Determination of Suitable Values for Parameters Governing B-spline Based Evolutionary Structural Optimisation Using the Boundary Element Method

by

Jie Wen

A thesis submitted for the degree of

Master of Science

School of Engineering

The University of Durham

2006

Determination of Suitable Values for Parameters Governing B-spline Based on Evolutionary Structural Optimisation Using the Boundary Element Method

by

Jie Wen

A thesis submitted for the degree of Master of Science

School of Engineering, the University of Durham, 2006

Abstract

The basic evolutionary structural optimisation concept (ESO) has been developed for several years. Recently, the first ESO algorithm based on the boundary element method (BEM) has been presented. In this thesis, this algorithm is used for the 2D shape optimisation. The aim is to develop a greater understanding of the role of certain governing parameters that drive the optimisation using this algorithm, and to make recommendations as to appropriate values of these parameters that give rise to good optimal solutions most efficiently. Two problems, a short cantilever beam and a fillet, are selected as test cases in this work. By using a wide range of numerical tests, the performance of the optimisation has been evaluated using a variety of methods including mean performance analysis and multi-objective optimisation approaches using Pareto curves and weighted sums. Recommendations are made as to appropriate values of these parameters that give rise to good optimal solutions most efficiently. Sensitivity analysis is another important method in engineering design. In this work a new algorithm to undertake a sensitivity analysis has been developed and used in a small number of investigations for boundary element structural optimisation process. ESO is selected when computational efficiency is thought the most important consideration, since it can reach the optimum in fewer iterations and lower run-time compared with sensitivity analysis in structural optimisation.

Acknowledgements

I sincerely appreciate my supervisor, Dr. Jon Trevelyan, for his invaluable guidance and enthusiastic supervision with my research and the thesis.

I would like to thank all the people in the school of Engineering. Thanks in particular to Derek and Gareth, their understanding and patience helped me to adapt the life in the UK.

Thank my parents Baijing Wen and Lihua Zhu, and my uncle Baicheng Wen.

Also thank my grandma deeply.

Table of Contents

Table of Contents 1
List of Figures 4
List of Tables 6
Nomenclature 7
1. Introduction 9
2. Structural Optimisation Review 11
2.1 Optimisation Methods 11
2.1.1 Sizing Optimisation 12
2.1.2 Shape Optimisation 12
2.1.3 Topology Optimisation 13
2.2 Evolutionary Structural Optimisation 14
2.2.1 Additive Evolutionary Structural Optimisation (AESO) 18
2.2.2 Bi-directional Evolutionary Structural Optimisation (BESO) 18
2.3. Numerical Methods in Elastic Stress Analysis 19
2.3.1 Comparison between Available Methods 19
3. Boundary Element Method 23
3.1 Review of Linear Elasticity 24
3.1.1 Stress and the Traction Vector 24
3.1.2 Strain 26
3.1.3 Stress-Strain Relationship 26
3.2 The Boundary Element Method for Stress Analysis 27
3.2.1 The Boundary Integral Equation 27
3.2.2 The BEM as a Matrix Method 31
3.2.3 The Solution 32
3.2.4 Internal Solution 34

3.2.5 Re-analysis 34

4. Nonuniform Rational B-splines 37

4.1 Introduction 37

4.2 B-spline Curves 39

4.2.1 Nonrational B-spline Curves 39

4.2.2 Rational B-spline Curves 42

5. Algorithm for ESO with BEM representation of B-splines 46

5.1 Geometry Definition 47

5.2 Boundary Element Analysis 48

5.3 Removal and Addition of Material 48

5.3.1 Area to Move 48

5.3.2 Distance to Move 49

5.3.3 Direction of Movement 51

5.4 Stopping Criterion 51

5.5 Geometry Control 52

5.5.1 Insertion of Control Points 52

5.5.2 Deletion of Control Points 54

5.5.3 Subdivision of Distorted Elements 55

5.6 Topology Optimisation 55

5.7 Concluding Remarks 57

6. Sensitivity Based Optimisation 58

6.1 SA in Structural Optimisation 60

7. Parameter Selection for Shape Optimisation 63

7.1 Introduction 63

7.2 Short Cantilever Beam 64

7.3 Fillet 69

7.4 Conclusion 78

8. Parameter Selection for Sensitivity Analysis 79

8.1 Fillet Problem 79

8.2 Parameter Selection 83

8.3 Comparison with ESO 85

8.4 Conclusion 87

9. Conclusions and Recommendation for Future Research 88

9.1 Conclusions 88

9.2 Recommendation for Future Research 89

Bibliography 91

List of Figures

Figure 2.1: Design domain for the two-bar frame structure 16
Figure 2.2(a-j): Evolutionary of two-bar frame structure 17
Figure 2.3: Comparison of FEM and BEM in the same problem 20
Figure 3.1: Components of the stress tensor 24
Figure 3.2: Quadratic boundary element 30
Figure 3.3: Modified portion of the matrix in a typical BEM reanalysis 35
Figure 4.1: Nonuniform B-spline basis functions for n=5, p=2 41
Figure 4.2: Initial B-spline curve 41
Figure 4.3: B-spline after moving P_2 to P_2' 42
Figure 4.4: NURBS basis functions for n=5, p=2, w_i=0.5 44
Figure 4.5: NURBS Curve 44
Figure 4.6: Rational cubic B-spline curves with w_3 varying 45
Figure 5.1: Outline of Optimisation Algorithm 47
Figure 5.2: Movement when removing material 50
Figure 5.3: Insertion of Control Point 53
Figure 5.4: Deletion of Control Point 54
Figure 5.5: Movement of the Control Point in the Internal Holes 56
Figure 7.1: Short cantilever problem 64
Figure 7.2: Diagrammatic representation of the parametric space used 66
Figure 7.3: Evolution of the short cantilever beam example 68
Figure 7.4: Evolution of objective function 68
Figure 7.5: Fillet example 70
Figure 7.6: Evolution of the process for fillet optimisation (30 iterations) 70
Figure 7.7: Evolution of the objective function 71
Figure 7.8: Evolution of the process for fillet optimisation (86 iterations) 73
Figure 7.9: Evolution of the objective function 74
Figure 7.10: Scatter plot of results for fillet problem, showing Pareto curve 75
Figure 7.11: Evolution of the fillet example 77
Figure 7.12: Evolution of objective function 77
Figure 8.1: Fillet problem 80
Figure 8.2: Scatter plot of results for fillet problem using sensitivity analysis 83

Figure 8.3: Optimum for the h = 800mm model.....84
Figure 8.4: Local and global optimisation.....85
Figure 8.5: Comparison between sensitivity analysis and ESO.....86

List of Tables

Table 7.1: Mean performances over range of RF values **66**
Table 7.2: Mean performances over range of ER_R and RR values **67**
Table 7.3: Results for the fillet problem **73**
Table 7.4: Best parameters for fillet problem (Pareto analysis) **75**
Table 7.5: Best parameters for fillet problem (weighted sum method) **76**
Table 8.1: Results for fillet problem by sensitivity analysis **82**

Nomenclature

a	distance
a_1, a_2	weighting factors
AF	addition factor
AR	addition ratio
b	distance
	body forces
c	distance
d	distance
	stepsize
E	Young's modulus
ER	evolution rate
ER_A	evolutionary rate for addition
ER_R	evolutionary rate for removal
F	representational force
f	objective function
IR	inclusion ratio
J	Jacobian
k	constant factor
L	reference length
L_e	least/most stressed element
M	minimum value of objective function
n	normal vector
	total number of degrees of freedom
	iteration number at which objective function is minimised
p	degree of the polynomial
	boundary element node
	iteration number
PI	performance indicator
Q_i	the i^{th} design variable
RF	removal factor
RR	rejection ratio

s_i	design sensitivity
s_{max}	maximum value of sensitivity
SS	steady state
t	traction
t	thickness
u	displacement vector
U	strain energy
V	volume of material in the structure
V_e	element volume
w	width of a narrow strip of material
w_i	weights
β_e	selected criterion
β_{max}	maximum value for the selected criterion
$\beta_x, \beta_y, \beta_z$	body forces
Γ	boundary
δ_{ij}	Kronecker delta
ε	strain
θ	internal angle
μ	shear modulus
ν	Poisson's ratio
ξ	local coordinate
σ	stress
$\sigma_1, \sigma_2, \sigma_3$	principal stresses
σ_e	the largest von Mises stress in the element
σ_{IP}	von Mises stress at an internal point
σ_{max}	maximum permissible value of von Mises stress
σ_{VM}	Von Mises stress
σ_{VMe}	the element von Mises stress
σ_y	yield stress or any other maximum stress criterion
Ω	volume

1. Introduction

Analysis plays an important role in engineering design. The primary purpose of analysis is to assist engineers in making key decisions in the structural design process. In the past, although some early work indicates engineers were already thinking about structural optimisation one hundred years ago (Michell (1904)), the designer's skills and experience were the most important prerequisite for a successful engineering design. More recently, the advances in computational performance have led to the development of structural optimisation algorithms and their increased usage in industrial settings. In its most rigorous mathematical meaning, the term optimisation implies the search for the design than is in some sense the best. The particular sense might, for example, be the design that uses the minimum amount of material within given stress constraints. In the engineering world, the term can equally mean the improving of a design, especially when we recognise that a mathematical optimisation may well yield a design that is impractical as an engineering solution. Recent work is incorporating such engineering concepts as design robustness into an optimisation algorithm in order to bring the engineering and mathematical definitions of 'optimisation' closer together. With the development of scientific methods, optimisation has been widely used in engineering design. It can be applied to solve problems requiring a high level of performance. Improvements in our ability to perform these complex calculations have been made possible by advances in analysis methods, such as design sensitivity analysis, and by the increasing speed and memory capacity of digital computers.

The finite element method (FEM) and the boundary element method (BEM), as computational methods have become essential tools in structural analysis. Since the FEM was developed in the late 1950s, it has been used in solving stress analysis problems for various types in almost all branches of engineering, such as aeronautical, civil, mechanical, and nuclear. It is an approximate technique in which the object is represented by discrete regions, or 'elements'. The entire domain is described in meshes in the FEM model. The displacement field is normally described in a piecewise polynomial fashion. A set of simultaneous equations is constructed by

applying the equations of equilibrium to each element, and the entire set of equations assembled for all elements can be solved for unknown displacement values using linear or nonlinear algebra as appropriate. Stresses and strains are then found using differentiation and by application of Hooke's Law. Refining the mesh, i.e. using more elements, will generally improve the accuracy of the FEM at the cost of the increased demand on computational resources.

The boundary element method (BEM) is used in this work as the numerical stress analysis tool. In last few decades, the BEM has become an effective alternative which improves modelling time and accuracy over the finite element method (FEM). Especially, the BEM is highly suitable for shape optimisation because of its particular advantages such as building the model in a reduced dimension and rapid re-analysis capability. In addition use of the BEM can avoid some of the drawbacks found in the use of finite elements as the stress analysis engine for shape optimisation.

In this thesis, reviews are firstly given of optimisation methods in general and also specifically of Evolutionary Structural Optimisation (ESO). Secondly, a brief comparison of the boundary element method with the finite element method will be made, before proceeding with a detailed overview of the boundary element method. Then the ESO algorithm used in this project is described. Results from some numerical tests are presented and recommendations made for the most appropriate and general values to be used for the parameters that guide the ESO scheme. Finally the results are discussed and conclusions drawn.

The first ESO algorithm using a boundary element discretisation on a B-spline boundary geometric representation was presented by Cervera (2003) and Cervera & Trevelyan (2005a, 2005b). The aim of the current work is to develop a greater understanding of the role of certain governing parameters that drive the optimisation using this algorithm, and to make recommendations as to appropriate values of these parameters that give rise to good optimal solutions most efficiently.

2. Structural Optimisation Review

Structural optimisation is an important field of study due to its contribution to cost, material and time savings in engineering design. Many basic techniques have been developed for achieving the optimum, such as *hill climbing* and *linear programming*. *Hill climbing* is a heuristic method. One characteristic of a so-called heuristic algorithm is that carries on solving the problem and ignores whether the solution is correct. It starts by comparing the value of a certain function at an arbitrary point with those of many points around it, and then carries out evolution of the design by the tendency of moving towards the points whose function values are lower than the appointed one. The problem with this method is that it may search in a local, but not in the global, design space. For modelling in real applications *linear programming* (Cormen et.al (2001)) has been developed and used in a wide variety of fields. It is expressed in matrix form as

$$\begin{aligned} &\text{maximize} && \mathbf{cx} \\ &\text{subject to} && \mathbf{Ax} \leq \mathbf{b} \\ & && \mathbf{x} \geq 0 \end{aligned} \tag{2.1}$$

where $\mathbf{x}$ is the vector of variables, $\mathbf{cx}$ is called the objective function and the $\mathbf{Ax} \leq \mathbf{b}$ and $\mathbf{x} \geq 0$ represents the constraints. The objective function and constraints are both linear. Commonly, the simplex algorithm first introduced by Dantzig (1951) is used for solving the linear programs. However, many optimisation problems are nonlinear. So nonlinear optimisation is widely used in variable fields, such as engineering design, economics, geography etc.

2.1 Optimisation Methods

There are many different methods available to solve structural optimisation problems: they may be broadly divided into the three categories of sizing, shape and topology optimisation.

2.1.1 Sizing Optimisation

In the sizing optimisation approach, the optimisation is achieved by varying a series of sizes and dimensions. There are two main application areas in sizing optimisation: discrete structures and continuum structures. In the late 1960s, optimality criteria methods which solved problems in discrete structures were developed by Venkayya, Khot, Berke (1973) and others. For the problem of continuum structures, the geometry must be converted into a finite element model with adaptable loads and boundary conditions. Fleury (1979) applied the dual theory of convex programming to a separable approximation of the design problem. A discrete-continuum optimization criteria (DCOC) (Zhou and Rozvany (1996)) iterative algorithm using the reciprocal linear approximation of the displacement constraints was presented which improved the iterative performance.

2.1.2 Shape Optimisation

In structural optimisation problems, shape optimisation always deserves attention first because it is simpler than topology optimisation yet still allows the solution of many interesting and industrially relevant problems. Numerical shape optimisation schemes must be based around some method of obtaining stress and displacement solutions for the object under analysis. The finite element method (FEM) appears most commonly in the literature. Schnack and Spörl (1986) used a mechanical dynamic programming algorithm for structure optimization based on FEM. Mattheck and Burkhardt (1990) introduced a Computer-Aided Optimization (CAO) method using a commercial finite-element code, by reducing localized notch stresses in 2D and 3D elastic structures based on a biological growth analogy. Another method called SKO (soft kill option), which found an optimum structural topology by simulating adaptive bone mineralization, was applied by Baumgartner et al (1992). This method combined with CAO to carry out the optimisation by varying the Young's modulus according to a calculated stress distribution. In spite of the success of the FEM in general structural and mechanical analysis, the boundary element

method (BEM) (Trevelyan (1994)) has more recently emerged as an alternative that has a number of desirable features for shape optimisation. Parvizian and Fenner (1997) have compared mathematical programming with normal movement techniques in the optimization of 2D boundary element models. Their numerical results indicate that the normal movement approaches are easier to use in design. Meric (1995) applied sensitivity analysis and material derivative concepts in the shape optimisation of 2D heat conduction problems models by BEM. In the papers of Yamazaki et al (1993 and 1994) the direct differentiation method of the discrete boundary integral equation is applied to determine optimum shapes (in both 2D and 3D models) of minimum weight subjected to stress constraints. Finally, gradientless methods such as the Response Surface Method and Genetic Algorithms have recently emerged as attractive options for optimisation of non-linear systems, or systems having multiple local optima. A Genetic Algorithm (GA) (Goldberg (1989)) is a search method based on natural selection and genetic processes like reproduction and mutation are used in the shape optimisation. Cervera and Trevelyan (2005(a)(b)) have implemented evolutionary structural optimisation ideas in a boundary element context with some success.

2.1.3 Topology Optimisation

Topology optimisation was pioneered by Michell (1904), who studied statically determinate trusses for a number of loading and support conditions. It is a more complex task than shape optimisation. Since the advent of fast numerical stress analysis methods this work has once again become a subject of some attention. For instance, a homogenisation method was advanced by Bendsøe and Kikuchi (1988). This method is a material distribution method using an artificial composite material with microscopic voids. Eschenauer et al (1994) described a bubble method which is based on a solution concept comprising an iterative positioning of new bubbles followed by a hierarchically secondary shape optimisation of the new bubbles together with all variable boundaries. A level set method operates by building a level set model embedded in a scalar function of a higher dimension (Wang et al (2003)). Such a level set model has flexibility in complex topological changes, and also

concisely describes the boundary shape of the structure. Another method for topology optimisation has been developed by Wang and Tai (2005). They further used GA to solve the topology optimisation problems by a bit-array representation method.

2.2 Evolutionary Structural Optimisation

The Evolutionary Structural Optimisation (ESO) method was first proposed by Xie and Steven (1993) and has been applied in many types of structural problems. It is a heuristic engineering design algorithm. ESO has been capable of solving a variety of problems of size, shape and topology optimisation in a wide variety of engineering problem types, including stress/displacement analysis, heat transfer, and fluid flow. The current work focuses on the stress/displacement problem. A concept of fully stressed design is inherently satisfied with this algorithm. It is considered to be fully stressed in the sense that all material in a structure is subject to its allowable stress under at least one of the load cases. The material can be removed if it is not used efficiently, i.e. has only small stress. The ESO in this context is based on the idea of removing inefficient material from an initially oversized domain. To remove the part of material that is not needed, a so-called *rejection criterion (RC)* is used. For example, many ESO researchers describe algorithms that use the *Von Mises stress* as a rejection criterion to determine when removal should take place and how much material to remove. Finite element analysis results are typically presented as *Von Mises stress*. In three dimensions *Von Mises stress* is

$$\sigma_{VM} = \sqrt{\frac{1}{2}\left[(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2\right]}$$

where σ_1, σ_2 and σ_3 are the principal stresses. In a plane stress case, $\sigma_3 = 0$. When σ_{VM} exceeds the material's yield stress, failure will occur. The process is carried out by removing material from low stress areas and possibly also adding material in regions of high stress by moving elements.

It has to be mentioned here that the essential operation in an ESO scheme is the removal of material that is not being used efficiently. For simple stress-based schemes this is usually performed by removing parts of the object under analysis that are subjected to low values of stress. It is common to use von Mises stress in the criterion, since this is a widely used failure criterion that, importantly, considers both tensile and compressive stresses using positive numbers. In this regard, this description of stress makes no distinction between compressive and tensile stresses. But other stress components, strain energy, or design sensitivities could also be used as appropriate to govern optimisation for different materials (e.g. concrete, in which tensile strength might be neglected) or to consider different objective functions.

In an FEM based scheme (Xie and Steven (1993)), elements are eliminated according to the criterion:

$$\sigma_e \leq RR\sigma_{\max} \tag{2.1}$$

where σ_e is the largest von Mises stress in the element, σ_{max} is some predefined maximum permissible value of von Mises stress (often related to the yield stress of the material) and *RR* is the *rejection ratio*. In BEM based schemes, the criterion described by equation 2.1 may be applied to regions of the boundary, and material removed by redefining the geometric description of the object's boundary.

Such a cycle is repeated using the same *RR* until a steady state is reached. Then an *evolution rate*, *ER*, is added to the *RR*.

$$RR_{i+1} = RR_i + ER \tag{2.2}$$

in which the subscript refers to the iteration number. The iterations take place again until a new steady state is attained.

Such an evolutionary process continues until a desired optimum is reached. Ideally the final structure becomes a fully stressed design where the material at each point of the structure is stressed to its full strength. Querin et al. (1998) introduced a *performance indicator* (*PI*) that measures how well the overall structure is performing against an idealised situation. The *PI* is defined as:

$$PI = \frac{\sum_{element} \sigma_{VMe} V_e}{FL} \tag{2.3}$$

where σ_{VMe} is the element von Mises stress, V_e is the element volume, F is a representational force and L is a reference length.

A large number of numerical examples demonstrate the ESO method is very efficient. A two-bar frame (Xie and Steven (1993)) as an example is followed. A design domain of the size 10 × 20 (m) as shown in figure 2.1 is discretized into 25 × 60 bilinear quadrilateral plane stress elements. Young's modulus $E = 100$GPa and Poisson's ratio $\nu = 0.3$ are assumed. The initial rejection ratio $RR_0 = 0.01$ and the evolution rate $ER = 0.01$. Figures 2.2 (a)-(j) show the evolutionary process of this model from a rectangular plate into the final tress type structure.

