Durham E-Theses

Evolutionary structural pptimisation based on boundary element representation of b-spline geometry

Cervera, Eva

How to cite:

Cervera, Eva (2003) *Evolutionary structural pptimisation based on boundary element representation of b-spline geometry*, Durham theses, Durham University. Available at Durham E-Theses Online: http://etheses.dur.ac.uk/2004/

Use policy

Evolutionary Structural Optimisation Based on Boundary Element Representation of B-spline Geometry

by

Eva Cervera

A thesis submitted for the degree of

Doctor of Philosophy

School of Engineering

The University of Durham

2003

2 8 APR 2004

Evolutionary Structural Optimisation Based on Boundary Element Representation of B-spline Geometry

by

Eva Cervera

A thesis submitted for the degree of Doctor of Philosophy

School of Engineering, The University of Durham, 2003

Abstract

Evolutionary Structural Optimisation (ESO) has become a well-established technique for determining the optimum shape and topology of a structure given a set of loads and constraints. The basic ESO concept that the optimum topology design evolves by slow removal and addition of material has matured over the last ten years. Nevertheless, the development of the method has almost exclusively considered finite elements (FE) as the approach for providing stress solutions.

This thesis presents an ESO approach based on the boundary element method. Non-uniform rational B-splines (NURBS) are used to define the geometry of the component and, since the shape of these splines is governed by a set of control points, use can be made of the locations of these control points as design variables. The developed algorithm creates internal cavities to accomplish topology changes. Cavities are also described by NURBS and so they have similar behaviour to the outside boundary. Therefore, both outside and inside are optimised at the same time. The optimum topologies evolve allowing cavities to merge between each other and to their closest outer boundary. Two-dimensional structural optimisation is investigated in detail exploring multi-load case and multi-criteria optimisation. The algorithm is also extended to three-dimensional optimisation, in which promising preliminary results are obtained.

It is shown that this approach overcomes some of the drawbacks inherent in traditional FE-based approaches, and naturally provides accurate stress solutions on smooth boundary representations at each iteration.

Declaration

The material contained within this thesis has not previously submitted for a degree at the University of Durham or any other University. The research reported within this thesis has been conducted by the author unless otherwise indicated.

Chance has conducted our affairs even better than we could either wish or hope for; look there, friend Sancho, and behold thirty or forty outrageous giants, with whom I intend to engage in battle, and put every soul of them to death, so that we may begin to enrich ourselves with their spoils; for it is a meritorious warfare, and serviceable both to God and man, to extirpate such a wicked race from the face of the earth. What giants do you mean?-said Sancho Panza in amazement...

Miguel de Cervantes
The Adventures of Don Quijote de la Mancha

Acknowledgements

Table of Contents

List of Figures

List of Tables

Nomenclature

a	distance
AF	addition factor
AR	addition ratio
b	distance, body force
c	distance
$C(u)$	parametric curve
C_{ij}	free term
d	distance
d_{min}	distance
e_T	truncation error
e_C	condition error
E	Young's modulus
ER_A	evolutionary rate for addition
ER_R	evolutionary rate for removal
f	objective function
f_k	stress concentration factor objective function
f_K	specific stiffness objective function
f_V	volume minimisation objective function
f_U	specific strain energy objective function
f_w	weight reduction objective function

f_Γ	stress levelling objective function
h	height, dimension
k	constant parameter
$k_{removal}$	constant parameter
$k_{addition}$	constant parameter
K_t	stress concentration factor
l	constant parameter
L_e	element length
LC	load case
L^S	serendipity shape function
L	shape function
m	number, constant parameter
n	number, normal vector
N	basis function, shape function (boundary element)
p	degree of B-spline, boundary element node
P	control point
q	degree of B-spline
r	radial coordinate
R^i_j	sensitivity ratio for control point i and criterion j
R_i	weighted sensitivity ratio for control point i
RF	removal factor

RR	removal ratio
S(u, v)	bi-parametric surface
S	surface
t	thickness
t	traction
T	traction vector, surface
u	independent variable, knot value, displacement
u_p	tangential vector at p
U	strain energy or compliance, knot vector
U_0	initial strain energy or compliance
v	independent variable
v_p	tangential vector at p
V	volume, knot vector
V_0	initial volume
w	weight
x	Cartesian coordinate, design variable
y	Cartesian coordinate
z	Cartesian coordinate
Γ	boundary
ν	Poisson's ratio
ρ	density

σ	stress
σ_{VM}	von Mises stress
σ_p	node stress of selected criterion
σ_{IP}	internal point stress of selected criterion
σ_y	Yield stress
$\sigma_1, \sigma_2, \sigma_3$	principal stresses
σ_{ref}	reference stress of selected criterion
σ_n	nominal stress of selected criterion
σ_{max}	maximum stress of selected criterion
η	local coordinate
ξ	local coordinate
θ	angle (polar coordinate)
ε	stopping criterion
Δs	step size
α_j^i	sensitivity number for control point i and criterion j
ψ	maximum dimension factor
β	stress ratio
γ	corner effect factor

1

INTRODUCTION

1.1 Overview

This thesis is concerned with the development of a computational algorithm to be used as a tool in structural optimisation. Firstly in this introductory chapter, a brief background to the theoretical features related to the algorithm is presented. The theoretical background, to be extended in the forthcoming chapters, allows introducing this research work into structural optimisation context. Later in the chapter, the aims of this thesis are stated. Finally, the layout of the thesis is presented.

1.2 Background

Structural optimisation is present in most engineering problems since changes in the design would normally imply changes in the performance of the structure. Until relatively recently, any design process was driven exclusively by the designer's experience. With the development of computers in the 1960s, complex optimisation techniques were implemented in the design process, allowing structural optimisation to be applied to problems requiring high level of structural performance, for example in the aerospace or automotive industries where material savings are crucial.

Moving from an engineering perspective to a living environment point of view some similarities are found between engineering structures and biological structures. In nature, organisms adapt to their environment in different ways. As well as species evolving through genetics and natural selection according to Darwin's Origin of Species, individual organisms react to their particular surroundings. Attempting to find a parallel in engineering to one of these phenomena, Mattheck *et al.* (1990) introduced the *biological growth method*. This method is based on the axiom of *constant stresses* and is derived from analogies observed in the growth of biological structures such as trees, bones and horns. This idea is similar to the gradientless shape optimisation presented by Schnack and Spörl (1986). Also in this context, *hard kill* (HK) methods (Hinton and Sienz (1995)) are based on finite element (FE) models and the solution is achieved iteratively by removing low stressed elements from the mesh. Starting from a similar concept, the *evolutionary structural optimisation* (ESO) approach proposed by Xie and Steven (1993) has become a well-established technique for determining the optimum shape and topology of a structure given a set of loads and constraints. The basic ESO concept, i.e. that the optimum topology design evolves by slow removal and addition of material, has matured over the last ten years. Element sensitivity numbers (Steven *et al.* (2002)) are calculated to consider different physical situations and various optimisation criteria. Nevertheless, the development of the method has almost exclusively considered finite elements (FE) as the approach for providing stress solutions.

1.3 Aims

The aim of this work is to present an alternative approach to ESO by using the boundary element method (BEM) to carry out the structural analysis. The geometry of the component is defined by non-uniform rational B-splines (NURBS) and, since the shape of these splines is governed by a set of control points, use can be made of the locations of these control points as design variables. The objective is to overcome some of the drawbacks inherent in traditional FE-based approaches, and naturally provide accurate stress solutions on smooth boundary representations. The algorithm is implemented into a computer program using Visual C++.

In two-dimensional problems, topology changes are performed by creating internal cavities which are also described by NURBS. Since the internal boundary exhibits a similar behaviour to the outside boundary both boundaries are optimised at the same time. The possibility that small cavities can merge to form bigger cavities is explored as a natural evolution of the topology of the structure. Also in the 2D context, multi-load case and multi-criteria optimisation are investigated. Finally, the basic features of the algorithm for 2D optimisation problems are extended to consider 3D structural optimisation problems.

1.4 Outline of the Thesis

The contents of this thesis have been structured into a total of ten chapters. Chapter 1 gives a brief background on structural optimisation and explains how this research work is relevant to the particular ESO field. The aims of this thesis are presented.

Chapters 2, 3 and 4 present a more detailed theoretical review of the main features investigated in this thesis. Thus, chapter 2 presents a review on structural optimisation methods. In chapter 3 the nonuniform rational B-splines (NURBS), used in the boundary representation, are detailed. Chapter 4 gives a comprehensive explanation of boundary element methods.

Chapters 5, 6, 7 and 8 show the development of the various algorithms and the results obtained. In chapter 5 the gradientless algorithm is developed for 2D shape optimisation problems. In chapter 6 the algorithm is applied to 2D topology optimisation problems, giving a detailed explanation of the creation and manipulation of cavities. Also in this chapter multi-load case problems are considered. In chapter 7 multi-criteria problems are investigated by using design sensitivity numbers and reanalysis techniques to speed up the process. In chapter 8 the gradientless algorithm is extended to 3D structural optimisation problems.

A discussion of the results is presented in chapter 9 and, finally, a summary of the thesis is given in chapter 10 showing the conclusions and some implications for future research.

2

STRUCTURAL OPTIMISATION
REVIEW

2.1 Introduction

Wasiutyński and Brandt (1963), amongst other authors, suggested that the beginnings of structural optimisation are dated in the 17th century. According to them, Galileo was the first who applied the notion of force and the laws of mechanics as a foundation for the design of structures. Methods of design in Galileo's time consisted of trials and experiments, and evolved then to methods of verification of strength such as admissible stresses or strains and the method of ultimate load. Perhaps the first analytical work in structural optimisation was by Michell (1904), on the basis of previous work by Maxwell (1869). Michell developed a design theory for the topology of frame structures that are optimal with respect to weight.

The classical theories and conventional view of optimization are presented by Wilde and Beightler (1976). According to their definition, optimisation deals with achieving *the best* in a *rational manner*, the term *the best* referring to maximum gain or minimum lost of an *objective function* for a specific environment. The expression *rational manner* can be interpreted as the selection of *design variables* within the limits or *constraints* placed on the structural behaviour, geometry or other factors.

Arora (1989) also refers to the concept of *design the best system* and by best means an efficient, versatile and cost-effective system.

Three basic features form the design problem - *design variables, objective function and constraints*. As *design variables* are considered the size of the structural components, cross-sectional dimensions, parameters that describe the geometry or the mechanical and physical properties of the material. The *objective function* $f(x)$ or objective functions $f(x) = [f_1(x), f_2(x), \ldots, f_n(x)]$ (*multi-criteria* optimisation) are used as a measure of the effectiveness of the design. Finally, the *constraints* are restrictions to be satisfied in order to make the design suitable. They can be *inequality constraints* which are the response limits imposed on the design. For example the maximum stress or limits on deflection. On the other hand, *equality constraints* are precise requirements to be accomplished for the design to be acceptable, such as equilibrium constraints.

2.2 Mathematical Definition of Optima

The mathematical formulation of the optimisation problem can be summarised in the following general expression where, for a set x of design variables, the optimum is achieved by minimising or maximising the objective function $f(x)$

$$
\begin{aligned}
&\text{min/max} \quad f(x) \\
&\text{subject to} \begin{cases} g_j(x) \leq 0 & j = 1,\ldots,l \\ g_k(x) = 0 & k = l+1,\ldots,m \\ x_i^L \leq x_i \leq x_i^U & i = 1,\ldots,n \end{cases}
\end{aligned}
\tag{2.1}
$$

where x denotes a vector of n design variables $x = [x_1, x_2, \ldots, x_n]$. The equality constraints are represented by $g_k(x)$ and the inequality constraints by $g_j(x)$. The lower and upper bounds, x^L and x^U respectively, are referred as the *side constraints* and they directly limit the region of search for the optimum.

5

The concept of *Lagrange multipliers* λ allows the incorporation of the constraints into an extended objective function L(x, λ), also know as the augmented *Lagrangian function*

$$L(x,\lambda) = f(x) + \sum_{j=1}^{m} \lambda_j g_j(x) \qquad (2.2)$$

A very important concept in constrained optimisation is use of *Kuhn-Tucker* conditions. The *Kuhn-Tucker* necessary conditions state that a solution set x^* is an optimum if there exist unique *Lagrange multipliers* λ^* such that

$$\frac{\partial L(x^*,\lambda^*)}{\partial x_i} = \frac{\partial f(x^*)}{\partial x_i} + \sum_{j=1}^{m} \lambda_j^* \frac{\partial g_j(x^*)}{\partial x_i} = 0 \qquad i = 1,...,n$$

$$\left.\begin{array}{l} g_j(x^*) \le 0 \\ \lambda_j^* g_j(x^*) = 0 \\ \lambda_j^* \ge 0 \end{array}\right\} \qquad j = 1,...,l \qquad (2.3)$$

$$g_j(x^*) = 0 \qquad j = l+1,...,m$$

Equations (2.3) imply a stationary point in the augmented Lagrangian function while constraints are all satisfied. The simultaneous solution of the $n+m$ variables, n from the initial design variables and m from the *Lagrange multipliers* (or *dual variables*), using the $n+m$ equations in (2.3) gives the set of all optimum points in the given domain.

To conclude with the mathematical definition of optima it is also important to consider, in spite of the conventional mathematical formulation of equation (2.1), that most structures during their service life are subjected to several loading conditions and moreover during their assembly, some of their assumed design variables are varied due to manufacturing imperfections. In the case of an optimum structure, this can have a dangerous effect on its performance. *Robust optimisation* (Su and Renaud (1997)) takes into account the random nature of the imperfections and incorporates a safety margin in the formulation of the optimisation problem. Therefore, in robust optimisation both the objective functions and the constraints

consist of two parts, the conventional function (equation 2.1) and an estimate of the variation of the functions in terms of probability functions.

2.3 Structural Optimisation Concepts

The concept of structural optimisation can be broadly grouped into *discrete structures* and *continuum structures*. Hence, different theories have emerged that are related to these two groups. Analytical and numerical methods have been applied to solve the optimisation process.

By discrete structures are identified structures such as beams, trusses and grillages. The optimisation of discrete structures is known as *sizing optimisation* and deals with the selection of the number and length of members defining the structure or definition of cross-sectional dimensions of these members. Research on this field started earlier than the one on continuum structures since discrete structures were easier to analyse using analytical methods.

The optimisation of continuum structures emerged with the development of computers and gained interest since. Its growth has been parallel to the development of the computational methods such as FEM and BEM allowing large-scale optimisation problems to be tackled with these modern methods of analysis. Methods in the optimisation of continuum structure include *shape* and *topology* optimisation. Shape optimisation modifies the boundary geometry of the structure to improve structural performance. In topology optimisation, the structure is allowed to change its topology during the optimisation process. The aim is to remove or redistribute material in order to arrive at a structural topology, which is in some sense optimal.

2.4 Solution Methods

There are two categories in which the solution methods can be divided, *optimality criteria methods* (OC) and *mathematical programming* (MP). Optimality criteria are

the conditions a function must satisfy at its minimum point. MP has a different philosophy, generally starting with an estimate of the optimum design for the problem and changing it iteratively until optimality conditions are satisfied.

The concept of *optimality criteria* as a basis of selection of a minimum-volume structure emerged in the early 1960s based on the seminal work of Michell (1904) on least weight plane trusses. Early work on Michell's theories led to the development of *optimal layout theory* by Prager (1967) which is the analytical form of OC, also in this field Venkayya, Gellatly and Berke (1973) pioneered the *discrete optimality criteria* (DOC) concept which derives from the extremum principles of structural mechanics, i.e. a numerical form of OC. Later developments in optimality criteria were based in a variational analysis approach so-called *continuum-based optimality criteria* (COC) (Rozvany (1995)) and more recently the introduction of the discrete formulation of COC known as *discrete continuum optimality criteria* (DCOC) by Zhou and Rozvany (1993).

The growth of structural optimisation has been marked by the growth of the finite and boundary element methods and the development of computers. Thus, complex structures are reduced to simple algebraic equations and easily solved by high-speed computers. Numerical search techniques, also referred as *mathematical programming* (MP) (Luenberger (1973)) methods, are applied to solve the optimisation problem being attractive to deal with the discrete design variables. A complete survey of optimum structural design in the context of mathematical programming procedures was presented by Schmit (1981), who was one of the precursors of using the features of numerical search techniques to the optimisation problem. Later developments include the introduction of *dual* methods where the variables are the *Lagrange multipliers*.

In addition to traditional MP and OC methods, new methodologies developed in the past twenty years have been successfully applied to both discrete and continuum structures. The *homogenization* method (Bendsøe and Kikuchi (1988)) introduces the concept of microstructure and composite elements. The parameters of the microstructure are the design variables. It uses the theory of homogenization to aggregate the macro structural properties to the microstructures which can vary from

being a complete void to a solid element. The homogenization methods create optimal topologies consisting of solid-empty-porous (SEP) elements with varying densities. More recent developments such as the *solid isotropic microstructure with penalty* (SIMP) method (Rozvany *et al.* (1992)) applies a power-law to the relative elemental property, thus varying between 0 and 1 and creating solid-empty (SE) topologies by explicitly penalising intermediate densities. Based on the *fully-stressed* techniques (Gallagher (1977)), the *evolutionary structural optimisation* (ESO) method (Xie and Steven (1993)) removes and adds elements of the FE mesh, according to some given criteria. Finally, heuristic methods such as GAs (Goldberg (1989)) have experienced an enormous growth with the implementation of parallel computing.

In the following sections a brief description of the methods presented above is given. Firstly, the methods of MP and OC are explained and followed by a description of the techniques used in size, shape and topology optimisation.

2.5 Algorithms

2.5.1 Mathematical Programming

Mathematical programming (MP) techniques were first applied to structural optimization in the late 1950s. MP methods deal directly with the design variables searching iteratively, based on gradient information, for the set of variables that minimise or maximise the objective function while satisfying the constraints. The incremental procedure starts with an initial estimation of the design variables. After that, as implemented in equation (2.4) following Arora's notation (Arora (1989)), a gradient-based search direction d and a step magnitude α are applied to improve the design from iteration k to the next iteration $k+1$ thus, iteratively driving the objective to its optimum value

$$x^{k+1} = x^k + \alpha d \qquad (2.4)$$

In general the problem posed in equation (2.4) cannot be solved in a finite number of iterations or it can attempt to find only a crude estimate of the minimum which is not desirable either.

In some formulations, associated to every MP is another MP called its *dual*. The original MP is called the *primal*. If the primal involves n variables and m constraints, the dual involves n constraints and m variables. Typically, the solution of either is sufficient for readily obtaining solution to the other.

The following sections explain briefly some of the MP methods and concepts used in optimisation; this description is based on Vanderplaats (1989). Also for more complete explanations about linear and non-linear mathematical programming see Luenberger (1973).

2.5.1.1 Unconstrained Optimisation

Unconstrained minimization problems are defined by equation (2.1) when the constraints are omitted. Although most engineering problems are not of this form they provide a basis for the constrained problem and are therefore used indirectly to solve constrained problems.

One of the best known unconstrained optimisation algorithms is the *steepest descent method*. In this method, the search direction d^k is calculated as the negative of the gradient of the objective function

$$d^k = -\nabla f(x^{k-1}) \tag{2.5}$$

Although this method is convergent, the rate of convergence cannot be reliable since it does not make use of the information on the previous iterations (Arora (1989)).

The method of *conjugate gradients* represents a simple modification to the steepest descent method but provides improvements in efficiency. This method uses the gradient information at the present design point and previous iterations to define the search direction as defined by

$$x^k = -\nabla f(x^{k-1}) + \beta x^{k-1}$$

$$\beta = \frac{\left|\nabla f(x^{k-1})\right|}{\left|\nabla f(x^{k-2})\right|}$$

(2.6)

2.5.1.2   Constrained Optimisation

Most engineering problems are constrained problems of the form of equation (2.1). The simplest method is known as *sequential unconstrained minimisation technique* (SUMT). The basic concept of this method is to convert the original problem to an unconstrained problem that can be solved by unconstrained optimisation methods. SUMT creates a penalty function that increases the objective for any constraint violation. Thus, the extended function has the form

$$\Phi(x) = f(x) + R\left(\sum_{j=1}^{l}\max\{0, g_j(x)\}^2 + \sum_{j=l+1}^{m}\{g_j(x)\}^2\right)$$

(2.7)

The penalty parameter R is initially taken as a relatively small value to avoid nonlinearity at the start and $\Phi(x)$ is minimised as an unconstrained function. Then R is increased and the process is repeated until no further improvements can be made on the objective function. This method is called the *exterior penalty function* method since the solution is iterated through the infeasible region. The method has the advantage of simplicity and reliability but, in contrast, it approaches the optimum from the infeasible region and the convergence is slow. Other methods approaching to the optimum from the feasible region are referred as *interior penalty function* methods. A more modern sequential unconstrained minimisation method is the *augmented Lagrange multiplier* method. Other applications of SUMT can be fund in Schmit and Miura (1976).

The *linear programming* (LP) (Luenberger (1973)) method has been used to solve linear systems being very efficient and reliable. These methods such as the *simplex algorithm*, are simple to implement however they are confined to applications where the objective function and constraints are linear functions of the design variables making its use more limited.

The *non-linear constrained optimisation* problem (Luenberger (1973)) can be approximated to a linear problem by using *sequential linear programming* (SLP). The basic concept is that the non-linear objective and constraint functions are first linearised and then this linear problem is solved by a linear optimiser. Thus, the constraints and objective can be approximated as follows

$$\tilde{f}(\delta x) = f(x^{k-1}) + \nabla f(x^{k-1})\delta x$$
$$\tilde{g}(\delta x) = g(x^{k-1}) + \nabla g(x^{k-1})\delta x$$

(2.8)

where $\delta x = x^k - x^{k-1}$ and the side constraints or also known as the *move limits*, $\delta x^L \leq \delta x \leq \delta x^U$. Thus, the optimisation problem is then reduced to a series of linear sub-problems that can be solved using the simplex algorithm. The move limits prevent unlimited moves as well that confine design changes to the feasible regions. Once the approximate optimum is found the problem is re-linearised and the process is repeated until it converges to the optimum. During the process, the move limits are reduced to ensure convergence. This reduction, often done through trial and error, makes the SLP problem very dependent on the move limits, showing lack of robustness as noted by different authors such as Arora (1989).

Optimisation problems with quadratic objective functions and linear constraints are defined as *quadratic programming* (QP). This problem is approximated to a linear one using the *Kuhn-Tucker* conditions and the introduction of *slack variables*. A variation of the QP formulation is the more powerful and robust *sequential quadratic programming* (SQP). In SQP the basic concept is to find a search direction d that minimises a quadratic approximation to the Lagrange function subject to linear approximation to the constraints, this is formulated as

$$\begin{aligned}
&\text{min} &&Q(d) = f(d^0) + \nabla f(d^0)d + \frac{1}{2}d^T \mathbf{B}d \\
&\text{subject to} &&g_j(x^0) + \nabla g_j(x^0)d \leq 0 \quad j = 1,\ldots,l
\end{aligned}$$

(2.9)

The matrix $\mathbf{B}$ is initially the identity matrix but it is updated using gradient information to approximate the Hessian of the Lagrange function $\nabla^2 L(x,\lambda)$. The solution of these problem yields a set of search directions d and Lagrange multipliers

λ. These values are then applied to perform a one-dimensional search using an exterior penalty formulation. SQP removes the need for a move limits and therefore it is more reliable and robust than the SLP. Nevertheless, it is computationally expensive involving careful formulation (Vanderplaats (1989)).

2.5.2 Optimality Criteria Methods

Optimality criteria (OC) methods are conditions which must be fulfilled by an optimal structure. One of the first OC to be considered was the intuitive concept of *fully stressed design* (FSD) (Gallagher (1977)). This concept refers to a design in which each structural member sustains a limiting permissible stress under at least one of the specified loading conditions. One feature of a FSD is the absence of an objective function whose extreme value is sought and therefore, there is no assurance that an algorithm for the calculation of an FSD will converge to the minimum weight. Moreover, the FSD approach is not unique, and will not always correspond to the optimum design for indeterminate structures under multiple load conditions.

OC methods based on mathematically derived conditions have their origins in the seminal work of Michell (1904). He applied elementary principles of mechanics and energy theorems to formulate a set of criteria for the general layout problem of isotropic plane trusses with stress constraints under a single load. Michell postulate the principles for the development of the *optimal layout theory*. The structures obtained through the solution of Michell's criteria, so called- Michell structures, for one-load case and under stress constraints, are constant strain structures and hence fully stressed designs. Moreover, the Michell theorems postulate some important concepts such as the term of *structural universe* as an initial design domain. A structural universe is the theoretical union of all possible members at every point in the design space.

2.5.2.1 Optimal Layout Theory and Continuum-Based Optimality Criteria

In spite of the important contributions of Michell's theorems his work remained ignored until the 1950s and 1960s, when his concepts were reconsidered by Hemp

(1973) and Prager and Shield (1967). They extended Michell's work to multiple loads and proved that Michell structures are optimum compliance structures. The systematic exploration of orthogonal nets for the geometry of Michell trusses by Hemp (1973) demonstrate that in some problems the optimum truss structure consists of infinitesimally spaced trusses. Based on the work initiated by Hemp and Prager, new approaches in OC emerged to cover a wider number of loading and structural conditions. These modern approaches are commonly referred to *continuum-based optimality criteria* (COC) and *optimal layout theory*. For a complete explanation on COC formulation see Rozvany *et al.* (1989).

The *optimal layout theory* is based on four fundamental concepts (Rozvany *et al.* (1995)): *structural universe*, *continuum-type optimality criteria* (COC), *adjoint structure* and a *layout criterion function*. The adjoint structure, concept introduced by Hemp (1973)), is a fictitious system with the same equilibrium and compatibility equations as the real structure but with different loads and initial strains. The layout criterion function Φ^e is derived from optimality criteria and normalised such that it takes the form

$$
\begin{aligned}
\Phi^e &= 1 && (\text{for } A^e \neq 0) \\
\Phi^e &\leq 1 && (\text{for } A^e = 0)
\end{aligned}
\tag{2.10}
$$

where A^e is the cross-sectional area of the member or element e.

The application of optimal layout theory has been restricted due to the difficulty of analytically satisfying the optimality conditions. An area where *optimal layout theory* has been successfully applied is in the design of least-weight grillages (Rozvany and Hill (1979)). A review on *optimal layout* theory can by found for example in Rozvany *et al.* (1995).

A variation to COC was the use of COC for iterative optimisation of large-scale systems in the so-called *iterative*-COC method (Rozvany *et al.* (1989)). This method implements the finite element analysis (FEA) into the optimisation; however the formulations continued to be continuous in nature since the FEA was used as a

14

black-box solver. Iterative-COC was applied to cross-sectional sizing of single beams and large truss systems.

2.5.2.2 Discrete Optimality Criteria

In spite of the COC methods and optimal layout theory, OC were simultaneously being developed for discretised systems as an alternative to the MP methods. This development was started in the late 1960s by Berke, Venkayya and Knot (1973). The *discrete optimality criteria* methods (DOC) are computationally more efficient than the primal MP allowing the application to large-scale problems. DOC methods are based on the idea that for a sufficiently constrained problem, the optimum lies on the constraints boundary in the solution space (Woon (2002)). Therefore, significant computational savings can be obtained by restricting the search to this sub-space through the Lagrange multipliers, also known as the *dual variables*. In these methods, the optimality criteria are defined through the application of the *Kuhn-Tucker* conditions as an integral part of the method formulation.

Another similar method conducting the search in the dual sub-space is known as the *dual method* introduced by Fleury (1979). This method uses the *Kuhn-Tucker* optimality conditions to produce primal-dual relations which are then used to get first-order approximations to the objective functions and the constraints. Thus, the primarily optimisation problem is replaced by a sequence of explicit sub-problems. For a convex primal solution space, the dual sub-space is also convex and has the same optimality conditions. Therefore, the solution to the dual problem simultaneously solves the primal problem. Examples and details of these methods can be found in Fleury (1979) and Fleury and Braibant (1986).

2.5.2.3 Discrete Continuum-Based Optimality Criteria

Rozvany and Zhou extended the basic ideas of the iterative-COC to discrete structural systems in the *discrete continuum-based optimality criteria* (DCOC) method (Zhou and Rozvany (1993)). The DCOC was formulated in a form suitable for incorporation in finite element codes. This allowed the application of the algorithm to more general structures. The concept of *adjoint structure* is still

fundamental in DCOC, reflecting the optimality criteria and constraints through initial strains and virtual loads. An improvement of DCOC over the iterative-COC method is due to the concepts acquired from DOC methods (Zhou and Haftka (1995)). *Kuhn-Tucker* optimality conditions were used to generate optimality criteria and an iterative algorithm for FE analysis was developed. The efficiency of DCOC lies in the especial feature that the Lagrange multipliers are evaluated explicitly at each step of the iterative procedure following the formulation proposed by Fleury and Schmit (1980).

2.5.3 Discussion of the Algorithms

Mathematical programming methods are in general robust and easy to apply to the optimisation problem. However, they present a number of drawbacks. Firstly, they depend on derivatives which are often expensive to calculate. Secondly, these methods can be inefficient if a good initial solution is not used. Thirdly, the number of computational calculations in MP methods is related to the number of design variables, and therefore, the computational cost can be prohibitive for large systems (Rozvany (1995)). Finally, most MP methods are primarily local searches which cannot guarantee a global optimum (Zhou and Haftka (1995)).

In spite of these drawbacks the capabilities of MP are still well considered in structural optimisation. While the global optimum may not be assured this may be less important in practical situations when the aim would be to improve the product. The key to success lies in the choice of an efficient approximation method and in taking the advantages of computer technology.

In OC methods, design improvements are carried out to satisfy *Kuhn-Tucker* conditions. Unlike MP methods, that update the design variables based on the local behaviour of the objective function, OC allow large design modifications. The efficiency of OC is related to the number of active constraints which in general are less than the number of design variables. Therefore, the combination of large design changes and reduced search space allow the application of OC to large-scale structural problems.

Nevertheless, these advantages are compromised if a large number of active constraints are present since they would require careful formulation (Patnaik *et al.* (1980)) and also make the problem highly dependent on the number of constraints (Zhou and Haftka (1995)). Moreover, analytical OC methods are less robust than MP due to the involved derivations necessary for each structural problem. A significant improvement is the implementation of the *Kuhn-Tucker* optimality conditions; however for some problems such as multi-criteria and multidisciplinary optimisation the formulations can be much more complex than using MP (Patnaik *et al.* (1980)).

2.6 Optimisation Methods

Structural optimisation problems can be solved using many different techniques, which can be broadly divided into *sizing, shape* and *topology* optimisation.

2.6.1 Sizing Optimisation

The structure is defined by a series of sizes and dimensions. Combinations of these sizes and dimensions are sought that achieve the optimisation criteria. There are two major categories of problems in size optimisation, *discrete* and *continuum* structures.

Optimisation of pin- and rigid-jointed frames (*discrete structures*) is the area that has received most attention over the last forty years. After defining the loads and supports of the structure, the sizes of the members are adjusted according to the optimisation criteria. Mathematical programming techniques and OC methods have been extensively applied for this type of problems. Examples of the application of such techniques in Zhou and Rozvany (1996) for OC and in Schmit and Farshi (1974) for MP. Recently, more heuristic techniques have emerged also for sizing optimisation, such as ESO (Xie and Steven (1997)).

In sizing optimisation of *continuum structures*, the structure can be described as a series of sizes or parameters. Optimisation techniques are then used to find the combination of design variables that give the best result. Methods generally used are

mathematical programming and heuristic methods such as ESO (Chu *et al.* (1997)) or genetic algorithms (Osyczka (2002)).

2.6.2 Shape Optimisation

In a *shape optimisation* problem the aim is to improve the shape of a given topology by minimising an objective function subjected to a certain equality and inequality constraints. Generally, sensitivities to changes in the geometry are evaluated together with optimisation techniques to produce optimal structural designs.

With the development of numerical methods in the late 1960s, the interest in shape optimisation has grown considerably. However, there have been some difficulties inherent to shape optimisation (Haftka and Grandhi (1986)). The first one is the continuously changing boundary, thus it is difficult to ensure that the accuracy of the structural analysis remains adequate throughout the design process. Second the effort of calculating the sensitivity derivatives can be considerable, especially for continuum structures. Nevertheless, these problems are being improved with the fast progress of numerical methods and the everyday more powerful computers.

2.6.2.1 Design Variables

Early work on shape optimisation was using the coordinates of the boundary nodes of the finite element model as design variables (Zienkiewicz and Campbell (1977)). In this approach, the number of design variables can be very large and if a mathematical programming method is used the computational effort for the sensitivity analysis calculations can be too expensive (Haftka and Grandhi (1986)). In addition, since the shape changes through the process the initial element mesh becomes too distorted to produce reliable results.

Alternatively, a boundary representation is introduced to overcome these drawbacks. Initially, polynomials are used to describe the boundaries and the coefficients of the polynomial being the design variables. Francavilla *et al.* (1975) used polynomials and their coefficients as the design variables. A more general approach is to use

shape functions to define the boundary with the coefficients being the design variables (Kristensen and Madsen (1976)).

The use of high-order polynomials can reduce the total number of design variables but can result in oscillatory boundaries for high-order polynomials due to numerical instability with higher order curves (Ding (1986)). Hereby, splines are introduced to avoid this problem since they are composed of low-order polynomial pieces combined to ensure smoothness. Bézier and B-splines were used by Braibant and Fleury (1984). These parametric geometries are defined by a control polygon whose vertices (control points) are used as design variables. The high flexibility in terms of geometry description results from the appropriate bending functions used. In addition, B-splines exhibit local control and therefore, changes in the control point position are not propagated throughout the whole spline.

More recently rational B-splines and more specifically the non-uniform rational B-splines (NURBS) have been implemented to shape optimisation since there are more flexible than the more general B-splines (Schramm and Pilkey (1995)). The advantage of rational B-splines comes with the inclusion of the vertex weights into the sets of design variables. Thus, the design space is extended to rational functions and therefore, the number of possible modifications to the structure is much greater.

An alternative technique is the use of *design elements*. In this method introduced by Imam (1982) the structure is divided into a few regions or design elements. These regions are described by a number of master nodes that control the geometry. These master nodes define movement directions of the nodes and their perturbation being the design variables. Braibant and Fleury (1984) used B-splines to define the boundaries of the design element. A recent application can be found in (Kegl (2000)) where three-dimensional design elements are parameterised with Bézier surfaces.

2.6.2.2   Structural Analysis

The finite element method is the most common numerical method to perform the structural analysis. However, the FE model imposed can result in inaccuracies when

applied to shape optimisation. Meshing considerations are necessary to prevent meshing distortion and inaccuracies due to the boundary changes. An initial solution for this problem was to perform manual mesh refinements to the badly distorted elements as we can see in Oda and Yamazaki (1977).

Nevertheless, the manual mesh refinement initially used was followed by the more efficient automated mesh generation with adaptive mesh refinement. Using analysis error estimators, regions which need further refinement are identified and remeshed appropriately to accommodate boundary changes. Work on automated mesh generation applied to shape optimisation can be found in Queau and Trompette (1980), Botkin (1981) and Botkin (1992), amongst others. The early mesh generators used were computationally costly but nowadays, great improvements have been made in automated mesh generators reducing very much the computational cost and increasing the efficiency (Sienz and Hinton (1997)).

Alternatively to the FEM, the boundary element method (BEM) (Becker (1992)) is also used in shape optimisation. This method is a natural alternative to the FE method since only the boundary is discretised and therefore, re-meshing is applied exclusively to the boundary. The number of elements is reduced but high accuracy and smoothness is ensured in the boundaries when a suitable mesh is used. Furthermore, this method is well suited for sensitivity design calculations due to the accuracy of the boundary representation.

Mota Soares *et al.* (1984) applied the BEM to optimise the shape of shafts using a variational approach to sensitivity calculation and linear elements. Kane (Kane (1986)) presented a boundary element formulation for plane strain or plane stress linear elasticity combined to implicit differentiation of the discretised boundary integral equations to perform the design sensitivity analysis. More recently, Zhao (1995), Parvizian and Fenner (1997) and Cerrolaza *et al.* (2000). Also Burcinsky and Orantek (2002) have applied the BEM combined with *genetic algorithms* (GAs) to optimisation problems and crack identification.

2.6.2.3 Solution Methods

Most of the methods used for solving shape design problems are related to mathematical programming (MP). Applications of these methods combined with finite element analysis can be found in Zienkiewicz and Campbell (1977). Sequential linear programming was applied by Braibant and Fleury (1984). Examples of application of penalty function methods are presented by Queau and Trompette (1980).

Alternatively, optimality criteria methods are used in shape optimisation. Dems and Mroz (1978) introduced a general approach that combines optimality criteria and variational methods. A similar approach was used by Na *et al.* (1984).

Heuristic methods have been also applied to shape optimisation. These methods are known as *zero order* or *gradientless methods* since they require no sensitivity calculations. Oda and Yamazaki (1977) developed an iterative FE-based technique for resizing the boundary elements based on their stress ratio. In a similar fashion, Schnack and Spörl (1986) introduced another method for the reduction of stress concentration on boundaries through the gradual removal of low stress material. The biological growth method was introduced by Mattheck and Burkhardt (1990) based on the hypothesis that in nature structures such as trees have a constant stress distribution over the boundaries.

2.6.3 Topology Optimisation

It is of great importance for the development of new products to find the best possible topology for given design objectives and constraints at a very early stage of the design process. For this reason, over the last decade, substantial research has been devoted to the development of topology optimisation techniques. These techniques can be divided very roughly into two kinds of topology design processes; the *material* or *microstructure technique* and the *geometrical* or *macrostructure technique.*

2.6.3.1 Microstructure Approach

Typically, this problem is represented by an initial design that is discretised with a suitable FE mesh. The optimisation consists in determining whether each element in the continuum should contain material or not. To this end, the density of material within each finite element is used as a design variable defined between the limits 1 ($\rho = 1$, solid material) and 0 ($\rho = 0$, void or very weak material) and known as the 1-0 problem. Such a formulation, however, has been reported to be mesh dependent and unstable (Bendsøe (1995)) and noting that the true optimum results may contain infinitesimal cavities of *grey areas* ($0 < \rho < 1$) which are not accomplished using 1-0 methods. Therefore, by using relaxation of the state equations composite material structures, which include a microstructure, are used. The material density function ρ, taken as design variable, allows the possibility of solid material ($\rho = 1$), voids ($\rho = 0$) and *grey areas* ($0 < \rho < 1$) composed of a porous composite model.

2.6.3.1.1 Homogenisation

Homogenisation has been implemented in structural optimisation by Bendsøe and Kikuchi (1988), and Hassani and Hinton (1998) amongst others. This method solves the material distribution problem of topology optimisation using the concept of microstructure combined with the theory of homogenisation (Sanchez-Palencia (1980)) to consider the material properties at a macroscopic level.

In a microstructure, the base cell, which is the smallest repetitive unit of material, is periodically distributed through the domain in the form of small holes or micro-voids creating a composite material with a varying density. The density function is a function of the micro-voids whose geometrical variables are the design variables. Typically the composite is orthotropic and the angle of rotation of the directions of orthotropy is considered as an additional design variable.

The formulation transforms the 1-0 topology problem into a sizing problem. Since the number of design variables is usually large, the solution methods are often generated based on OC. For details of the implementation of OC and variable updates see Bendsøe (1995). For a given microstructure, the macroscopic properties

such as the elastic modulus are computed from the microscopic properties of density and cell orientation by using the theory of homogenisation. For the mathematics involved in this theory see (Sanchez-Palencia (1980)).

Basically, after choosing the reference domain and finite element discretisation, it is assumed that each element consists of a cellular material with a specific microstructure. In homogenisation methods there are a number of types of basic unit cells. These can be divided into two categories: *rank-layered material* cells and *microcells with internal voids*. In the rank-layered material cells each cell is composed of layers of different materials and voids. In practice, to avoid singularity in the stiffness matrix of the structure, instead of voids a very flexible (weak) material is used. For example the rank-2 composite is where the first layer of strong and weak material is used as the weaker material in a second layering that is orthogonal to the first. Studies (Avellaneda (1987)) on the material properties of composites of two materials have shown that in plane elasticity problems, the rank-2 microstructure generates the stiffest composite material. For more practical solutions, sub-optimal microstructures can be used in the form of micro-voids within each finite element. The most commonly used are square or rectangular micro-voids (Eschenauer and Olhoff (2001)).

Nevertheless, structures optimised based on an optimal microstructure produce topologies with the so-called *checkerboard pattern*, i.e. with large grey or porous areas with intermediate densities. Topologies presenting this pattern give the structure an artificially high stiffness compared with a topology with uniform material distribution, therefore they are impractical in most engineering applications as noted by Bendsøe (1995) and Sigmund and Petersson (1998). Other numerical problems which can occur are the mesh-dependency problem and the local minima as reviewed by Sigmund and Petersson (1998).

2.6.3.1.2 Artificial Materials

There is another group of methods that solve the material distribution problem by sizing directly the elasticity matrix E or other macro scale properties. These methods are referred to as using an *artificial material* since they use a continuous artificial

density function $0 \leq \rho \leq 1$ which becomes the design variable. Intermediate densities are penalised in order to encourage solid-void topologies. Topology optimisation applications that use an artificial material model include Maute and Ramm (1995). A similar concept of using a penalised artificial material for the generation of solid-void solutions was derived by Rozvany *et al.* (1992) in the *solid isotropic material with penalty* (SIMP) method. This method penalises and suppresses in the solution porous regions by adding to the material costs the cost of manufacturing of holes. Results suggest (Rozvany *et al.* (1992)) that the topologies obtained through SIMP are closer to the analytical optimum than the ones obtained by using square cells. Bendsøe (1995) noted that artificial material schemes suffer from a non-existence of solutions and are mesh dependent. However, Rozvany claimed that their results are not highly mesh dependent (Rozvany *et al.* (1992)).

2.6.3.2 Macrostructure Approach

Another approach to the topology optimisation problem is to formulate the problem where the basic units are the discrete elements of the FE mesh. Unlike the microstructure approach, these methods are heuristic and simple to implement in any FE package. However, they are known to be mesh dependent.

The *soft kill* (Baumgartner *et al.* (1992)) strategy simulates adaptive bone mineralization in which high stresses lead to a higher degree of mineralization than lower stresses. This is accomplished by varying the Young's modulus according to a calculated stress distribution. Hereby, elements at higher stress rates have increased their Young's modulus and the elements at lower stress states decreased. In this way, the inefficient elements increasingly withdraw themselves from carrying the load and once they do not contribute significantly, they are removed from the set of elements.

More dramatically, the *hard kill* (Hinton and Sienz (1995)) strategies change the element properties, usually the Young's modulus, to a near zero value representing a softening of the material. Thus, while soft kill methods vary their Young's modulus according to a linear relationship of the stress measure, the hard kill method uses a step function which changes from one to zero (or near zero). In this fashion, in the

24

evolutionary structural optimisation (ESO) method introduced by Xie and Steven (1993) the domain is discretised using a uniform rectangular element mesh. Elements are then slowly removed based on a selected criterion and the structure evolves towards an optimum. For example, for stress considerations, the lightly stressed elements are removed. Querin *et al.* (1998) introduced the *bidirectional* ESO (BESO) that not only removes material in low stressed areas but allows the addition of elements to highly stressed areas. Since then, ESO has been broadly applied to a wide range of optimisation problems considering stiffness, frequency, buckling, and multicriteria optimisation. For a review of applications see Xie and Steven (1997), and the latest developments see Steven *et al.* (2002).

Similar methods to ESO include the so-called *reverse adaptivity* (RA) (Reynolds *et al.* (1999)) and *evolutionary material translation* (EMT) (Reynolds *et al.* (2001)). In RA the low-stressed regions are re-meshed in order to increase the accuracy in these regions. This method produces topologies to great detail but for each iteration, two FE analyses are required. In the case of the second method presented, the EMT is very similar to the BESO.

A different approach to the previous methods is the *bubble* method (Eschenauer *et al.* (1994)), which uses an iterative positioning and a hierarchically structured shape optimisation of new holes, so-called bubbles. This means that the boundaries of the component are taken as parameters. Shape optimisation of the new bubbles and of the other variable boundaries of the component is carried out as a parameter optimisation.

More recently, probabilistic methods such as genetic algorithms and simulated annealing have emerged as heuristic approaches to be applied to optimisation problems. Both algorithms rely on observed phenomena in nature and in their implementation they use random selection processes which are guided by probabilistic decisions. These methods can find near and good optimum solutions; however the computing time to achieve these solutions can be very expensive.

Genetic algorithms (GAs) were introduced by Holland (1975) in the 1970s and then extended by Goldberg (1989). They were designed to mimic the natural evolutionary

process, which leads to the survival of the fittest individuals. The structural optimisation can be achieved by allowing a *population* of *chromosomes,* traditionally binary mathematical strings, representing possible solutions to evolve through the basic random operations of *selection, crossover* and *mutation* towards an optimum solution. Since the evolutionary nature of GA they are often termed as *evolutionary algorithms* or *evolutionary optimisation methods.* However, GA should not be confused to the *evolutionary methods* presented in the following section 2.7.

GAs have been proven to be robust (Goldberg (1989)), and effective in finding global optima. These techniques have been applied in diverse optimisation problems. Hajela and Lee (1995) applied GAs to develop optimal topologies of truss structures. Kita and Tanie (1999) used GAs to optimise the topology and shape problems of continuum structures. The boundary profiles are expressed by spline functions. The GA is used to find the number of internal boundaries and the position of the control points of the spline functions that result in the optimum topology. Burczyński and Orantek (2002) applied evolutionary algorithms to a variety of optimisation problems such as truss structures, elastoplastic problems, dynamic loading and crack identification. Recent reviews include the book of Osyczka (2002) on evolutionary algorithms for single and multicriteria optimisation and S.Y. Woon's thesis (Woon 2002) on GA-based optimisation of continuum structures.

GAs are excellent when there is a complex design space with multiple local minima and also for very rapid assessment of fitness of an individual member of the population. However, the question of computational efficiency is the reason why these methods have been slow in being adopted for continuum shape and topology optimisation. Most FE/BE runs will take too long when considering GA, possibly several thousand runs, requiring powerful computers and parallel systems.

Simulated annealing (Laarhoven and Aarts (1987)) has been developed motivated by studies in statistical mechanics of annealing in solids. A low energy state usually means a highly ordered state. To accomplish this, the material is first melted, i.e. heated to a temperature that permits atomic rearrangements. Then it is cooled slowly and as the temperature is reduced the atoms migrate to a more ordered state with lower energy. The final degree of order depends on the temperature cooling rate.

The analogy between the *annealing* and the optimisation of functions was established by Kirkpatrick *et al.* (1983) by replacing the energy state with an objective function *f*, and using the design variables for the configurations of the particles. Applications of *simulated annealing* to frame structures with discrete valued variables were considered by Balling (1991). Blachut (2003) has applied simulated annealing to maximise the external hydrostatic pressure in a barrel shell. More recently, to reduce the computational effort, parallel computing has been implemented into the algorithm. Examples of this can be found in Leite and Topping (1999) amongst others.

Recently, new methodologies have emerged that not only optimise the design but also take into account the process involved during the design. These techniques, often identified as *design and process optimisation* (DPO), include *robust design optimisation* (RDO), *reliability based design optimisation* (RBDO) and *design of experiments* (DoE) generally, coupled with *response surface analysis* (RSA). GA and simulated annealing are usually implemented together with these methods.

Robust design optimisation (RDO) (Su and Renaud (1997)) yields a system that performs with minimal variability in the response in the case of input variations or uncertainties. Graphically, the minimum value of the RDO objective function that represents the optimum is as flat as possible rather than a steep valley obtained in conventional optimisation. The aim of RDO is not only to minimise or maximise the primary objective function but also to minimise the sensitivity of the solution to exhibit uncertain deviations from the nominal state. In this sense the RDO can be regarded as bi-objective requiring not only that the performance of the solution be brought towards a target but also the variation from the target is minimised. Recent applications of RDO are found in Messac and Ismail-Yahaya (2002) and Bates *et al.* (2002).

In the *reliability based design optimisation methods* (RBDO) (Frangopol and Maute (2003)) the purpose is to produce structures which are economic and also reliable by introducing safety criteria into the optimisation process in order to consider different sources of uncertainty. These uncertainties refer to changes between the final design and how it is eventually made, for example using a different material or changing a

dimension of the structure. RBDO methods are based on stochastic analysis methods and therefore regarded as computationally more expensive compared to deterministic approaches. Techniques used here are GA, simulated annealing and also Monte-Carlo simulations (Frangopol and Maute (2003)).

Response surface analysis (RSA) and *design of experiments* (DoE) (Gambling *et al.* (2001)) are applied to design optimisation generally to reduce the cost of expensive analysis. RSA allows to construct approximations using results of the response analysis calculated at certain points in the design variable space and thus, minimising the number of response evaluations. The selection of the location of the points where the response should be evaluated is done using DoE methodologies.

2.7 Evolutionary Methods

This last section is devoted to explain in more detail the optimisation methods in which this PhD has its basis. Firstly, the *biological growth* method is presented, followed by *hard-kill* and *self-designing structures* methods. Finally, the *evolutionary structural optimisation* method is extensively described and discussed.

2.7.1 Biological Growth

Mattheck and Burkhardt (1990) introduce the shape optimisation method called CAO (*computer aided optimisation*). This method is based on the axiom of *constant stresses* derived from analogies observed in the growth of biological structures such as trees, bones and horns. He proposes the idea of copying the mechanism of tree growth in order to optimise mechanical engineering structures. He found that trees adjust their growth in a fashion that the stresses on the surface are equally distributed by reducing the stress peaks. This idea is similar to the gradientless shape optimisation presented by Schnack and Spörl (1986) which also argues that local stress peaks are the main reason for fatigue failure of structures. In CAO, a thermal analogy is implemented for the swelling phenomenon, i.e. heating the structure in areas of non-admissible stresses and letting the surface expand. The structures

obtained from this approach do not show any significant stress peaks and their fatigue life is proved to have increased.

In addition, Baumgartner *et al.* (1992) observed that in nature, all unnecessary weight is avoided and that material decays where it is no longer needed. He introduced the so-called *soft kill option* (SKO). This topology optimisation algorithm simulates adaptive bone mineralization in which high stresses lead to a higher degree of mineralization than lower stresses; i.e. parts under higher loads will become stiffer than areas exposed to lower loads. This is accomplished by varying the Young's modulus according to a calculated stress distribution. Thus, elements at higher stress rates have their Young's modulus increased and the elements at lower stress states have it decreased. Repeating this procedure, the inefficient elements increasingly withdraw themselves from carrying the load and once they do not contribute significantly, they are removed from the set of elements.

This heuristic approach, which has been integrated to finite elements, proves that combining both methods, SKO followed by CAO, leads to a lightweight design that is cost efficient and durable. Nevertheless, the objective function of the method is to reduce stress peaks along the optimization boundary/domain and hence primarily provides an optimum solution with respect to fatigue and fracture failure.

2.7.2 Hard Kill Methods

The *hard kill* (HK) (Hinton and Sienz (1995)) methods form the basis for the *evolutionary structural optimisation* methods and *self-defining structures*. The HK methods are based on FE models and the solution is achieved iteratively by removing low stressed elements from the mesh. Since it is important to maintain a smooth transition between iterations the amount of material removed at each stage is small.

The removal of material from an evolving design is typically done by changing the material properties, usually the Young's modulus, to a near zero value, representing softening of the material. If the element stress is less than a certain fraction, usually

called removal ratio and determined by the user, of the maximum stress, then the element is killed by changing its material properties.

Usually, the number of elements removed at each iteration will stabilise, for a given removal ratio, representing a stable evolution for the structure. To promote further change the removal rate is increased by a constant amount, typically called the evolutionary rate and also decided by the user, which has the effect of increasing the threshold at which the elements are killed. Following this process, in all regions of the design the stress will eventually be within a certain value of the maximum producing a *fully stressed design*.

HK methods are conceptually very simple and easy to implement, however they present some drawbacks, mainly related to mesh dependency such as *checker-board patterns* and *jagged edges* (Li *et al.* (2001)).

2.7.3   Self-Designing Structures

The so-called *self-designing structures* approach embraces two methods trying to produce low-weight and low-stress structures. These methods are identified as *evolutionary material translation* (EMT) and *reverse adaptivity* (RA).

2.7.3.1   Reverse Adaptivity (RA)

The *reverse adaptivity* (RA) method developed by Reynolds *et al.* (1999) is based upon a combination of existing adaptive finite element methods and evolutionary structural optimisation (ESO) methods. Although it is a HK method it differs in that the finite element mesh is substantially modified during evolution. The technique works identifying which elements are low stressed. The criterion to identify such elements follows a *cut-off* stress value related to the *area removal limit* (AR). Then, these low-stress elements are subdivided to a calculated target size. The refined model is then re-analysed and the elements with a stress value lower than the cut-off stress removed. Structural integrity checking algorithms operate after removal of material controlling if the loads have become disconnected from the constraints and

avoiding *checkerboard* patterns (see figure 2.1). In such cases the model is restored, the cut-off parameter is reduced and the processes repeated.

A main feature of this method is that RA refines the mesh in areas of interest. Unlike conventional adaptivity like *h*-adaptive meshes, which refines the mesh in the zones of high stress, this refinement takes place in zones of low stress and reason why this method has been named *reverse*. In the optimisation literature, adaptivity methods have also been applied by many authors such as Maute and Ramm (1995) and discussed by Sienz and Hinton (1997), in contrast to RA, they applied techniques based on error estimators, instead of exclusively remeshing the low-stress areas.

checkerboard pattern   jagged edges
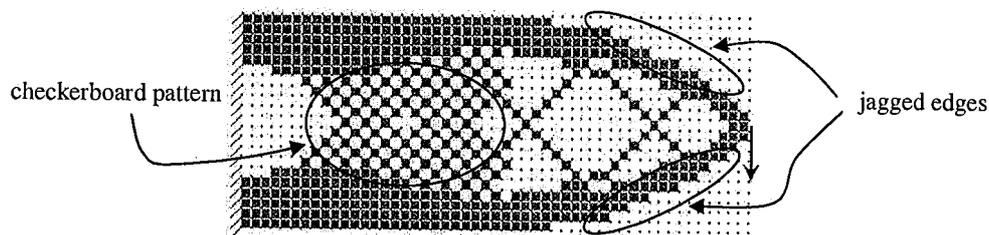
Figure 2.1: Example of a model with checkerboard patterns and jagged edges (Li *et al.* (2001))

Another important feature is that the elements are physically removed from the mesh of the structure, as opposed to simply being switched off by a decrease in the Young's modulus (HK). So that difficulties in handling very large data structures are avoided. Another difference with HK is the improved boundary definition. With successive decrements of adapted element size, structures evolve with an increasingly well-defined shape. However, the disadvantage of the RA method is that it requires two analyses for each iteration in contrast to one in the HK method.

2.7.3.2 Evolutionary Material Translation (EMT)

The *evolutionary material translation* method (Reynolds *et al.* (2001)) is based on the RA method (presented in the previous section) combined with original methods for material addition. Like the HK and SK methods, the main idea of this method is to mimic, on the computer, how bones grow. The mechanical simulation uses the stress obtained from the FEM to suppress finite elements in the low stressed regions

and at the location of high stress; finite elements are added to increase the volume of the structure.

The procedure starts with the definition of the initial mesh for the problem together with the loads and constraints. Unlike HK or RA methods there is no need for an over-large mesh to be defined. An *area removal limit* (*AR*) and an *area addition limit* (*AA*) are used to specified the amount of material to remove and add, respectively, as a proportion of the existing area. The process of material removal is done following the RA method and after that, a new FE analysis is carried out before the process of material addition. Once the optimum is reached a mesh improvement can be done, by Laplacian smoothing and mesh relaxation techniques.

One of the advantages of EMT is that is capable of reducing the maximum stress in a structure while reducing the amount of material required. This feature is usually impossible to achieve in a HK method, where material is only removed, as discussed by Querin *et al.* (2000). In fact, the EMT method has many similarities to the bidirectional evolutionary structural optimisation (BESO) (Querin *et al.* (2000)).

2.7.4 Evolutionary Structural Optimisation (ESO)

The Evolutionary Structural Optimisation method (ESO) was presented by Xie and Steven in 1993 (Xie and Steven (1993)) and since then, it has been under continuous development. It is based on the simple concept that by slowly removing inefficient material from a structure, the residual shape evolves in the direction of making the structure better.

Initially, a topology large enough to cover the area of the final design is divided into a finite element mesh. Then, a set of loads and constraints is applied and the stress analysis is performed. Henceforth, elements that satisfy the *rejection criterion* shown in the following equation are eliminated

$$\beta_e \geq RR\beta_{max} \qquad (2.11)$$

where β_e is the selected criterion, typically the element von Mises stress, β_{max} is the maximum value of the selected criterion, such as the maximum von Mises stress and *RR* is the *rejection ratio*.

Such procedure is repeated using the same value of *RR* until a *steady state* is reached, which means that no more elements are removed. At this stage an *evolution rate* (*ER*) is introduced and added to *RR*. Further iterations take place until a new steady state is reached. This evolutionary process continues until a desired optimum or specific target is reached.

To help measure the quality of the optimisation methods, Querin *et al.* (1998) have introduced the concept of a performance indicator (*PI*) that measures how well the overall structure is performing against an idealised situation such as a *fully stressed design* (FSD). The *PI* is defined as

$$PI = \frac{\displaystyle\sum_{elements} \sigma_{VMe} V_e}{FL} \qquad (2.12)$$

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. This is a non-dimensional number and while it is problem specific there is often a target value for each problem.

This simple evolutionary procedure has as natural corollaries the *additive evolutionary structural optimisation* (AESO) (Querin *et al.* (2000)) and the *bidirectional evolutionary structural optimisation* (BESO) (Querin *et al.* (1998)) which are presented in the following subsections.

2.7.4.1 Additive Evolutionary Structural Optimisation (AESO)

The additive evolutionary structural optimisation (AESO) method (Querin *et al.* (2000)) generates optimum structures by an additive technique that starts from a minimum possible design space rather than a maximum and evolves by adding material rather than removing it.

The AESO procedure starts specifying the maximum allowable physical domain with marked location of loads and supports. Then, the entire domain is subdivided in a finite element mesh and the minimum number of elements that will connect the supports to the loads are specified to avoid a singular finite element solution. All elements within the design domain not marked as initially connecting elements are given a property value of zero, i.e. they are still stored in the data file for the structure but they do not physically exist as part of both the structure and the finite element solution. After carrying out a FE analysis, the material is added in an evolutionary manner, similar to classic ESO. This evolutionary process is achieved by using the following driving criterion

$$\beta_e \geq IR\beta_{max} \tag{2.13}$$

where β_e is the selected criterion such as the element von Mises stress, β_{max} is the maximum value for selected criterion such as the maximum von Mises stress and IR is the *inclusion ratio* which is generally determined from numerical experience.

If a *steady state* or local optimum is reached, i.e. when no more elements are added to the critical regions, then the IR is modified by the *evolutionary rate ER* and the process continues. The steps of FEA and addition of material are repeated until the stopping criterion is achieved such as the objective function has been reduced, the design domain has been fully populated or until the *performance index* (PI) (see equation. 2.12) has reached the appropriate target value within a prescribed tolerance.

To conclude, this method adds material where it is most needed, but unlike ESO, does not remove any from the inefficient regions. Therefore, a new method that combines the properties of AESO together with the removal capabilities of ESO has been developed so-called the *bidirectional structural optimisation* (BESO).

2.7.4.2   Bidirectional Evolutionary Structural Optimisation (BESO)

The bidirectional algorithm (Querin *et al.* (1998)) allows the method to search all possible directions which include not only removal of material but also the addition of material where needed.

The addition and removal processes are the AESO and ESO methods respectively; however, when both methods are combined to work together, some of the governing equations that drive both methods may be modified for the BESO method to work properly. The initial design can be either the entire domain (ESO) or the minimum connecting domain between the loads and supports (AESO). After the FEA, elements can be removed or added to the structure if they satisfy these equations, respectively

$$\beta_e \leq RR\beta_{\max}$$
$$\beta_e \geq IR\beta_{\max}$$

(2.14)

where β_e is the selected criterion such as the element von Mises stress, β_{max} is the maximum of the selected criterion such as the maximum von Mises stress, RR is the *rejection ratio* and IR is the *inclusion ratio*. Similar to ESO and AESO, if a *steady state* is reached, then the RR and IR are increased and decreased, respectively by the *evolutionary rate ER*. This process is repeated until the *performance index (PI)* (equation 2.12) reaches its minimum, or until the criterion limits are reached.

2.7.4.3 Discussion on ESO

ESO was originally formulated using the von Mises stress criterion (Xie and Steven (1993)). Thus, it optimises the structure by slowly removing elements with low stress, approaching towards a fully stressed design (for a single load case). Moreover, as previously presented, the method was later developed to add as well as to remove elements in the BESO (Querin *et al.* (1998)). More recently, ESO has been extended to consider different physical situations and various optimisation criteria, therefore becoming a more practical method. In such cases, the element's alteration is determined in terms of its effect on the design objective using the concept of element sensitivity. Detailed explanation on the element-based sensitivities can be found in (Steven *et al.* (2002)) covering a broad range of criteria and physical situations. ESO developments include the implementation of stiffness and displacements (Chu *et al.* (1997)), multi-criteria problems (Proos *et al.* (2001)), the application to multiple load (Xie and Steven (1997)), general physical field problems (Xie and Steven (2000)) such as thermal problems, elastic torsion shafts,

incompressible fluid flow, elastostatic field and magnetostatic field and frequency and buckling problems (Xie and Steven (1997)).

ESO presents some drawbacks and weaknesses as has been reported by Zhou and Rozvany (2001). These shortcomings are related to the rejection criteria caused if the element sensitivity number for the rejected element increases significantly as a result of being normalised. The ESO formulation is mesh dependent and consequently, the basis for its main disadvantages. Although the definition of the initial mesh is fairly simple, usually involving a uniform mesh of quadrilateral (in 2D) or hexahedral (in 3D) elements, the final solution, due to the fixed nature of the mesh, can result in jagged edges and structural interconnections. For example figure 2.1 represents a boundary whose elements are joined in a chain by their corner nodes. It is clear that this is an undesirable situation, since the accuracy of the FE results is in doubt. In addition, post-processing, such as spline construction, must be carried out to smooth the boundary for manufacturing reasons (Hinton and Sienz (1995)).

The fixed nature of the mesh also leads to the problem that when material is removed at each iteration, it has fewer and fewer elements with which to approximate the stress solution. An alternative would be to decrease the element size as the evolution proceeds to keep the computational efficiency on the results.

Another drawback is found during the evolution process since it can produce disintegration of the structure if excessively large values are used. The removal of material can often disconnect the loads from the constraints. The method used by Hinton and Sienz (1995) is to impose an upper area limit on material removed.

Checkerboard patterns refer to the phenomena of alternating presence of void and solid elements ordered in a checkerboard fashion. These patterns, which are depicted in figure 2.1, are quite common in various fixed grid finite element based structural optimisation methods. Such patterns complicate the interpretation of optimal material distribution and geometry extraction for manufacturing reasons. The cause of checkerboard formation is likely to be related to the finite element approximations when using low-order elements and occurring in different design

criteria. With the ESO method, material replacement of an element is determined in terms of its relative reference level. In this sense the appearance of the checkerboard patterns reflects an improper estimation on the elemental evolutionary criterion. Such a situation is also caused by the poor numerical behaviour of four-node elements (in 2D models). The use of higher order elements, e.g. eight-node elements, can significantly reduce the occurrence of checker boarding. However, this solution may not be appropriate due to the considerable increase in computational time. Kim *et al.* (2000) developed an algorithm of *intelligent cavity creation* (ICC), in which the checker boarding patterns (with numerous cavities) can be eliminated through controlling the number and scale of structural cavities in the final topology. Also in the fashion of perimeter control, Yang *et al.* (2003) presented a perimeter control technique incorporated into BESO. Li *et al.* (2001), in order to improve the estimation quality of elemental sensitivity or reference level in low order elements, suggested a weighted average algorithm to balance the *over* or *under* evaluation of the evolution criteria. This intuitive smoothing filter technique does not alter the mesh of the finite element model nor increase the degrees of freedom of the structural system, and therefore does not affect the computational efficiency.

Finally, another task pointed out by Querin *et al.* (1998) is related to the solution time, that is, the necessity to make the ESO process faster so that the designer is able to get the optimum design within a few seconds of describing the environment. Research in this topic has been carried out combining ESO and the *fixed grid* (FG) method (Kim *et al.* (2000)), instead of FE, to increase the performance without losing the accuracy of the results.

3

BOUNDARY REPRESENTATION. NONUNIFORM RATIONAL B-SPLINES

3.1 Overview

Nonuniform rational B-splines (NURBS) have become the standard for describing and modelling curves and surfaces in *computer aided design* (CAD) and computer graphics. They are used to model everything from automobile bodies and aircraft components to skis and vacuum cleaners. To fully exploit the flexibility of NURBS some knowledge of the mathematics involved is necessary. In this chapter an introduction to NURBS is presented to help to understand these parametric curves and surfaces. The purpose is to present a clear definition of NURBS and their properties since some of them are key features in the research work of this thesis. A detailed historical review on NURBS is given in Rogers (2001). Also, for a description of parametric curves and surfaces see Farin (1988).

It is worth saying that each figure and graphical illustration presented in the following chapter has been precisely computed using the developed algorithm for boundary representation.

3.2 Introduction

This discussion of NURBS curves and surfaces begins with looking at their antecedents, specifically *Bézier curves* (Farin (1988)). Pierre Bézier developed a method for shape description using the *Bernstein basis* or polynomial approximation function. Thus, mathematically, a parametric nth-degree Bézier curve is defined by

$$C(u) = \sum_{i=0}^{n} B_{i,n}(u)P_i \qquad 0 \le u \le 1 \qquad (3.1)$$

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

$$B_{i,n}(u) = \frac{n!}{i!(n-i)!} u^i (1-u)^{n-i} \qquad (3.2)$$

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

- The degree of the polynomial defining the curve segment is one less than the number of control points.

