/8; % diff(N4,s);
N5s = ((t - 1)*(u + 1))/8; % diff(N5,s);
N6s = -((t - 1)*(u + 1))/8; % diff(N6,s);
N7s = ((t + 1)*(u + 1))/8; % diff(N7,s);
N8s = -((t + 1)*(u + 1))/8; % diff(N8,s);

```



```

N1t = -(s/8 - 1/8)*(u - 1); % diff(N1,t);
N2t = (s/8 + 1/8)*(u - 1); % diff(N2,t);
N3t = -(s/8 + 1/8)*(u - 1); % diff(N3,t);
N4t = (s/8 - 1/8)*(u - 1); % diff(N4,t);
N5t = (s/8 - 1/8)*(u + 1); % diff(N5,t);
N6t = -(s/8 + 1/8)*(u + 1); % diff(N6,t);
N7t = (s/8 + 1/8)*(u + 1); % diff(N7,t);
N8t = -(s/8 - 1/8)*(u + 1); % diff(N8,t);
N1u = -(s/8 - 1/8)*(t - 1); % diff(N1,u);
N2u = (s/8 + 1/8)*(t - 1); % diff(N2,u);
N3u = -(s/8 + 1/8)*(t + 1); % diff(N3,u);
N4u = (s/8 - 1/8)*(t + 1); % diff(N4,u);
N5u = (s/8 - 1/8)*(t - 1); % diff(N5,u);
N6u = -(s/8 + 1/8)*(t - 1); %diff(N6,u);
N7u = (s/8 + 1/8)*(t + 1); % diff(N7,u);
N8u = -(s/8 - 1/8)*(t + 1); % diff(N8,u);

Nxyz= J\[N1s N2s N3s N4s N5s N6s N7s N8s;
          N1t N2t N3t N4t N5t N6t N7t N8t;
          N1u N2u N3u N4u N5u N6u N7u N8u];
N1x = Nxyz(1,1); N2x = Nxyz(1,2); N3x = Nxyz(1,3); N4x =
Nxyz(1,4);
N5x = Nxyz(1,5); N6x = Nxyz(1,6); N7x = Nxyz(1,7); N8x =
Nxyz(1,8);
N1y = Nxyz(2,1); N2y = Nxyz(2,2); N3y = Nxyz(2,3); N4y =
Nxyz(2,4);
N5y = Nxyz(2,5); N6y = Nxyz(2,6); N7y = Nxyz(2,7); N8y =
Nxyz(2,8);
N1z = Nxyz(3,1); N2z = Nxyz(3,2); N3z = Nxyz(3,3); N4z =
Nxyz(3,4);
N5z = Nxyz(3,5); N6z = Nxyz(3,6); N7z = Nxyz(3,7); N8z =
Nxyz(3,8);

% Linear strain-displacement transformation matrix
B = [N1x 0 0 N2x 0 0 N3x 0 0 N4x 0 0 N5x 0 0 N6x 0 0 N7x 0 0 N8x 0
0;
     0 N1y 0 0 N2y 0 0 N3y 0 0 N4y 0 0 N5y 0 0 N6y 0 0 N7y 0 0 N8y
0;
     0 0 N1z 0 0 N2z 0 0 N3z 0 0 N4z 0 0 N5z 0 0 N6z 0 0 N7z 0 0
N8z;
     N1y N1x 0 N2y N2x 0 N3y N3x 0 N4y N4x 0 N5y N5x 0 N6y N6x 0
N7y N7x 0 N8y N8x 0;
     0 N1z N1y 0 N2z N2y 0 N3z N3y 0 N4z N4y 0 N5z N5y 0 N6z N6y 0
N7z N7y 0 N8z N8y;
     N1z 0 N1x N2z 0 N2x N3z 0 N3x N4z 0 N4x N5z 0 N5x N6z 0 N6x
N7z 0 N7x N8z 0 N8x];

BD = transpose(B)*D*B*detJ;
ke=ke+BD;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This Functions return the stiffness matrix of a 3D brick element
% using X-FEM approximation

function KE =Kxfem3D(elcoord,nelcoord,sth,vst,D)

x1=elcoord(1,1);y1=elcoord(1,2);z1=elcoord(1,3);
x2=elcoord(2,1);y2=elcoord(2,2);z2=elcoord(2,3);

```

```

x3=elCOORD(3,1);y3=elCOORD(3,2);z3=elCOORD(3,3);
x4=elCOORD(4,1);y4=elCOORD(4,2);z4=elCOORD(4,3);
x5=elCOORD(5,1);y5=elCOORD(5,2);z5=elCOORD(5,3);
x6=elCOORD(6,1);y6=elCOORD(6,2);z6=elCOORD(6,3);
x7=elCOORD(7,1);y7=elCOORD(7,2);z7=elCOORD(7,3);
x8=elCOORD(8,1);y8=elCOORD(8,2);z8=elCOORD(8,3);