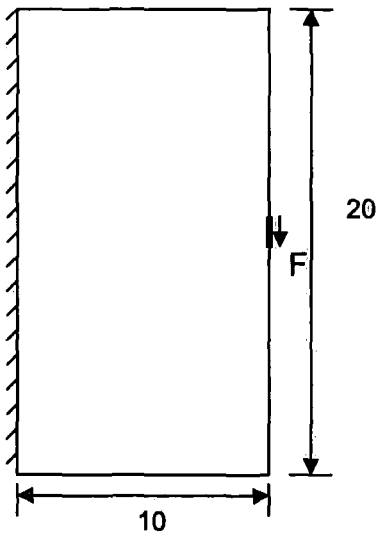


Figure 2.1: Design domain for the two-bar frame structure

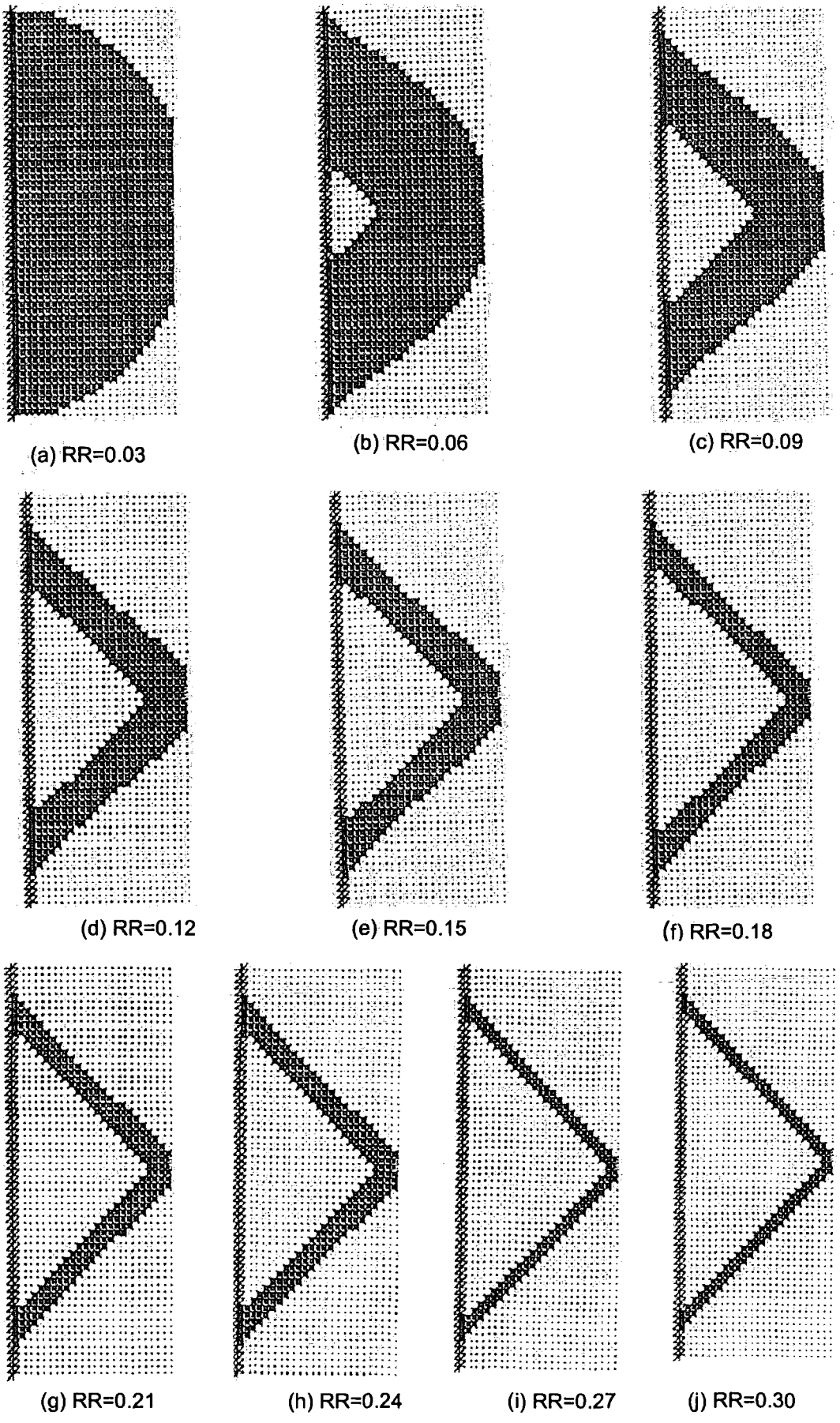


Figure 2.2(a-j): Evolutionary of two-bar frame structure (Xie & Steven (1993))

2.2.1 Additive Evolutionary Structural Optimisation (AESO)

AESO (Querin et al. (2000)) is a similar method to the basic ESO procedure described above. The AESO only adds the material where is most needed, but not remove any from the inefficient areas. This evolutionary process is driving by the following criterion:

$$\beta_e \geq IR\beta_{\max} \tag{2.4}$$

where β_e is the selected criterion, β_{max} is the maximum value for the selected criterion and IR is a parameter called the *inclusion ratio*. This parameter may be determined by

$$IR = a_0 - a_1 SS - a_2 SS^2 - a_3 SS^3, \ldots. \tag{2.5}$$

where SS is *steady state* number, $0 \leq IR \leq 1$, a_0=1, and $a_1 \ldots. a_n$ are determined from numerical experiments with AESO.

When a steady state is achieved, elements are not added any more. At this stage, SS is increased by 1 and the IR is re-calculated. Repeat such a process until the maximum criterion has been reduced or the *performance indicator* (PI) in equation (2.3) has been minimised.

2.2.2 Bi-directional Evolutionary Structural Optimisation (BESO)

In the BESO method (Querin et al. (1998)), elements of the structure can not only be added, but also removed. The addition and removal processes are the AESO and ESO methods. For example, when the optimisation criterion is the von Mises stress, elements can be added or removed if they satisfy the equations:

$$\sigma_e \leq RR\sigma_{\max} \tag{2.6}$$

$$\sigma_e \geq IR\sigma_{\max} \tag{2.7}$$

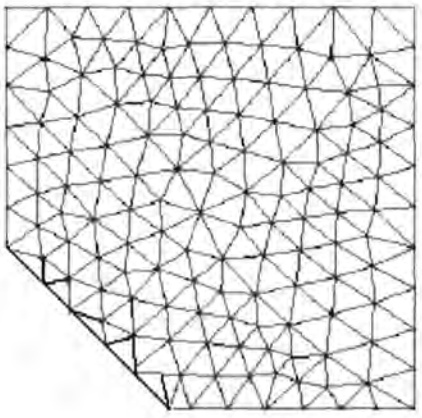
where σ_e is the element von Mises stress, σ_{max} is the maximum value of the von Mises stress, *RR* is the *rejection ratio,* and *IR* is the *inclusion ratio.*

When a steady state is achieved, elements are not added any more. Such a process is repeated until the criterion reaches the limit value or the *performance indicator* (*PI*) (equation 2.3) has been minimised, similar to AESO and ESO.

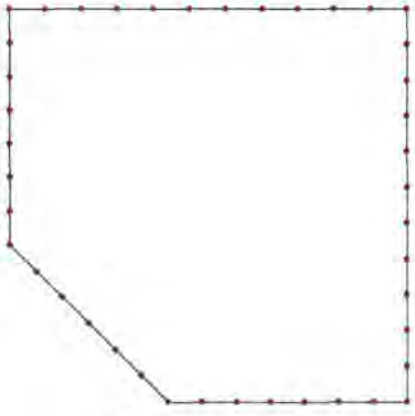
2.3. Numerical Methods in Elastic Stress Analysis

2.3.1 Comparison between Available Methods

The vast majority of numerical stress analysis computations performed today are done using the finite element method (FEM). This is an approximate technique based on a discretisation of the domain, and may be applied to a wide variety of problem types, including linear and non-linear stress analysis, heat transfer, vibrations, acoustics, electromagnetics, and many others. In the case of elasticity, which is the subject of the current work, the displacement field is assumed to be described in a piecewise polynomial fashion. Confining this description to two-dimensional problems for simplicity, the finite elements will be (generally) triangular and quadrilateral in shape, and fill the complete extent of the material under analysis. Engineers familiar with the technique will know also that it is not sufficient only to fill the domain with elements, but in order to achieve accurate stress solutions it must be ensured that the elements are sufficiently small, and sufficiently undistorted, so that each element is capable of capturing the stress field within its area.



(a) Mesh by FEM



(b) Mesh by BEM

Figure 2.3: Comparison of FEM and BEM in the same problem

However, this method has some shortcoming as checkerboarding (Rozvany (2004)) and non-smooth boundaries (Reynolds (1999)). As an analysis tool in the typical structure optimisation (Xie and Steven(1993)), the drawbacks to use of the FEM are presented (Wen and Trevelyan (2005)):

- Typically shape optimisation is a boundary related process so the vast majority of the finite elements are not producing information that directly guides the process.

- Careful checking needs to be made to identify and remove 'islands', i.e. elements or groups of elements that become unconnected to the model and cause singularities in the solution.

- The method tends to give rise to a mesh character known as 'checkerboarding', in which the elements and voids fill regions in the same pattern as the black and white squares on a chess board. This results in single node contact between diagonally adjacent elements that is undesirable.

- In the later stages of the optimisation, the geometry has converged to a truss-like state, and care needs to be taken to ensure this is not represented in the finite element model by a line of elements joined diagonally at their corners.

- Problems converging to surfaces that are not horizontal or vertical (i.e. almost all problems!) will be represented in the finite element model by non-smooth, 'staircase' type meshes on the boundary. These can give rise to artificial stress concentrations that impede the optimisation process.

The boundary element method is similar to the finite element method. In brief, the two methods are both based on the use of matrix algebra to solve large systems of simultaneous equations. They both use the concept of node points to define the displacement on each element and shape functions to describe the variation of this displacement over the element. The difference between them is that only the surface area of the object should be defined in BEM, but the entire volume must be defined in FEM. Compared with FEM, BEM is highly suitable for shape optimisation. A model can be built easily and quickly, and importantly the elements are confined to the boundary of the object which is where the majority of concerns lie in an optimisation scheme. The accuracy of computation is high because the boundary integral equation can be used directly to determine the derivatives of the objective and constraint functions. In the optimisation process re-generation of BEM models, which is to accommodate a change in the design geometry, is both straightforward and fast (Tafreshi and Fenner (1991)). (Cervera and Trevelyan (2002)) also use the BEM as the analytical tool because it does not require remeshing of the domain, which can reduce the computational effort per iteration and eliminates perturbations due to changes in the mesh. Further advantages are available in optimisation through the use of re-analysis. Geometric perturbations of the boundary from one iteration to

the next often involve only a small part of the model changing. It has been shown by Trevelyan and Wang (2001) that much of the computation in the re-analysis can be saved by re-using matrix terms from the previous iteration, and by using an iterative solver for rapid re-solution of the updated equation set. More detail on this aspect of the BEM is provided in section 3.2.5.

3. Boundary Element Method

More than 100 years ago, the fundamentals of the boundary element method have been investigated in depth by Abel and Helmholtz. Fredholm (1903) was the first to solve the elasticity problems by discretization procedures. But his idea was held back by the lack of calculating speed before the appearance of the fast computers. In the 1950s Mikhlin (1957) and Muskhelisvili (1953) improved application of this type of technique in engineering.

In the early sixties, the development of the high-speed computer made the BEM become more applicable. Jaswon (1963) and Symm (1963) solved the Fredholm equations using a collocation procedure for simple 2D potential flow problems. At the same time, Hess and Smith (1967) worked on solving potential flow problems for general geometries. Their works were extended to elasticity problems also in the 1960's, in which the works of Rizzo (1967), Cruse and Rizzo (1968) and Cruse (1968) who were the first to describe the direct boundary integral equation method which is still popular in engineering software. Lachat (1976) introduced to BEM the concept of higher-order elements using quadratic shape functions. The first text book describing the newly named boundary element method by Brebbia (1978) was published in 1978. Since then BEM has continued to develop. Brebbia (1989), Beer (1992), Becker (1992), Kane (1994) and Trevelyan (1994) have presented other text books in the field.

There are two formulations in the BEM, the direct formulation and the indirect formulation. In the former, the physical variables, e.g. displacements and tractions in elasticity, are unknown. In the latter, the unknown physical variables are obtained by imaginary densities. The current work considers only the direct formulation.

3.1 Review of Linear Elasticity

The concept of linear elasticity as the basis of the boundary element method has to be presented firstly in order to introduce the BEM.

3.1.1 Stress and the Traction Vector

The stress on any particular face is defined as the resultant force divided by the area. Stress can have units of Newtons per square metre (N/m^2) or Pascal (Pa) where 1 Pa = 1 N/m^2. Three normal stresses (σ_{xx}, σ_{yy}, σ_{zz}) are parallel to the coordinate axes, as illustrated in figure 3.1.

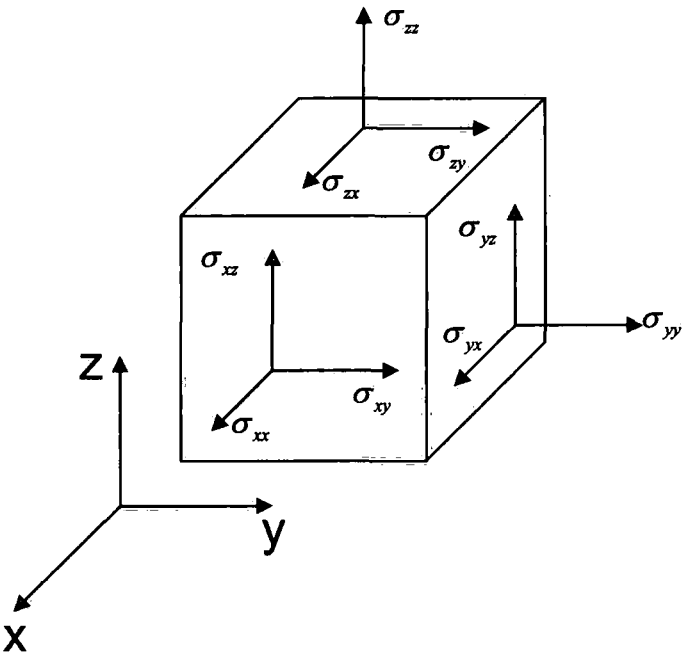


Figure 3.1: Components of the stress tensor

The stress tensor is symmetric in the equilibrium condition as shown

$$\sigma_{ij} = \sigma_{ji} \qquad (i, j = 1, 2 \text{ or } 3) \qquad (3.1)$$

The components of the stress at the various points in a body must satisfy the equilibrium equations.

$$\begin{aligned}
\frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{yx}}{\partial y} + \frac{\partial \sigma_{zx}}{\partial z} + \beta_x &= 0 \\
\frac{\partial \sigma_{xy}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} + \frac{\partial \sigma_{zy}}{\partial z} + \beta_y &= 0 \\
\frac{\partial \sigma_{xz}}{\partial x} + \frac{\partial \sigma_{yz}}{\partial y} + \frac{\partial \sigma_{zz}}{\partial z} + \beta_z &= 0
\end{aligned} \tag{3.2}$$

where β_x, β_y and β_z are body forces.

Another way to define stresses is by the *traction vector* which is derived from the stress components in the coordinate directions, and is denoted t_x, t_y and t_z. A set of equations can be written

$$\begin{aligned}
t_x &= \sigma_{xx} n_x + \sigma_{yx} n_y + \sigma_{zx} n_z \\
t_y &= \sigma_{xy} n_x + \sigma_{yy} n_y + \sigma_{zy} n_z \\
t_z &= \sigma_{xz} n_x + \bar{\sigma}_{yz} n_y + \sigma_{zz} n_z
\end{aligned} \tag{3.3}$$

where n_x, n_y and n_z are the components in the (x, y, z) directions of the outward pointing normal to the surface.

For two dimensional plane stress problems $\sigma_{zz} = \sigma_{xz} = \sigma_{yz} = 0$. A similar set of equations can be written:

$$\begin{aligned}
\frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{yx}}{\partial y} + \beta_x &= 0 \\
\frac{\partial \sigma_{xy}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} + \beta_y &= 0
\end{aligned} \tag{3.4}$$

$$\begin{aligned}
t_x &= \sigma_{xx} n_x + \sigma_{yx} n_y \\
t_y &= \sigma_{xy} n_x + \sigma_{yy} n_y
\end{aligned} \tag{3.5}$$

3.1.2 Strain

A solid body deforms when it is subjected to stress. The deformation of a body is often measured and characterised in terms of *strain*, which is denoted ε, and is formed from the derivatives of the displacement vector (u_x, u_y, u_z).

$$\begin{aligned}
&\varepsilon_{xx} = \frac{\partial u_x}{\partial x}, \quad \varepsilon_{yy} = \frac{\partial u_y}{\partial y}, \quad \varepsilon_{zz} = \frac{\partial u_z}{\partial z} \\
&\varepsilon_{xy} = \left(\frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x}\right), \quad \varepsilon_{xz} = \left(\frac{\partial u_x}{\partial z} + \frac{\partial u_z}{\partial x}\right), \quad \varepsilon_{yz} = \left(\frac{\partial u_y}{\partial z} + \frac{\partial u_z}{\partial y}\right) \\
&\varepsilon_{yx} = \varepsilon_{xy}, \quad \varepsilon_{zx} = \varepsilon_{xz}, \quad \varepsilon_{zy} = \varepsilon_{yz}
\end{aligned} \tag{3.6}$$

For two dimensional plane strain problems, the strain components $\varepsilon_{zz} = \varepsilon_{xz} = \varepsilon_{yz} = 0$. The non-zero strains are

$$\varepsilon_{xx} = \frac{\partial u_x}{\partial x}, \quad \varepsilon_{yy} = \frac{\partial u_y}{\partial y}, \quad \varepsilon_{xy} = \varepsilon_{yx} = \left(\frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x}\right) \tag{3.7}$$

$$\frac{\partial^2 \varepsilon_{yy}}{\partial x^2} + \frac{\partial^2 \varepsilon_{xx}}{\partial y^2} - \frac{2\partial^2 \varepsilon_{xy}}{\partial x \partial y} = 0 \tag{3.8}$$

Equation 3.8, which may be verified from equation 3.6, is known as the *compatibility condition*.

3.1.3 Stress-Strain Relationship

For an isotropic elastic state, the stress-strain relationship can be simply derived from the definitions of Young's modulus and Poisson's ratio using the principle of superposition. These relationships can be written

$$\begin{aligned}
\varepsilon_{xx} &= \frac{1}{E}\left[\sigma_{xx} - \nu(\sigma_{yy} + \sigma_{zz})\right] \\
\varepsilon_{yy} &= \frac{1}{E}\left[\sigma_{yy} - \nu(\sigma_{zz} + \sigma_{xx})\right] \\
\varepsilon_{zz} &= \frac{1}{E}\left[\sigma_{zz} - \nu(\sigma_{xx} + \sigma_{yy})\right] \\
\varepsilon_{xy} &= \frac{\sigma_{xy}}{\mu}, \varepsilon_{yz} = \frac{\sigma_{yz}}{\mu}, \varepsilon_{zx} = \frac{\sigma_{zx}}{\mu}
\end{aligned} \tag{3.9}$$

where E is the Young's modulus, ν is the Poisson's ratio and μ is the shear modulus. In an isotropic material, these material constants satisfy the following relationship:

$$\mu = \frac{E}{2(1+\nu)} \tag{3.10}$$

3.2 The Boundary Element Method for Stress Analysis

3.2.1 The Boundary Integral Equation

The reciprocal theorem is well known in mechanics. There are different ways of expressing the reciprocal theorem; one of them is given here as the beginning of the derivation of the boundary element method for stress analysis.

Consider an object subjected to two load cases; we will call them A and B. Load case A contains some forces and displacement constraints. Load case B consists of a different set of forces and displacement constraints. The reciprocal theorem states that the work done by the forces from load case A on the displacements from load case B is equal to the work done by the forces from load case B on the displacements from load case A. They can be written

$$\text{Forces}_A \text{ x } \text{Displacements}_B = \text{Forces}_B \text{ x } \text{Displacements}_A \tag{3.11}$$

Being a little more scientific about the statement, in load case A the forces are made up of boundary tractions t and body forces b, and the displacements u. For load case B a similar notation is used but we write t^*, b^* and u^*. Since tractions are applied only at the boundary of the object, work is done only over the boundary, Γ. Body forces (e.g. gravitational or thermal loads) act throughout the volume, Ω. So we can sum, or integrate, the work done by writing the reciprocal theorem as

$$\int_\Gamma t^* u d\Gamma + \int_\Omega b^* u d\Omega = \int_\Gamma u^* t d\Gamma + \int_\Omega u^* b d\Omega \tag{3.12}$$

Let load case B take the form of a concentrated point force at some position *'p'* in the volume. Kelvin's *fundamental solutions*, which are sometimes termed the "free-space Green's function", are u^* and t^*. The fundamental solutions for 2D linear elastic stress analysis (Trevelyan (1994)) are

$$u^*_{ij} = \frac{1}{8\pi\mu(1-\nu)}\left[(3-4\nu)\ln\frac{1}{r}\delta_{ij} + \frac{\partial r}{\partial x_i}\frac{\partial r}{\partial x_j}\right] \tag{3.13}$$

$$t^*_{ij} = \frac{-1}{4\pi(1-\nu)r}\frac{\partial r}{\partial n}\left[(1-2\nu)\delta_{ij} + 2\frac{\partial r}{\partial x_i}\frac{\partial r}{\partial x_j}\right] + \frac{1-2\nu}{4\pi(1-\nu)r}\left[\frac{\partial r}{\partial x_j}n_i - \frac{\partial r}{\partial x_i}n_j\right] \tag{3.14}$$

where i and j represent direction indices, u^*_{ij} is the displacement in direction x_i at some location due to a concentrated point force in the direction x_j, μ and ν are the shear modulus and the Poisson's ratio, r is the distance from the point *'p'* to the point at which the displacement is required, δ_{ij} is the Kronecker delta, which takes the value zero if $i \neq j$, and 1 if $i = j$.

By the particular choice of the load case B the volume integral on the left hand side of equation 3.12 reduces to

$$\int_\Omega b^* \cdot u \, d\Omega = u(p) \tag{3.15}$$

If we assume the body forces in the real load case, b, to be zero, we can reduce equation 3.12 to a much simpler form

$$u(p) + \int_\Gamma t^* \cdot u \, d\Gamma = \int_\Gamma u^* \cdot t \, d\Gamma \tag{3.16}$$

The volume integrals have been removed, and the only term that remains in this expression that relates to the inside of the material is the first one '$u(p)$', which is the displacement at the point 'p' inside the object.

Moving the point 'p' to the boundary, we have the final form of the equation which is called the Boundary Integral Equation.

$$c(p)u(p) + \int_{\Gamma} t^* u d\Gamma = \int_{\Gamma} u^* t d\Gamma \tag{3.17}$$

A multiplier *c(p)* is introduced as a result of treating a singular boundary integral as the limiting value of a non-singular integral, taken in the sense of Cauchy principal value. Often *c(p)* takes the value of 0.5 because '*p*' is on a smooth boundary. To be more general, *c(p)* takes the value of $\theta / 2\pi$, where θ is the internal angle subtended at point '*p*'.