- 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*. Basically, 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 that

- High degree is required in order to satisfy a large number of constraints and also to accurately fit some complex shapes. Moreover, high degree curves are more complex to process and are numerically unstable.

- Single-segment curves are not well-suited to interactive shape design; although Bézier curves can be shaped by means of their control points and weights, the control is not sufficiently local.

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

3.3 B-spline Curves

B-Splines can be regarded as a generalisation of Bézier curves (Rogers (2001)). Similarities between both curves are that they 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.

3.3.1 Nonrational B-spline Curves

The most general *nonrational* B-Spline curves are those defined by *nonuniform* 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))

$$C(u) = \sum_{i=0}^{n} N_{i,p}(u)P_i \qquad a \le u \le b \qquad (3.3)$$

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

$$N_{i,0}(u) = \begin{cases} 1 & \text{if } u_i \le u < u_{i+1} \\ 0 & \text{otherwise} \end{cases} \qquad (3.4a)$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \qquad (3.4b)$$

The u_i are *knot* values which form the *knot vector* $U = \{u_0, u_{1,...}, u_m\}$. 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

$$n + p + 1 = m \qquad (3.5)$$

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

$$U = \left\{ \underbrace{a,...,a}_{p+1}, u_{p+1},...,u_{m-p-1}, \underbrace{b,...,b}_{p+1} \right\} \qquad (3.6)$$

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.*

Notice that for equation (3.4b), the value of any expression that has a division by 0 is taken to be 0.

To get a better understanding of these curves the following example is presented. The B-spline curve represented has a degree $p = 2$; i.e. quadratic curve, and six control points so that $n = 5$. The basis functions $N_{i,2}(u)$ are obtained following equations (3.4a) and (3.4b), $0 \le u \le 1$ and the *knot vector* is $U = \{0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1\}$. This is a *nonuniform* knot spacing, and the *knot* values correspond to the joints between the curve segments. Because of the recursive nature of the basis function equations, we must first compute the $N_{i,0}(u)$ and then the $N_{i,1}(u)$, strictly in that order and according to the knot values U, before computing the $N_{i,2}(u)$. These $N_{i,p}(u)$ functions act like switches, turning on and off the terms that they control. The basis functions $N_{i,2}(u)$ are plotted in figure 3.1.
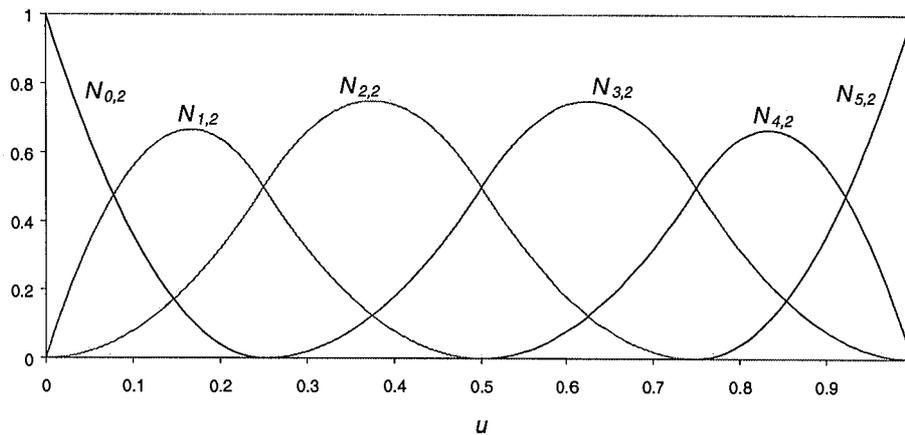
Figure 3.1: Nonuniform B-spline basis functions for $n = 5, p = 2$

Applying the basis functions $N_{i,2}(u)$ depicted in figure 3.1 to a set of six control points $\{P_0, P_1, P_2, P_3, P_4, P_5\}$ using equation (3.3) a B-spline curve $C(u)$ is obtained. The resulting curve is a composite sequence of four curve segments; i.e. $C_1(u)$, $C_2(u)$, $C_3(u)$, $C_4(u)$, connected with C^1 continuity as shown in figure 3.2. Notice that the curve passes through only the first and last points, P_0 and P_5, and it is tangent to

P_1-P_0 and P_5-P_4 at these same points P_0 and P_5. This piecewise quadratic curve $C(u)$ is tangential to each successive side of the control polygon (only for $p = 2$) at the joints between curve segments. These joints are indicated by dot marks in figure 3.2.

In figure 3.3 point P_2 defining curve $C(u)$ is moved to P'_2 causing $C'(u)$ to be the resulting curve. This local change affects only three segments of the curve $C(u)$. From the same figure 3.3, we can notice that only a maximum of three control points influence each curve segment. Conversely, a control point can influence the shape of at most only three curve segments. This can be generalised; each segment of a B-spline curve is influenced by only $p+1$ control points, and conversely, each control point influences only $p+1$ curve segments.
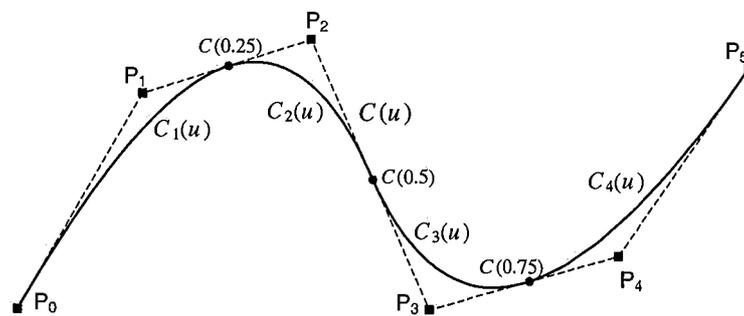
Figure 3.2: Initial B-spline curve
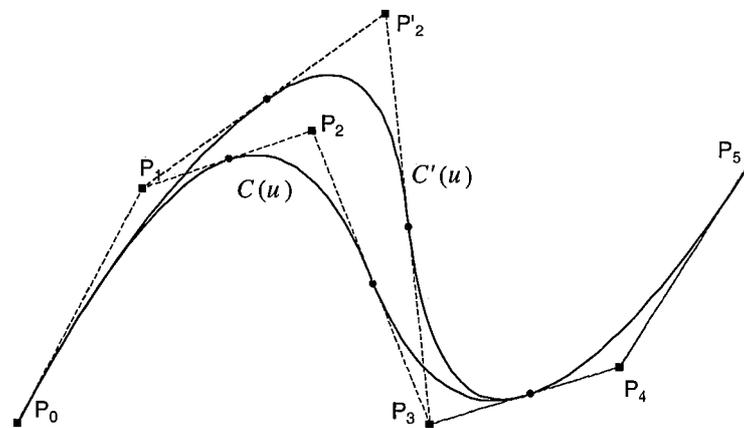
Figure 3.3: Control point P_2 moved to P'_2

Figures 3.4 (a) and 3.4 (b) show the effect on the curve of deleting and inserting, respectively, control points. In figure 3.4 (a) P_3 is removed from the control polygon,

43

thus the new B-spline $C'(u)$ consists of three curve segments. Conversely, figure 3.4 (b) shows the new curve $C'(u)$ when P' is inserted into the control polygon.

(a)   (b)
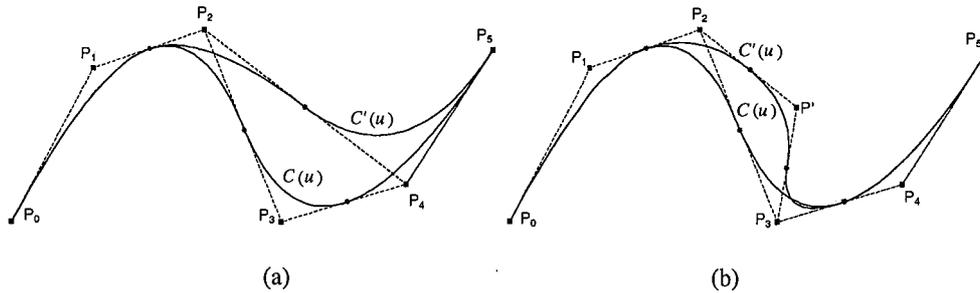
Figure 3.4: Control point deleted (a). Control point inserted (b)

The influence of p on the B-Spline shape and the independence of the degree from the number of control points is shown in figure 3.5. The three B-splines represented are defined by the same control polygon but for different values of p. Case of $p = 1$, the curve is linear and it is coincident with the control polygon. When $p = 2$, the curve is quadratic. It moves away from the intermediate control points and only passes through the end points. Indeed, the quadratic curve touches the midpoints in the intermediate segments. If p is increased to 3, the cubic curve moves further away from the control points. As a conclusion, it can be said that the lower the degree of the curve, the closer it resides to the control points.
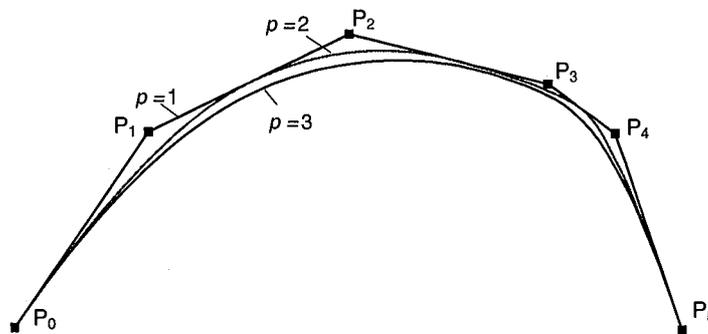
Figure 3.5: Linear ($p=1$), quadratic ($p=2$) and cubic ($p=3$) B-splines

The analytical properties of the B-spline basis functions determine the geometric characteristics of the curves. The most significant properties are (Piegl and Tiller (1997))

- Locality: $N_{i,p}(u) = 0$ if $u \notin [u_i, u_{i+p+1})$

- Nonnegativity: $N_{i,p}(u) \geq 0$

- Partition of unity: $\sum_{i=0}^{n} N_{i,p}(u) = 1$

- Differentiability: All derivatives of $N_{i,p}(u)$ exist in the interior of a knot span. At a knot $N_{i,p}(u)$ is p-k times continuously differentiable, with k the multiplicity of the knot. Thus increasing the degree increases the continuity and increasing knot multiplicity decreases continuity

- A knot vector of the form $U = \left\{ \underbrace{0,...,0}_{p+1}, \underbrace{1,...,1}_{p+1} \right\}$ yields the Bernstein polynomials of degree p

As a consequence of the properties of the basis functions, the B-spline curve exhibits the following geometric characteristics

- If $n = p$ and $U = \{0,..., 0, 1,..., 1\}$ then the B-spline curve $C(u)$ is a Bézier curve.

- Local modification scheme: If a control point P_i is moved, it will affect the curve in the interval $[u_i, u_{i+p+1})$, i.e. only in $p+1$ knot spans, following the *locality* property.

- End point interpolation: $C(0) = P_0$ and $C(1) = P_n$

- Strong convex hull property: The curve is contained in the convex hull of its control polygon; in fact if $u \in [u_i, u_{i+p+1})$ and $p \leq i \leq m - p - 1$, then $C(u)$ lies within the convex hull of the control points $P_{i-p},..., P_i$.

- Invariance under affine and perspective transformations.

- The control polygon represents a piecewise linear approximation to the curve; such approximation can be improved by knot insertion or degree elevation.

Generally, the lower the degree, the closer a B-spline curve follows its control polygon. The extreme case is p = 1, for which every point $C(u)$ is just a linear interpolation between two control points. In this case, the curve is the control polygon.

- The $N_{i,p}(u)$ functions act like switches, as u moves past a knot, one $N_{i,p}(u)$ and thus its corresponding P_i switches off, and the next one switches on.

- The continuity and differentiability of $C(u)$ follow that of the basis functions, since the curve is a linear combination of the $N_{i,p}(u)$. Thus, $C(u)$ is infinitely differentiable in the interior of knot intervals, and it is at least p-k times continuously differentiable at a knot of multiplicity k (see table 3.1). Therefore, multiple or repeated knot-vector values, or multiple coincident control points, introduce discontinuities (Rogers (2001)). For example, for a cubic curve, a double knot defines a join point with curvature discontinuity and a triple knot produces a corner point in the curve.

p-k	Continuity
0	Disjointed points
1	C^0(connected linear segments)
2	C^1(tangent continuous)
3	C^2(curvature continuous)

Table 3.1: Continuity C for different degree p and multiplicity k

3.3.2 Rational B-spline Curves

The use of rational parametric curves is advantageous in that they allow us to represent conics and circles more precisely than do nonrational parametric curves. Rational curves are defined on the basis of homogeneous coordinates. For example in 3D, homogeneous coordinates represent three-dimensional points in terms of a four-dimensional point with an additional coordinate axis w. If $P = (x, y, z)$ is a point in three-dimensional space, the corresponding point in the four-dimensional space is $P^h = (wx, wy, wz, w)$, where the homogeneous coordinate $w > 0$. Conversely, a four-

dimensional point is converted back to a three-dimensional one by dividing the three first coordinates by the fourth one.

Using homogeneous coordinates the equation (3.3) 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))

$$C(u) = \frac{\sum_{i=0}^{n} N_{i,p}(u) w_i P_i}{\sum_{i=0}^{n} N_{i,p}(u) w_i} \qquad a \leq u \leq b \qquad (3.7)$$

where P_i are the *control points*, the w_i are the so-called *weights*, and the $N_{i,p}(u)$ are the pth-degree basis functions (equations (3.4a) and (3.4b)) defined on the nonperiodic and nonuniform knot vector in equation (3.6). In most cases assuming $a = 0$, $b = 1$ and $w_i > 0$.

Setting

$$R_{i,p}(u) = \frac{N_{i,p}(u) w_i}{\sum_{j=0}^{n} N_{j,p}(u) w_j} \qquad 0 \leq u \leq 1 \qquad (3.8)$$

The equation (3.7) can be rewritten into the following equivalent form

$$C(u) = \sum_{i=0}^{n} R_{i,p}(u) P_i \qquad 0 \leq u \leq 1 \qquad (3.9)$$

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 3.6 shows a rational cubic B-spline curve and the effect that changing the single weight w_3 (associated to P_3) has on the curve. Therefore, starting from a set of weights all equal to 1; i.e. $w_i = 1$, the qualitative effect of modifying a single weight, w_3, is that if w_3 increases (decreases), then the curve $C(u)$ moves closer to (further

from) P_3, and hence the curve is pulled toward (pushed away from) P_3. Obviously, this observation applies to any weight defining the B-spline curve.
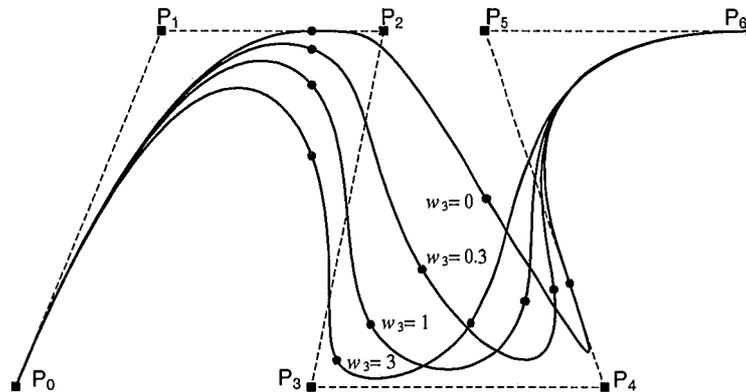
Figure 3.6: Rational cubic B-spline curves with w_3 varying

$R_{i,p}(u)$ properties which determine the geometric behaviour of curves are derived from equation (3.8) and the corresponding properties of the $N_{i,p}(u)$. The most significant properties are (Piegl and Tiller (1997))

- Nonnegativity: $R_{i,p}(u) \geq 0$

- Locality: $R_{i,p}(u) = 0$ if $u \notin [u_i, u_{i+p+1})$. In general, $R_{i-p,p}(u), \ldots, R_{i,p}(u)$ are nonzero in $[u_i, u_{i+p+1})$

- Partition of unity: $\displaystyle\sum_{i=0}^{n} R_{i,p}(u) = 1$

- $R_{0,p}(0) = R_{n,p}(1) = 1$

- Differentiability: In the interior of a knot span, the $R_{i,p}(u)$ are infinitely continuously differentiable if the denominator is different from zero. At a knot they are p-k times continuously differentiable where k is the multiplicity of the knot

- If all the weights are equal to one; i.e. $w_i = 1$ for all i, then $R_{i,p}(u) = N_{i,p}(u)$ meaning that the $N_{i,p}(u)$ are special cases of the $R_{i,p}(u)$

- $R_{i,p}(u, w_i = 0) = 0$

- $R_{i,p}(u, w_i \to +\infty) = 1$

- $R_{i,p}(u, w_j \to +\infty) = 0 \quad j \neq i$

The NURBS curve exhibits similar geometric characteristics to the nonrational B-splines curves presented in the previous section 3.3.1. Moreover, as a consequence of the rational basis functions they exhibit also the following features

- Bézier and nonrational B-spline curves are special cases of NURBS curves

- Local approximation: If a control point is moved or a weight is changed, it will affect the curve only in $p+1$ knot spans. This property is very important for interactive shape design. Using NURBS curves, we can utilise both control point movement and weight modification to attain local shape control

- If a particular weight is set to zero then the corresponding control point has no effect at all on the curve

- If $w_i \to +\infty$ then $C(u) = \begin{cases} P_i & \text{if } u \in [u_i, u_{i+p+1}) \\ C(u) & \text{otherwise} \end{cases}$

3.4 B-spline Surfaces

3.4.1 Nonrational B-spline Surfaces

The formulation of B-spline surfaces follows a similar pattern to the B-spline curves previously presented in section 3.3. Therefore, the same advantages found in B-spline curves can be extrapolated for the surfaces. Thus, adding another parametric variable v forms the general equation of a B-spline surface patch. Such surfaces are obtained by taking a bidirectional net of control points, two knot vectors and the products of the B-spline basis functions (Piegl and Tiller (1997))

$$S(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)P_{i,j} \qquad\qquad (3.10)$$

where $P_{i,j}$ are the $(n{+}1)\mathrm{x}(m{+}1)$ *control points* which are the vertices of the *control polygon*. The $N_{i,p}(u)$ and $N_{j,q}(v)$ are the *basis functions* calculated from the equations (3.4a) and (3.4b); p and q are the degrees of the basis functions polynomials in u and v, respectively. The *knot vectors* are

$$U = \left\{ \underbrace{0,...,0}_{p+1}, u_{p+1},...,u_{r-p-1}, \underbrace{1,...,1}_{p+1} \right\} \qquad\qquad (3.11a)$$

$$V = \left\{ \underbrace{0,...,0}_{q+1}, u_{q+1},...,u_{s-q-1}, \underbrace{1,...,1}_{q+1} \right\} \qquad\qquad (3.11b)$$

where the end knots are repeated with multiplicities $p{+}1$ and $q{+}1$, respectively, and $r = n{+}p{+}1$ and $s = m{+}q{+}1$.

The previous equation (3.10) is applied to display a B-spline surface having the following parameters: $p = 3$, $q = 3$, $n = 4$, $m = 4$. As shown in figure 3.7 (a), this surface is determined by set of control points, forming the control polyhedron. Figure 3.7 (b) shows the resulting nonrational B-spline surface.

(a)   (b)
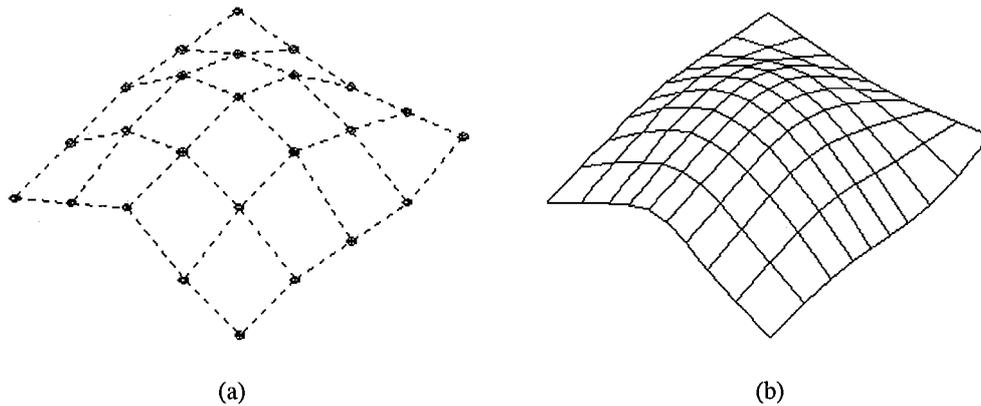
Figure 3.7: Control points (a). Nonrational B-spline surface (b); $p = 3$, $q = 3$, $n = 4$,

$$m = 4$$

The properties of the tensor product basis functions follow from the corresponding properties of the univariate basis functions listed in the previous section 3.3.1 (Piegl and Tiller (1997))

- Nonnegativity: $N_{i,p}(u), N_{j,q}(v) \geq 0$

- Partition of unity: $\sum_{i=0}^{n} \sum_{j=0}^{m} N_{i,p}(u) N_{j,q}(v) = 1$

- If $n = p$, $m = q$, $U = \{0,..., 0, 1,..., 1\}$ and $V = \{0,..., 0, 1,..., 1\}$ then B-spline functions degenerate to products of Bernstein polynomials

- Locality: $N_{i,p}(u) N_{j,q}(v) = 0$ if $u, v \notin [u_i, u_{i+p+1}) \times [v_j, u_{j+q+1})$

- In the interior to the rectangles formed by the u and v knot lines all partial derivatives of $N_{i,p}(u)$ $N_{j,k}(v)$ exist. At a u knot (v knot) it is $p-k$ ($q-k$) times continuously differentiable in the u (v) direction, where k is the multiplicity of the knot

As a consequence, the B-spline surface will exhibit the following properties

- If $n = p$, $m = q$, $U = \{0,..., 0, 1,..., 1\}$ and $V = \{0,..., 0, 1,..., 1\}$ then, the nonrational B-spline surface $S(u, v)$ is a Bézier surface

- The surface $S(u, v)$ interpolates the four corner control points: $S(0,0) = P_{0,0}$, $S(1,0) = P_{n,0}$, $S(0,1) = P_{0,m}$ and $S(1,1) = P_{n,m}$

- Affine invariance

- Strong convex hull property: If $u, v \in [u_{i0}, u_{i0+p+1}) \times [v_{j0}, v_{j0+q+1})$, then $S(u, v)$ lies within the convex *hull* of the control points $P_{i,j}$, $i_0-p \leq i \leq i_0$ and $j_0-q \leq j \leq j_0$

- The control polygon represents a piecewise planar approximation to the surface. Like the B-spline curves, the lower the degree the better the approximation

- Local modification scheme: Following the locality property, if a control point $P_{i,j}$ is moved, it will affect the surface only in the rectangle $[u_i, u_{i+p+1}) \times [v_j, v_{j+q+1})$.

- The continuity and differentiability of $S(u, v)$ follows the same property for the basis functions. Thus, $S(u, v)$ is $p-k$ $(q-k)$ times differentiable in the u (v) direction at a u (v) knot of multiplicity k.

3.4.2 Rational B-spline Surfaces

A NURBS surface of degree p in the u direction and degree q in the v direction is a vector piecewise rational function of the form (Piegl and Tiller (1997)).

$$S(u,v) = \frac{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)w_{i,j}P_{i,j}}{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)w_{i,j}} \qquad 0 \leq u,v \leq 1 \qquad (3.12)$$

This surface is the rational generalisation of the tensor product nonrational B-spline surface (equation (3.10)), where $P_{i,j}$ form a control point net, the $w_{i,j}$ are the weights, and the $N_{i,p}(u)$ and $N_{j,q}(v)$ are the pth-degree and q^{th}-degree basis functions in the u and v directions defined over the knot vectors equations (3.11a) and (3.11b).

The piecewise rational basis functions are defined as

$$R_{i,j}(u,v) = \frac{N_{i,p}(u)N_{j,q}(v)w_{i,j}}{\sum_{k=0}^{n}\sum_{l=0}^{m} N_{k,p}(u)N_{l,q}(v)w_{k,l}} \qquad (3.13)$$

Therefore, the surface equation (3.12) can be written as

$$S(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} R_{i,j}(u,v)P_{i,j} \qquad (3.14)$$

Examples of NURBS surfaces are shown in figures 3.8 (b) and 3.8 (c). Figure 3.8 (a) shows the corresponding control polygon (dashed lines) with the control points

(dots). Each control point has a weight w associated and, depending on its value it is identified as w_i or w_j. Figures 3.8 (b) and 3.8 (c) represent the resultant surfaces for such a polygon and parameters $p = 3$, $q = 3$, $n = 3$, $m = 4$. The geometrical differences between both surfaces, i.e. fig 3.8 (b) and (c), are as a result of the values of the weights. In figure 3.8 (b) all the weights are set to 1 ($w_i = w_j = 1$) whereas in figure 3.8 (c) weights have different values ($w_i = 1$, $w_j = 0.5$).

(a)　　　　　　(b) $w_i = w_j = 1$　　　(c) $w_i = 1$, $w_j = 0.5$
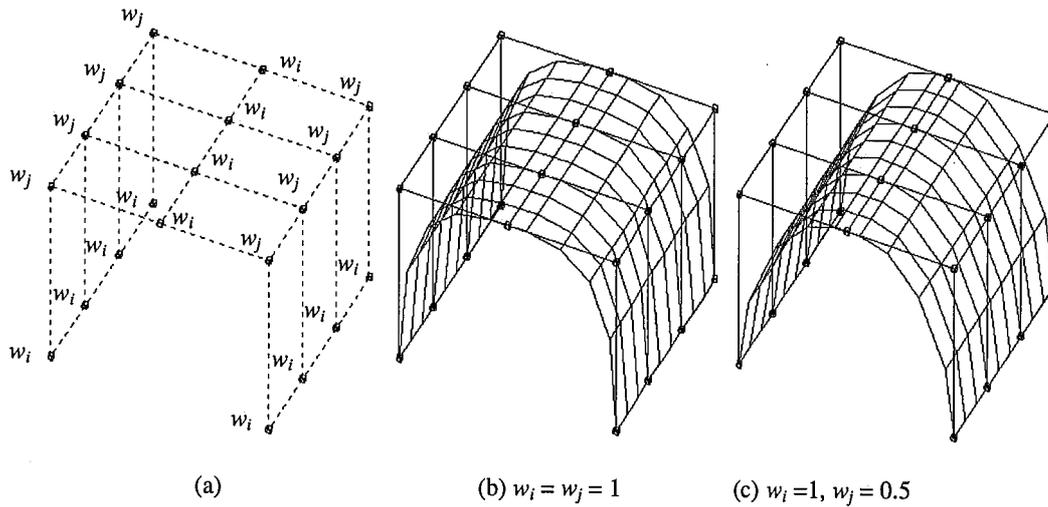
Figure 3.8: Control points (a). NURBS surfaces (b) (c); $p = 3$, $q = 3$, $n = 3$, $m = 4$

The properties of the rational functions $R_{i,j}(u, v)$ are roughly the same as those given for nonrational basis functions $N_{i,p}(u)$ $N_{j,q}(v)$ in the previous section 3.4.1 but additionally (Piegl and Tiller (1997))

- If all $w_{i,j} = a$ for $0 \leq i \leq n$, $0 \leq j \leq m$, and $a \neq 0$, then $R_{i,j}(u, v) = N_{i,p}(u)$ $N_{j,q}(v)$

The properties of the rational B-spline surfaces are similar to the ones presented for the nonrational B-spline ones, moreover there are some properties as a result of using the homogeneous coordinate $w_{i,j}$

- Nonrational B-spline and Bézier surfaces are special cases of NURBS surfaces

- Local modification scheme: If a control point $P_{i,j}$ is moved, or $w_{i,j}$ is changed, it will affect the surface only in the rectangle $[u_i, u_{i+p+1}) \times [v_j, v_{j+q+1})$. Thus, we can use both control point movement and weight modification to locally change the shape of NURBS surfaces

3.5 Discussion and Conclusions

The boundary representation presented in this chapter has begun with Bézier curves, giving a brief description of their mathematical definition and properties. Then, the analytical description of nonrational B-splines curves has been introduced and depicted with some examples. This general formulation has also been applied to the rational B-spline curves. Finally, the mathematical expressions for curves have been extended to nonrational and rational B-spline surfaces, respectively, complemented with some graphical illustrations.

To conclude, the main properties and features of NURBS are reviewed. They are efficient tools used in the graphics and CAD industry especially as trimmed surfaces. Reasons why NURBS are suitable graphic tools and hence of special interest in this work are summarised as follows

- They offer a common mathematical form for precisely representing and designing both standard analytic shapes (conics, surface of revolution, etc) and free-form curves and surfaces

- NURBS provide the flexibility to describe a large variety of shapes by manipulating the control points and the weights. Moreover, they present a powerful geometric tool kit allowing operations, such as insertion, refinement and removal of knots, degree elevation, splitting, etc. These features are found especially attractive for design optimisation purposes

- Evaluation is reasonably fast and computationally stable (Piegl and Tiller (1997)).

- NURBS are invariant under scaling, rotation and translation as well as parallel and perspective projection (affine invariance).

- NURBS are generalisations of nonrational B-spline forms as well as rational and nonrational Bézier curves and surfaces.

- The geometry of the NURBS is governed locally by the nearest control points. Movement, removal or addition of more remote control points has no effect.

However, NURBS have several drawbacks

- Extra storage is needed to define traditional curves and surfaces. For example, to represent a circle using a circumscribing square requires seven control points with their corresponding weights and ten knots (cubic B-spline). Traditional representation requires the centre, the radius and the plane of the circle.

- Improper application of the weights can result in a bad parameterisation resulting in distorted surface constructions.

4

STRUCTURAL ANALYSIS.
THE BOUNDARY ELEMENT METHOD

4.1 Overview

Computational methods for structural analysis such as the *finite element method* (FEM) and the *boundary element method* (BEM) have become essential tools in engineering. The FEM is a well-established analytical tool used in a very wide range of problems. However, the BEM's applicability is becoming an effective alternative to FEM in some areas of engineering analysis.

Although the first ideas related to the *boundary element method* were developed in the early 1900's, the method, like most computational ones, only emerged when modern computers became available. The reason is that, except for very simple problems, the number of computations involved is too great for analytical calculation.

The first part of the present chapter presents a brief description of the FEM followed a more detailed theoretical overview of the BEM. The BEM explanation is focused on its applications to elastostatic problems since these are the type of problems to be tackled in this thesis. However, it should be noted that the BEM is applicable to many different fields, e.g. fracture mechanics, corrosion protection, waves, thermal problems, etc. Advantages and disadvantages of the BEM are also discussed.

56

4.2 Introduction

There are three main computational techniques for the approximation of the solution of problems in continuum mechanics, i.e. the *finite difference* (FDM), the *finite element* (FEM) and the *boundary element method* (BEM). See figure 4.1 for a depiction of the three methods. In all these numerical techniques the governing equations are replaced with approximate algebraic relationships. The geometry is described by a grid (*mesh*) and the solution of a set of simultaneous algebraic equations is calculated to determine the unknown response distribution on the grid.

However these methods have some fundamental differences. The FDM operates directly on the governing differential equations. It uses a simple approximation over local cells to generate a system of algebraic equations. In contrast, FEM and BEM transform the governing equations into equivalent integral equations. The FEM operates on the equivalent governing integral using element interpolations of node-point response quantities. Unlike the FEM, the BEM uses element interpolations of boundary node point response quantities. Therefore, FEM requires the volume of the object to be divided into elements but BEM requires only the boundary of the object to be divided.

(a)   (b)   (c)
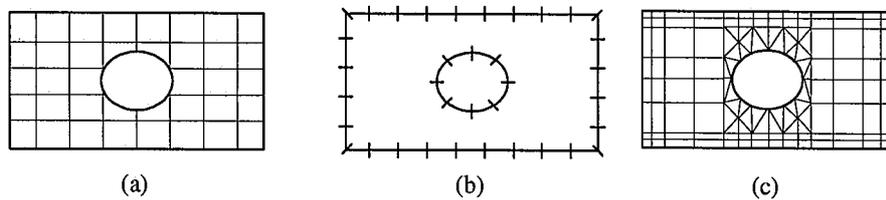
Figure 4.1: Finite difference (a), boundary element (b), finite element mesh (c)

In the following section, an abridged explanation of FEM is presented. The scope of this section is to discuss the reasons why the BEM is chosen as an alternative tool for the structural analysis.

4.3   The Finite Element Method

As previously stated, it is not the aim of this section to develop the formulation on which FEM is based on but to present the principal aspects concerning this theory. For further information regarding the FE formulation refer to Zienkiewicz (1989) and Bathe (1982). The essence of *finite elements* is to divide a continuous body into a set of elements which are joined to each other at their node points. The system of external loads acting on the actual solid is replaced by an equivalent system of forces acting at the nodes. The overall finite element equation of equilibrium can be written as (Zienkiewicz (1989))

$$\mathbf{Ku} = \mathbf{f} \qquad (4.1)$$

where $\mathbf{K}$ is the global *stiffness matrix*, $\mathbf{u}$ is the global displacement vector and $\mathbf{f}$ is the nodal vector of forces. The forces in the overall system $\mathbf{f}$ are obtained by adding all the forces at each node, and $\mathbf{K}$ is obtained by assembling the *stiffness matrix* of each element $\mathbf{K_e}$ (Zienkiewicz (1989))

$$\mathbf{K_e} = \int_\Omega \mathbf{B^T D B} d\Omega \qquad (4.2)$$

where $\mathbf{D}$ ($\sigma = \mathbf{D}\varepsilon$) and $\mathbf{B}$ ($\varepsilon = \mathbf{Bu}$) denote the *elasticity* and *strain* matrices, respectively.

Finally, once the nodal displacements $\mathbf{u}$ are known from equation (4.1), the stresses in each element σ_e are calculated from the displacements, $\mathbf{u_e}$, over the element using

$$\sigma_e = \mathbf{DBu_e} \qquad (4.3)$$

Within each element, the variation of displacement is assumed to be determined by nodal displacements and polynomial *shape functions* (Zienkiewicz (1989)). Figure 4.2 shows a typical two-dimensional *triangular* element with three nodes identified as 1, 2 and 3, respectively. Every node has a force and displacement associated for each degree of freedom, e.g. f_{x1}, u_{x1} for node 1 in the x direction.
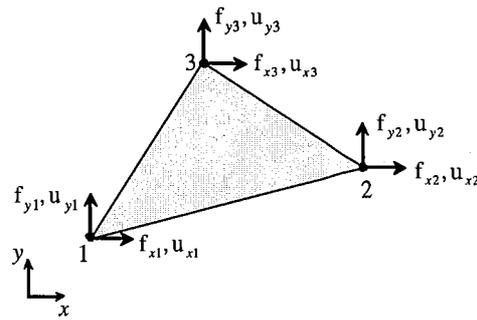
Figure 4.2: Two-dimensional triangular element

According to their shape, elements can be of different types such as *quadrilateral* or *triangular* (in 2D) and *hexahedral* or *tetrahedral* (in 3D). The *degree of the element*, i.e. *linear, quadratic, cubic*, is determined by the number of nodes per element, for example *linear* elements have one node placed in each vertex, *quadratic* have one node in each vertex but also one additional node in the middle of each edge, etc. Thus, the element in figure 4.2 is a two-dimensional *triangular linear* element.

The type and number of elements used can be decided by the analyst. In general, the accuracy of the solution increases with the number of elements and also with the degree of the element. However, the computer time and cost also increase with the number of elements and nodes. Thus, it is sensible to use a dense concentration of elements (and/or higher order elements) in the critical areas of the model while a coarse mesh (and/or lower order elements) in regions of less interest. Moreover, there are modern procedures, *error estimators*, for estimating the error arising from a given density of elements and adaptively refining the mesh or increasing the degree of the element to ensure the accuracy of the results.

In spite of the general applicability of FEM to engineering problems, they have some drawbacks such as that models can be difficult and time consuming to build, check, and also to change. Moreover, it may not be simple to reach the required accuracy, particularly for problems involving stress concentrations.

In a continuum structure discretised using low-order finite elements, the *shape functions* can become discontinuous across the element boundaries which in turn produces discontinuity of the solution between adjacent elements. Extrapolating this

problem to FE-based optimisation, it leads to meshes with alternating void and solid elements displayed in a checkerboard fashion (*checkerboard pattern*).

The BEM addresses the above drawbacks of FE modelling since it is very easy and quick to build a model as well as to change a model to reflect design changes or to try different design options. Finally, it is highly accurate, since it makes approximations only on the surface area of the component instead of throughout its entire volume.

4.4 The Boundary Element Method. A Brief Overview

The BEM is a computational technique in which the governing equations are transformed into equivalent integral equations. Using relationships from vector calculus related to the *Gauss-Green* or *divergence* theorem, these integral equations, involving both volume and surface integrals, are then transformed into integral equations (*boundary integral equations*) that contain no volume integrals involving the unknown response. This last transformation involves certain known solutions (*fundamental solutions*) to the original differential equation. These *fundamental solutions* generally describe the response of an infinite medium to a point excitation. They are singular functions, and the strength of the singularity varies with the problem type, being $1/r$, $1/r^2$, $\ln(1/r)$, etc. For non-linear problems or with variable material properties the derivation of these solutions can be very complex which limits the BEM versatility.

The *boundary integral equations* are approximated by a set of discretised integral equations. The boundary is divided into elements, and for each element there are some nodes associated. The response inside each boundary element is given by the node-point response and simple interpolation formulae.

The interior is continuous since no discretisation of the interior is needed; this leads to a high resolution of interior stresses and displacements. Internal point solutions may be calculated after the boundary unknowns are obtained. In general, the density and location of the internal points have no relation to the boundary mesh or boundary

unknowns. However, there is a minimum distance to be maintained between the internal points and the boundary for numerical reasons, explained in more detail in section 4.9

One of the main features of BEM is this reduction in the dimensionality of the problem. That is, for two-dimensional problems only the line boundary is discretised, and for three-dimensional problems only the surface is discretised. This results in a reduction in modelling effort compared with other computational methods such as the FEM. Moreover, other features of this method will be introduced later in this chapter.

4.5 Evolution of Boundary Element Methods

The fundamentals of the boundary element methods can be found in the classical formulations by Somigliana and Betti, for elasticity, and by Fredholm, for potential problems. These classical formulations were followed in the 1950's by Muskhelishvili (1953) and Mikhlin (1957).

In the early sixties, with the introduction of high-speed computers, the *boundary element* method (BEM) emerged becoming a more generally applicable technique in the 1970's. The major breakthrough came when Jaswon and Symm (1977) discretised the integral equations of two-dimensional potential problems governed by Laplace's equation into straight-line elements. Also in this period, Rizzo (1967) was the first to use displacements and tractions in an integral equation applicable over the boundary. The extension of this approach to 3D problems was presented by Cruse (1969).

Later in the seventies Lachat (1976) contributed towards BEM becoming an effective-numerical technique. He introduced to BEM the concept of *higher-order elements* using *quadratic shape functions*. Therefore, at this stage many of the algorithms and numerical modelling methods used in FE formulations also were also being implemented in the BE formulations.

Since then, the *boundary integral equation* approach has continued to develop being extended in a very wide range of continuum mechanics as Aliabadi (2002), Brebbia (1989), Becker (1992), Kane (1994) and Trevelyan (1994) present on the many contributions to the modern boundary element method.

Boundary element formulations may be divided into two different but closely related categories. The first, and most popular, is the so-called *direct formulation*, in which the unknown functions appearing in the formulation are actual physical variables of the problems. In elasticity these unknown functions are the displacement and traction fields. The other approach is called the *indirect formulation*, in which fictitious source densities represent unknown functions. Once these source densities are found, the values of physical parameters can be obtained by simple integrations. Since this research is focused on the *direct formulation*, the following sections are explained according to this category.

4.6 Fundamentals of Elasticity

In this section, basic ideas and relationships are presented from the theory of elasticity that are necessary for developing the *direct formulation* of the BEM.

4.6.1 Stress and Tractions

Stress is defined as the average force per unit area. The stresses can be visualised by cutting a solid on a particular plane, as shown in figure 4.3 for an infinitesimally small cube. Three stress components are defined normal to the cutting planes σ_{xx}, σ_{yy}, σ_{zz} (*normal stress*) and three tangential to the cutting planes σ_{xy}, σ_{xz}, σ_{yz} (*tangential or shear stress*). The stress tensor has two subscripts. The first refers to the direction of the outward normal to the plane on which it acts and the second one to the direction of the stress. Note that the *shear stresses* according to the notation must be equal (or *complementary*), i.e. $\sigma_{xy} = \sigma_{yx}$. Therefore, the subscripts can be interchangeable for all practical purposes. For *plane stress* problems, such as thin

62

plates subject to in-plane loading, all stresses associated with the z direction are assumed to be zero, i.e. $\sigma_{zz} = \sigma_{xz} = \sigma_{yz} = 0$.

Another way of defining stresses is as distributed forces in any direction per unit area acting on a plane section. Stresses such defined are known as *boundary stresses* or *tractions*. The *traction* vector is a stress vector, which has a magnitude and direction. On the other hand, *stresses* are a tensor and cannot be resolved in any direction.
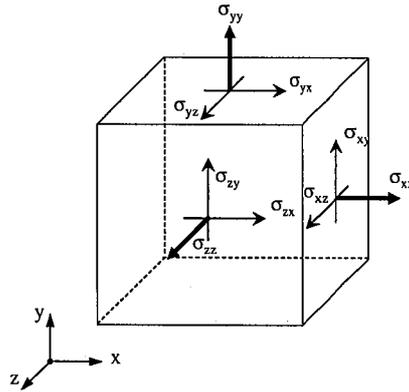
Figure 4.3: Stresses acting on an infinitesimal cube

For three dimensional problems the relationship between *tractions* and *stresses* is given by

$$t_x = \sigma_{xx} n_x + \sigma_{xy} n_y + \sigma_{xz} n_z$$
$$t_y = \sigma_{yx} n_x + \sigma_{yy} n_y + \sigma_{yz} n_z \qquad (4.4)$$
$$t_z = \sigma_{zx} n_x + \sigma_{zy} n_y + \sigma_{zz} n_z$$

where n_x, n_y and n_z are the x, y and z components of the unit vector normal to the boundary. The relationship between *tractions* and *stresses* can be summarised in the tensor notation as

$$t_i = \sigma_{ij} n_j \qquad (4.5)$$

assuming the *Einstein summation convention*[*]

4.6.2 Equilibrium

The *equations of equilibrium* for a three-dimensional system subjected to external forces and body forces b are given by

$$\frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} + \frac{\partial \sigma_{xz}}{\partial z} + b_x = 0$$

$$\frac{\partial \sigma_{yx}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} + \frac{\partial \sigma_{yz}}{\partial z} + b_y = 0 \qquad (4.6)$$

$$\frac{\partial \sigma_{zx}}{\partial x} + \frac{\partial \sigma_{zy}}{\partial y} + \frac{\partial \sigma_{zz}}{\partial z} + b_z = 0$$

where b_x, b_y, b_z are the components of the body force vector. In a tensor notation

$$\sigma_{ij,j} + b_i = 0 \qquad (4.7)$$

where the use of a comma denotes differentiation with respect to the direction associated with the index following the comma.

4.6.3 Strain

The external and internal forces result in linear and angular displacements in a deformable body. These displacements can be defined in terms of *strains*, in tensor notation as

$$\varepsilon_{ij} = \frac{1}{2}\left(u_{i,j} + u_{j,i}\right) \qquad (4.8)$$

---

[*] $a_i b_i \equiv \sum_i a_i b_i$

A state of *plane strain* can be assumed if the solid extends a long distance in the z-direction, the loading is uniform in this direction and $u_z = 0$ everywhere. In such a situation, certain shear strain components are taken to be zero $\varepsilon_{xz} = \varepsilon_{yz} = 0$.

4.6.4 Compatibility Equations

Differentiating (4.8) with respect to the *Cartesian coordinates* we obtain the strain *compatibility equations* which can be written in tensor notation as

$$-2\frac{\partial^2 \varepsilon_{ij}}{\partial x_i \partial x_j} + \frac{\partial^2 \varepsilon_{ii}}{\partial x_j^2} + \frac{\partial^2 \varepsilon_{jj}}{\partial x_i^2} = 0$$

$$\frac{\partial^2 \varepsilon_{ij}}{\partial x_i \partial x_k} - \frac{\partial}{\partial x_i}\left(-\frac{\partial \varepsilon_{jk}}{\partial x_i} + \frac{\partial \varepsilon_{ik}}{\partial x_j}\frac{\partial \varepsilon_{ij}}{\partial x_k}\right) = 0$$

(4.9)

where $i, j, k = 1, 2, 3$ and $i \neq j \neq k$

4.6.5 Stress-Strain Relationship

Consider a linear elastic state for a homogeneous material, i.e. with the same properties at all points and whose properties at all points are the same in all directions (isotropic). *Hooke's law* can be written as

$$\begin{aligned}
\sigma_{xx} &= \lambda e + 2\mu\varepsilon_{xx} & \sigma_{xy} &= \mu\varepsilon_{xy} \\
\sigma_{yy} &= \lambda e + 2\mu\varepsilon_{yy} & \sigma_{yz} &= \mu\varepsilon_{yz} \\
\sigma_{zz} &= \lambda e + 2\mu\varepsilon_{zz} & \sigma_{xz} &= \mu\varepsilon_{xz}
\end{aligned}$$

(4.10)

where $\lambda = 2v\mu/(1-2v)$ is the *Lamé constant*, $\mu = E/2(1+v)$ is the *shear modulus* of elasticity and $e = \varepsilon_{xx} + \varepsilon_{yy} + \varepsilon_{zz}$ is the *volumetric strain*. The *stress-strain equations* of elasticity can be written in tensor notation as

$$\sigma_{ij} = \delta_{ij}(\lambda\varepsilon_{kk} + \mu\varepsilon_{ij}) + \mu\varepsilon_{ij}$$

(4.11)

where δ_{ij} is the *Kronecker delta* whose properties are

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \tag{4.12}$$

4.6.6 Governing Equations of Elasticity

The governing equations of elasticity, known as *Navier's equations*, are the conditions of equilibrium expressed in terms of displacements. They can be obtained by substituting the *stress-strain relationship* (4.11) into *equations of equilibrium* (4.6) using the *strain-displacement relationships* (4.8), to give

$$\mu \nabla^2 u_x + (\lambda + \mu)\frac{\partial e}{\partial x} + b_x = 0$$

$$\mu \nabla^2 u_y + (\lambda + \mu)\frac{\partial e}{\partial y} + b_y = 0 \tag{4.13}$$

$$\mu \nabla^2 u_z + (\lambda + \mu)\frac{\partial e}{\partial z} + b_z = 0$$

where $\nabla^2 = \left(\dfrac{\partial^2}{\partial x^2} + \dfrac{\partial^2}{\partial y^2} + \dfrac{\partial^2}{\partial z^2} \right)$ is the *Laplacian operator*. Thus, the *Navier equation* can be written in tensor notation as

$$\mu u_{i,jj} + (\mu + \lambda)u_{j,ji} + b_i = 0 \qquad \text{for } i, j = 1, 2, 3 \tag{4.14}$$

4.7 Boundary Element Formulation

The formulation of BEM, here presented based mainly in Aliabadi (2002) notation, is based on the idea that only functions satisfying the differential equations exactly should be used to approximate the solution inside the domain. If we use these functions, we only need to approximate the actual boundary conditions. Therefore, the solutions inside the domain satisfy the differential equations exactly. Moreover, since the functions are defined globally, there is no need to subdivide the domain into elements. On the contrary, these *fundamental solutions* must be available and sometimes their singular nature leads to difficulties with integration.

4.7.1 Betti's Reciprocal Theorem

From the *equilibrium equations* (4.7) it is possible to write the following weighted residual statement

$$\int_{\Omega} (\sigma_{ij,j} + b_i) u_i^* d\Omega = 0 \qquad (4.15)$$

where Ω denotes the domain with boundary Γ of the problem. In the conventional direct collocation approach, the weighting function u_i^* is taken to be the displacement field associated with a point force applied at a *collocation point*. The stresses, body forces and displacements are a function of p, which is the collocation point, $p \in \Omega$ (p $\equiv x, y, z$) for a three-dimensional body. The integral involving the $\sigma_{ij,j} u_i^*$ term in (4.15) can be rewritten as

$$\int_{\Omega} \sigma_{ij,j} u_i^* d\Omega = \int_{\Omega} (\sigma_{ij} u_i^*)_{,j} d\Omega - \int_{\Omega} \sigma_{ij} \varepsilon_{ij}^* d\Omega \qquad (4.16)$$

since

$$\sigma_{ij} \varepsilon_{ij}^* = \frac{1}{2} (\sigma_{ij} u_{i,j}^* + \sigma_{ij} u_{j,i}^*) = \frac{1}{2} (\sigma_{ij} u_{i,j}^* + \sigma_{ji} u_{j,i}^*) = \sigma_{ij} u_{i,j}^* \qquad (4.17)$$

where ε_{ij}^* is the strain field associated with displacements u^*.

To transform the previous equation (4.16) to involve only values at the boundary, it is necessary to introduce here the concept of the *divergence theorem*, also known as *Green's theorem*. This is a fundamental identity that relates a volume integral to a surface integral. The *divergence theorem*, in tensor notation and assuming the *Einstein summation convention*, is as follows

$$\int_{\Omega} f_{i,i} d\Omega = \int_{\Gamma} f_i n_i d\Gamma \qquad (4.18)$$

67

where Ω is the volume enclosed by a piecewise smooth surface Γ, f is a function with continuous first derivatives with respect to the *Cartesian coordinates*, and n is the unit outward normal vector.

Therefore, applying the *divergence theorem* (4.18) to the first integral in (4.16)

$$\int_{\Omega}(\sigma_{ij}u_i^*)_{,j}\,d\Omega \;=\; \int_{\Gamma}\sigma_{ij}n_j u_i^*\,d\Gamma = \int_{\Gamma}t_i u_i^*\,d\Gamma \qquad (4.19)$$

thus

$$\int_{\Omega}\sigma_{ij,j}u_i^*\,d\Omega \;=\; \int_{\Gamma}t_i u_i^*\,d\Gamma - \int_{\Omega}\sigma_{ij}\varepsilon_{ij}^*\,d\Omega \qquad (4.20)$$

Substituting (4.20) in (4.15)

$$\int_{\Gamma}t_i u_i^*\,d\Gamma + \int_{\Omega}b_i u_i^*\,d\Omega = \int_{\Omega}\sigma_{ij}\varepsilon_{ij}^*\,d\Omega \qquad (4.21)$$

The integral on the right-hand side of (4.21) can be written as

$$\int_{\Omega}\sigma_{ij}\varepsilon_{ij}^*\,d\Omega = \int_{\Omega}\left[\lambda\delta_{ij}\varepsilon_{ij}^*\varepsilon_{kk} + 2\mu\varepsilon_{ij}\varepsilon_{ij}^*\right]\,d\Omega \qquad (4.22)$$

Since $\delta_{ij}\varepsilon_{ij}^* = \varepsilon_{mm}^*$ and $\varepsilon_{kk} = \delta_{ij}\varepsilon_{ij}$

$$\int_{\Omega}\sigma_{ij}\varepsilon_{ij}^*\,d\Omega = \int_{\Omega}\left[\lambda\delta_{ij}\varepsilon_{mm}^* + 2\mu\varepsilon_{ij}^*\right]\varepsilon_{ij}\,d\Omega = \int_{\Omega}\sigma_{ij}^*\varepsilon_{ij}\,d\Omega \qquad (4.23)$$

Finally, we get the following expression known as *Betti's reciprocal work theorem*

$$\int_{\Gamma}t_i u_i^*\,d\Gamma + \int_{\Omega}b_i u_i^*\,d\Omega = \int_{\Gamma}t_i^* u_i\,d\Gamma + \int_{\Omega}b_i^* u_i\,d\Omega \qquad (4.24)$$

4.7.2 The Boundary Integral Equation

The aim of using this mathematics is to derive a form of the BIE which does not include any volume integrals. For this reason the concept of the *Dirac delta function* is introduced at this stage. Thus, the *boundary integral equation* for elastostatic

problems can be derived from *Betti's reciprocal work theorem* (4.24) by taking the fictitious body force b^*_i to correspond to a point force. This force can be represented by the *Dirac delta function*. The main feature of this function is that it is zero at all points x except at x = X, where it becomes infinity. Thus it represents a point singularity at the source point X.

$$\Delta(X, x) = \begin{cases} \infty & \text{for } x = X \\ 0 & \text{otherwise} \end{cases} \tag{4.25}$$

Therefore, by using the *Dirac delta function*, the body force b^*_i can be represented as

$$b^*_i = \Delta(X, x) e_i \tag{4.26}$$

where the unit vector component e_i corresponds to a unit positive force in the i direction applied at X, with $X \in \Omega$. In two-dimensional problems, e_i is a force per unit thickness, and in three-dimensional problems it is a pure concentrated force. The *Dirac delta function* has the property

$$\int_a^b \Delta(X, x) f(x) dx = f(X) \tag{4.27}$$

where $-\infty \le a, \ b \le \infty$ and $a < X < b$

Using this property, the last integral in the equation (4.24) can be written for a source point p as

$$\int_\Omega b^*_i u_i d\Omega = \int_\Omega \Delta(p, x) e_i u_i d\Omega = u_i(p) e_i \tag{4.28}$$

The displacement and traction fields corresponding to the point force solution can be written as

$$u^*_i = U_{ij}(p, Q) e_j \tag{4.29}$$

$$t^*_i = T_{ij}(p, Q) e_j \tag{4.30}$$

where U_{ij} and T_{ij} are the *fundamental solutions* or also known as *kernel functions* (to be presented in section 4.7.3), Q is any point on Γ, p is a point inside the solution domain Ω (not on the boundary).

Using (4.28) to (4.30) the equation (4.24) can be written as

$$u_i(\mathrm{p}) = \int_\Gamma U_{ij}(\mathrm{p},\mathrm{Q})t_j(\mathrm{Q})d\Gamma - \int_\Gamma T_{ij}(\mathrm{p},\mathrm{Q})u_j(\mathrm{Q})d\Gamma + \int_\Omega U_{ij}(\mathrm{p},\mathrm{q})b_j(\mathrm{q})d\Omega \qquad (4.31)$$

where p and P refer to the source point and q and Q refer to the field point. Lower case notation stands for points inside the domain, i.e. p and q, whereas upper case, i.e. P and Q, for points on the boundary. The equation (4.31) is known as *Somigliana's identity for displacements*. It relates the value of displacements at a source point p to boundary values of the displacements and tractions. This is an important step since only the term $u_i(\mathrm{p})$ and the integral containing the body forces $b_i(\mathrm{q})$ remain related to Ω.

4.7.3 Fundamental Solutions

To calculate the displacement field u^*_i the *Navier equations* (4.14) are used. They can now be written for a unit point force applied to the body at a point p, as

$$\mu u^*_{i,jj} + \frac{\mu}{1-2v}u^*_{j,ji} + \Delta(\mathrm{p},\mathrm{Q})e_i = 0 \qquad (4.32)$$

The solutions of the governing equations due to a point force are commonly referred to as *fundamental solutions*. The *Galerkin vector approach* is used to evaluate *the fundamental solutions* due to a unit point force in an infinite medium. The displacements are expressed in terms of the *Galerkin vector* as

$$u^*_i = G_{i,kk} - \frac{1}{2(1-v)}G_{k,ik} \qquad (4.33)$$

Substituting into equation (4.32)

$$\mu G_{i,kkjj} + \Delta(\mathrm{p},\mathrm{Q})e_i = 0 \qquad (4.34)$$

Since $G_{k,ikjj} = G_{k,jjki}$, $G_{j,kkjj} = G_{k,jjki}$ and $G_{k,jkij} = G_{k,jjki}$ the previous equation can also be written as

$$\mu \nabla^2 (\nabla^2 G_i) + \Delta(p,Q) e_i = 0 \tag{4.35}$$

If $F_i = \nabla^2 G_i$ then the previous equation (4.35) is transformed to

$$\nabla^2 F_i + \frac{1}{\mu} \Delta(p,Q) e_i = 0 \tag{4.36}$$

Three-Dimensional Problems

For three-dimensional problems the solution to the above problem is the well-known solution of a point force in an infinite medium, called the *Kelvin solution*.

$$F_i = \frac{1}{4\pi\mu} \left(\frac{1}{r} \right) e_i \tag{4.37}$$

where r is the distance between the points p and Q.

The corresponding *Galerkin vector* is

$$G_i = \frac{1}{8\pi\mu} r e_i \tag{4.38}$$

Substituting the derivatives of (4.38) into (4.33) gives

$$u_i^* = \frac{1}{8\pi\mu} \left[r_{,kk} e_i - \frac{1}{2(1-v)} r_{,ik} e_k \right] \tag{4.39}$$

and rewriting (4.39)

$$u_i^* = \frac{1}{16\pi\mu(1-v)} \left(\frac{1}{r} \right) \left[(3-4v)\delta_{ij} + r_{,i} r_{,j} \right] e_j \tag{4.40}$$

thus, the *displacement fundamental solution* is obtained from (4.40) and (4.29)

$$U_{ij}(p,Q) = \frac{1}{16\pi\mu(1-v)}\left(\frac{1}{r}\right)\left[(3-4v)\delta_{ij} + r_{,i}r_{,j}\right] \tag{4.41}$$

where $U_{ij}(p,Q)$ represents the displacement in the j direction at point Q due to a unit point force acting in the i direction at p.

The *traction fundamental solution* is obtained from (4.40), through the *displacement-strain* and *strain-stress relationships* and $t_i^* = \sigma_{ij}^* n_j$

$$t_i^* = \frac{-1}{8\pi(1-v)}\left(\frac{1}{r^2}\right)\left\{\frac{\partial r}{\partial n}\left[(1-2v)\delta_{ij} + 3r_{,i}r_{,j}\right] - (1-2v)\left(r_{,i}n_j - r_{,j}n_i\right)\right\}e_j \tag{4.42}$$

where n_j denotes the components of the outward normal at the field point Q, from (4.30)

$$T_{ij}(p,Q) = \frac{-1}{8\pi(1-v)}\left(\frac{1}{r^2}\right)\left\{\frac{\partial r}{\partial n}\left[(1-2v)\delta_{ij} + 3r_{,i}r_{,j}\right] - (1-2v)\left(r_{,i}n_j - r_{,j}n_i\right)\right\} \tag{4.43}$$

where $T_{ij}(p,Q)$ represents the traction in the j direction at point Q due to a unit point force acting in the i direction at p.

Two-Dimensional Problems

For two-dimensional problems a similar procedure is used to determine the *fundamental solutions*.

$$F_i = \frac{-1}{2\pi\mu}\ln(r)e_i \tag{4.44}$$

The corresponding *Galerkin vector* is

$$G_i = \frac{-1}{8\pi\mu}r^2 \ln(r)e_i \tag{4.45}$$

Following the steps detailed in 3D problems, the *displacement* and *traction fundamental solutions* in the two-dimensional plane strain condition are given by

$$U_{ij}(p,Q) = \frac{1}{8\pi\mu(1-v)}\left[(3-4v)\ln\left(\frac{1}{r}\right)\delta_{ij} + r_{,i}r_{,j}\right]$$ (4.46)

$$T_{ij}(p,Q) = \frac{-1}{4\pi(1-v)}\left(\frac{1}{r}\right)\left\{\frac{\partial r}{\partial n}\left[(1-2v)\delta_{ij} + 2r_{,i}r_{,j}\right] - (1-2v)(r_{,i}n_j - r_{,j}n_i)\right\}$$ (4.47)

Once the formulation of the *fundamental solutions* has been carefully explained, there are some comments to be made related to the singular nature of these functions. Having noticed that $U_{ij}(p,Q)$ in 2D is a function of $\ln(1/r)$ and a function of $1/r$ in 3D, thus as r becomes very large, i.e. the source point p and the point at which we are finding the displacement Q are very far from each other, in 3D cases $U_{ij}(p,Q)$ tends to zero, but for 2D problems the $\ln(1/r)$ term tends to $-\infty$. This means that if the analysis is performed on an infinite domain problem it will assume a far-field displacement of $-\infty$.

On the other hand if r becomes very small, i.e. as the field point Q approaches the fictitious point force at p, the value of $U_{ij}(p,Q)$ becomes very large, even worse in 3D for $T_{ij}(p,Q)$ since it is a function of $1/r^2$. Thus, the *fundamental solutions* head towards infinity rapidly as the field point approaches the source point. This produces some careful considerations in the evaluation of the integrations of these functions. Indeed for those integrals in which the collocation point lies on the element containing Q, the integrand becomes singular and special forms of numerical integration are required.

Finally, in spite of all the complexity related to the mathematical terms defining the *fundamental solutions*, they are all known as they contain only mathematical properties and distances between known points, and therefore they can easily be computed.

4.7.4 Boundary Displacement Equation

The *boundary integral equation* (4.31) is valid for any source point within the domain Ω. Thus, we have two sets of points: points p where the unit source are applied and points Q where we have to satisfy boundary conditions. In order to

obtain a solution for the points on the boundary, the source point is moved to the boundary Γ, i.e. $p{\to}P \in \Gamma$. This selection brings in the problem that some integrals in equation (4.31) only exist in the sense of a limiting value as p approaches Q (Aliabadi (2002)). Thus, introducing a multiplier $C_{ij}(P)$ which depends only on the geometry of the problem at the source point the *displacements boundary integral equation* can be written as

$$C_{ij}(P)u_j(P) + \int_\Gamma T_{ij}(P,Q)u_j(Q)d\Gamma = \int_\Gamma U_{ij}(P,Q)t_j(Q)d\Gamma$$
$$+ \int_\Omega U_{ij}(P,q)b_j(q)d\Omega$$

(4.48)

where the first integral is taken in the sense of the *Cauchy principal value*[*], the free term $C_{ij}(P)$ describes the local geometry around the point P. This multiplier takes value of 1 when p is complete inside the volume as we have seen from equation (4.28). If p is completely outside of the material C_{ij} is 0. If we consider a smooth boundary, the value of the free term C_{ij} is 1/2. For non-smooth boundaries, the common practice is to evaluate the free-term from consideration of *rigid body motion*. This approach is described in more detail in section 4.8.2.

4.8 Numerical Implementation

4.8.1 Division of the Boundary into Elements

Solving the BIE analytically is only possible for very simple problems. General problems therefore need to be solved numerically. Thus, the integral is divided into small segments, which in this case are called *boundary elements*, and each element is

---

[*] Cauchy principal value of a finite integral about a point c with $a \leq c \leq b$

$$\int_a^b f(x)dx \equiv \lim_{\varepsilon \to 0}\left[\int_a^{c-\varepsilon} f(x)dx + \int_{c+\varepsilon}^b f(x)dx \right]$$

defined by some *nodal points*. These elements are generally straight lines, quadratic curves or cubic splines in 2D, or triangular or quadrilateral surface elements in 3D.

The first step in the discretisation procedure is to divide the boundary Γ into N elements, so that, without considering the body forces, the equation (4.48) becomes

$$C_{ij}(P)u_j(P) + \sum_{n=1}^{N} \int_{\Gamma_n} T_{ij}(P,Q)u_j(Q)d\Gamma = \sum_{n=1}^{N} \int_{\Gamma_n} U_{ij}(P,Q)t_j(Q)d\Gamma \qquad (4.49)$$

where $\Gamma = \Gamma_1 \cup \Gamma_2 \cup \Gamma_3 \cup ... \cup \Gamma_N$

Over each element, the variation of the geometry and the variables must be described. This variation can be constant, linear, quadratic, cubic or even higher order. Generally, the higher the order of variation the more accurate solutions but also the higher CPU required. Results of acceptable engineering accuracy can consistently be obtained by a well-defined mesh of quadratic elements, so most BEM implementations use no higher order elements than this.