KE=zeros(24,24);

[GP1,GP]=tetquad(sth,nelCOORD); %This function performs Gauss
quadrature for tetrahedral. GP1: 1 integration point per
tetrahedron, GP: 4 integration points per tetrahedron

for ii=1:size(sth,1)
k1=zeros(24,24);

for jj=1:4
s=GP(ii,jj,1);t=GP(ii,jj,2);u=GP(ii,jj,3);

xs = (x2*(t - 1)*(u - 1))/8 - (x1*(t - 1)*(u - 1))/8 - (x3*(t +
1)*(u - 1))/8 + (x4*(t + 1)*(u - 1))/8 + (x5*(t - 1)*(u + 1))/8 -
(x6*(t - 1)*(u + 1))/8 + (x7*(t + 1)*(u + 1))/8 - (x8*(t + 1)*(u +
1))/8; % diff(x,t);

xt = x2*(s/8 + 1/8)*(u - 1) - x1*(s/8 - 1/8)*(u - 1) - x3*(s/8 +
1/8)*(u - 1) + x4*(s/8 - 1/8)*(u - 1) + x5*(s/8 - 1/8)*(u + 1) -
x6*(s/8 + 1/8)*(u + 1) + x7*(s/8 + 1/8)*(u + 1) - x8*(s/8 -
1/8)*(u + 1); % diff(x,t);

xu = x2*(s/8 + 1/8)*(t - 1) - x1*(s/8 - 1/8)*(t - 1) - x3*(s/8 +
1/8)*(t + 1) + x4*(s/8 - 1/8)*(t + 1) + x5*(s/8 - 1/8)*(t - 1) -
x6*(s/8 + 1/8)*(t - 1) + x7*(s/8 + 1/8)*(t + 1) - x8*(s/8 -
1/8)*(t + 1); % diff(x,u);

ys = (y2*(t - 1)*(u - 1))/8 - (y1*(t - 1)*(u - 1))/8 - (y3*(t +
1)*(u - 1))/8 + (y4*(t + 1)*(u - 1))/8 + (y5*(t - 1)*(u + 1))/8 -
(y6*(t - 1)*(u + 1))/8 + (y7*(t + 1)*(u + 1))/8 - (y8*(t + 1)*(u +
1))/8; % diff(y,s);

yt = y2*(s/8 + 1/8)*(u - 1) - y1*(s/8 - 1/8)*(u - 1) - y3*(s/8 +
1/8)*(u - 1) + y4*(s/8 - 1/8)*(u - 1) + y5*(s/8 - 1/8)*(u + 1) -
y6*(s/8 + 1/8)*(u + 1) + y7*(s/8 + 1/8)*(u + 1) - y8*(s/8 -
1/8)*(u + 1); % diff(y,t);

yu = y2*(s/8 + 1/8)*(t - 1) - y1*(s/8 - 1/8)*(t - 1) - y3*(s/8 +
1/8)*(t + 1) + y4*(s/8 - 1/8)*(t + 1) + y5*(s/8 - 1/8)*(t - 1) -
y6*(s/8 + 1/8)*(t - 1) + y7*(s/8 + 1/8)*(t + 1) - y8*(s/8 -
1/8)*(t + 1); % diff(y,u);

zs = (z2*(t - 1)*(u - 1))/8 - (z1*(t - 1)*(u - 1))/8 - (z3*(t +
1)*(u - 1))/8 + (z4*(t + 1)*(u - 1))/8 + (z5*(t - 1)*(u + 1))/8 -
(z6*(t - 1)*(u + 1))/8 + (z7*(t + 1)*(u + 1))/8 - (z8*(t + 1)*(u +
1))/8; % diff(z,s);

zt = z2*(s/8 + 1/8)*(u - 1) - z1*(s/8 - 1/8)*(u - 1) - z3*(s/8 +
1/8)*(u - 1) + z4*(s/8 - 1/8)*(u - 1) + z5*(s/8 - 1/8)*(u + 1) -
z6*(s/8 + 1/8)*(u + 1) + z7*(s/8 + 1/8)*(u + 1) - z8*(s/8 -
1/8)*(u + 1); % diff(z,t);

```

```

zu = z2*(s/8 + 1/8)*(t - 1) - z1*(s/8 - 1/8)*(t - 1) - z3*(s/8 +
1/8)*(t + 1) + z4*(s/8 - 1/8)*(t + 1) + z5*(s/8 - 1/8)*(t - 1) -
z6*(s/8 + 1/8)*(t - 1) + z7*(s/8 + 1/8)*(t + 1) - z8*(s/8 -
1/8)*(t + 1); % diff(z,u);

J = [xs ys zs; xt yt zt; xu yu zu]; % Jacobian operator
detJ = xs*(yt*z u - zt*yu) - ys*(xt*z u - zt*xu) + zs*(xt*yu -
yt*xu);