In practice, it is very difficult, for all but the simplest cases, to solve the boundary integral equation analytically. The integration has to be done numerically by an approximate method. Engineers are usually familiar with classical numerical integration methods such as the Trapezoidal Rule and Simpson's Rule (Abramowitz and Stegun (1972)) for numerical integration. It is most common for both FEM and BEM implementations to use Gauss-Legendre quadrature (Davis & Rabinowitz (1984)). Like the Trapezoidal and Simpson's rules this involves the evaluation of the integrand at a number of points (or abscissae) and the weighted sum of such evaluations. In order to achieve accuracy of this numerical integration we perform it over a large number of subdivisions of Γ. These subdivisions are the boundary elements. This subdivision into small elements is also required to provide for interpolation of displacements and tractions between nodal values. So it is seen that the discretisation fulfils two purposes, one for ease and accuracy of numerical integration and one for interpolation. A subdivided form of the Boundary Integral Equation can be written in which the integrals are expressed as the sum of the integrals over all the elements

$$c(p)u(p) + \sum_{elements} \int_{\Gamma} t^* u d\Gamma_e = \sum_{elements} \int_{\Gamma} u^* t d\Gamma_e \tag{3.18}$$

Boundary elements have nodes which are often placed at the end and at the mid-points of the elements like finite elements. These can define the geometry of the element and the displacement, traction and stress variation over the element. A

quadratic line element for 2D analysis is illustrated in figure 3.2, in which the local coordinate ξ is defined.

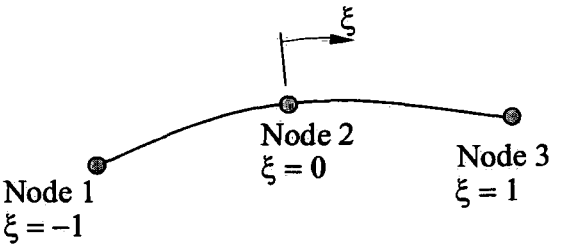


Figure 3.2: Quadratic boundary element

Interpolation is performed using shape functions which are identical in concept to their use in finite element methods. For the quadratic boundary element shown, the shape functions N_i in the local coordinate ξ are given by

$$\begin{aligned} N_1 &= \frac{1}{2}\xi(\xi-1) \\ N_2 &= (1-\xi)(1+\xi) \\ N_3 &= \frac{1}{2}\xi(\xi+1) \end{aligned} \tag{3.19}$$

So the displacement, u, at any point on the element will be found using the interpolation

$$u(\xi) = \sum_{i=1}^{3} N_i u_i \tag{3.20}$$

Expressing equation 3.19 as a vector multiplication,

$$u(\xi) = \mathbf{N}^T\mathbf{u} \tag{3.21}$$

where $\mathbf{N}^T$ is the transpose of a vector containing the shape functions and $\mathbf{u}$ is a vector containing the values of the displacement at the three nodes.

The traction t, at any point is given by

$$t(\xi) = N_1 t_1 + N_2 t_2 + N_3 t_3 \tag{3.22}$$

$$t(\xi) = \mathbf{N}^T \mathbf{t} \tag{3.23}$$

3.2.2 The BEM as a Matrix Method

The next important step is replacing the displacement u and traction t in equation 3.17 by the interpolated forms. We can not perform the integration directly since the terms u and t, both remain unknown. By replacing with the interpolated form equation 3.17 becomes

$$c(p)u(p) + \sum_{elements} \int_{-1}^{1} t^* \mathbf{N}^T \mathbf{u} J d\xi = \sum_{elements} \int_{-1}^{1} u^* \mathbf{N}^T \mathbf{t} J d\xi \tag{3.24}$$

where J is called the Jacobian (Gradshteyn and Ryzhik (2000)) which is used to transform the variables of integration or differentiation from one set of variables to another. Recognise that the vectors **u** and **t** may be treated as constants and removed from the integral.

$$c(p)u(p) + \sum_{elements} \int_{-1}^{1} t^* \mathbf{N}^T J d\xi \mathbf{u} = \sum_{elements} \int_{-1}^{1} u^* \mathbf{N}^T J d\xi \mathbf{t} \tag{3.25}$$

For every element on the boundary, the value of the integral can be calculated since every term in the integral is known. An expression is given as follows for p at node 1

$$c(1)u_1 + \mathbf{h}_{11}u_1 + \mathbf{h}_{12}u_2 + \mathbf{h}_{13}u_3 + \ldots + \mathbf{h}_{1n}u_n = \mathbf{g}_{11}t_1 + \mathbf{g}_{12}t_2 + \mathbf{g}_{13}t_3 + \ldots + \mathbf{g}_{1n}t_n \tag{3.26}$$

where the **h** terms arise from the integrals of the t^* terms and the **g** terms arise from the u^* integrals. This expression is somewhat simplified for clarity. Here it is needed

to notice the fundamental solutions are singular at the point '*p*'. In reality, in a 2D analysis we would also include similar terms involving displacement and traction in the y-direction as well as the x-direction. The term c(1) and all **h** and **g** terms are known. Obviously we have an equation including many unknowns, the displacement and traction at each node. However, a similar expression can be found by placing the force in the y-direction at node 1. Place the point '*p*' at every node in turn, repeat the integration step in both directions at each node and a set of equations will be developed. These can be written in matrix form.

$$\mathbf{Hu} = \mathbf{Gt} \tag{3.27}$$

where **H** is a matrix of all the **h** coefficients, **G** is a matrix of all the **g** coefficients, and the vectors **u** and **t** contain the displacements and tractions in each direction at the nodes on the boundary.

3.2.3 The Solution

Equation 3.27 is a statement of a set of n simultaneous equations with $2n$ unknowns, where n is the total number of degrees of freedom. So it can not be solved without reducing the number of unknowns. This is most simply achieved by applying the boundary conditions in such a way that at each node and in each direction we prescribe either the displacement or the traction. In practice this does not present a difficulty since a free surface has zero traction.

The first step in the solution is to swap the columns of the matrices to bring all the terms that remain unknown to the left hand side and take all the terms as boundary conditions to the right hand side as follows:

$$\mathbf{Ax}=\mathbf{By} \tag{3.28}$$

where **A** and **B** are matrices containing columns of **H** and **G**, **x** is a vector of mixed displacements and tractions that remain unknown, and **y** is a vector of known displacement and tractions as boundary conditions. The matrix **B** and the vector **y** are known, so we can multiply out the matrix vector product to leave

$$\mathbf{Ax=b} \tag{3.29}$$

where **b** is the vector result of multiplying out the right hand side of equation 3.28.

Equation 3.29 can be solved using a variety of techniques. Popular solvers are *direct solvers* like Gauss Elimination, and *iterative solvers* like GMRES (Saad (1986)) (Generalised Minimum Residual Method).

- **Gauss Elimination.** A combination of row operations reduces the system matrix to upper triangular form and then the solution is obtained through back-substitution. This method is guaranteed to arrive at a solution for a non-singular system.

- **GMRES.** This is a conjugate gradient type solver that is applicable to the non-symmetric systems that characterise boundary element equations. The advantage of this method is speed, especially for larger systems of equations, though occasionally the solver may fail to converge satisfactorily. Convergence properties can be greatly improved through suitable preconditioning.

The solution of equation 3.29 provides us with a full description of the displacements and tractions on the boundary, which allow ready calculation of stress components and principal stresses, etc.

3.2.4 Internal Solution

We can find the stresses and displacements in the similar process when '*p*' is at the inside the material. In this case the internal angle $\theta = 2\pi$ at the collocation point '*p*', so $c(p) = 1$. The equation 3.25 can be expressed

$$u(p) + \sum_{elements} \int_{-1}^{1} t^* \mathbf{N}^T J d\xi \mathbf{u} = \sum_{elements} \int_{-1}^{1} u^* \mathbf{N}^T J d\xi \mathbf{t} \tag{3.30}$$

Here **u** and **t** are both known. The only unknown is $u(p)$, the displacement at the internal point, which can be found by evaluating the integrals. In the same way, the stresses at the internal point can be found by the derivative of the equation 3.30.

3.2.5 Re-analysis

By the term re-analysis we mean the solution of a system of equations that is similar to a set that has already been solved, and for which a solution is available. This is applicable to both finite element and boundary element systems. It is clear that this type of situation is a common one in any iterative shape optimisation scheme in which the geometry in successive iterations is formed by considering a small perturbation from the previous geometry.

The early work in reanalysis techniques has been reviewed by Arora (1976), and later by Abu Kassim and Topping (1987). In the early years of finite element analysis the motivation for re-analysis was simply that the run-time could be extremely long for analysis jobs that would today be performed in seconds. Therefore, any schemes that could provide savings would be helpful. More recent work is derived from more advanced motivations, usually the acceleration of a shape optimisation scheme or the acceleration of computations to allow contours to be updated dynamically as a model is changed, introducing a greater degree of interactivity than is possible using a full analysis for each design change. Mackie (1998) developed an object-oriented approach with FEM reanalysis. However,

although the FEM reanalysis can reduce run time by inheriting the advantage of the previous runs for the next run, the subsequent remeshing is generally more cumbersome than the straightforward BEM reanalysis. However, Mackie has presented a functioning approach based on substructuring of FE models and making use of multithreading of operations.

In the BEM, the use of re-analysis is more advanced and appears more commonly in the literature. This is because a geometric change is much more simply and robustly accommodated in the mesh, and the effects of the geometric change do not propagate as much through the model as they do in the FEM. The typical pattern of rows and columns that require updating in the governing matrix in a re-analysis is shown in figure 3.3.

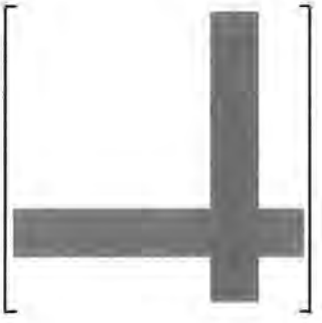


Figure 3.3: Modified portion of the matrix in a typical BEM reanalysis

Note that this governing matrix, such as **A** in equation 3.28, is a dense and unsymmetric matrix. The updated **A** is largely similar to that in the previous analysis.

BEM re-analysis was first applied by Kane *et al.* (1990), whose scheme allows large geometric perturbation but is not guaranteed to converge. Leu (1999) presented a scheme based on a reduction method, in which the new solution is expressed as a linear combination of orthogonal basis vectors produced through a Gram-Schmidt procedure. This was applied to simple shape optimisation problems.

Both schemes of Kane and Leu suffer from the fact that they do not allow well for multiple perturbations of different parts of the model. The re-analysis makes use of an inversion or decomposition of the original system matrix. This means that the efficiency of the algorithms degrades after a number of different perturbations have been performed. Trevelyan and Wang (2001) presented a simple BEM reanalysis scheme which did not suffer from the drawback and was also guaranteed to converge given sufficient iterations. This approach is based on the GMRES iterative solver, in which the previous matrix is rewritten and a full matrix solution is performed in each analysis. The approach was further accelerated by Trevelyan, Scales, Morris and Bird (2005), who introduced an approximate complete LU preconditioner that greatly reduced the iterations to convergence.

4. Nonuniform Rational B-splines

Nonuniform rational B-splines (NURBS) have become a common standard for the representation of geometric information processed by computers. NURBS have excellent mathematical and algorithmic properties, which have contributed to their enormous popularity. The B-spline curves and surface have the similar pattern and properties. This chapter only introduces the B-spline curves.

4.1 Introduction

Pierre Bézier developed a method for shape description using the Bernstein basis or polynomial approximation function. A parametric *n*th-degree Bézier curve is defined by

$$C(u) = \sum_{i=0}^{n} B_{i,n}(u) P_i \qquad 0 \le u \le 1 \tag{4.1}$$

where the geometric coefficients P_i are called the *control points*, which form the *control polygon*, and the basis or blending functions $B_{i,n}$ are the *Bernstein polynomials* given by

$$B_{i,n}(u) = \frac{n!}{i!(n-i)!} u^i (1-u)^{n-i} \tag{4.2}$$

The use of such basis functions gives the Bézier curves, also applicable to Bézier surfaces, some important properties (Rogers (2001))

- The basis functions are real.
- The degree of the polynomial defining the curve segment is one less than the number of control points.
- The curve generally follows the shape of the control polygon.

- The first and last points on the curve are coincident with the first and last points of the control polygon.
- The tangent vectors at the ends of the curve have the same direction as the first and last polygon spans, respectively.
- The curve is contained within the convex hull of the control polygon, i.e., within the largest convex polygon defined by the control polygon vertices.
- The curve exhibits the *variation-diminishing property*. This means that the curve does not oscillate about any straight line more often than the control polygon, or, in other words, no straight line has more intersections with the curve than with the control polygon.
- The curve is invariant under an affine transformation.

However, curves consisting of just one polynomial or rational segment can be inadequate. Their shortcomings are

- **The limitation of flexibility** The number of specified polygon vertices fixes the order of the resulting polynomial that defines the curves.

- **The limitation of control** Specified polygon controls the shape of curves. However, the control gets inefficient when the specified polygon has more vertices that leads to a higher order of curves.

- **The limitation of changing** Any point is a result of blending the values of all control vertices, so a change in one vertex affects the entire curve. This eliminates the ability to produce a local change.

For all these reasons, an alternative solution can be to use curves which are *piecewise polynomial*, or *piecewise rational* such as B-splines.

4.2 B-spline Curves

B-splines are a generalisation of Bézier curves. Both curves are controlled by a set of points P_i (*control points*) lying on a polygon (*control polygon*). In general they do not necessarily interpolate their endpoints. However, the *nonuniform* B-spline basis functions allow this, passing through the first and last points. A B-spline curve differs from a Bézier curve in that it usually consists of more than one curve segment. Each segment is defined and influenced by only a few control points, which are the coefficients of the B-Spline basis function polynomials. The degree of the curve is independent of the total number of control points. These characteristics allow local changes in shape; i.e. changes do not propagate beyond one or only a few local segments.

4.2.1 Nonrational B-spline Curves

The most general *nonrational* B-spline curves are those defined by *nonrational* basis functions. That is, the basis function defining one segment may differ from those defining another. This allows us to interpolate one or more of the control points, depending on the modelling situation. The *nonrational* p^{th}*-degree B-spline curve* is given by (Piegl and Tiller (1997))

$$C(u) = \sum_{i=0}^{n} N_{i,p}(u) P_i \qquad a \le u \le b \qquad (4.3)$$

where the P_i are the $(n+1)$ control points and the piecewise polynomials $N_{i,p}(u)$ are the p^{th}*-degree* basis functions defined recursively as

$$N_{i,0}(u) = \begin{cases} 1 & \text{if } u_i \le u \le u_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \qquad (4.4)$$

The u_i are *knot* values which form the *knot vector* $U = \{u_0, u_1, \ldots, u_m\}$. They relate the parametric variable u to the control points P_i. The parameters determining the number of control points, $n+1$, knots, $m+1$, and the degree of the polynomial, p, are related by

$$n+p+1=m \tag{4.5}$$

For *nonuniform* and *nonperiodic* B-spline curves, the knot vector is characterised by

$$U = \left\{ \underbrace{a,\ldots,a}_{p+1}, u_{p+1}, \ldots, u_{m-p-1}, \underbrace{b,\ldots,b}_{p+1} \right\} \tag{4.6}$$

where end knots a and b are repeated with multiplicity p to interpolate the initial and final control points. If the entire curve is parameterised over the unit interval, then for most practical situations, $a=0$ and $b=1$. Spacing the knots at equal intervals of the parameter describes a *uniform* nonrational B-spline curve; otherwise it is *nonuniform.*

For example, the following B-spline curve presented has a degree $p=2$, *i.e.* it is a quadratic curve, and six control points. The basic functions $N_{i,2}(u)$ can be obtained by equation (4.4), $0 \leq u \leq 1$ and $U= \{0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1\}$. We in turn compute $N_{i,0}(u)$, $N_{i,1}(u)$ and $N_{i,2}(u)$. The basis functions $N_{i,2}(u)$ are plotted in figure 4.1.

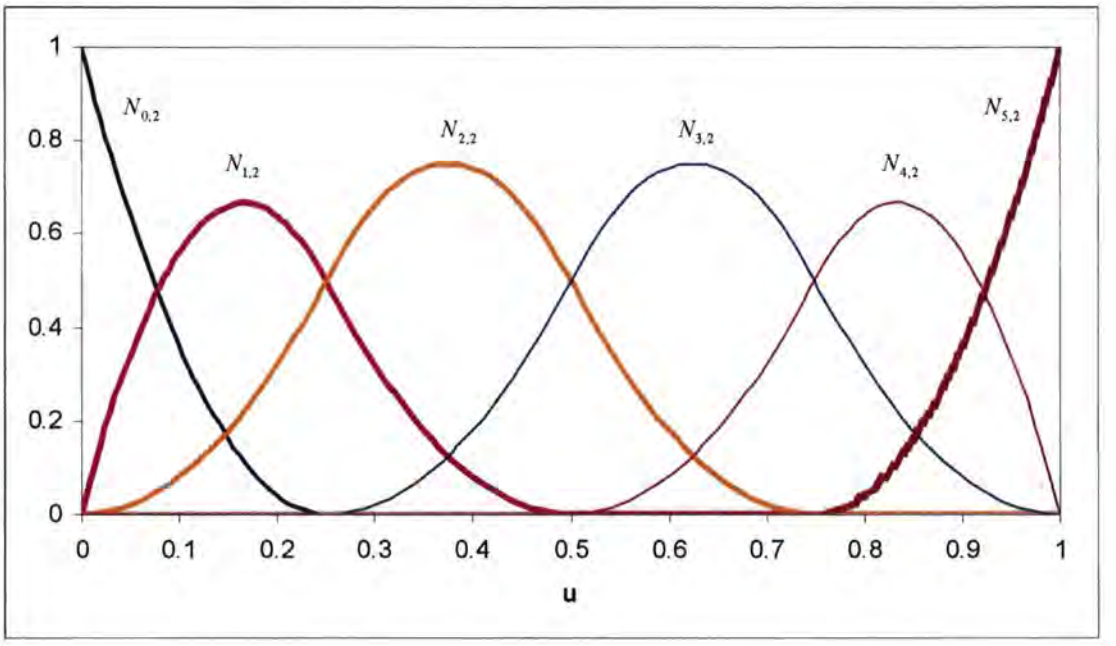


Figure 4.1: Nonuniform B-spline basis functions for n=5, p=2

The B-spline curve is obtained as shown in Figure 4.2. We can see a set of six control points $\{P_0(0,0), P_1(2,4), P_2(4,5), P_3(6,0), P_4(8,1), P_5(10,5)\}$, the resulting curve $C(u)$ including four segments $\{C_1(u), C_2(u), C_3(u), C_4(u)\}$.

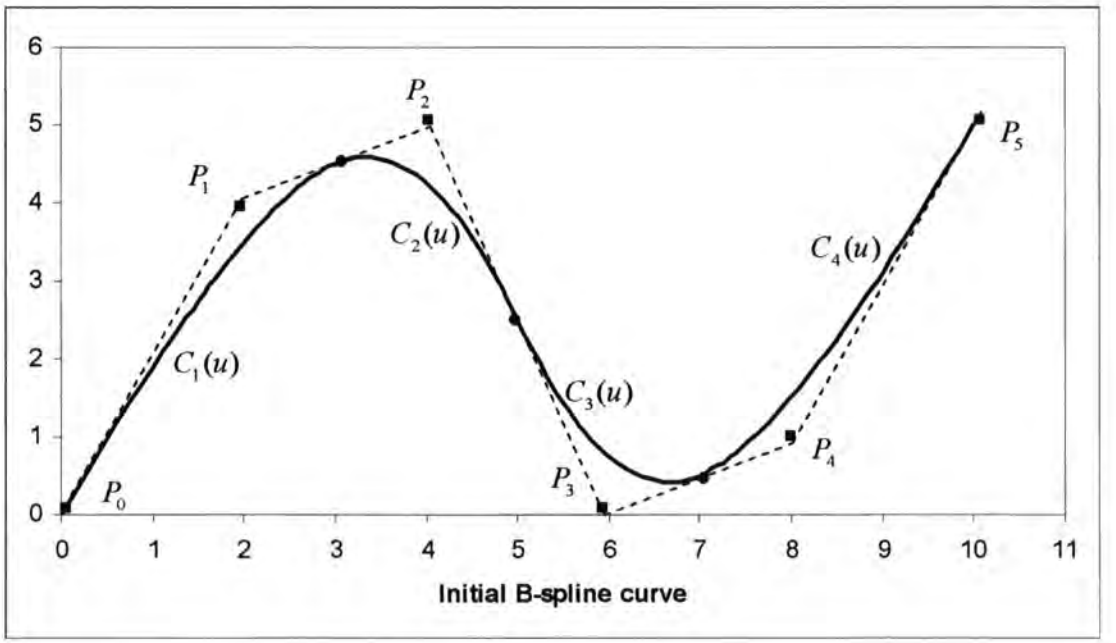


Figure 4.2: Initial B-spline curve

The B-spline curve is governed by these control points. In figure 4.3, the new B-spline curve is shown by moving $P_2(4,5)$ to $P'_2(5,6)$. This change only affects three segments of the curve. A conclusion can be achieved from figure 4.2 and 4.3 that three control points influence each curve segment and three segments are influenced by a control point.

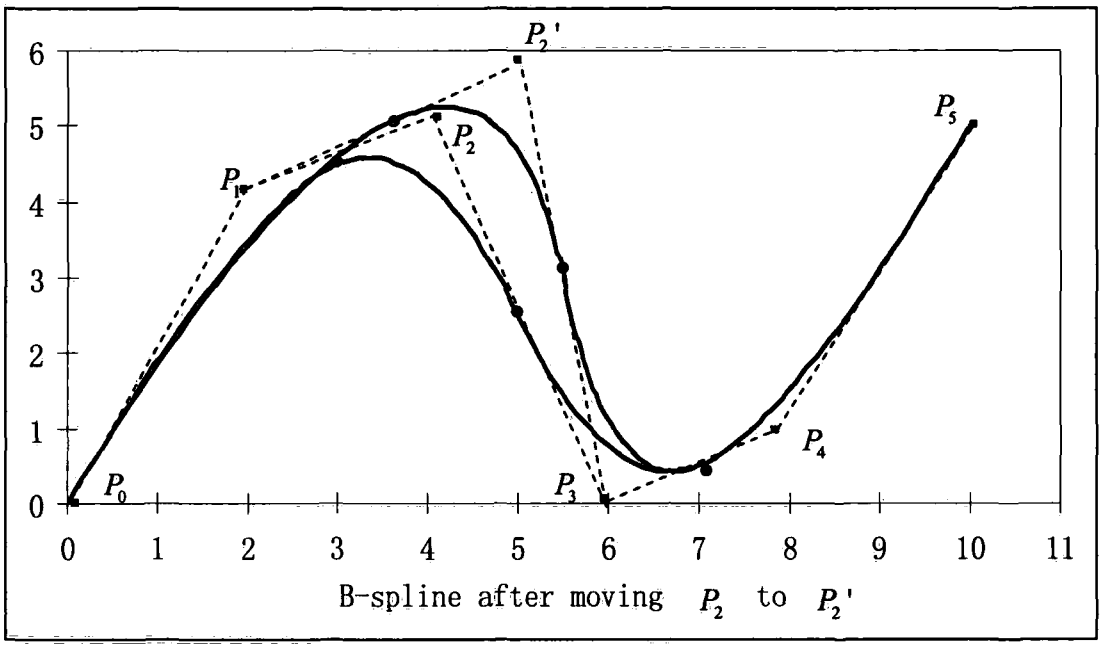


Figure 4.3: B-spline after moving P_2 to P_2'

4.2.2 Rational B-spline Curves

Rational curves are defined based on homogeneous coordinates. As the generalization of the nonrational B-spline, rational curves are more advantageous than nonrational curves, because their mathematical forms offer us to represent some standard shapes, such as conics and circles, more efficiently.