Two-dimensional formulation

Isoparametric elements use the same order of variation for both the geometry and unknown variables. In this formulation, the boundary geometry x_i, the unknown displacement field u_j and the traction field t_j are approximated using interpolation functions over an element containing m nodes

$$x_j = \sum_{\alpha=1}^{m} N_\alpha(\xi)x_j^\alpha$$

$$u_j = \sum_{\alpha=1}^{m} N_\alpha(\xi)u_j^\alpha \qquad (4.50)$$

$$t_j = \sum_{\alpha=1}^{m} N_\alpha(\xi)t_j^\alpha$$

where N_α, which are called the *shape functions*, are polynomials of degree $m-1$, and have the property that they are equal to 1 at node α and 0 at all other nodes, x_j^α, u_j^α, and t_j^α are the values of the functions at node α. These *shape functions* are

defined in terms of local coordinates ξ (-1 ≤ ξ ≤ 1). Figure 4.4 shows this new coordinate system for a quadratic element. The local variable ξ has its origin at the midpoint node and values −1 and +1 at the end nodes.
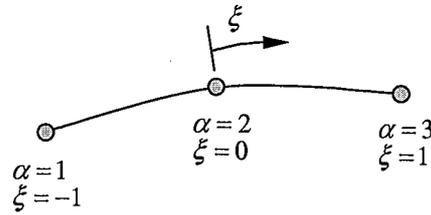
Figure 4.4: Quadratic two-dimensional boundary elements

The *shape functions* consider a number of nodes on each element where the variable value is given. Therefore, for a quadratic element ($m = 3$), we need three nodes on each element: one at the midpoint and one at either end (see figure 4.4). The shape functions for quadratic elements are

$$N_1 = \frac{1}{2}\xi(\xi - 1) \quad N_2 = (1 - \zeta^2) \quad N_3 = \frac{1}{2}\xi(1 + \xi) \tag{4.51}$$

In general, *shape functions* can be derived from the *Lagrangian polynomials*, which are defined for degree (m-1), as

$$N_\alpha(\xi) = \prod_{i=0, i \neq \alpha}^{m} \frac{\xi - \xi_i}{\xi_\alpha - \xi_i} \tag{4.52}$$

A discretised boundary element formulation can be obtained by substituting the expressions (4.50) into the integral equation (4.49)

$$C_{ij}(P)u_j(P) + \sum_{n=1}^{N} \int_{\Gamma_n} T_{ij}(P, Q(\xi))N_\alpha(\xi)J(\xi)d\xi \, u_j^\alpha =$$

$$\sum_{n=1}^{N} \int_{\Gamma_n} U_{ij}(P, Q(\xi))N_\alpha(\xi)J(\xi)d\xi \, t_j^\alpha \quad i,j = 1, 2 \tag{4.53}$$

To define the element formulation using the local coordinate ξ, the concept of the *Jacobian of transformation* is introduced. The *Jacobian* is used to transform the variables of integration or differentiation from one set of variables to another. For

example, the variable x in the following one-dimensional integral can be transformed into another variable ξ as follows

$$\int_{x_1}^{x_2} f(x)dx = \int_{\xi_1}^{\xi_2} f[x(\xi)]J(\xi)d\xi$$

(4.54)

The Jacobian $J(\xi)$ is defined as

$$J(\xi) = \frac{\partial x(\xi)}{\partial \xi}$$

(4.55)

In this case, because of the transformation of the variable from the boundary curve Γ to the intrinsic coordinate ξ, the Jacobian $J(\xi)$ of the transformation is calculated as

$$J(\xi) = \sqrt{\left(\frac{dx}{d\xi}\right)^2 + \left(\frac{dy}{d\xi}\right)^2}$$

(4.56)

Using the Jacobian, the coefficients $P_{ij}^{n\alpha}$ and $Q_{ij}^{n\alpha}$ are defined in terms of integrals over the surface Γ where $d\Gamma$ becomes $J(\xi)d\xi$ that is

$$P_{ij}^{n\alpha} = \int_{-1}^{1} T_{ij}(P,Q(\xi))N_\alpha(\xi)J(\xi)d\xi$$

$$Q_{ij}^{n\alpha} = \int_{-1}^{1} U_{ij}(P,Q(\xi))N_\alpha(\xi)J(\xi)d\xi$$

(4.57)

Substituting (4.57) into (4.53), a discretised boundary element formulation is finally expressed as follows

$$C_{ij}(P)u_j(P) + \sum_{n=1}^{N}\sum_{\alpha=1}^{m} P_{ij}^{n\alpha}u_j^{n\alpha} = \sum_{n=1}^{N}\sum_{\alpha=1}^{m} Q_{ij}^{n\alpha}t_j^{n\alpha} \quad i,j = 1,2$$

(4.58)

As we have seen, if the same *shape functions* are used for approximation of geometry and functions, the formulation is referred to as *isoparametric*. On the contrary, if the approximation is a higher order polynomial than that used for the geometry, the formulation is referred to as *superparametric* and conversely if the

77

function is represented by a lower order polynomial than the geometry then the formulation is referred to as *subparametric*.

Three-dimensional formulation

In three-dimensional problems the surface Γ is divided into triangular or quadrilateral elements. As in two-dimensional problems, *isoparametric* elements are chosen and therefore the same *shape functions* $N_\alpha(\xi, \eta)$ are used to approximate the boundary geometry x_j, and the displacement u_j and traction fields t_j respectively

$$x_j = \sum_{\alpha=1}^{m} N_\alpha(\xi, \eta) x_j^\alpha$$

$$u_j = \sum_{\alpha=1}^{m} N_\alpha(\xi, \eta) u_j^\alpha \qquad (4.59)$$

$$t_j = \sum_{\alpha=1}^{m} N_\alpha(\xi, \eta) t_j^\alpha$$

where the *shape functions* $N_\alpha(\xi, \eta)$ are functions of two local variables ξ and η ($-1 \leq \xi, \eta \leq 1$) for quadrilateral elements. This coordinate transformation is equivalent to mapping the boundary elements into a local square or triangle in the (ξ, η) plane, as shown in figure 4.5.
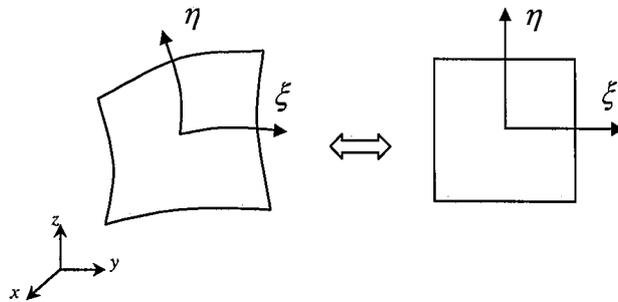
Figure 4.5: Transformation of boundary elements into a local system of coordinates
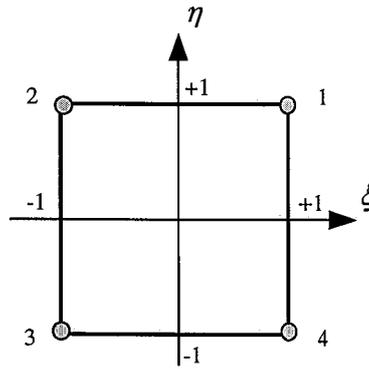
Figure 4.6: Linear 4-node quadrilateral element

The *shape functions* for quadrilateral elements can be obtained from the *Lagrangian polynomials*. From equation (4.52), the polynomials may be generalized to two dimensions by simply forming products of the shape functions. For example, consider a quadrilateral element shown in figure 4.6, defined by its four corners (± 1, ± 1).

$$\phi(\xi,\eta) = N_1(\xi,\eta)\phi^1 + N_2(\xi,\eta)\phi^2 + N_3(\xi,\eta)\phi^3 + N_4(\xi,\eta)\phi^4 \qquad (4.60)$$

$$N_1(\xi,\eta) = N_1(\xi)N_1(\eta) = \frac{1}{4}(\xi+1)(\eta+1)$$

$$N_2(\xi,\eta) = N_2(\xi)N_2(\eta) = \frac{1}{4}(1-\xi)(\eta+1)$$

$$N_3(\xi,\eta) = N_3(\xi)N_3(\eta) = \frac{1}{4}(\xi-1)(\eta-1)$$

$$N_4(\xi,\eta) = N_4(\xi)N_4(\eta) = \frac{1}{4}(1+\xi)(1-\eta)$$

(4.61)

Higher order *shape functions* for quadrilateral elements can be formulated in a similar way. In general, for *Lagrangian elements* (see figure 4.7 (a))

$$N_\alpha(\xi,\eta) = N_\alpha(\xi)N_\alpha(\eta) \qquad (4.62)$$

There is another commonly used set of quadrilateral elements, known as the *serendipity family* (see figure 4.7 (b)). These elements do not contain any interior nodes In terms of the local coordinate system (ξ,η), the *serendipity shape functions* for linear elements are the same as those given in (4.60).

(a)   (b)
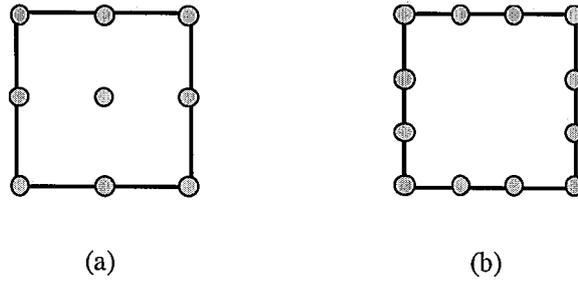
Figure 4.7: *Lagrangian* (a) and *serendipity* (b) elements

Similarly, for two-dimensional integrals the transformation of the variables has the following form

$$\int_{y_1}^{y_2}\int_{x_1}^{x_2} f(x,y)dxdy = \int_{\eta_1}^{\eta_2}\int_{\xi_1}^{\xi_2} f[x(\xi,\eta), y(\xi,\eta)]J(\xi,\eta)d\xi d\eta \tag{4.63}$$

where the Jacobian $J(\xi,\eta)$ can be written as

$$J(\xi,\eta) = \frac{\partial(x,y)}{\partial(\xi,\eta)} = \begin{vmatrix} \dfrac{\partial x}{\partial \xi} & \dfrac{\partial x}{\partial \eta} \\ \dfrac{\partial y}{\partial \xi} & \dfrac{\partial y}{\partial \eta} \end{vmatrix} = \frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta} - \frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \xi} \tag{4.64}$$

Note that the functions $x(\xi,\eta)$ and $y(\xi,\eta)$ must be continuous and have continuous first partial derivatives in the specified region of integration.

Like the 2D case, the *boundary integral equation* can be written in the discretised form

$$C_{ij}(P)u_j(P) + \sum_{n=1}^{N}\sum_{\alpha=1}^{m} P_{ij}^{n\alpha}u_j^{n\alpha} = \sum_{n=1}^{N}\sum_{\alpha=1}^{m} Q_{ij}^{n\alpha}t_j^{n\alpha} \qquad i,j = 1,2,3 \tag{4.65}$$

The coefficients $P_{ij}^{n\alpha}$ and $Q_{ij}^{n\alpha}$ are now defined in terms of double integrals

$$P_{ij}^{n\alpha} = \int_{-1}^{1}\int_{-1}^{1} T_{ij}(P,Q(\xi,\eta))N_\alpha(\xi,\eta)J(\xi,\eta)d\xi d\eta$$

$$Q_{ij}^{n\alpha} = \int_{-1}^{1}\int_{-1}^{1} U_{ij}(P,Q(\xi,\eta))N_\alpha(\xi,\eta)J(\xi,\eta)d\xi d\eta \tag{4.66}$$

Thus, with the incorporation of the *shape functions* and the transformation of the variables, the discretised *boundary integral equation* for 2D (4.58) and for 3D (4.65) consists of integrals which have to be performed over each element. This can be performed using a numerical integration procedure such as *Gauss Legendre quadrature* (Trevelyan (1994)).

Finally, careful consideration must be taken when the integrals are calculated for the element containing the *collocation point*; in such cases the *fundamental solution* is *singular* at the *collocation point*, as explained in section 4.7.3. To integrate numerically *singular* or *nearly singular* functions, when the *source node* is close to the element under consideration, it requires a more complex integration scheme. For example the use of *high-order Gauss quadrature* instead of the *low-order integration* scheme applied for large enough r. Thus, depending on the nature of the *fundamental solution* and the relative position of the *source point* with respect to the element on which integration is being carried out, different integrals schemes are carried out (see Aliabadi (2002) for more details on numerical integration).

4.8.2 Assembly of System of Equations

The *point collocation* method is used to obtain the solution for the integral equation. In this method, a set of equations is obtained by placing the load point P in turn at each point on the surface. The point load P is first placed at node-point 1 that gives only one set of equations relating all M variables (nodes) on the surface. Thus, figure 4.8 shows an example of the point load placed at node-point 1 and integrated over the element with nodes 12 and 13. Then the point load is placed at node-point 2 yielding another set of equations, and so on until all M sets of equations are formed. It should be noticed that two equations, three in the case of 3D, are formed for each collocation point by applying the source in each coordinate direction.
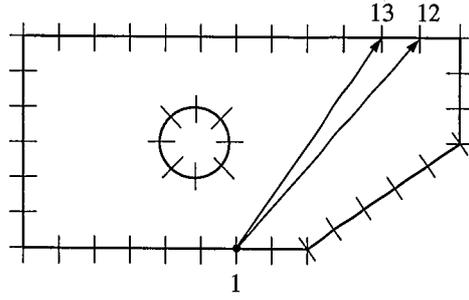
Figure 4.8: Point collocation at node 1 over element with nodes 12 and 13

Therefore, following this *point collocation* method, equations (4.58) for two-dimensions and (4.65) for three-dimensions are evaluated at nodal points

$$C_{ij}(\mathrm{P}^c)u_j(\mathrm{P}^c) + \sum_{n=1}^{N}\sum_{\alpha=1}^{m} P_{ij}^{n\alpha}(\mathrm{P}^c)u_j^{n\alpha} = \sum_{n=1}^{N}\sum_{\alpha=1}^{m} Q_{ij}^{n\alpha}(\mathrm{P}^c)t_j^{n\alpha} \quad c=1,...,\mathrm{M} \quad (4.67)$$

where P^c is the collocation point. The double sum in (4.67) must be evaluated considering that some nodes are shared between elements, and since the displacement values $u_j^{n\alpha}$ are uniquely defined at these nodes, they can be combined to give a sum over all nodes; thus, rewriting (4.67)

$$C_{ij}(\mathrm{P}^c)u_j(\mathrm{P}^c) + \sum_{\gamma=1}^{M} \overline{H}_{ij}^{c\gamma} u_j^{\gamma} = \sum_{n=1}^{N}\sum_{\alpha=1}^{m} G_{ij}^{cn\alpha} t_j^{n\alpha} \quad c=1,...,\mathrm{M} \quad (4.68)$$

where $\overline{H}_{ij}^{c\gamma}$ is made up from $P_{ij}^{n\alpha}(\mathrm{P}^c)$ and $G_{ij}^{cn\alpha}$ is equal to $Q_{ij}^{n\alpha}(\mathrm{P}^c)$. Building the first term into the other displacement unknowns in (4.68) gives

$$\sum_{\gamma=1}^{M} H_{ij}^{c\gamma} u_j^{\gamma} = \sum_{n=1}^{N}\sum_{\alpha=1}^{m} G_{ij}^{cn\alpha} t_j^{n\alpha} \quad (4.69)$$

where $H_{ij}^{c\gamma} = C_{ij}(\mathrm{P}^C)\delta_{c\gamma} + \overline{H}_{ij}^{c\gamma}$ and $\delta_{c\gamma}$ is the *Kronecker delta function* (4.12). Therefore, the discretised boundary element equation may be written in matrix form as

$$\mathbf{Hu} = \mathbf{Gt} \quad (4.70)$$

82

where **H** is the *displacement coefficient matrix* and **G**, the *traction coefficient matrix* containing known integrals of the product of the *shape functions*, *the Jacobians* and the *fundamental solutions* T_{ij} and U_{ij}, respectively. The displacement vector **u** and the traction vector **t** contain unknowns and also the known boundary conditions. It is worth pointing out that the matrix **H** is a 2Mx2M (2D) or 3Mx3M (3D) square matrix. **G** is a rectangular matrix with 2M (2D) or 3M (3D) rows and 2Nm or 3Nm, respectively, columns. Therefore, the vector **t** has 2Nm (2D) or 3Nm (3D) components and is larger than **u**, which has 2M (2D) or 3M (3D) components, respectively. This formulation allows traction to be discontinuous across element boundaries, which is common occurrence in application of boundary conditions for engineering problems.

The parameter C_{ij} contributes only to the diagonal terms. This parameter can be evaluated directly from a consideration of *rigid body motion* (constant displacement of all the nodes in any direction). This motion results in zero traction everywhere, which makes the right-hand side of equation (4.68) zero

$$C_{ij}(P^c) = -\sum_{\gamma=1}^{M} \overline{H}_{ij}^{c\gamma} \tag{4.71}$$

Thus the diagonal terms in $H_{ij}^{c\gamma}$ can now be evaluated as the (negative) sum of all the other non-diagonal coefficients, since equation (4.71) can be rewritten as

$$C_{ij}(P^c) + \overline{H}_{ij}^{cc} = -\sum_{\substack{\gamma=1 \\ \gamma \neq c}}^{M} \overline{H}_{ij}^{c\gamma} \tag{4.72}$$

that is

$$H_{ij}^{cc} = -\sum_{\substack{\gamma=1 \\ \gamma \neq c}}^{M} \overline{H}_{ij}^{c\gamma} \tag{4.73}$$

Applying the boundary conditions, for 2D problems, since there are 2M equations, we need 2M prescribed values. Therefore for each node and in each direction k either the displacement u_k or the traction t_k must be prescribed. For three-dimensional problems the number of equations is 3M and consequently the number of boundary

values prescribed is now 3M. In this case the number of node variables is six, $(u_x, u_y, u_z, t_x, t_y, t_z)$. After applying the boundary conditions, the resulting system of algebraic equations may be rewritten as

$$\mathbf{Ax} = \mathbf{By} \qquad\qquad (4.74)$$

The vector $\mathbf{x}$ contains all the unknown boundary displacements or tractions, $\mathbf{A}$ and $\mathbf{B}$ are coefficient matrices and $\mathbf{y}$ contains the prescribed boundary conditions. This stage requires careful consideration of conditions at sharp corners and edges with discontinuous tractions or displacements on either side of the corner (Becker (1992)). Multiplying the known matrix $\mathbf{B}$ by the known vector $\mathbf{y}$, the previous equation can be rewritten in the following form

$$\mathbf{Ax} = \mathbf{b} \qquad\qquad (4.75)$$

where $\mathbf{b}$ is the new vector formed from the product $\mathbf{By}$. In the above system of equations, $\mathbf{A}$ is fully populated, unsymmetric, square matrix. Generally, *Gauss elimination* or some other direct technique is chosen as an equation solver. As the solution time for this type of matrix is proportional to the cube of the total number of degrees of freedom, the required computational time can become large for complex structural models (particularly 3D). Iterative techniques suitable for unsymmetric systems are becoming popular, e.g. BiCGSTAB (Van der Vorst (1992)) and GMRES (Saad and Schultz (1986)) (with appropriate preconditioning) have been shown to offer reasonable convergence.

The reason why the BEM matrices are fully populated is that each nodal point *influences* all the other nodal points on the boundary. This is in contrast to the FEM, in which the nodes influence only their neighbouring nodal points and these results in sparsely populated matrices that can be exploited to provide efficient storage and solution schemes. However, the BEM matrices may require less storage than the FEM ones because the interior is not modelled and a lesser degree of mesh refinement is required for the same level of accuracy (Brebbia and Trevelyan (1986)) so the rank of the matrix is considerable smaller for many problems (see figure 4.9).

FEM   BEM
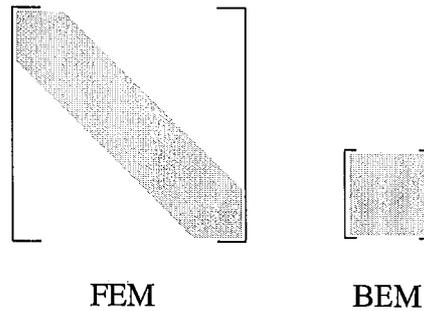
Figure 4.9: Comparison of FEM and BEM system matrices

It is worth noting that in elastostatic problems the displacement magnitudes are generally several orders of magnitude less than the tractions. Therefore, when these magnitudes occur together in equation (4.75) there is a great difference on the order of magnitude. This means that the solution may suffer from inaccuracies due to an inherent ill-conditioning of the system. To solve this problem the equation (4.70) is multiplied by a suitable scaling factor S

$$Hu = SGS^{-1}t \tag{4.76}$$

A suitable value for S can be $S = E/L_{max}$, with E the Young's modulus and L_{max} the maximum distance between any two nodes.

4.9 Internal Solution

The solution of the *integral equation* provides values of u and t only on the boundary of the domain. Evaluation of stresses and displacements at any point within the body is obtained from the solution of the system of equations. Once the values of all displacements and tractions are known on the boundary, then the BIE (4.31) for the unknown interior displacements can be discretised in a similar way to the boundary element formulation. Now, moving the source point to some point inside the material we can evaluate the solution u(p) around the boundary for this source point in a similar way as we have done for the source point on the boundary, knowing now the values of u and t. Thus, for any point p^i wholly within the domain, C_{ij}(P) = 1, and the

interior displacements at p^i may be evaluated using another application of the integral equation

$$u_j(p^i) + \sum_{\gamma=1}^{M} \overline{H}_{ij}^{c\gamma} u_j^\gamma = \sum_{n=1}^{N} \sum_{\alpha=1}^{m} G_{ij}^{cn\alpha} t_j^{n\alpha} \quad c = 1,...,M \tag{4.77}$$

Since the interior points never lie on the boundary, the points p and Q are not coincident and therefore the *fundamental solutions* are no longer singular. However, it is not desirable to place an interior point too close to the boundary to avoid inaccuracies in the results, which are caused by numerical problems in integrating functions $1/r$, $1/r^2$ or $\ln(1/r)$ over a domain in which $r \to 0$.

It should be pointed out that the solutions at interior points are very accurate since there are no discretisations or approximations imposed on the solution inside the solution domain. The accuracy is therefore of the same order as that in the solution over the boundary Γ.

4.10 Stress Computation

The solution of (4.75) computes the solution for tractions and displacements around the boundary. However, it still remains to calculate the values of the stress components. There are two methods of calculating the boundary stresses. In the first method, the stresses are calculated directly from the displacements and tractions obtained from the BEM solution, by differentiation using the *shape functions*. On the other hand, the second method calculates the stresses operating directly on the BIE. Thus, the strains at any interior point can be obtained by differentiating the displacements in equation (4.31) with respect to the coordinate direction k to give

$$u_{i,k}(p) = \int_\Gamma U_{ij,k}(p,Q)t_j(Q)d\Gamma - \int_\Gamma T_{ij,k}(p,Q)u_j(Q)d\Gamma$$

$$+ \int_\Omega U_{ij,k}(p,q)b_j(q)d\Omega \tag{4.78}$$

where $U_{ij,k}$ and $T_{ij,k}$ are the derivatives of the *fundamental solutions*. Finally, applying Hooke's law (4.11) into equation (4.78) to give

$$\sigma_{ik}(p) = \int_{\Gamma} D_{kij}(p,Q)t_k(Q)d\Gamma - \int_{\Gamma} S_{kij}(p,Q)u_k(Q)d\Gamma$$
$$\int_{\Omega} D_{kij}(p,q)b_k(q)d\Omega$$

(4.79)

where D_{kij} and S_{kij} are obtained from $U_{ij,k}$ and $T_{ij,k}$.

As we can see, the second method is more accurate but more complex to solve than the first method since it implies derivatives of the *fundamental solutions* and numerical integrations. More difficult still is the fact that if the *fundamental solution* is of $1/r$ singularity, its derivative will be $1/r^2$. Likewise a $1/r^2$ *fundamental solution* will have a derivative that becomes like $1/r^3$. This complicates the numerical integration further. Therefore, the first method is used to explain the calculation of the boundary stresses. Moreover, this has also been the method implemented in the software used (Trevelyan (1994), Trevelyan and Wang (2001)). The method is explained for two-dimensions for the sake of simplicity.

To calculate the stress components, it is necessary to calculate first the strains from the derivatives of displacement results, and then to calculate the stress components from the strains using elasticity theory. To start with, we define the local directions 1 and 2 as the tangential and normal directions, respectively, to the surface Γ. Therefore the local tangential component of the displacement vector u_1 can be written as follows

$$u_1(\xi) = u_x(\xi)m_x + u_y(\xi)m_y$$

(4.80)

where m_x and m_y are the x and y components of the unit tangential vector. Using the *shape functions* and differentiating with respect to the tangential direction the tangential strain ε_{11} can be obtained as

$$\varepsilon_{11}(\xi) = \frac{1}{J(\xi)}\left\{\left[\sum_{c=1}^{3}\frac{\partial N_c(\xi)}{\partial \xi}u_x\right]m_x + \left[\sum_{c=1}^{3}\frac{\partial N_c(\xi)}{\partial \xi}u_y\right]m_y\right\}$$

(4.81)

The local components of the traction vector, t_1 and t_2, are the tangential and normal tractions to the surface as shown in figure 4.10. These local tractions are written in terms of the Cartesian global traction as

$$t_1 = -t_x \sin\alpha + t_y \cos\alpha$$
$$t_2 = t_x \cos\alpha + t_y \sin\alpha$$

$$(4.82)$$
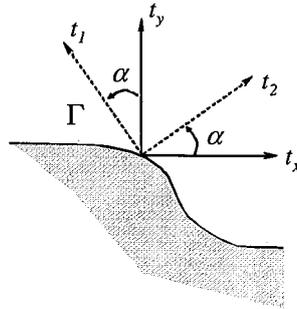
Figure 4.10: Local and global components of the traction vector

The stresses in the local directions 1 and 2, are obtained applying the stress-strain relationships

$$\sigma_{11} = \left(\frac{E}{1-v^2}\right)\varepsilon_{11} + \left(\frac{v}{1-v}\right)t_2$$
$$\sigma_{22} = t_2$$
$$\sigma_{12} = t_1$$

$$(4.83)$$

Finally, to transform these local stresses into global ones, the transformation matrix is used

$$\begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \end{bmatrix} = \begin{bmatrix} \sin^2\alpha & \cos^2\alpha & -2\sin\alpha\cos\alpha \\ \cos^2\alpha & \sin^2\alpha & 2\sin\alpha\cos\alpha \\ -\sin\alpha\cos\alpha & \sin\alpha\cos\alpha & (\cos^2\alpha - \sin^2\alpha) \end{bmatrix} \begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{bmatrix}$$

$$(4.84)$$

with the angle $\alpha = \tan^{-1}\left(\frac{m_y}{m_x}\right)$

4.11 Treatment of Edges and Corners

In assembling the equations, some problems must be considered carefully, such as modelling geometries with sharp corners and edges with discontinuous tractions on either sides of the corner. In a discretised boundary corner, the outward normal changes its value sharply from one element to another as shown in figure 4.11. Therefore, a corner node can have two sets of traction values, one for each side of the corner.

A number of procedures have been used to solve this problem. Generally, if the multi-valued tractions are assumed to be equal, then the errors are concentrated mainly at the corners and are not significant at other nodes. One of these approaches uses *discontinuous elements*, in which the collocation points at the corners are taken inside the element as shown in figure 4.11. In this approach the geometry is still approximated using the continuous *shape functions* similar to those presented in previous sections, but the displacement and traction fields are now represented using *discontinuous elements*.
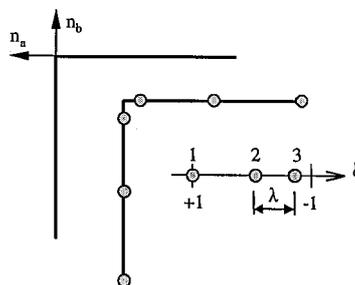
Figure 4.11: Discontinuous elements for corner nodes (2D)

Another approach is based on identifying the possible situations in a corner. Figure 4.12 shows the different boundary conditions at the node j common to the element $i-1$ and the element j. That is, at node j the unknowns may be

- Displacement u_j: Both, t_{i-1} and t_i are needed

- Traction t_{i-1}: u_j and t_i are needed

- Traction t_i: u_j and t_{i-1} are needed

- Traction $t_i = t_{i-1}$: Only u_j is needed

Therefore, the assembly of the equation (4.74) ($\mathbf{Ax} = \mathbf{By}$) depends on the boundary conditions considered in this way. These considerations get more complicated when symmetry conditions are used since the boundary condition in such a case is normal displacement equals to zero.
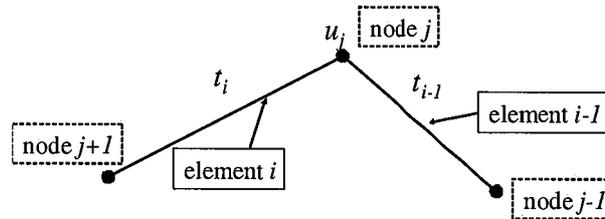
Figure 4.12: Boundary conditions in a corner

4.12 Multi-zone Formulation

In some problems, it is necessary to divide the model into several regions or *zones*, e.g. non-homogeneous problems, notch and crack problems and problems with two or more materials. The treatment of the *multi-zone formulation* is a straightforward extension of the BEM procedures described in the early sections of this chapter.
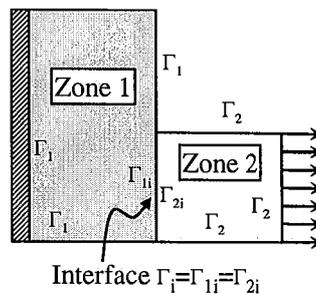
Figure 4.13: Multi-zone problem

Consider a domain consisting of two zones, Zone 1 and Zone 2, which have boundaries Γ_1 and Γ_2 and an interface Γ_i, as shown in figure 4.13. For each zone we have the following variables

- $\mathbf{u}_1$, $\mathbf{t}_1$: nodal displacements and tractions at the external boundary Γ_1

90

- u_{1i}, t_{1i}: nodal displacements and tractions at the interface Γ_{1i}

- u_2, t_2: nodal displacements and tractions at the external boundary Γ_2

- u_{2i}, t_{2i}: nodal displacements and tractions at the interface boundary Γ_{2i}

The system of equations for zone 1 can be written as

$$[\mathbf{H}_1 \quad \mathbf{H}_{1i}]\left\{\begin{matrix} \mathbf{u}_1 \\ \mathbf{u}_{1i} \end{matrix}\right\} = [\mathbf{G}_1 \quad \mathbf{G}_{1i}]\left\{\begin{matrix} \mathbf{t}_1 \\ \mathbf{t}_{1i} \end{matrix}\right\} \tag{4.85}$$

Similarly for zone 2

$$[\mathbf{H}_2 \quad \mathbf{H}_{2i}]\left\{\begin{matrix} \mathbf{u}_2 \\ \mathbf{u}_{2i} \end{matrix}\right\} = [\mathbf{G}_2 \quad \mathbf{G}_{2i}]\left\{\begin{matrix} \mathbf{t}_2 \\ \mathbf{t}_{2i} \end{matrix}\right\} \tag{4.86}$$

The compatibility and equilibrium conditions at the interface Γ_i are

$$\begin{aligned} \mathbf{u}_{1i} &= \mathbf{u}_{2i} \equiv \mathbf{u}_i \\ \mathbf{t}_{1i} &= -\mathbf{t}_{2i} \equiv \mathbf{t}_i \end{aligned} \tag{4.87}$$

Using the above equations, we can combine the coefficient matrices (4.85) and (4.86) for the two regions and applying the interface boundary conditions (4.87)

$$\begin{bmatrix} \mathbf{H}_1 & \mathbf{H}_{1i} & 0 & 0 \\ 0 & 0 & \mathbf{H}_2 & \mathbf{H}_{2i} \\ 0 & \mathbf{I} & 0 & -\mathbf{I} \\ 0 & 0 & 0 & 0 \end{bmatrix} \left\{\begin{matrix} \mathbf{u}_1 \\ \mathbf{u}_{1i} \\ \mathbf{u}_2 \\ \mathbf{u}_{2i} \end{matrix}\right\} = \begin{bmatrix} \mathbf{G}_1 & \mathbf{G}_{1i} & 0 & 0 \\ 0 & 0 & \mathbf{G}_2 & \mathbf{G}_{2i} \\ 0 & 0 & 0 & 0 \\ 0 & \mathbf{I} & 0 & \mathbf{I} \end{bmatrix} \left\{\begin{matrix} \mathbf{t}_1 \\ \mathbf{t}_{1i} \\ \mathbf{t}_2 \\ \mathbf{t}_{2i} \end{matrix}\right\} \tag{4.88}$$

Rearranging (4.88) in terms of u_i and t_i

$$\begin{bmatrix} \mathbf{H}_1 & \mathbf{H}_{1i} & -\mathbf{G}_{1i} & 0 \\ 0 & \mathbf{H}_{2i} & -\mathbf{G}_{2i} & \mathbf{H}_2 \end{bmatrix} \left\{\begin{matrix} \mathbf{u}_1 \\ \mathbf{u}_i \\ \mathbf{t}_i \\ \mathbf{u}_2 \end{matrix}\right\} = \begin{bmatrix} \mathbf{G}_1 & 0 \\ 0 & \mathbf{G}_2 \end{bmatrix} \left\{\begin{matrix} \mathbf{t}_1 \\ \mathbf{t}_2 \end{matrix}\right\} \tag{4.89}$$

Finally after the substitution of the boundary conditions, the resulting system of equations can be written as

$$\begin{bmatrix} \mathbf{A}_1 & \mathbf{H}_{1i} & -\mathbf{G}_{1i} & 0 \\ 0 & -\mathbf{H}_{i2} & -\mathbf{G}_{2i} & \mathbf{A}_2 \end{bmatrix} \begin{Bmatrix} \mathbf{x}_1 \\ \mathbf{u}_i \\ \mathbf{t}_i \\ \mathbf{x}_2 \end{Bmatrix} = \begin{bmatrix} \mathbf{B}_1 & 0 \\ 0 & \mathbf{B}_2 \end{bmatrix} \begin{Bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{Bmatrix} \qquad (4.90)$$

It can be seen that the coefficient matrices are block-banded with one block for each region and overlaps between blocks in the **A** matrix at the common interface (see figure 4.14). For large complex models it is often more economical to split the model into several smaller, simpler sub-models. These sub-models or zones, which may also have different material properties, are modelled independently and then joined together along an interface as described above. At these interfaces the compatibility and equilibrium conditions are imposed for the interface requirements. This strategy leads to an overall matrix system, which has a blocked, sparse and unsymmetrical character. This character of multi-zone formulation extends the range of problems that can be solved, due to the large savings in storage and CPU required to build and solve the matrix. However, this advantage has to be balanced with the fact that multi-zone modelling adds additional unknowns to the overall problem; these unknowns are associated with the response at the zone interfaces, and therefore this tends to require more computer cost for problems which require substantial numbers of elements on the interface(s).
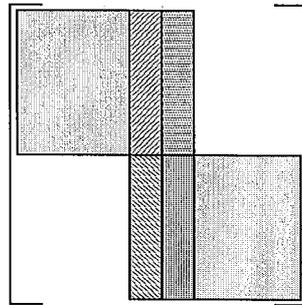
Figure 4.14: Block-banded matrix produced by zoning

4.13 Further Applications

This chapter has focused on the BEM applied to elastostatic problems. However, this method has many other applications in engineering, described briefly in this section.

Potential Problems

They are governed by a differential equation that satisfies Laplace's or Poisson's equation. Examples of potential problems are ideal fluid flow, steady-state heat conduction and electrostatics can be found in Becker (1992). Moreover, the BEM has also been proved to be an effective tool to solve the Helmholtz equation using plane waves to express the potential field, for example, in acoustic problems (Perrey-Debain *et al.* (2003)).

Thermoelastic, Gravitational, Centrifugal and Inertia Body Force Problems

These problems are concerned with the induced displacements and stresses caused by a gradient across the body. This effect is treated as a body force over the solution domain.

In the previous sections related to elastostatics, body forces have been assumed to be zero. However, if this is not the case, the body force integral in equation (4.31) must be evaluated. There are several methods developed for the evaluation of this integral such as the *cell integration approach* or the *Galerkin method* (Aliabadi (2002), Becker (1992)). In this section none of these methods is described for the sake of simplicity.

Contact Problems and Fracture Mechanics

The numerical modelling of contact problems and fracture mechanics are fields where the BE is more suitable than FE. The reasons are related to the accuracy of the stresses on the boundary and the easy re-meshing in BE (Trevelyan (1994)). The

stress analysis of contact problems has many applications such as bearings, gears, pressure vessels and bio-engineering (Aliabadi (2002), Becker (1992)).

Fracture mechanics is applied in problems related to lifetime prediction and crack growth control. In the study of stresses around cracks, the stresses at the tip of the crack are assumed to be infinite and are characterised by the stress intensity factor, K. When K reaches a critical value K_c then a catastrophic failure occurs. Therefore, the theory of fracture has been developed to determine K_c for each specific problem.

Early application of the BEM to crack growth was due to Cruse (1971). Since then, the method improved using *singularity elements* (Becker (1992)). As the crack grows in the sub-critical regime, a BE model can readily be updated to model the extended crack surface. This method thus lends itself well to the simulation of fatigue crack growth when used in conjunction with an appropriate crack growth law. Another method also used is the *dual boundary element method* (DBEM, Portela *et al.* (1992)). The main idea of DBEM is to model the crack as two separate surfaces facing each other, with corresponding discretisation points. Then, the *displacement boundary integral equation* (4.31) is applied to one surface (Γ^+) meanwhile the *traction boundary integral equation*, obtained multiplying equation (4.79) by the outward unit normal u_i, is applied to the second surface (Γ^-).

Coupling the BEM and FEM

Combining BE and FE techniques offers promise as an efficient way of tackling problems that contain features requiring both BE and FE capabilities. For example, a structure containing a thin section and a crack. For such situation the thin section should be modelled using the FEM and the crack area using the BEM.

There are two alternative approaches to combining these techniques; either converting the BE into equivalent stiffness matrices or transforming the FE forces into nodal tractions. Compatibility and equilibrium conditions must be satisfied on the interface of the BE and FE mesh (Aliabadi (2002)).

Sensitivity Analysis and Shape Optimisation

The boundary element method can be used to determine the best shape required to certain structural situation. There are several reasons why the BEM is attractive to shape optimisation such as the accuracy in evaluating displacements and stresses, as well as the easy adaptation to changes in the geometry of the structure during the optimisation process (Parvizian and Fenner (1997)). More information on this topic is given in detail in the chapter related to the optimisation methods (see chapter 2).

4.14 Discussion and Conclusions

To conclude this chapter, the main advantages and disadvantages of the *boundary element method* are summarised here. The main advantages are as follows

- It simplifies mesh data preparation, because only the surface of the component or structure to be analysed needs to be discretised.

- High accuracy on the resolution of stresses because within the solution domain the governing differential equations are satisfied exactly, rather than approximately as in the case of the FEM.

- For the same level of accuracy, the BE method generally uses fewer nodes and elements, but has a fully populated matrix. Sparse blocked matrices are obtained in the multi-zone approach.

- Using the boundary element method, less unwanted information about internal points is obtained. Since in most linear static engineering problems the worst situations occur on the surface, using a boundary element model offers much more efficient use of computing resources than modelling a three dimensional body using a large number of volumetric elements.

- Since it is a surface-oriented technique, it is particularly well suited for shape optimisation problems.

- The method may be used for a variety of problems, including structural, thermal, acoustic, dynamics, fluid analysis and infinite problems.

In spite of these features the BEM has some disadvantages and drawbacks

- The BEM involves dealing with complex mathematics, since it requires an integral equation relation, plus transformation to a boundary formulation using fundamental solutions.

- The interior must be modelled in non-linear problems. However, in some non-linear cases, such as elastoplasticity, interior modelling can be restricted to selected areas such as the region around a crack tip (Pineda and Aliabadi (2003)). The DRBEM (*dual reciprocity boundary element method*, Nardini, and Brebbia (1982), Partridge *et al.* (1992)) has been developed for the BE solution of such problems but, in spite of a large volume of research in the late 1980's and early 1990's, the method has not matured to the extent that it is widely used.

- The BE method uses unsymmetric and fully populated solution matrix whereas the FE solution matrices are usually much larger but sparsely populated. However, to obtain the same level of accuracy as the FE solutions, the BE method needs only a relatively modest number of nodes and elements.

- The method is not suitable for thin shell analysis. This is because of the large surface/volume ratio and that the separation between nodal points becomes very small. This causes inaccuracies in the numerical integrations. Nevertheless, recent publications (Di Pisa and Aliabadi (2003)) have shown a BEM formulation for thin-plate structures obtained by assembling 2D elasticity with plate bending theory.

Summing up, the most important advantages of the BEM concern to mesh generation and manipulation. Moreover, the applicability of the BEM seems to be superior to the FEM in problems where the boundary stresses are important.

5

SHAPE OPTIMISATION IN 2D

5.1 Overview

This chapter describes the algorithm developed to perform shape optimisation for two-dimensional problems. The approach is based on the evolutionary structural optimisation (ESO) method. The boundary element method is used to carry out the structural analysis in elastostatic problems. The boundary is described using NURBS curves, each of them defined by a set of control points. The iterative procedure adds and removes material progressively depending on the stress level within the structure. This is accomplished by moving sets of control points instead of dealing directly with elements of the mesh. Furthermore, to obtain a better control of the geometry, the number of control points is not fixed for the entire process. Some examples are presented to show the effectiveness of the algorithm.

5.2 Introduction

In shape optimisation, the topology of the structure is fixed and only the shape of the boundary can change. Extensive work has been done in shape optimisation using both numerical methods, FEM and BEM, for the structural analysis (Haftka and Grandhi (1986)). Of especial interest has been the application of BEM when design sensitivity analysis is implied in the optimisation algorithm. This is due to the high

accuracy of the BEA results on the boundary. Indeed, since the method deals with integrals over the boundary, only the boundary needs to be discretised which is a clear advantage for remeshing purposes.

Considering these advantages and also bearing in mind some drawbacks of FE-based methods such as checkerboard patters and jagged edges (see chapter 2) an ESO-based approach is implemented for shape optimisation. This technique considers the BE as a tool to carry out the structural analysis. The optimisation approach is stress-based selecting only the region of interest to be optimised; in ESO this process is called *nibbling* (Xie and Steven (1997)).

The geometry is described using the nonuniform rational B-spline curves (NURBS). These parametric curves allow free-form representation with total geometry control over the curve. This feature is very interesting since, a priori, the final design is not known. The process is fully integrated within the in-house boundary element software (Concept Analyst, Trevelyan and Wang (2001)) allowing a straightforward communication between software and optimisation code.

5.3 Algorithm

The optimisation process is based on the principal idea of the ESO method, i.e. that inefficient material is slowly removed from the structure, and material is added to critical areas as required. Unlike the FE-based ESO, the numerical analysis is carried out with the BEM and so the design variables of the problem are exclusively related to the boundary. Figure 5.1 shows the flow chart of the basic shape optimisation process. Later in the chapter, the algorithm and flowchart would be enhanced by adding control point insertion and deletion checks, evolution ratios and other specific issues. Thus in figure 5.1 the basic algorithm has been divided into several steps in order to clarify the process.

Step 1: The geometry of the structure is defined and the changeable boundaries modelled by NURBS curves. This initial design is subjected to a set of loads and constraints.

Step 2: A boundary element analysis, BEA, follows the model description.

Step 3: Searching for the least stressed nodes within the boundary mesh carries out the *removal* of material. Therefore, the nearest control points to those nodes are identified and moved to a more efficient shape. In addition, if a node is found with a von Mises stress (or any selected criterion) higher than the yield stress (or any maximum criterion) then a similar process to removal is undertaken but, however, the opposite direction of movement of the control points results in a material *addition*.

Step 4: Such a procedure is repeated (from step 2) until the stopping criterion is satisfied. This minimum is not mathematically searched but it is found according to the evolution of the objective function.
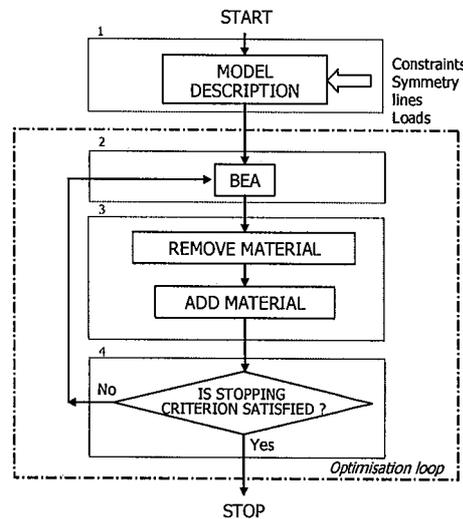
Figure 5.1: Flow chart of the shape optimisation process.

5.4   Geometry Definition

According to the applied constraints, loads or any other design requirements, the structural domain can be identified as *design* and *non-design domain*. The design domain is any area that can be modified and is free to change during the optimisation. On the contrary, the non-design domain is any area that cannot be

modified and remains fixed during the optimisation. In this way, the shape optimisation can consider as much (or as little) of the boundary as is required for any individual problem. Following this notation, on the *geometry definition step* the boundary domain can be divided into three different types of curves (see figure 5.2); i.e. *changeable*, *non-changeable* and *symmetry lines*. Therefore, lines that can change freely along the process are identified as changeable lines, whereas those lines that cannot change due to constraints or design restrictions are identified as non-changeable. Symmetry lines can be regarded as an intermediate step between changeable and non-changeable lines. They can be modified by changing their length but only if the adjoining line is a changeable line; furthermore, their changes are as a result of changes in the changeable line. These symmetry lines are always straight lines; therefore, their variations are restricted along the direction of the line.
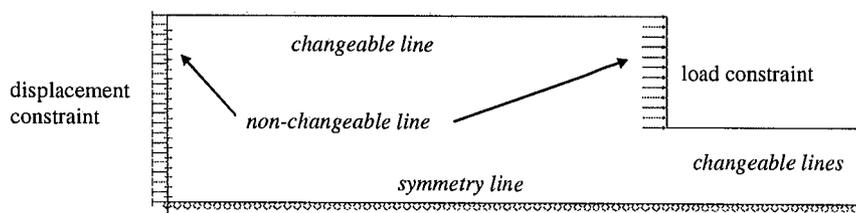
Figure 5.2: Definition of the changeable and non-changeable lines

NURBS curves, already presented in chapter 3, define the changeable geometry in order to control the curvature and tangency of the boundary. Usually, these changeable lines are initially straight lines, since the shape is not determined. Nevertheless the program converts them automatically into NURBS curves with their corresponding control points.

Although the concept of a symmetry line is well known, this section explains how these types of lines vary according to the changeable lines next to them. In the event of any adjoining changeable line being modified, then the line of symmetry can be either shortened or lengthened depending on the new position of the changeable line. However, the direction of movement of the symmetry line (see figure 5.3) is only permitted along the plane of symmetry. The quantity of movement is determined by the adjoining changeable line movement.
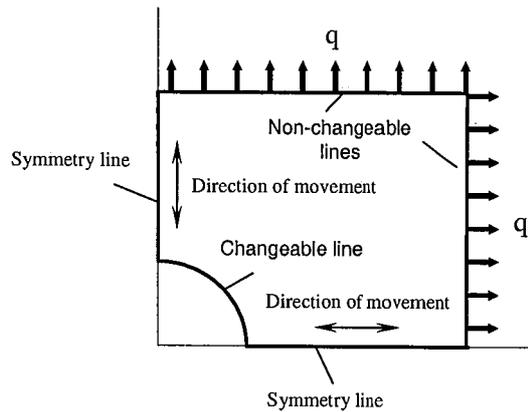
Figure 5.3: Movement of symmetry lines

5.5 Boundary Element Model

As previously stated in chapter 4, in the boundary element method the boundary integral equations are approximated by a set of discretised integral equations. As a result, the boundary surface is divided into elements as shown in figure 5.4, thus the response is given at the nodal points associated with the elements.

The main advantage of this method relates to the mesh, because only the surface of the structure needs to be discretised. The analytical results inside the structure are calculated at an arbitrary number of *internal points*. In the software used, these points are randomly distributed throughout the interior domain.

Concept Analyst (Trevelyan and Wang (2001)) allows three different types of element mesh so called *fine*, *standard* and *coarse* mesh. Generally, all the examples are discretised using a *standard* boundary element mesh and quadratic order elements, i.e. three nodes per element. This standard element mesh produces 0.5% – 1.0% accuracy in peak stress. Furthermore, areas having high stress gradients would require a finer meshing as is shown in the example in figure 5.4.

101
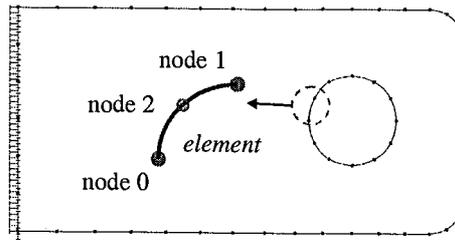
Figure 5.4: Illustration of a standard quadratic boundary element mesh

5.6 Removal and Addition of Material

5.6.1 Identifying Inefficient Areas

The von Mises stress is the stress description chosen in this case to drive the optimisation process. For elasticity, the *von Mises stress* can be formulated as

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

where σ_1, σ_2 and σ_3 are the *principal stresses*. The von Mises stress is a useful measure since it not only provides a failure criterion applicable to a wide range of ductile materials if it exceeds the material's yield stress, but also has the advantage of providing a single stress value to compute the stress situation in each node or element. This is a stress type commonly used in computational mechanics since a single contour plot provides a good overall picture of the stress state of a component, including the relation to this failure criterion for both tensile and compressive stresses.

Thus, the von Mises stress is computed for each node in the elastostatic boundary element analysis carried out. These nodes are sorted in ascending order according to their von Mises stress level. Material is removed from areas of low von Mises stress and added in areas of high von Mises stress. In other words, material can be removed from the structure if any node p satisfies equation (5.2)

$$\sigma_p \leq RR\sigma_{max} \tag{5.2}$$

and added to the structure if any node p satisfies equation (5.3)

$$\sigma_p \geq \sigma_y \text{ OR } \sigma_p \geq AR\sigma_{max} \tag{5.3}$$

where σ_p is the node von Mises stress, σ_{max} is the maximum von Mises stress, which varies along the process, 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 conceptually similar to the ones used in FE-ESO (Xie and Steven (1997)). In the same way as classical ESO, if a steady state is reached in which no nodes, or only a few nodes, can satisfy equation (5.2) then the RR is increased by the *evolutionary rate for removal*, ER_R, as follows

$$RR_i = RR_{i-1} + ER_R \tag{5.4}$$

where i is the current iteration. In a similar fashion, if only a few nodes can satisfy equation (5.3) then the AR is decreased by the *evolutionary rate for addition*, ER_A,

$$AR_i = AR_{i-1} - ER_A \tag{5.5}$$

Typical values for suitable ratios are $RR_0 = 0.01$, $ER_R = 0.01$, $AR_0 = 0.99$, $ER_A = 0$. These values are determined from numerical experience.

It should be noted that the von Mises stress is the criterion generally used in this work but, nevertheless, other stress criteria can be applied. For example, the principal stresses σ_1 and σ_2 can be considered to be taken into account the differences between tension and compression, which may be important in cases having two materials; for example, steel and concrete.

5.6.2 Distance to Move

In any iteration, material is either removed or added to the structure by changing its boundary definition. Since control points define the NURBS curves, a set of control

points is associated to each specific area to be moved. Every set consists of the three nearest control points to this area. The justification for this choice of three points comes from numerical tests with different numbers of control points. For sets of less than three control points it has been found that the distortion of the curve is very localised causing extreme curvature, while for sets of more than three control points the distortion is propagated to a wider region of the curve and therefore, losing control of the areas to be removed/added. These effects are especially noticeable in the first iterations of the process since, as it is explained in the next paragraph, the larger geometric changes are performed at the beginning of the process. Smoother shapes are obtained for sets of three control points due to the *local control feature* of the NURBS, which was presented in chapter 3. Also, due to the fact of using quadratic B-splines since the order of the curve determines the number of piecewise polynomials influenced by each control point. It was shown in section 3.1 (chapter 3) that for a quadratic curve only a maximum of three control points influence each curve segment.

Each set of control points is moved a distance related to the following parameters:

- Length of the least/most stressed element, L_e

- Distance of the set of three control points from the least/most stressed element. These distances are denoted a, b, c, respectively

- A factor related to the stress situation within the structure at the current iteration, which is called the *removal factor* (*RF*) if removing material, and the *addition factor* (*AF*) if adding it. This factor allows larger geometric changes in early iterations and smaller changes when fine-tuning is required in the later iterations

The *RF* and *AF* are features which ESO has never included. ESO is based mainly on once reached a steady state and then increasing *RR* (or decrease *AR*), in this case the material removal and addition is also controlled according to the stage of the process. Using this feature we gain in computational efficiency without losing the accuracy of the final solution.

For example, in figure 5.5 (a) the control point (P₂) situated at a distance a from the node of lowest stress is moved a quantity calculated as follows

$$\text{Movement of } P_2 = L_e \left(\frac{1}{\dfrac{1}{a} + \dfrac{1}{b} + \dfrac{1}{c}} \right) \frac{1}{a} RF \tag{5.6}$$

(a)   (b)
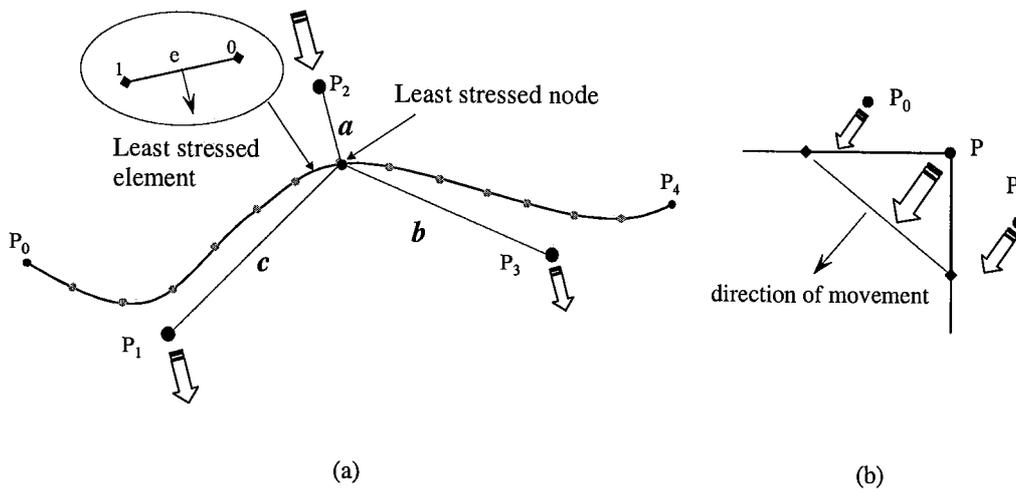
Figure 5.5: Direction of movement when removing material, one B-spline (a), two B-splines (b)

5.6.3 Direction of Movement

The direction of movement for each set is perpendicular to its nearest least/most stressed boundary element. The only difference between removal and addition is the inwards movement when removing material, shown in figure 5.5 (a) and outwards movement when adding it.

In the case of two B-splines, depicted in figure 5.5 (b), an interpolated normal to those NURBS curves is calculated. To get this normal quickly, first a line is obtained by joining the first node from the last element of the first B-spline with the last node from the first element of the last NURBS curve. The normal to this line gives us the direction of movement for the control points.

Note that the automatic meshing procedure used at each step aims to have equal, or close to equal, element lengths where two lines meet, so that this quick approximation to the average normal vector is likely to be a good approximation.

The *removal and addition of material process* detailed in this section can be summarised in the flow chart in figure 5.6. This flow chart improves the basic step 3 presented in figure 5.1.
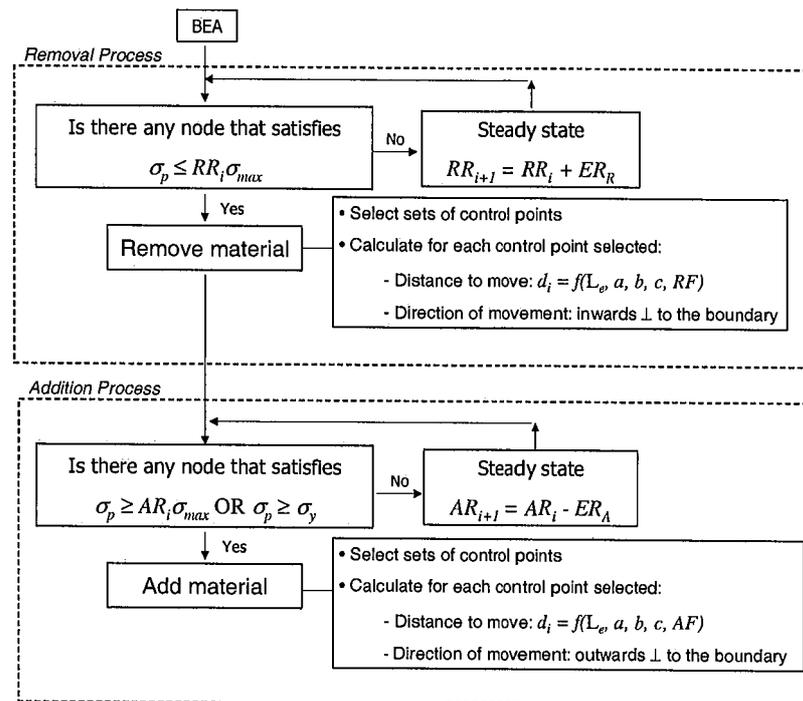
Figure 5.6: Flow chart of the removal and addition process

5.7 Geometry Control

The number of control points defining each NURBS curve is never fixed. On the contrary, it has been found to be important that control points are automatically, either inserted or deleted as required throughout the process. This capability has been implemented in order to control geometry changes and keep smoothness. Another important feature related to the geometry control process is the mesh subdivision. This subdivision is performed in areas in which the NURBS curve distortion has caused extreme curvature of an element. At this point, the optimisation algorithm is

enhanced with the control point insertion and deletion checks as well as mesh subdivision.

5.7.1 Insertion of Control Points

The insertion of a new control point in between two existing control points is done when these control points are moved away from each other more than a constant factor, k, times the initial distance between them

$$|P_j - P_{j-1}|_i > kd_0 \tag{5.7}$$

where i is the iteration number, $d_0 = |P_j - P_{j-1}|_0$, and generally $k \geq 1.5$.

The *control point insertion algorithm* is implemented to avoid areas whose defining control points are placed too far away and therefore, loosing control over the B-spline curve (in this area) since changes on their position are slightly reflected on the final curve. This effect is illustrated in figure 5.7. Figure 5.7 (a) depicts the original curve $C(u)$ with $d_0 = P_1 - P_2$. As the optimisation evolves former curve $C(u)$ changes according to the movement of its control points. In this case, P_1 is moving too separate from the curve compared to the other control points. Figure 5.7 (b) shows the curve $C(u)$ at iteration i for which the equation (5.7) is satisfied. Thus, $C(u)$ is updated to $C'(u)$ inserting a control point P' to the control point set. The new control point is placed at the mid-point of the line joining both control points P_1 and P_2.

(a)   (b)
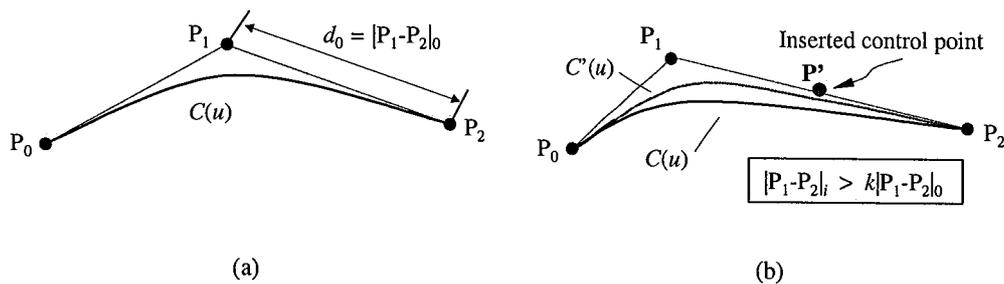
Figure 5.7: Insertion of control point P'

5.7.2 Removal of Control Points

The removal of a control point is performed when a set of control points are placed too close together. This *control point removal process* is implemented to avoid regions of excessive curvature as a result of the movement of control points in the removal and addition processes. Also, to keep each control polygon edge proportional to the overall edges. The criterion to remove the point from this set is based on the comparison of the different lengths of the lines of the control polygon

$$|P_j - P_{j\text{-}1}| < l|P_{j\text{-}1} - P_{j\text{-}2}| \text{ OR } |P_j - P_{j\text{-}1}| < l|P_{j+1} - P_j| \tag{5.8}$$

where $l < 1$, $|P_j - P_{j\text{-}1}|$ is the distance of the actual line checked, $|P_{j\text{-}1} - P_{j\text{-}2}|$ and $|P_{j+1} - P_j|$ are the distances of the previous and next control polygon lines respectively. Thus, if in equation (5.8), the inequality on the left is satisfied, then $P_{j\text{-}1}$ would be removed. On the other hand, if it is the inequality on the right that is satisfied then P_j is the control point to be removed.

Figure 5.8 (a) displays the original curve $C(u)$ and the distribution of control points defining the curve. Figure 5.8 (b) shows the curve $C(u)$ some iterations after, when P' and P_2 have been moved. At this stage P' has approached too close to P_3 and the distance between the two control points satisfies equation (5.8) since $|P_3\text{--}P'| < l|P_2\text{--}P'|$. Finally, $C'(u)$ is created by removing control point P' and so modifying the control polygon definition.

(a)   (b)
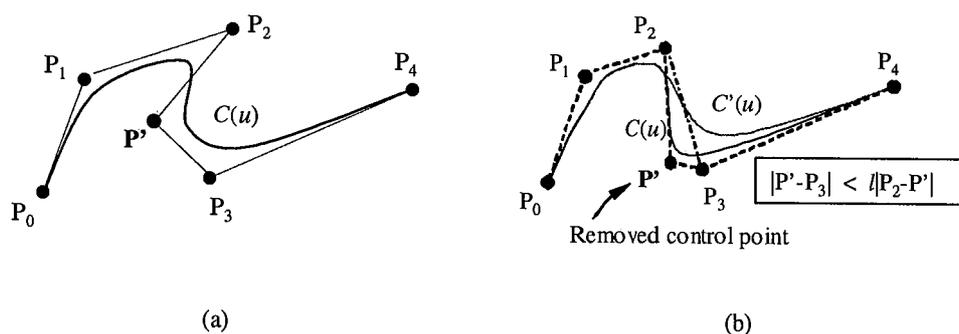
Figure 5.8: Removal of control point P'

One could argue here that to keep the geometry under control an alternative to adding or deleting control points would be to increase or decrease the degree of the curve. This is not a feasible solution since the structure is defined by several NURBS curves which are connected to each other and therefore they would behave differently if they had different degree. Moreover, an increase on the degree of the curve also increases its complexity. As discussed in chapter 3, section 3.1, if p is the degree of a B-spline curve, each segment of the curve is influenced by only $p+1$ control points, and conversely, each control point influences only $p+1$ curve segments. This *locality feature* would be affected by a degree increment since the higher the degree of the curve the more control points would influence a curve segment or in other words, changes of a control point would affect many more segments on the curve.

5.7.3 Mesh Subdivision

Mesh subdivision is performed in areas in which the NURBS curve distortion has caused extreme curvature of a single element. Since the optimisation algorithm is implemented together with the boundary element analysis the mesh is checked while the boundary is discretised and just before the structural analysis is carried out. Thus, if there is any element whose overall length exceeds certain factor of the distance between the edges of this element as stated in equation (5.9), then the element is subdivided

$$length_e > m|p_j - p_{j-1}| \tag{5.9}$$

where $length_e$ is the length of the element, e, under study. p_j and p_{j-1} are the first and last node of the element and $|p_j - p_{j-1}|$ is the distance between both nodes. m is a factor which generally, $m \geq 1.05$.

The *geometry control process* is summarised in the following flow chart in figure 5.9. This chart shows both the geometry control process related to *mesh subdivision* and the geometry control process related to *control point insertion and removal*. The mesh subdivision process is carried out after a boundary element analysis, whereas

the control point insertion and removal processes are performed after the removal and addition of material.
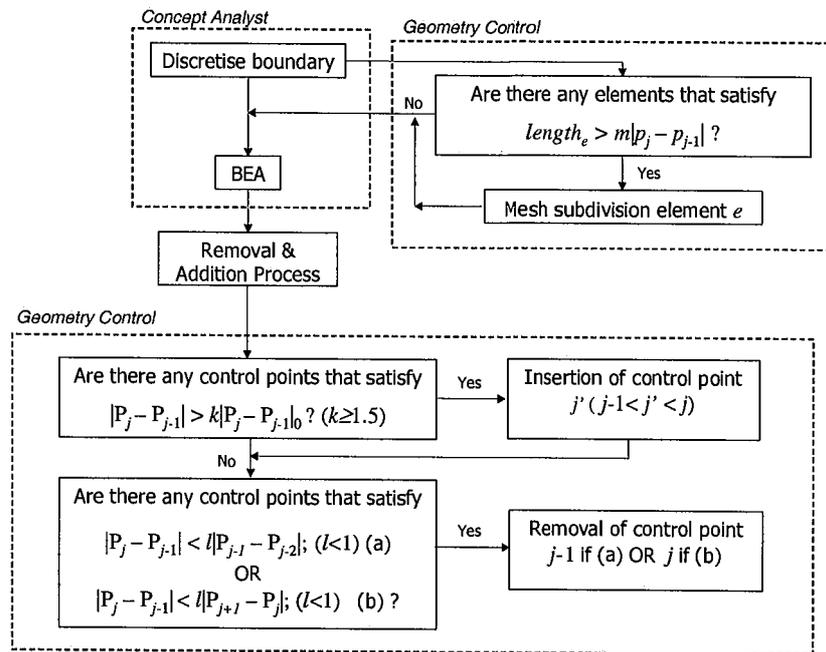
Figure 5.9: Flow chart of the geometry control process

5.8 Stopping Criteria

The failure of structures under service frequently takes place in areas of locally high stresses. Therefore, it is crucial to avoid stress peaks in order not only to prevent localised yielding but also to increase the fatigue life of the structure and avoid crack initiation. For this reason, the stopping criteria implemented in this shape optimisation algorithm are mainly related to the control of the stress level on the boundary. For the current work four different objective functions have been considered depending on the problem. Monitoring of the objective function towards, and ultimately beyond, an optimum provides a stopping criterion. This stopping criterion can be quantified in the following normalised form

$$\varepsilon = \left| \frac{f^{i+1} - f^{i}}{f^{i}} \right| \leq 10^{-4} \tag{5.10}$$

where f^i is the value of the objective function at iteration i and f^{i+1} is the value of the objective function one iteration after.

5.8.1 Stress Concentration Factor

The objective is to minimise the stress concentration factor on the boundary

$$f_k = \frac{\sigma_{max}}{\sigma_n} \tag{5.11}$$

where σ_{max} is the maximum stress at any point in the region of interest and σ_n is the nominal applied stress. For the majority of cases, in which the stress is applied on the non-design domain, this objective is equivalent to one in which the maximum stress is minimised.