N1s = -((t - 1)*(u - 1))/8; % diff(N1,s);
N2s = ((t - 1)*(u - 1))/8; % diff(N2,s);
N3s = -((t + 1)*(u - 1))/8; % diff(N3,s);
N4s = ((t + 1)*(u - 1))/8; % diff(N4,s);
N5s = ((t - 1)*(u + 1))/8; % diff(N5,s);
N6s = -((t - 1)*(u + 1))/8; % diff(N6,s);
N7s = ((t + 1)*(u + 1))/8; % diff(N7,s);
N8s = -((t + 1)*(u + 1))/8; % diff(N8,s);
N1t = -(s/8 - 1/8)*(u - 1); % diff(N1,t);
N2t = (s/8 + 1/8)*(u - 1); % diff(N2,t);
N3t = -(s/8 + 1/8)*(u - 1); % diff(N3,t);
N4t = (s/8 - 1/8)*(u - 1); % diff(N4,t);
N5t = (s/8 - 1/8)*(u + 1); % diff(N5,t);
N6t = -(s/8 + 1/8)*(u + 1); % diff(N6,t);
N7t = (s/8 + 1/8)*(u + 1); % diff(N7,t);
N8t = -(s/8 - 1/8)*(u + 1); % diff(N8,t);
N1u = -(s/8 - 1/8)*(t - 1); % diff(N1,u);
N2u = (s/8 + 1/8)*(t - 1); % diff(N2,u);
N3u = -(s/8 + 1/8)*(t + 1); % diff(N3,u);
N4u = (s/8 - 1/8)*(t + 1); % diff(N4,u);
N5u = (s/8 - 1/8)*(t - 1); % diff(N5,u);
N6u = -(s/8 + 1/8)*(t - 1); %diff(N6,u);
N7u = (s/8 + 1/8)*(t + 1); % diff(N7,u);
N8u = -(s/8 - 1/8)*(t + 1); % diff(N8,u);

Nxyz= J\[N1s N2s N3s N4s N5s N6s N7s N8s;
        N1t N2t N3t N4t N5t N6t N7t N8t;
        N1u N2u N3u N4u N5u N6u N7u N8u];

N1x = Nxyz(1,1); N2x = Nxyz(1,2); N3x = Nxyz(1,3); N4x =
Nxyz(1,4);
N5x = Nxyz(1,5); N6x = Nxyz(1,6); N7x = Nxyz(1,7); N8x =
Nxyz(1,8);
N1y = Nxyz(2,1); N2y = Nxyz(2,2); N3y = Nxyz(2,3); N4y =
Nxyz(2,4);
N5y = Nxyz(2,5); N6y = Nxyz(2,6); N7y = Nxyz(2,7); N8y =
Nxyz(2,8);
N1z = Nxyz(3,1); N2z = Nxyz(3,2); N3z = Nxyz(3,3); N4z =
Nxyz(3,4);
N5z = Nxyz(3,5); N6z = Nxyz(3,6); N7z = Nxyz(3,7); N8z =
Nxyz(3,8);

% Linear strain-displacement transformation matrix
B = [N1x 0 0 N2x 0 0 N3x 0 0 N4x 0 0 N5x 0 0 N6x 0 0 N7x 0 0 N8x 0
0;
     0 N1y 0 0 N2y 0 0 N3y 0 0 N4y 0 0 N5y 0 0 N6y 0 0 N7y 0 0 N8y
0;
     0 0 N1z 0 0 N2z 0 0 N3z 0 0 N4z 0 0 N5z 0 0 N6z 0 0 N7z 0 0
N8z;
     N1y N1x 0 N2y N2x 0 N3y N3x 0 N4y N4x 0 N5y N5x 0 N6y N6x 0
N7y N7x 0 N8y N8x 0];

```

```
      0 N1z N1y 0 N2z N2y 0 N3z N3y 0 N4z N4y 0 N5z N5y 0 N6z N6y 0
N7z N7y 0 N8z N8y;
      N1z 0 N1x N2z 0 N2x N3z 0 N3x N4z 0 N4x N5z 0 N5x N6z 0 N6x
N7z 0 N7x N8z 0 N8x];
```

```
BD = transpose(B)*D*B*detJ;
k1=k1+l/4*BD;
```

```
end
KE=vst(ii)*k1+KE;
End
```

# Appendix C:

## Matlab Code for 3D version of BESO

The Matlab code presented in this appendix is the 3D version of BESO which is developed based on Huang and Xie's (2010a) study. This code can be used for topology optimization of cuboid design domains. The code accepts the geometric properties of the design domain including length (L), width (W) and height (h), number of grids in x, y and z (nelx, nely & nelz), material properties including Young's modulus (E) and Poisson's ratio (nu), applied force (f), as well as the optimization parameters including volume evolution rate (er) and volume constraint factor (volfrac).