Using homogeneous coordinates the equation (4.3) can be modified to define *rational* B-splines, commonly known as NURBS (NonUniform Rational B-Splines).

Thus a NURBS curve is a vector-valued piecewise rational polynomial function of the form (Piegl and Tiller(1997))

$$C(u) = \frac{\sum_{i=0}^{n} N_{i,p}(u) w_i P_i}{\sum_{i=0}^{n} N_{i,p}(u) w_i} \qquad a \le u \le b \qquad (4.7)$$

where P_i are the *control points*, the w_i are the so-called *weights*, and the $N_{i,p}(u)$ are the p^{th} *-degree* basis functions (equations (4.4)) defined on the nonperiodic and nonuniform knot vector in equation (3.6). In most cases $a = 0$, $b = 1$ and $w_i > 0$ are assumed.

Setting

$$R_{i,p}(u) = \frac{N_{i,p}(u) w_i}{\sum_{j=0}^{n} N_{j,p}(u) w_j} \qquad 0 \le u \le 1 \qquad (4.8)$$

The equation (4.7) can be rewritten into the following equivalent form

$$C(u) = \sum_{i=0}^{n} R_{i,p}(u) P_i \qquad 0 \le u \le 1 \qquad (4.9)$$

where the $R_{i,p}(u)$ are the *rational basis functions*. They are piecewise rational functions defined on the unit interval $u \in [0, 1]$.

Figure 4.4 shows the Rational B-spline basis functions $R_{i,2}(u)$ which is obtained by equation (4.8), $0 \le u \le 1$ and $U = \{0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1\}$. Here the $w_i = 0.5$.

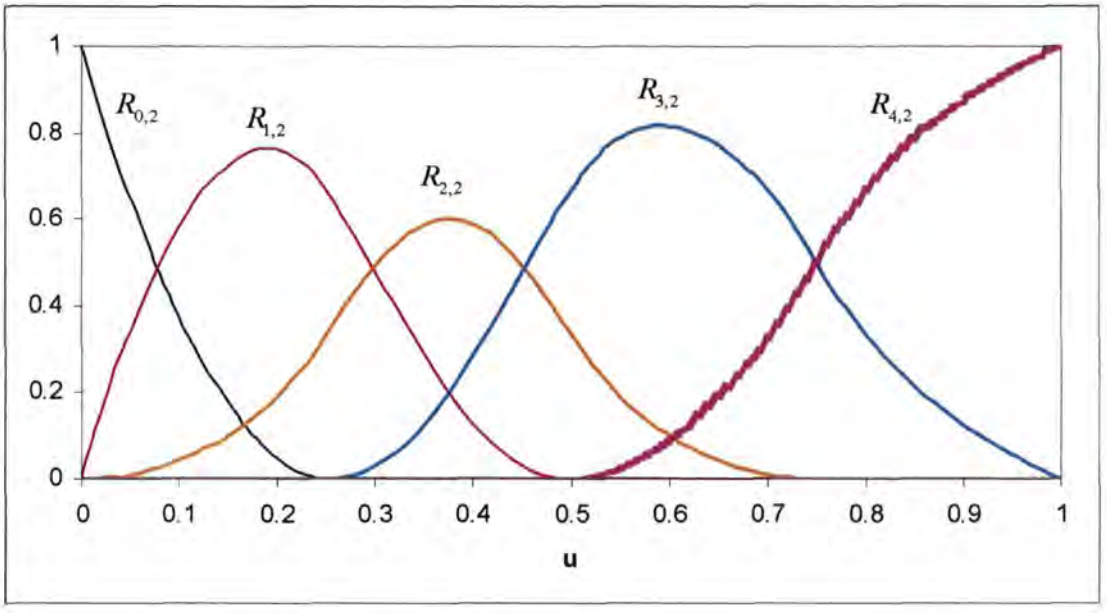


Figure 4.4: NURBS basis functions for n=5, p=2, w_i=0.5

Figure 4.5 shows the NURBS curve with five control points $\{P_0(0,0), P_1(2,4), P_2(3,1), P_3(5,0), P_4(7,4)\}$

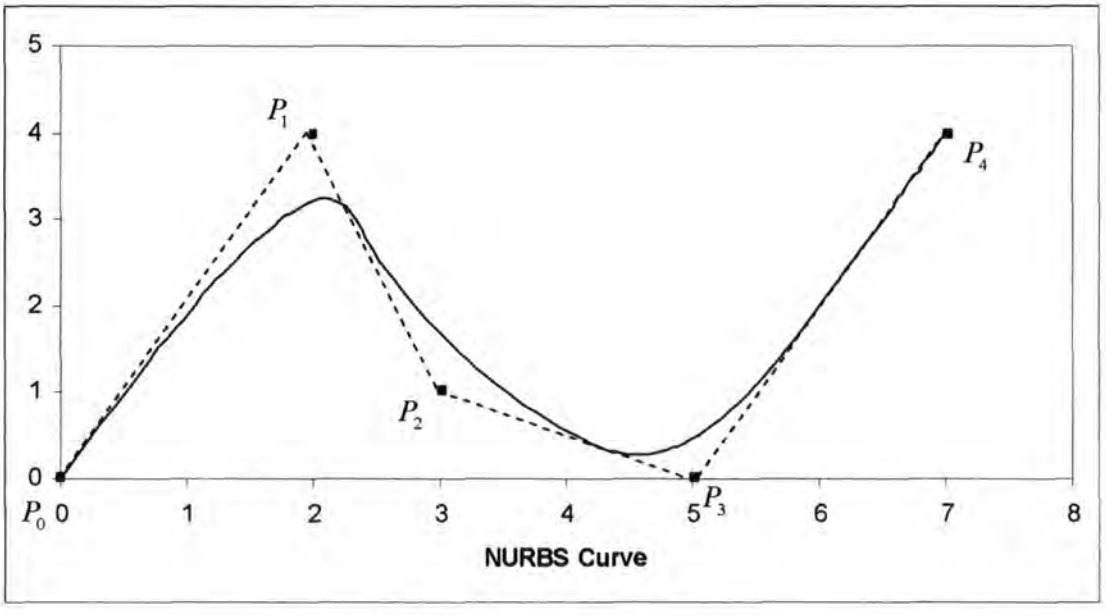


Figure 4.5: NURBS Curve

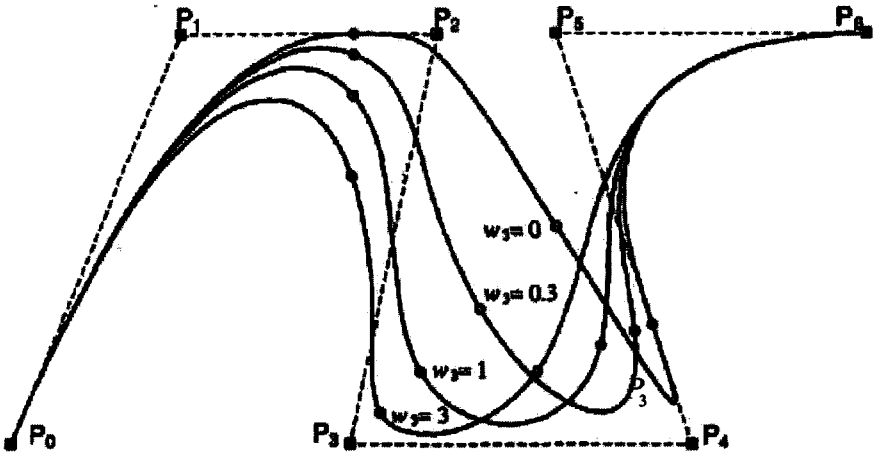


Figure 4.6: Rational cubic B-spline curves with w_3 varying (Cervera(2003))

A rational cubic B-spline curve is shown in figure 4.6. We also can see how w_i affects the curve. Here the single weight w_3 associated to P_3. If w_3 increases (decreases), the curve moves closer to (further from) P_3, and so the curve is pulled toward (pushed away from) P_3. (Cervera(2003))

5. Algorithm for ESO with BEM representation of B-splines

Cervera (2003) has developed the first ESO algorithm using boundary element and has implemented it in a program in a Microsoft Visual C++ environment. The Boundary Element Method (BEM) is used for the analysis and NURBS curves for describing the design domain where lines can change freely. NURBS are defined by their control points, and the optimisation progresses by iteratively moving these control points to accommodate a change in the geometry based on the results of a BEM stress analysis. In this part the complete optimisation algorithm is presented in detail. The main steps of evolutionary optimization are as follows:

Step 1: Geometry Definitions: The geometry of the structure is defined and the boundaries modelled by using B-splines; loads and constraints are applied;

Step 2: Structural Analysis: A boundary element analysis is carried out;

Step 3: Removal or Addition of Material: Material is removed from areas of low stress is added in areas of high stress.

Step 4: Repeat such a procedure (from **Step 2**) until a stopping criterion is reached.

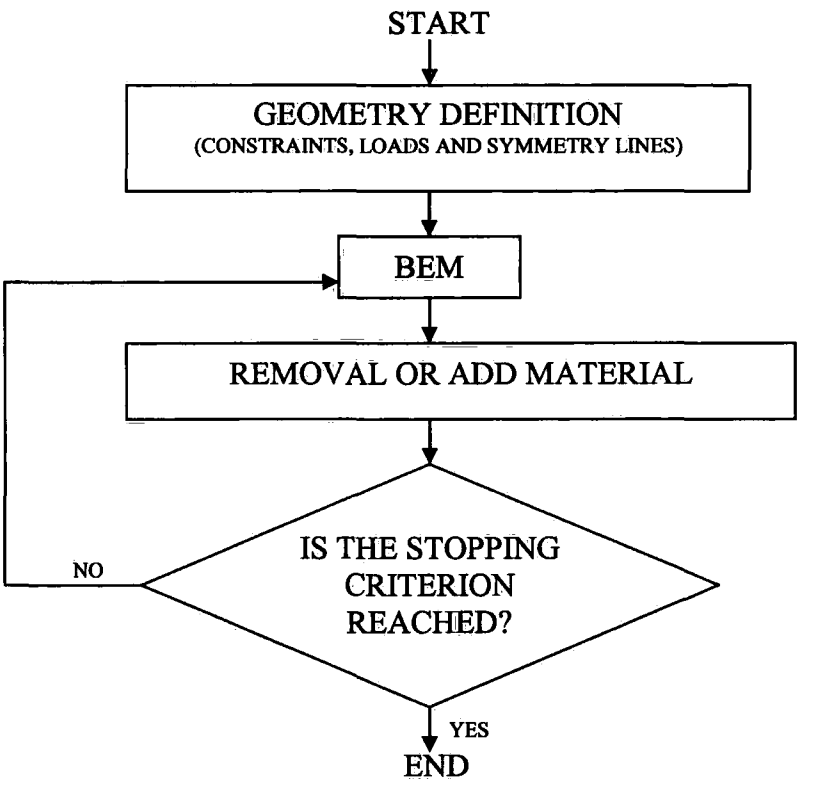


Figure 5.1: Outline of Optimisation Algorithm

5.1 Geometry Definition

According to the applied constraints, loads or some design requirements, the boundary domain can be divided into three types of curves: *changeable, non-changeable and symmetry lines.* Changeable lines can change freely in the optimisation process, whereas those lines that can not change are identified as non-changeable. Symmetry lines can change but in a limited way. They are always straight lines; therefore, their variations are restricted along the direction of the line. When the adjoining line is a changeable line, the symmetry lines can be modified by changing their length. NURBS curves define the entire changeable geometry since moving of control points is the only mechanism for modifying the geometry.

5.2 Boundary Element Analysis

In the BEM the boundary integral equations are approximated by a set of discrete integral equations. The boundary surface is divided into elements, thus the response is given at the nodal points associated with the elements. Only the surface of the structure needs to be discretised. The results inside the structure are calculated at an arbitrary number of internal points. These points are randomly distributed throughout the interior domain in the algorithm described. That is in a slight contrast with the conventional engineering usage of the method in 2D which concentrates the internal points in areas of high stress gradient to improve the fidelity of contour representations of results.

5.3 Removal and Addition of Material

5.3.1 Area to Move

The von Mises stress is chosen to drive the optimisation process. It is a useful measure since it not only provides a failure criterion that is reasonably generally applicable for ductile materials, but it is also provides a single stress value to compute the stress situation in each element. This is defined as

$$\sigma_{vm} = \sqrt{\frac{1}{2}\left[(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2\right]} \tag{5.1}$$

where σ_1, σ_2 and σ_3 are the *principal stresses*. This can be thought of as the norm of the three Mohr's circle diameters. It relates directly to the von Mises failure criterion, which is derived from the shear strain energy in a material, i.e. the strain energy associated with change in shape but not in volume. Most usefully it is a single positive number that expresses the severity of the stress situation for both tensile and compressive regions.

Material is removed from areas of low von Mises stress and added in areas of high von Mises stress. Material can be removed from the structure at any node p which satisfies

$$\sigma_{vm,p} \leq RR\sigma_{vm,\max} \tag{5.2}$$

and added to the structure if any node p satisfies

$$\sigma_{vm,p} \geq \sigma_y \; OR \; \sigma_{vm,p} \geq AR\sigma_{vm,\max} \tag{5.3}$$

where $\sigma_{vm,p}$ is the node p von Mises stress, $\sigma_{vm,max}$ is the maximum von Mises stress in the model, and σ_y is the yield stress or any other maximum stress criterion. RR is the *removal ratio* and AR is the *addition ratio* ($0 \leq RR, AR \leq 1$). These ratios are conventional in the finite element based ESO (Xie and Steven (1997)). If a steady state is reached, in other words if no nodes can satisfy equation (5.2), then the RR is increased using

$$RR_{j+1} = RR_j + ER_R \tag{5.4}$$

where j is the current iteration number and the parameter ER_R is termed the *evolutionary rate for removal*. Similarly, if no nodes satisfy equation (5.3) then the AR is decreased by the *evolutionary rate for addition*, ER_A,

$$AR_j = AR_{j-1} - ER_A \tag{5.5}$$

5.3.2 Distance to Move

At each iteration, material is either removed or added to the structure by changing the boundary definition. A set of control points is first identified that is in the near vicinity of those areas to be moved. Each set includes the three control points nearest to a specific area of high or low stress. It might be noted that in a topology

optimisation it has been found important to include only the single closest control point in the set. However, we focus in the current work on shape optimisation so will include the three closest control points.

The key parameters to guide how far the set of control points is moved are:

- Length, L_e, of the least/most stressed boundary element.
- Distances of the three control points from the least/most stressed boundary node. These distances are denoted *a*, *b*, *c*.
- An important factor related to the stress situation at the current iteration, which is called the *removal factor* (*RF*) if removing material, and the *addition factor* (*AF*) if adding.

For instance, in the figure 5.2 the control point (P_2) situated at a distance *a* from the node of lowest stress is moved a distance calculated as follows

$$Movement\ of\ P_2 = L_e \left[\frac{1}{\frac{1}{a}+\frac{1}{b}+\frac{1}{c}} \right] \frac{1}{a} RF \tag{5.6}$$

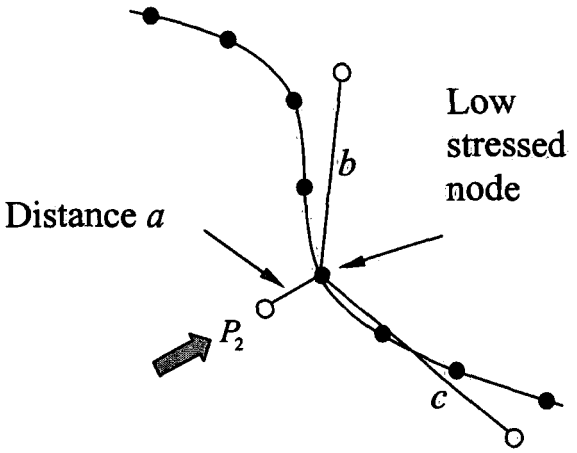


Figure 5.2: Movement when removing material

It is by reducing *RF* as the optimisation progresses that one can achieve rapid geometric change at the outset and more of a fine-tuning at the later iterations.

5.3.3 Direction of Movement

The direction of movement for each set is perpendicular to its nearest least/most stressed boundary element (figure 5.2). The material moves inwards for removal, but outwards for addition.

5.4 Stopping Criterion

According to the required aims in design, different objective functions may be used. It is inherent in the described algorithm that the objective is a fully stressed design, but we discuss approaches should other objective functions be adopted, such as some function of stiffness or strain energy. In this algorithm, a stopping criterion is defined, which is the simple method of monitoring the objective function during the progress of the algorithm. The process stops when the objective function (f) reaches a minimum/maximum. This is deemed to have been reached when the minimum/maximum value of f has not been improved upon in 10 iterations. Two objective functions have been considered in this work, a strain energy criterion and a criterion based on minimising the maximum principal stress.

Strain energy on unit volume is computed by the boundary integral

$$U = \int_{\Gamma} \frac{1}{2} \mathbf{t}\mathbf{u} d\Gamma \tag{5.7}$$

where $\mathbf{t}$ are the tractions over the boundary, and $\mathbf{u}$ the displacements over the part of the boundary Γ where the tractions are applied. Since most of the boundaries are usually traction-free, equation (5.7) reduces to integration only over those elements at which loads or constraints have been applied as boundary conditions.

It is generally desirable to minimise the strain energy. However, a simple objective function stated in this way will result in an optimum design in which the available space is completely filled with material, since this will provide the maximum stiffness and therefore minimum strain energy. It is much more useful to minimise a specific strain energy UV, where V is the volume of material in the structure.

Another objective function is related to maximum stress. It is defined as

$$\bar{F} = \min(\max Stress) \tag{5.8}$$

The term maxStress may be any desired stress component, but it is expected that maximum principal stress or von Mises stress will be the most likely to be useful. Although this method is simple, there is no guarantee that the resulting shape is actually the optimum for any arbitrary objective function except the fully stressed design. A better method is to use sensitivity analysis, which will be presented in the next chapter.

5.5 Geometry Control

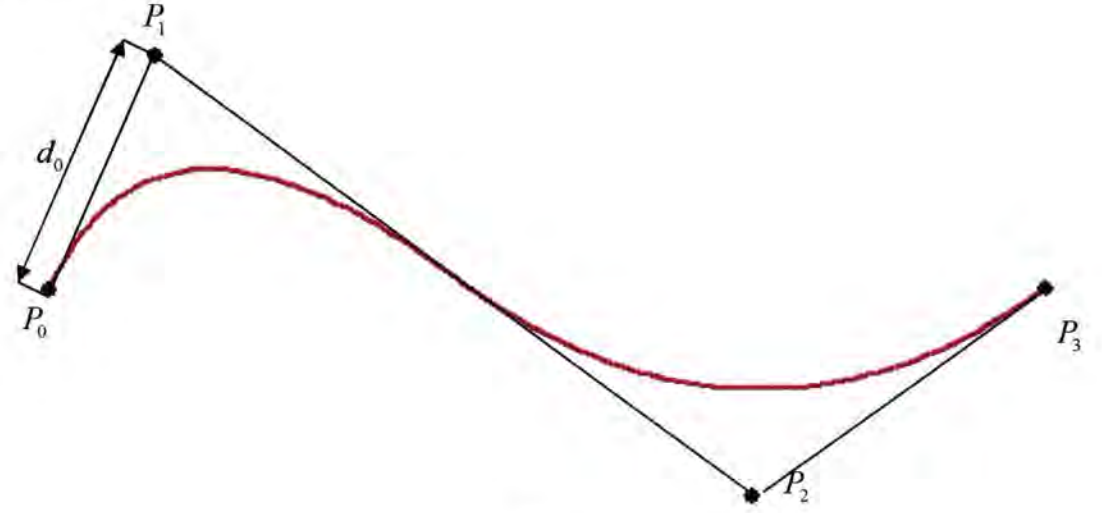
The geometry is represented by NURBS whose definition is governed by a set of control points. Those control points can move freely for controlling the geometry change. A useful feature of NURBS is the fact that each control point has an influence over only a localised portion of the spline. This allows a detailed and localised control of the geometry through movement of control points as suggested by local stress distributions. In spite of this, and other benefits, numerical tests have shown that the algorithm presented in section 5.3 tends to distort, elongate or compress splines as the optimisation progresses. This ultimately causes a degradation of control over the spline geometry. This can be rectified by periodically inserting and/or deleting control points as required.