5.8.2 Stress Levelling

The objective chosen is to make uniform the stress along a detailed part of the boundary (Γ). Therefore the objective deals with minimising the following function

$$f_\Gamma = \int_\Gamma (\sigma_p - \sigma_{ref})^2 d\Gamma \tag{5.12}$$

where σ_p is the node stress and σ_{ref} is a reference stress.

5.8.3 Weight Reduction

The objective function relates to the minimisation of the weight of the structure over the domain Ω, and is defined as

$$f_w = \int_\Omega \rho dV \tag{5.13}$$

where ρ is the density and V is volume of the structure. Since ρ is constant for these homogeneous material problems the equation (5.13) can be transformed and thus, the initial weight minimisation can be regarded as a volume minimisation

$$f_V = \int_\Omega dV \qquad (5.14)$$

5.8.4 Strain Energy Criterion

A common objective in engineering is to obtain a structure stiff enough to produce a good performance under certain loads and constraints. Such an optimisation problem consists of maximising the stiffness in the structure. This problem can be transformed by dividing the stiffness (K) by volume (V) and thus, a function termed *specific stiffness* (Steven *et al.* (2002)) is obtained

$$f_K = \frac{K}{V} \qquad (5.15)$$

Therefore, the overall specific stiffness is maximised which implies that the stiffness is maximised while the volume is reduced. An equivalent expression can be written in terms of strain energy (U) as

$$f = \frac{1}{UV} \qquad (5.16)$$

Using BEA notation, the strain energy U is defined as

$$U = \int_\Gamma \frac{1}{2} Tu \; d\Gamma \qquad (5.17)$$

where T are the tractions over the boundary and u the displacements over the part of the boundary Γ where the tractions are applied. Therefore, maximising the specific stiffness f_K of the structure is equivalent to minimising the function f_U where

$$f_U = UV \qquad (5.18)$$

5.9 Numerical Examples

5.9.1 Hole in a Biaxial Stress Field

This classical example addresses the shape optimisation of a hole in an infinite plate under a biaxial stress field. Due to the symmetry, only a quarter of the plate is modelled. Figure 5.10 shows the initial shape and boundary conditions. The plate is subjected to biaxial stress $\sigma_x = 20$ N/mm^2 and $\sigma_y = 10$ N/mm^2 under the plane stress assumption. The following isotropic material properties are assumed: Young's modulus $E = 210000$ N/mm^2, Poisson's ratio $v = 0.3$ and thickness $t = 1$ mm. The hole is modelled by a quadratic NURBS curve (line B-A).

Two optimal design problems are investigated for this example:

- Optimisation for stress levelling (equation 5.12)

- Optimisation for minimum volume (equation 5.14)
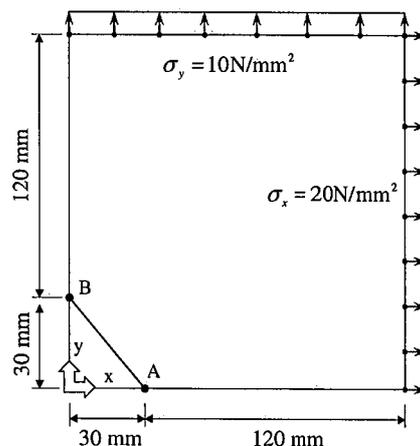
Figure 5.10: Problem definition for a quarter of the plate

5.9.1.1 Optimisation for Stress Levelling

The objective for this optimisation is to smooth out local stress concentrations, thus minimising the square of the deviation of the stresses from a desired uniform stress σ_{ref} (equation 5.12). In this case, the maximum von Mises stress σ_{max} is taken as σ_{ref}.

113

The theoretical solution (Kristensen and Madsen (1976)) is an ellipse with an aspect ratio b/a equivalent to the loading ratio σ_y/σ_x, i.e. 1/2, with the stresses uniformly distributed around the hole. This represents the optimum shape condition, in which all the material at the edge of the hole is at the same maximum permissible stress. The problem was also studied by Tafreshi and Fenner (1991) using the BEM and design sensitivity analysis. The sensitivity calculations were carried out by implicit differentiation of the structural response; i.e. the boundary integral equation is differentiated directly to give equations for the required derivatives.

The optimisation parameters are set to $RR_0 = 0.01$, $ER_R = 0.01$, $AR = 0.99$, $ER_A = 0$, $RF = 0.05$ and $AF = 0.025$. RF and AF are taken to be constant and of small value in order to keep smoothness and thereby do not give rise to artificial stress peaks over the boundary.

Initially, the element model contains 44 quadratic elements, a number that increases to only 45 in the optimised design. Figure 5.11 (a) shows the initial and final boundary mesh in the area of the hole. There is one quadratic NURBS curve defining the central hole. This curve consists initially of, 6 control points. At the end of the process this number has increased to 9. See figure 5.11 (b) for the initial and final distribution of the control points.

(a)   (b)
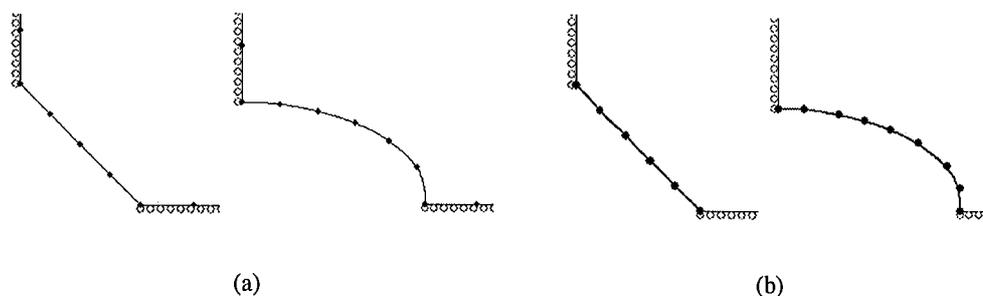
Figure 5.11: Initial and final element mesh (a). Control point distribution (b)

The final design for the cut-out is reached after 60 iterations in a total CPU time of 14.162 seconds on a Pentium 4 (2 GHz). Figure 5.12 compares the numerical optimum shape with the theory. In order to compare both results, the analytical optimum is calculated taking the major axis a = 47.8 mm. Since a = 2b for this

problem, the minor axis is b = 23.9 mm. Thus, the ellipse (analytical optimum) is calculated following the equation for an ellipse being centred at (0,0)

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \longrightarrow \frac{x^2}{(47.8)^2} + \frac{y^2}{(23.9)^2} = 1 \tag{5.19}$$

The numerical results show a slight deviation from the theoretical solution. The ratio of the minor axis of the elliptic hole to the major is b/a = 25.29 mm/45.81 mm $\approx$ 0.5521 (analytical optimum b/a = 0.5). However, the overall solution shows good agreement with the theory.
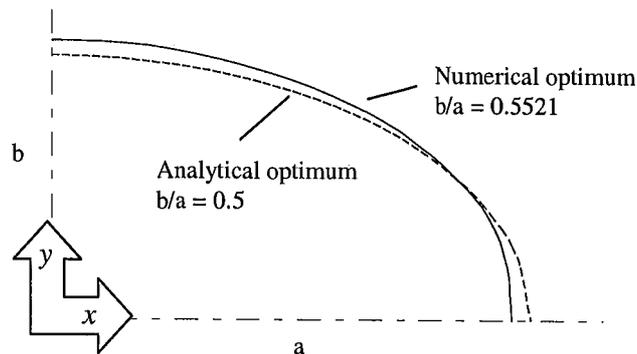
Figure 5.12: Comparison between the theoretical and analytical solution

Comparing the initial (figure 5.13 (a)) and final (figure 5.13 (b)) stress distribution, as expected, the stresses around the curve become more uniform when approaching the final solution. The ratio between the maximum von Mises stress at the edge of the hole and the larger of the two biaxial components is 1.691 (σ_{max}/σ_x = 33.828 N/mm^2/20 N/mm^2 = 1.691), which is high if compared to the analytical 1.5 (Kristensen and Madsen (1976)) and other numerical results 1.59 (Tafreshi and Fenner (1991)). A reason for this can be that the theoretical results are calculated assuming an infinite plane whereas in our case the geometry is described by finite segments and the effects of the loading and displacement constraints cannot be avoided.

enlarged cut-out

(a)

enlarged cut-out

(b)
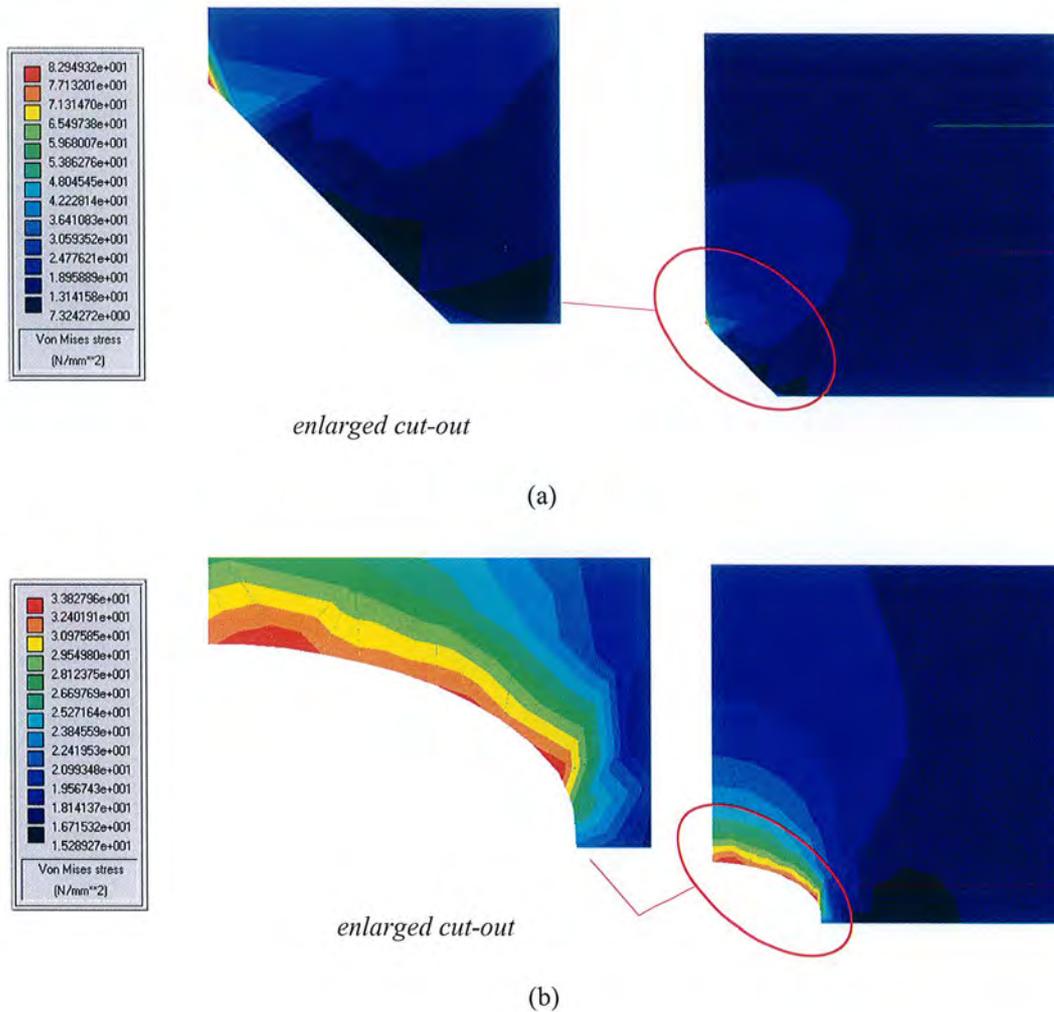
Figure 5.13: Initial (a) and final (b) von Mises stress contour plot

By using the stress based criterion; i.e. adding material to very high stressed areas and removing from lightly stressed regions, the boundary evolves into a state of near constant von Mises stress at its surface. The structure reduces any localised stress peaks which leads to a constant stress state at the surface. This situation is observed in figure 5.13 (b) especially in the enlarged view of the final elliptical shape.

Figure 5.14 shows the distribution of the von Mises stress along the hole for the initial and final designs. In the arbitrary initial design there is a stress peak of 80 N/mm^2 that occurs in a localised area at $\theta = 0°$. As we move along θ, the von Mises stress drops sharply to 10 N/mm^2 and only increases nearby $\theta = 90°$; i.e. at the right end of the hole. In figure 5.14 we compared these results to the optimum design; in the latter, the stress concentration has been effectively removed and the stress levels

116

are clearly uniform around the cut-out. Moving from $\theta = 0°$ to $\theta = 80°$ the von Mises stresses smoothly vary between the values of 30 N/mm² and 33 N/mm². It is only nearby the right end of the hole that the stresses drop to 22 N/mm² ($\approx$ 31 % from the uniform stress average 32 N/mm²). The optimum analytical solution (Kristensen and Madsen (1976)) $\sigma_\theta = (1 + \sigma_y / \sigma_x) \sigma_x$, displayed in red, has a constant value $\sigma_\theta = 30$ N/mm². Thus, there is a very good general agreement between the analytical and the numerical optimum.
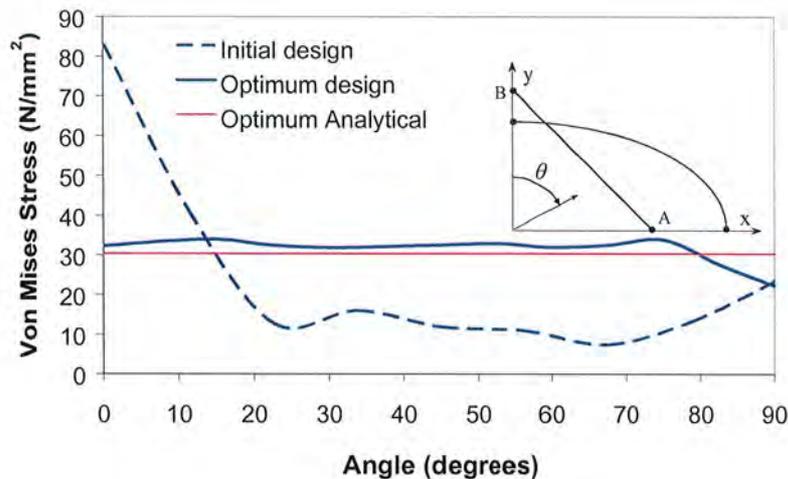
Figure 5.14: Distribution of von Mises stress along the hole

5.9.1.2   Optimisation for Minimum Volume

The same plate example is now optimised for minimum volume. The purpose of this optimisation is to remove as much material as possible from around the hole while avoiding yielding (a yield stress of 280 N/mm² is adopted). In addition to this maximum stress criterion, since the plate is considered to be infinite, the stress level in the constrained edges has to be smaller than a percentage (50%) of the maximum stress in the plate. The optimisation parameters are set to $RR = 0.01$, $ER_R = 0.01$, $AR = 0.99$, $ER_A = 0$, $RF = 0.2$ and $AF = 0.5$.

The standard boundary element mesh consists of 44 quadratic elements (figure 5.15 (a)) at the initial design, and 46 elements (figure 5.15 (b)) at the end of the process. The hole is described by a NURBS curve, initially with 6 control points (figure 5.16 (a)). The number of control points has increased to 10 at the end of the process

(figure 5.16 (b)). Figure 5.17 depicts the von Mises stress distribution, initial (a) and final (b) after performing 65 iterations. The optimum shape for volume minimisation has no longer an elliptical shape as the previous example. This is due to the local bending in the right corner, which produces high stresses and therefore and addition of material. As a result of this addition of material, the structure tends to increase the second moment of area[*]. Similar results for this problem were found by Esping (1985) and Sienz (1994).

(a)   (b)
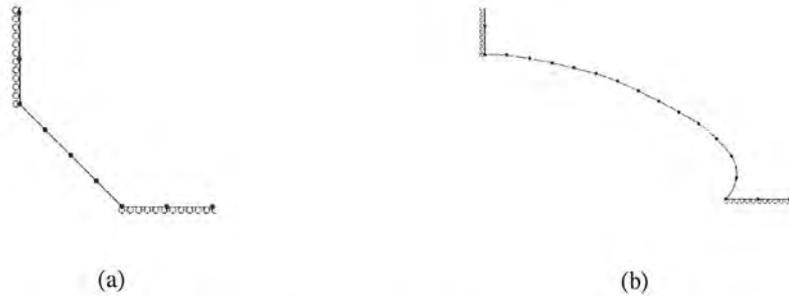
Figure 5.15: Initial (a) and final (b) boundary element mesh

(a)   (b)
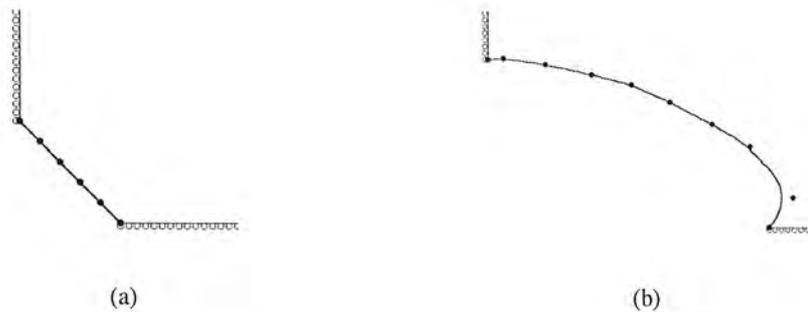
Figure 5.16: Initial (a) and final (b) control point distribution

---

[*] Second moment of area about the z-axis $I_z = \int_A y^2 dA$

(a)   (b)
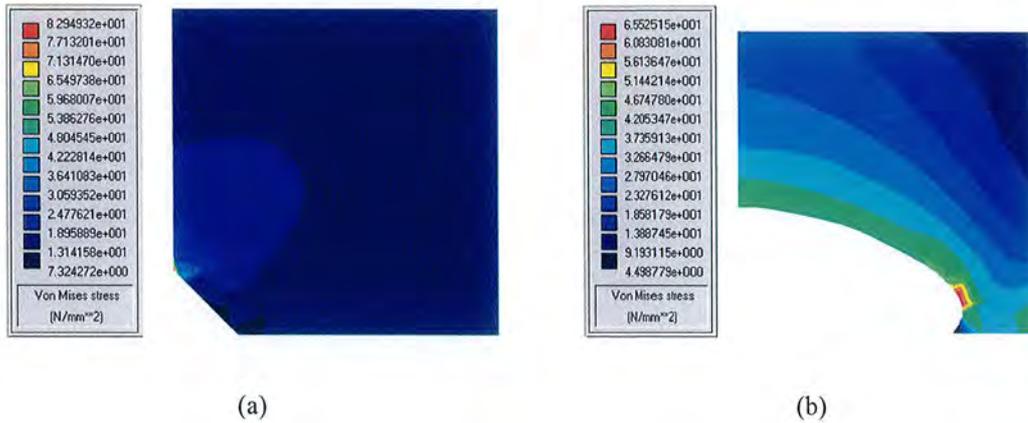
Figure 5.17: Initial (a) and final (b) von Mises stress distribution

5.9.2 Fillet

The second example deals with the minimisation of the stress concentration (equation 5.11) around a fillet, or in other words, to find the shape for which the maximum stress on the fillet is a minimum. This problem has been studied by Francavilla *et al.* (1975) using penalty-function methods and extended to axisymmetric shapes by Pedersen and Laursen (1983). Also, the fillet design has been investigated by FE-ESO (Xie and Steven (1997)). Figure 5.18 (a) shows the shaft with shoulder fillet loaded in tension; due to the symmetry only half of the structure is modelled. A priori, the shape of the transition area is not known and it is therefore modelled initially as a rectangular shape with L/H = 2; being L = 200 mm and H = 100 mm. Figure 5.18 (b) depicts the *changeable boundary* as displayed by Concept Analyst.

The material properties assumed are: Young's modulus $E = 210000$ N/mm^2, Poisson's ratio $\nu = 0.3$ and thickness $t = 1$ mm. The structure is in pure tension due to a stress of magnitude 100 N/mm^2 applied on the right-hand side.

(a)   (b)
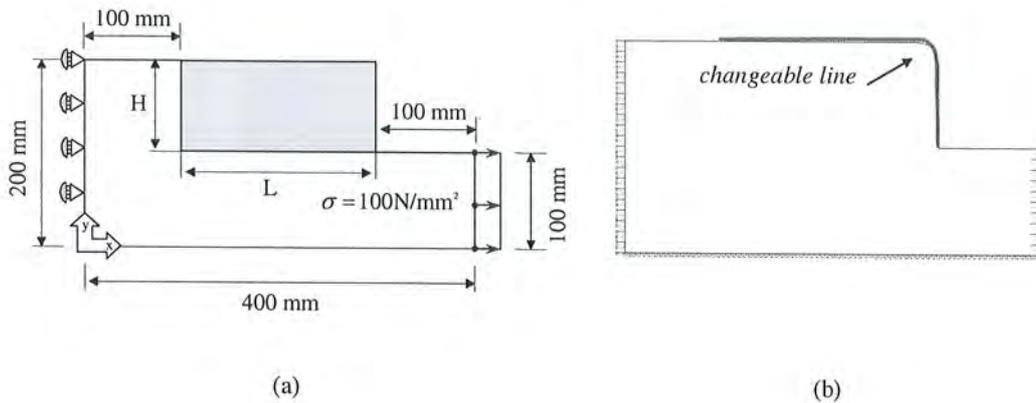
Figure 5.18: Problem definition for a fillet

The optimisation parameters are set as $ER_R = 0.01$, RR initially 0.01 but increasing gradually throughout the process to 0.2, $ER_A = 0$, $AR = 0.99$, $RF = 0.02$ and $AF = 0.025$. These small and constant values for RF and AF produce a slow removal and addition, respectively, so that smooth changes are ensured alongside the fillet. The optimum design is accomplished after performing 123 iterations within a CPU time of 21.876 seconds in a Pentium 4 (2 GHz).

(a)   (b)
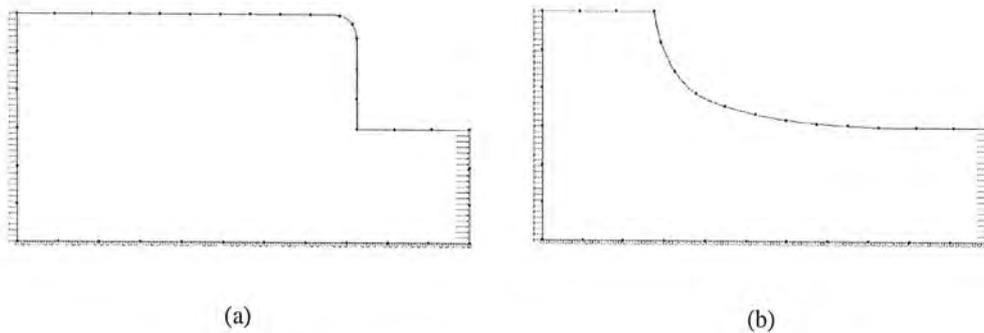
Figure 5.19: Initial (a) and final (b) boundary element mesh

The standard mesh consists of 38 quadratic elements in the initial design (figure 5.19 (a)), and 35 in the final design (figure 5.19 (b)).

(a)   (b)
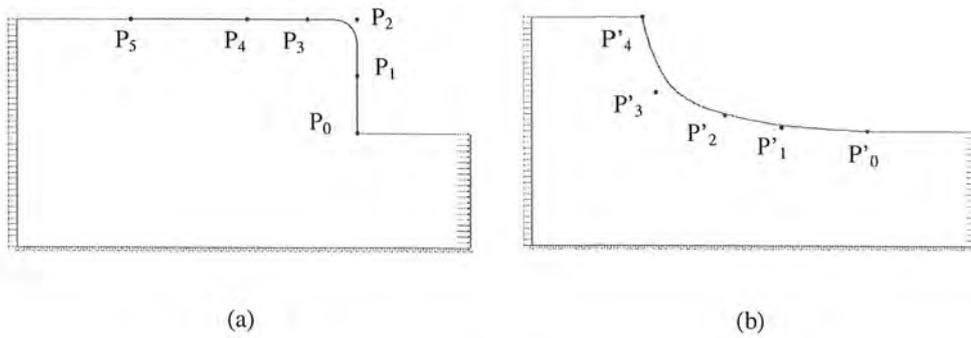
Figure 5.20: Initial (a) and final (b) control point distribution

P	X (mm)	Y (mm)
P_0	300	100
P_1	300	150
P_2	300	200
P_3	256	200
P_4	203	200
P_5	100	200

P'	X (mm)	Y (mm)
P'_0	300	100
P'_1	223.42	102.93
P'_2	173.21	113.42
P'_3	110.82	133.53
P'_4	100	200

Table 5.1: Coordinates of the control points

The fillet is described by a quadratic NURBS curve with six control points. The weights associated to each control point are taking equal to 1; i.e. $w_i = 1$ (see chapter 3). Figure 5.20 shows the control point distribution for the initial (a) and final (b) models. Notice that the number of control points has decreased to five. Such a phenomenon occurs as a result of the application of the *control point removal algorithm* following equation (5.8) with $l = 1.5$. The insertion of control points is controlled following equation (5.7) for $k = 2$ but for this example this part of the algorithm does not turn out to be required. The coordinates of these control points are displayed in table 5.1. Points P_0 and P_5, initial and final, respectively, control points in the control point polygon, are fixed since they belong to the changeable line but also to the neighbouring lines which are *non-changeable*. Note that the control point P_5 is latter identified as P'_4 in the final design.
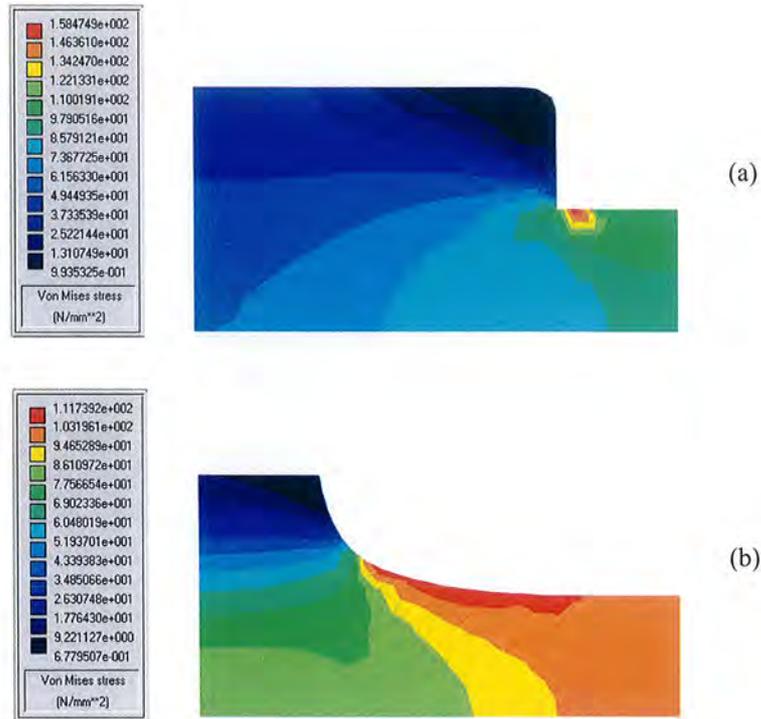
Figure 5.21: Initial (a) and final (b) von Mises stress distribution

Figure 5.21 depicts the von Mises stress distribution. In the initial design (a), the highest stress is concentrated at the bottom right side of the fillet. As it can be noticed, the stress peak is not at the corner. This is due to an artifact of the stress averaging of Concept Analyst since the two elements at the corner have different traction situation. The element situated on the vertical line of the corner is under low tractions while the other element, on the horizontal line of the corner, has very high tractions. Numerical tests refining the elements in area have shown that this effect is reduced and the stress peak is much closer to the actual corner. The von Mises stress maximum is 158.47 N/mm^2; this value of the maximum von Mises stress is reduced to 111.73 N/mm^2 in the optimum proposed design figure 5.21 (b). Therefore, the stress concentration in the fillet has been reduced by approximately 30%. From the von Mises stress distribution displayed in figure 5.21 it should be noted that a larger control area would have resulted in less stress variation at the design domain and therefore, a more homogeneous stress distribution.

The distribution of von Mises stress along the fillet is shown in figure 5.22. The NURBS curve defining the fillet varies from coordinate $x = 100$ mm to coordinate x

= 300 mm. However, in order to show the effect of low and high stress situation in the right hand side corner of the fillet (explained at the beginning of this paragraph); the von Mises stress in the boundary nearby the corner ($x > 300$) is also displayed.
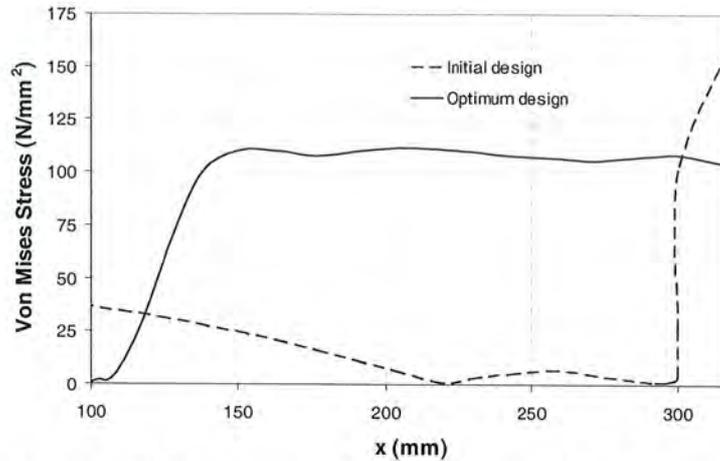
Figure 5.22: Distribution of von Mises stress along the fillet

Figure 5.23 shows the evolution history of the stress concentration factor K_t. The minimum for this objective is clearly obtained at iteration 122. However, different local minima can also be identified in the evolution of the graph. These peaks or jumps in the objective are due the fact that the removal and addition of material is based on a stress criterion rather than driven by the objective function.

Figure 5.25 shows the evolution process of volume ratio. The minimum stress is obtained for a volume ratio $V/V_0 = 17\%$. There is a similarity between the resulting shape and the one obtained using FE-ESO in figure 5.24 (b) where the minimum stress is found at the volume ratio of 12%. However, FE-ESO solution bears a strong resemblance to an ellipse whereas the BE-ESO result is closer to a parabola shape. Some authors (Francavilla *et al.* (1975)) have obtained the optimised shape by assuming a transition low-order curve and tangential straight lines in the definition of the initial problem. As a result of such restrictions, the optimal shape generally becomes an ellipse rather than a general parabola. In our case, the control points do not follow any predetermined direction but simply move perpendicular to the nearest boundary. No bounds have been imposed to the control points apart from fixing the first and last control point.
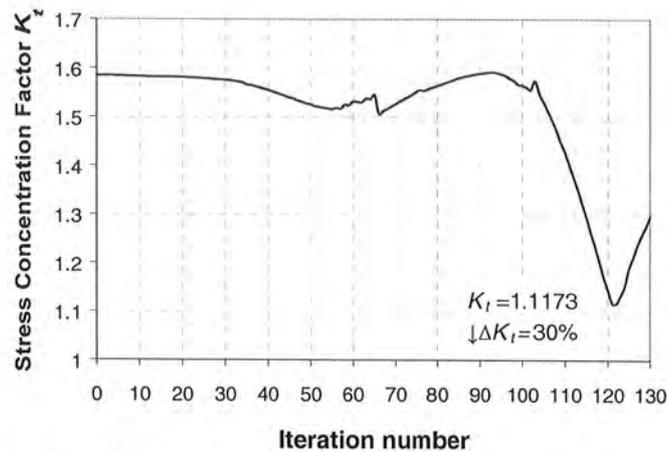
Figure 5.23: Evolution history of stress concentration factor K_t

(a)   (b)
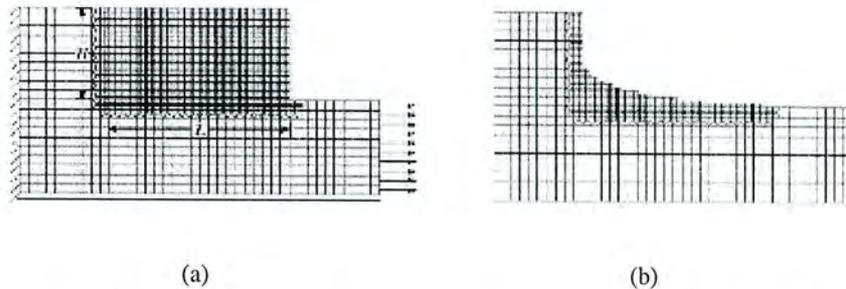
Figure 5.24: FE-ESO (Xie and Steven (1997). Initial design (a). $V/V_0 \approx 12\%$ minimum stress (b)

The same fillet problem is investigated but imposing bounds to the movement of the control points. Thus, displacement constraints are applied to restrict the movement of any control point inside the design domain area, displayed in grey in figure 5.26. The same material properties, optimisation parameters, dimensions, loads and constraints, as depicted in figure 5.18, are assumed. However, the design domain in this case it is not a rectangle but a triangle, since the *changeable line* is now the hypotenuse as shown in figure 5.26.
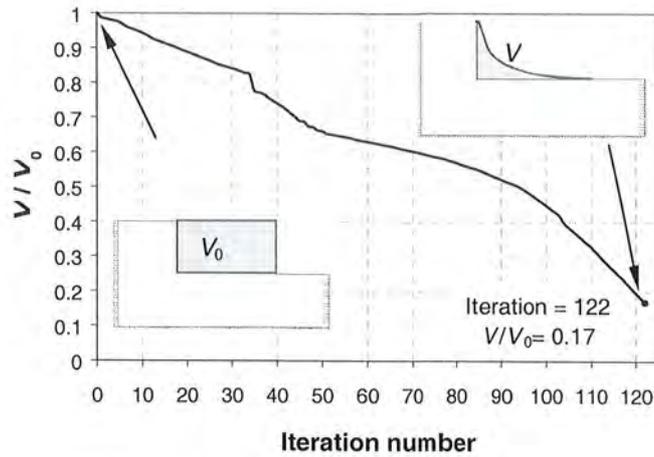
Figure 5.25: History plot of volume ratio
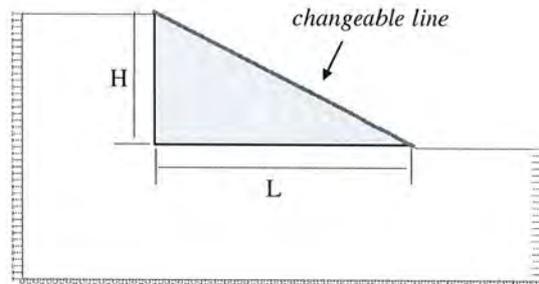
Figure 5.26: Problem definition for a fillet

The boundary element mesh comprises 34 quadratic elements in the initial design (figure 5.27 (a)), and 36 elements in the final design (figure 5.27 (b)). The fillet is described by a quadratic NURBS curve with seven control points equally distributed along the fillet. Figure 5.28 shows the control point distribution for the initial (a) and final (b) models. The number of control points in the resulting model has increased after applying the *control point insertion algorithm*; also notice that the control points accomplish the constraint of being within the design domain.
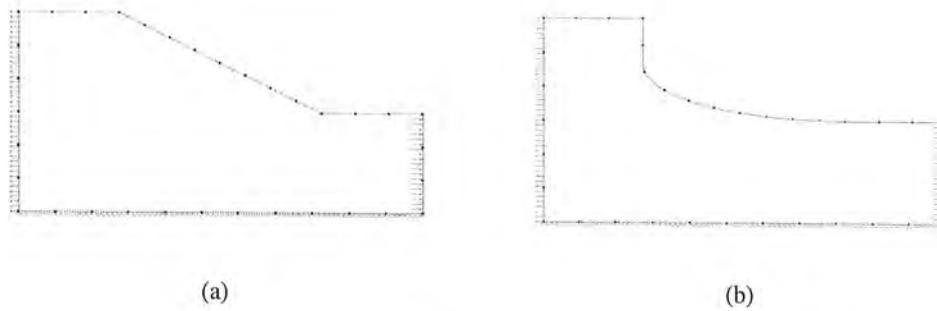
Figure 5.27: Initial (a) and final (b) boundary element mesh
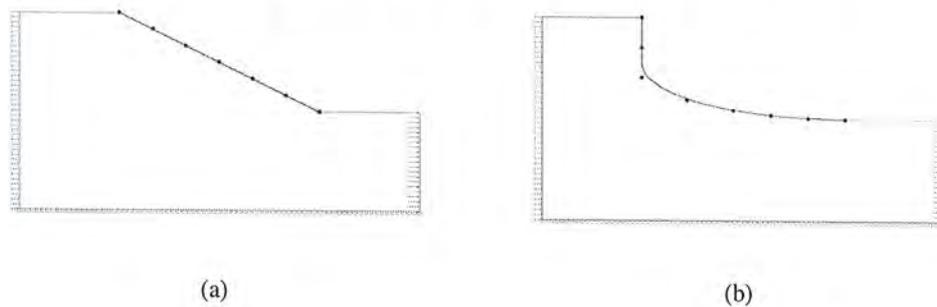
Figure 5.28: Initial (a) and final (b) control point distribution

The optimum design is obtained after 75 iterations in a CPU time of 13.23 seconds. Figure 5.29 depicts the von Mises stress distribution. In the initial design (a), there is a high stress peak of 162.6 N/mm^2; after the optimisation this value is reduced to 108.6 N/mm^2 in the optimum proposed design (b). Figure 5.30 shows a comparison between the initial and final stress distribution around the fillet. It is clear that the stress peak, initially localised at $x = 300$ mm, has been reduced by 33.2%. The volume ratio is $V/V_0 = 0.1182$; where $V_0 = H \times L$ in order to compare to FE-ESO and to the previous result for the fillet. Both final shape and volume ratio show a very good agreement with the results obtained by Xie and Steven (1997) and displayed in figure 5.24 (b).
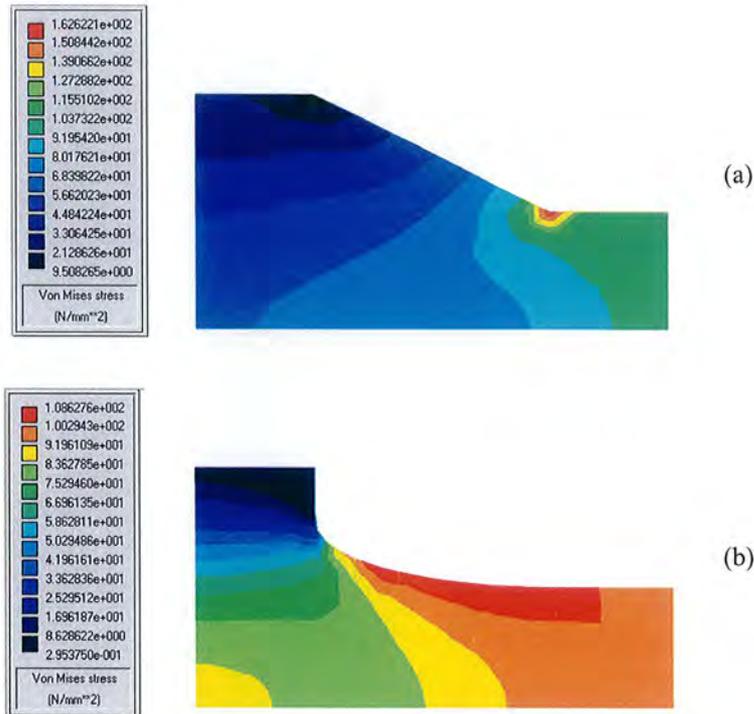
Figure 5.29: Initial (a) and final (b) von Mises stress distribution
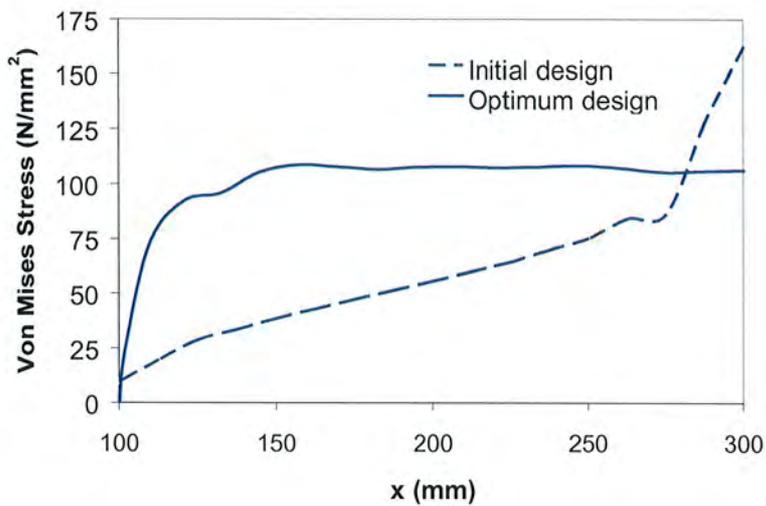
Figure 5.30: Distribution of von Mises stress along the fillet

Finally, the two proposed designs are compared in figure 5.31. The dashed line (final design I) is the fillet initially obtained without applying any restriction to the control points but the first and last control point fixed. In the second case (final design II) the control point movement is restricted to the design space. Evaluating both results, the minimum value for the maximum von Mises stress is obtained in the second solution

127

(final design II). Moreover this solution is closer to the optimum proposed by Xie and Steven (1997) and by Francavilla *et al.* (1975), and also exhibits a better volume reduction.
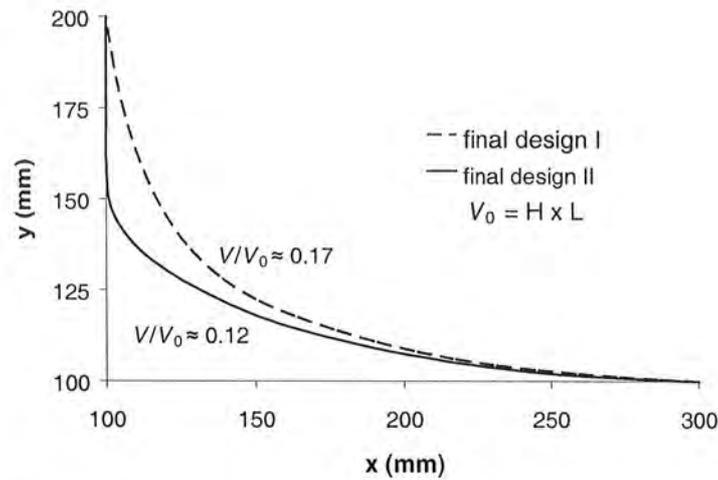
Figure 5.31: Comparison of fillets according to the constraints imposed

5.9.3 Two-bar frame

The last example of this chapter shows the optimisation of a rectangular beam subjected to a vertical load of magnitude 100 N. The definition of the initial problem is shown in figure 5.32. The height of the beam (100 mm) is twice the width (50 mm). The material properties are: Young's modulus $E = 210000$ N/mm², Poisson's ratio $v = 0.3$ and thickness $t = 1$ mm. The optimisation process follows the strain energy criterion defined in equation (5.18). The analytical solution for this problem is a two-bar frame structure (Hemp (1973)).

The optimisation parameters are set to $ER_R = 0.01$, and RR, initially 0.01, rises up to 0.26 during the iterative process. RF also varies from 0.3 to 0.06. $AF = 0.5$ and a constant value of $AR = 0.75$ is adopted.

The applied load of 100 N is deliberately chosen to take a small value so that the material addition is inhibited, the yield stress being unlikely to be reached. This allows a more direct comparison with the analytical optimum presented by Hemp (1973).
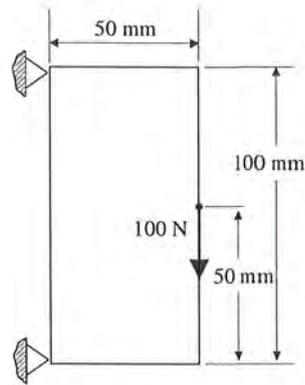
Figure 5.32: Problem definition for a two-bar frame

Figures 5.33 (a) and 5.33 (b) show the boundary element mesh used for the analysis. The initial boundary element model contains 35 quadratic elements but at the end of the optimisation process this number has increased to 43. The final geometry depicts a structure similar to a two bar truss with an internal angle of 90°, which is the analytical solution. In figure 5.33 (b) the *mesh subdivision algorithm*, explained in section 5.7.3, is applied to the element with extreme curvature.

(a)   (b)   (c)   (d)
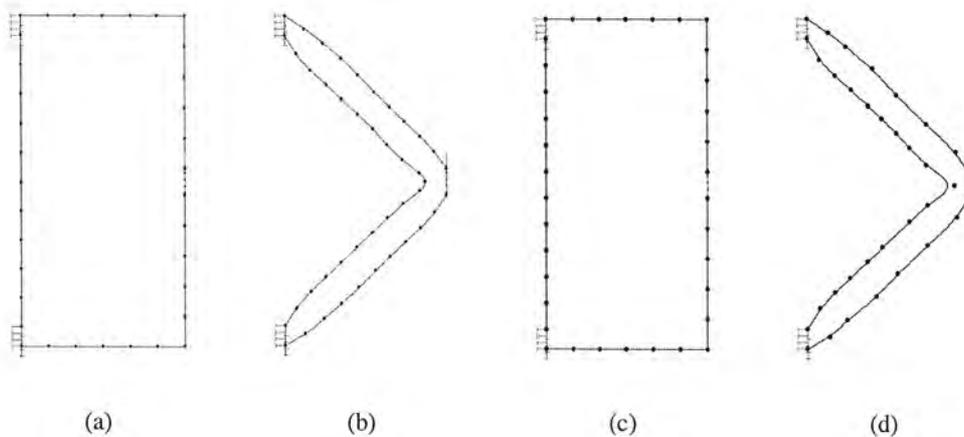
Figure 5.33: Initial (a) and final (b) boundary element mesh. Initial (c) and final (d) control point distribution

Figures 5.33 (c) and 5.33 (d) show the number and position of the control points related to the five quadratic NURBS curves describing the changeable boundary shape. Notice from figure 5.33 (c) that initially all curves are modelled as straight lines.

Figure 5.34 displays the evolution of the process according to the von Mises stress contour plot. The optimum is reached after 43 iterations in 11.68 seconds using a Pentium 4 (2 GHz) processor. Comparing the initial geometry and the final one; in the final design there is approximately a 75% reduction from the initial volume. It should be noted that this final design is approaching a *fully stressed design*, in which ideally all the material is at the same stress level.

iteration 0, $V/V_0 = 1$

iteration 7, $V/V_0 = 0.75$

iteration 19, $V/V_0 = 0.50$

iteration 42, $V/V_0 = 0.25$
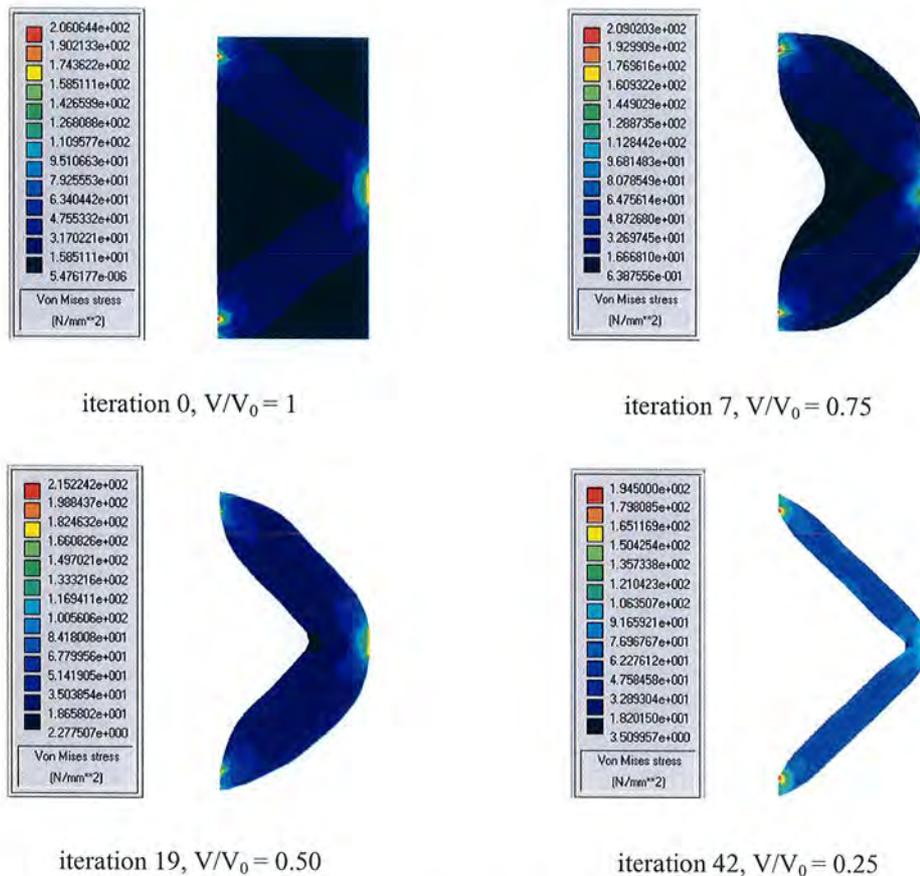
Figure 5.34: Von Mises stress contour plot for several steps in the process

5.10 Discussion and Conclusions

The problem of shape optimisation has been successfully tackled using the new ESO approach. Initial structural applications demonstrate the ability of the developed method to produce optimal shapes based on a stress-based criterion. The method identifies and removes material that does not contribute to increase the performance

of the structure according to the sought objective, as well as adding material to critical regions. Smooth boundaries are obtained by working directly with the control points defining the geometry.

In a similar way to the *biological growth method* (Mattheck and Burkhardt (1990)) a homogeneous state of von Mises stresses is obtained on the surface of the structure considered. Small movements of the control points defining the NURBS curve under design tend to reduce the maximum von Mises stress and to increase the minimum von Mises stress in order to get a more homogeneous von Mises stress distribution. In this sense, localised stress peaks are avoided in the structure and eventually a constant stress state is reached at the surface.

The boundary movements do not produce mesh distortion problems since a remeshing is performed for each analysis. This remesh process is only required over the boundary and therefore it is simple and fast, which is one of the advantages of the boundary element analysis.

A practical application of the method is the optimisation of holes in plates under in-plane loading. Two different objectives are considered, i.e. stress levelling and weight reduction. As expected, the shape results are different depending on the objective to be achieved. The results agree with both theory and numerical results obtained using other optimisation methods. Table 5.2 shows a chart comparing von Mises stress levels and volume results for the two optimisation objectives considered.

Initial V_0 (mm^3)	22050	
Final V_f (mm^3)	Stress levelling	Volume
	21577.69	17069.50
Initial σ_{max} (N/mm^2)	82.95	
Final σ_{max} (N/mm^2)	Stress levelling	Volume
	33.828	65.52

Table 5.2: Stress levelling vs. volume minimisation

The problem of a fillet on a tension bar is optimised to minimise the stress concentration factor. Again, the proposed fillet shape exhibits very good agreement with benchmark results. Moreover, the algorithm has been proved to be effective for specific strain energy minimisation. By integrating structural analysis and optimisation the computational performance shows very good performance and fast solution times.

6

TOPOLOGY OPTIMISATION IN 2D

6.1 Overview

In this chapter the optimisation approach is applied to topology optimisation problems. The approach has been proved to be successful for shape optimisation problems (chapter 5). Hence, the algorithm for topology optimisation is implemented in a similar way to the shape optimisation case. Likewise, the algorithm is based on the evolutionary structural optimisation (ESO) method. The boundary element method (BEM) is used as the analytical tool and a boundary representation is implemented in terms of NURBS curves. Furthermore, holes can be created in the inner low stressed areas to remove inefficient material from the inside regions and, therefore, performing topological changes. These holes are also defined by NURBS. Accordingly, they exhibit the same behaviour features as the outside boundary. Several examples are presented to prove the efficiency of the algorithm.

6.2 Introduction

In the field of structural optimisation, topology optimisation refers to optimal design problems where the topology of the structure varies. This optimisation class is regarded as one of the most challenging optimisation problems since the topology is allowed to change in order to increase the performance of the structure. For this

reason, much research has been devoted to topology optimisation over the last ten years. This has included methods based on microstructures such as the *homogenisation method* (Bendsøe and Kikuchi (1988)) and *SIMP method* (Rozvany, *et al.* (1992)), and methods that deal with elements on a macro basis approach such as the *biological growth* (Baumgartner *et al.* (1992)), the *evolutionary structural optimisation* (ESO) *method* (Xie and Steven (1997)) and the *bubble method* (Eschenauer *et al.* (1994)). Most recently, probabilistic methods such as *genetic algorithms* (GAs) (Cerrolaza *et al.* (2000)) have been applied to topology optimisation since they are particularly robust in finding global optima. However, this can come at a high computational cost. Also, the concepts of *robust* and *reliable design* (Frangopol and Maute (2003)) are in growing interest in topology optimisation since they consider input variations and uncertainties in the design and manufacturing process. From these different topology optimisation techniques, presented in detail in chapter 2, the evolutionary structural optimisation method ESO (Xie and Steven (1997)) is chosen as the basis for the present work.

Most of the work carried out on ESO is based on the finite element method. The classical ESO approach is based on the idea of removing inefficient material from an initially oversized domain. The removal process is carried out deleting regions occupied by elements with low stresses. By repeating this process and removing small amounts of material at each stage, the topology for the structure gradually evolves to a more efficient shape. After this basic approach there have been a number of modifications and refinements such as BESO (Querin *et al.* (1998)) where not only elements are removed but also added in high stress areas. Recent applications of ESO (Steven *et al.* (2002)) would consider element modification sensitivity terms applied in a wide range of physical situations. These sensitivities are used to drive the removal and addition process in order to achieve a minimum (or maximum) of some objective/s or to target certain fitness values.

Nevertheless, ESO presents some drawbacks and weaknesses as have been reported by Zhou and Rozvany (2001). Another problem already reported in the previous chapter (chapter 5) is the presence of jagged edges and checkerboard patterns resulting from the dependency on the finite element model. These problems are

134

illustrated in figure 2.1 (see chapter 2). Although increasing the density of the mesh improves the quality of the boundary definition, it also implies longer solution times.

To cope with these shortcomings, the boundary element method (BEM), detailed in chapter 3, is used as the analytical tool. The boundary element software used in topology problems is the in-house program Concept Analyst (Trevelyan and Wang (2001)). In addition, the boundary shape is represented by NURBS curves defined by control points. These curves are explained in detail in chapter 4. The present approach optimises the structure by treating the control points as the design variables. Therefore, the boundary smoothness is achieved by migrating control points instead of deleting and adding elements.

6.3   Algorithm

The basic topology optimisation process can be summarised in the flow chart shown in figure 6.1. The optimisation process follows the same stages already presented in shape optimisation (figure 5.1). However, a new step is implemented in the topology optimisation process related to the creation of holes in the inner low stressed areas. The main steps of the basic algorithm are summarised as follows

Step 1: The geometry of the structure is defined. The boundary lines able to change, so-called *changeable lines*, are selected and modelled as NURBS curves. Loads and constraints are applied to the initial design problem.

Step 2: A boundary element analysis (BEA) is carried out.

Step 3: The *material removal process* is performed by selecting the least stressed nodes within the boundary mesh and effectively moving the control points nearest to those nodes. Following a similar process, *material addition* is carried out if a node is found with a stress higher than the yield stress or a certain maximum stress criterion. The control points nearest to the high stressed nodes would be moved. However, movement takes place in the opposite direction to that in the removal process.

Step 4: To complete the topology optimisation, the inside regions are also considered and internal holes are created in the inner low stressed areas. The algorithm considers both the external and internal boundaries to be optimised at the same time. Under certain circumstances, holes can merge between each other or into the outer boundary. Thus, if the topology changes, i.e. a hole is created or merged, then the process is repeated from step 2.

Step 5: Finally, such a procedure is repeated, from step 2, until the evolution of the objective function shows no improvements.

Notice that since, during the optimisation process, the control points are moved and in consequence the shape changes, the boundary element mesh is updated every iteration.
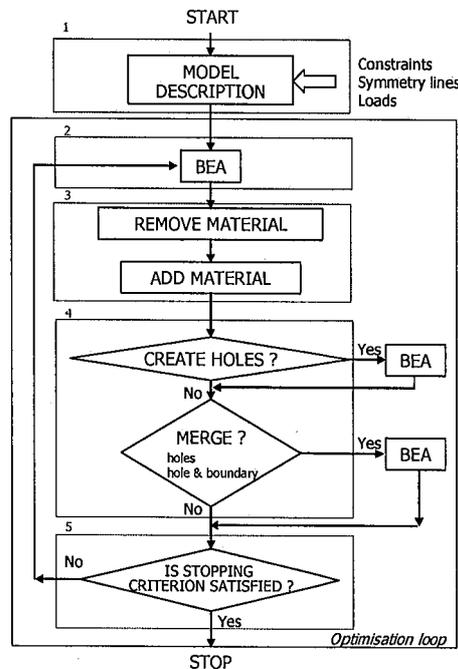
Figure 6.1: Flow chart of the optimisation process

This basic topology optimisation algorithm is very similar to the one presented for shape optimisation (figure 5.1). Therefore, only the distinctive features are covered in this chapter. The reader is referred to chapter 5 for a more detailed explanation about the different stages of the process not considered here.

6.4 Removal and Addition of Material

6.4.1 Identifying Inefficient Areas

In a similar way to shape optimisation, the von Mises stress is computed for each node after the boundary element analysis is carried out. These nodes are sorted in ascending order according to their von Mises stress level. As a result, material can be removed from the structure if any node satisfies

$$\sigma_p \leq RR\sigma_{max} \tag{6.1}$$

and added to the structure if any node satisfies the following equation

$$\sigma_p \geq \sigma_y \quad OR \quad \sigma_p \geq AR\sigma_{max} \tag{6.2}$$

where σ_p is the node von Mises stress, σ_{max} is the maximum von Mises stress, which varies along the process, σ_y is the yield stress, AR is the *addition ratio* $0 \leq AR \leq 1$ and RR is the *removal ratio* $0 \leq RR \leq 1$. These ratios are similar to the ones defined in ESO (Xie and Steven (1997)) and are called the *rejection rate* (RR) and *addition rate* (AR) (Querin *et al.* (1998)), respectively. If a steady state is reached in which no nodes, or only a few nodes, can satisfy equation (6.1) then the RR is incremented by the *evolutionary rate* ER_R. Similarly, if equation (6.2) is not satisfied then the AR is reduced by the *evolutionary rate* ER_A

$$\begin{aligned} RR_{i+1} &= RR_i + ER_R \\ AR_{i+1} &= AR_i - ER_A \end{aligned} \tag{6.3}$$

Typical values for suitable ratios are $RR_0 = 0.01$, $ER_R = 0.01$, $AR_0 = 1$ and $ER_A = 0$. These values are determined from numerical experience.

The distance to move the selected control points as well as the direction of movement for each control point follow the details given in sections 5.6.2 and 5.6.3 in the previous chapter. Geometry control checks (section 5.7) are also crucial in the topology optimisation approach.

6.5 Creation of Holes

Topology changes are brought about by creating internal cavities or holes in low stressed areas. These holes are shaped according to the internal von Mises stress contours. The free-shapes are created using NURBS curves to define their geometry.

In a BEA the results at points contained within the boundary of the object are calculated at *internal points*. In the software used in the current study (Trevelyan and Wang (2001)), these internal points are scattered at random (though concentrated near stress concentrations and their distribution smoothed using a Laplacian algorithm). It is the results at these points, and not at the boundary nodes, that form the starting point of the algorithm for the creation of holes in the domain.

The basis to identify the low stress areas is similar to the one used for the outside boundary (equation 6.1) but satisfying equation (6.4). Internal points that satisfy this equation are selected and put in a stack of internal points, *stackIP*. This *stackIP* is sorted in ascending von Mises stress order. Starting from the least stressed internal point, the adjacent internal points within a stress level lower than a *threshold stress* are searched. The internal points dealing with the creation of the new contours are removed from the *stackIP* to a temporal stack, *temporalstackIP*. The surrounding contour line is built as the convex polygon of internal points from the *temporalstackIP*. This contour is described by NURBS curves. The polygon is split into two pieces. Furthermore, the internal points are the control points defining the new NURBS. This procedure is repeated until there are no more internal points in the *stackIP*. The *hole creation process* is presented in the flow chart in figure 6.3 and summarised in the following steps

Step 1: Check for any internal point that satisfies equation (6.4). If so, add this internal point to the *stackIP*

$$\sigma_{IP} \leq RR\sigma_{\max} \tag{6.4}$$

where σ_{IP} is the internal point von Mises stress, σ_{max} is the maximum von Mises stress and RR is the removal ratio used in equation (6.1)

Step 2: The *stackIP* which consists of a group of selected internal points is sorted in von Mises stress ascending order.

Step 3: The first point of the *stackIP* is identified as *minimum* and removed from the *stackIP* to a *temporalstackIP*.

Step 4: This *minimum* is taken as the initial central point to be enclosed by the new contour. Internal points surrounding this *minimum* and having a von Mises stress lower than a threshold are also moved from the *stackIP* to the *temporalstackIP*.

Step 5: A convex polygon is created from the internal points in the *temporalstackIP* as displayed in figure 6.2 (a). This polygon is split up into two open polygons, *control polygons*, and the internal points are taken as the control points. Therefore, these control polygons together with the control points define two NURBS curves creating a new hole (figure 6.2 (b)).

Step 6: If the remaining *stackIP* is not empty, the process is repeated from point 2 until there are no more internal points in the *stackIP*.

(a)   (b)
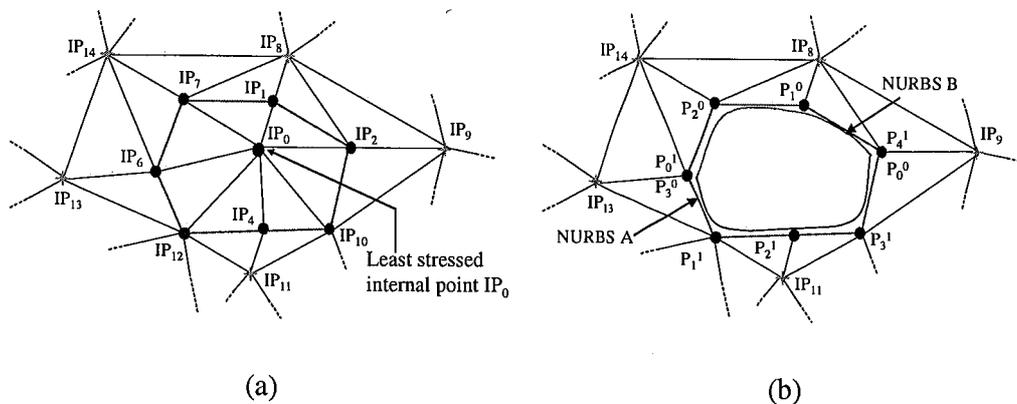
Figure 6.2: Creation of holes from internal points (a). New hole consists of two NURBS curves (b)

Note that if the von Mises stress in most of the internal points surrounding the *minimum* is higher than the threshold, then the *temporalstackIP* consists of only a few internal points and therefore, no holes are inserted. The minimum number of control points necessary to create a hole is chosen to be five. The reason for this is

that each hole makes use of two NURBS curves and at least three control points are required for a proper (and more flexible) definition of each NURBS curve.
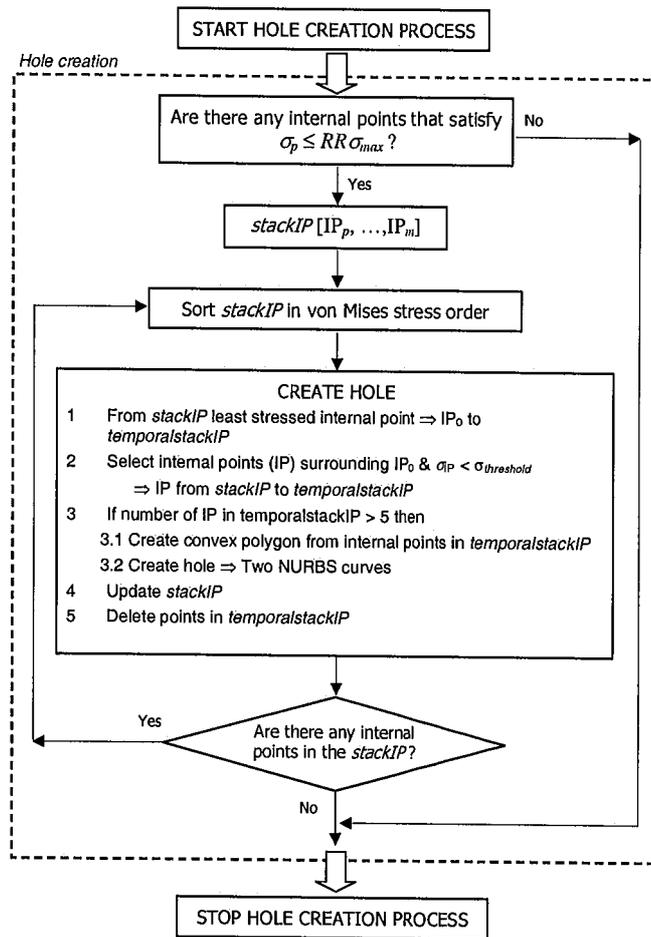
Figure 6.3: Flow chart of the hole creation process

Holes, which are described by two NURBS curves per hole, have the same form of boundary representation as the outside boundary. Consequently, they have similar properties and behaviour such that they can either grow or shrink. The fact that they are defined by two pieces of NURBS instead a single B-spline is due to some code limitations. Of course, the cavities could be split even into more pieces but for simplicity in the definition of the hole, two pieces were adopted here.