```
%%% 3D Iso-XFEM topology optimization code by Meisam Abdi,  
University of Nottingham %%%  
% This code generates topology optimization solutions for 3D  
structures  
% using Iso-XFEM method.  
  
function BESO3D(L,W,h,nelx,nely,nelz,volfrac,E,NU,er,rmin)  
  
% L: length, W: width, h: height  
% nelx: number of elements in x direction  
% nely: number of elements in y direction  
% nelz: number of elements in z direction  
% er: volume evolution rate  
% volfrac: volume constraint  
% E: Young's modulus, nu: Poisson's ratio  
% f: magnitude of load  
% BESO3D(40,2,20,20,2,10,0.5,1,0.3,0.01,1.5) % example  
  
[con,coord]=concoord3D(L,W,h,nelx,nely,nelz); % Element  
connectivity and Nodes' coordinate matrix  
nel=size(con,1);nnd=size(coord,1); %number of nodes and elements  
  
% Element elasticity matrix  
D = (E/((1+NU)*(1-2*NU)))*[1-NU NU NU 0 0 0 ; NU 1-NU NU 0 0 0 ;  
NU NU ...  
1- NU 0 0 0 ;  
0 0 0 (1-2*NU)/2 0 0 ; 0 0 0 0 (1- 2*NU)/2 0 ; 0 0 0 0 0 (1-  
2*NU)/2];
```

```

% initial design
x=ones(nel,1); % set all design variables as 1 (full solid design
domain)

% Display and save the start mesh (full solid design domain)
surfcon=[];
for i=1:nel
    surfcon(end+1:end+6,1:4)=[con(i,1:4);con(i,5:8);con(i,[1
4 8 5])
    con(i,[2 3 7 6]);con(i,[4 3 7 8]);con(i,[1 2 6 5])];
end
figure (1)
for i=1:size(surfcon,1)
    for j=1:4
        xcoord(j,i)=coord(surfcon(i,j),1);
        ycoord(j,i)=coord(surfcon(i,j),2);
        zcoord(j,i)=coord(surfcon(i,j),3);
    end
end
h=patch(xcoord,ycoord,zcoord,'w');axis equal;
axis([min(coord(:,1)) max(coord(:,1)) min(coord(:,2))
max(coord(:,2)) min(coord(:,3)) max(coord(:,3))]);axis off;
set(h,'LineWidth',1.5)
set(h,'FaceColor',[.85 .85 .85])
hgsave('0')

% Force and Boundary Conditions
U=sparse(3*nnd,1);
F = sparse(3*nnd,1);
fixeddofs=[1:3*(nely+1)*(nelz+1)]; % Fixed DoF
fdof=3*(nelx*(nely+1)*(nelz+1) + (nely/2+1)*(nelz+1)); % DoF of
the bottom of the free end
F(fdof,1) = 1*nely/(nely+1); % apply force
alldofs = [1:3*nnd];
freedofs = setdiff(alldofs,fixeddofs);

% elements connected to each node
ncon=zeros(nnd,8);
for i=1:nnd
    row=[];col=[];
    [row,col] = find(con==i);
    d=size(row,1);
    ncon(i,1:d)=row;
end

%volume and stiffness of the elements
for el=1:nel
    elcoord=coord(con(el,:),'');
    [~,Ael(el,1)] =
convhulln([elcoord(1,:);elcoord(2,:);elcoord(3,:);...
elcoord(4,:);elcoord(5,:);elcoord(6,:);elcoord(7,:);elcoord(8,:)])
;
    Kel(:, :, el)=Brick_Stiffness(D,el,coord,con); % element
stiffness at start mesh
end

A=sum(sum(Ael)); % area of the whole structure

```

```

Ke=Kel;Ae=Ae1; % set the current stiffness and volume as those of
the initial design

if rmin>=0
    [nbnodes,nbdis]=nodedis(coord,rmin);
end

volfracit=1; % defining initial volume fraction of the design

for it=1:200

    if it>1; olddc=dc; end
    volfracit=max(volfracit*(1-er),volfrac); % calculate volume of the
    next design

    [K]=stiffness(nel,nnd,con,Ke,x); % Global stiffness matrix
    U(freedofs,1)=K(freedofs,freedofs)\F(freedofs,1); % FEA for
    initial structure

    for el = 1:nel
        edof = [3*con(el,1)-2; 3*con(el,1)-1; 3*con(el,1)
                3*con(el,2)-2; 3*con(el,2)-1; 3*con(el,2)
                3*con(el,3)-2; 3*con(el,3)-1; 3*con(el,3)
                3*con(el,4)-2; 3*con(el,4)-1; 3*con(el,4)
                3*con(el,5)-2; 3*con(el,5)-1; 3*con(el,5)
                3*con(el,6)-2; 3*con(el,6)-1; 3*con(el,6)
                3*con(el,7)-2; 3*con(el,7)-1; 3*con(el,7)
                3*con(el,8)-2; 3*con(el,8)-1; 3*con(el,8)];
        Ue=U(edof,1);
        ese(el,1)=0.5*Ue'*Ke(:, :, el)*Ue; % element strain energy
        dc(el,1) = ese(el,1)/Ae1(el,1); % element strain energy
        density (elemental sensitivities)
    end
    SE=sum(sum(ese));

    for i=1:nnd
        a=find(ncon(i, :)>0);
        b=dc(ncon(i, a), 1);
        %     ndc(i,1)=mean(b);
        ndc(i,1)=sum(b)/max(8,numel(a)); % nodal sensitivities
    end