5.5.1 Insertion of Control Points

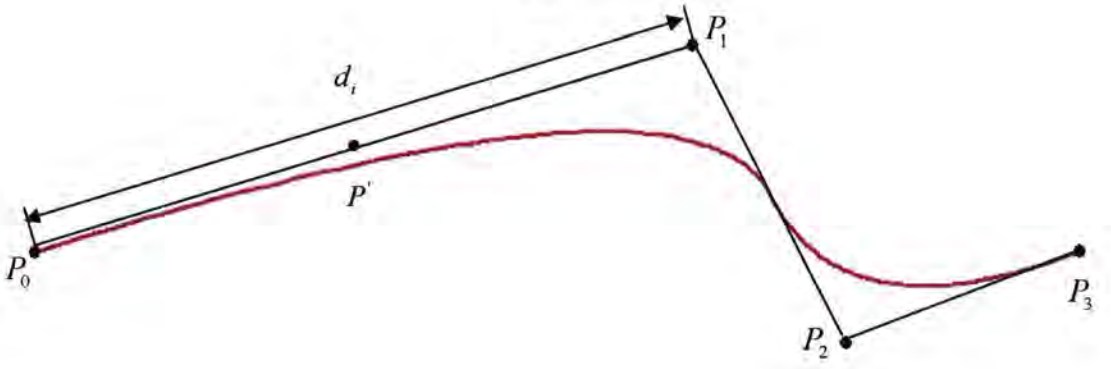
When the distance between two existing control points becomes too large to control the boundary geometry in sufficient detail, a new control point is considered to be inserted between them. The insertion is implemented when the distance satisfies

$$d_i > k_{Insert} d_0 \tag{5.9}$$

where k_{Insert} is a constant factor ($k_{Insert} \geq 1.5$), d_0 is the initial distance between two existing control points, i is the iteration number and d_i is the distance at iteration i. Cervera (2003) used $k_{Insert} = 1.5$ for topology optimisation and 2.0 for shape optimisation.



(a)



(b)

Figure 5.3: Insertion of Control Point

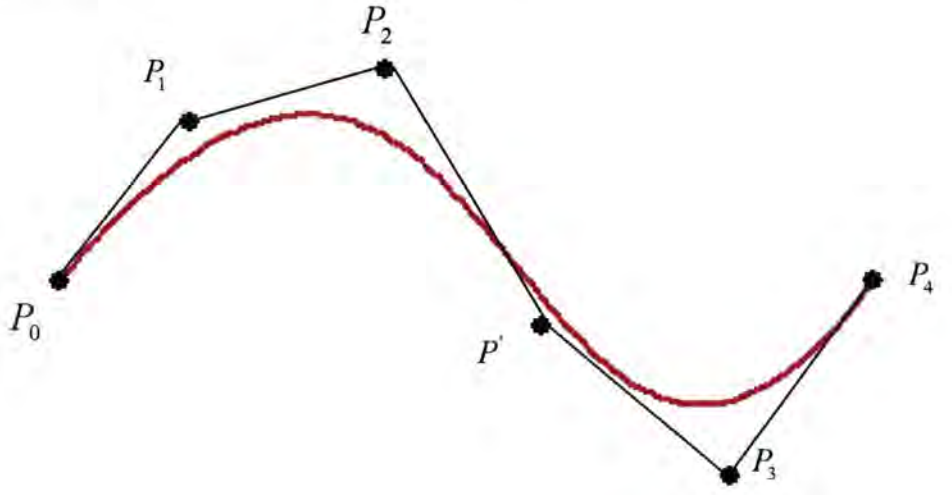
Figure 5.3(a) shows the initial placement of the control points. At the iteration i, the movement of the control points leads to the distance between P_0 and P_1, d_i, satisfying the equation 5.9. The new control point P' is inserted at the mid-point of P_0 and P_1 as shown in figure 5.3(b), and the algorithm advances as before.

5.5.2 Deletion of Control Points

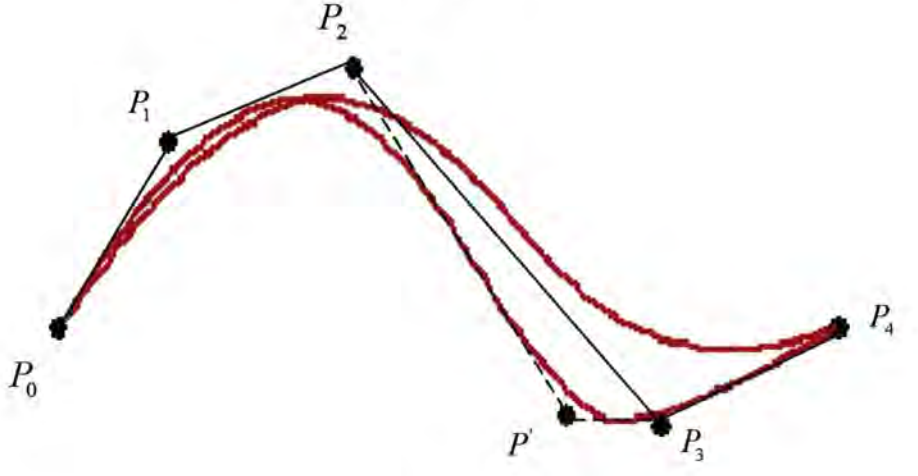
When the locations of a set of control points are too close, a control point is deleted to avoid localised boundary distortions. The lengths of the lines of the control polygon provide the criterion to delete the point P_j using

$$\left|P_j - P_{j-1}\right| < k_{delete}\left|P_{j-1} - P_{j-2}\right|$$
$$OR\ \left|P_j - P_{j-1}\right| < k_{delete}\left|P_{j+1} - P_j\right| \qquad (5.10)$$

where $|P_j - P_{j-1}|$ is the length of the line checked, $|P_{j-1} - P_{j-2}|$ and $|P_{j+1} - P_j|$ are the lengths of the previous and next control polygon lines and k_{delete} is constant factor ($k_{delete} \leq 1.5$).



(a)



(b)

Figure 5.4: Deletion of Control Point

Figure 5.4(a) shows the initial spline geometry, which during the optimisation process becomes changed to the shape shown in figure 5.4(b). Now the distance between P' and P_3 satisfies $|P_j - P_{j-1}| < k_{delete}|P_{j-1} - P_{j-2}|$, so the control point P' is deleted and the curve is updated.

5.5.3 Subdivision of Distorted Elements

It is not unusual that an element becomes distorted in optimisation algorithm. The optimisation runs too fast to achieve the stable geometry. The boundary element mesh is generated automatically at each iteration in this algorithm, of course, to allow for the required speed of progressing the optimisation. The automesh is carried out by comparing element sizes and then reducing the larger element of outward and inward element (start and end element as well) to fit the special grading ratio. Considering the character of the B-spline, an individual element may need to be subdivided to maintain suitable solution accuracy if it is located in a portion of the B-spline that exhibits a high degree of curvature. An element is subdivided if its length satisfies

$$L_e > m\left|p_j - p_{j-1}\right| \tag{5.11}$$

where L_e is the length of the element, $|p_j - p_{j-1}|$ is the distance between its end nodes and m is constant factor ($m \geq 1.05$).

5.6 Topology Optimisation

By topology optimisation we mean a shape optimisation that is further enhanced by the ability to insert, delete and merge holes. Holes may be inserted at the low stressed areas to realise the topology optimisation. The criterion to identify low stressed internal regions is similar to the one to the outside domain, as expressed in equation 5.2, and may be written

$$\sigma_{IP} \leq RR\sigma_{\max} \tag{5.12}$$

where σ_{IP} is the von Mises stress at an internal point, $\sigma_{\max}$ is the maximum von Mises stress in the model and RR is the removal ratio.

The process of inserting holes starts with calculating the von Mises stress for any internal points by equation 3.29. The internal point whose von Mises stress is the minimum locates the centre of the hole, and then other internal points around it are taken as the control points to create a new polygon.

For the internal holes, the distance and direction to move the control points are both similar to the outside domain, though some important differences are described by Cervera and Trevelyan (2005a) relating to holes that approach external boundaries.

The distance the control points are moved is related to:

- Length of the least/most stressed element, L_e;
- Distance of the control point from the least/most stressed boundary node, a;
- The *removal factor* (*RF*) if removing material, and the *addition factor* (*AF*) if adding;
- The width, *w*, of a narrow strip of material between the hole and an external boundary.

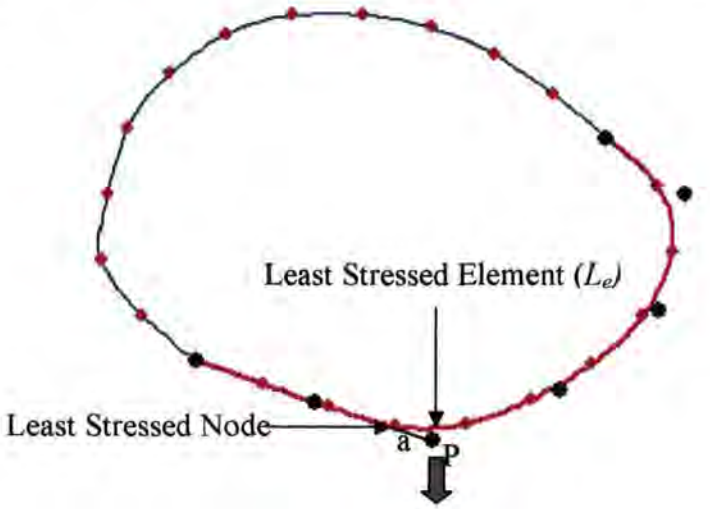


Figure 5.5: Movement of the Control Point in the Internal Holes

The direction of movement is generally perpendicular to the nearest least/most stressed element, unless in the case that the hole is very close to the boundary, in which case the direction of movement is parallel to the boundary which is the closest to the control point.

5.7 Concluding Remarks

This section has described in detail the algorithm of Cervera. We note here that it is specifically the values of parameters *RF*, *RR* and ER_R that are the subject of the current work. Cervera & Trevelyan (2005(a)(b)) have developed a corresponding algorithm for shape optimisation in 3D problems based on the use of NURBS surfaces as a boundary description. Similar governing parameters are used. However, the scope of the current work is limited to plane stress problems.

6. Sensitivity Based Optimisation

Sensitivity analysis (SA) is a powerful tool that can play an important role in any modelling field. In engineering design, SA extends the engineer's information about the behaviour of a complex model and helps him/her make decisions in guiding the design. In a design optimisation process sensitivities are the gradients of the objective functions, so that we define a design sensitivity, s_i, as

$$s_i = \frac{\partial f}{\partial Q_i} \tag{6.1}$$

where f is the objective function considered in the optimisation and Q_i is the i^{th} design variable.

Supposing that the sensitivity can be evaluated for each design variable in the process, the engineer is provided with readily useable information to determine the next design iteration. For example, in the optimisation for minimum weight of a pin-jointed frame or truss, the design variables are often the Cartesian coordinates of the member connection points, or nodes. At iteration p, we calculate the sensitivities, being the partial derivatives of the weight with respect to these coordinates, and in determining the geometry for iteration p+1 the engineer might choose to move the nodes in proportion with the sensitivity values.

Design sensitivities are the central feature of the gradient based methods of optimisation, and there are numerous works that describe their features and numerical stability, as well as examples of their application. A boundary element implementation, for example, is described by Tafreshi and Fenner (1991).

Sensitivities may be obtained in two ways:

- Analytical differentiation. Here the governing matrix system is differentiated with respect to the design variables. In a BEM context, this means that the fundamental solutions used in the integration and assembly stage are found in this derivative form, and this gives the most accurate sensitivity. The sensitivity is found to the same degree of accuracy as the boundary solution, and no further approximations are made. It can arrive at the analytical derivative for other than a trivial geometry only to the same degree of approximation as the boundary element solution. So it is not really a truly analytical derivative.

- Finite difference method. Here the model is rebuilt following a change in one of the design variables (e.g. a coordinate value is moved) and a second analysis is performed. The objective function is evaluated for both the initial and changed models allowing direct computation of the sensitivity. Mathematically, if design variable Q_i is changed to $Q_i + \Delta Q_i$, causing a change in the objective function from f to $f + \Delta f$, the sensitivity may be given by

$$s_i = \frac{\Delta f}{\Delta Q_i} \tag{6.2}$$

  This is less accurate than the analytical differentiation method, because of the finite Δf (in fact the analytical differentiation approach may be considered as the limit of the finite difference approach as $\Delta f \rightarrow 0$). The attractiveness of the method lies in its ease of computation.

In the current work gradient based optimisation is not applied, of course, and instead we use the evolutionary structural optimisation approach as described in chapter 5. This does not necessarily rule out the use of sensitivity information. The approach has been used successfully in a finite element evolutionary structural optimisation implementation by Steven *et al.* (2002), in a scheme which guarantees to achieve a true optimum. They presented a few element-based sensitivities, such as stiffness sensitivity, displacement sensitivity, stress sensitivity, and so on, and those

sensitivities are also used to achieve a minimum or maximum of some composite fitness functions or certain values to drive the evolutionary strategy. We now discuss how design sensitivities can be used in a BEM-based ESO procedure. This part presents the basic of sensitivity analysis briefly and its use in structural optimisation.

6.1 SA in Structural Optimisation

As described in detail in chapter 2, the main concept of ESO is the removal and/or addition of material, in response to stress levels, in order to improve structural efficiency. However, the method has significant limitation in the shape optimisation of design geometry for an arbitrary objective function. However, it is quite common that a specific objective function should be desirable that is not compatible with the ESO approach. For example, one might desire to converge to a solution in which the stress is uniform around a hole, or around multiple holes.

One way of including arbitrary objective functions in an ESO framework is simply to monitor the objective function as the iterations progress (Cervera (2003)). A stopping criterion might be defined such that the process halts if, at iteration $p + q$ the objective function has not improved upon its value at iteration p. For example, the objective function reaches a minimum at iteration 60 ($p = 60$) and in all iterations between 60 and 68 (if $q = 8$) the objective function at iteration 60 has not been bettered. This accounts for the typically non-monotonic convergence of this type of scheme, by allowing the objective function to become worse in the hope that it will later improve further. The value of q is a measure of how long we are prepared to wait before we conclude that the optimum has been reached.

This approach is valuable, but provides no guarantee that the design at iteration p is in fact the optimum design for this objective function. We can say only that it is the best design among all designs passed through during the stress-level driven ESO optimisation process.

In this work, we use the finite difference approach, so that at each iteration the control points are individually perturbed a small amount in each coordinate direction

and an analysis performed to determine the associated change in the objective function at each iteration. Equation (6.2) provides the sensitivities with respect to both the x- and y-coordinates (separately) of each control point. We denote the sensitivity s_{ij} where

$$s_{ij} = \frac{\Delta f}{\Delta x_{ij}} \tag{6.3}$$

where x_{ij} is the coordinate of the i^{th} control point in the x_j direction ($x_1 = x$, $x_2 = y$).

In determining the design geometry for the next iteration we move the control points such that

$$x_{ij}^{p+1} = x_{ij}^{p} + \frac{s_{ij}}{s_{\max}} d \tag{6.4}$$

where $s_{\max}$ is the maximum value of sensitivity found in the iteration, d is some predetermined stepsize usually a function of the model dimension, and the superscript on the x variables denotes the iteration.

As a final remark, the use of finite difference sensitivities in preference to analytical differentiation of the fundamental solution is adopted for the following reasons:

- Ease of implementation and control

- The fact that our design domain is defined using B-splines so that each control point exerts an influence only over the local geometry. This means that each analysis for a perturbed geometry may be efficiently performed using a reanalysis (see section 3.2.4). Only the few elements in the immediate vicinity of the perturbed control point will change location and so only a small portion of the governing system matrix will become changed.

- We recognise that in defining a change in geometry from iteration p to iteration p+1 we will be moving the control points a finite distance given by (6.4). It is not difficult to make a case that the accuracy of the sensitivity using finite differences will be greater than that using analytical differentiation if $\Delta x_{ij} \approx d$ and we define the accuracy of the sensitivity as the ability to predict the value of the objective function at iteration p+1.

7. Parameter Selection for Shape Optimisation

Chapter 5 included a description of an evolutionary structural optimisation algorithm based on a boundary element discretisation of a B-spline representation of the problem boundary. The optimisation can proceed by defining where the material should be removed or added and how much material should be removed or added at each iteration step. The algorithm uses a small number of key parameters to define them. This part presents some numerical experiments that aim to determine suitable values of these parameters in order to produce good optimal designs with the best computational efficiency.

7.1 Introduction

In the algorithm, the main parameters for optimisation are *RR*, ER_R, *RF*, *AR*, ER_A and *AF*. Their definition has been presented in chapter 5. *RR* is the *removal ratio* and *AR* is the *addition ratio,* which govern the areas in which material should be removed or added. ER_R and ER_A are termed the *evolutionary rates*. *RF* and *AF* are parameters which govern the distance through which control points are moved. In traditional ESO, typical values determined from numerical experience are $RR_0 = 0.01$, $ER_R = 0.01$, $AR_0 = 0.99$, $ER_A = 0$ (Cervera (2003)). Although small values of these parameters tend to produce the most optimal designs, it is also desirable to consider the computational efficiency. For different requirements in design, saving the calculation time sometimes is regarded as the emphasis.

Two problems of different character are selected as test cases. One is a short cantilever beam with whole geometric change during ESO process, the other is a fillet with small geometric change. The following isotropic material properties are assumed: Young's modulus $E = 210000 N/mm^2$, Poisson's ratio $\nu = 0.3$ and an arbitrary thickness $t = 1mm$.

7.2 Short Cantilever Beam

Figure 7.1 shows a problem that is classically called a short cantilever beam. This consists of an initial domain (figure 7.1(a)) of height 100mm and width 50mm subjected to a vertical load over a short length of boundary on the right edge of the domain, with displacement constraints applied at the top and bottom of the left hand edge. The non-design domain where lines can not change freely consists only of the short boundary segments over which loads and constraints are applied, so all other boundaries form the design domain where lines can change freely. Classically the problem converges to a minimum weight optimum of a two-bar frame (Xie and Steven (1993)) as illustrated in figure 7.1(b).

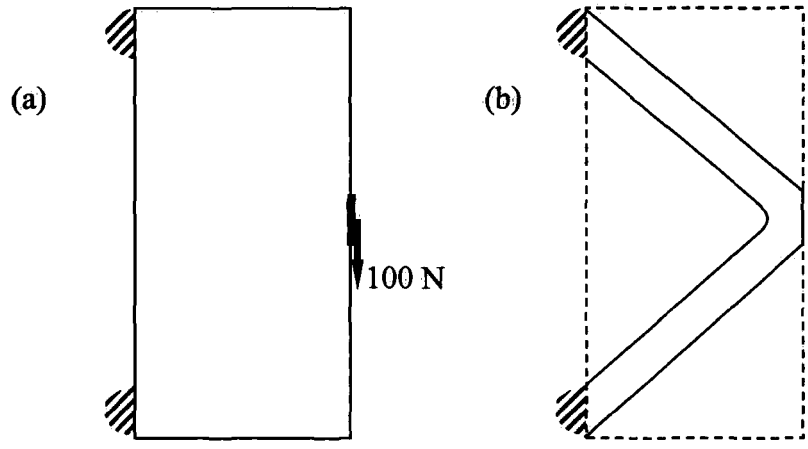


Figure 7.1: Short cantilever problem;
(a) original domain, (b) form of optimum solution

The objective function, F, to be minimised in this problem is based on a specific strain energy criterion

$$F(i)=U(i)\,V(i) \tag{7.1}$$

where i is the iteration number, U is the strain energy in the material and V is the volume. We define variables M and n by which we characterise the performance of an ESO optimisation scheme. We write

$$M=minimum\ (F(n)) \tag{7.2}$$

where M is the minimum value of F that is achieved at any iteration, and n is the iteration number at which F is minimised.

Based on the numerical experience gained during the current work, in addition to values reported in previous work, we consider a range of parameters in our study to be $0.005 \leq RR \leq 0.095$, $0.0005 \leq ER_R \leq 0.0505$, $0.05 \leq RF \leq 0.75$. The short cantilever beam model has been optimised a total of 14535 times using different combinations of the parameter values within these ranges. In order to analyse the results of such a large number of runs, we break the three-dimensional parameter space (RR, ER_R, RF) into 105 subspaces, each containing a smaller region, and form the mean values of M and n, to be denoted below using overbars, within each subspace. Figure 7.2 shows ranges of these parameters. It is made up of 105 small cubes. Each small cube contains many sets of parameters values that have been tested. RR, ER_R and RF have 19, 51 and 15 values respectively within their respective ranges defined above.

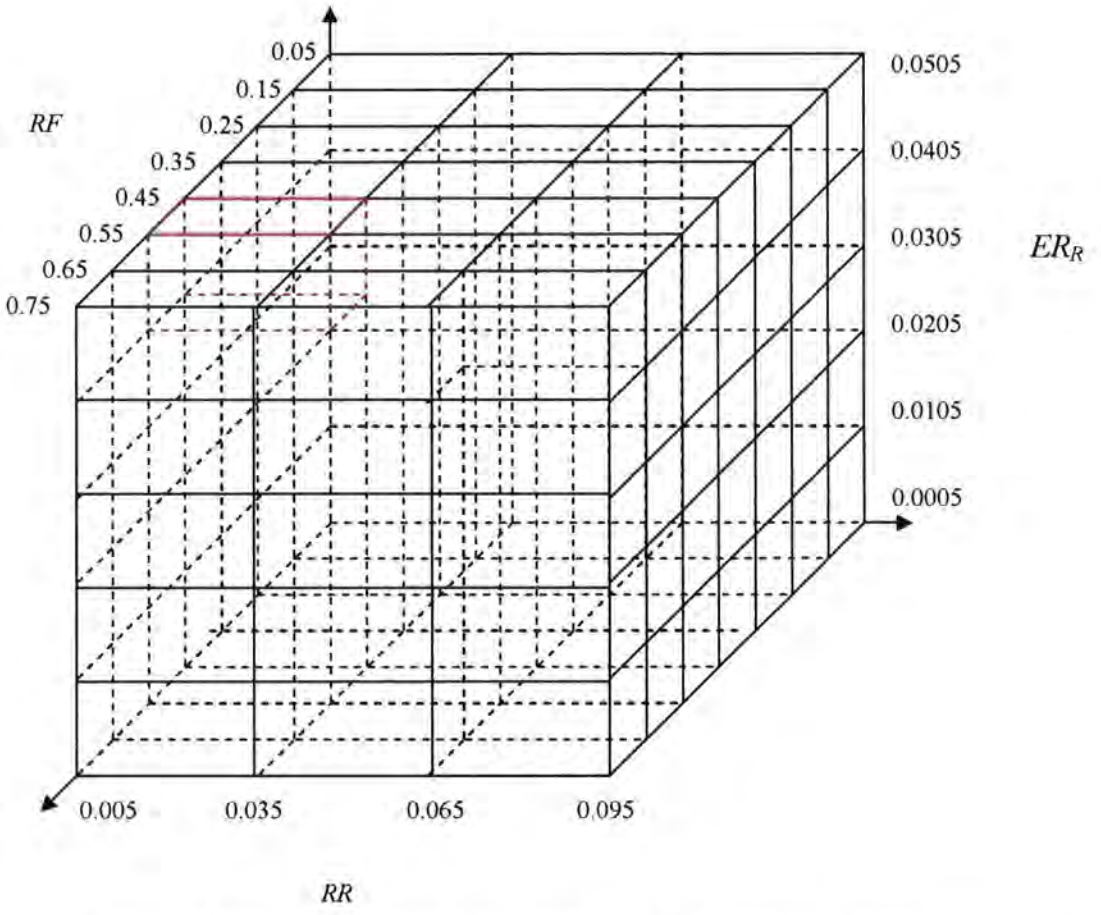


Figure7.2: Diagrammatic representation of the parametric space used

From these results, the values of M and n are most sensitive to the value of the parameter RF. That means that RF is the most important parameter in the evolution. It was therefore decided to focus on a broad grouping of parameter subspaces so that the mean effects of variation in RF could be investigated over the entire range of RR and ER_R. Table 7.1 shows the average M and n results for the subspaces contained in various RF ranges as shown. As might be expected the best optimum, *i.e.* minimum value of $\overline{M}$, occurs with the smallest RF, but an acceptable optimum is obtained in far fewer iterations in the subspaces encompassing $0.5 \leq RF \leq 0.55$.