Often, as the external boundaries shift inwards, there are a few adjacent low stress points. Moreover, low stress points also appear near to existing cavities. The areas covered by these low stress points are generally very small and they usually

disappear after the next iterations. Hence, cavities should be introduced within the domain only as long as their locations are neither near existing cavities nor adjacent to the structure boundary, both situations depicted in figure 6.4.
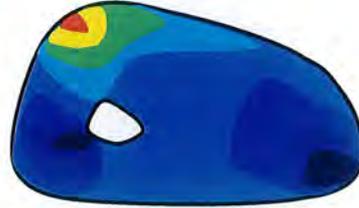
Figure 6.4: Low stress locations near an existing cavity and adjacent to the external boundary

6.6   Distance to Move Control Points Defining Holes

To calculate the distance to move the control points in internal holes, a similar procedure to the outer boundary is followed (equation 5.6). Unlike the outer boundary, the set of control points associated to each specific area to be moved consists of only one control point, the closest point to this area. The reason for moving exclusively the closest control point is that around cavities, high stress areas are alternated close to low stress regions. This situation can be illustrated using the example shown in figure 6.5 of a circular hole in a plate under uniaxial loading. Timoshenko and Goodier (1970) give the stress σ_θ to be

$$\sigma_\theta = \frac{S}{2}(1 + \frac{a^2}{r^2}) - \frac{S}{2}(1 + \frac{3a^4}{r^4})\cos 2\theta \tag{6.5}$$
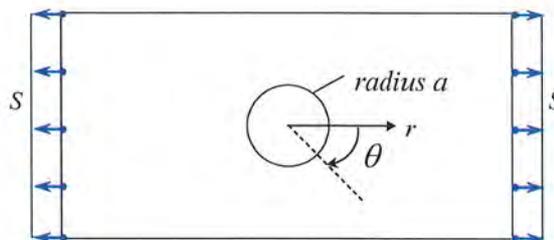
Figure 6.5: Circular hole in a plate under uniaxial loading

So on the boundary of the circle, where $r = a$,

$$\sigma_\theta = S(1 - 2\cos 2\theta) \tag{6.6}$$

On an unloaded circle σ_r (radial) $= \tau_{r\theta} = 0$ giving two zero valued principal stresses (in plane stress) and the third principal stress equal to σ_θ the von Mises stress varies as shown in figure 6.6 (b).

(a)   (b)
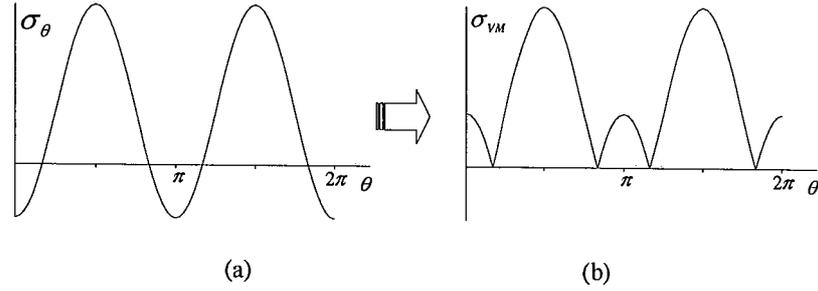
Figure 6.6: (a) Stress $\sigma_\theta (r = a)$, (b) von Mises stress $(r = a)$

Thus, moving the adjacent control points would cause a counter productive removal or addition of material. In addition, generally, the cavities are created after a few iterations and therefore, the factors RF and AF have decreased compared to the initial steps. A direct effect of this is that the process is slowed down and hence changes in the boundary of the hole are smooth. Therefore, the control point is moved a distance related to the following parameters

- Length of the least/most stressed element, L_e

- Distance, a, of the control point from the low/high stressed node

- The removal factor (RF) if removing material, or the addition factor (AF) if adding it

For example, in figure 6.7 (a) the closest control point from the node of lowest stress is moved a quantity calculated as follows

$$\text{Movement of } P_2 = L_e \frac{1}{a} RF \tag{6.7}$$

Generally, the optimisation parameters (*RR*, *RF*, *AF*, etc.) adopted in the creation and evolution of the cavities at each iteration have the same value as the parameters adopted for the outer boundary. However, by changing the parameters *RR* and *RF* in an appropriate way the number of cavities to insert can be controlled in a similar fashion as the *intelligent cavity creation algorithm* (ICC) (Kim *et al.* (2000)) implemented in ESO.

(a)   (b)
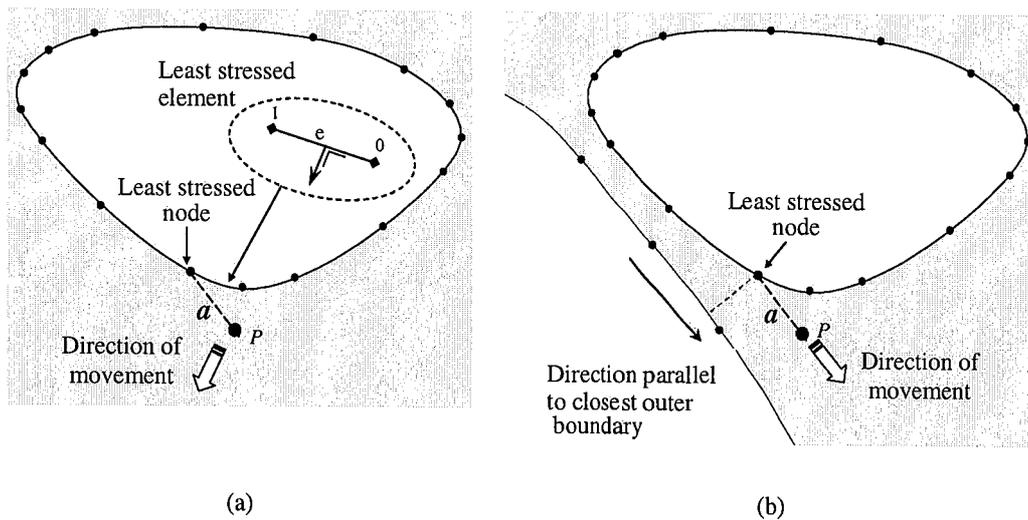
Figure 6.7: Direction of movement. Isolated hole (a), hole with a close boundary (b)

6.7 Direction of Movement in Holes

Generally, the direction of movement for a control point in a hole is perpendicular to its nearest least/most stressed element (figure 6.7 (a)) as has been explained in section 5.6.3. Nevertheless, in the case of a close outer boundary (figure 6.7 (b)) the direction of movement is determined by this borderline, i.e. parallel to the boundary that is closest to the least stressed node. In a summary, there are two directions of movement related to the boundaries of holes and which are determined according to the neighbouring situation. For a hole, if there is no boundary closer than certain distance (d_{min}) related to the element size then the material would move perpendicular to its boundary. On the other hand, it would move parallel to a boundary closer than this threshold.

6.8 Merging Holes

It appears that the algorithm works towards an optimum by producing multiple holes which later merge to form large cavities. For this purpose an algorithm to merge holes is implemented in order to complete this topology optimisation approach. Therefore, holes can merge between each other or with the outside boundary if necessary. Moreover, it is found that the presence of multiple holes would produce thin filaments of material which are not efficient from a manufacture point of view.

6.8.1 Merging Two Holes

This situation occurs when the cavities evolve approaching too close to each other, hence creating low stress filaments of material in between these cavities. The criterion for merging holes is based on the following two premises

- *Minimum distance.* If the distance between two holes is less than a minimum those holes are likely to be merged. The distances are calculated between control points from each hole. The largest element length, from the elements in the closest area, is taken as the minimum distance.

- *Under stressed area in between.* Internal points situated in the region between the holes satisfy equation (6.4).

Therefore, adjacent holes merge only if they satisfy the two previous conditions. As shown in figure 6.8, merging the holes implies the creation of a new one. Consequently, it follows a similar process to the *hole creation process* previously stated in section 6.5. In this case, *stackIP* consists of all the control points related to the holes to be merged.
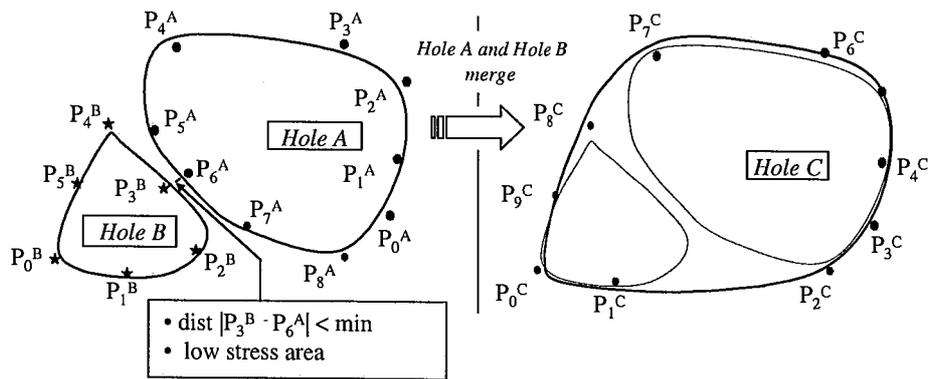
Figure 6.8: Merging two holes (A, B) into a new one (C)

6.8.2 Merging Hole to Outside Boundary

The criterion for merging a hole with the outside boundary is similar to the one for merging two holes, that is, *minimum distance* and *under stressed area in between*.

Figure 6.9 depicts how a hole is merged to its closest boundary. It implies a change of the boundary definition and also the deletion of the hole. The extreme (maximum and minimum x y coordinates) control points defining the hole are identified and added to the set of control points describing the boundary. In addition, control points from the boundary in between these extreme points are removed.
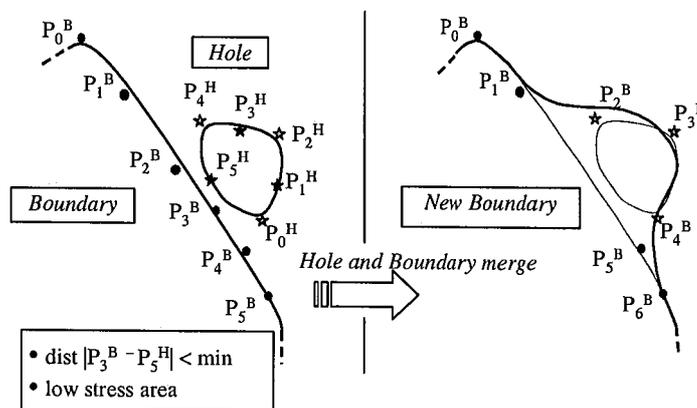
Figure 6.9: Merging hole (*Hole*) into boundary (*Boundary*)

The *merging holes process* is presented in flow chart form in figure 6.10.
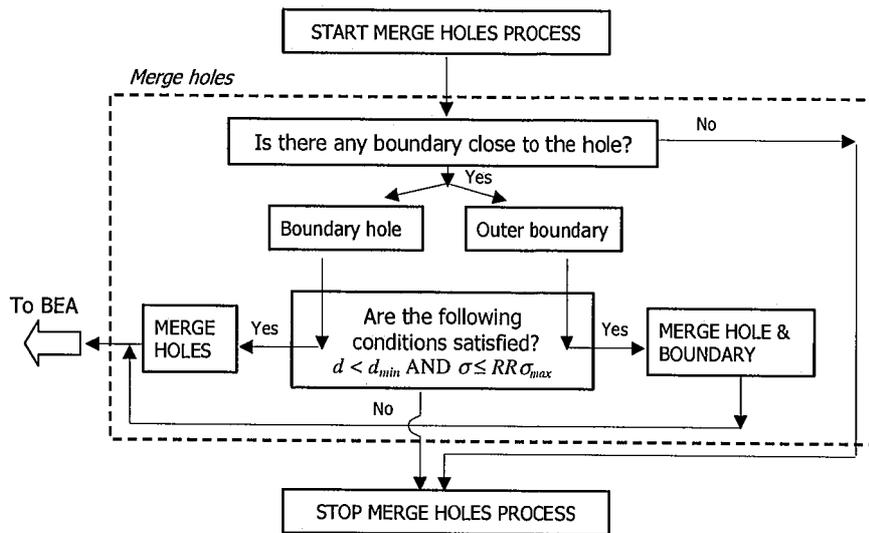
Figure 6.10: Flow chart of merge holes process

6.9 Stopping Criterion

The objective function adopted in these topology optimisation problems is related to the minimisation of the specific strain energy f_U in the structure. This criterion has already been used in the shape optimisation problem of the two-bar frame (section 5.9.3) following equation (5.18) and rewritten here as follows

$$f_U = UV \tag{6.8}$$

where V refers to the volume of the structure and U to the strain energy which in BEA notation is defined as

$$U = \int_\Gamma \frac{1}{2} Tu \, d\Gamma \tag{6.9}$$

where T are the tractions over the boundary and u the displacements over the part of the boundary Γ where the tractions are applied.

Thus, the stopping criterion can be quantified in the following normalised form

$$\varepsilon = \left| \frac{f_U^{i+1} - f_U^{i+1}}{f_U^{i+1}} \right| \leq 10^{-4} \qquad\qquad (6.10)$$

where f_U^i is the value of the objective function at iteration i and f_U^{i+1} is the value of the objective function one iteration after. Finally, the flow chart of the basic algorithm, which was presented in figure 6.1, is enhanced in the following figure 6.11.

Geometry Control

Are there any control points that satisfy

$|P_j - P_{j-1}| > k|P_j - P_{j-1}|_0$? $(k \geq 1.5)$

Yes → CONTROL POINT INSERTION
j' $(j-1 < j' < j)$

No

Are there any control points that satisfy

$|P_j - P_{j-1}| < l|P_{j-1} - P_{j-2}|$; $(l<1)$ (a)
OR
$|P_j - P_{j-1}| < l|P_{j+1} - P_j|$; $(l<1)$ (b) ?

Yes → CONTROL POINT REMOVAL
$j-1$ if (a) OR j if (b)

No

4

Hole creation

Are there any internal points that satisfy
$\sigma_p \leq RR\sigma_{max}$?

No

Yes

stackIP $[IP_{p_1} ..., IP_m]$

Sort *stackIP* in von Mises stress order

CREATE HOLE

1   From *stackIP* least stressed internal point $\Rightarrow IP_0$ to *temporalstackIP*
2   Select internal points (IP) surrounding IP_0 & $\sigma_{ip} < \sigma_{threshold}$
        $\Rightarrow$ IP from *stackIP* to *temporalstackIP*
3   If number of IP in temporalstackIP > 5 then
        3.1   Create convex polygon from internal points in *temporalstackIP*
        3.2   Create hole $\Rightarrow$ Two NURBS curves
4   Update *stackIP*
5   Delete points in *temporalstackIP*

Yes

Are there any internal
points in the *stackIP*?

No

148

Merge holes

Is there any boundary close to the hole? — No

Yes

Boundary hole   Outer boundary

To 2

MERGE HOLES — Yes — Are the following conditions satisfied? $d < d_{min}$ AND $\sigma \leq RR\sigma_{max}$ — Yes — MERGE HOLE & BOUNDARY

No

5

To 2 — No — IS STOPPING CRITERION SATISFIED?

Yes

STOP

Figure 6.11: Flow chart of the topology optimisation algorithm

6.10 Examples

In this section, classical examples are presented in order to prove the developed algorithm. The following isotropic material properties are assumed: Young's modulus $E = 210000\text{N/mm}^2$, Poisson's ratio $v = 0.3$ and an arbitrary thickness $t = 1$ mm. The Yield stress for this material is 280 N/mm^2. Plane stress conditions are also assumed. The weights associated to the NURBS curves (see chapter 4) are taken to be $w_i = 1$ in all cases.

6.10.1 Short Cantilever Beam

A common benchmark problem is the cantilever beam with a central load and an aspect ratio of 1.6. The definition of the initial problem is shown in figure 6.12. This cantilever of dimensions 160 mm x 100 mm is fixed at the top and bottom corners of

the left hand side; whereas it is subjected to a vertical load at the centre of the free end. The load has been chosen arbitrary of value 10 N. This value can be regarded to be very small compared to the failure stress; however it produces an appropriate stress distribution to study removal of material as well as the creation and evolution of holes. The optimisation parameters are $AF = 1.0$, constant $AR = 0.98$, $ER_R = 0.01$, RR, varies from 0.01 to 0.2, and RF changes from 0.8 to 0.5 according to the stress levels within the structure.
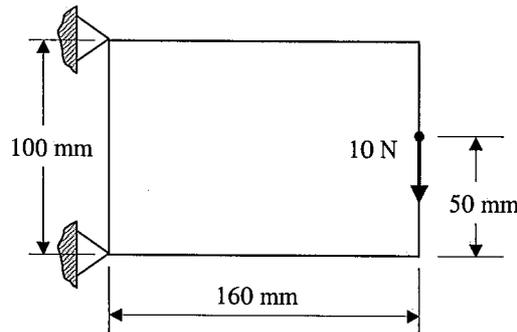
Figure 6.12: Problem definition for a short cantilever with a point load at the middle right hand

The boundary domain is initially discretised into 37 quadratic elements (fig.6.13 (a)). At the end of the process this number has increased to 165 elements (fig.6.13 (b)). The number and position of the control points for the initial and final topology design are depicted in the figure 6.14. The initial 5 non-constrained straight lines are converted into 5 quadratic NURBS curves. In the final design this number has increased, with the creation of internal holes, to 11 NURBS.

(a)   (b)
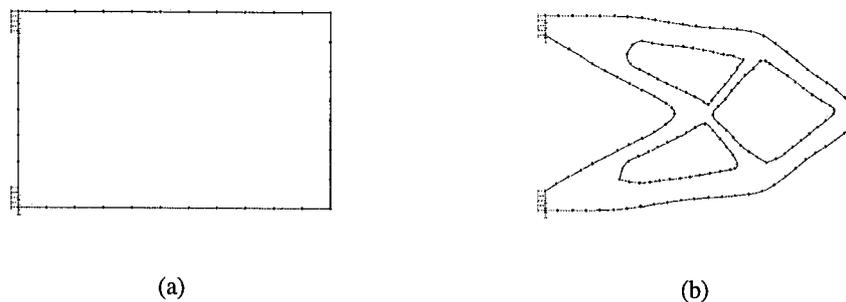
Figure 6.13: Initial and final boundary element mesh

150

(a)   (b)
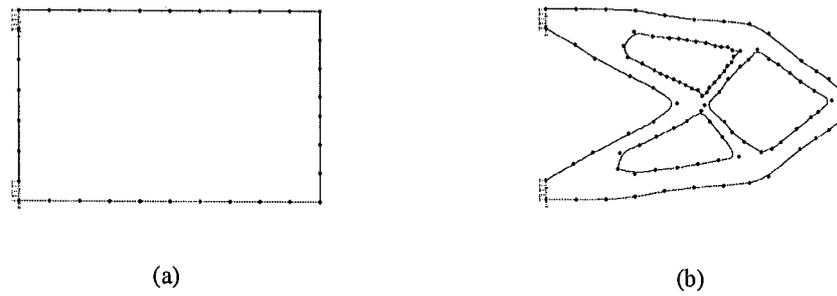
Figure 6.14: Initial and final control point distribution

Figure 6.15 compares the optimum design obtained with this approach (boundary-ESO) to the optimal layout theory by Rozvany (1995) and to a solution obtained by Chu *et al.* (1997) using FE-ESO. The analytical solution figure 6.15 (a) is a truss structure. Figure 6.15 (b) shows the ESO solution obtained under stiffness constraints. Sensitivity numbers indicate the change in the overall stiffness or a specified displacement due to the removal of an element. The final solution is obtained for an element mesh of 32x20 and a displacement limit of 1.0 mm. The optimum design obtained with the current approach (figure 6.15 (c)), shows a good agreement with the FE-ESO.

(a)   (b)   (c)
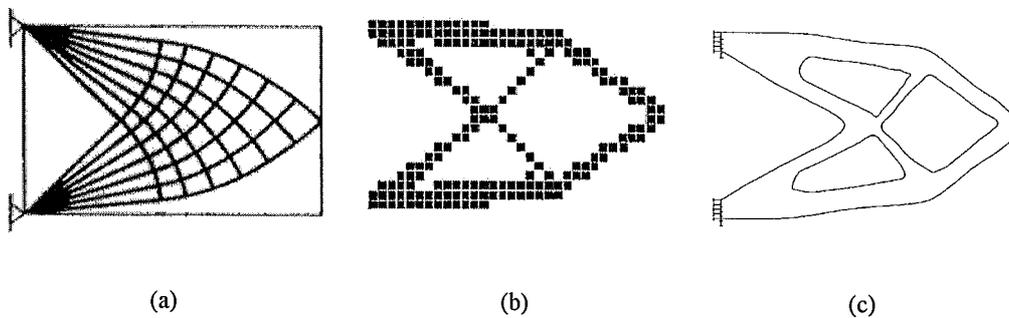
Figure 6.15: Optimum designs. (a) Analytical (Rozvany (1995)). (b) ESO (Chu *et al.* (1997)). (c) boundary-ESO

Figure 6.16 shows, in terms of von Mises stress contours, how the structure evolves towards the final solution. As the optimisation progresses, small cavities are merged into larger cavities. Also cavities that are very close together or to the outer boundary may merge. Thus, this algorithm for merging adjacent holes avoids thin filaments of material. The optimum topology design is accomplished after 38 iterations. The

151

volume reduction is approximately 63% ($V/V_0 \approx 0.37$) from the initial design. Although the overall process shows a clear reduction of volume (removal of material) in the structure, it should be noticed that material is also added if any stress concentration appears as a result of the movement of the control points. The final design is obtained in a CPU time of 5 minutes and 27 seconds (Pentium 4 (2 GHz)).

iteration 0

iteration 10

iteration 17

iteration 21

iteration 27
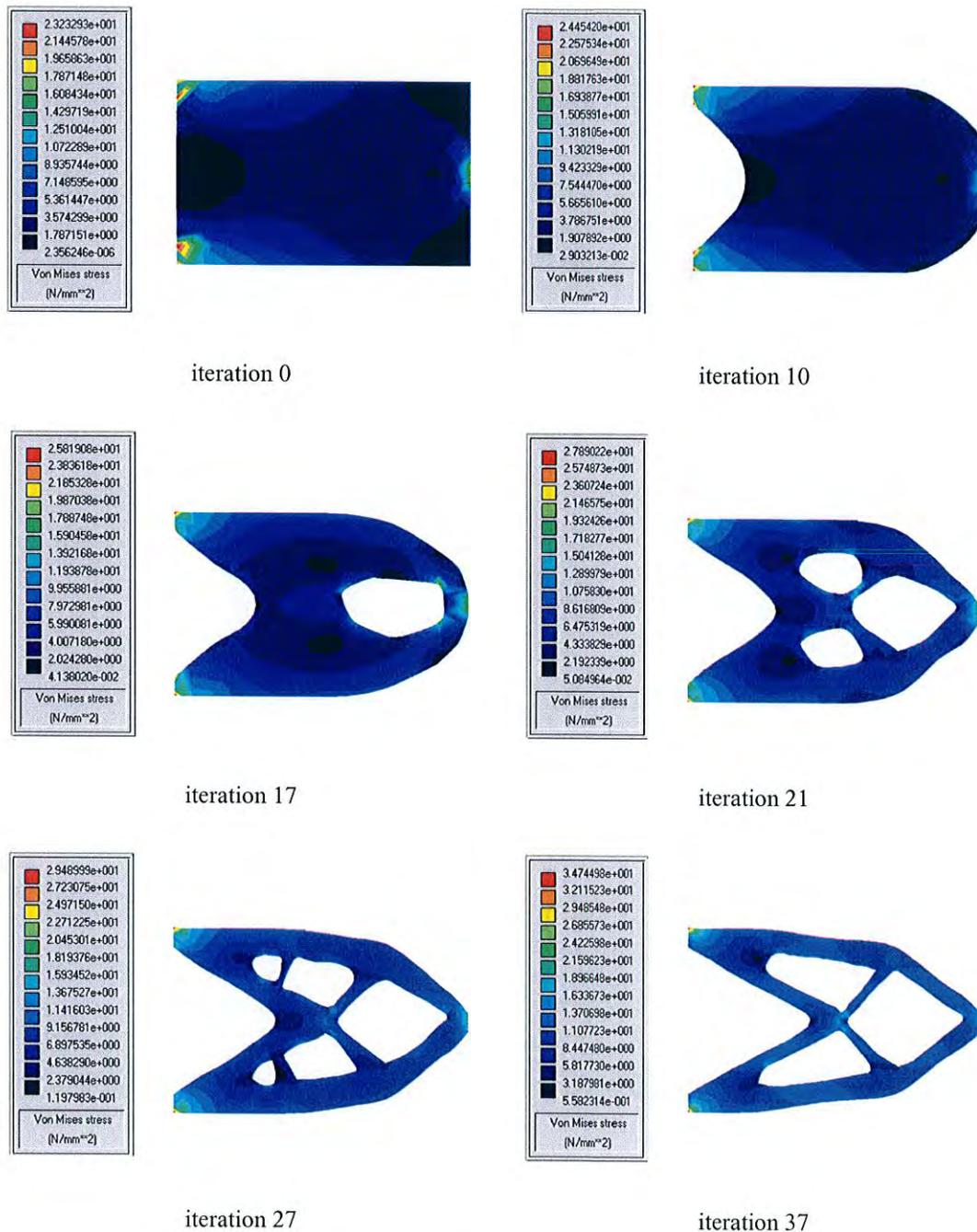
iteration 37

Figure 6.16: Von Mises stress contour plot

152
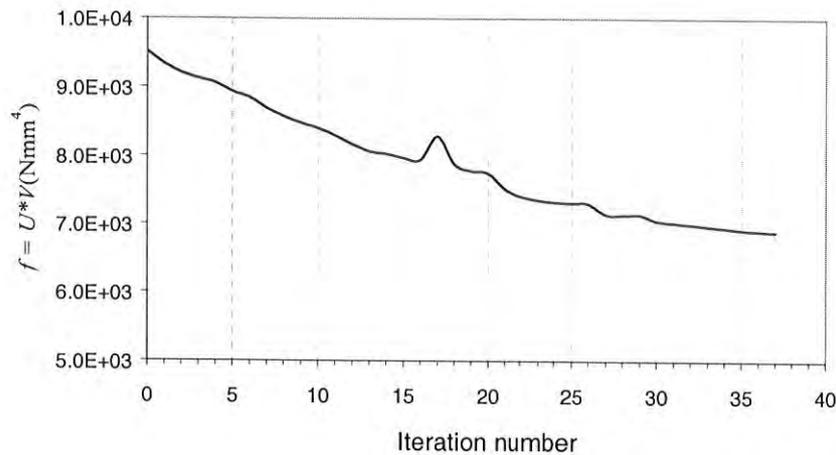
Figure 6.17: Evolution of the objective function

The evolution of the objective function (equation (6.8)) is displayed in figure 6.17. As it can be observed, the function decreases smoothly until iteration 16. At this iteration there is a big jump in the function due to the insertion of the first hole. A few iterations later, at iteration 21, two new holes are created. The effect of the creation of these latter holes is not as dramatic as the first one because the size of the holes is much smaller. As the process evolves another two holes are created at iteration 27. These last cavities are very small compared to the existing ones and eventually are merged to the larger holes. The final topology is achieved when no further improvements are shown in the objective function which shows a 28% reduction compared to the initial situation.

6.10.2 Short Cantilever-II

The previous short cantilever is investigated but loaded at the top right hand corner with a point load of 100 N as shown in figure 6.18. The initial dimensions are the same as the ones defined in example 6.10.1. The optimisation parameters are set to $AF = 1.0$ and $AR = 0.98$. Parameters related to the removal process are $ER = 0.01$, RR, initially set to 0.01 rises up to 0.18 and RF varies from 0.3 to 0.1.

The original boundary domain is discretised into 45 quadratic elements (fig 6.19 (a)). At the end of the process, the model size has increased to 103 elements (fig. 6.19

(b)). Initially, 3 quadratic B-splines define the geometry (fig 6.20 (a)), increasing to 5 NURBS curves (fig 6.20 (b)) in the final design.
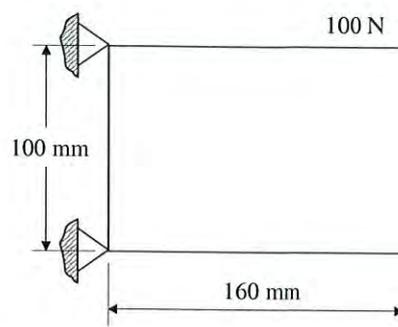
Figure 6.18: Problem definition for a short cantilever beam with a point load on the top right hand corner

(a)   (b)
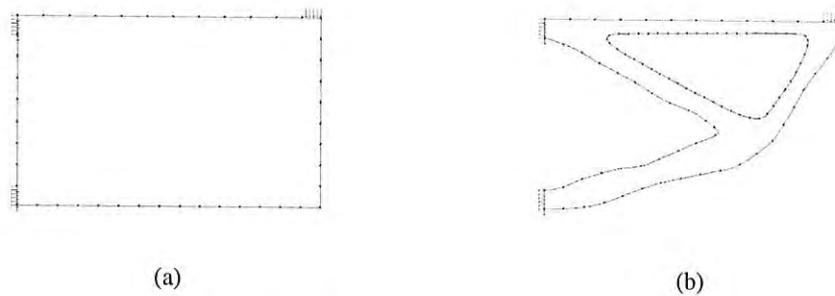
Figure 6.19: Initial and final element mesh

(a)   (b)
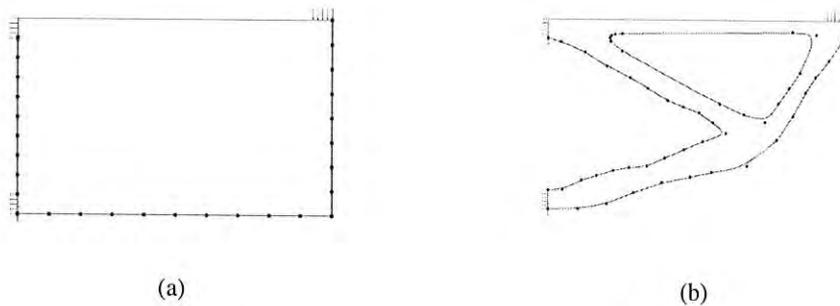
Figure 6.20: Initial and final control point distribution

Figure 6.21 compares the optimum design (c) to the optimal design obtained by Eschenauer *et al.* (1994) using the *bubble method* (a) and by Hinton and Sienz (1995) using the *hard-kill method* (HK) (b). The numerical solution in figure 6.21 (a) is obtained by iterative positioning of new holes (so-called bubbles) in the structure.

The iterative positioning of holes is carried out using variational methods. The objective followed in this example was the minimisation of the mean compliance while considering a volume constraint and certain equality conditions. This solution is obtained for a load applied at bottom hand corner therefore we compare the final topology to our results qualitatively, considering this limitation. Figure 6.21 (b) shows the solution obtained using an automated fully stressed design approach based on ESO. The fully stressed design is obtained following a von Mises stress criterion. The final continuum topology can be interpreted as 4 bar trusses.

(a) load at bottom right hand   (b) load at top right hand corner   (c) load at top right hand corner
corner
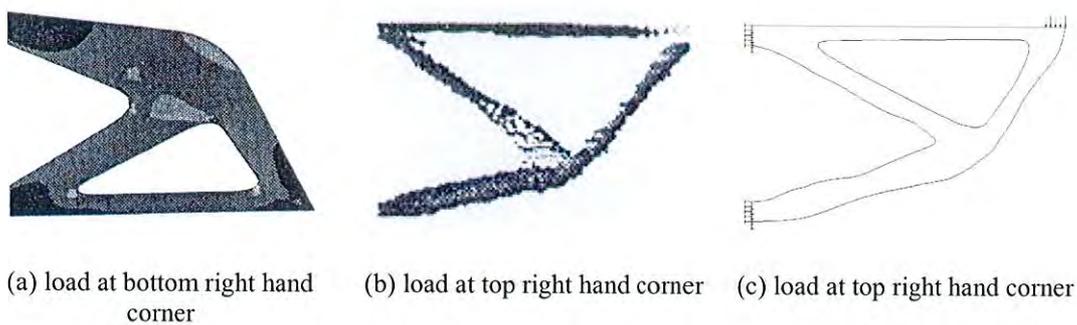
Figure 6.21: Optimum designs. (a) Bubble method, Eschenauer *et al.* (1994). (b) HK, Hinton and Sienz (1995). (c) boundary-ESO

Figure 6.22 displays the von Mises stress contour plots for different iterations. The solution is accomplished after 47 iterations, allowing the insertion and merging of internal holes. These holes are merged not only between each other but also with their closest boundary. As a result, the final design has only one hole covering most of the internal domain. The initial volume has been reduced 70% ($V/V_0 = 0.30$).

iteration 0   iteration 10

iteration 20   iteration 30

iteration 35   iteration 37

iteration 40   iteration 46
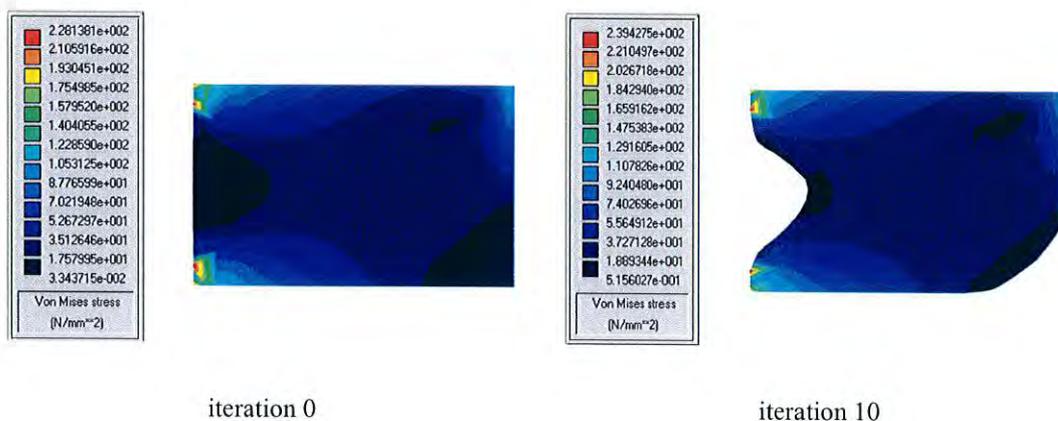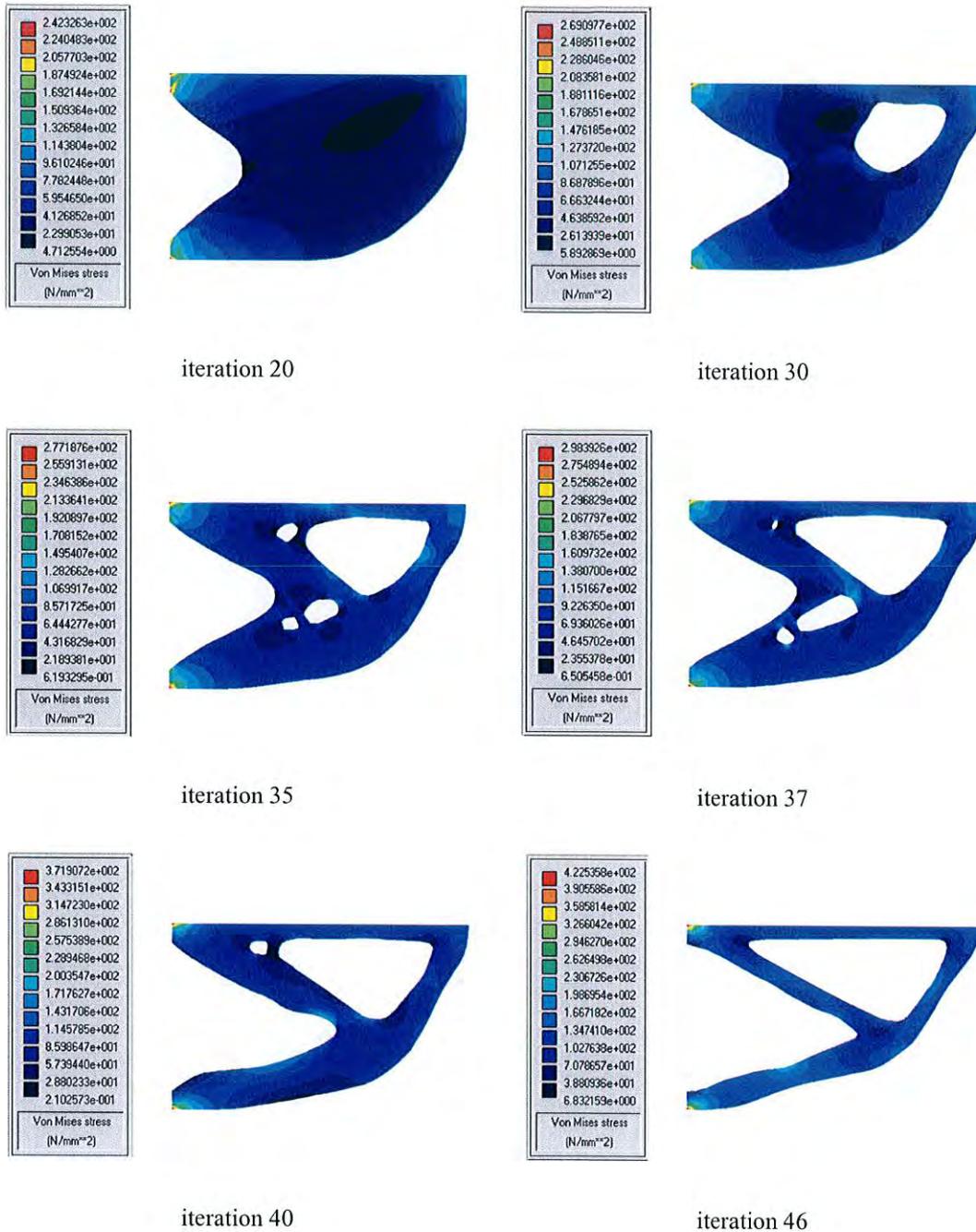
Figure 6.22: Von Mises stress contour plot

The creation and merging of cavities in the process is evident in the evolution of the objective function displayed in figure 6.23. The function decreases $\approx 25\%$ compared to the initial situation. Local minima are shown at any iteration previous to the iteration where holes are created and/or merged. Indeed, a change in the topology of the structure exhibits an increase in the objective function. For example, at iteration

22 a hole is created and at iteration 40 several existing holes are merged. The final topology obtained after 1 minute and 32 seconds on a Pentium 4 (2 GHz) is achieved when no further improvements are shown in the objective function.
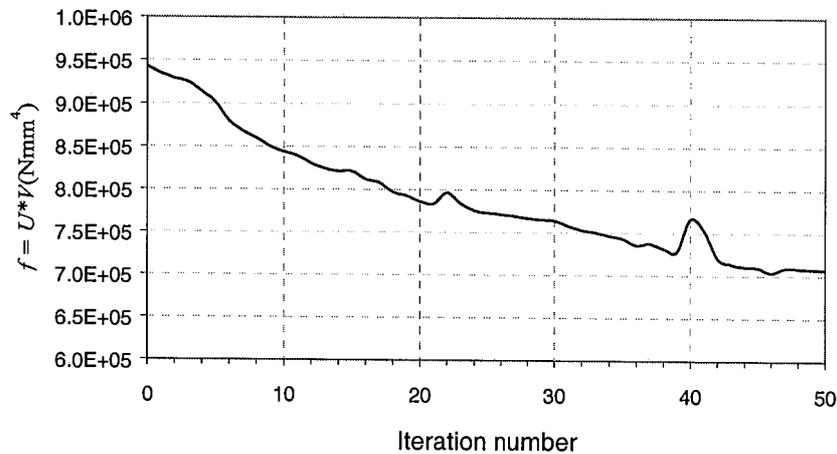
Figure 6.23: Evolution of the objective function

Figure 6.24 shows the effect of the mesh size on the final solutions. Three different types of mesh are tested, which according to the software used can be labelled as *fine*, *standard* and *coarse* mesh (see section 5.5). The topology results show a dependency on the mesh size. In the Concept Analyst software, the number of internal points happens to be related to the mesh size. Thus, the finer the mesh the more internal points are used. On the contrary, the coarser the mesh the fewer internal points are dispersed around the domain. Generally, the outer boundary behaves in a similar manner regardless of the element size. However, the final solutions present different topologies since the cavities (their size and number) vary depending on the mesh size. It is observed that a coarse mesh would give rise to the creation of bigger holes, because of the larger internal point spacing, which in turn may destabilise the structure. This problem can be overcome by controlling the number of cavities that are allowed in the structure. As the number of elements in the mesh is increased, there is a general convergence to the same final topology. This is observed in the solutions obtained using a *fine mesh* (65 elements in the initial design) and a *standard mesh* (45 elements initial design). Also, as the mesh increases numerical tests have shown that the volume (V) decreases and the strain energy (U)

increases. Nevertheless, the results from a standard mesh (in the example above) and from a fine mesh are very similar. Since the reliability of the analysis results decreases with increasing element size, coarse meshes may not be appropriate. Also, since the constrained lines are non changeable lines, the final solution is clearly dependent on the initial definition of these lines. Any other load or constraint situation would evolve into a different final design.

COARSE MESH
Initial mesh: 31 elements
Final mesh: 80 elements
$U/U_0 = 1.46$
$V/V_0 = 0.54$

FINE MESH
Initial mesh: 65 elements
Final mesh: 208 elements
$U/U_0 = 2.85$
$V/V_0 = 0.276$
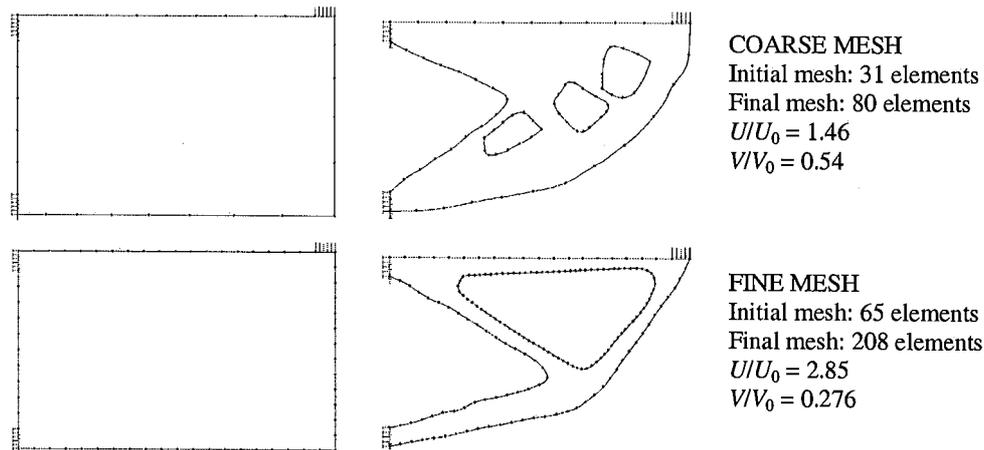
Figure 6.24: Effect of the mesh size on the final solution

6.10.3 Bridge

This example aims to reproduce a Michell type structure (Michell (1904)). This problem has been solved analytically by Hemp (1973) and it has also been investigated by Li et al. (1999) using FE-ESO. The initial domain displayed in figure 6.25 is a rectangle with the horizontal dimension (200 mm) twice the height (100 mm). The two bottom corners are assumed to be fixed; also a point mid-span load of 100 N is applied at the bottom. The parameters are set to $ER_R = 0.01$, RR, initially set to 0.01 rises to 0.12. RF varies from 0.3 to 0.06, $AF = 0.5$, $AR = 0.75$ and $ER_A = 0$.

Figure 6.26 shows the boundary element mesh for both the initial (a) and final stages (b). The boundary domain is initially discretised into 34 quadratic elements. At the end of the process this number has increased to 114 elements. Figure 6.27 depicts the number and position of the control points for the initial (a) and final (b) topology design. Initially, the changeable lines are modelled by 5 quadratic NURBS curves.

158

Since 3 internal holes are inserted in the final design, the number of curves increases to 11.
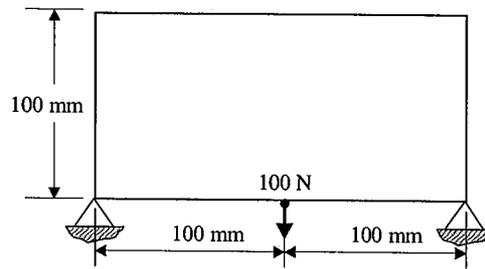
Figure 6.25: Problem definition for the Michell type structure

(a)   (b)
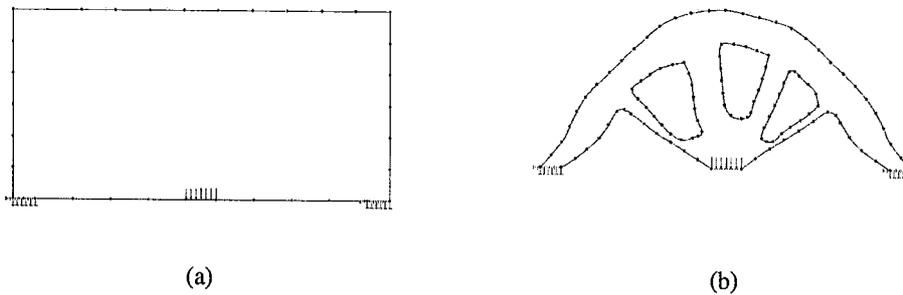
Figure 6.26: Initial and final boundary element mesh

(a)   (b)
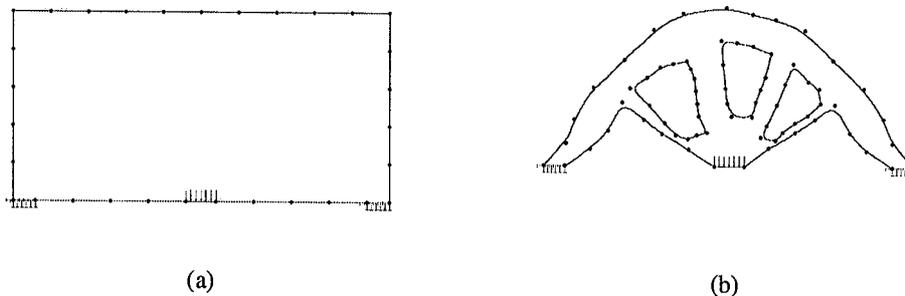
Figure 6.27: Initial and final control points distribution

Figure 6.28 shows the analytical results and the numerical ESO solution using the von Mises stress criterion. According to the seminal work by Michell (1904) and later corroborated by Hemp (1973), considering a truss structure with one load condition and stress constraints, the optimum design is a fully stressed design. The Michell trusses are also structures of maximum stiffness for a given volume or, in other words, structures of minimum strain energy. As stated by Li et al. (1999), the

original Michell model is a pin jointed frame. However in the 2D elasticity analysis a continuum domain is assumed and therefore the results show rigid jointed structures. This is the reason why the numerical results differ from the analytical ones. The solution obtained using this approach has a comparison to classical ESO.

—— Tension bars
--- Compression bars

(a) Michell structure in Rozvany (1995)   (b) ESO Li *et al.* (1999)
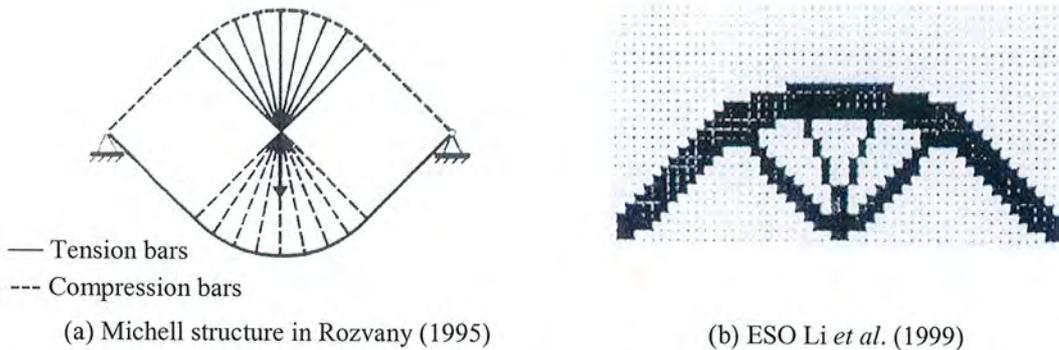
Figure 6.28: Exact optimum layout (a). FE-ESO solution (b)

Figure 6.29 displays the von Mises stress contour plot each 10 iterations. The final design is reached after 29 iterations. In terms of volume/weight reduction, the final design has a volume reduction of 65% from the initial design. Notice that in spite of the totally random position of the internal points, the proposed design is almost symmetric.

iteration 0   iteration 9

iteration 19   iteration 28
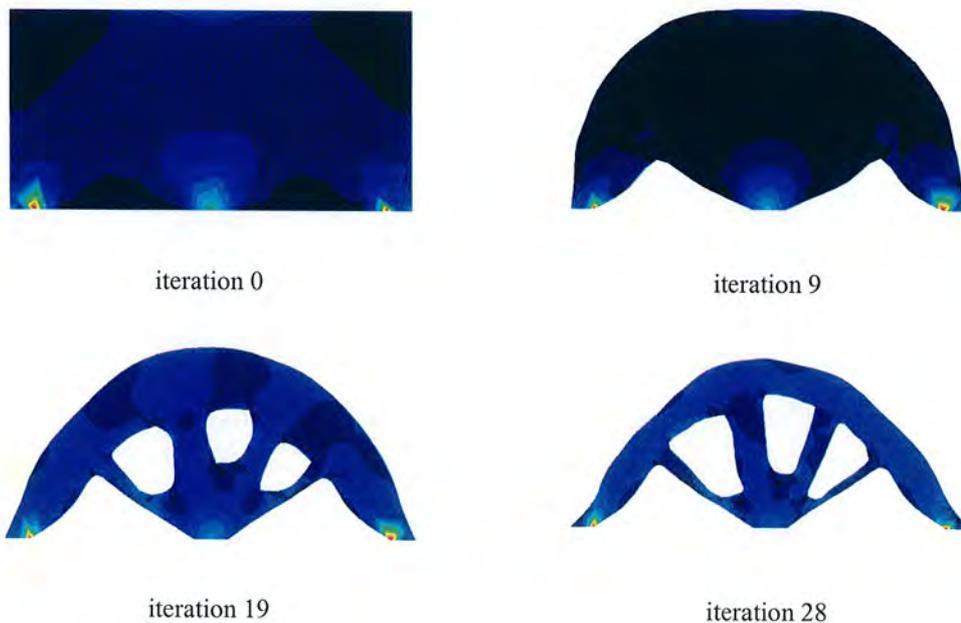
Figure 6.29: Von Mises stress contour every 10 iterations

The evolution of the objective function is displayed in figure 6.30. Step-like areas appear as a result of the creation of holes. The insertion of holes tends to increase the strain energy which can be identified with the presence of local minima in the graph. The final shape is achieved when no further improvements are shown in the objective function.
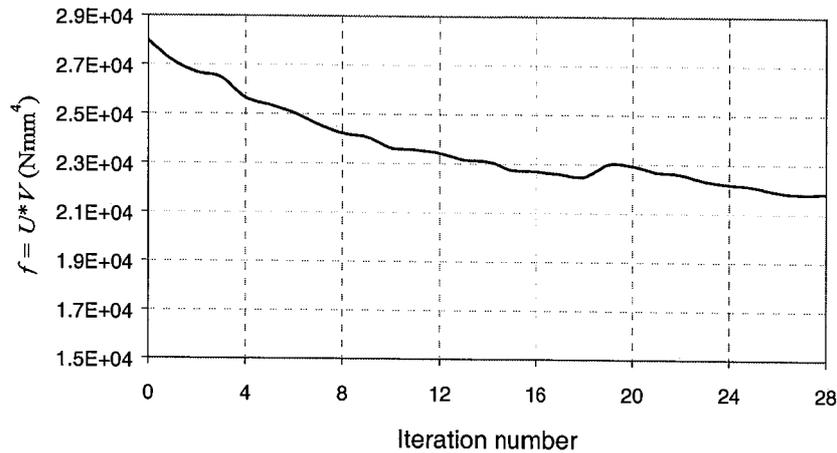
Figure 6.30: Evolution of the objective function

This example is studied further to explore the effect of the control points on the final solution. From the following figures (fig 6.31), it can be seen that using only a few control points to define the geometry or, conversely, many control points produces in general the same resemblance of an arch structure. Increasing the number of control points does not improve the convergence towards an optimum; on the contrary, it tends to create non smooth and oscillatory areas on the shape. This slows down convergence because the oscillatory boundary will give lots of stress concentrations inhibiting material removal. Smoothing algorithms and control point interpolation algorithms may alleviate this problem but increasing the complexity of the algorithm. It has been found that starting with equally distributed sets of control points and implementing the control point removal and insertion processes (see section 5.7) produces good results compared to benchmark problems. A general rule of thumb is that a control point spacing of 1/10 to 1/15 of the length of the longest changeable line gives the best stability to the process.

COARSE DISTRIBUTION OF CONTROL POINTS
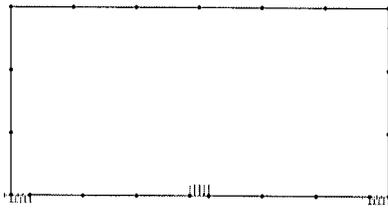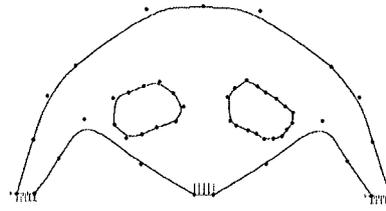
Initial control point distribution

Final control point distribution
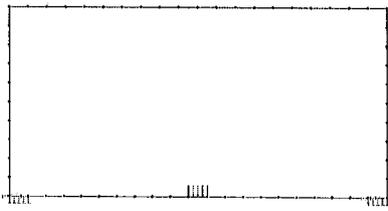
Initial boundary element mesh
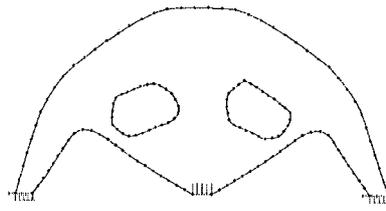
Final boundary element mesh

FINE DISTRIBUTION OF CONTROL POINTS
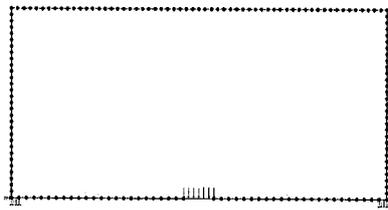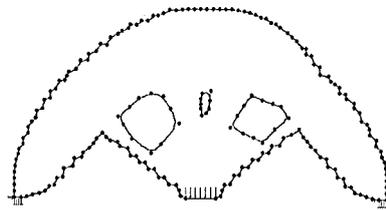
Initial control point distribution

Final control point distribution
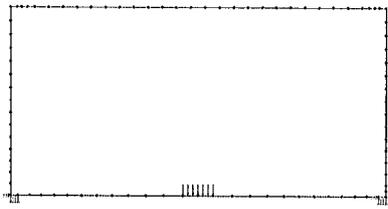
Initial boundary element mesh
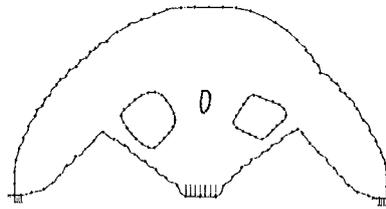
Final boundary element mesh

Figure 6.31: Effect of the control point distribution on the final solution

6.10.4 Beam under Multiple Load Cases

The previous examples have been applied to problems of one load case. However, in most engineering situations multiple load problems are frequent. For these cases the optimum design is not always a fully stressed design. Moreover, the solutions depend on the scheme adopted to compute the effect of each load case. The criterion explored in this work is the *logical AND/OR* algorithm (Li *et al.* (1999)) which is a simple scheme also used in FE-ESO. In the logical AND/OR algorithm a node is selected for the removal process only if the equation (6.1) is satisfied for *all* load cases and, it is selected for the addition process if the equation (6.2) is satisfied for *any* of the load cases.

The equations (6.1) and (6.2) presented previously refer to a single load case problem. In the case of multiple load cases these equations can be slightly changed as appropriate. Thus, for each node p and each load case j, a ratio between stresses is calculated as follows

$$R_p^j = \frac{\sigma_p^j}{\sigma_{max}^j} \tag{6.11}$$

Computing the above equation for all the load cases would give a set of ratios associated to each node. For example the node p has associated N ratios, where N is the number of load cases.

$$R_p = \left\{ R_p^0, R_p^1, R_p^2, ..., R_p^N \right\} \tag{6.12}$$

Dealing with these values, material can be removed from the structure if *all* the ratios $\{R_p\}$ are low stressed, that is; for any node p the following equation is satisfied

$$(R_p^1 \leq RR) \text{ AND} (R_p^2 \leq RR) ... \text{AND} (R_p^N \leq RR) \tag{6.13}$$

In the same way, material can be added if for *any* node p

$$(R_p^1 \geq AR) \text{ OR } (R_p^2 \geq AR) ... \text{OR } (R_p^N \geq AR) \tag{6.14}$$

where the RR and AR are the same optimisation parameters as the ones related to equations (6.1) and (6.2).

To illustrate these concepts, the example of the vertical beam (2LxL) studied in the shape optimisation chapter (in section 5.9.3) is optimised under two load cases which are identified as load case 0 and load case 1 and depicted in figure 6.32. In both load cases the load applied has the same magnitude of 100 N. Load case 0 has a vertical load applied at the middle of the right hand side edge whereas in load case 1 the applied load is at the top right hand side corner.
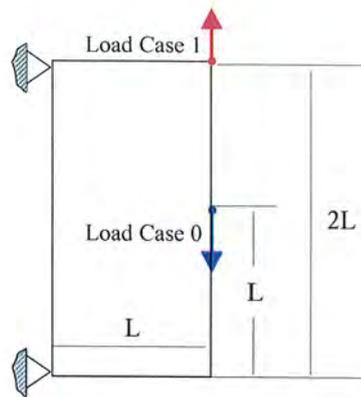
Figure 6.32: Beam under two load cases

The final solution for the two load cases combined using the logical AND/OR scheme is displayed in figure 6.33. This figure shows the von Mises stress contour plots for each load case at the initial and final iteration which was obtained after 38 iterations. The volume ratio at the end of the process is $V/V_0 = 0.43$ and RR varying from 0.01 to 0.29. Comparing this result to the solution for one load case (figure 5.34) it can be seen that the fully stressed design is no longer achieved.

initial design   final design

Load case 0 (a)

initial design   final design
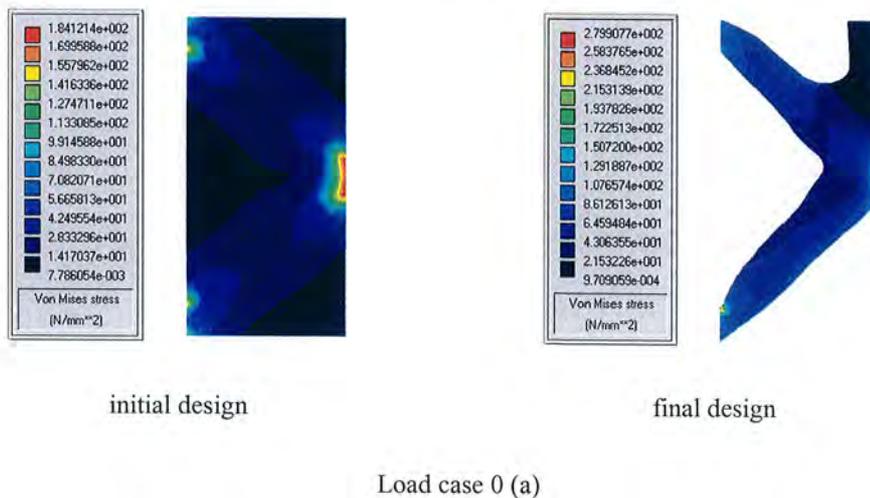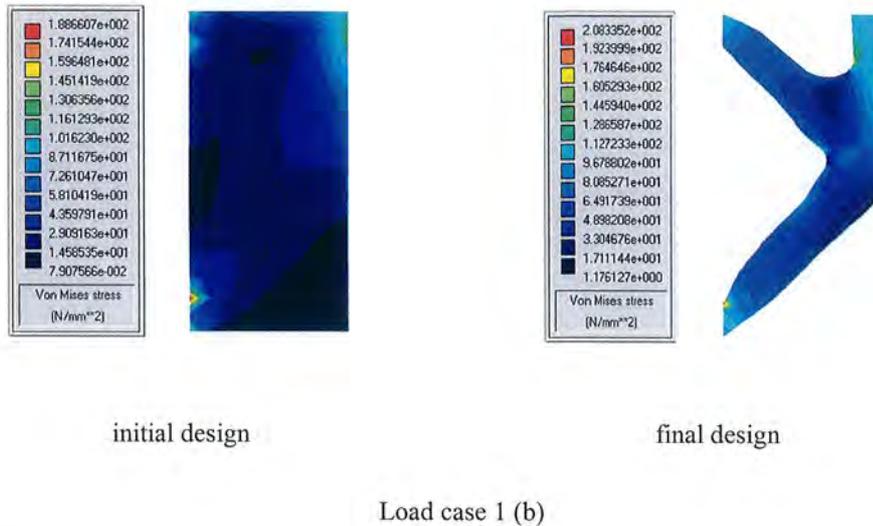
Load case 1 (b)

Figure 6.33: von Mises stress contour plots for load case 0 (a) and load case 1 (b)

6.10.5 Bridge under a Moving Load

This final example tackles the problem of a bridge structure under multiple loading conditions. The initial design for the bridge is shown in figure 6.34 (a). The dimensions of the bridge are 160 mm x 50 mm. The bridge is supported at four points in the bottom surface. A point load of 100 N is moving along the top surface and this effect is approximated by five load cases. The logical AND/OR schemes represented by equations (6.13) and (6.14) are also adopted. The top surface is considered non design domain as well as the four supports. Moreover, there is a limit for the maximum vertical displacement of 0.02 mm. The rest of the domain is free to change and defined by 5 NURBS curves. The optimum design, shown in figure 6.34 (b), that satisfies this problem is found after 42 iterations with RR varying from 0.01 to 0.27.