    % Filtering The Sensitivites
    if rmin>=0
        [ndc,dc]=filter(nbnodes,nbdis,ndc,dc,rmin,con);
    end

    if it>1; dc=(dc+olddc)/2; end % stabilization of evolutionary
    process

    vf(it)=sum(sum(Ae))/A; % volume fraction of current design

    % BESO Algorithm
    [x]=ADDDEL(volfracit,A,dc,x,Ae);
    sol=x==1;
    vol(it)=sum(Ae1(sol,1));

    %%%%%%%%%%%%%% visualization of the topology

```

```

surfcon=[];xcoord=[];ycoord=[];zcoord=[];
for i=1:nel
    if x(i)==1
        surfcon(end+1:end+6,1:4)=[con(i,1:4);con(i,5:8);con(i,[1
4 8 5])
        con(i,[2 3 7 6]);con(i,[4 3 7 8]);con(i,[1 2 6 5])]];
    end
end

figure (it)
for i=1:size(surfcon,1)
    for j=1:4
        xcoord(j,i)=coord(surfcon(i,j),1);
        ycoord(j,i)=coord(surfcon(i,j),2);
        zcoord(j,i)=coord(surfcon(i,j),3);
    end
end
h2=patch(xcoord,ycoord,zcoord,'w');axis equal;
axis([min(coord(:,1)) max(coord(:,1)) min(coord(:,2))
max(coord(:,2)) min(coord(:,3)) max(coord(:,3))]);axis off;
set(h2,'LineWidth',1.5)
set(h2,'FaceColor',[.85 .85 .85])
camlight
colormap(gray(100))
lighting gouraud
hgsave(num2str(it))
close(it)

disp([' It.: ' sprintf('%4i',it) ' Obj.: ' sprintf('%10.4f',SE) '
Volfrac.: ' sprintf('%6.3f',volfracit )]);

end

%%%%%%%%%%%% BESO add-remove algorithm
%%%%%%%%%%%%
function [x]=ADDDEL(volfrac,domain_vol,dc,x,Ae)
l1 = min(min(min(dc))); l2 = max(max(max(dc)));
while ((l2-l1)/l2 > 1.0e-5)
    th = (l1+l2)/2.0;
    x = max(0.001,sign(dc-th));
    if sum(x.*Ae)-volfrac*(domain_vol) > 0;
        l1 = th;
    else
        l2 = th;
    end
end
end

%%%%%%%%%%%%% filtration
%%%%%%%%%%%%%
function [a,b]=nodedis(coord,rmin)
for i=1:size(coord,1)
    coordi=zeros(size(coord));

coordi(:,1)=coord(i,1);coordi(:,2)=coord(i,2);coordi(:,3)=coord(i,
3);
    ndis=coordi-coord;
    for j=1:size(coord,1)
        r(i,j)=sqrt(sum(ndis(j,1)^2+ndis(j,2)^2+ndis(j,3)^2));
    end
end

```



```

        a{i}=find(r(i,:)<rmin);
        b{i}=r(i,a{i});
end

function [ndc,dc]=filter(a,b,ndc,dc,rmin,con)
oldndc=ndc;
for i=1:size(ndc,1)
    wi=rmin-b{i};
    alphai=oldndc(a{i},1)';
    ndc(i,1)=sum(wi.*alphai)/sum(wi);
end
for i=1:size(dc,1)
    dc(i,1)=mean(ndc(con(i,:),1));
end

%%%%%%%%%% global stiffness matrix
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [K]=stiffness(nel,nnd,con,Ke,x)
K = sparse(3*nnd, 3*nnd);
for el = 1:nel
    edof = [3*con(el,1)-2; 3*con(el,1)-1; 3*con(el,1)
            3*con(el,2)-2; 3*con(el,2)-1; 3*con(el,2)
            3*con(el,3)-2; 3*con(el,3)-1; 3*con(el,3)
            3*con(el,4)-2; 3*con(el,4)-1; 3*con(el,4)
            3*con(el,5)-2; 3*con(el,5)-1; 3*con(el,5)
            3*con(el,6)-2; 3*con(el,6)-1; 3*con(el,6)
            3*con(el,7)-2; 3*con(el,7)-1; 3*con(el,7)
            3*con(el,8)-2; 3*con(el,8)-1; 3*con(el,8)];
    K(edof,edof) = K(edof,edof) + Ke(:, :, el)*x(el);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