RF	0.3-0.35	0.4-0.45	**0.5-0.55**	0.6-0.65
$\overline{M}(Nmm^4)$	16464	16698	**16519**	16551
$\overline{n}$	89	94	**51**	52

Table 7.1: Mean performances over range of RF values

In order to establish the best values of the other parameters, we consider the subspaces over which $0.5 \leq RF \leq 0.55$, and look in more detail at the effects of RR and ER_R. The best portion of these subspaces is shown in table 7.2.

ER_R \ RR	0.005-0.035		0.04-0.065	
	$\overline{M}(Nmm^4)$	$\bar{n}$	$\overline{M}(Nmm^4)$	$\bar{n}$
0.0315-0.0405	16631	50.7	**16432**	**48.4**
0.0415-0.0505	**16593**	**48.8**	16768	51.8
0.0515-0.0605	16596	48.9	17038	49.4

Table 7.2: Mean performances over range of ER_R and RR values

According to the data in Table 7.2, we consider that the best combination of parameters is in the region of $RR \approx 0.05$, $ER_R \approx 0.04$ and $RF=0.5$. The best M is achieved in this region, requiring 31 iterations to convergence. The evolution of the geometry and of the stress distribution for this 'optimal' set of parameters is shown in figure 7.3. No attempt should be made to compare stresses between the four solutions at the various iterations. The contours show distributions only, and the values of stress defining each contour colour are different in the four cases. Figure 7.4 shows the corresponding evolution of the objective function. The optimum of M =16144 at the 31st iteration is highlighted.

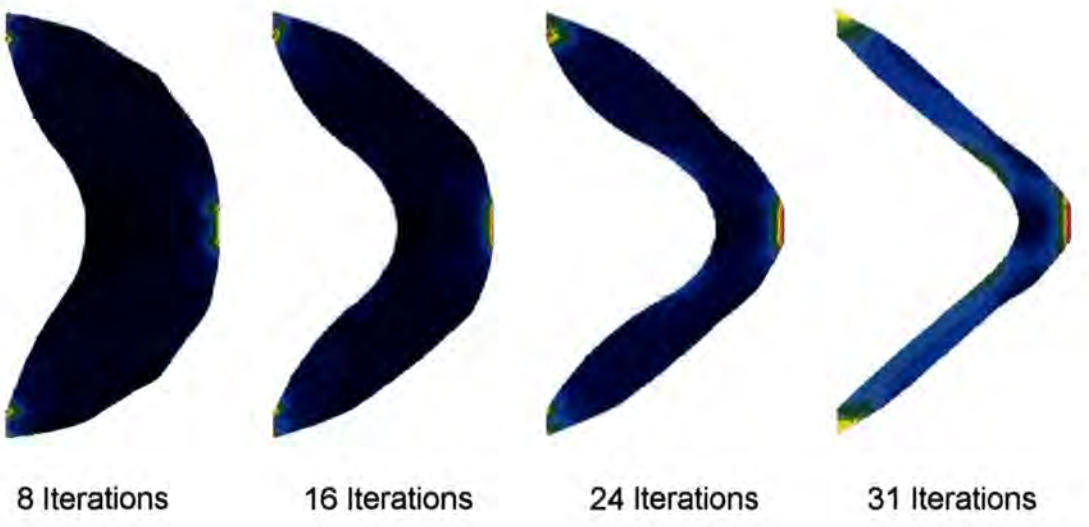


Figure 7.3: Evolution of the short cantilever beam example
(von Mises stress contours)

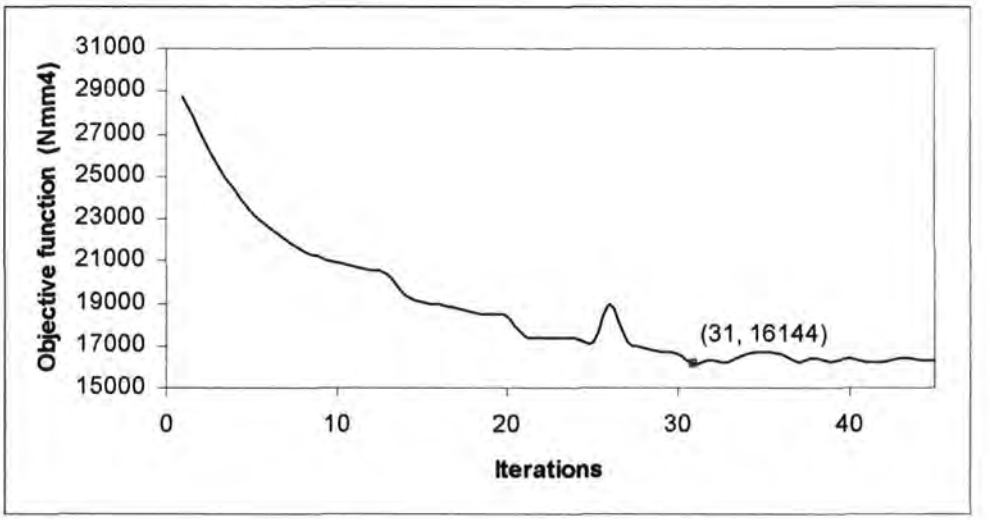


Figure 7.4: Evolution of objective function

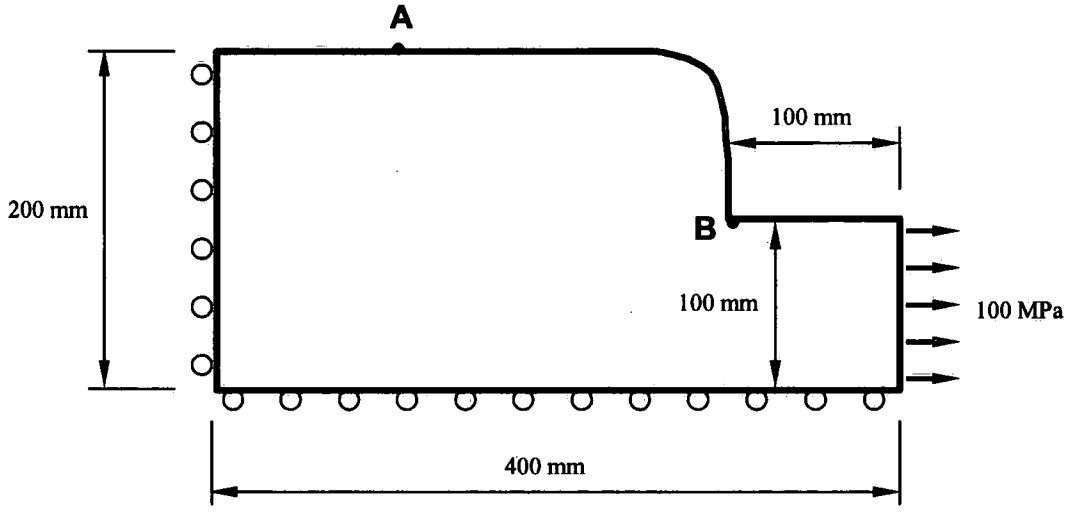
We note that, since RR is incremented many times during the process using equation (5.4), this initial value is of significantly lower importance than ER_R.

7.3 Fillet

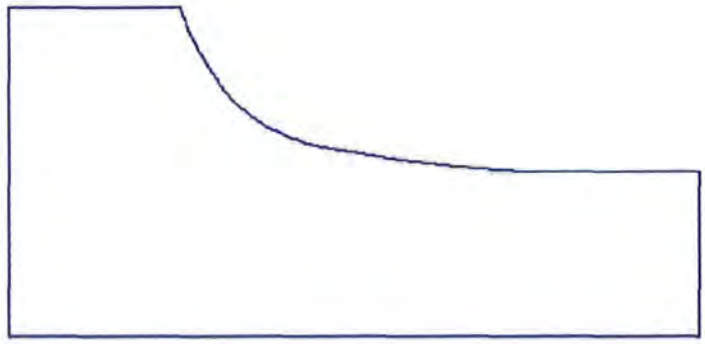
The second problem of interest is the optimisation of a fillet profile (figure 7.5) to minimise the maximum principal stress in the fillet. Here only the line AB forms the design domain and the problem takes on an altogether different character to the first problem. In addition to the two problems having different objective functions, the short cantilever beam optimisation involves whole geometry changes with removal of over half the initial volume, while the fillet optimisation is more of a fine tuning of geometry. This difference in character has been reflected in the literature by different values of governing parameters being applied. The current work aims to determine a unified set of parameter values along with a suitable algorithm that will be successful for both problems.

In this problem, we use a minimax condition as the objective function.

$$F = \min(\max(\sigma_1)) \tag{7.3}$$



(a)



(b)

Figure 7.5: Fillet example; (a) original domain, (b) form of optimum solution

As an initial exercise for this problem, we investigated the optimisation process using the governing parameters $RR = 0.05$, $ER_R = 0.04$, $RF = 0.5$ which are considered the best sets in the first problem. The evolution proceeds as depicted in figure 7.6.

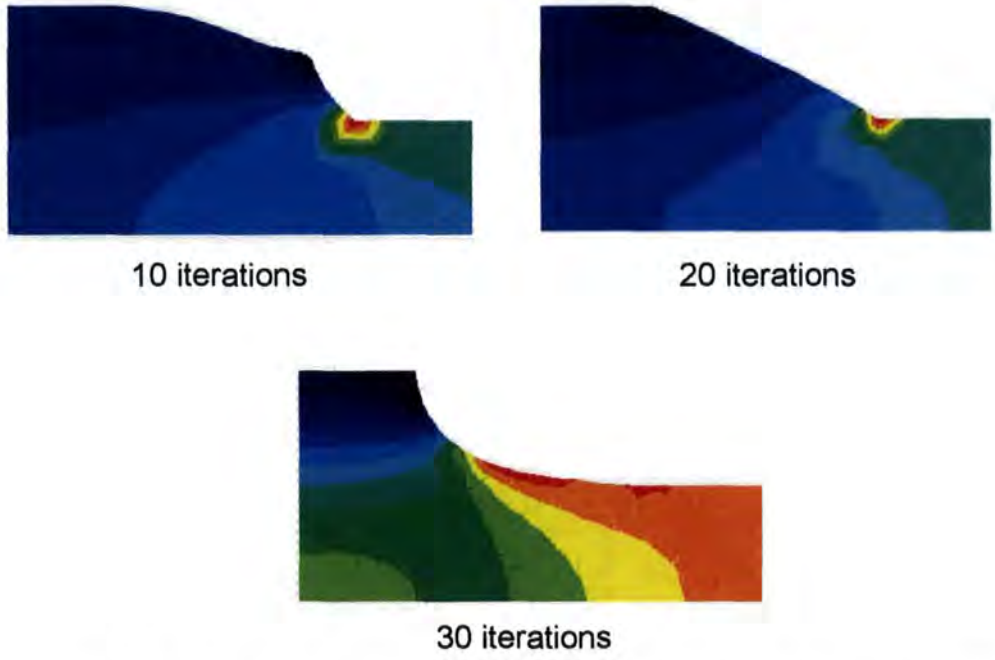


Figure 7.6: Evolution of the process for fillet optimisation (30 iterations)

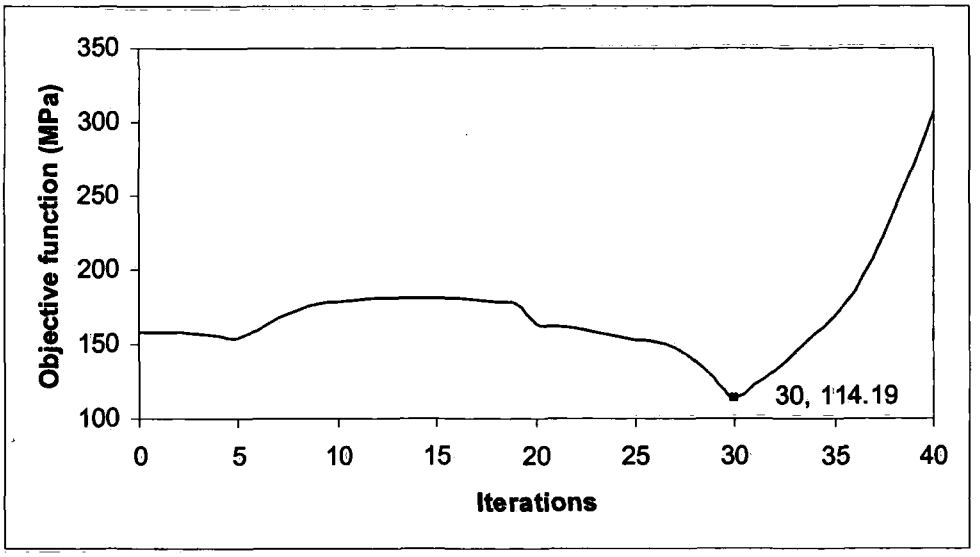


Figure 7.7: Evolution of the objective function

Figure 7.7 shows the evolution of the objective function. A minimisation of the objective function is found at the 30th iteration. It is interesting to comment on the shape of the curve. While the corresponding plot for the short cantilever beam example (figure 7.4) presents a very classical convergence to an optimum, figure 7.7 shows a rather different character. Indeed, it is only at around the 20th iteration that the objective function starts to improve on the initial model. We speculate that this is due to the fact that the objective function in the first example more closely matches that inherent in the ESO scheme, which is aiming to minimise weight to give a fully stressed design. The relatively flat nature of the curve also reflects the nature of the problem being solved, i.e. one of fine tuning and not one of whole geometric changes.

The results of the short cantilever example allow us to refine our range of parameters. In addition, we can reduce the number of optimisation runs by using a more sophisticated scheme to optimise the performance of the algorithm. Specifically, since we are aiming to minimise both M and n, this becomes a multi-objective optimisation problem in its own right. We will therefore apply classical methods of multi-objective optimisation to the determination of the best values of the governing parameters. The fillet example was therefore run 96 times using different values of

the parameters $0.035 \leqslant RR \leqslant 0.075$, $0.02 \leqslant ER_R \leqslant 0.05$ and $0.2 \leqslant RF \leqslant 0.5$. The results are shown in table 7.3.

RR / ER_R	0.035		0.045		0.055		0.065		0.075		0.085	
	M	*n*	*M*	*n*	*M*	*n*	*M*	*n*	*M*	*n*	*M*	*n*
0.02	117.20	62	119.61	76	113.5	83	112.83	96	112.44	101	112.65	100
0.03	116.92	64	119.08	72	113.54	89	114.09	92	112.44	101	112.54	103
0.04	118.11	60	119.63	71	113.52	77	113.31	95	112.90	94	113.07	91
0.05	117.24	64	120.73	79	114.65	83	114.38	82	**112.01**	**86**	112.42	107

(a) RF=0.2

RR / ER_R	0.035		0.045		0.055		0.065		0.075		0.085	
	M	*n*	*M*	*n*	*M*	*n*	*M*	*n*	*M*	*n*	*M*	*n*
0.02	117.66	40	121.24	54	114.97	57	114.40	62	112.87	65	113.32	69
0.03	118.51	36	117.21	43	114.89	60	113.20	60	114.55	56	113.26	69
0.04	116.98	39	119.61	49	115.19	51	114.6	62	113.50	59	113.06	62
0.05	117.75	43	119.64	46	114.97	55	115.27	53	115.05	53	114.29	72

(b) RF=0.3

RR / ER_R	0.035		0.045		0.055		0.065		0.075		0.085	
	M	*n*	*M*	*n*	*M*	*n*	*M*	*n*	*M*	*n*	*M*	*n*
0.02	118.26	29	115.92	41	117.13	44	115.83	47	113.53	49	113.60	49
0.03	117.01	31	122.68	36	114.81	45	115.3	44	114.26	49	114.59	48
0.04	117.09	30	118.83	31	113.89	39	115.17	45	113.35	45	114.84	44
0.05	116.73	29	122.39	39	116.52	40	117.1	40	114.93	41	113.35	51

(c) RF=0.4

RR / ER_R	0.035		0.045		0.055		0.065		0.075		0.085	
	M	*n*	*M*	*n*	*M*	*n*	*M*	*n*	*M*	*n*	*M*	*n*
0.02	119.46	23	119.79	30	118.76	31	115.32	36	114.62	40	114.0	37
0.03	115.47	25	122.30	29	113.85	32	113.80	33	112.94	40	115.0	37
0.04	119.82	24	121.08	28	117.63	29	113.78	33	115.5	36	114.26	35
0.05	118.93	25	122.02	25	114.14	30	115.24	30	116.46	33	114.85	39

(d) RF=0.5

Table 7.3: Results for the fillet problem

From the above data, the best optimum is found using the parameters $RR = 0.075$, $ER_R = 0.05$ and $RF = 0.2$. Using this set of parameters the evolution proceeds as shown in figure 7.8. At the 86th iteration the objective function has reduced to a minimum of F = 112.01MPa.

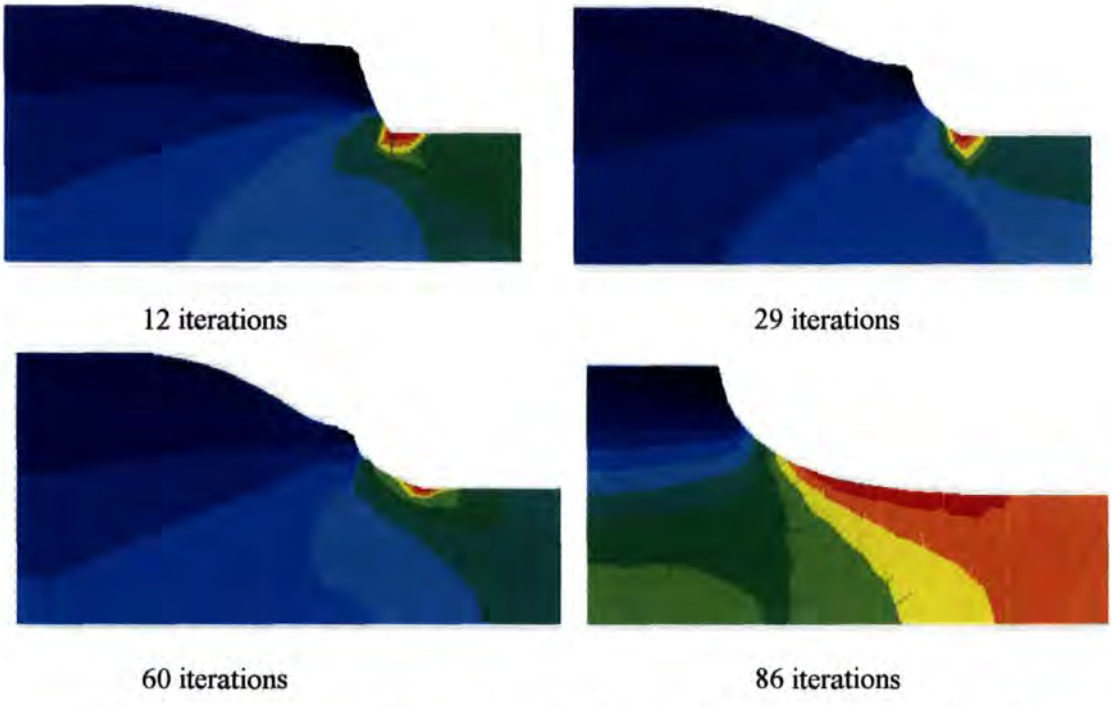


Figure 7.8: Evolution of the process for fillet optimisation (86 iterations)

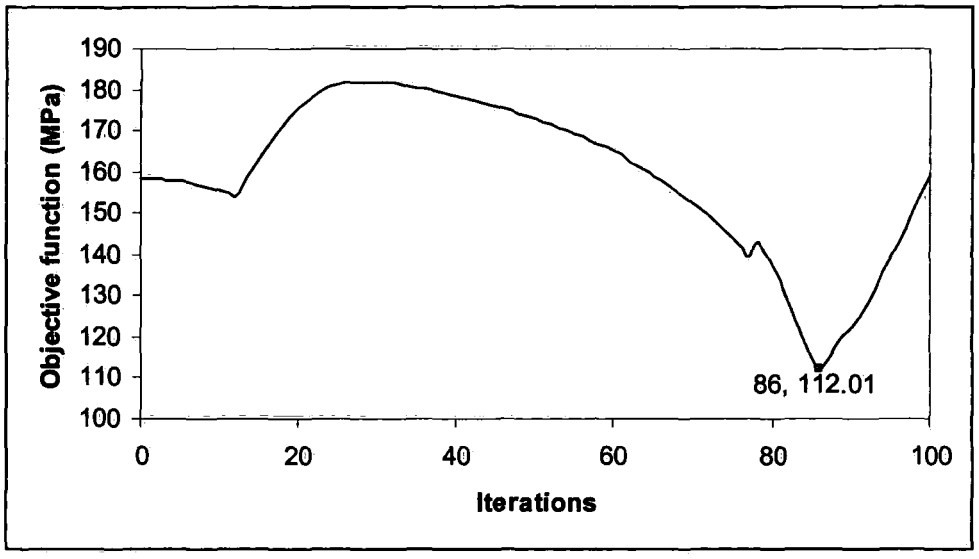


Figure 7.9: Evolution of the objective function

Figure 7.9 shows the evolution of the objective function which reaches M at 86 iterations.