Figure 6.35 shows the von Mises stress contours, at the initial and final iteration, for each of the five load cases considered $\{LC_0, LC_1, LC_2, LC_3, LC_4\}$. The ratio of the final volume to the initial volume is 0.54. The maximum vertical displacement (u_{max}) for any load case is below the limit constraint of 0.02mm. Certainly, these results would have varied if the constraints over the maximum vertical displacement for

each load case were had been considered differently however, no further study has been considered related to this issue.
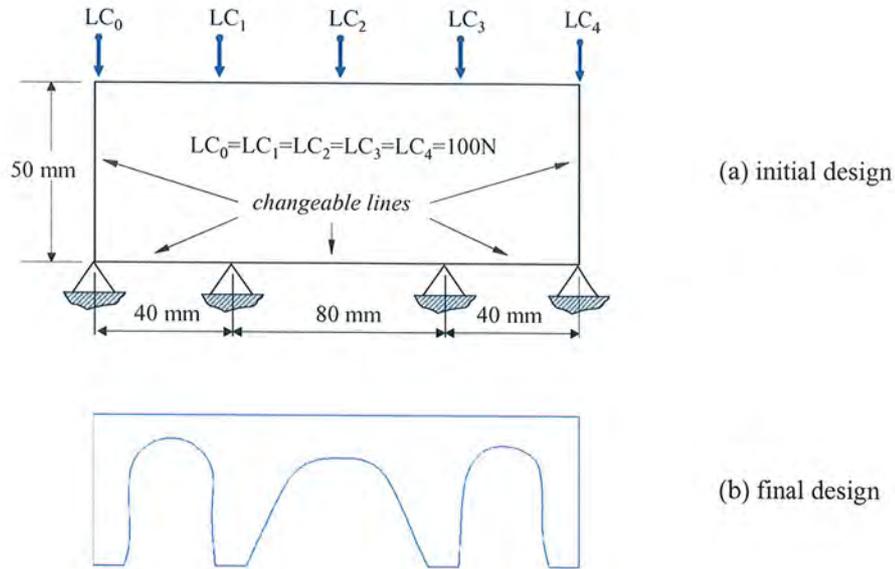
Figure 6.34: Initial and final design for a bridge under moving load
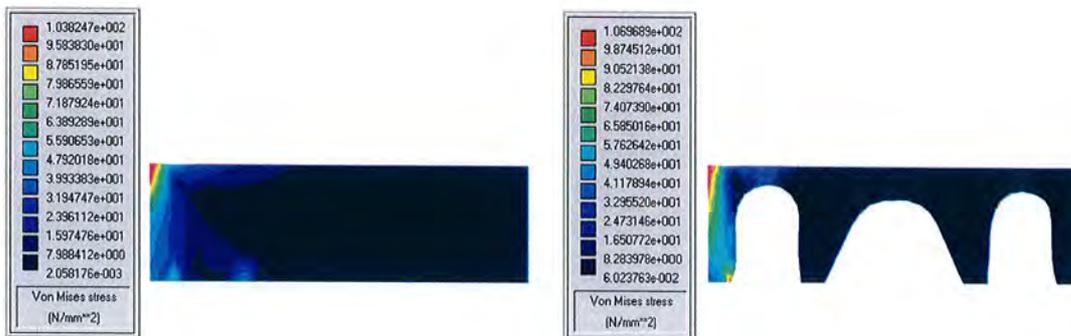
initial design

final design, $u_{max} = -0.01823$ mm

Load case 0 (a)

166
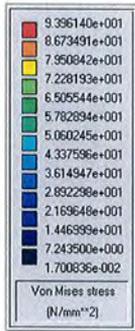
initial design   final design, u_{max} = - 0.01195 mm

Load case 1 (b)
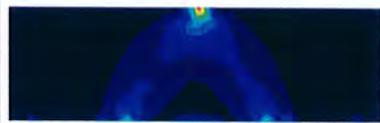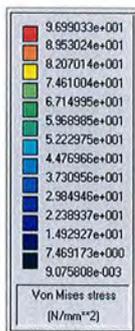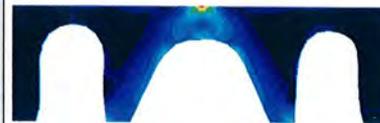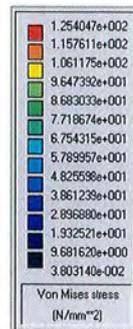
initial design   final design, u_{max} = - 0.01688 mm

Load case 2 (c)
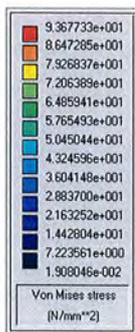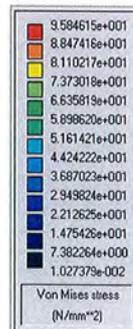
initial design   final design, u_{max} = - 0.01167 mm

Load case 3 (d)

initial design

final design, $u_{max} = -0.01807$ mm
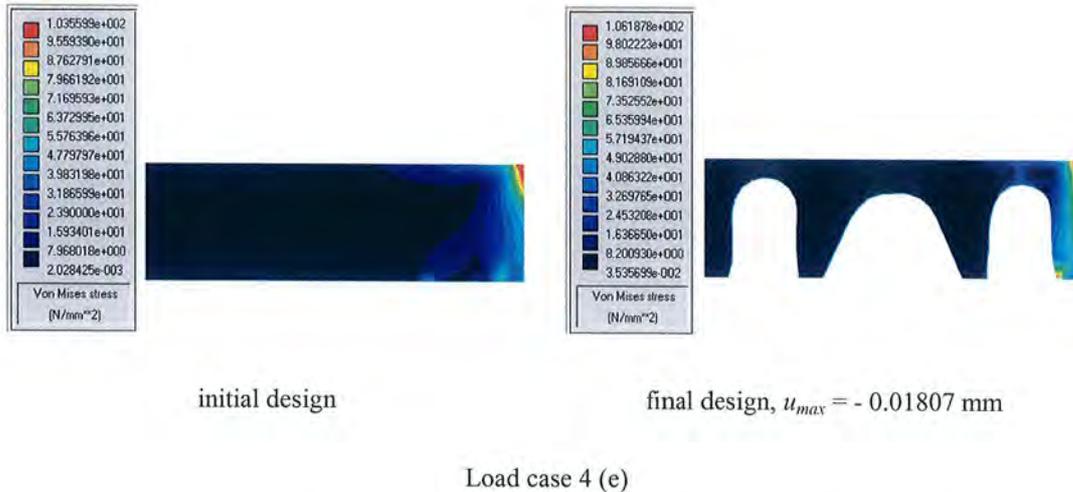
Load case 4 (e)

Figure 6.35: Von Mises stress contour plots for each load case

6.11 Discussion and Conclusions

This chapter has presented a new topology optimisation algorithm for evolutionary structural optimisation (ESO). The algorithm exploits the concept that the optimum topology evolves by slow removal and addition of material. As has already been concluded where this algorithm was applied to shape optimisation (chapter 5), the boundary element method is used for the analysis and NURBS curves for describing the changeable lines. The coordinates of the control points defining the NURBS curves are the design variables; in consequence the number of design variables compared to FE-based ESO has been decreased. Also, it has been proved that working directly with the boundary provides a large flexibility in the design. It has been shown how at each iteration the boundary remains smooth without artificial stress concentrations.

Topology optimisation is accomplished allowing the insertion of internal holes in the inner low stressed areas of the structure. Thus, inefficient material from inside regions is removed. Holes are also described by NURBS curves and so they have similar behaviour to the outside boundary. Therefore, both outside and inside are optimised at the same time.

To prove the algorithm, benchmark examples have been reproduced showing the effectiveness of the method generating the final topologies as well as rapid solution times. In the case of problems with an initial symmetry, almost symmetric shapes are obtained, in spite of the totally random position of the internal points and no symmetry being forced throughout the process. Multiple load cases are investigated using the logical AND/OR scheme. As expected, the final solution under multiple load cases is not necessarily a fully stressed design.

Numerical tests have shown that the solution is mesh dependent. This dependency can be overcome by controlling the number of cavities that are allowed to be created. The initial distribution of control points also determines the final topology. More control points do not increase the fidelity of the solution. On the contrary it induces the creation of spikes and oscillations on the shape that slow the convergence and interfere with the ability of this algorithm to reach a good optimum. A rule of thumb has been presented for effective control point spacing for general problems.

7

MULTI-CRITERIA OPTIMISATION

7.1 Overview

In previous chapters, shape (chapter 5) and topology (chapter 6) optimisation have been presented based on a single objective. Such individual optimisation criteria have included strain energy, volume and stress minimisation. In this chapter a transition is made from the optimisation of the single criterion to the optimisation of multiple criteria.

In multi-criteria problems optimum designs are obtained considering more than one objective function at the same time. Generally, these are the kind of problems presented in engineering and therefore, this study investigates a more robust and practical design approach in optimisation. Thus, the dilemma of obtaining an optimal solution that best satisfies a number of objectives is examined. In particular the use of Pareto frontier techniques which provide the trade-off among the desirable goals. Multi-criteria solutions are obtained for a well-known connecting rod example.

7.2 Introduction

Multi-criteria optimisation techniques (Eschenauer (2000), Osyczka (2002)) allow the designer to model a specific problem from a more realistic approach since it

considers the satisfaction of several objectives simultaneously. It presents the dilemma of obtaining an optimal design that best satisfies a number of objectives. The multi-criteria problem can be defined with the following function $f(x)$

$$f(x) = [f_1(x), f_2(x), \ldots \ldots f_N(x)] \tag{7.1}$$

Where $f_i(x)$ are the objective functions for the N different criteria and x is the vector of design variables.

One approach to solving multi-criteria problems is to generate and examine the *Pareto solution*. This concept represents a design alternative where an objective cannot be improved without worsening at least one of the other objectives. A *Pareto frontier* contains all Pareto solutions while a *Pareto set* is a discrete approximation of the Pareto frontier.

The concept of *Pareto optimality* is introduced in this work to solve the multi-criteria problem. It presents a range of optimum solutions to the designer. No solution stands alone as being the optimum but a trade-off solution. To find a compromise between the different objectives, an overall approach is taken in terms of a weighted average scheme. In general, an efficient formulation must be able to capture the entire Pareto points including non convexities and ensure that they are uniformly distributed.

7.2.1 Strategies to Tackle Multi-criteria. The Pareto Front

The solution of a multi-objective problem is difficult to obtain because the objective functions conflict with each other. The Pareto concept is used to gain knowledge about trade-offs between objectives, and from which the most desirable solution may be chosen. The Pareto point set is used to construct a point-wise approximation to the Pareto curve or surface (Pareto frontier). Then, from amongst those Pareto points the designer has to choose the design that achieves the most adequate objective function. There are several techniques to obtain the Pareto set such as the *weighted sum method*, the *normal boundary intersection method* or the *physical programming method*. In this work the *weighted sum* (WS) strategy is investigated.

7.2.1.1 The Weighted Sum Method

This technique defines a vector w of weights, which represent the relative importance of each criterion to the final solution, and transforms the multiple objective problem into a new single objective one as follows

$$F = \sum_{i=0}^{N} w_i f_i \qquad\qquad (7.2)$$

where generally $\sum_{i=0}^{N} w_i = 1$ as a way of normalising the objective function.

Solving the multi-criteria problem for different sets of weights yields the Pareto front or, in other words, each point in the Pareto front correspond to a particular set of weights.

The WS is a popular technique but nevertheless it presents the drawback that an even spread of weights does not necessarily produce an even spread of points on the Pareto curve/surface. The deficiency of the WS approach has its roots in the linear dependence of the objective function, more specifically by its inability to capture solutions that lie on concave boundaries of the feasible design space (Das and Dennis (1997)). With the difficulty of finding enough Pareto points, the designer cannot have an estimation of the shape of the trade-off curve/surface.

Different procedures, such as *physical programming* and *normal boundary intersection* (NBI) have been developed to overcome this problem. *Physical programming* (Messac (2000)) employs a flexible problem formulation framework. The designer does not need to specify weights in the problem formulation; rather the designer specifies ranges of different degrees of desirability for each design measure. The *normal boundary intersection* (NBI) (Das and Dennis (1998)) is an alternative method for a general nonlinear multicriteria optimisation problem. This method is independent of the relative scales of the functions and is successful in producing an evenly distributed set of points in the Pareto set given an evenly distributed set of parameters.

7.2.2 Sensitivities

The use of sensitivities allows us to determine the effect that changes in the design variables produce on the performance of the structure. In the literature (Haftka and Gürdal (1992)), there are two methods used in sensitivity analysis calculations, the *variational* method and the *implicit differentiation* or *discrete* method. The former method is based on the material derivative concept. The latter involves differentiating the basic equations of the structural response.

There are three approaches to calculate the sensitivities using discrete methods. The first is the *finite difference* (FD) approach whereby the response analysis is applied to both the current and perturbed models, and the differences in response values are divided by the design perturbation (step-size) to obtain the sensitivity derivatives. It is a general approach and simple to implement. However, numerical difficulties and inaccuracies appear if a non-suitable step-size is used. The second approach is the *analytical differentiation* (AD). In this technique the governing equations are implicitly differentiated to treat the response derivatives. This is a more accurate method, but more complex to implement because there are analytical derivatives involved. Finally, the third approach is the *semi-analytical approach* (SA), which is developed mixing various levels of analytical with finite difference derivatives.

In practical situations, the two most commonly used sensitivity algorithms are the FD and the SA method. The FD is simple to implement, as said before, but is expensive because a complete analysis is required for each design variable. The SA method requires only a resolution for each design variable and, therefore, is less expensive than finite differences. However, both of these methods entail a perturbation of the design variable, and they are dependent on the choice of perturbation step-size.

7.2.3 Fast Reanalysis Techniques

In this work, sensitivity calculations are implemented together with reanalysis techniques. In this context, reanalysis consists of updating the analysis results

(*current state*) over a system after a small geometric change (*perturbed state*). Hence, the use of an accurate and efficient reanalysis method can speed up the process.

Some of the techniques available in the reanalysis literature (Abu Kassim and Topping (1987)) yield the exact modified response (*exact* or *direct* methods) whereas others give only an approximate solution for the modified structure (*approximate* or *iterative* methods). In general, the *exact* reanalysis methods tend to be slow and only suitable for small modifications. On the other hand, the *iterative* methods are efficient but the modified solution can be less accurate. In these methods, solution accuracy and rate of convergence are both important. Indeed, iterative methods may fail to converge in some cases, particularly for larger geometric changes (Kane *et al.* (1990)). Therefore, some compromise amongst the computational effort and efficiency has to be made.

In the FEM context, Kirsch has investigated different reanalysis methods for structural analysis such as the *reduction* method (Kirsch (1991)) or *combined approximations* (CA) (Kirsch (2002)). Most recently, Mackie (1998) presented a FEM reanalysis in an object-oriented framework. This approach uses substructuring and multithreading to improve the analysis performance. However, these FEM implementations present some difficulties of remeshing after geometric changes.

Reanalysis in the BEM framework takes the advantage of some of the techniques previously presented in FEM. However in this case, the system matrix A is dense and unsymmetric so the resolution techniques used in the FEM cannot be applied here. Starting from the matrix equation system (Trevelyan (1994)) previously presented in chapter 4

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{7.3}$$

the equation system following a design change can be updated as follows

$$\mathbf{A}_0\mathbf{x}_0 = \mathbf{b}_0 \tag{7.4}$$

where $A_0 = (A + \Delta A)$, $b_0 = (b + \Delta b)$ and the updated solution vector $x_0 = (x + \Delta x)$. Consider first the *exact* reanalysis methods for updating matrix inverses. At first, the matrix inverse of the original system A^{-1} is calculated. It is well known that an efficient solver would not operate by computing explicitly the inverse A^{-1}, but the computational effort taken due to the calculation of this inverse may be a price worth paying if the performance achieved in the reanalysis stage were exceptional. Hence, in the reanalysis the original system has the form

$$x = A^{-1}b \tag{7.5}$$

and the modified system

$$x_0 = A_0^{-1}b_0 \tag{7.6}$$

To obtain a rapid solution to the updated inverse $A_0^{-1} = (A + \Delta A)^{-1}$, where A^{-1} is known, there are classical approaches based on the Sherman-Morrison-Woodbury formulae (Press (1986)). However, these formulae are only efficient for problems in which A and A_0 differ only in a few rows and columns, i.e. changes in a relatively small number of elements. Since a BEM reanalysis, for single zone problems, involves changes to a large number of rows and columns, the Sherman-Morrison-Woodbury formulae can be inefficient. An alternative formula for updating matrix inverses has been presented by Castillo *et al.* (1998). However this approach is also computationally expensive for the type of modified matrices in the BEM.

Kane *et al.* (1990) presented an iterative method for reanalysis of BEM systems applying these techniques to shape optimisation problems. The method reformulates equation (7.4) after substitutions of the form

$$(A + \Delta A)(x + \Delta x) = (b + \Delta b) \tag{7.7}$$

Obtaining the following expression

$$A\Delta x = (\Delta b - \Delta Ax) - \Delta A\Delta x \tag{7.8}$$

The vector Δx is present on both sides of the expression, and this leads to an iterative solution. However, since A is involved in the iterative solution, this method will

converge for small to moderate changes. To improve convergence to more highly modified models the method may be extended by a scaling of **A**, and a reformulation of the recurrence.

More recently, Leu (1999) proposed an iterative scheme for reanalysis of BEM systems also applied to shape optimisation problems. This approach is based on a reduction method introduced by Kirsch (2002). In this method the updated solution vector is expressed as a linear combination of a reduced set of basis vectors. This has the advantage that the number of these vectors is far less than the number of structural degrees of freedom, i.e. the modified behaviour of the structure is approximated using fewer degrees of freedom.

Trevelyan *et al.* (2002) presented a reanalysis scheme, based on the use of the iterative solver GMRES (*generalized minimum residual method*, Saad and Schultz (1986)). In this approach, the previous system matrix **A** is overwritten and a full matrix solution performed for each reanalysis. The solution vector from the previous equation, i.e. vector **x**, is used as the initial estimate of the perturbed solution vector x_0. Hence, the matrix level description of the problem updates with each modification.

7.3 Algorithm

To carry out the optimisation, an integrated computational procedure is implemented. The integrated system incorporates several tools such as boundary representation, BE analysis, design sensitivity analysis and reanalysis techniques, all of them linked to the optimisation algorithm.

The FD is used in this study to calculate the sensitivity calculations. To this end, n perturbed systems for each intermediate design are performed; n being the number of design variables. Each perturbed system corresponds to the state where only one design variable is modified by a step size Δs and the rest of the design variables (n-1) remain unchanged. To evaluate the structural situation for each perturbed system, reanalyses are performed instead of a full analysis. Thus, the iterative solution

obtained using reanalysis is based on the results for the non-perturbed system. The analysis results such as displacements or stresses for each perturbed system are then used to evaluate the FD sensitivities of each design variable. Further, sensitivity numbers related to the variables are calculated and used as optimality criteria.

7.3.1 Design Sensitivities

The use of sensitivities allows us to determine the effect that changes in the design variables, control points in this case, produce on the performance of the structure. The FD approach is implemented together with the reanalysis technique available in Concept Analyst (Trevelyan *et al.* (2002)). The use of reanalysis avoids full additional analyses for each design variable. Therefore, the response analysis is carried out for the current design (original design) and also for the perturbed design. The difference between them is divided by the step size.

$$\alpha^s = \frac{\partial f(s)}{\partial s} \approx \frac{f(s + \Delta s) - f(s)}{\Delta s} \tag{7.9}$$

where s is the perturbed variable. Following equation (7.9) a sensitivity number α_j^i is calculated for each changeable control point i and each criterion j. In this way, the sensitivities are computed for all the design variables. In some cases, the numerical quantities of various sensitivities of different criteria may have significant variations with respect to each other; therefore it is more practical to normalise them. These sensitivities are normalised by taking the maximum value of the sensitivity number; i.e. all the sensitivities are rated according to this maximum value. Thus, a global ratio R_j^i is associated to each design variable i and each criterion j.

$$R_j^i = \frac{\alpha_j^i}{\alpha_j^{max}} \tag{7.10}$$

α_j^i is the jth criteria sensitivity number for each control point i and α_j^{max} is the maximum value of the jth criteria sensitivity number.

177

The notation used here for the sensitivity numbers is the same as the one used in most of ESO literature (Steven *et al.* (2002), Li *et al.* (2001)). Such an identical notation allows us a more straightforward comparison of results and concepts.

7.3.2 Removal and Addition Criteria

Regarding ESO, once the sensitivities are normalised (equation (7.10)), then they are assigned a weighting factor w_j. These sensitivity numbers are finally summed to form a new single criterion number R_i for each control point following the weighted sum method

$$R_i = \sum_{j=0}^{N} w_j R_j^i \qquad (7.11)$$

Thus R_i is the multiple criteria sensitivity and determines the removal and addition of material. Material can be removed from the structure if any design variable i satisfies

$$R_i \leq RR \qquad (7.12)$$

where RR is the *removal ratio*. In addition, material can be added if

$$R_i \geq AR \qquad (7.13)$$

where AR is the *addition ratio*. Generally, these ratios, in a similar fashion to that used in shape and topology optimisation, are not fixed for the entire process but can be updated by the *evolutionary rate ER*. For example in the case of material removal $RR_m = RR_{m-1} + ER$. The overall process of multi-criteria optimisation can be summarised in the following steps and in the flow chart of figure 7.1

1. Define the structural optimisation problem: Geometry, loads, constraints, Δs.

2. Perform the BE analysis

3. Save current solution and geometry

4. Sensitivity calculations

    4.1. For $i = 0$ to number of design variables n

4.1.1. Perform a perturbation Δs on the design variable i

4.1.2. Carry out reanalysis for the perturbed system

4.1.3. Calculate sensitivity of the criterion to the perturbed design variable i

4.1.4. Retrieve solution and geometry

5. Modify the design according to the sensitivity calculations

6. Check the optimisation convergence criteria

   6.1. Stop if they are satisfied

   6.2. Otherwise update design and go to 2

Boundary Element Analysis

⟱

Calculation of sensitivity numbers

---

1. for ($i=0$; $i<n$; $i++$)

   1.1. $P_i(s + \Delta s)$: Perturb model by moving Δs design variable i

   1.2. Reanalysis

   1.3. Sensitivity α_j^i for each criterion j to ΔP_i by finite difference approximation

   1.4. Restore current model. Retrieve current model solution

2. Calculate a sensitivity ratio R_j^i for each design variable: $R_j^i = \dfrac{\alpha_j^i}{\alpha_j^{\max}}$

⟱

Optimisation Criteria

---

1. Calculate the Pareto front using Weighted Sum Method

   1.1. for ($j=0$; $j<N$; $j++$)

      1.1.1. Select the weights w_j: $\sum\limits_{j=0}^{N} w_j = 1$

      1.1.2. Calculate weighted sensitivity ratio R_i for each i: $R_i = \sum\limits_{j=0}^{N} w_j R_j^i$

2. Removal criterion: $R_i \leq RR$

3. Addition criterion: $R_i \geq AR$

Figure 7.1: Flow chart

179

7.3.3 Influence of the step-size

When using finite differences to approximate derivatives there are two sources of error; i.e. *truncation* and *condition errors* (Haftka and Gürdal (1992)). Truncation error e_T is a result of the neglected terms in the Taylor series expansion of the perturbed function. For example for the Taylor series expansion of $f(s+\Delta s)$

$$f(s + \Delta s) = f(s) + \Delta s \frac{\delta f(s)}{\delta s} + \frac{(\Delta s)^2}{2!} \frac{\delta^2 f(s)}{\delta s^2} + \frac{(\Delta s)^3}{3!} \frac{\delta^3 f(s)}{\delta s^3} + \dots \qquad (7.14)$$

the truncation error for the finite difference approximation is proportional to the step size

$$e_T(\Delta s) = \frac{\Delta s}{2!} \frac{\delta^2 f(s)}{\delta s^2} + \dots \qquad (7.15)$$

Condition error e_C is the difference between the numerical evaluation of the function and its exact value. It is related to the round-off error and can be comparatively important if the step size is particularly small. Condition error is also related to ill-conditioned numerical problems and iterative process.

$$e_C(\Delta s) = \frac{2}{\Delta s} \varepsilon \qquad (7.16)$$

where ε is the bound on the absolute error of the function f.

Both errors yield to the *step size dilemma*. If we select a step size to be small enough to reduce the truncation error, then we may have an excessive condition error and vice versa. The question is then how to establish these limits in our problems since the condition error is directly related to the boundary element software used.

To compute the range of valid step sizes a simple test is carried out for a general example using Concept Analyst. Thus, by taking a geometry defined by NURBS curves, and perturbing Δs the position of one selected control point, then the displacement of its nearest mesh point A is obtained. The sensitivities of the displacement of mesh point A to the perturbation are computed and the process

repeated for a different range of values. The results of the finite difference calculations for different step sizes are shown in figure 7.2. We do not compare the accuracy of the results with the analytical solution but as an indicator of the suitable step size values for the problems tackled and software used. The horizontal axis in figure 7.2 shows the ratio of the step size Δs to the maximum dimension h, i.e. (Δs/h). Note that for very small step size ratios Δs/h $\leq$ 0.00001 the finite difference result decreases sharply, which is due to the condition error. On the contrary, for higher step size ratios Δs/h $\geq$ 0.01 the function rapidly increases, in this case this behaviour is driven by the truncation error. Clearly, for step size ratio such as 0.00001 $\leq$ Δs/h $\leq$ 0.01, the finite differences results exhibit a stable region and therefore, this is the interval chosen for the step size ratio in the sensitivity calculations

$$0.00001 \leq \Delta s/h \leq 0.01 \tag{7.17}$$

where Δs is the step size and h is the maximum dimension of the object. Further, for practical reasons, this step size is also set for the actual movement of the control points when removing and adding material ensuring accuracy in the sensitivity results as well as in the evolutionary process.
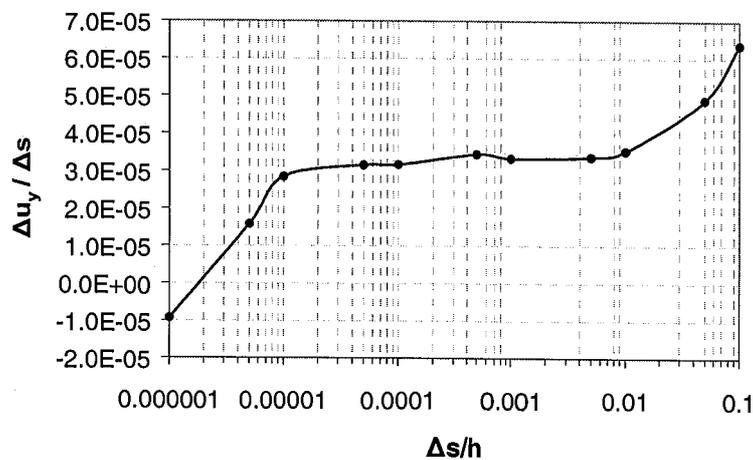
Figure 7.2: Effect of step size on FD of the vertical displacement at mesh point A

7.4 Numerical Example

One example in two-dimensional elasticity is presented below to verify the accuracy and efficiency of the proposed method for multi-criteria.

7.4.1 Connecting Rod

In this example, the sensitivity approach is applied in order to solve the multi-criteria problem. This problem was analysed by different authors for a single criterion (Sienz and Hinton (1997), volume minimisation) and more recently for multiple criteria (Afonso *et al.* (2002), volume and strain energy minimisation). In order to compare results, the objective functions considered are the volume ($f_V = V$) and the strain energy ($f_U = U$).

The initial domain is shown in figure 7.3 (a). The connecting rod has the following material properties: Young's modulus $E = 210000 N/mm^2$, Poisson's ratio $\nu = 0.3$, thickness $t = 1$ mm. Plane stress conditions are assumed. The rod is clamped in the middle of the left hand side edge ($u_x = u_y = 0$) and roller conditions are imposed over the top and bottom of this edge ($u_x = 0$). The load condition is applied in the normal direction with a cosine distribution over the right hand side of the inner circle, such that the resultant force has a value of $q_x = 1000$ N.

(a)   (b)
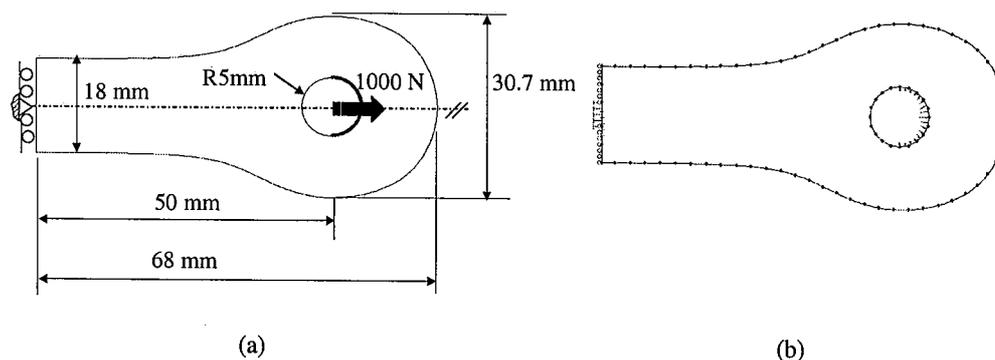
Figure 7.3: Problem definition of a connecting rod (a). Boundary element mesh (b)

Figure 7.3 (b) shows the boundary element mesh used. This element mesh has 87 quadratic elements and 174 nodes. A convergence analysis shows that this mesh is

fine enough to produce accurate solutions for the sensitivity calculations. Figure 7.4 shows the initial distribution of control points. Two NURBS curves are used to define the changeable geometry. The first curve is defined by 6 control points identified as P_0, P_1, P_2, P_3, P_4 and P_5. The second NURBS curve is defined by the control points P_5, P_6, P_7, P_8, P_9 and P_{10}. Note that both curves are joined at control point P_5.

To achieve the minimum strain energy design or stiffest design, the overall strain energy of the structure is minimised (min U). In fact, the strain energy sensitivity number R^i_U (equation (7.10)) gives the gradient of the strain energy U with respect to the variation Δs in the position of control point P_i. This sensitivity number is used to guarantee that f_U steadily decreases during a change of the location of the control points.

	Coordinates (mm)	
	x	y
P_0	0	-8
P_1	20	-8
P_2	34	-8
P_3	46.25	-18.75
P_4	67.5	-12.5
P_5	68	0
P_6	67.5	12.5
P_7	46.25	18.75
P_8	34	8
P_9	20	8
P_{10}	0	8
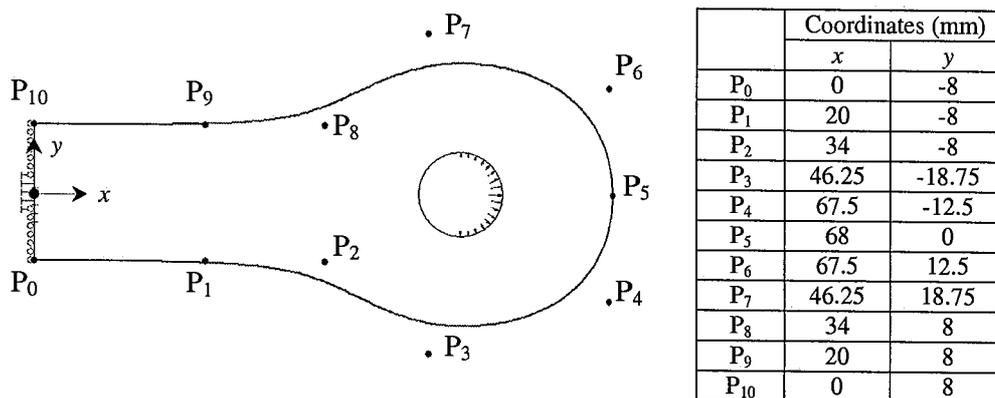
Figure 7.4: Control point distribution

To achieve a minimum volume design, the volume of the structure is minimised (min V). Similarly, the volume sensitivity number R^i_V (equation (7.10)) represents the gradient of the volume V with respect to the control point, P_i, variation. This sensitivity is used to direct the control point movement so that the volume is continuously reduced.

To achieve both strain energy minimisation and volume minimisation, an overall objective function, constructed using a weighted sum scheme, is minimised; i.e. min $f = (w_U U + w_V V)$. The 11 control points, shown in figure 7.4, are taken as design

variables. Constraints on the maximum displacement are imposed to control points P_0, P_1, P_9 and P_{10} as stated in the equation below. This constraint is imposed to avoid the creation of notches on the symmetry line. Moreover control point P_5 is also constrained to movement along the x axis; i.e. symmetry axis.

$$\min \quad f = w_U U + w_V V$$

$$\text{subject to} \quad \begin{cases} \left| P_i - y \right|_y \geq 5\,\text{mm} & i = 0,1,9,10 \\[2mm] \left| P_5^{m+1} - P_5^m \right|_y = 0 \end{cases}$$

where m is the current iteration and $m+1$ is the perturbed iteration. w_U and w_V represent weighting factors for strain energy and volume criteria respectively, and are taken as $w_U + w_V = 1$. The gradient of the new single objective function with respect to the control point variable P_i (Δs_i) is as equation (7.11)

$$\frac{\Delta f}{\Delta s_i} = w_U R_U^i + w_V R_V^i = R_i$$

which is used to estimate an overall effect of the variation of control point location on the structural strain energy and the volume. The weighting factors emphasise the importance of the various criteria differently, in addition they allow balancing the two criteria, with different physical meanings. It should be noticed that the strain energy and volume sensitivities have been normalised since they may have different dimensions and so that, the weighting factor would not appropriately reflect an equal emphasis for each criterion. The weighting factors obviously have to be modified depending on the units used to express strain energy and volume.

Thus, in the ESO procedure, the sequence of material removal and addition is determined by the sensitivity numbers R_i. In other words, the control point sensitivity numbers are regarded as their relative contribution to variations in the design objectives. The optimisation process is implemented by progressively adding and removing material from the model according to these sensitivity numbers. Thus material is removed from the control point regions with the smallest sensitivity

numbers (R_i < 0; i.e. RR = 0) and then added to the elements with the largest sensitivity numbers (R_i > 0; i.e. AR = 0).

The objective is achieved in a progressive manner since the material is removed/added according to the step size used in the sensitivity calculations, which in this case is set to Δs = 0.05. The step size ratio is $\Delta s/h$ = 0.05 mm/68 mm = 0.000735. This value is within the interval $0.00001 \leq \Delta s/h \leq 0.01$ (equation (7.17)). Moreover, numerical tests carried out on this example for different step sizes have shown the same agreement. As said before, the movement of the control points is related to the step size. Control points with the most significant sensitivities (highest or lowest sensitivity values) are moved a distance equal to the step size. In addition, control points with less significant sensitivities are moved a percentage of this step size; i.e. $k_{removal}\Delta s$ and $k_{addition}\Delta s$. This percentage is taken proportional to the maximum ($k_{addition}$) or minimum ($k_{removal}$) sensitivity numbers.

Generally, to calculate the structural response, Concept Analyst performs full analyses using Gauss elimination. Besides, one useful feature implemented in this software is the rapid response under small geometry changes by using reanalysis. The reanalysis scheme used in Concept Analyst (Trevelyan *et al.* (2002)) is based on the use of an iterative solver GMRES. In this approach, the system matrix **A** (equation (7.3)) is overwritten and a full matrix solution is performed for each reanalysis. The previous solution (to the unperturbed state) provides the initial *guess* as a starting vector in the iterative process. For perturbations of the size used in sensitivity calculations, the first guess is likely to be close to the solution of the perturbed state. In our case, during the sensitivity calculations, for each perturbed state the boundary geometry slightly changes, and so it makes possible to consider the implementation of reanalysis techniques.

Consider the connecting rod example with 11 design variables and for a set of weights w_U = 0.3 and w_V = 0.7. Reanalysis vs. full analysis is compared for a single iteration that comprises 11 sensitivity calculations and consequently, 11 analyses. In the first case, full analyses are carried out and reanalyses in the second case. On a Pentium 4 (2 GHz) processor, the computation of 11 reanalyses and sensitivities takes 2.0350 seconds. In the case of full analyses, it takes 9.0904 seconds. Therefore,

185

for this particular example, a single iteration of the optimisation process using reanalysis is 4.46 times faster than performing a full analysis.

Figure 7.5 shows the Pareto front obtained using the WS method by a set of weighting factors $\lambda = w_U = [0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1]$ and consequently $w_V = [1, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1, 0]$. In this example, the Pareto curve is successfully obtained from an evenly distributed set of weights which produce a spread distribution of points defining the Pareto curve.
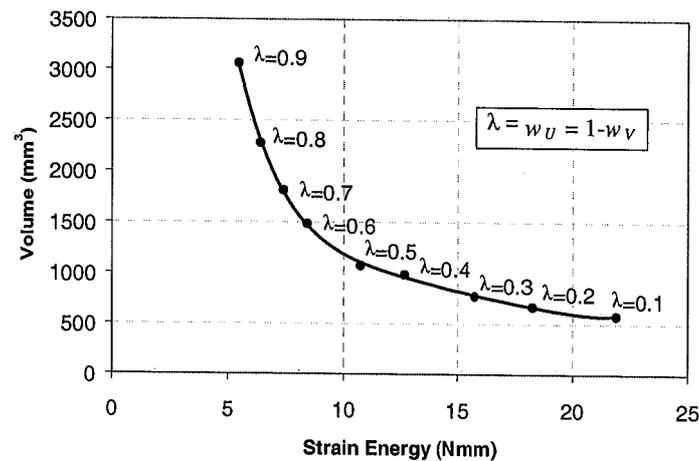
Figure 7.5: Pareto front

In fig.7.6 are shown different optimum solutions according to the selected weights for the strain energy criterion (w_U) and the volume criterion (w_V), respectively. For the case of figures (b) and (c) a *neck* shape appears due to the constraint limit to the vertical movement of points P_0, P_1, P_9 and P_{10}, but not to points P_3 and P_8. Moreover, examining these figures (b) and (c) we can state that more control points defining the circular region would increase the smoothness in this area, especially closer to the union between the two curves. Also notice that there is no symmetry being forced apart from the horizontal movement of point P_5. In spite of this, almost symmetric geometries are produced for all cases.

(a) Original geometry
V= 1367.85 mm³, U= 9.76 Nmm

(b) w_U = 0.1, w_V = 0.9
V= 579.65 mm³, U= 21.89 Nmm

(c) w_U = 0.2, w_V = 0.8
V= 667.83 mm³, U= 12.25 Nmm

(d) w_U = 0.3, w_V = 0.7
V= 772.73 mm³, U= 15.76 Nmm

(e) w_U = 0.4, w_V = 0.6
V= 982.93.65 mm³, U= 12.62 Nmm

(f) w_U = 0.5, w_V = 0.5
V= 1071.85 mm³, U= 10.711 Nmm

(g) w_U = 0.6, w_V = 0.4
V= 1483.14 mm³, U= 8.41 Nmm

(h) w_U = 0.7, w_V = 0.3
V= 1812.57 mm³, U= 7.40 Nmm

(i) w_U = 0.8, w_V = 0.2
V= 2276.50 mm³, U= 6.40 Nmm

(j) w_U = 0.9, w_V = 0.1
V= 2908.60 mm³, U= 5.60 Nmm
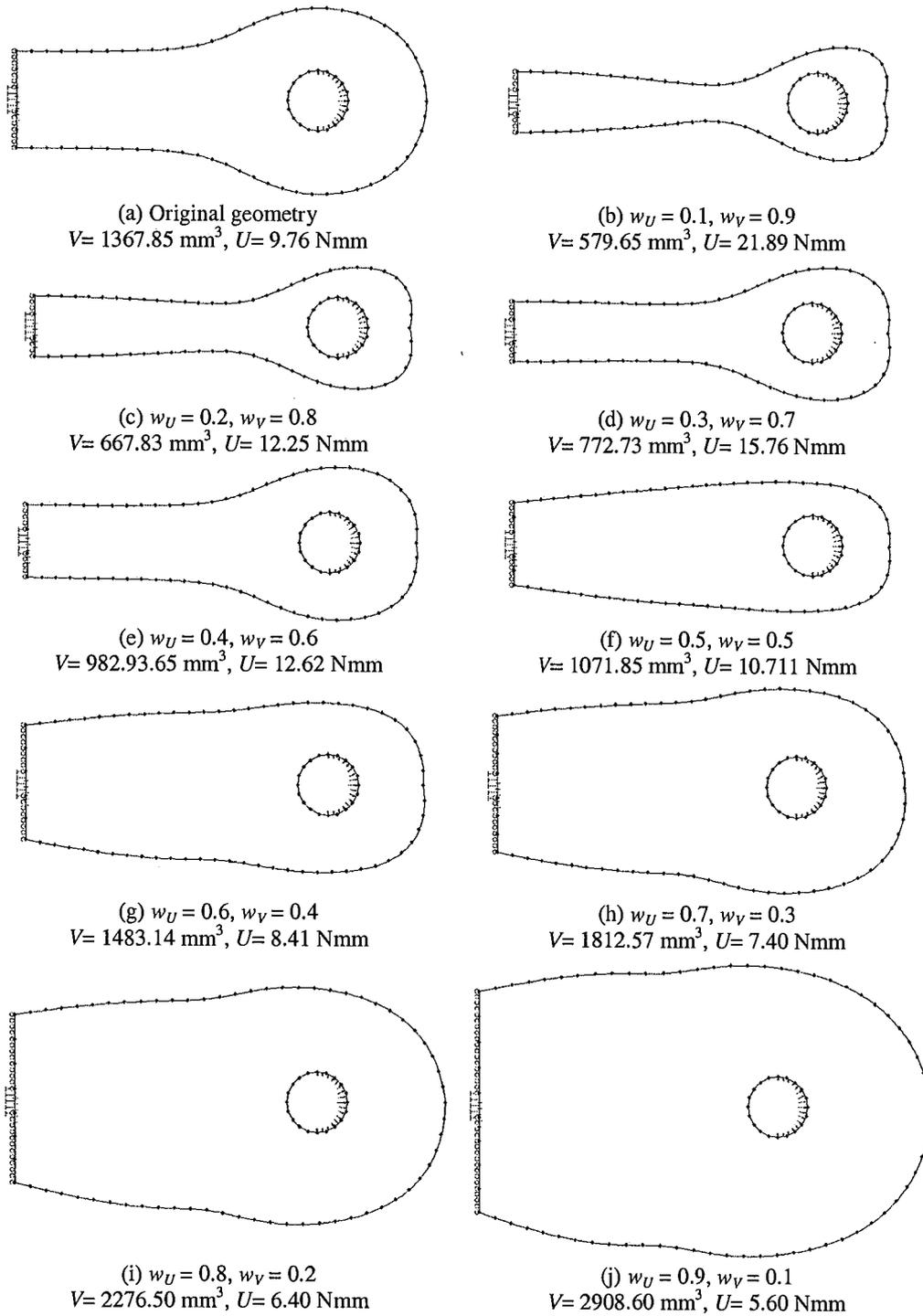
Figure 7.6: Optimum designs for different set of weights

Comparing these results to Afonso *et al.* (2002) which uses MP and FEA, the solutions depicted in figures (e), (f), (g), and (h) are the most similar. Different results are found for the other cases. The reason is that for moving the control points

187

a percentage of the step size is used, that is, *taking into account* the control points with higher positive or lower negative sensitivities and *disregarding* the control points with near zero sensitivities. This means that the initial shape of the connecting rod is not necessarily maintained even for cases in which only removal or only addition takes place. The fine boundary element mesh used in the present study is highly likely to generate more accurate results than those presented in Afonso *et al.* (2002) in which a more coarse finite element mesh is used, even for critical areas. In Sienz and Hinton (1997) the accuracy of the results (for one single load case) is considerably improved by using adaptivity techniques.

Figures 7.7 (a) and 7.7 (b) show the evolution history of the normalised strain energy (U/U_0) versus the normalised volume (V/V_0) and the averaged objective function which results from the weighted sum of the normalised strain energy and volume. Figure 7.7 (a) plots the evolution for a set of weights of $w_U = 0.3$ and $w_V = 0.7$. Figure 7.7 (b) displays this result for a set of weights $w_U = 0.8$ and $w_V = 0.2$. It is shown from this graphs that the two objectives are in conflict.

(a) $w_U = 0.3$, $w_V = 0.7$ (b) $w_U = 0.8$, $w_V = 0.2$
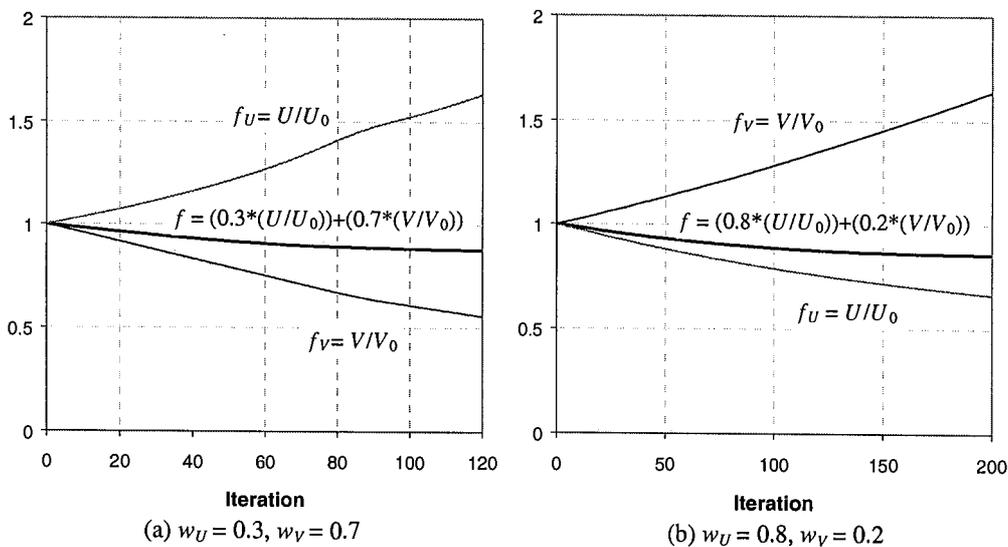
Figure 7.7: Evolution history of objective functions and averaged objective

The effects of weighting factor w_U and w_V on the strain energy objective (U/U_{max}) and volume objective (V/V_{max}) are displayed in figure 7.8. By increasing the volume the strain energy is reduced and conversely, the volume is reduced if the strain

energy increases. This effect is not linear which means that a 50%-50% allocation of weights may not reflect an equal satisfaction of both criteria. In this figure 7.8, the relation between the stress weights and the optimality criteria is shown. It is clear that, with the increase in the strain energy weight ($w_U\uparrow$) the strain energy reduction becomes more significant, but in turn this leads to higher volume. The critical weights are found for $\hat{w}_U = 0.56$ and $\hat{w}_V = 0.44$, meaning that if the strain energy minimisation is a major objective, the strain energy weighting factor should be 0.56 $\leq w_U \leq 1$ ($0 \leq w_V \leq 0.44$); and if the volume minimisation is a major objective then the strain energy weighting factor should be $0 \leq w_U \leq 0.56$ ($0.44 \leq w_V \leq 1$).
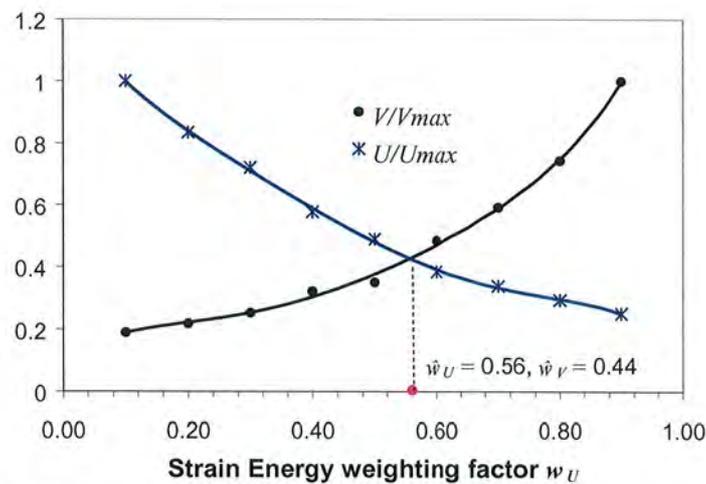
Figure 7.8: Effects of the weighting factors on the optimal objectives

Finally, the von Mises stress contour plots are displayed in figure 7.9 for different sets of weights. The consideration of such stress distributions is essential in the design of suitable FE or BE meshes for the optimisation. It can be noticed that for all cases the von Mises stress maximum is localised around the inner circle where the load is applied but there are no high stress peaks in the design regions.

(a) *Original design*   (b) $w_U = 0.56$, $w_V = 0.44$

(c) $w_U = 0.3$, $w_V = 0.7$ (d) $w_U = 0.8$, $w_V = 0.2$
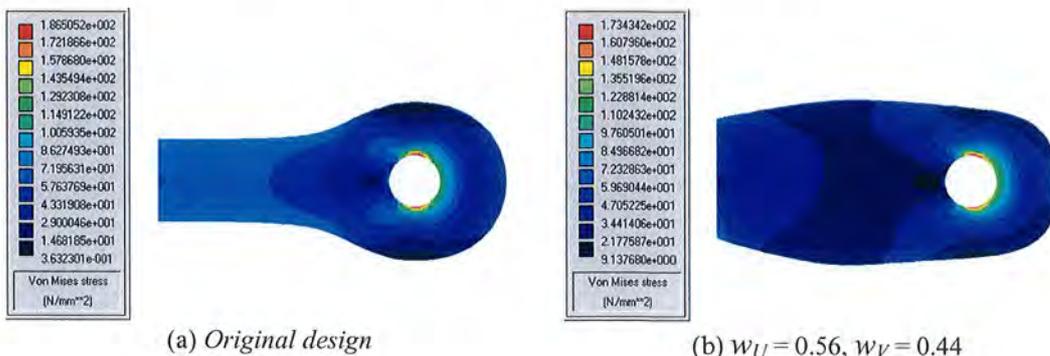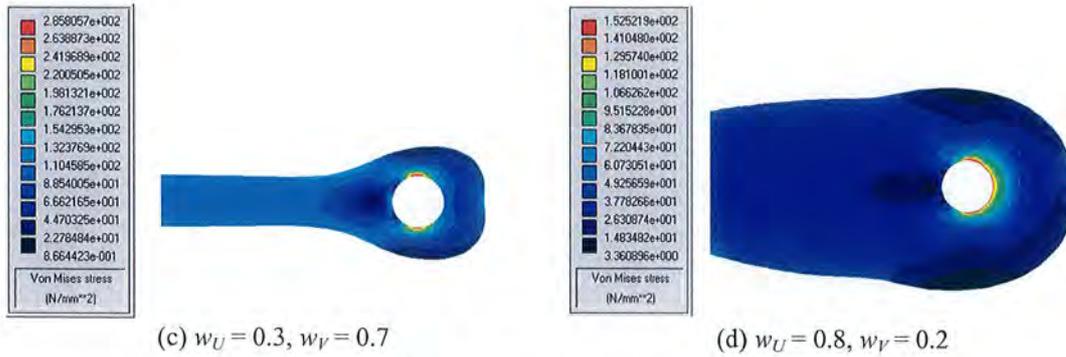
Figure 7.9: Von Mises stress contour plots

7.5 Discussion and Conclusions

Multi-criteria optimisation problems have been investigated in this chapter. In such problems the objective functions are in conflict. None of the possible solutions simultaneously fulfils all the objectives, and also the individual solutions of the objective functions are different. The solution of the multi-criteria problem is obtained by transforming the multiple objective functions into a single objective function using the weighed sum method (WS). The Pareto concept is used to gain knowledge about trade-offs between objectives and also allows the choice of the most desirable solution. To calculate the effect of changes in the design variables on the objective functions, sensitivity numbers are calculated using finite difference (FD) derivatives. The technique has been applied to the connecting rod example, showing it to be efficient obtaining an even spread set of Pareto optimal points for an even distribution of weights. However, for other examples this cannot be the case (Das and Dennis (1997)). The sensitivity calculations are performed using BEM reanalysis based on the non-perturbed solution rather than full analyses for each perturbed state. It has been shown that this considerably reduces the computing time for each iteration.

8

STRUCTURAL OPTIMISATION IN 3D

8.1 Overview

This chapter presents the developed approach applied to three-dimensional structural optimisation problems. This algorithm for 3D optimisation is an extension of the two-dimensional gradientless method presented previously in the chapters *shape optimisation in* 2D and *topology optimisation in* 2D, which are chapter 5 and chapter 6, respectively. Similarly, the optimisation algorithm is based on the *evolutionary structural optimisation method* (ESO) and *boundary elements* (BE) are used to carry out the structural analysis. The geometries are represented using *nonuniform rational B-spline* (NURBS) curves and surfaces. Unlike the 2D algorithm, a commercial boundary element software package is used. The optimisation algorithm is no longer fully integrated in the structural analysis program which introduces some limitations handling the analysis data. Examples are presented to show the effectiveness of the algorithm for some preliminary results.

8.2 Introduction

There are different techniques used in 3D structural optimisation problems, and most of them are a straightforward extension from 2D optimisation techniques. The general formulation of these techniques has been presented in the structural

optimisation review chapter (chapter 2) and also recalled in the 2D optimisation chapters (chapters 5, 6 and 7). These optimisation techniques can be broadly divided into *deterministic-gradient based* methods and *heuristic* or *scholastic* techniques.

The deterministic-gradient based methods applied to 3D can be subdivided into *discrete* and *continuum methods*. In the discrete method the finite (Jacobsen *et al.* (1998)) or boundary element (Kane *et al.* (1992)) equations are directly differentiated. In the continuum approach, the variational equilibrium equations are differentiated and discretised (Haug *et al.* (1986)). Alternatively to the gradient-based methods, applications of the heuristic or scholastic methods in 3D include the evolutionary structural optimisation method (ESO) (Young *et al.* (1999)) and genetic algorithms (Annicchiarico and Cerrolaza (2001)).

An important feature in three-dimensional optimisation problems is the model representation and how it is related to the structural analysis results. Most of the work on shape optimisation relates the finite/boundary element data with the model geometry (Haftka and Grandhi (1986)). Initially, some authors defined the nodal coordinates of the discrete finite/boundary (Kane *et al.* (1992)) element model as design variables. However, this requires a large number of design variables and also it is difficult to maintain an adequate finite/boundary element mesh during the optimisation process (Haftka and Grandhi (1986)). New methods appeared to overcome these initial drawbacks such as the mesh parameterisation (Haug *et al.* (1986)) and solid modelling, which comprises constructive solid geometry (CSG) (Kodiyalam and Vanderplaats (1992)) and boundary representation (B-Rep) (Schramm and Pilkey (1993)).

ESO applied to three-dimensional domain problems is a straightforward extension from 2D and therefore, it is based on the simple idea that the optimum design evolves by slowly removing and adding material. The approach in 3D described in this chapter is focused on the classical gradientless ESO (Xie and Steven (1997)) i.e. material is removed from low-stressed areas and added to high-stressed regions.

The boundary element method (BEM) is used to carry out the elastostatic structural analysis in 3D. In addition, the boundary shape is represented by NURBS curves and

192

surfaces, defined by control points. The NURBS representation is attractive since it can be exchanged with CAD/CAM systems. Moving the basic parameters of the NURBS (control points) can modify the geometry of the NURBS curve or surface.

8.3 Algorithm

The optimisation algorithm in 3D is an extension of the gradientless algorithm presented in 2D, chapters 5 and 6, i.e. the optimum topology evolves by slowly removing and adding material from low and high-stressed areas, respectively. Similarly to 2D, the numerical analysis is carried out with the BEM but in this case, a commercial software package called BEASY (BEASY users manual (2002)) is used. The boundaries are represented using NURBS surfaces defined by control point nets. Use is made of the locations of these control points as design variables of the problem. Figure 8.1 shows a basic flow chart of the optimisation process. This process can be divided into several steps which are associated with either BEASY (structural analysis) or the OPTIMISER (boundary-ESO algorithm).

Step 1 (BEASY): The geometry of the structure is defined. The initial design is subjected to a set of loads and constraints.

Step 2 (BEASY): A boundary element analysis follows the model description.

Step 3 (OPTIMISER): The surface information is read together with the structural analysis results

- Read solution from BEASY and sort boundary nodes according to their stress levels.

- Read surface information. Process NURBS surface data and calculate normal vectors to surfaces.

Step 4 (OPTIMISER): Removal of material. The nearest control points to the least stressed nodes are identified and moved in the direction of the normal vector to that area to generate a more efficient shape.

193

Step 5 (OPTIMISER): Addition of material. If any node is found with a stress higher than the yield stress or any other maximum stress criterion, then a similar process to removal is undertaken but, however, the opposite direction of movement of the control points results in a material addition.

Step 6 (OPTIMISER): The output file is written for the modified geometry.

Step 7: Such a procedure is repeated (from step 2) until the stopping criterion is reached. This minimum is not mathematically searched but is found according to the evolution of the objective function.

Step 8 (BEASY): Post-process final design.

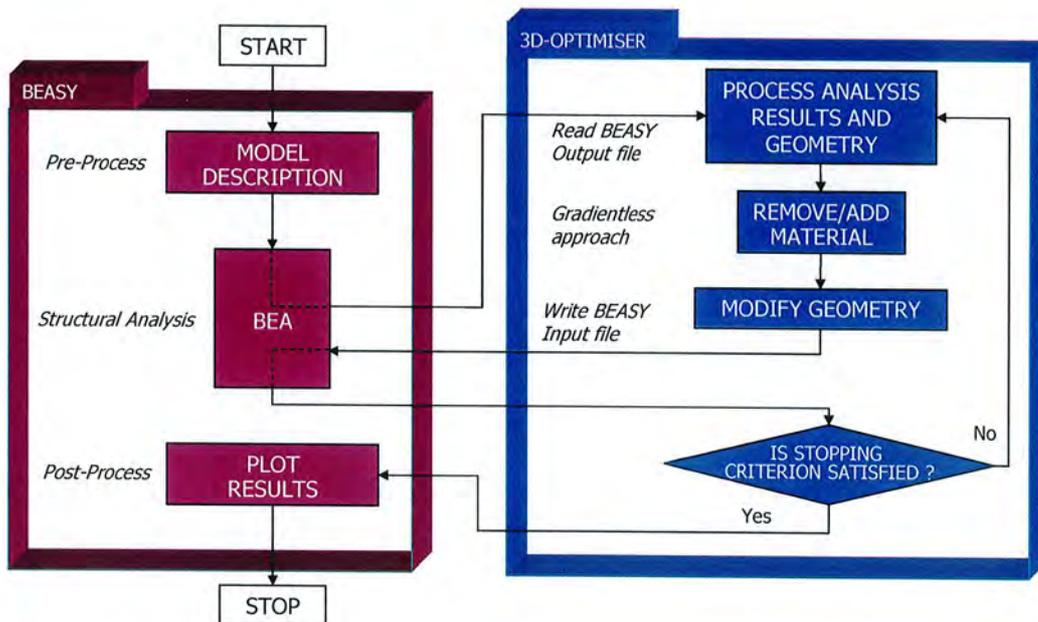**Figure 8.1**: Flow chart of the basic optimisation process in 3D

8.4   Geometry Definition

The initial geometry is defined in the first step of the optimisation process. This geometry can be created directly in the BEASY pre-processor or read into BEASY from an IGES file exported from a CAD program. This geometry is subjected to a set of loads and constraints, which determine the two possible types of surfaces, i.e.

design and *non-design* surfaces. Therefore, surfaces that can change freely along the process are identified as design surfaces, whereas these surfaces that cannot change due to loads, constraints or design restrictions are identified as non-design.

NURBS surfaces are adopted to define the geometry in order to control the curvature and tangency of the changeable boundary. A control point net is specified for each surface. Adjacent surfaces are required to share control points over their coincident edge and therefore ensure continuity between surfaces along the process.

Spline curves define the boundaries of the surfaces, acting as trim lines. Splines are used instead of NURBS since BEASY does not recognise these curves but only the more basic splines.

8.5   Boundary Element Model

The boundary element method is used for the structural analysis of the 3D topologies. As previously stated in the BEM chapter (chapter 4), and later referred to in the 2D optimisation chapters (chapters 5, 6 and 7), in this computational method, the boundary integral equations are approximated by a set of discretised integral equations. As a result, the boundary surface is divided into elements, thus the response is given at the nodes associated with the elements.

As presented before, in 3D problems a commercial software called BEASY (BEASY (2002)) is used. The use of this commercial software instead of the in-house software used in 2D problems brings some considerable differences in the problem definition and also while processing and managing the analysis data. The reason for the adoption of different software in this case is because at the current stage, Concept Analyst (Trevelyan and Wang (2001)) was not developed enough for 3D problems. In addition, the use of alternative software would prove the portability and flexibility of the algorithm. Figure 8.2 shows a typical example of a 3D quadratic boundary element mesh displayed using BEASY. It should be noted that the program includes a class of elements known as discontinuous, in which nodes are not shared with neighbouring elements, and which allow meshes to become discontinuous. Areas of

discontinuous meshing are evident in this figure and in others that follow in later sections of this chapter.
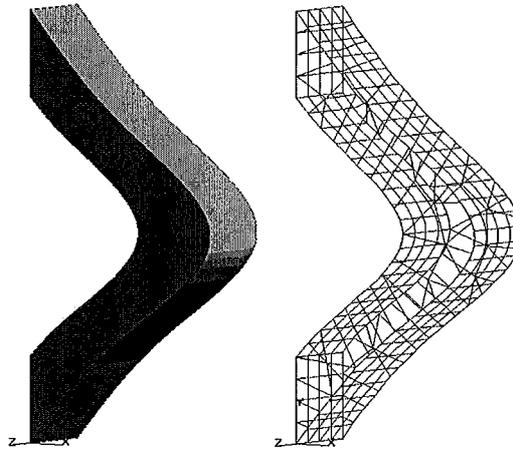
Figure 8.2: Illustration of a quadratic boundary element mesh in 3D

To assess the accuracy of the analysis BEASY provides several features to check the results and also identify any error in the model definition that can be the source of further potential errors. Model convergence is checked by examining the *equilibrium check* results and the *error norms* display. The equilibrium check verifies that the loads that have been applied to the model are in equilibrium with the reactions at the constraints. A poor equilibrium is a clear sign that the model has not converged. Error norms are calculated from inter-element stress discontinuities, and therefore provide a local control estimator that complements well the global equilibrium check. The error norm contours would give an estimated accuracy of the results for each element. They provide an estimation of the error in the form of the percentage of the stress. These error norms are treated as a guide only since it is not possible to compute the exact error.

In cases in which the mesh of the model needs to be improved to ensure convergence the alternatives are to increase the order of the element or to refine the mesh. The feasible option in this case is to refine the mesh by increasing the number of elements. This alternative is more general since an automatic meshing is set to the whole domain which keeps the same element mesh properties for all surfaces.

8.6 Removal and Addition of Material

8.6.1 Identifying Inefficient and Critical Areas

In a similar way to the 2D gradientless method the nodes from the boundary element mesh are sorted in ascending order according to their stress level, generally the von Mises stress is the stress criterion used (see section 5.6.1). Basically, although all nodes are initially sorted only nodes related to changeable regions would be considered later. Material would be removed from the structure if there is any node p that satisfies the following equation (8.1)

$$\sigma_p \leq RR\sigma_{max} \tag{8.1}$$

and added to the structure if any node satisfies equation (8.2)

$$\sigma_p \geq \sigma_y \quad OR \quad \sigma_p \geq AR\sigma_{max} \tag{8.2}$$

where σ_p is the node von Mises stress, σ_{max} is the maximum von Mises stress within the structure, which varies along the process, σ_y is the yield stress or any other maximum stress criterion, RR and AR are the *removal* and *addition ratio*, respectively, with $0 \leq RR, AR \leq 1$.

If a *steady state* is reached in which no nodes, or only a few nodes, can satisfy the previous equation (8.1) then the RR is incremented by the *evolutionary rate ER_R*

$$RR_i = RR_{i-1} + ER_R \tag{8.3}$$

Similarly, if only a few nodes can satisfy equation (8.2) then AR is decreased by the *evolutionary rate ER_A*

$$AR_i = AR_{i-1} - ER_A \tag{8.4}$$

Typical values determined from numerical experience for suitable ratios are $RR_0 = 0.05$, $ER_R = 0.01$, $AR = 0.95$ and $ER_A = 0$.

For each selected node p_i, the program searches its nearest control point P_i. Therefore, each node has a control point related as shown in figure 8.3, where p_0 is related to P_0 situated a distance a_0 away. The positions of these control points are treated as the design variables.

Sometimes the same control point can be associated to different nodes as shown in figure 8.3 for P_1. To consider such situations each node has associated a weight ω_k. This weight expresses the influence of the node on the associated control point. Therefore, the closer the node to the control point the higher its influence and therefore the weight. Consequently the weights are calculated as follows

$$\omega'_k = \frac{a_k}{\sum_{l=0}^{m-1} a_l} \tag{8.5}$$

with a_k the distance between the node and the control point. m is the number of nodes associated to the control point. Notice that for a unique node related to the control point ($m = 1$) this weight is 1 ($\omega'_k = 1$).

$$\omega''_k = \frac{1}{\omega'_k} \tag{8.6}$$

To avoid numerical instabilities there is a minimum value allowed for a_k. This value, called ψ, is a factor related to the maximum dimension of the structure. After numerical tests this value is defined as

$$\psi = \text{max dimension}/100 \tag{8.7}$$

The algorithm forces a_k to be always greater than or equal to ψ ($a_k \geq \psi$). Finally, the weight ω_k

$$\omega_k = \frac{\omega''_k}{\sum_{l=0}^{m-1} \omega''_l} \tag{8.8}$$

198

and therefore as a result of defining w_k in such a way , $\sum_{k=0}^{m-1} \omega_k = 1$.
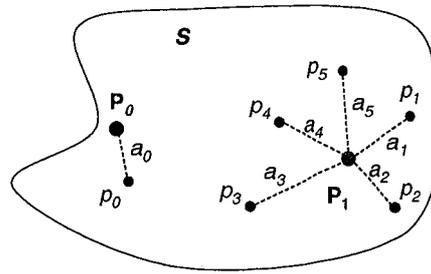
Figure 8.3: Selection of control points associated to nodes

8.6.2 Direction of Movement

The direction of movement of each control point selected is perpendicular to the NURBS surface defining the associated inefficient or critical area, i.e. where the nodes (p) are placed. The perpendicular movement (see figure 8.4 (a)) follows the normal vector n_p. This vector is calculated using the analytical definition of NURBS surfaces, which have been defined in chapter 3, its derivatives, i.e. v_p and u_p, and vector relationships (cross product $u_p \times v_p = n_p$). Similarly to 2D problems, the movement is inwards when removing material and outwards when adding it.
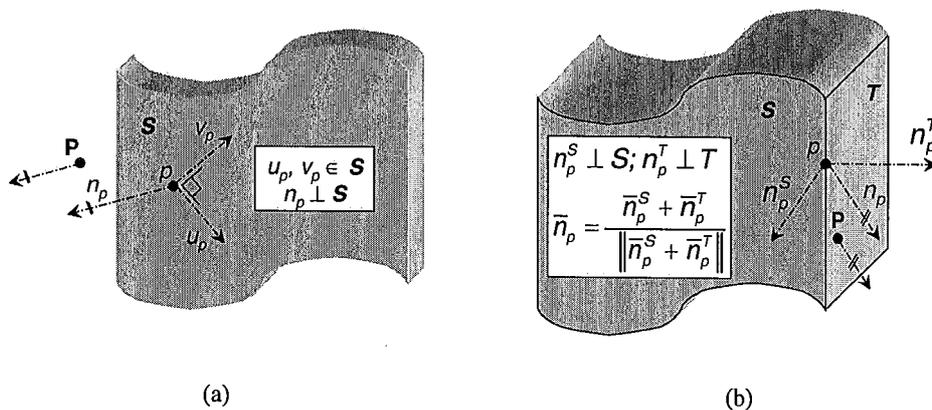
Figure 8.4: Direction of movement for control point P associated to node p. (a) p on surface S. (b) p on surfaces S and T

In the case of nodes placed on edges between surfaces (figure 8.4 (b)) an averaged normal is calculated. Therefore, knowing the vectors n^S_p and n^T_p which are the normal vectors at p to the surfaces S and T respectively, the direction of the new normal vector at p is calculated as that of a vector addition of vectors n^S_p and n^T_p. This criterion is not only applicable to two surfaces but to any number of surfaces sharing a node p.

For situations in which a single control point is associated to a group of nodes, as is illustrated in figure 8.5, the direction of movement n is calculated using the weight factors explained in the previous section (equation (8.8)) and mathematically stated

$$ n = \sum_{k=0}^{m-1} \omega_k n_k \tag{8.9} $$

where n_k is the normal vector to the node k and ω_k is its associated weight.
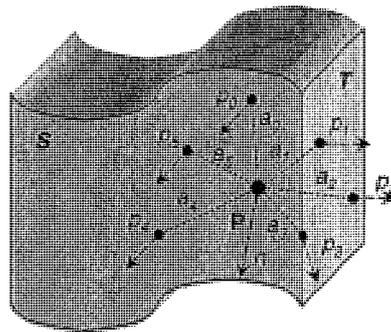
Figure 8.5: Direction of movement for control point P associated to a group of nodes

8.6.3   Distance to Move

In any iteration the material is either removed or added to the structure by migrating control points. Each control point is moved a distance related to the following parameters

- Distance of the control point from its associated node. This distance is denoted as a.