# Appendix D:

## Geometrically Non-linear FEA Matlab Code

This Matlab code presents the non-linear FEA method used in chapter 6 for modelling the structures experiencing large deformations. The code operates by dividing the load into a number of increments and implementing Newton-Raphson method to find the equilibrium state in each load increment. In this code, N denotes the number of load increments which can be controlled by the user. In order to regenerate the simulation presented in section 6.4.1, one can use `GNL_FEA(100,2,1,50,2,6000,30e6,0.3,5)`.

```
function GNL_FEA(L,H,t,nelx,nely,f,E,nu,N)

% parameters representing the cantilever beam problem in section
6.4.1:
% GNL_FEA(100,2,1,50,2,6000,30e6,0.3,5)

[con,coord]=con_coor(nelx,nely,L,H); %nodes and elements matrices
coord1=coord; % coordinates in undeformed structure

%----- initialization
D=E/(1-nu^2)*[1,nu,0;nu,1,0;0,0,(1-nu)/2]; % element elasticity
Matrix (plain stress)
% D=E/(1+nu)/(1-2*nu)*[1-nu,nu,0;nu,1-nu,0;0,0,.5-nu]; %element
elasticity Matrix (plain strain)
U=zeros(2*(nelx+1)*(nely+1),1);
F = sparse(2*(nely+1)*(nelx+1),1);
fixeddofs=[1:2*(nely+1)];
alldofs = [1:2*(nely+1)*(nelx+1)];
freedofs = setdiff(alldofs,fixeddofs);
fnode=(nelx+1)*(nely+1)*2; %bottom of the free end
F(fnode,1)=-f;
%-----

for el=1:nelx*nely
    Ke0(:, :,el)=stiffnessmat(D,t,el,coord,con);
end
```

```

[K]=stiffness(nelx,nely,con,Ke0) ;
U1=U;
U1(freedofs,:) = K(freedofs,freedofs) \ F(freedofs,:); %Linear
solution

dF=F/N;Kt=K;se2=0;Wc2=0;
for inc=1:N
    Uel=U;Ut=U;Kold=Kt;
    Uel(freedofs,:) = Kt(freedofs,freedofs) \ dF(freedofs,:);
    %Linear solution to be used as a guess

    [dU,Kt,se]=newton(coord,coord1,con,nelx,nely,freedofs,Uel,Ut,D,t,d
    F*inc); %Newton-Raphson method
    U=U+dU;
    for ih=1:size(U,1)/2
        dis(ih,1)=U(2*ih-1);
        dis(ih,2)=U(2*ih);
    end
    coord=coord1+dis;
    se2=se2+dU'*( (inc-1)*dF+dF/2);%calculating the strain energy
    using the increments and linearity
    Wc2=Wc2+(U-dU/2)'*dF;

    for ih=1:size(U,1)/2
        dis(ih,1)=U1(2*ih-1);
        dis(ih,2)=U1(2*ih); %nodal displacements in x and y
    end
    coordel=coord1+dis/N*inc;

    coordi=coord;
    figure (1)
    for i=1:size(con,1)
        for j=1:4
            xcoord0(j,i)=coord1(con(i,j),1);
            ycoord0(j,i)=coord1(con(i,j),2);
        end
    end
    for i=1:size(con,1)
        for j=1:4
            xcoord1(j,i)=coordel(con(i,j),1);
            ycoord1(j,i)=coordel(con(i,j),2);
        end
    end
    for i=1:size(con,1)
        for j=1:4
            xcoord(j,i)=coordi(con(i,j),1);
            ycoord(j,i)=coordi(con(i,j),2);
        end
    end
    ax=0;bx=1.05*L;ay=-1.5*L;by=H;
    figure (inc)
    h0=patch(xcoord0,ycoord0,'w');axis equal; axis([ax bx ay by]);axis
    off;
    h1=patch(xcoord1,ycoord1,'w');axis equal; axis([ax bx ay by]);axis
    off;
    h=patch(xcoord,ycoord,'w');axis equal; axis([ax bx ay by]);axis
    off;
    set(h,'FaceColor','r')

end