We seek a set of parameters to give a good compromise between minimisation of both M and n, which will not happen simultaneously using the same set of parameters. We analyse the results first by displaying a scatter plot of the results of the 96 runs as shown in figure 7.10. Each point represents an optimisation process using a set of parameters, and the performance is depicted graphically by plotting a point showing the minimum objective function, M, and the number of iterations, n, required to reach this optimum. A Pareto curve is displayed on the figure using a dashed line showing the practical bounds on the optima that may be achieved. Five points are identified as good solutions that provide both a reasonable optimum and good computational efficiency, and are labelled using 1-5 in the figure. Table 7.4 shows the parameter values and performance relating to these five points. This suggests $RF = 0.5$ is important, with no strong conclusions to be formed about the other parameters.

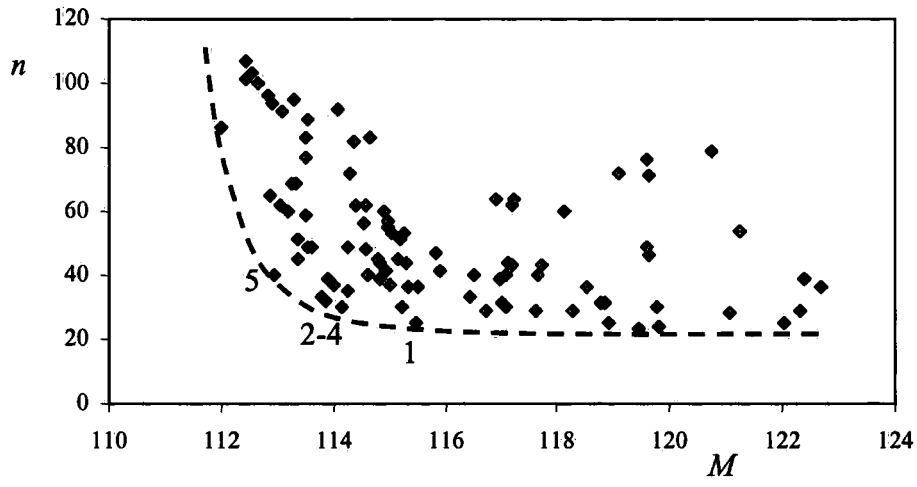


Figure 7.10: Scatter plot of results for fillet problem, showing Pareto curve

No.	*n*	*M*	*RR*	ER_R	*RF*
1	25	115.47	0.035	0.03	0.5
2	30	114.14	0.055	0.05	0.5
3	32	113.85	0.055	0.03	0.5
4	33	113.78	0.065	0.04	0.5
5	40	112.94	0.075	0.03	0.5

Table 7.4: Best parameters for fillet problem (Pareto analysis)

We can also apply a weighted sum method, in which we define a new objective function F_1 where,

$$F_1 = \alpha_1 M + \alpha_2 n \tag{7.4}$$

so that α_1 and α_2 are weighting factors that reflect the relative importance of the quality of the optimum (M) and the convergence rate (n). The best sets of parameters from the 96 parameter sets investigated found using different weights are shown in table 7.5.

α_1	α_2	F_1	n	M	RR	ER_R	RF
0	1	23	23	119.46	0.035	0.02	0.5
0.1	0.9	32.65	23	119.46	0.035	0.02	0.5
0.25	0.75	47.12	23	119.46	0.035	0.02	0.5
0.5	0.5	70.24	25	115.47	0.035	0.03	0.5
0.75	0.25	92.85	25	115.47	0.035	0.03	0.5
0.9	0.1	105.65	40	112.94	0.075	0.03	0.5
1	0	112.01	86	112.01	0.075	0.05	0.2

Table 7.5: Best parameters for fillet problem (weighted sum method)

It is interesting to note that, once again, the value of $RF = 0.5$ is a dominant conclusion. It is only in the case ($\alpha_1 = 1$, $\alpha_2 = 0$) that a different value ($RF = 0.2$) is suggested. This is not altogether surprising since $\alpha_2 = 0$ represents a situation in which an engineer is wholly concerned about the quality of the optimum, and is entirely unconcerned with the speed of attaining the optimum. In this case, a very small RF, i.e. removal of only a very small amount of material at each iteration, will prevail. Note also that the best RF value reverts quickly to 0.5 even for small $\alpha_2 = 0.1$.

The evolution towards optimum using $RF = 0.5$, $RR = 0.075$, $ER_R = 0.03$ is shown in figure 7.11 and the corresponding evolution of the objective function shown in Figure 7.12. This is the case in table 7.5 considering $\alpha_1 = 0.9$, $\alpha_2 = 0.1$.

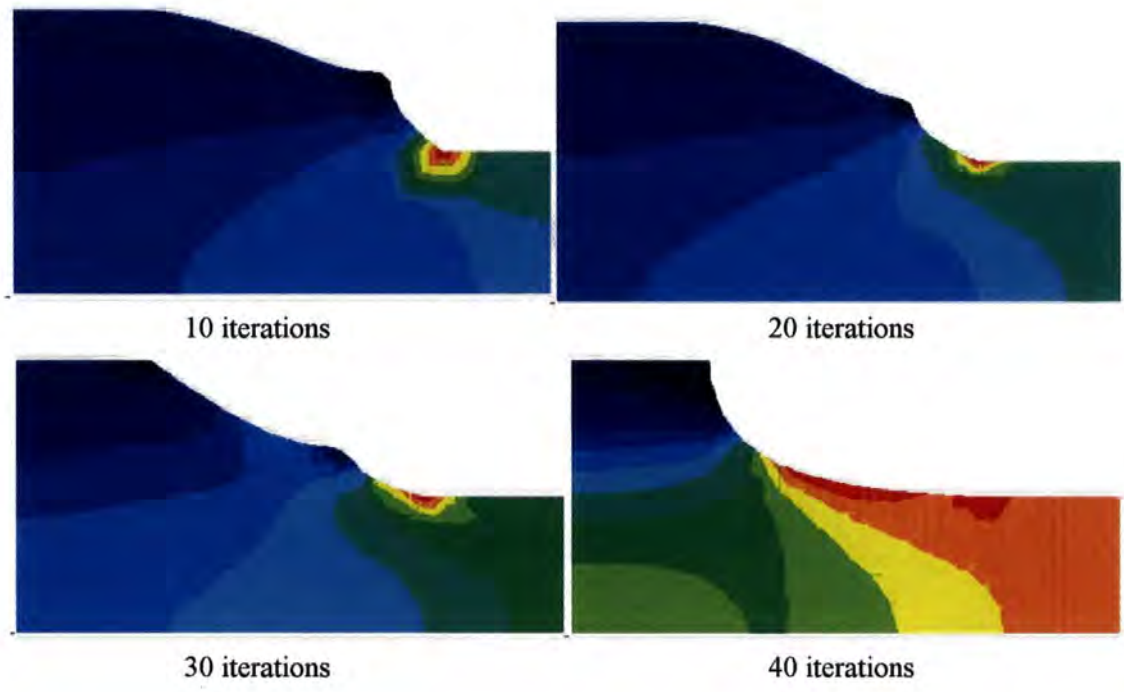


Figure 7.11: Evolution of the fillet example

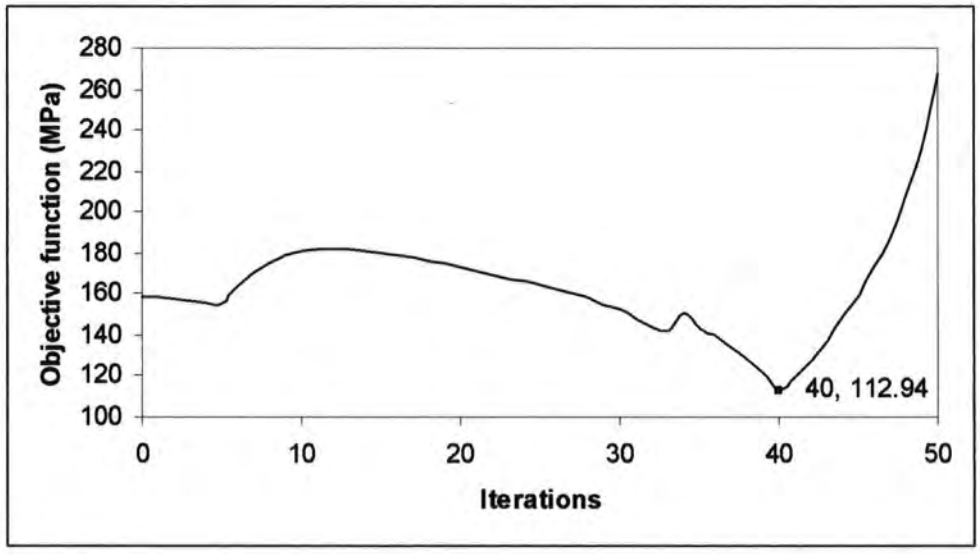


Figure 7.12: Evolution of objective function

It is interesting to note that a set of parameters that emerges, from table 7.6, as having good convergence properties to a good optimum, i.e. $RF = 0.5$, $RR = 0.035$, $ER_R = 0.03$, is the same set of parameters that were found from the short cantilever

beam example. This suggests that this set of parameters might be useful for a range of shape optimisation problems.

7.4 Conclusion

A series of tests have been performed to establish appropriate values for the guiding parameters governing shape optimisation problems using the boundary element method on a spline-based ESO algorithm. These parameters determine the extent of the boundary that is deformed at each iteration, and also the extent of the geometric change over those sections of boundary.

Two example models have been investigated using a wide range of numerical tests, and the performance of the optimisation has been evaluated using a variety of methods including mean performance analysis and multi-objective optimisation approaches using Pareto curves and weighted sums.

It is found that the parameter *RF* is the most important, i.e. the parameter determining the distance through which spline control points are moved. This is found to have an optimum value of *0.5*. The other parameters that define the extent of the boundary to be modified are of lesser importance, but recommended values are $RR = 0.05$ and $ER_R = 0.03$.

8. Parameter Selection for Sensitivity Analysis

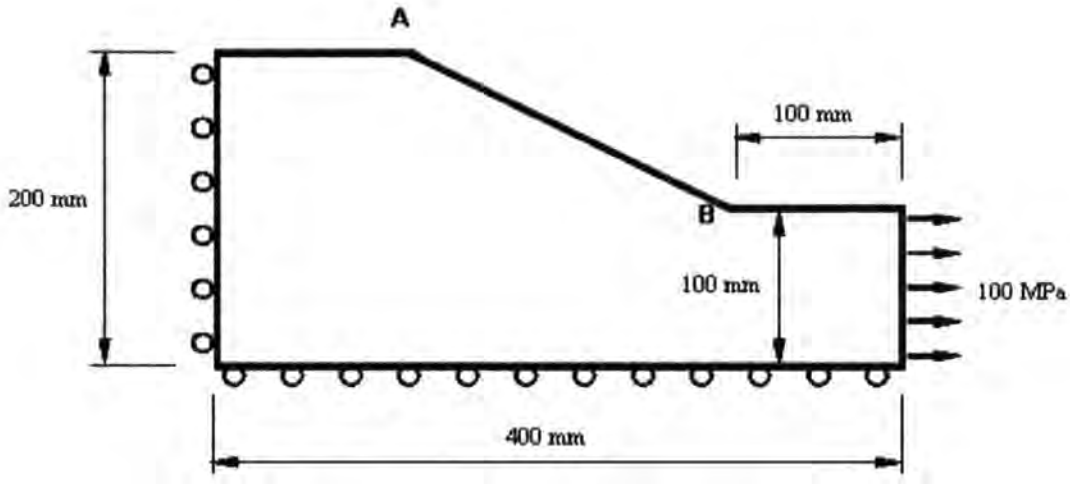
In chapter 6 has been presented the optimisation based on sensitivity analysis, which guides the design by providing the gradients of the objective functions. The finite difference approach is applied in which the control points are individually perturbed a small amount in each coordinate direction and an analysis performed to determine the associated change in the objective function at each iteration as shown in equation 6.3. This chapter gives numerical experiments to verify the efficiency of this approach. We choose the fillet as the model to investigate, the same model as in the chapter 7. The determination for stepsize d is presented in detail.

8.1 Fillet Problem

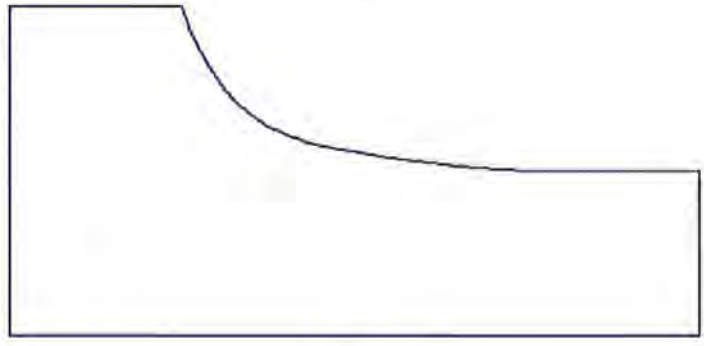
The fillet profile is shown in figure 8.1. The design domain is the changeable line, AB. This example model is similar to one that has been investigated using a wide range of numerical tests based on the ESO method. In those studies, the objective function was related to minimising the maximum principal stress or minimising the variation of maximum principal stress from its mean value on AB. In the current work, the maximum principal stress is still used as the objective function, but in this design optimisation process sensitivities provide the gradients of the maximum principal stress. The objective function f in equation 6.3 is defined as

$$f = max(\sigma_1) \tag{8.1}$$

and we require $M = \min(f)$ which provides a primary measure of the optimisation algorithm.



(a)



(b)

Figure 8.1: Fillet problem; (a) original domain, (b) form of optimum solution

Another variable in equation 6.4, stepsize d, should be predetermined. There are a number of considerations for determining d. The conclusion reached in chapter 7 is that the removal factor, RF, i.e. the parameter that governs the distance through which control points are moved, is the most important parameter. This can also be considered to be a factor affecting the stepsize d in a sensitivity based scheme. We denote

$$d = d_{sa}(i)\ RF \tag{8.2}$$

where $RF = 0.5$ as before, and $d_{sa}(i)$ is related to the model size and the iteration i. The initial value, d_{sa} (0), is defined by the ratio of the model dimension to a key

parameter c, e.g. for the model shown in figure 8.1 $cd_{sa}(0) = 400$mm. The selection for the value of c will be discussed later. In order that the evolution of the geometry tends to increase in stability, the stepsize is desired to decrease as the number of iterations increases. So we define

$$d_{sa}(\text{i+1}) = d_{sa}(\text{i})\, F_{sa}(\text{i}) \tag{8.3}$$

The value of the stepsize directly affects the change of the model volume, and the evolution history of the volume can reveal the extent of the change of the geometry. The term $F_{sa}(\text{i})$ is defined as followed

$$F_{sa}(i) = \frac{V_i}{V_0} \tag{8.4}$$

where V_i is the volume at the iteration i and V_0 is the initial volume.

To set the value of c, the range of the stepsize has to be considered. Cervera (2003) obtained the ratio range of stepsize (Δd) to the model dimension (h) as $10^{-5} \le \Delta d / h \le 10^{-2}$, by a series of tests for a general case using Concept Analyst, where $\Delta d = RF(d_{sa}(i) - d_{sa}(i+1))$. This leads us to suggest a value of c in the range ($10^2 \le c \le 10^5$). However, in order to accelerate the convergence to the optimum, we can investigate the effect of lower values of c. In the current work three models with different dimension h (400mm, 600mm and 800mm) are chosen as the object. The range of c, $10 \le c \le 80$ is selected for carrying out the test with sensitivity analysis. Every process stops if the stepsize reduces to approximately zero (<0.0005).

We separately run the three different dimension models repeatedly with different values of c. Table 8.1 presents the dimensions of the example models, c, the initial stepsize $d_{sa}(0)$, the minimum of the objective function, M, and the number of iterations required to achieve the minimum, n. A scatter plot is displayed in figure 8.2 showing M and n. Three kinds of spots are used to distinguish different dimension models.

h (mm)	c	$d_{sa}(0)$ (mm)	M(Nmm^4)	n
400	50	8	117.4893	157
	40	10	116.3375	101
	25	16	114.7256	102
	20	20	113.5133	102
	16	25	114.8869	102
	13.33333	30	112.4154	101
	11.42857	35	112.1416	91
	10	40	110.7306	85
600	50	12	109.946	102
	40	15	114.3205	178
	30	20	113.1647	179
	25	24	111.0253	97
	20	30	110.4599	40
	15	40	109.8822	46
	10	60	109.3947	102
800	50	16	111.9853	225
	40	20	111.7787	30
	25	32	110.2489	38
	20	40	111.4217	42
	16	50	108.9301	16
	13.33333	60	109.5317	76
	11.42857	70	109.5423	76
	10.66667	75	107.1972	49
	10	80	107.2199	72

Table 8.1: Results for fillet problem by sensitivity analysis

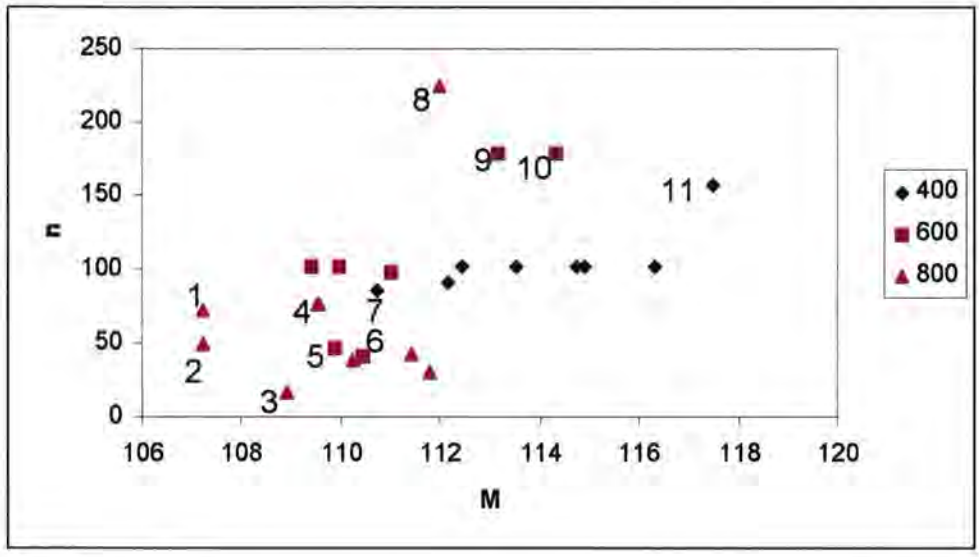


Figure 8.2: Scatter plot of results for fillet problem using sensitivity analysis

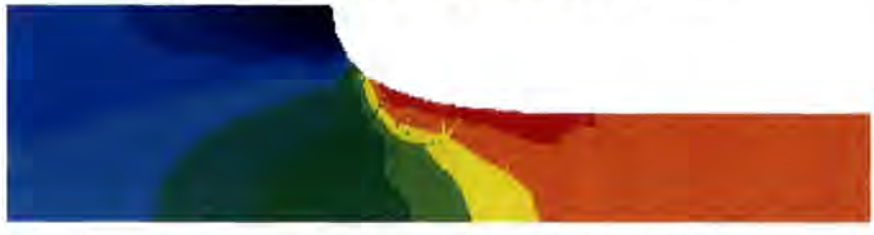
It appears from the figure that perhaps not enough runs have been performed for a clear Pareto curve to emerge. However, points labelled 1 to 7 can clearly be seen to outperform those labelled 8, 9, 10 and 11. Points for different h tend to cluster in the figure, and this is not surprising since the different initial geometry will lead to slightly different optima in the three cases. With this information, we can learn about suitable parameters in the algorithm.

8.2 Parameter Selection

From the data we obtain above, the initial stepsize $d_{sa}(0)$ seems to be a key parameter affecting the optimum. The points labelled 8, 9, 10 and 11 indicate that the algorithm converges to an optimum after a large number of iterations because of the small initial stepsize. In optimisation algorithms, it is normally the case that a smaller stepsize should be expected to yield an improved optimum having lower M. Viewing the data in Table 8.1 shows that the reverse is found. It is believed that the cause of this behaviour is that the stepsize decreases with the increasing the number of iterations, which leads to the process stopping before the models achieve the

optimum. This highlights a flaw in the algorithm, and further work is required in determining a suitable reduction in stepsize with the number of iterations to give more reliable convergence. However, for the algorithm presented in chapter 6, because $d_{sa}(0)$ is defined by the ratio of the model dimension to c, we consider $c \approx$ 10-20 is the best range as shown around the points labelled 4, 5, 6 and 7. However, it is interesting to note that for $h = 800mm$ model the points labelled 1 and 2 reach an improved optimum having a different shape to the optima for most runs (figure 8.3).

(a) $d_{sa}(0)$ = 80 for the point 1 in figure 8.2



(b) $d_{sa}(0)$ = 70 for the point 4 in figure 8.2

Figure 8.3: Optimum for the h = 800mm model

Although they both converge to an optimum, different minima of the objective function are obtained. It is proposed that this is a problem having multiple local optima, and exhibiting an instability for certain values of stepsize that allows the algorithm to jump to a different optimum. Here we demonstrate this process graphically figure 8.4. The aim is to minimise f(x) using a simple gradient approach. A scheme starting with a larger initial stepsize, directly jumps to A and reaches the global optimum. On the other hand, a smaller stepsize is destined to arrive at B, only to achieve the local optimum. In the current work, we consider for the h = 800mm model $c \geq 12$ is the ideal range to get a good optimum.