- A factor related to the stress situation within the structure at the current iteration. This factor is called the *removal factor* (*RF*) if removing material, and the *addition factor* (*AF*) if adding it. As presented in chapter 5, these parameters allow larger geometric changes in early iterations and smaller changes when fine-tuning is required in the later iterations.

For example in figure 8.6 the control point P_i situated at a distance a_i from the node is moved a quantity d_i calculated as follows

$$d_i = \frac{1}{a_i} RF \qquad a_i \geq \psi \qquad (8.10)$$

where ψ is defined in equation (8.7). Likewise, for addition of material

$$d_i = \frac{1}{a_i} AF \qquad a_i \geq \psi \qquad (8.11)$$

Therefore, the smaller the distance a_i the higher d_i, i.e. more material is moved. For cases in which several nodes influence a control point (figure 8.5), a more general expression of the distance of movement for the control point P_i is given as follows

$$d_i = \left[\sum_{k=0}^{m-1} \frac{1}{a_k} \right] RF \qquad (8.12)$$
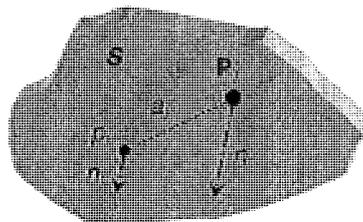
Figure 8.6: Control point and node distance relationship

In the case of removing material, there is another parameter to be considered, the so-called β ratio. This is a ratio that takes into account the least stressed areas amongst the lightly stressed areas within the structure. For each node p the β ratio can be defined as

$$\beta_p = \frac{\sigma_{min}}{\sigma_p} \qquad (8.13)$$

where σ_{min} ($\sigma_{min} > 0$) is the minimum von Mises stress within the structure and σ_p is the von Mises stress at node p ($\sigma_p \geq \sigma_{min}$). Thus, the β ratio is introduced into the equation (8.12) which is transformed in the new following expression

$$d_i = \left[\sum_{k=0}^{m-1} \frac{\beta_k}{a_k} \right] RF \qquad (8.14)$$

This ensures that removal takes place primarily in the most lightly stressed regions in preference to somewhat higher stressed regions that happen to be more densely meshed. The removal and addition process can be summarised in the flow chart presented in figure 8.7.
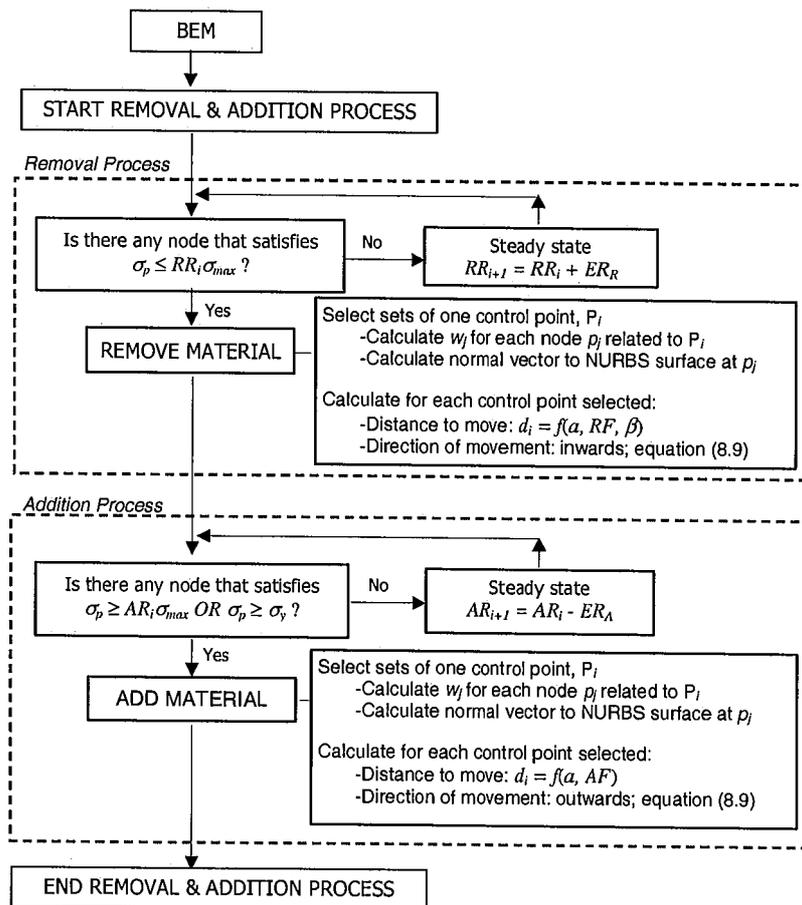
Figure 8.7: Flow chart of the removal and addition process

8.7 Geometry Control

8.7.1 Smoothing Algorithm

It is found that as the optimisation process evolves, edges and corners may become sharper. This problem is undesirable not only because the stresses are likely to increase but also from the manufacturing point of view. To overcome this drawback an algorithm to identify and to smooth sharp areas is implemented.

In the literature, different ways of smoothing geometries can be found. *Laplacian smoothing* is a very popular technique to smooth element meshes on surfaces. This method changes the position of nodes without modifying the topology of the mesh but adjusting the location of each node to the geometric centre of its neighbour nodes. Moreover, smoothing techniques used in a NURBS context (Piegl and Tiller (1997)) could be also considered. However a simple but effective alternative procedure is used here which also avoids the angular geometries.

Firstly, the algorithm identifies any non-smooth edge or corner. To do so, the angles between the geometric lines, which are the trim lines to the surfaces, are computed using the properties of the *dot* and *cross* vector product. Therefore, having two vectors in a surface, say u (u_x, u_y, u_z) and v (v_x, v_y, v_z), and the normal vector to this surface, say vector n, which has an associated specific known direction, the angle between the two vectors u and v is calculated using the *dot* product

$$\cos\theta = \frac{u.v}{\|u\|\|v\|} \tag{8.15}$$

To identify the clockwise angle from vector u to vector v calculated using the *dot* product some extra information needs to be calculated this time using the *cross* product. Since n points perpendicular to u and v, it points either parallel or opposite to the vector *cross* product $u \times v$

$$u \times v = (u_y v_z - u_z v_y)\,\hat{i} + (u_z v_x - u_x v_z)\,\hat{j} + (u_x v_y - u_y v_x)\hat{k} \tag{8.16}$$

203

Assuming the right-hand-rule conventions for the *cross* product, if n points parallel to u x v along the direction of n, the angle θ that has just been calculated above in equation (8.16) is the clockwise angle. Otherwise, if n points opposite to u x v, then the clockwise angle is the $360° - \theta$ angle.

Once all the angles have been calculated, the sharp edges are identified. The criterion for sharp edges is that any edge angle greater than $180°$ or less than $45°$ would be considered as *sharp*. The reason for choosing these limits is because any values out of this range would produce numerical errors in the meshing and Jacobian calculations.

To smooth a corner the control point in corner is moved to the midpoint of the line joining the previous and next control points. Figure 8.8 shows how these sharp edges are smoothed. Each sharp corner has a related control point P. The shape of the corner is approximated using two vectors u and v. The vector u points from P to its previous control point P_1 whereas the vector v points from P to its following control point P_2. The corner is smoothed by moving P to the midpoint of the line that joins P_1 and P_2.

(a) initial model   (b) final model after smoothing

(c) initial model   (d) final model after smoothing
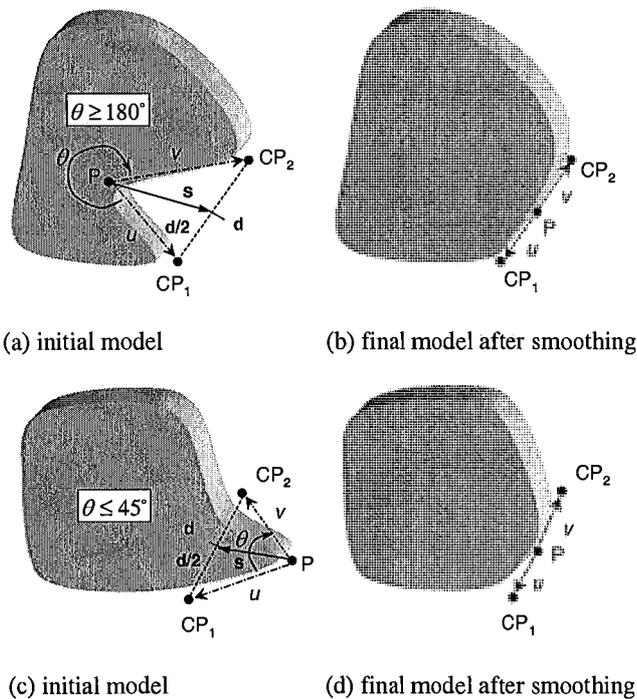
Figure 8.8: Smoothing edges and corners

The main steps of this smoothing algorithm can be summarised in the following flow chart (figure 8.9).
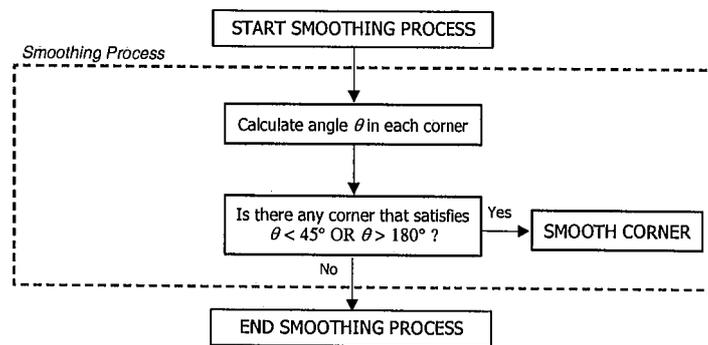
Figure 8.9: Flow chart of smoothing process

8.7.2 Corners Effect

When moving control points situated in corners, it is found important to treat each corner differently according to its angle. A factor γ is introduced to accelerate or decelerate the movement in such cases. The motivation for introducing this new factor is because equal distance gives rise to undesirable distortion that tends to sharpen corners as shown in figure 8.10 (a). Thus, the effect of moving material in corners which are very convex/concave or sharp is larger than moving material in areas on an even surface.

(a)   (b)
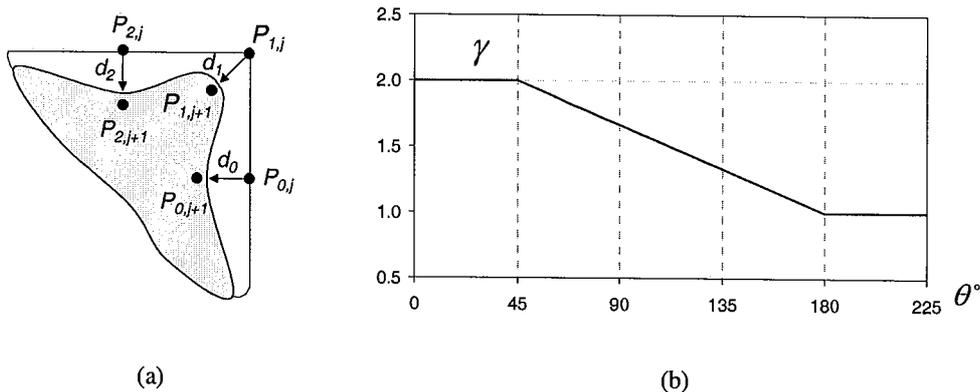
Figure 8.10: (a) Illustration of sharpen a corner. (b) Corner effect factor γ

This factor γ is related to the angle in the corner θ° (see section 8.7.1) thus, the smaller the angle the higher the influence of the factor. Equation (8.10) is modified as follows

$$d_i^* = \gamma \, d_i \qquad\qquad (8.17)$$

The factor γ is calculated following a linear interpolation. Figure 8.10 (b) plots the values of γ according to the angle θ°. Further investigations would consider other types of interpolation such as parabolic. At this stage the flow chart of the removal and addition process can be updated to consider the factor γ as shown in figure 8.11.
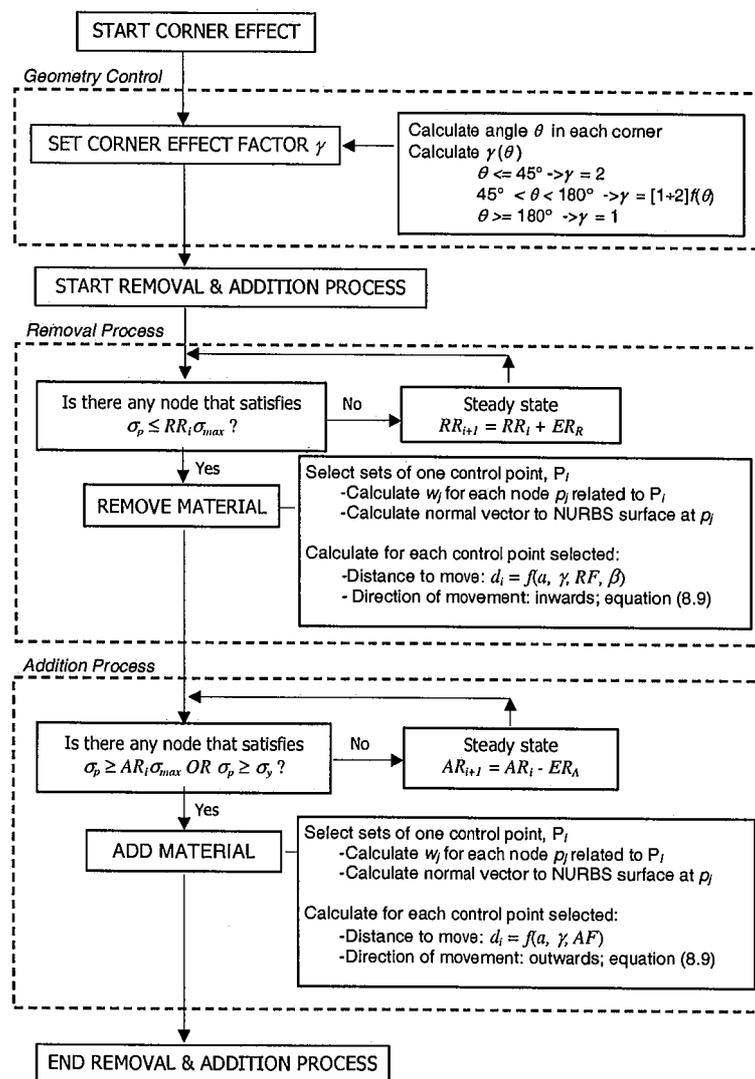
Figure 8.11: Flow chart of γ inserted into the removal and addition process

8.7.3 Mapping the Geometry

In certain problems, the control point movement can be restricted to be performed exclusively in one plane, i.e. movement in XY, or XZ or YZ. This might be desirable, for example, to test a 3D algorithm by reproducing conditions found in 2D modelling, and therefore seems a good starting point for the evaluation and development of the 3D optimiser. For these situations control points on the boundaries of the control point net are moved freely (in the plane of movement) but the points situated inside the net, which would normally move in the perpendicular direction to the surface, cannot move since their movement has been constrained. In this case, like a potential flux problem, the internal control points are moved in a manner influenced by the change of the boundaries. In this work, the relationship between the movement of the outer control points and the inner control points is established by using the concept of shape functions.

Shape functions are used in the finite and boundary element formulations to relate any point in an element to the nodes defining this element as described in the BEM chapter (section 4.8.1). This concept is extrapolated here but applied to the control point nets instead of elements. In order to calculate inner control point locations it is possible to use the shape functions for *serendipity finite elements* (Bathe (1982)). For a surface defined by a given number of boundary control points (*numBCP*), the coordinates at any inner control point can be found from the values of the coordinates at the control points on the boundary of the control point net

$$x = \sum_{i=0}^{numBCP} L_i^S x_i$$

$$y = \sum_{i=0}^{numBCP} L_i^S y_i \qquad (8.18)$$

$$z = \sum_{i=0}^{numBCP} L_i^S z_i$$

where L_i^S terms are the shape functions and (x_i, y_i, z_i) locates the ith boundary control point. Considering $numBCP = 16$ and a local coordinate system as shown in figure 8.12, the shape functions L_i^S for control point P_i are calculated by finding a function

of the local coordinates which gives a value 1 at control point P_i and a value 0 at all other control points on the boundary of the control point net. Figure 8.13 illustrates some of the corresponding shape functions for *numBCP* = 16.
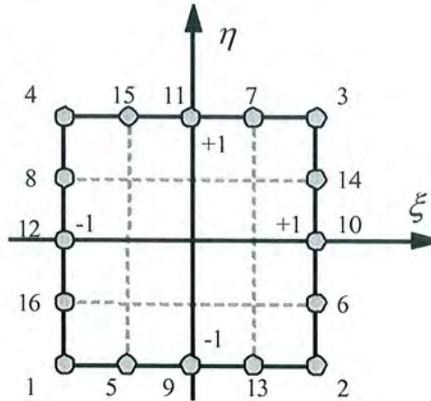
Figure 8.12: Local coordinates used for mapping the geometry

$L_1(\xi,\eta)$

$L_2(\xi,\eta)$

$L_5(\xi,\eta)$

$L_6(\xi,\eta)$

L$_9$(ξ,η)

L$_{10}$(ξ,η)

L$_{13}$(ξ,η)

L$_{14}$(ξ,η)
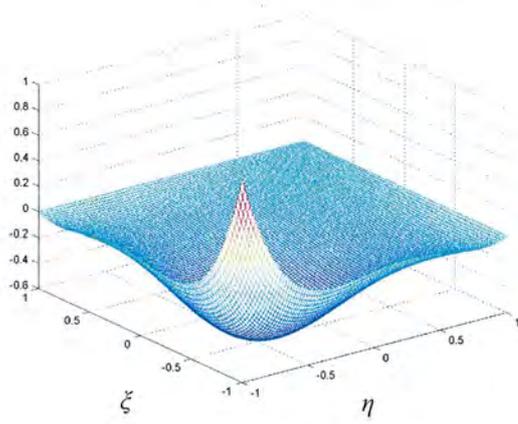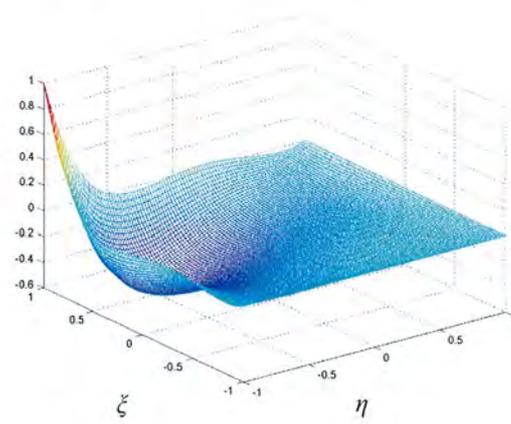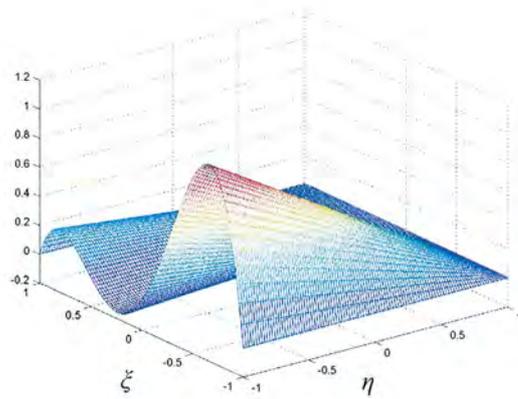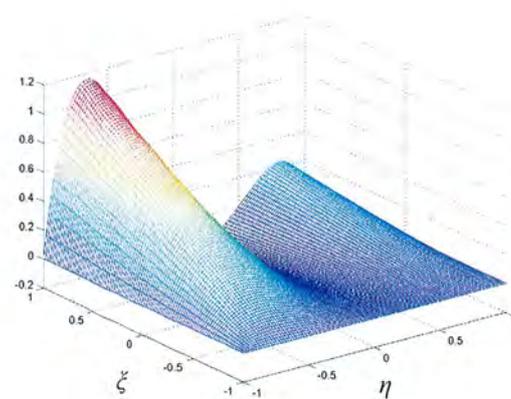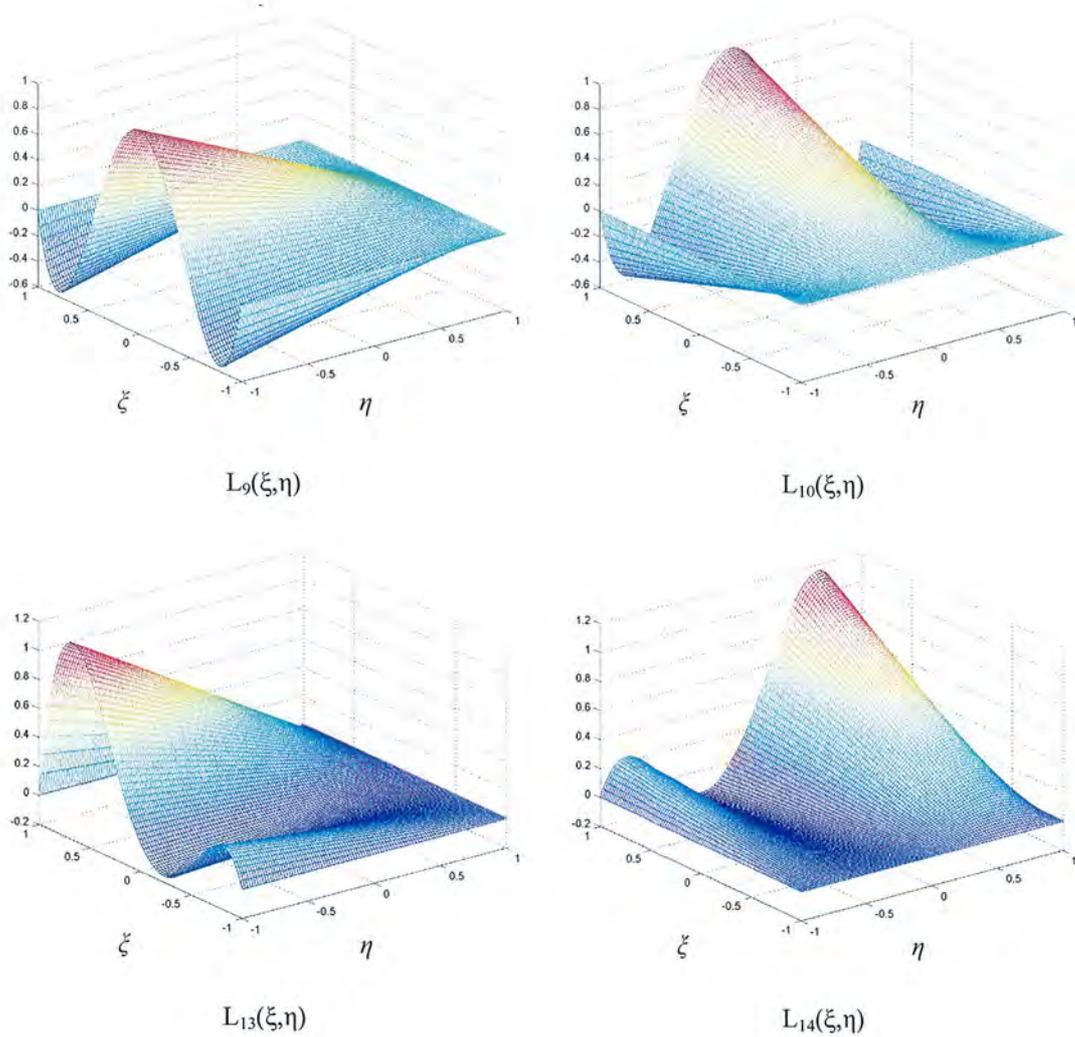
Figure 8.13: Shape functions

When there are no restrictions on the control point movement, then the points situated inside the net can be moved according to equations (8.10) (removal) and (8.11) (addition). However, as the process evolves and more material is removed, it is found important to introduce a similar idea of relationship between the movement of the outer control points and the inner control points.

Nevertheless, the serendipity shape functions (as explained previously) are not applied in the general 3D algorithm since redistribution of the inner control points is not desired. In this case, the current position of the control point is slightly modified by the effect of moving its adjacent control points on the boundaries. Therefore, the global movement can be split into two movements; i.e. the *main* movement and the

propagation movement. The *main* movement follows the general equations (8.10) and (8.11) and the *propagation* movement depends on the movement of the edges. Hence, for a surface defined by a given number of boundary control points (*numBCP*), the coordinates (*x*, *y*, *z*) at any inner control point can be defined as follows

$$x = x_m + k\Delta x_p; \qquad \Delta x_p = \sum_{i=0}^{numBCP} L_i \Delta x_i$$

$$y = y_m + k\Delta y_p; \qquad \Delta y_p = \sum_{i=0}^{numBCP} L_i \Delta y_i \qquad\qquad (8.19)$$

$$z = z_m + k\Delta z_p; \qquad \Delta z_p = \sum_{i=0}^{numBCP} L_i \Delta z_i$$

where x_m, y_m, z_m are the coordinates resulting from the *main* movement; k is a parameter that determines the influence of the *propagation* movement over the global movement; Δx_p, Δy_p, Δz_p are the quantities related to the *propagation* movement; and the $L_i(\xi,\eta)$ terms are the shape functions calculated as follows

$$L_i(\xi,\eta) = \left(\xi_{max} - |\xi - \xi_i|\right)\left(\eta_{max} - |\eta - \eta_i|\right) \qquad\qquad (8.20)$$

It should be remarked that the global movement of the inner control points is essentially determined by the *main* movement. The *propagation* movement effect is smaller and simply avoids overlapping due to the faster movement of the edges.

The mapping geometry algorithm is depicted in the flow chart in figure 8.14. As shown, this algorithm relates directly to the removal and addition process.

8.7.4 Restart Procedures

In 3D, the shape of the component evolves by changing the position of the control points defining the surfaces. Nevertheless, these local changes can result in localised distortion produced by, for example, a number of control points being compressed into a smaller and smaller area. In 2D optimisation problems this phenomenon was overcome by inserting and removing control points (sections 5.7.1 and 5.7.2). In 3D this insertion and removal implies a redefinition of the patches performed in the so-

called *restart procedure*. Therefore, every restart stage implies a reduction or addition, depending on the geometric requirements, of the number of control points.
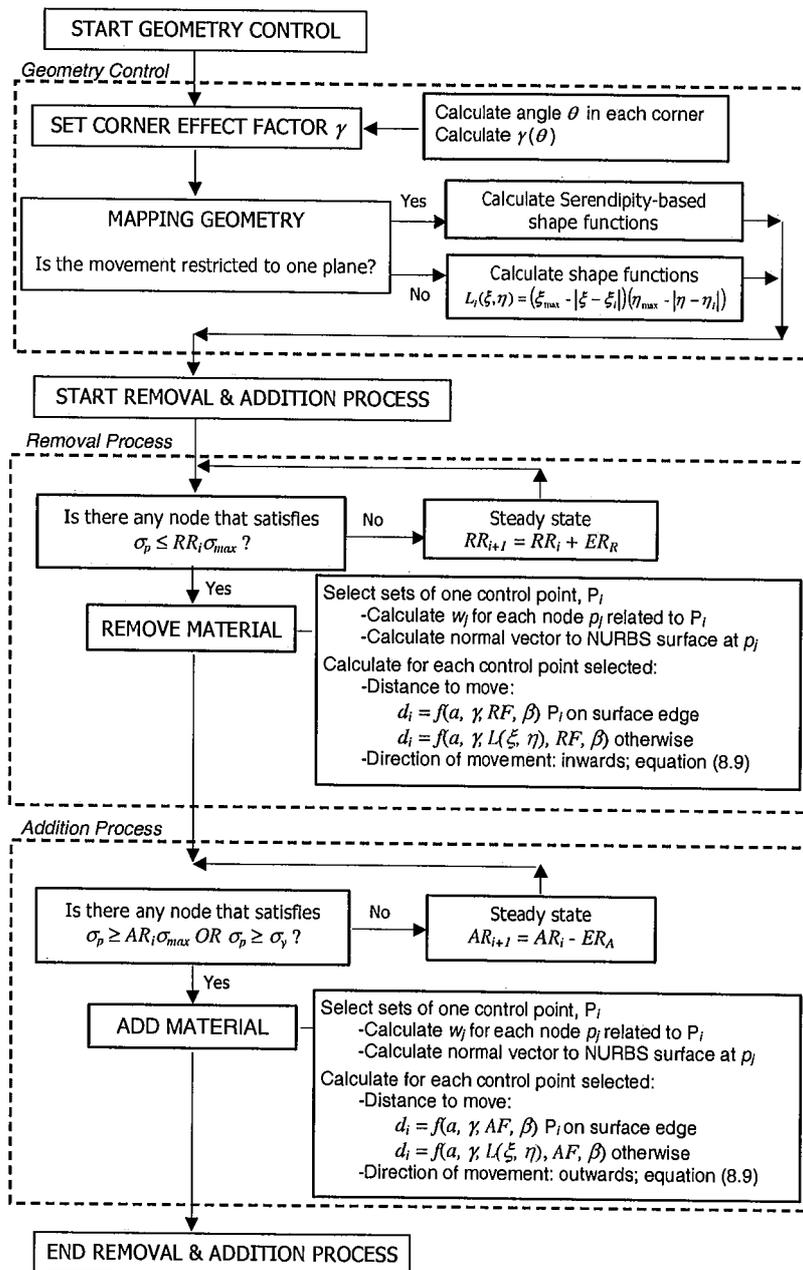
Figure 8.14: Flow chart of mapping geometry inserted into the removal and addition process

Generally, the areas with high curvature would increase their control point definition whereas flat areas of the surface would have retained or even decreased the number

of control points. These areas of high or low curvature are identified using the mathematical definition of the NURBS surfaces which can be processed appropriately in order to obtain the curvature contours, as is shown later in the example 8.9.4.

Another application of the restart procedures is to avoid meshing problems caused by excessive distortion of a surface. In these cases, even a finer mesh would not improve the convergence of the solution. Therefore, a restart is performed and the optimisation process would continue further for this new geometry.

Finally, restart procedures are used to perform topology changes. Similarly to 2D problems, inner low-stressed areas are removed from the structure by creating cavities. Unlike 2D, these areas are not identified using internal points (section 6.5) but the nodes on the surfaces. Hence, inner low stressed areas are identified when low stressed nodes are present in the same region in two opposite close surfaces. Since elastostatic problems are assumed and therefore no body forces are present, the material between the two facing surfaces can be considered to be at the same low stress levels.

The creation of the cavities can be seen as an extrusion cut process where a hole is extruded through the low stressed area. Once the new surfaces are created a BEA is carried out and the optimisation process continues with the new topology. The example 8.9.4, introduced later, clarifies graphically this concept of restarts to perform topology optimisation.

8.8   Stopping Criteria

The objective function adopted in 3D optimisation problems is related to the minimisation of the specific strain energy f_U in the structure

$$f_U = UV \tag{8.21}$$

where V refers to the volume of the structure and U to the strain energy which is defined in BEA notation as

$$U = \int_{\Gamma} \frac{1}{2} \mathbf{T} u \; d\Gamma \tag{8.22}$$

where T are the tractions over the boundary and u the displacements over the part of the boundary Γ.

Accordingly, the stopping criterion ε related to the objective function f_U can be quantified as follows

$$\varepsilon = \left| \frac{f_U^{i+1} - f_U^{i+1}}{f_U^{i+1}} \right| \leq 10^{-4} \tag{6.23}$$

where f_U^i is the value of the objective function at iteration i and f_U^{i+1} is the value of the objective function at iteration $i+1$.

Thus, the basic flow chart of the optimisation algorithm introduced in figure 8.1 is enhanced in figure 8.15, which presents the fully developed flow chart of the algorithm resulting from this research.
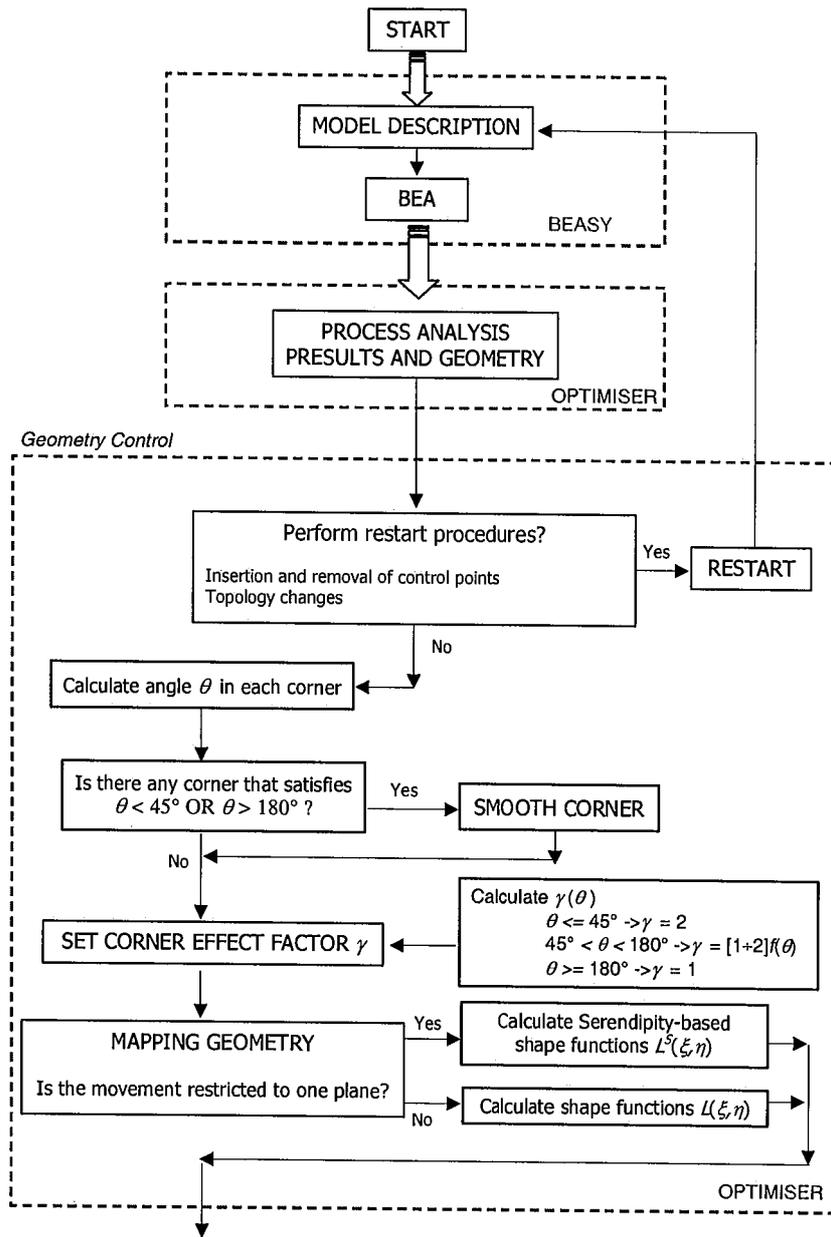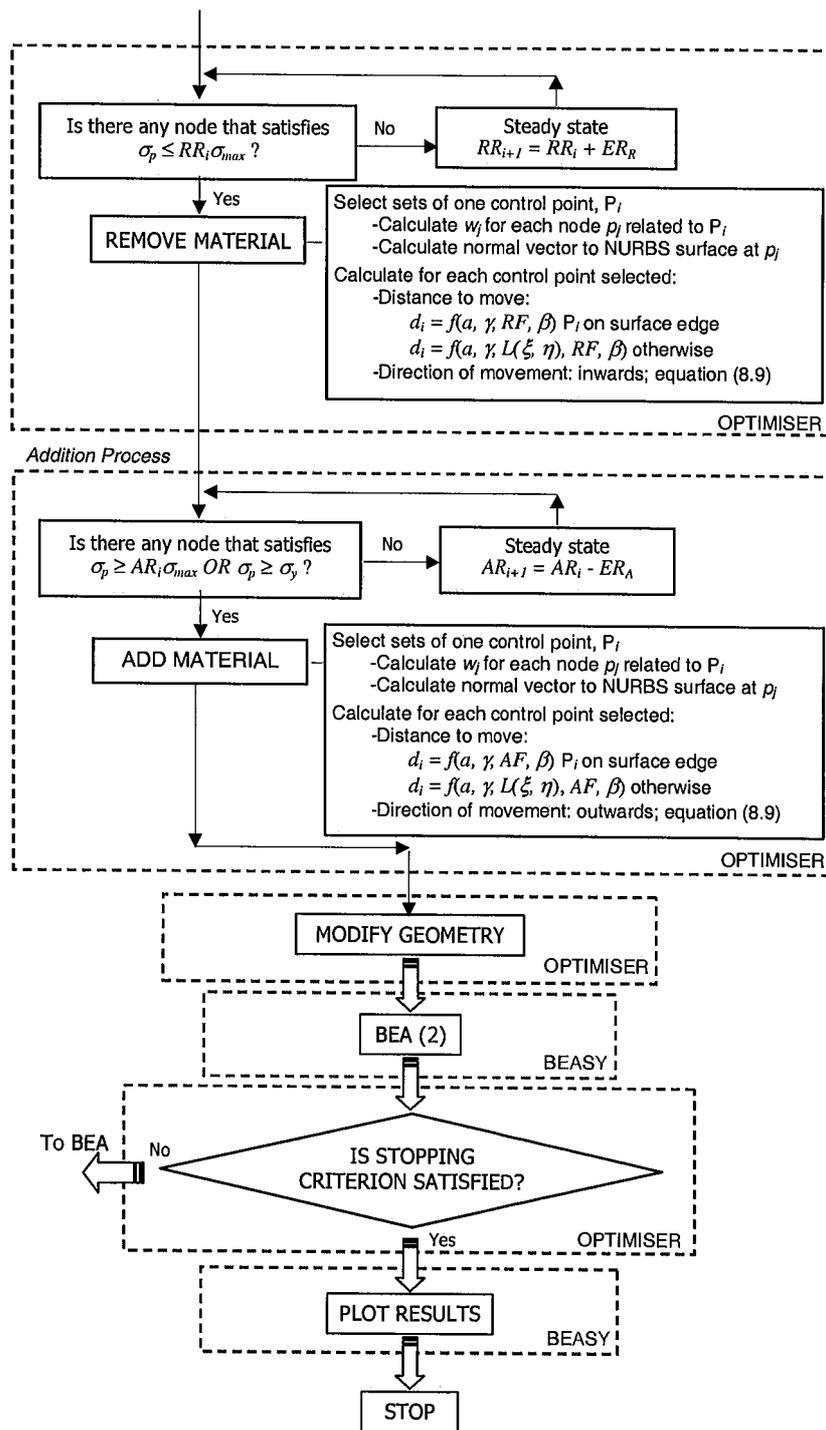
Figure 8.15: Flow chart of the process

8.9 Examples

8.9.1 Beam under Bending I

The following example is presented in order to prove the algorithm. The following isotropic material properties are assumed: Young's modulus E = 210000N/mm² and Poisson's ratio ν = 0.3. This example is the optimisation of a rectangular beam fixed at surface S1 and subjected to a load in the x-direction of magnitude 100 N at surface S2, as shown in figure 8.16. The dimensions of the beam are 100 mm x 50 mm x 20 mm. The objective of this optimisation problem is to minimise the strain energy U (equation (8.22)).
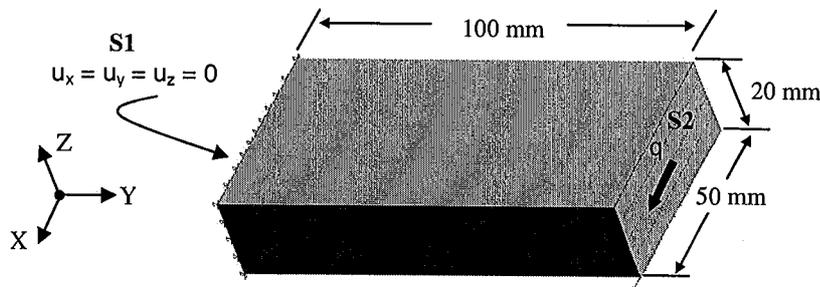
Figure 8.16: Problem definition for a beam under bending

The optimisation parameters are set to $RF = 10$, $AF = 10$, $AR = 0.90$, $ER_A = 0$, $ER_R = 0.05$, $RR_0 = 0.05$. RR increases from 0.05 to 0.10. In the case of removing material, the parameter β (equation (8.13)) is also considered.
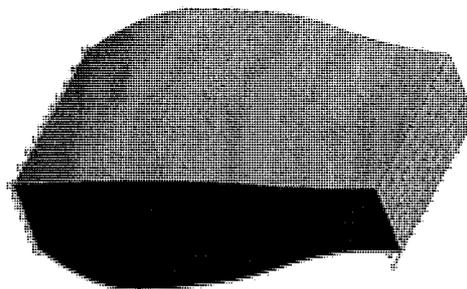
Figure 8.17: Optimum design

216

Figure 8.17 shows the optimum design for the objective of this problem. The initial boundary element model contains 160 elements but at the end of the optimisation process this number increases to 195.

Figure 8.18 displays the evolution of the process according to the von Mises stress contour plot. Comparing the initial topology and the final one (after 9 iterations); in the final design there is approximately a 50% increase from the initial volume.

| iteration 0 | iteration 2 | iteration 4 |

| iteration 6 | iteration 8 | iteration 9 |
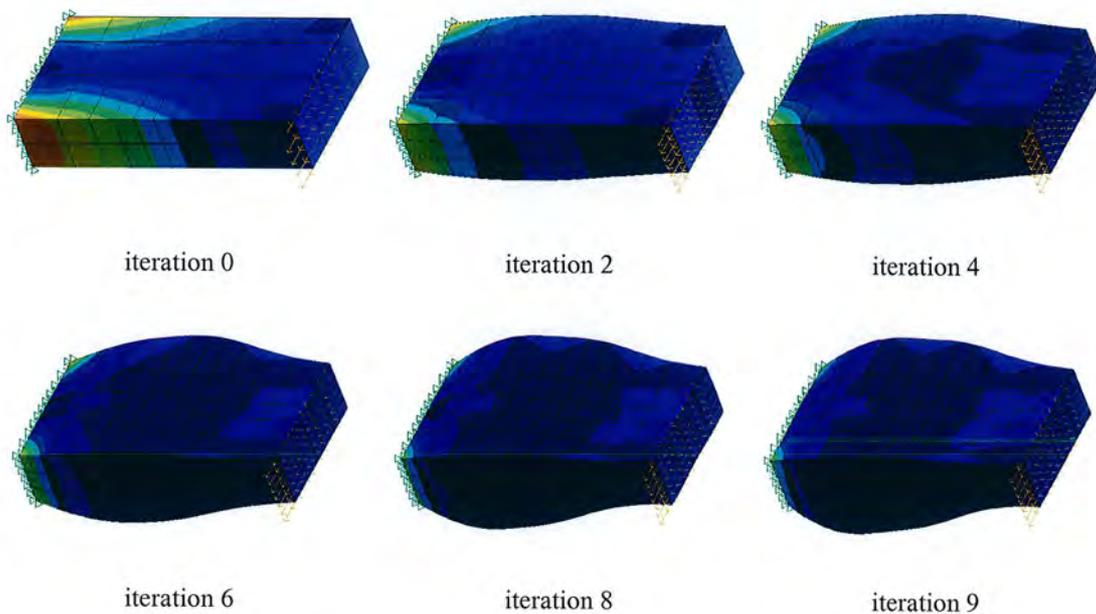
Figure 8.18: Von Mises stress contour plot

Figure 8.19 displays the evolution of the process according to the objective function. The objective of minimising the strain energy is accomplished after 9 iterations when there are no further changes in the objective function. As expected, the stiffest design is obtained by a redistribution, and mainly addition, of material. Although in most engineering problems the objective is to reduce material, this example is chosen to show and test the effectiveness of the algorithm in any situation, addition or removal.

Notice that this optimum design is accomplished for the load and displacement restrictions set. A different load or restraints situation would result in a different optimum design.
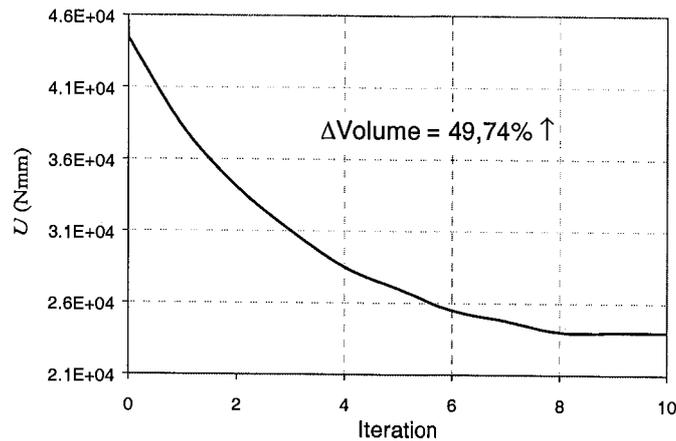
Figure 8.19: Evolution history of the objective function

8.9.2 Beam under Vertical Load I

The following example shows the optimisation of a beam whose movement is restricted to the plane ZY. The following isotropic material properties are assumed: Young's modulus E = 210000N/mm^2, Poisson's ratio ν = 0 (fictitious ν to avoid undesirable stress concentrations at the fixed supports). Figure 8.20 depicts the problem definition, the beam of dimensions 100 mm x 50 mm x 20 mm is fixed at the top and bottom surfaces (S1, S2) on the plane Y = 0. A vertical load of 100 N is applied on the surface (S3) situated in the middle of the plane Y = 50 mm. The optimisation aims to minimise $f = UV$ (equation (8.21)).

From the type of elements available in the BEASY library, Q3_8 are used. These elements are quadratic and interpolate the solution using eight nodes. The initial element mesh depicted in figure 8.21 (a) consists of 720 elements and 3300 nodes. The automatic meshing parameters remain constant for the whole process. These parameters, which describe the minimum and maximum element size, are set to 4 mm for the minimum size and 5 mm for the maximum, respectively to give a reasonably uniform mesh. The final design (figure 8.21 (b)) has 718 elements and 3286 nodes.
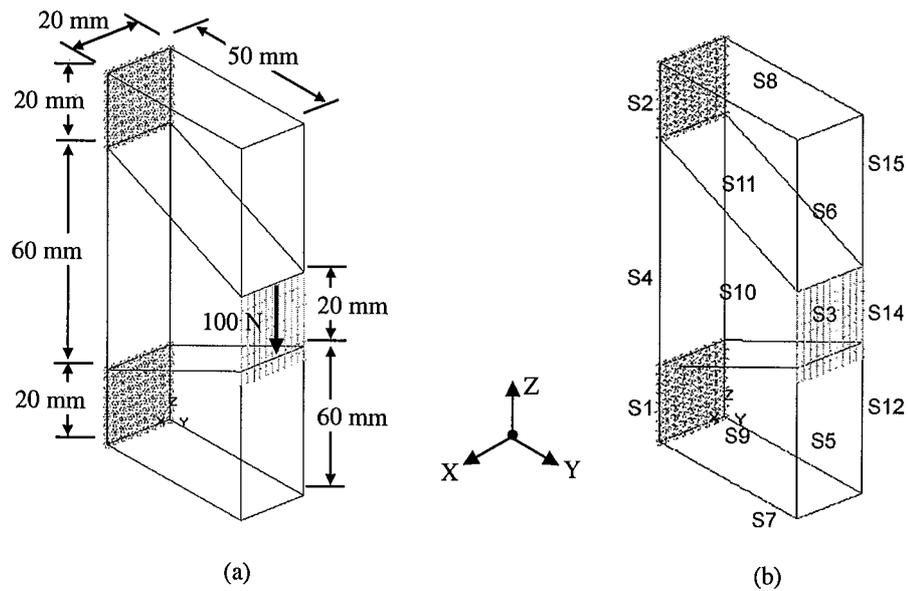
Figure 8.20: Beam problem. (a) Problem definition. (b) Surfaces

Isometric View   YZ view   Isometric View   YZ view
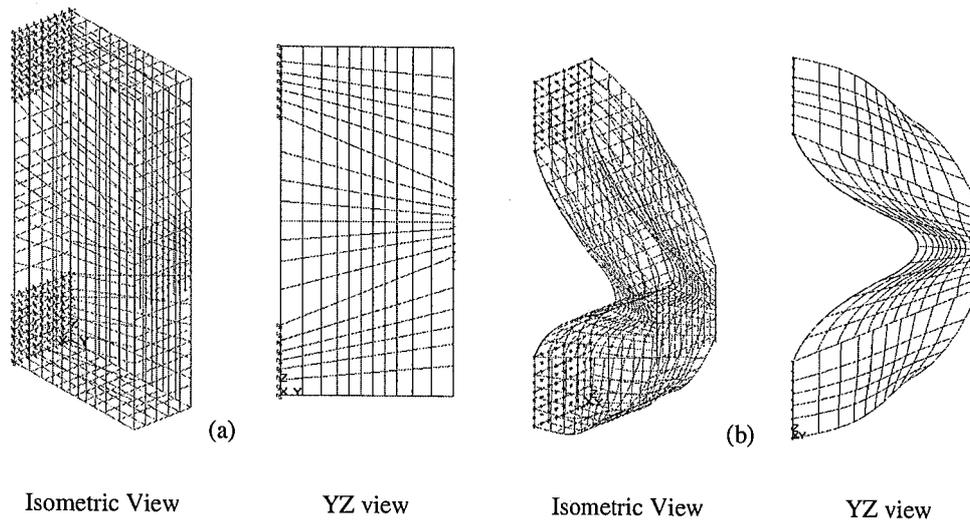
Figure 8.21: Boundary element mesh. (a) Initial design. (b) Final design

The position of the control points can vary during the process but in this example the number of control points does not change. The initial distribution of control points viewed from plane YZ and X = 0 is shown in figure 8.22 (points in blue). This picture displays also the control points at iteration 1 (points in red). From the current YZ view, there are 65 control points, this distribution is repeated along the parallel plane (X = 20). Moreover, at X = 10, a group of control points is analogously

distributed to the control points on the edges of the control point net in figure 8.22, therefore the total number of control points defining the surfaces is 162.
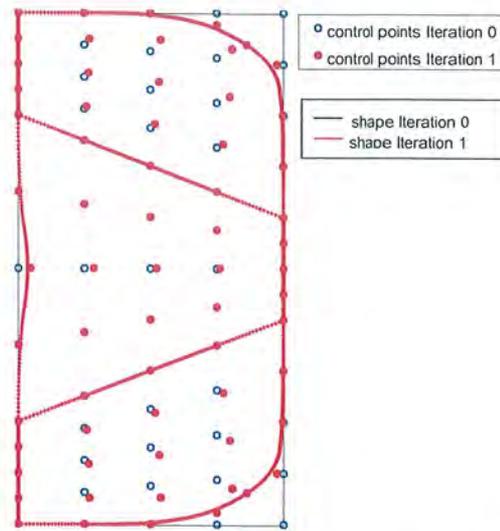
Figure 8.22: Example of mapping control points at iteration 1

Control points situated in the inside of the control point net are moved according to the movement of the control points on the boundaries. In other words, for each surface, the inner control points are mapped to the outer control points. The shape functions used in the mapping are those shown in figure 8.13. Figure 8.22 shows the distribution of the control points at iteration 1 after the mapping. The position of the control points can be compared to the previous iteration (iteration 0). According to these results, points near the corners are moved much less than points in other areas and moreover, the movement inwards would be expected for control points near the corners. The explanation for this apparently contrary behaviour is shown in the display of the shape functions (figure 8.13), which change sharply from 1 to negative values in the vicinity of the source control point.

The von Mises stress contour plots are displayed in figures 8.23 (Isometric view) and 8.24 (Front view). These figures show the stress situation every 10 iterations until iteration 43. For this final design the volume ratio is $V/V_0 = 56\%$. According to the von Mises stress contours it could be argued that the final design presented is not a fully stressed design (as expected). The reason is that, due to current limitations dealing with the code and analysis data, much manual effort is involved in order to

progress along the iterations. Nevertheless, this solution gives a clear understanding about the structure evolving towards the optimum.

iteration 0   iteration 10   iteration 20

iteration 30   iteration 40   iteration 43
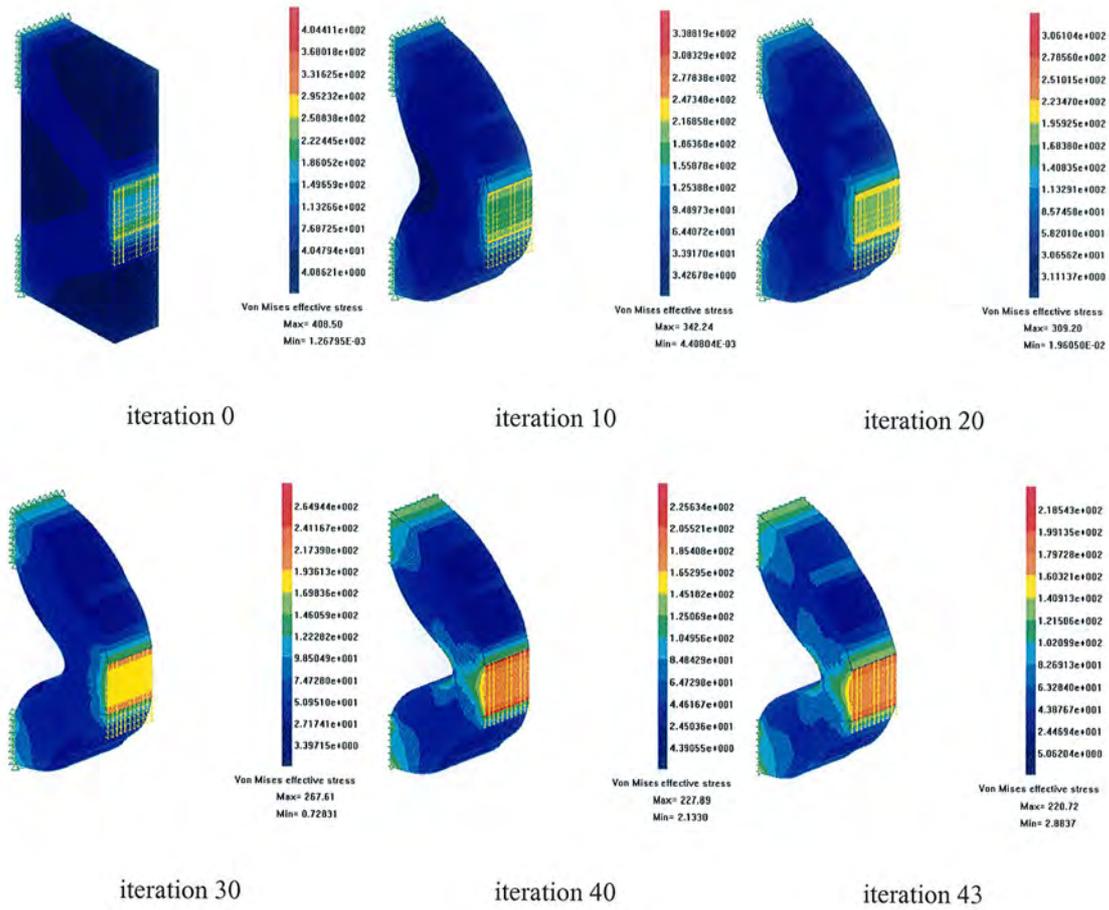
Figure 8.23: Von Mises stress contour plots. Isometric View

iteration 0   iteration 10   iteration 20

| iteration 30 | iteration 40 | iteration 43 |
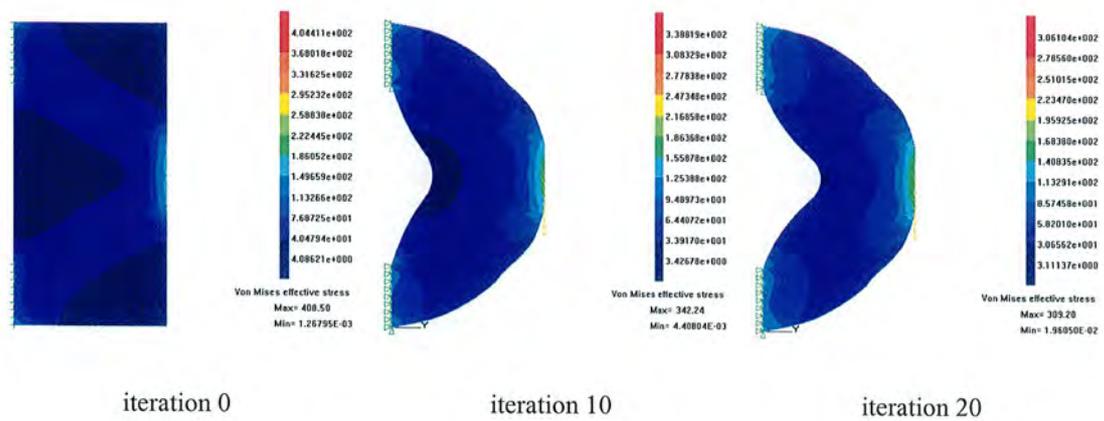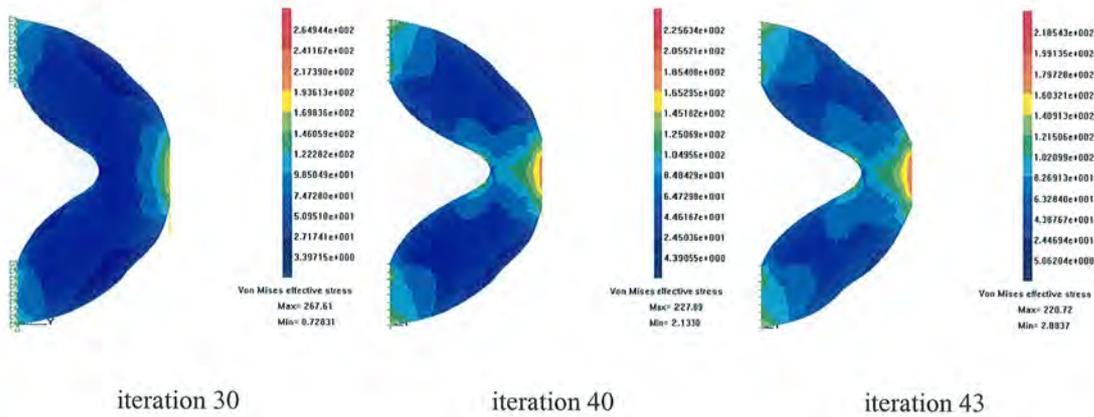
Figure 8.24: Von Mises stress contour plots. Lateral View (XY)

8.9.3 Beam under Vertical Load II

In the present example the beam under vertical load is optimised to minimise f_U (equation (8.21)). In this case, unlike example 8.9.2, there are no restrictions on the 3D movement. The following isotropic material properties are assumed: E = 210000N/mm² and ν = 0. Figure 8.25 depicts the beam (100 mm x 50 mm x 20 mm) fixed at the top and bottom surfaces (S1, S2) on the plane Y = 0. A vertical load of 100 N is applied on the surface (S3) situated in the middle of the plane Y = 50.

Initially 319 quadratic elements type Q3_8 are used. The initial element mesh depicted in figure 8.27 (a) consists of 319 elements and 1560 nodes. The automesh parameters are set to 6 mm for the minimum size and 8 mm for the maximum, respectively.

The objective function evolution is plotted in figure 8.26. After 6 iterations the geometry is redefined in the so-called *restart step* (see figure 8.26 at iteration 5). The new geometry at this stage, shown in figure 8.27 (b), is discretised with 314 elements and 1590 nodes. As the process evolves care must be taken to ensure convergence of the results. At iteration 17 since the error estimators indicated possible inaccuracy (see figure 8.26) the convergence is checked by refining the mesh (automesh = 4, 5). The change between the results is greater than 1%-5% showing evidence to suggest that the coarse model has not converged. Therefore, at this stage the automesh parameters are changed from 6, 8 to 4, 5. The new model has now 716 elements and

3354 nodes (see figure 8.27 (c)). A new restart step is carried out for reasons of geometric distortion at iteration 21, defined as restart2 and geometrically depicted in figure 8.27 (d). The statistics in this case are 638 elements and 2988 nodes. The process evolves showing the minimum at iteration 44 using a model of 677 elements and 3116 nodes; as shown in figure 8.26 further iterations do not improve the objective function.

(a)   (b)
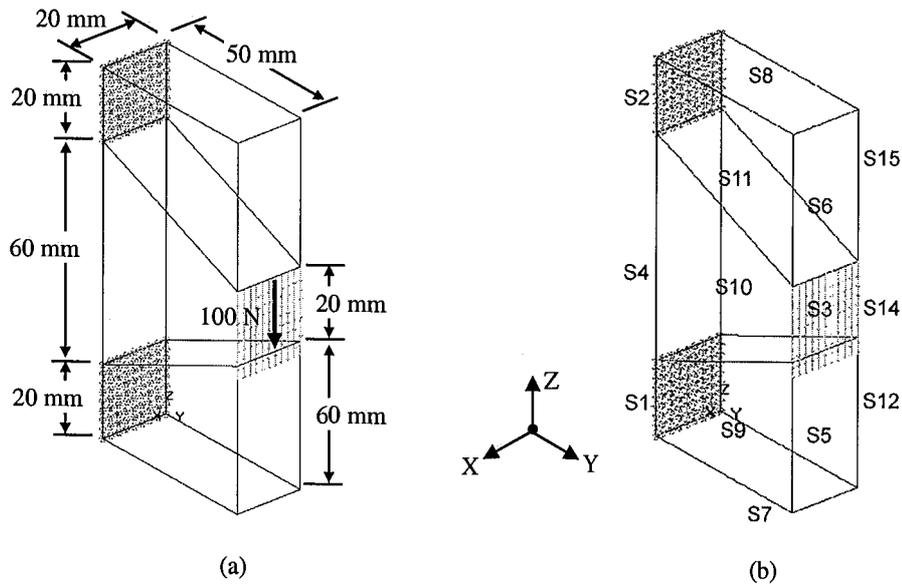
Figure 8.25: Beam problem. (a) Problem definition. (b) Surface definition
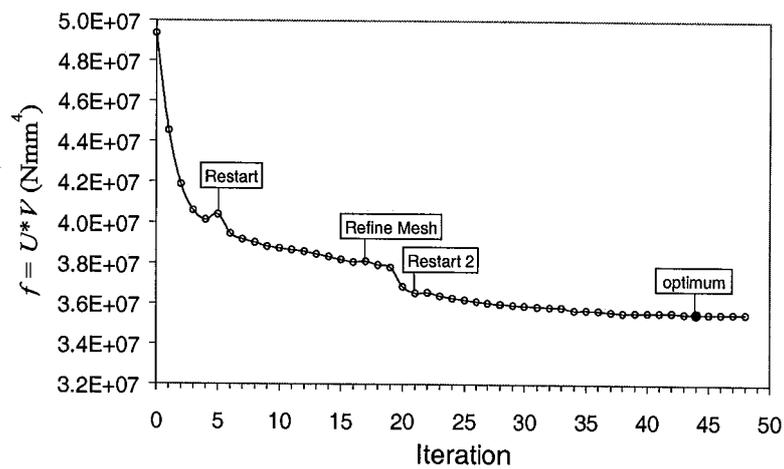
Figure 8.26: Objective function evolution

(a)　　　　　　(b)　　　　　　(c)　　　　　　(d)　　　　　　(e)
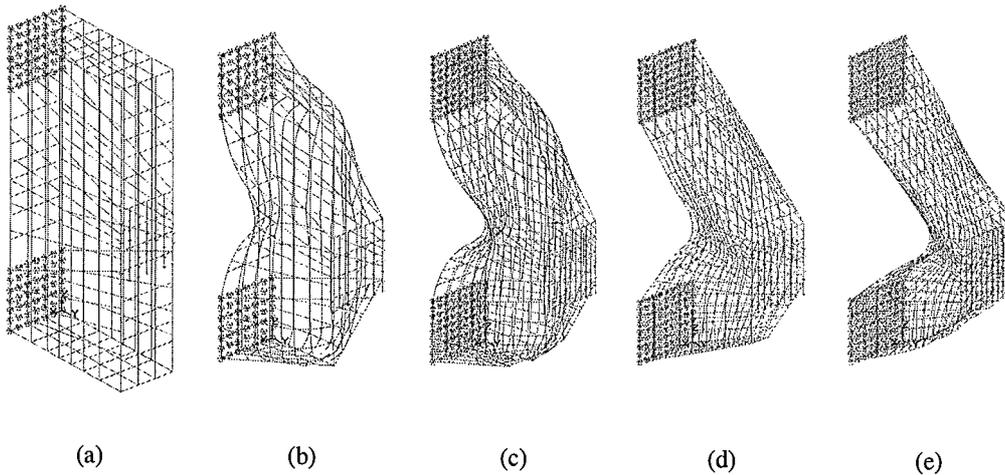
Figure 8.27: Boundary element mesh. (a) initial design, (b) restart, (c) finer mesh,

(d) restart2, (e) optimum design

There is a total number of 162 control points defining the surfaces. These control points, initially equally distributed, are moved according to the algorithm, to achieve a more efficient profile. Control points situated inside of the control point net are influenced by the movement of the control points on the boundaries. Hence, their movement follows equation (8.19) using the parameter $k = 0.5$. The number of control points is not automatically changed by the algorithm but manually in the restart procedures. Therefore, every restart implies a reduction or addition, depending on the geometric requirements, of control points. Areas with high curvature either increase or maintain their control point definition whereas large flat regions would have retained or even decreased the number of control points.