```

```

Linear_deflection=U1(end) % linear tip deflection
nonlinear_deflection=U(fnode) % non-linear tip deflection

%%%%%%%%%% connectivity and coordinate
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [connectivity,coord]=con_coor(nelx,nely,L,h)
connectivity=zeros(nelx*nely,4);
for ii=1:nelx*nely
    rw=mod(ii,nely);
    cl=fix((ii-1)/nely)+1;
    connectivity(ii,1)=cl-1+ii;
    connectivity(ii,4)=connectivity(ii,1)+1;
    connectivity(ii,2)=connectivity(ii,1)+nely+1;
    connectivity(ii,3)=connectivity(ii,2)+1;
end

for ii=1:(nelx+1)*(nely+1)
    coord(ii,1)=fix((ii-1)/(nely+1))*L/nelx;
    coord(ii,2)=mod((ii-1),(nely+1))*h/nely;
end

%%%%%%%%%% Element Stiffness Matrix %%%%%%%%%%%
function [ke]=stiffnessmat(D,t,en,coord,con)

GP=[-1/sqrt(3),-1/sqrt(3);1/sqrt(3),-
1/sqrt(3);1/sqrt(3),1/sqrt(3);-1/sqrt(3),1/sqrt(3)];

i=con(en,1);j=con(en,2);k=con(en,3);l=con(en,4);
x01=coord(i,1);y01=coord(i,2);
x02=coord(j,1);y02=coord(j,2);
x03=coord(k,1);y03=coord(k,2);
x04=coord(l,1);y04=coord(l,2);

ke=zeros(8,8);
for i=1:4

    r=GP(i,1);s=GP(i,2);

    dN1r=-1/4*(1-s);dN1s=-1/4*(1-r);
    dN2r=1/4*(1-s);dN2s=-1/4*(1+r);
    dN3r=1/4*(1+s);dN3s=1/4*(1+r);
    dN4r=-1/4*(1+s);dN4s=1/4*(1-r);

    j11=x01*dN1r+x02*dN2r+x03*dN3r+x04*dN4r;
    j12=y01*dN1r+y02*dN2r+y03*dN3r+y04*dN4r;
    j21=x01*dN1s+x02*dN2s+x03*dN3s+x04*dN4s;
    j22=y01*dN1s+y02*dN2s+y03*dN3s+y04*dN4s;
    J=[j11 j12;j21 j22];detj=det(J);

    dNxy=inv(J)*[dN1r dN2r dN3r dN4r; dN1s dN2s dN3s dN4s];
    dN1x=dNxy(1,1);dN2x=dNxy(1,2);dN3x=dNxy(1,3);dN4x=dNxy(1,4);
    dN1y=dNxy(2,1);dN2y=dNxy(2,2);dN3y=dNxy(2,3);dN4y=dNxy(2,4);

    BL0=[dN1x 0 dN2x 0 dN3x 0 dN4x 0
         0 dN1y 0 dN2y 0 dN3y 0 dN4y

```

```

        dN1y dN1x dN2y dN2x dN3y dN3x dN4y dN4x];

    ke=ke+t*BL0'*D*BL0*detj;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%% global stiffness matrix
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [K]=stiffness(nelx,nely,con,Kel)
K = sparse(2*(nelx+1)*(nely+1), 2*(nelx+1)*(nely+1));
for el = 1:nelx*nely
    edof = [2*con(el,1)-1; 2*con(el,1); 2*con(el,2)-1;
2*con(el,2); 2*con(el,3)-1; 2*con(el,3); 2*con(el,4)-1;
2*con(el,4)];
    K(edof,edof) = K(edof,edof) + Kel(:, :, el);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%% global internal forces
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [Fint]=internalforces(nelx,nely,con,Fe)
Fint=zeros((nelx+1)*(nely+1)*2,1);
for el = 1:nelx*nely
    edof = [2*con(el,1)-1; 2*con(el,1); 2*con(el,2)-1;
2*con(el,2); 2*con(el,3)-1; 2*con(el,3); 2*con(el,4)-1;
2*con(el,4)];
    Fint(edof,1) = Fint(edof,1) + Fe(:,el);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%% element properties in large deformation
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% elements' tangent stiffness matrix, internal force and strain
energy
function [kt,Fel,se]=nonlinearmat(D,t,en,coord,coordt,u,con)

GP=[-1/sqrt(3),-1/sqrt(3);1/sqrt(3),-
1/sqrt(3);1/sqrt(3),1/sqrt(3);-1/sqrt(3),1/sqrt(3)];

i=con(en,1);j=con(en,2);k=con(en,3);l=con(en,4);
x01=coord(i,1);y01=coord(i,2);
x02=coord(j,1);y02=coord(j,2);
x03=coord(k,1);y03=coord(k,2);
x04=coord(l,1);y04=coord(l,2);

ux1=u(2*i-1);uy1=u(2*i);
ux2=u(2*j-1);uy2=u(2*j);
ux3=u(2*k-1);uy3=u(2*k);
ux4=u(2*l-1);uy4=u(2*l);

x1=x01+ux1;x2=x02+ux2;x3=x03+ux3;x4=x04+ux4;
y1=y01+uy1;y2=y02+uy2;y3=y03+uy3;y4=y04+uy4;

k0=zeros(8,8);k1=k0;kn1=k0;Fel=zeros(8,1);se=0;
for i=1:4
    r=GP(i,1);s=GP(i,2);
    N1=1/4*(1-r)*(1-s);
    N2=1/4*(1+r)*(1-s);
    N3=1/4*(1+r)*(1+s);
    N4=1/4*(1-r)*(1+s);