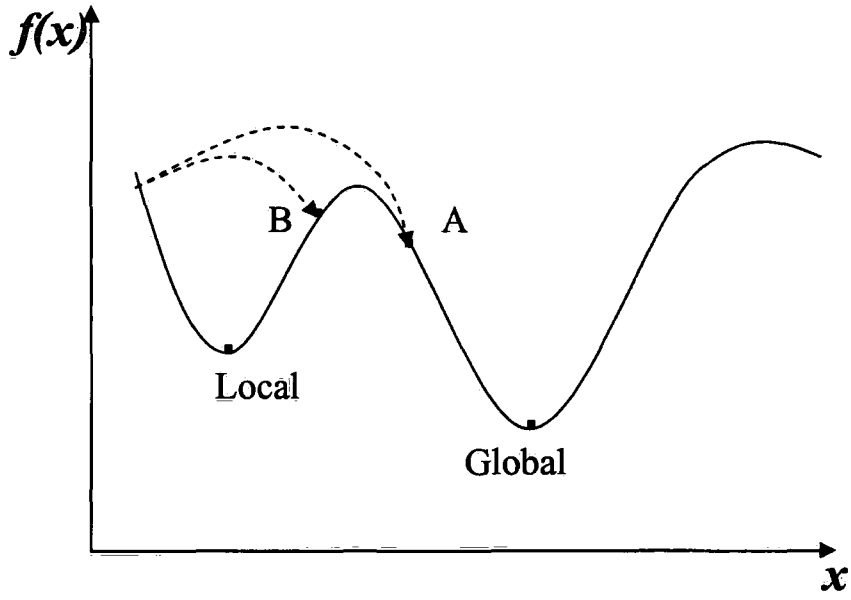
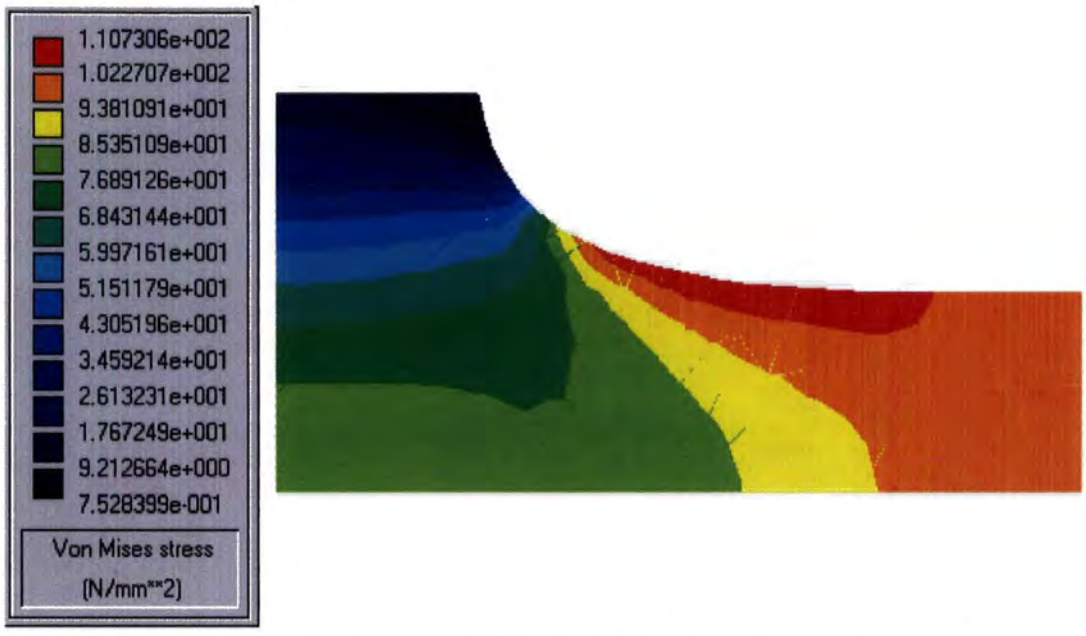


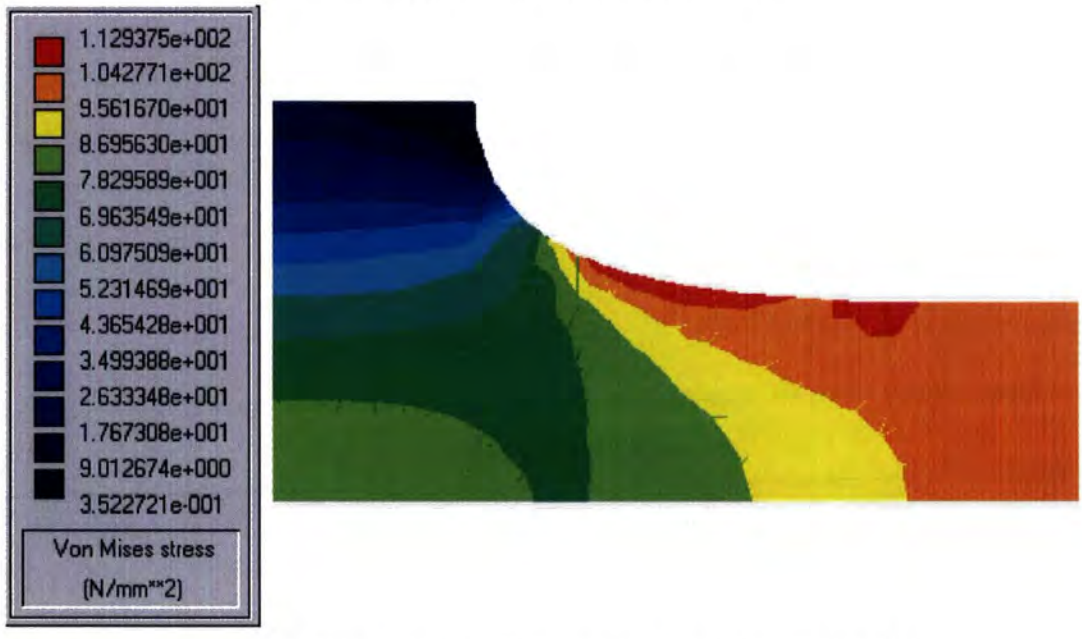
Figure 8.4: Local and global optimisation

8.3 Comparison with ESO

It is necessary to compare the sensitivity analysis method in structural optimisation with ESO, the main method in this work. In order to provide a clear comparison, we use the same dimension fillet model as chapter 7. In the figure 8.5(a), $M = 110.73$MPa is achieved at 85 iterations by sensitivity analysis, while $M = 112.94$MPa at 40 iterations by ESO. In sensitivity analysis the optimisation process is directly guided by the objective function. Although longer run-time is needed, it undoubtedly is a good choice in practice when a high level of solution accuracy is required. In other words, if the objective function is one other than a fully stressed design or the equivalent specific stiffness or strain energy objective functions, ESO should be selected if the run-time is the first factor.



(a) Von Mises stress contour plot at 85 iterations by sensitivity analysis
(the width of the model is 400mm)



(b) Von Mises stress contour plot at 40 iterations by ESO
(the width of the model is 400mm)

Figure 8.5: Comparison between sensitivity analysis and ESO

8.4 Conclusion

Sensitivity analysis has been applied in the design optimisation process based on the boundary element method in this chapter. The finite difference approach is applied to calculate the effect of changes in the design variables on the objective function. The stability of this approach has been displayed by a series of tests. From the small number of runs that have been performed, we estimate c is a key parameter which determines the initial stepsize to guide the optimisation process. It is found that a good range of c is about [10, 20] for the problem considered. Moreover, the design domain in this algorithm is described by a B-spline which is influenced only by the local control points. So it should be efficient to use reanalysis together with sensitivity analysis.

The current work also compares the sensitivity analysis method in structural optimisation with ESO. Obviously, although longer run-time is a drawback for sensitivity analysis, it has the advantage in that it is guided by objective function and consequently produces an optimum that is better than ESO. In this chapter, a minimisation of maximum principal stress is the only objective function to be used. An extension to multi-criteria optimisation has been made by Steven *et al.* (2002) in a finite element based ESO scheme. The promising results presented in this section suggest that a similar success would be likely in the BEM based ESO scheme.

9. Conclusions and Recommendation for Future Research

9.1 Conclusions

In this work the evolutionary structural optimisation algorithm (Cervera (2003) and Cervera & Trevelyan (2005a, 2005b)) based on Boundary Element Method has been used in 2D shape optimisation. An improved understanding has been developed of the role of some governing parameters driving the optimisation using this algorithm. As a result of a series of numerical tests, recommendations have been made as to appropriate values of these parameters by which good optimal solutions may be achieved efficiently.

Two problems of different character are selected as test cases in this work. One is a short cantilever beam with whole geometric change during the ESO process; the other is a fillet with a small geometric change. They have been investigated using a wide range of numerical tests, and the performance of the optimisation has been evaluated using a variety of methods including mean performance analysis and multi-objective optimisation approaches using Pareto curves and weighted sums. It is found that the parameter RF is the most important, i.e. the parameter determining the distance through which spline control points are moved at each iteration. This is found to have an optimum value of 0.5. The other parameters that define the extent of the boundary to be modified are of lesser importance, but recommended values are $RR = 0.05$ and $ER_R = 0.03$. Precise definitions of these parameters may be found in section 5.3. Since these parameter values have been found appropriate for these two problems of very different character, it is hoped that they may be used reliably for a wide class of structural shape optimisation problems. Further work is required to validate the use of these parameters on a wider group of optimisation problems.

A small number of investigations into a boundary element design optimisation process based on sensitivity analysis have been carried out in this work. By a series of tests the stability of this method has been obtained by applying the finite

difference approach to calculate the effect of small changes in the design variables (in this algorithm, the spline control points) on the objective function. A parameter *c* which is related to the dimension of the model is considered a key one to guide the optimisation process. The approximate range is 10-20 for the fillet problem considered. Like *RF* in the conventional ESO, c provides a measure of the amount of geometric change at each iteration. Difficulty was experienced in obtaining an appropriate set of parameters to provide for progressive reduction in the amount of geometric change per iteration as the optimisation converges to a solution, i.e. the fine tuning. In some cases, the reduction was performed too rapidly so that the true optimum was never achieved. It was also noted that the fillet problem converged to two different 'optimal' solutions, depending on the governing parameters adopted. The classical optimum shape (figure 7.1) is obtained with most parameters used, but an undercut fillet (figure 7.4) is obtained when a large initial step size is used. The undercut fillet provides a better optimum for the objective function chosen. The convergence towards a local optimum, and missing a global optimum, is a well recognised drawback of gradient approaches.

The current work also compares the sensitivity analysis method in structural optimisation with ESO. In sensitivity analysis the optimisation process is directly guided by the objective function. Although a longer run-time is needed, it is undoubtedly a good choice for requiring high level solution in practice. However, the advantage of the ESO is that the optimum can be reached in fewer iterations and considerably lower run-time, which is why it is selected when computational efficiency is thought the most important consideration. Moreover, for certain objective functions, the ESO scheme has been shown to provide the optimum solution.

9.2 Recommendation for Future Research

The initial aims of this work have been accomplished. However, some questions found during the work should be considered for future research.

Future research might expand the application scope of this work which is limited to plane stress problems. In addition to confirming the validity of the recommended parameter values for a wider set of problems, different objective functions might be used, such as displacement, natural frequency, buckling load and thermo-elastic; also extend to the multi-criteria and multidisciplinary structural optimisation.

The design domain in this algorithm is described by a B-spline which is influenced only by the local control points. So it should be efficient to use reanalysis together with sensitivity analysis. This may be expected to reduce the run-time of the sensitivity based scheme significantly. More research is also required in determining a scheme for the fine tuning of the process so that the amount of geometric change per iteration reduces as the optimisation progresses. A scheme has been presented in this work, but it is recognised that it needs further refinement to ensure that the scheme does not slow down to a stop before the optimum is reached.

In this work, 3D problems were not investigated at all. There would be benefit in extending to structural optimisation for 3D problems. B-spline surfaces can be used for representation of the geometry.

Bibliography

1. Abramowitz, M., Stegun, I.A. (Eds.). (1972). *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, 9th printing. New York: Dover; 885-886.
2. Abu Kassim, A.M.,Topping, B.H.V. (1987). Static re-analysis: a review. *Journal of structural Engineering, ASCE*; 113(5): 1029-1045.
3. Arora, J.S. (1976). Survey of structural reanalysis techniques. *Journal of Structural Division, ASCE*; 102(ST4): 783-802.
4. Baumgartner, A., Harzheim, L., Mattheck, C. (1992). SKO (soft kill option): the biological way to find an optimum structure topology. *International Journal of Fatigue*; 14(6): 387-393.
5. Becker, A.A. (1992). *The Boundary Element Method in Engineering*, McGraw-Hill.
6. Beer, G., Watson, J.O. (1992). *Introduction to Finite and Boundary Element Methods for Engineers*, Wiley, New York.
7. Bendsøe, M.P., Kikuchi, N. (1988). Generating optimal topologies in structural design using a homogenisation method. *Computer Methods in Applied Mechanics & Engineering*; 71: 197-224.
8. Brebbia, C.A. (1978). *The Boundary Element Method for Engineering*, Pentech Press London.
9. Brebbia, C.A., Dominguez, J. (1989). *Boundary Elements: An introductory Course, Computational Mechanics Publications and McGraw-Hill*, First Edition 1989, Second Edition 1992.
10. Cervera, E. (2003). *ESO Based on Boundary Element Representation of B-spline Geometry*, PhD Thesis, Durham University,UK.
11. Cervera, E, Trevelyan, J. (2002). Evolutionary structural optimization using B-spline representation. *Proceedings of the Fifth World Congress on Computational Mechanics (WCCMV). Vienna, Austria.*
12. Cervera, E., Trevelyan, J. (2005a). Evolutionary structural optimisation based on boundary representation of NURBS. Part I: 2D algorithms. *Computers & Structures*; 83: 1902-1916.

13. Cervera, E., Trevelyan, J. (2005b). Evolutionary structural optimisation based on boundary representation of NURBS. Part II: 3D algorithms. *Computers & Structures*; 83: 1917-1929.
14. Cormen, H.T., Leiserson, E.C., Rivest, L.R, Stein, C. (2001). *Introduction to Algorithms*, Second Edition. MIT Press and McGraw-Hill. Chapter 29: 770-821.
15. Cruse, T.A. (1968). A direct formulation and numerical solution of the general transient elastodynamic problem. II. *Journal of Mathematical Analysis and Applications*; 22: 341-355.
16. Cruse, T.A., Rizzo, F.J. (1968). A direct formulation and numerical solution of the general transient elastodynamic problem. I. *Journal of Mathematical Analysis and Applications*; 22:244-259.
17. Davis, P.J., Rabinowitz, P. (1984). *Methods of Numerical Integration* 2^{nd}, Academic Press.
18. Dantzig, B.G. (1951). Maximization of a linear function of variables subject to linear inequalities, In: T. C. Koopmans, ed. *Activity Analysis of Production and Allocation*, John Wiley, New York, 339-347.
19. Eschenauer, H.A., Kobelev, V.V, Schumacher, A. (1994). Bubble method for topology and shape optimisation of structures. *Structural optimization*; 8: 42-51.
20. Fredholm, I. (1903). ` Sur une class d'equations fonctionelles, *Acta Math.* 27: 365-390.
21. Fleury, C. (1979). Structural weight optimization by dual methods of convex programming. *International Journal for Numerical Methods in Engineering*; 14:1761-1783.
22. Hess, J.L., Smith, A.M.O. (1967). Calculation of potential flow around arbitrary bodies. D. Kuchemann(ed), *Progress in Aeronautical Sciences*; Vol.8, Pergamon.
23. Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*, Kluwer Academic Publishers, Boston, MA.
24. Gradshteyn, I.S., Ryzhik, I.M. (2000)."Jacobian Determinant." §14.313, In: *Tables of Integrals, Series, and Products*, 6th ed. San Diego, CA: Academic Press, 1068-1069.
25. Jaswon, M.A. (1963). Integral equation methods in potential theory I. *Proceeding of the Royal Society of London Series A*; 275: 23-32.

26. Kane, J.H., Keshava Kumar, B.L., Gallagher, R.H. (1990). Boundary-element iterative re-analysis for continuum structures. *Journal of Structural Mechanics ASCE*; 116(10): 2293-2309.
27. Kane, J.H. (1994). *Boundary Element Analysis in Engineering Continuum Mechanics*. Prentice-Hall.
28. Lachat, J.C, Watson, J.O. (1976). Effective numerical treatment of boundary integral equations. *International Journal for Numerical Methods in Engineering*; 10: 991-1005.
29. Leu, L-J. (1999). Shape Optimization by the Boundary Element Method with a Reduced Basis Reanalysis Technique. *Structural Engineering and Mechanics*; 8(1): 73-84.
30. Mackie, R.I. (1998). An object-oriented approach to fully interactive finite element software. *Advanced in Engineering Software*; 29(2): 139-149.
31. Mattheck, C, Burkhardt, S. (1990). A new method of structural shape optimization based on biological growth. *International Journal of Fatigue*; 12(3):185-190.
32. Meric, R.A. (1995). Differential and integral sensitivity formulations and shape optimization by BEM. *Engineering Analysis with Boundary Element*; 15: 181-188.
33. Michell, A.G.M. (1904). The limits of economy of material in frame-structures, *Philosophical Magazine*; 8: 589-597.
34. Mikhlin, S.G. (1957). *Integral Equations*. Pergamon Press. Now York.
35. Muskhelishvili, N.I. (1953). *Some Basic Problems of the Mathematical Theory of Elasticity*. Noordhoff, Holland.
36. Parvizian, J, Fenner, R.T. (1997). Shape optimization by the boundary element method: a comparison between mathematical programming and normal movement approaches, *Engineering Analysis with Boundary Element*; 19: 137-145.
37. Piegl, L., Wayne, T. (1997). *The NURBS Book*. Springer, Berlin.
38. Querin, O.M., Steven, G.P., Xie, Y.M. (2000). Evolutionary structural optimisation using an additive algorithm. *Finite Elements in Analysis and Design*; 34: 291-308.

39. Querin, O.M., Xie, Y.M., Steven, G.P. (1998). Evolutionary structural optimization (ESO) using a bidirectional algorithm. *Engineering Computations*; 15:1031-1048.

40. Reynolds, D., McConnachie, J., Bettess, P., Christie, W.C., Bull, J.W. (1999). Reverse Adaptivity-Anew evolutionary tool for structural optimisation. *International Journal for Numerical Methods in Engineering*; 45: 529-552.

41. Rizzo, F.J. (1967). An integral equation approach to boundary value problems of classical elastostatics. *Quarterly of Applied Mathematics*; 25: 83-95.

42. Rogers, D.F. (2001). *An Introduction to NURBS: with Historical Perspective.* Morgan Kaufmann publishers. Academic Press.

43. Rozvany, G.I.N., Pomezanski, V., Querin, O.M., Gaspar, Z., Logo, J. (2004). Corner contact suppression in topology optimization. In: *Proceedings of 5th ASMO UK/ISSMO conference on engineering design optimization. Stratford-upon-Avon, July 12-13, 2004.*

44. Schnack, E., Spörl, U. (1986). A mechanical dynamic programming algorithm for structure optimization. *International Journal for Numerical Methods in Engineering*; 23:1985-2004.

45. Saad, Y., Schultz, M.H. (1986). GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*; 7(3): 856-869.

46. Steven, G.P, Qing, L., Xie, Y.M. (2002). The role of physical sensitivity in evolutionary topology design optimization with multi-criteria and multi-physics. In: *Proceedings of the Fifth World Congress on Computational Mechanics (WCCMV). Vienna, Austria.*

47. Symm, G.T. (1963). Integral equation methods in potential theory II. *Proceeding of the Royal Society of London Series A*; 275: 33-46.

48. Tafreshi, A., Fenner, R.T. (1991). Design optimization using the boundary element method. *Journal of Strain Analysis*; 26(4):231-240.

49. Trevelyan, J. (1994). *Boundary Elements for Engineers.* Computational Mechanics Publications.

50. Trevelyan, J., Scales, D., Morris, R. and Bird, G. (2004). Acceleration of boundary element computations in reanalysis of problems in elasticity. In: *Computational Mechanics WCCM VI in conjunction with APCOM'04, Sept. 5-10.*

51. Trevelyan, J., Wang, P. (2001). Interactive re-analysis in mechanical design evolution. Part I. Background and implementation. *Computers and Structures*; 79: 929-938.

52. Venkayya, V.B., Knot, N.S., Berke, L. (1973). Application of Optimality Criteria Approaches to Aautomated Design of Large Practical Structures. In: *Proceedings of 2nd Symposium on Structural Optimization. Milan, Italy, AGARD-CP-123.*

53. Wang, M.Y., Wang, X., Guo, D. (2003). A level set method for structural topology optimization. *Computer Methods in Applied Mechanics & Engineering*; 192: 227-246.

54. Wang, S.Y., Tai, K. (2005). Structural topology design optimization using Genetic Algorithm with a bit-array representation. *Computer methods in applied mechanics and engineering*; 194: 3749-3770.

55. Wen, J., Trevelyan, J. (2005). Tuning of parameters guiding B-spline based ESO optimisaton with boundary elements. In: *Proceedings of 5th UK Conference on Boundary Integral Methods, Liverpool.*

56. Xie, Y.M., Steven, G.P. (1993). A simple evolutionary procedure for structural optimization. *Computers & Structures*; 49(5): 885-896.

57. Xie, Y.M., Steven, G.P. (1997). *Evolutionary Structural Optimization (ESO).* Springer.

58. Yamazaki, K., et al. (1993). An efficient shape optimization technique of a two-dimensional body based on the boundary element method. *Computers and Structures*; 48(6): 1073-1081.

59. Yamazaki, K., et al. (1994). Three-dimensional shape optimization using the boundary element method. *AIAA Journal*; 32(6): 1295-1301.

60. Zhou, M., Rozvany, G.I.N. (1996). An improved approximation technique for the DCOC method of sizing optimization. *Computers & Structures*; 60(5): 763-769.