```

```

dN1r=-1/4*(1-s); dN1s=-1/4*(1-r);
dN2r=1/4*(1-s); dN2s=-1/4*(1+r);
dN3r=1/4*(1+s); dN3s=1/4*(1+r);
dN4r=-1/4*(1+s); dN4s=1/4*(1-r);

j11=x01*dN1r+x02*dN2r+x03*dN3r+x04*dN4r;
j12=y01*dN1r+y02*dN2r+y03*dN3r+y04*dN4r;
j21=x01*dN1s+x02*dN2s+x03*dN3s+x04*dN4s;
j22=y01*dN1s+y02*dN2s+y03*dN3s+y04*dN4s;
J=[j11 j12;j21 j22];detj=det(J);

dNxy=inv(J)*[dN1r dN2r dN3r dN4r; dN1s dN2s dN3s dN4s];
dN1x=dNxy(1,1);dN2x=dNxy(1,2);dN3x=dNxy(1,3);dN4x=dNxy(1,4);
dN1y=dNxy(2,1);dN2y=dNxy(2,2);dN3y=dNxy(2,3);dN4y=dNxy(2,4);

dxx11=dN1x*x1+dN2x*x2+dN3x*x3+dN4x*x4;
dxx12=dN1y*x1+dN2y*x2+dN3y*x3+dN4y*x4;
dxx21=dN1x*y1+dN2x*y2+dN3x*y3+dN4x*y4;
dxx22=dN1y*y1+dN2y*y2+dN3y*y3+dN4y*y4;

XX=[dxx11 dxx12; dxx21 dxx22];

BL0=[dN1x 0 dN2x 0 dN3x 0 dN4x 0
      0 dN1y 0 dN2y 0 dN3y 0 dN4y
      dN1y dN1x dN2y dN2x dN3y dN3x dN4y dN4x];

L11=dN1x*ux1+dN2x*ux2+dN3x*ux3+dN4x*ux4;
L12=dN1y*ux1+dN2y*ux2+dN3y*ux3+dN4y*ux4;
L21=dN1x*uy1+dN2x*uy2+dN3x*uy3+dN4x*uy4;
L22=dN1y*uy1+dN2y*uy2+dN3y*uy3+dN4y*uy4;

BL1=[L11*dN1x L21*dN1x L11*dN2x L21*dN2x L11*dN3x L21*dN3x
      L11*dN4x L21*dN4x
      L12*dN1y L22*dN1y L12*dN2y L22*dN2y L12*dN3y L22*dN3y
      L12*dN4y L22*dN4y
      (L11*dN1y+L12*dN1x) (L21*dN1y+L22*dN1x)
      (L11*dN2y+L12*dN2x) (L21*dN2y+L22*dN2x) ...
      (L11*dN3y+L12*dN3x) (L21*dN3y+L22*dN3x)
      (L11*dN4y+L12*dN4x) (L21*dN4y+L22*dN4x)];

BNL=[dN1x 0 dN2x 0 dN3x 0 dN4x 0
      dN1y 0 dN2y 0 dN3y 0 dN4y 0
      0 dN1x 0 dN2x 0 dN3x 0 dN4x
      0 dN1y 0 dN2y 0 dN3y 0 dN4y];

eps=1/2*(XX'*XX-[1 0;0 1]); % green lagrange strain tensor
GLS=[eps(1,1);eps(2,2);eps(1,2)*2];
PK2=D*GLS;
PK2mat=[PK2(1) PK2(3) 0 0;PK2(3) PK2(2) 0 0; 0 0 PK2(1)
        PK2(3); 0 0 PK2(3) PK2(2)];

k0=k0+t*BL0'*D*BL0*detj;
k1=k1+t*detj*(BL0'*D*BL1+BL1'*D*BL0+BL1'*D*BL1);
kn1=kn1+t*BNL'*PK2mat*BNL*detj;
Fel=Fel+(BL0+BL1)'*PK2*t*detj;
se=se+PK2'*GLS*t*detj/2;

```

```

end
kt=k0+k1+kn1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Newton-Raphson method
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function
[dU,Kt,se]=newton(coord,coord1,con,nelx,nely,freedofs,Uel,Ut,D,t,F
)

cond=1;change=1;U=Uel/100;ru=sparse(2*(nely+1)*(nelx+1),1);it=0;
while cond>1e-6
    oldU=U;it=it+1;
    for ih=1:size(U,1)/2
        dis(ih,1)=U(2*ih-1);
        dis(ih,2)=U(2*ih);
    end
    coordi=coord+dis;
    for el=1:nelx*nely

[Ket(:, :, el), Fe(:, el), se(el)]=nonlinearmat(D,t,el,coord1,coordi,Ut
+U,con);
        end
        [Kt]=stiffness(nelx,nely,con,Ket) ;
        Fint=internalforces(nelx,nely,con,Fe);
        ru=Fint-F;
        U(freedofs,:)=U(freedofs,:)-
Kt(freedofs,freedofs)\ru(freedofs,:);

        cond=norm(U-oldU)/norm(U);
    end
dU=U;

```