The RR is set up initially to 0.05 and remains constant until the first mesh refinement at iteration 17. After that, RR increases steadily to reach the value of 0.35 when the optimum is reached (iteration 44). To control the convergence the factor RF is also considered. Initially, this factor is 1.0. Nevertheless, throughout the process the von Mises stress levels in the overall structure increase showing that the structure is evolving towards a fully stressed design optimum. Therefore, the process is slowed down according to the stress levels by setting $RF = 0.5$ from iteration 35 when fine tuning is needed near the optimum.
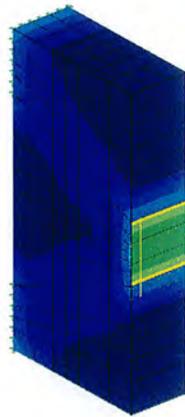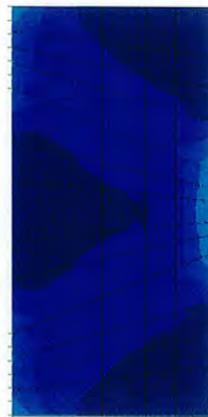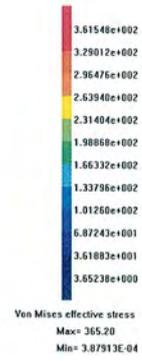
(a)   (b)   iteration 0
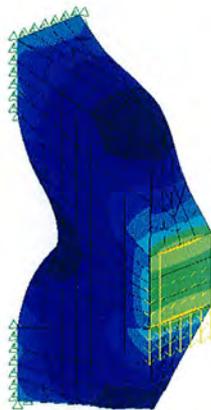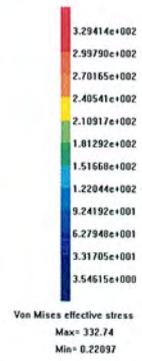
(a)   (b)   iteration 5 (restart)
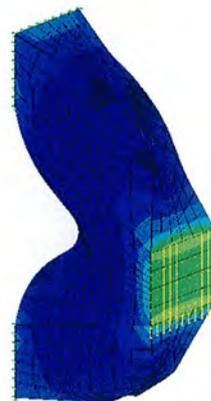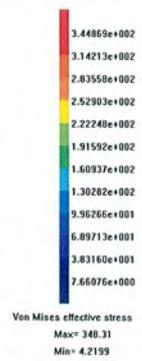
(a)   (b)   iteration 17 (refine mesh)

(a)   (b)   iteration 24 (restart 2)

(a)   Front view   (b)   iteration 44 (optimum)
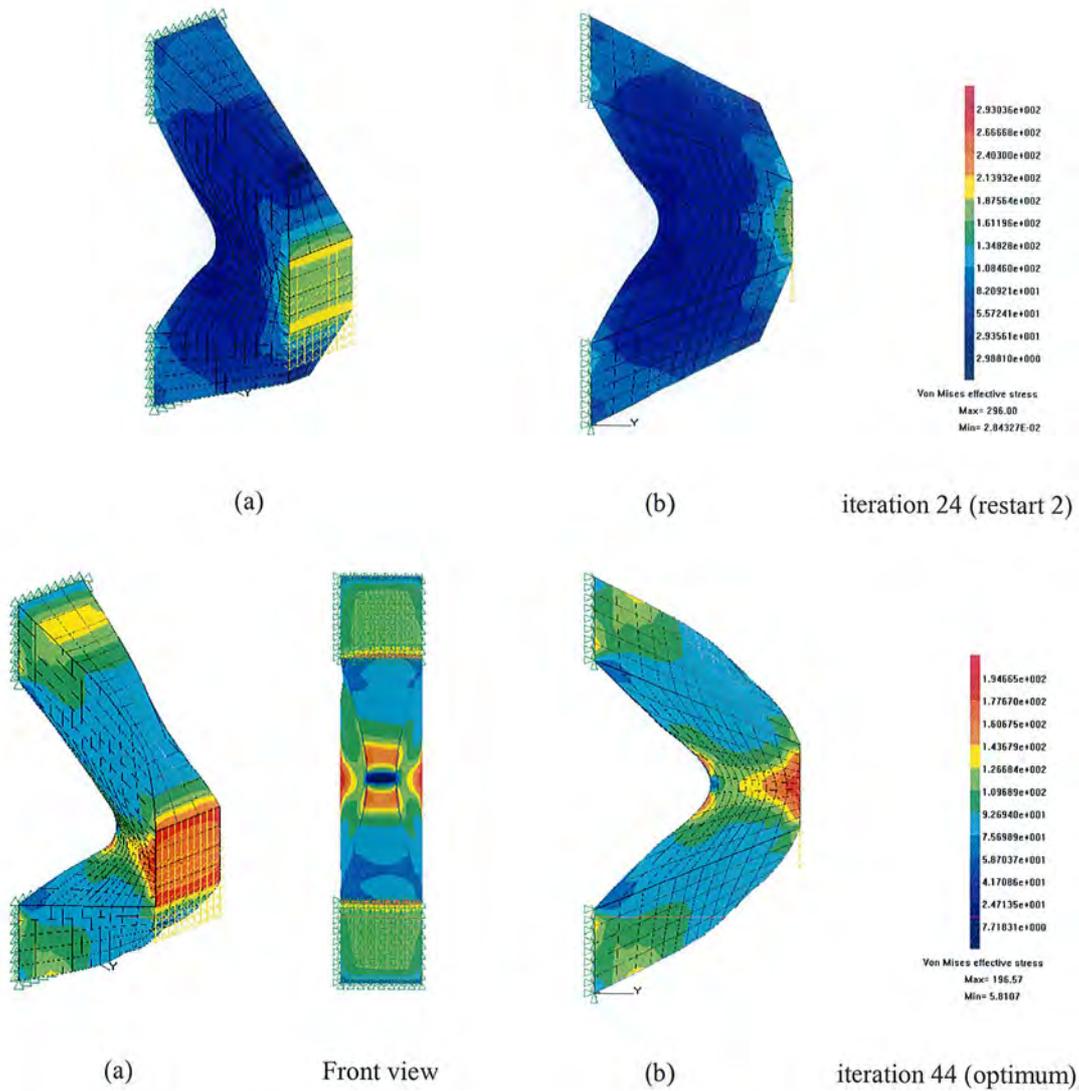
Figure 8.28: Von Mises stress contour plots. (a) Isometric View, (b) Lateral View (XY)

Figure 8.28 depicts the von Mises stress contour plots at different stages of the process. The optimum design is obtained after 45 iterations for a volume ratio $V/V_0 = 0.408$, i.e. $\approx 60\%$ reduction from the initial design. At iteration 44 the value of the von Mises stress for most of the material is between 40% and 50% of the maximum von Mises stress within the structure.

8.9.4 Beam under Bending II

A similar problem to the example in section 8.9.1 is considered changing the patch definition for the same isotropic material properties (Young's modulus $E = 210000N/mm^2$ and Poisson's ratio $v = 0.3$). The dimensions of the beam in this case are 100 mm x 50 mm x 30 mm. The problem is defined in figure 8.29 (a), the initial geometry is divided into 10 surfaces. Surface S1 is fixed ($u_x = u_y = u_z = 0$) and surface S2 has a horizontal load applied of 1000 N. These surfaces are defined as non-design domain, i.e. are fixed throughout the process. The rest of the surfaces are considered design domain and therefore free to change. The optimisation follows a stress driven criterion which aims to minimise the function $f = UV$ (equation (8.21)).
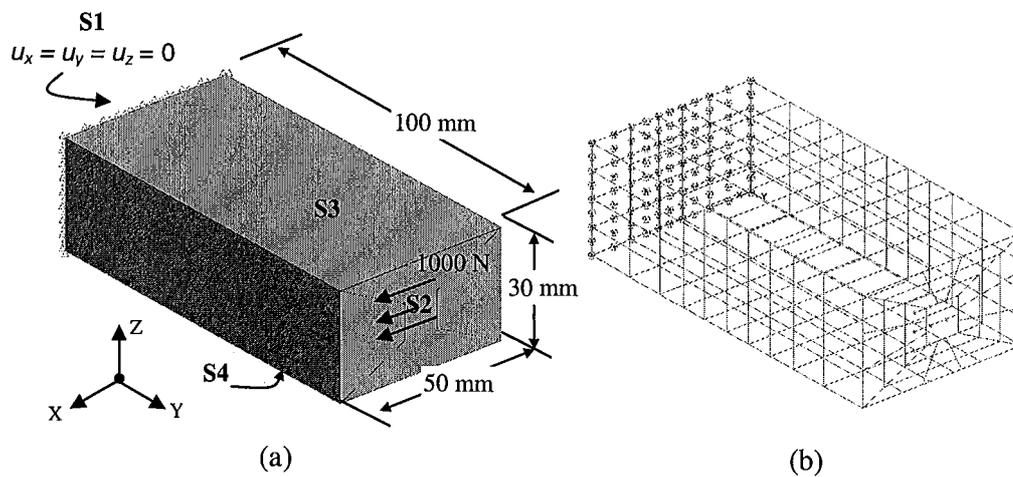
Figure 8.29: (a) Problem definition. (b) Initial boundary element mesh

The initial number of quadratic boundary elements is 237 (see figure 8.29 (b))). At each iteration, a new remeshing is done and the elements are distributed according to the automesh parameters initially set. However, as the optimisation process evolves the surfaces change and therefore, refinements of the mesh are needed to improve the accuracy of the analysis results. By modifying the automesh parameters appropriately a finer mesh is set from iteration 27 (see figure 8.30)). The new boundary element mesh consists of 504 elements.
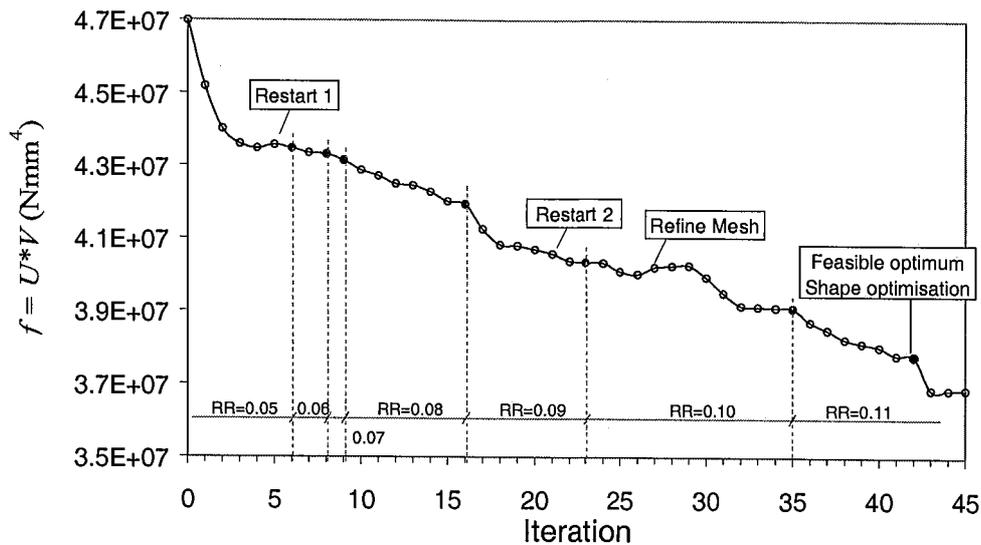
Figure 8.30: Evolution of the objective function

Figure 8.30 shows the evolution of the objective function. Initially a *RR* of 0.05 is set; this ratio is increased gradually by the ER_R (ER_R = 0.01) when a *steady state* is reached (equation 8.3). In figure 8.30 every *RR* stage is shown delimited by dotted lines. At the end of the process *RR* has increased to 0.11. The *AR* is set constant (ER_A = 0) and equal to 0.90.

From figure 8.30 it can be seen that restarts are performed at iteration 5 (Restart 1) and iteration 21 (Restart 2). Using the restart feature, the surfaces of the model are redefined to avoid meshing problems caused by excessive distortion of a surface. The optimisation process would continue further for this new geometry. Figure 8.31 shows a wireframe view of the geometry for the different critical steps mentioned in figure 8.30. The changes introduced at Restart 1 are shown between figures 8.31 (b) and 8.31 (c). The number of surfaces is maintained but their definition is slightly modified due to some very distorted elements. This element problem cannot be improved even by refining the mesh and therefore, only by doing a restart can we ensure the reliability of the analysis results. Similarly, at iteration 20 a new restart is performed but this time the number of surfaces is decreased (See fig. 8.31 (d) and 8.31 (e)). Surfaces which show almost a C^1 continuity are merged and a new surface is defined by the control points of the two former surfaces.

228

(a) iteration 0   (b) iteration 4   (c) iteration 5 (Restart)

(d) iteration 20   (e) iteration 21 (Restart2)   (f) iteration 42 (final design)
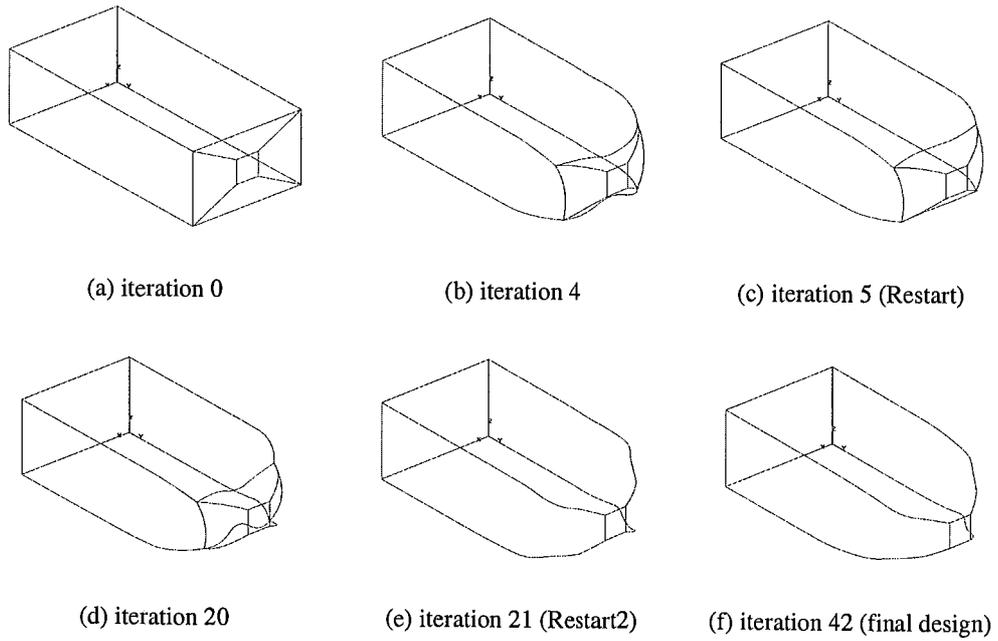
Figure 8.31: Critical steps of the optimisation process

(a)   (b)

(c)
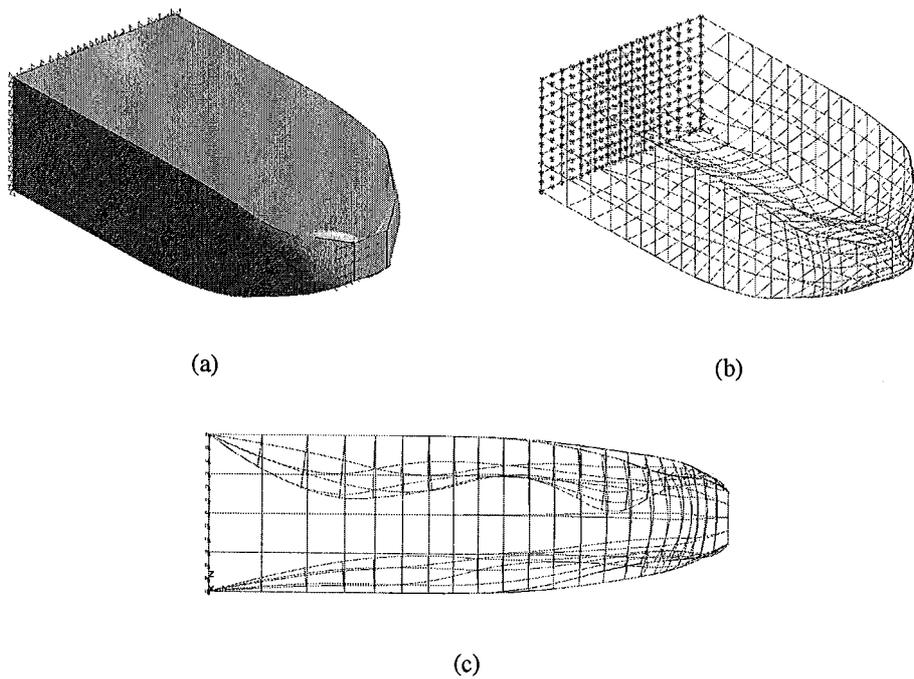
Figure 8.32: Optimum design

Following the evolution of the objective function, the feasible minimum (for shape optimisation) is found at iteration 42, since further iterations result in a very distorted

shape. Figure 8.32 displays the optimum feasible design obtained at iteration 42. The display of the elements makes it evident (unlike the wireframe view in figure 8.31) that the section is undergoing a significant amount of thinning in two locations, as might be expected from the 2D study of the related plane stress case. This is the optimum for shape optimisation; topology optimisation would be accomplished doing a new redefinition of the surfaces. Looking at the lateral view in figure 8.32 (c) there are some areas from the bottom and top surfaces that become closer, therefore surfaces between these areas would be potential new cavities.

According to the von Mises stress situation, only removal of material is performed. No addition of material is carried out since the highest von Mises stresses appear in the non-design areas. At the end of the process there has been a 38% reduction from the initial volume (V_0 = 150000 mm^3). Figure 8.33 shows the von Mises stress contour plots at different iterations of the process.

iteration 0

iteration 5

iteration 10

iteration 15

iteration 20

iteration 25

iteration 30

iteration 35

iteration 40
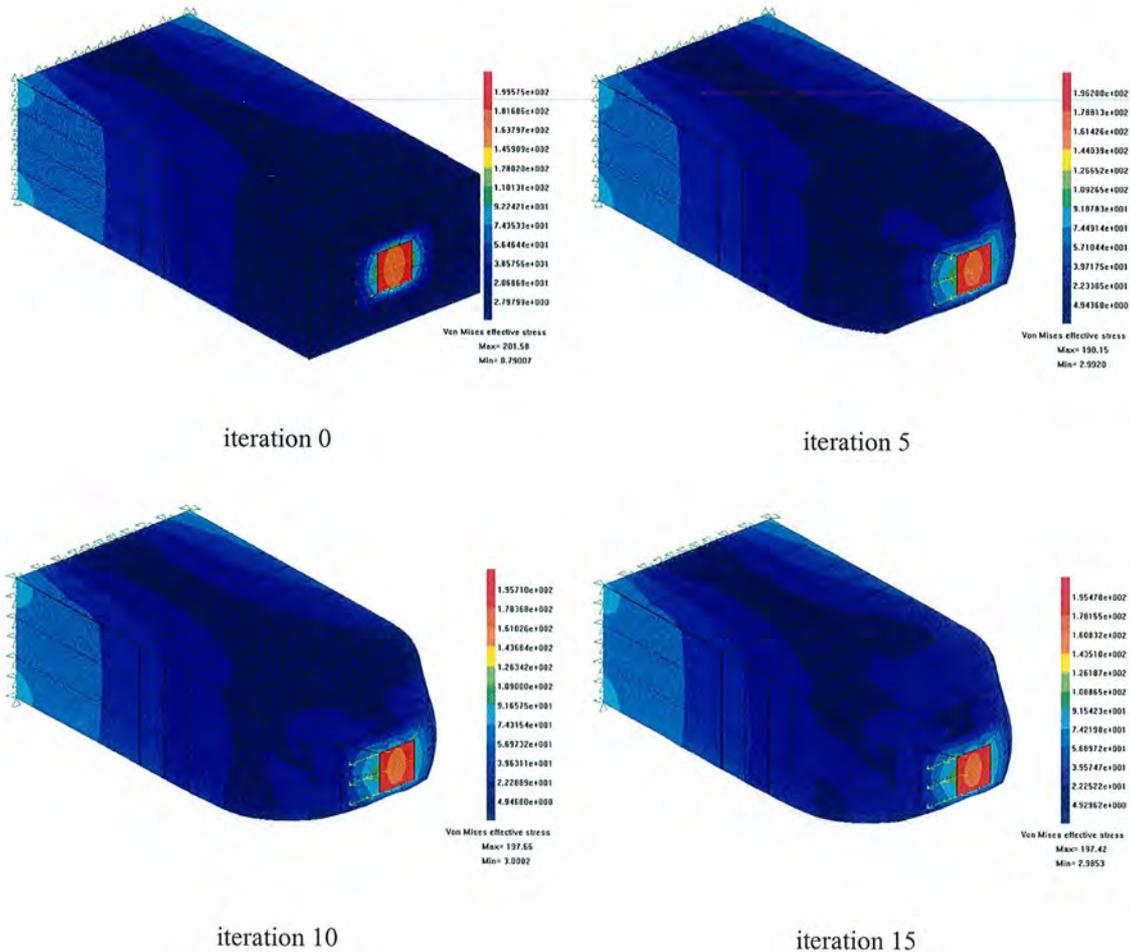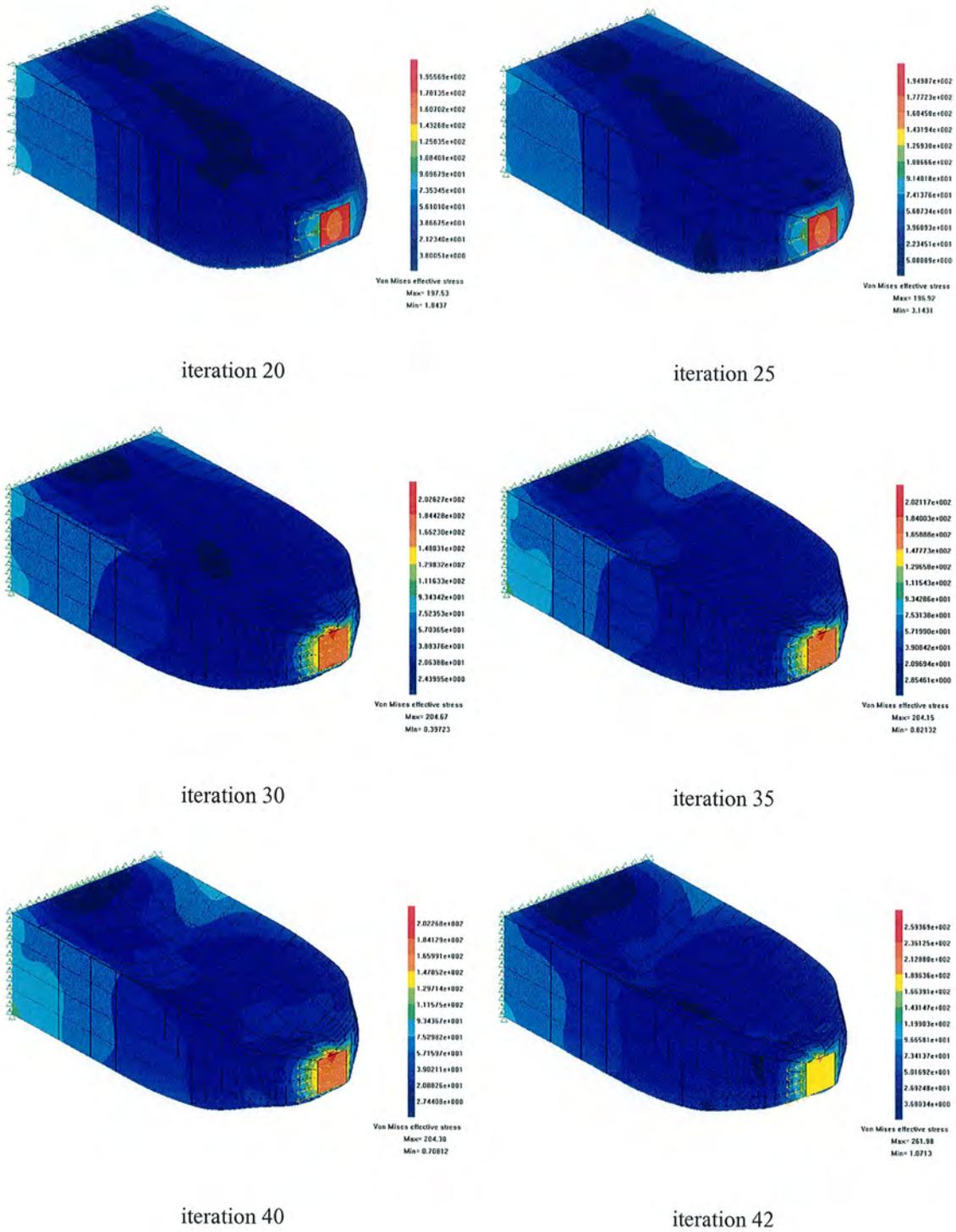
iteration 42

Figure 8.33: Von Mises stress contour plots every 5 iterations

8.9.4.1 Topology Optimisation Step

The topology changes are performed by redefining the shapes as appropriate and therefore using the restart scheme. From figure 8.32 (b) the potential areas to be redefined are identified. Surfaces S3 and S4 present different areas of curvature and moreover, in the areas of high curvature these surfaces get very close.
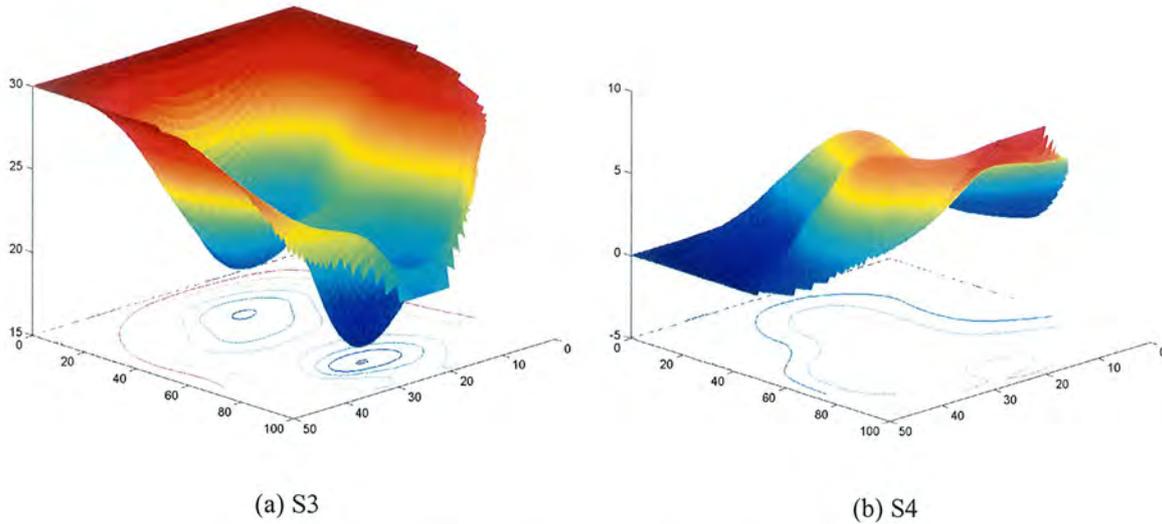
(a) S3   (b) S4

Figure 8.34: Shape contours. (a) Surface S3. (b) Surface S4

(a)   (b)
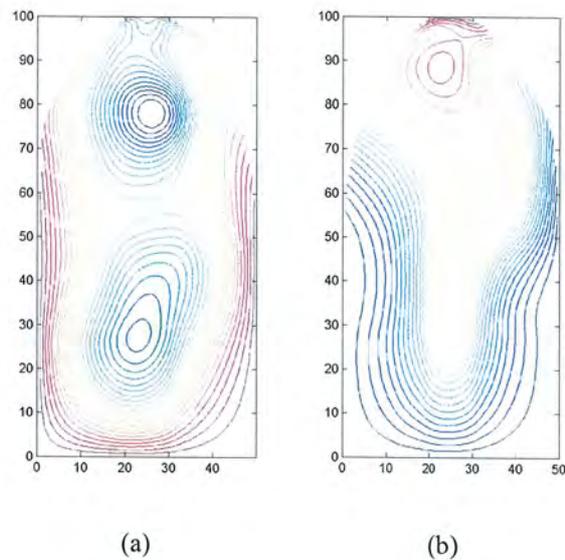
Figure 8.35: Shape contours. (a) S3. (b) S4

In order to compute and therefore identify in more detail these areas, the curvature contours of both surfaces S3 and S4 are displayed in figures 8.34 and 8.35. From figure 8.34 the two peaks can be clearly observed in both surfaces. Figure 8.34 (a) represents the surface S3 whereas figure 8.34 (b) refers to S4. It happens that for both surfaces the convex/concave areas occur at similar coordinates (x, y) as depicted in the latter figure 8.35. Thus, joining the iso-lines of maximum curvature in figure 8.35 two cylindrical cavities are created in the solid. It should be noted that the asymmetric solution of this symmetric problem is due to the automesh, since it produces non symmetric meshes which later influence on the movement of the control points.

Figure 8.36 shows the evolution of the whole process including topology optimisation. Topology changes are performed manually at Restart 3 (iteration 43), Restart 4 (iteration 56) and Restart 5 (iteration 59). These restart procedures create new cavities in the structure whenever the algorithm shows a inner low stressed area in facing surfaces at a small distance. Inner low stressed areas must satisfy equation (8.1). The minimum distance d_{min} is taken as a percentage of the minimum dimension of the problem, in this case $d_{min} = 10$ mm; $\approx 33\%$ of 30 mm. It is observed in figure 8.36 that the effect of performing a topology change considerably increments the objective function. As expected, the insertion of new cavities significantly increases the strain energy of the structure.
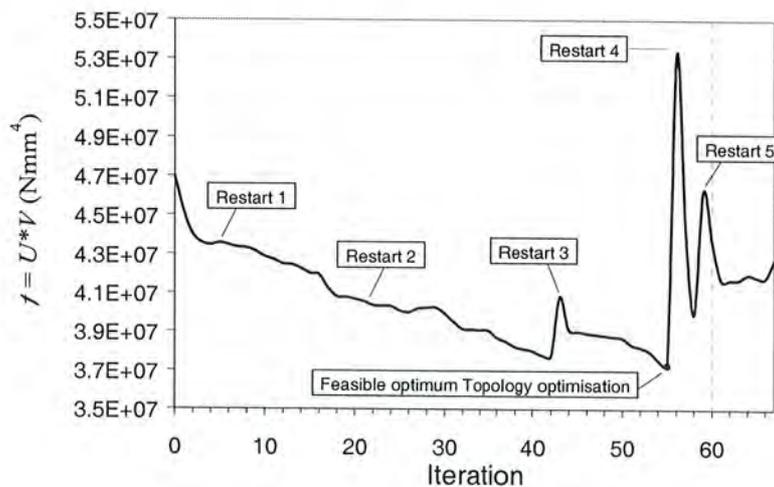
Figure 8.36: Evolution of the objective function

iteration 43 (Restart 3)

iteration 55

iteration 56 (Restart 4)

iteration 58

iteration 59 (Restart 5)
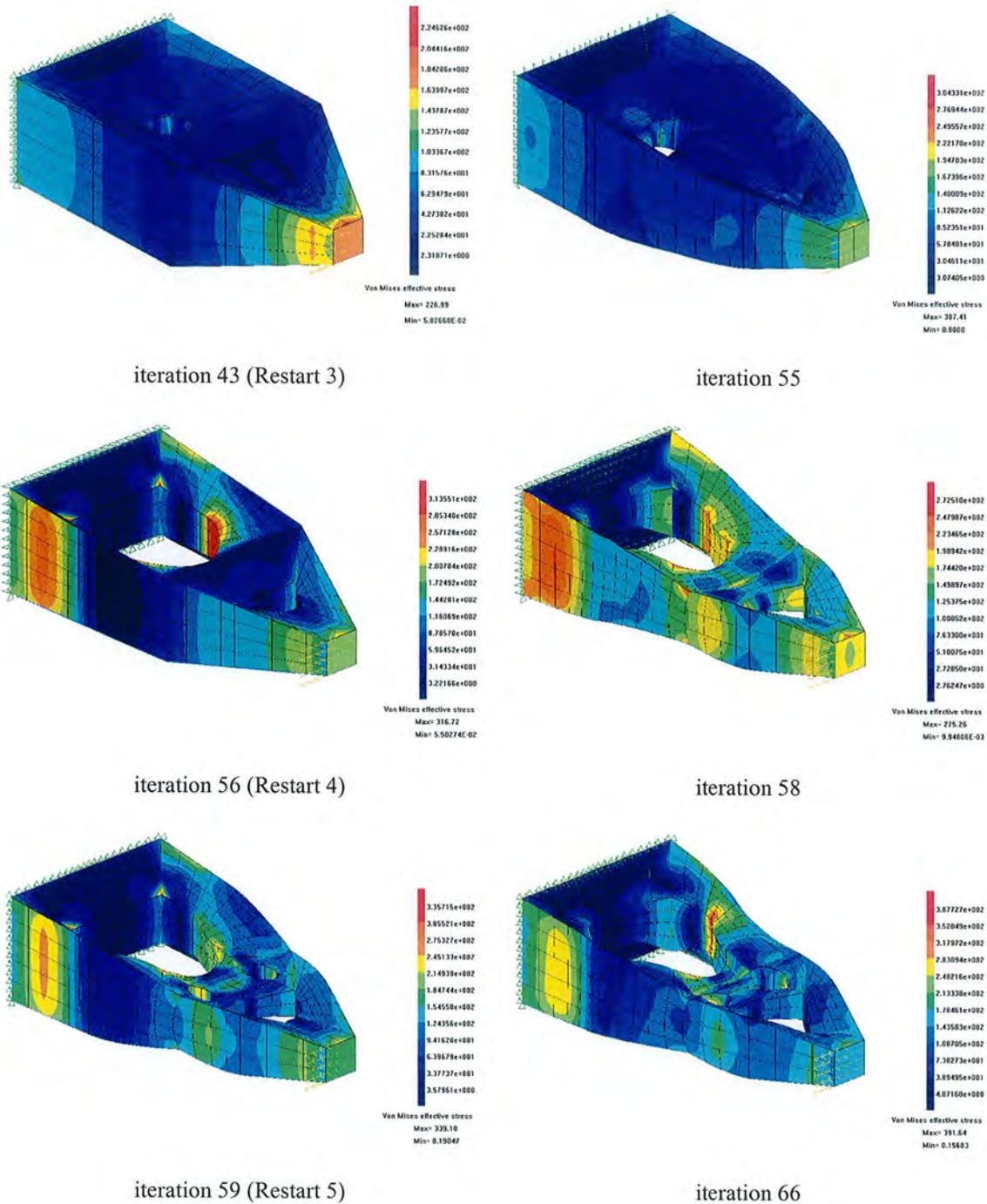
iteration 66

Figure 8.37: Von Mises stress contour plots

It is found that after the Restart 3 the objective function decreases until iteration 55 where it shows a minimum. At this stage, since the two premises of inner low stressed area and minimum distance are satisfied, new topology changes are implemented. Figure 8.37 shows the von Mises stress at the critical steps of the process. Restart 4 performs changes enlarging the cavities and at Restart 5 two new

cavities are created. From the related 2D plane stress case it should be expected that these changes would increase the performance of the structure. However, the evolution of the objective function shows that from iteration 55 further topology changes do not contribute to reach a minimum. By looking at the last iterations of the process, for example iteration 66 (figure 8.37), we can observe the presence of very low material near the fixed region (S1) which is not removed due to design constraints. This situation shows the clear dependence of the final design on the constraints imposed. Further investigation should consider the possibility of redefining such areas which do not contribute to increase the performance of the structure in spite of being constrained.

8.10 Discussion and Conclusions

In this chapter an algorithm for 3D structural optimisation has been presented. This algorithm is an extension from the gradientless 2D algorithm. Similarly, the structural analysis is boundary element-based and the geometry is parameterised with NURBS. Control points, which define the NURBS, migrate evolving to the optimum design. New features are inserted to control the geometry. Corners and shape edges are specially handled by using the *smoothing* and the *corner effect* algorithms, both implemented to ensure smoothness on the surfaces. Moreover, the perpendicular movement of the control points inside the control polyhedron is also influenced by the movement of the control points on the boundaries of this polyhedron.

The complexity of dealing with surfaces has restricted the automatic geometry control and in this initial study some parts of the algorithm have been implemented as a manual procedure. A defined algorithm has been strictly adhered to in all manual modifications to the model. Restarts are inserted to perform addition and removal of control points as well as topology changes. These restarts imply redefinition of the surfaces. Computational applications of the algorithm have been presented proving the algorithm for shape optimisation and showing preliminary results for topology optimisation.

9

DISCUSSION

9.1 Overview

An alternative approach to the evolutionary structural optimisation method (ESO) is presented. This approach uses the boundary element method (BEM) to carry out the structural analysis and nonuniform rational B-splines (NURBS) to define the geometry of the component. Since the shape of these B-splines is governed by a set of control points, the locations of these control points are used as design variables. The algorithm is not only applied to 2D structural optimisation but also extended to 3D structural optimisation problems.

The algorithm is implemented into a computer program using Visual C++. At the end of this work, about 15000 lines of code have been written, 10000 lines in the 2D algorithm and roughly 5000 lines in the 3D one. This code comprises the optimisation algorithms and the data processing algorithms dealing with both the structural analysis results and geometry definition. The algorithm in 2D is fully integrated within the in-house analysis software Concept Analyst. However, in 3D this software was not mature enough and the commercial program BEASY was used. In this latter case, the communication between software and algorithm involves extra coding and also special treatment of the analysis results and geometry.

9.2 Evaluation of the Algorithm and Results

9.2.1 Main Features of the Algorithm

The first chapters (chapters 1, 2, 3 and 4) of this thesis present a theoretical explanation of the main features on which the developed algorithm is based. A background to the field of structural optimisation is presented and the different optimisation methods are discussed. The state of structural optimisation is reviewed giving special mention to the evolutionary structural method (ESO). ESO is critically discussed with a detailed description of the method and its importance in the field of structural optimisation.

The boundary representation is explained introducing the nonuniform rational and nonrational B-splines. The mathematical definition of these parametric curves and surfaces is presented and graphically pictured with explanatory examples produced using the boundary representation code. The shape of the nonuniform rational B-spline (NURBS) curves and surfaces is defined by sets of control points and weights which are used as design variables. Therefore, complex geometries can be described in a very compact way using a small set of design variables as it has already been presented in the literature for example by Braibant and Fleury (1984). NURBS present the attractive feature of providing flexibility to describe a large variety of shapes by manipulating these control points and weights. Also of interest is that the number of design variables compared to FE-based ESO, which uses elements of the finite element mesh, is decreased.

The boundary element method (BEM), used to carry out the structural analysis, is detailed for elastostatic problems. The most important advantages of the BEM concern to mesh generation and manipulation of nodes and boundary elements. These are shown to be of great importance to this work in order to maintain smoothness along the boundaries and fast solution times. Moreover, the applicability of the BEM seems to be superior to the FEM in problems where the boundary stresses are important. This feature is not only important from a structural analysis

point of view but also from a structural optimisation perspective when design sensitivity calculations are involved (Kane (1986)).

9.2.2 Shape Optimisation

Shape optimisation problems are successfully solved using this new ESO approach. Numerical applications demonstrate the ability of the developed method to produce optimal shapes based on a stress-based criterion, generally von Mises stress. The method identifies and removes material that does not contribute to increase the performance of the structure from low stressed areas as well as adding material to high stressed and critical regions. Material removal and addition is performed by migrating sets of control points in the perpendicular direction to the boundary.

Migrating in an appropriate way the control points defining the NURBS generally, decreases the maximum von Mises stress and increases the minimum von Mises stress, although of course this stress description always passes through zero in regions of transition between tensile and compressive stress states on free boundaries in plane stress problems. As a result a more homogeneous von Mises stress distribution is obtained over the boundary of the component. In this sense, localised stress peaks are avoided in the structure and eventually a uniform stress state is reached at the surface in a similar way to the *biological growth method* (Mattheck and Burkhardt (1990)).

Smooth boundaries without artificial stress concentrations are obtained by working directly with the control points defining the geometry. Geometry control is enhanced by inserting or removing these control points automatically. Since a remeshing is performed for each structural analysis, and therefore for each iteration, there are no mesh distortion problems due to boundary movements. This remeshing process is only required over the boundary and therefore it is shown to be simple and fast. The integration of the optimisation algorithm together with the structural analysis shows a good computational performance of the algorithm and fast solution times.

9.2.3   Topology Optimisation

Topology optimisation is accomplished allowing the insertion of internal cavities in the inner low stressed areas of the structure, which are identified from the internal points. These cavities are also described by NURBS curves and so they have similar behaviour to the outside boundary. Therefore, both outside and inside are optimised at the same time. During the removal and addition process, the movement of the control points defining the cavities is either perpendicular to the boundary of the cavity or parallel to the closest outer boundary. Thus, it has been presented that if the outer boundary is closer than a certain minimum distance then the movement of the cavity is mainly influenced by the position of this closest boundary. Results in section 6.10 have shown that this movement produces areas where both boundaries (outer and inner cavity) remain parallel approaching to truss-like structures and thus, reproducing analytical optimum results.

Small cavities merge to form bigger cavities as a natural evolution of the topology of the structure. These cavities can merge to other cavities or to the closest outer boundary. Indeed, it appears the algorithm tends to evolve towards optima having large cavities by first creating small cavities and then merging them together or with the external boundary. It is unusual that a large cavity evolves simply by growth of a small cavity.

The effectiveness of the method in generating the optimum topologies as well as rapid solution times is shown by applying the 2D algorithm to benchmark problems. In the case of problems with an initial symmetry, almost symmetric shapes are obtained, in spite of the totally random position of the internal points and no symmetry being forced throughout the process. Results show that problems of checkerboard patters and jagged edges which are present in FE-based optimisation methods, as reported in chapter 2, are overcome with this approach.

In two-dimensional problems, multiple load cases are investigated using the logical AND/OR scheme also used in FE-ESO (Li *et al.* (1999)). The simple concept of this scheme is that material is removed from the structure if it is low stressed in all load cases. In contrast, material is added if it is high stressed in any of the load cases

considered. As expected, it is found that the final solution under multiple load cases is not necessarily a fully stressed design. Nevertheless, the study carried out has not considered other schemes such as a weighted average. This scheme makes use of weights to reflect the importance of each load case and thus, it considers a broader design space. Further research on this issue would complement and enhance the results obtained.

Numerical tests have shown that the solution can be mesh dependent. This dependency can be overcome by controlling the number of cavities that are allowed to be created. The initial distribution of control points also determines the final topology. It can be thought that increasing the number of control points increases the fidelity of the solution. Nevertheless this is not the case. On the contrary, it induces the creation of spikes and oscillations on the shape that slow the convergence and interfere with the ability of this algorithm to reach a good optimum. The explanation for this relates to the use of NURBS since changes in their definition are performed locally and these changes are related to the degree of the curve which in turn depends on the number of control points. A rule of thumb is presented for effective control point spacing for general problems.

9.2.4 Multi-Criteria Optimisation

In a two-dimensional context, multi-criteria optimisation problems are investigated. In such problems the different objective functions are in conflict. Generally, none of the possible solutions simultaneously fulfils all the objectives. The solution of this problem is obtained by transforming the multiple criteria problem into a single criterion one using the weighed sum method (WS). The Pareto concept produces the set of weights used in this weighted average scheme. This Pareto front is used to gain knowledge about trade-offs between objectives and also allows the choice of the most desirable solution.

Design sensitivity calculations are performed to explore the effect that changes in the design variables produce on the objective functions. Thus, sensitivity numbers are calculated for each design variable using finite difference (FD) derivatives. The FD

method is simple to apply and fast to obtain the derivatives but it is dependent on the step-size used. Therefore, the step-size is appropriately chosen to ensure the accuracy of the results. Moreover, in practice, the distance to move the control points is set to be equal (or related) to this step-size. The computing time required is successfully reduced by using BEM reanalysis to perform the sensitivity calculations. This reanalysis is based on the non-perturbed solution rather than full analyses for each perturbed state.

The technique for the example considered is shown to be efficient obtaining an evenly spread set of Pareto optimal points for an even distribution of weights. However, this multi-criteria algorithm is only in the early steps and further research using different objectives would show that this is not always the case (Das and Dennis (1997)).

The results obtained for shape and topology optimisation in 2D show good agreement with the results presented by FE-based ESO methods (Xie and Steven (1997)). Benchmark problems are reproduced successfully proving the effectiveness of the method for 2D structural optimisation problems. The use of boundary elements produces fast, simple and reliable meshing, as concluded in chapter 4. This approach inherits these BE features and consequently, shows the advantage of fast solutions compared to FE-based methods.

However, the use of boundary elements has restricted the applicability of the algorithm for example to non-linear problems. In such cases the reliability of boundary elements is limited which is a clear disadvantage over the FE. Thus, FE-based algorithms can be applicable to a larger range of physical situations. An alternative to this problem should be the application of this boundary-ESO algorithm based on FE combined to the boundary representation. In this context, Schramm and Pilkey (1993) applied NURBS coupled with FE to sensitivity analysis-based shape optimisation. Alternatively, strategies presented in chapter 4 such as coupling FE-BE or using for example the dual reciprocity boundary element method (DRBEM) should be further studied. In the literature, Burzyńscki and Orantek (2002) have implemented the FE and BE with GAs to optimise structures under mechanical, thermo-mechanical and dynamical loading. The implementation of such an

alternative structural analysis method would bring the possibility of considering other structural conditions such as frequency optimisation and thermal problems. As a result of the consideration of further conditions the multi-criteria algorithm would be applicable to a much broader range of objectives.

9.2.5 3D Problems

The gradientless algorithm that has proved successful for two-dimensional problems is extended to three-dimensional optimisation problems. Similarly, the structural analysis is boundary element-based but the commercial program BEASY is used. The optimisation algorithm is no longer fully integrated to the structural analysis and manual intervention is required to deal with the geometric models and data produced by BEASY.

The geometry is parameterised with NURBS surfaces and curves. The implementation of surfaces is a new aspect compared to the 2D optimisation. These surfaces are defined by control point nets and therefore these control points migrate evolving to the optimum design.

New features are considered in order to control the geometry. Corners and shape edges are especially handled by using the *smoothing algorithm* and the *corner effect algorithm*, both of which are implemented to ensure smoothness on the surfaces. Moreover, the perpendicular movement of the control points inside the control point nets is also influenced by the movement of the control points on the boundaries of this control point net.

At this state of the research, the complexity of dealing with surfaces has restricted the automatic geometry control to become a manual process. Restart procedures redefining the surfaces are inserted to perform addition and removal of control points as well as topology changes. Computational applications of the algorithm prove the algorithm for shape optimisation and show preliminary results for topology optimisation. Nevertheless, these results are still not comparable to the equivalent FE-ESO algorithms (Young *et al.* (1999)). In 3D problems, the FE-ESO algorithm is a direct application of the 2D one and, therefore based on the same concept of slow

removal and addition of elements according to certain criteria. In this case, the 3D algorithm is not a straightforward extension of the 2D algorithm. The use of NURBS to define the geometry brings difficulties involved with the limitation of the boundary element software package dealing with these types of surfaces. Moreover, the manual communication between the software and the optimisation code reduces the effectiveness of the overall process. It can be concluded that further research is needed to be able to develop an algorithm that in 3D produces results analogous to the FE-based ones.

10

CONCLUSIONS AND IMPLICATIONS FOR FUTURE RESEARCH

10.1 Overview

In this thesis the development of a computational algorithm to be used as a tool for shape and topology optimisation has been presented. In each chapter a detailed explanation of the algorithm and its results is given as well as the corresponding discussion and conclusions at the end of the chapter. This final chapter summarises the main points of these conclusions and gives some implications for further work.

10.2 Achievements

The achievements of this thesis can be stated as follows

- The developed structural optimisation approach is based on the evolutionary structural optimisation method (ESO).

- The boundary element method (BEM) is used for the structural analysis. In 2D structural optimisation the in-house software Concept Analyst is fully integrated with the optimisation algorithm. In 3D problems this is not the case and the commercial program BEASY is used.

- Nonuniform rational B-spline (NURBS) curves (2D) and surfaces (3D) are implemented to describe the changeable geometry.

- The coordinates of the control points defining the NURBS curves and surfaces are the design variables.

- The structural optimisation algorithm is successfully applied for shape optimisation.

- As the classical ESO basic concept, the optimal shapes evolved following a gradientless method or stress-based criterion, generally von Mises stress.

- The method identifies and removes material from low stressed regions as well as adds material to high stressed areas by migrating sets of control points.

- Smooth boundaries are obtained by working directly with the control points defining the geometry.

- The remesh process is performed every iteration and it is only required over the boundary. This provides robustness to the process since it is straight forward to generate a reliable boundary mesh.

- Practical applications are presented to optimise the shape of holes in plates under in plane loading and fillets, considering different objectives such as stress levelling, weight reduction and minimisation of the stress concentration factor.

- For one load case and strain energy minimisation fully stressed designs are obtained.

- By integrating the structural analysis and the structural optimisation algorithm good performance and fast solution times are achieved.

- Topology changes are performed creating holes in the inner low stressed areas of the structure.

- These holes are also described by NURBS curves and so they have similar behaviour to the outside boundary. This appears to be the natural method of evolution of optima containing large cavities.

- Holes merge to other holes or to the closer boundary as a natural evolution of the topology of the structure.

- Examples are reproduced showing the effectiveness of the method generating optimum topologies for fast solution times.

- Multiple load cases are investigated using the logical AND/OR scheme. The results show that the final solution under multiple load cases is not necessarily a fully stressed design.

- Numerical tests show the dependency of the solution on the boundary mesh and the initial distribution of control points. A rule of thumb is presented for effective control point spacing.

- Multi-criteria optimisation problems are investigated.

- The solution of the multi-criteria problem is obtained transforming the multiple objective functions into a single objective function using the weighted sum method (WS).

- Trade-offs between objectives are considered using the Pareto concept.

- Sensitivity numbers are calculated to assess the effect of changes in the design variables on the objective functions. These sensitivity calculations are performed using finite difference (FD) derivatives.

- A modified version of a reanalysis scheme present in Concept Analyst is used to carry out the FD calculations and considerably reduces the computing time for each iteration.

- The technique applied to a connecting rod example shows it to be efficient at obtaining an even spread set of Pareto optimal points for an even distribution of weights.

- The gradientless algorithm for 2D structural optimisation is extended to 3D problems.

- In 3D, corners and shape edges are especially handled by using a smoothing algorithm and a corner effect algorithm to ensure smoothness on the surfaces.

- The perpendicular movement of the control points inside the control point nets is also influenced by the movement of the control points on the boundaries of this control point net. An algorithm mapping the surfaces of the geometry is developed for this purpose.

- Restart procedures redefining the surfaces perform addition and removal of control points as well as topology changes.

- Computational applications of the algorithm for three-dimensional problems prove the algorithm for shape optimisation and show preliminary results for topology optimisation.

10.3 Conclusions

The developed algorithm, which is based on the evolutionary structural optimisation (ESO) method, is an effective tool for shape and topology optimisation. The implementation of the boundary element method (BEM) for the elastostatic structural analysis ensures fast and reliable meshes every iteration. As a feature of BEM, this mesh is only required over the boundary. In 2D structural optimisation the in-house software Concept Analyst is fully integrated within the optimisation algorithm which allows savings in computational time and improves the performance.

Moreover, boundary representation is used in terms of nonuniform rational B-splines (NURBS). The use of the locations of the control points defining these NURBS as design variables allows a direct manipulation of the geometry and ensures smooth boundaries.

Multi-criteria problems can be studied by transforming them into a single criterion problem using the weighted sum method (WS) and trade-offs between the different criteria. These trade-offs are performed using the Pareto concept. When applied to multi-criteria problems, this approach considers the use of sensitivity numbers to assess the effect on the objective functions of changes in the design variables. The calculation of these sensitivity numbers using finite differences is reduced considerably by implementing reanalysis schemes.

The extension of the gradientless algorithm from 2D to 3D problems is not straightforward since care must be taken dealing with the NURBS surfaces. More importantly, the optimisation algorithm is no longer fully integrated within the structural analysis, which is carried out using the commercial program BEASY. For this reason, the advantages in performance and computing time found in 2D are not fully achieved yet. However, a working optimisation algorithm is successfully implemented as shown in the selected examples.

10.4 Implications for Future Research

The approach for structural optimisation has been developed showing good results. The initial aims of this PhD have been accomplished however, by no means has the research been completed. While working on this topic new paths and implications for future research have arisen.

Future work might involve expanding the methods to include other objectives such as natural frequency and buckling as an extension of the multi-criteria and multidisciplinary structural optimisation and towards robust designs.

The results for three-dimensional structural optimisation are preliminary. Further lines of investigation will incorporate in-house software or alternatively, will enhance the communication between the commercial boundary element software and the structural optimisation algorithm.

The use of trim lines in 3D has been restricted purely to geometry definition however future research will include these lines as part of the optimisation process and fully independent of the design variables.

There is considerable justification also for research into the surface management, i.e. the NURBS surface patching, re-patching algorithms that might be automated (possibly using trimmed surfaces) and the association of the surface description to the element meshing. Work on identification of appropriate mesh density and control point spacing is justified, particularly in 3D modelling.

The restart procedures performed manually will be implemented for the automatic insertion and deletion of holes as well as the topology changes. This improvement will clearly increase the efficiency of the algorithm in 3D generating further savings in computational effort.

BIBLIOGRAPHY

[1]   Abu Kassim, A.M., Topping, B.H.V. (1987) *Static reanalysis: a review*. J. Structural Div, ASCE 113(5): 1029-1045.

[2]   Afonso, S.M.B., Macedo, C.M.H., Oliveira, D.A.P. (2002). *Structural shape optimization under multicriteria conditions*. Proceedings of the Fifth World Congress on Computational Mechanics (WCCM V), Vienna, Austria.

[3]   Aliabadi, M.H. (2002). *The Boundary Element Method. Volume 2: Applications in Solids and Structures*. John Wiley & Sons, Ltd.

[4]   Annicchiarico, W., Cerrolaza M. (2001). *Structural shape optimization 3D finite-element models based on genetic algorithms and geometric modeling*. Finite Elements in Analysis and Design, 37: 403-415.

[5]   Arora, J.S. (1989). *Introduction to optimum design*. McGraw-Hill.

[6]   Avellaneda, M. (1987) *Optimal Bounds and microgeometries for elastic two-phase composites*. SIAM J. Appl. Math. 47(6): 1216-1240.

[7]   Balling, R.J. (1991). *Optimal steel frame design by simulated annealing*. Journal for Structural Engineering 117: 1780-95.

[8]   Bates, S.J., Sienz, J., Pittman, J.F.T. (2002).*Comparison of conventional Optimization with Robust Design techniques for Slit Die Design*. Proceedings of the 4th ASMO UK/ISSMO conference, Newcastle-upon-Tyne, UK.

[9]   Bathe, K. (1982). *Finite element procedures in engineering analysis*. Prentice-Hall, Inc., Englewood Cliffs. New Jersey.

[10] 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.

[11] BEASY *Users Manual.* Computational Mechanics BEASY Ltd (2002), Ashurst Lodge, Ashurst, Southampton, Hampshire, SO40 7AA, England.

[12] Becker, A.A. (1992). *The boundary element method in engineering*, McGraw-Hill.

[13] Bendsøe, M.P., Kikuchi, N. (1988). *Generating optimal topologies in structural design using a homogenisation method.* Comp. Meth. in Appl. Mech. Engng 71: 197-224.

[14] Bendsøe M.P. (1995). *Optimization of Structural Topology, Shape and Material.* Springer. Berlin.

[15] Botkin, M.E. (1981). *Shape Optimization of Plate and Shell Structures.* AIAA Journal 20: 268-273.

[16] Botkin, M.E. (1992). *Three-Dimensional Shape Optimization Using Fully Automatic Mesh Generation.* AIAA Journal 30(7): 1932-1934.

[17] Blachut, J. (2003). *Optimal barreling of steel shells via simulated annealing algorithm.* Computers & Structures 81: 1941-1956.

[18] Braibant, V., Fleury, C. (1984). *Shape optimal design using B-splines.* Computer Methods in Applied Mechanics & Engineering 44: 247-267.

[19] Brebbia, C.A., Trevelyan, J. (1986). *On the accuracy and convergence of Boundary Element results for the Floyd pressure vessel problem.* Computers & Structures 24(3): 513-516.

[20] Brebbia, C.A., Dominguez, J. (1989). *Boundary Elements. An introductory course.* Computational Mechanics Publications.

[21]   Burczyński, T., Orantek, P. (2002). *Evolutionary Algorithms in Computational Mechanics: Applications in Optimization and Identification.* Proceedings of the Fifth World Congress on Computational Mechanics (WCCM V), Vienna, Austria.

[22]   Castillo, E., Cobo, A., Fernandez-Canteli, A., Jubete, F., Pruneda, E. (1998). *Updating inverses in matrix analysis of structures.* Int. J. Numer. Meth. Engng. 43: 1479-1504.

[23]   Cerrolaza, M., Annicchiarico, W., Martinez, M. (2000). *Optimization of 2D boundary element models using β-splines and genetic algorithms.* Engineering Analysis with Boundary Elements 24: 427-440.

[24]   Chu, D.N., Xie, Y.M., Hira, A., Steven, G.P. (1997). *On various aspects of evolutionary structural optimization for problems with stiffness constraints.* Finite Elements in Analysis and Design 24(4): 197-212.

[25]   Cruse, T.A. (1969). *Numerical solutions in three dimensional elastostatics.* International Journal of Solids and Structures 5: 1259-74.

[26]   Cruse, T.A., VanBuren, W. (1971). *Three-dimensional elastic stress analysis of a fracture specimen with an edge crack.* International Journal of Fracture Mechanics 7(1):1-15.

[27]   Das, I., Dennis, J.E. (1997). *A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems.* Structural Optimization 14: 63-69.

[28]   Das, I., Dennis, J.E. (1998). *Normal-boundary intersection: a new method for generating the Pareto surface in nonlinear multicriteria optimisation problems.* SIAM, J. Optim. 8(3): 631-657.

[29]   Dems, K., Mroz Z. (1978). *Multiparameter structural shape optimization by the finite element method.* International Journal for Numerical Methods in Engineering 13: 247-263.

[30]   Ding, Y. (1986). *Shape optimisation of structures: a literature survey.* Computers & Structures 24(6): 985-1004.

[31]   Di Pisa, C., Aliabadi, M.H. (2003). *BEM analysis of thin-walled assembled plate structures.* Proceedings of the fourth UK conference on boundary integral methods. Ed. Sia Amini. The University of Salford. Salford. UK

[32]   Eschenauer, H.A, Kobelev, V.V, Schumacher, A. (1994). *Bubble method for topology and shape optimisation of structures.* Structural optimization 8: 42-51.

[33]   Eschenauer, H.A. (2000). *Multidisciplinary optimization procedure in design processes – Basic ideas, aims, scope, concepts.* In Emerging methods for multidisciplinary optimization. Ed. J. Blachut and H. Eschenauer. CISM Courses and Lectures N° 425.

[34]   Eschenauer, H. A., Olhoff N. (2001). *Topology optimization of continuum structures: A review.* AIAA Journal 54(4): 331-390.

[35]   Esping, J.D. (1985) *A CAD approach to the minimum weight design problem,* International Journal for Numerical Methods in Engineering 21: 1049-1066.

[36]   Farin, G.E. (1988). *Curves and surfaces for computer aided geometric design: a practical guide.* Boston; London: Academic Press.

[37]   Fleury, C. (1979). *Structural weight optimization by dual methods of convex programming.* International Journal for Numerical Methods in Engineering 14: 1761-1783.

[38]   Fleury, C., Schmit, L.A. (1980). *Structural synthesis by combining approximate concepts and dual methods,* AIAA J. 18: 1252-1260.

[39]   Fleury, C., Braibant, V. (1986). *Structural Optimization. A new dual method using mixed variables.* International Journal for Numerical Methods in Engineering 23: 409-428.

[40] Francavilla, A., Ramakrishnan, C.V., Zienkiewicz, O.C. (1975). *Optimization of shape to minimize stress concentration.* Journal of Strain Analysis 10(2): 63-70.

[41] Frangopol, D.M., Maute, K. (2003). *Life-cycle reliability-based optimization of civil and aerospace structures.* Computers & Structures 81: 397-410.

[42] Gallagher, R.H (1977). Gallagher, R.H., Zienkiewicz, O.C. (eds.) *Optimum Structural Design*, Wiley.

[43] Gambling, M., Jones, R.D., Toropov, V.V., Alvarez, L.F. (2001). *Application of optimization strategies to problems with highly non-linear response.* Proceedings of the 3rd ASMO UK/ISSMO conference, Harrogate, UK.

[44] Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimisation and Machine Learning*, Addison-Wesley.

[45] Haftka, R.T., Grandhi, R.V. (1986). *Structural Shape Optimization, a survey.* Computer Methods in Applied Mechanics & Engineering 57(1): 91-106.

[46] Haftka, R.T. and Gürdal Z. (1992). *Elements of Structural Optimization.* Dordrecht, Kluwer Academic Publishers.

[47] Hajela, P., Lee, E. (1995). *Genetic algorithms in truss topological optimization.* Int. J. Solids Structures 32(22): 3341-3357.

[48] Hassani, B., Hinton, E. (1998). *A review of homogenization and topology optimization III-topology optimization using optimality criteria.* Computers & Structures 69: 739-756.

[49] Haug E. J., Choi, K.K., Komkov, V., (1986). *Design sensitivity analysis of structural systems*, Academic Press, New York, NY.

[50] Hemp, W.S. (1973). *Optimum Structures.* Oxford, Clarendon Press.

[51] Hinton, E., Sienz, J. (1995). *Fully Stressed Topological Design of Structures using an evolutionary procedure.* Engineering Computations 12: 229-244.

[52]   Holland, J.H. (1975) *Adaptation in natural and artificial systems*. London: MIT Press.

[53]   Imam, M.H. (1982). *Three-dimensional shape optimization*. International Journal for Numerical Methods in Engineering 18: 661-673.

[54]   Jacobsen, J.B, Olhoff, N., Rønholt, E. (1998). *Generalized shape optimization of three-dimensional structures using materials with optimum microstructures*. Mechanics of Materials, 28: 207-225.

[55]   Jaswon, M., Symm, G.T. (1977). *Integral Equations Methods in Potential Theory and Elastostatics*. Academic Press. London.

[56]   Kane, J.H. (1986). *Shape optimization utilizing a boundary element formulation*. BETECH 86, MIT, CAMBRIDGE, MA.

[57]   Kane, J.H., Keshava Kumar, B.L., Gallagher, R.H. (1990). *Boundary element iterative reanalysis for continuum structures*. J. Engng. Mech, ASCE 116(10): 2293-2309.

[58]   Kane, J.H., Zhao, G., Wang, H., Guru Prasad, K. (1992). *Boundary Formulations for Three-Dimensional Continuum Structural Shape Sensitivity Analysis*. Journal of Applied Mechanics, 59: 827-834.

[59]   Kane, J.H. (1994). *Boundary Element Analysis in Engineering Continuum Mechanics*. Prentice-Hall.

[60]   Kegl, M. (2000). *Shape optimal design of structures: an efficient shape representation concept*. International Journal for Numerical Methods in Engineering 49: 1571-1588.

[61]   Kim, H., Querin, O. M., Steven, G.P., Xie, Y.M. (2000). *A method for varying the number of cavities in an optimized topology using evolutionary structural optimization*. Structural and Multidisciplinary Optimization 19: 140-147.

[62] Kim, H., Garcia, M.J., Querin, O.M., Steven, G.P., Xie, Y.M. (2000). *Introduction of fixed grid in evolutionary structural optimisation.* Engineering computations 17(4): 427-439.

[63] Kirkpatrick, S., Gellatt, C.D., Vecchi, M.P. (1983). *Optimization by simulated annealing.* Science 220: 671-680.

[64] Kirsch, U. (1991). *Reduced basis approximations of structural displacements for optimal design.* AIAA Journal 29: 1751-1758.

[65] Kirsch, U. (2002). *Design-Oriented Analysis of Structures.* In P. Gosling (Ed.). Proceedings of the 4th ASMO-UK/ISSMO conference, Newcastle, UK, University of Newcastle, UK.

[66] Kita, E., Tanie, H. (1999) *Topology and shape optimization of continuum structures using GA and BEM.* Struct. Optim. 17(2-3): 130-139.

[67] Kristensen, E.S., Madsen, N.F. (1976). *On the optimum shape of fillets in plates subject to multiple in-plane loading cases.* International Journal for Numerical Methods in Engineering 10: 1007-1019.

[68] Kodiyalam, S., Vanderplaats, G.N. (1992). *Constructive Solid Geometry Approach to Three-Dimensional Structural Shape Optimization.* AIAA Journal, 30(5): 1408.

[69] Laarhoven, P.J.M. van, Aarts, E. (1987). *Simulated annealing: theory and applications.* D. Reidel Publishing, Dordrecht, The Netherlands.

[70] Lachat, J.C., Watson, J.O. (1976). *Effective numerical treatment of boundary integral equations.* International Journal for Numerical Methods in Engineering 10: 991-1005.

[71] Leite, J.P.B., Topping, B.H.V. (1999). *Parallel simulated annealing for structural optimization.* Computers & Structures 73: 545-564.

[72]   Leu, L. (1999). *Shape optimization by the boundary element method with a reduced basis reanalysis technique*. Structural Engineering and Mechanics 8(1): 73-84.

[73]   Li, Q., Steven, G.P., Xie, Y.M. (1999). *On equivalence between stress criterion and stiffness criterion in evolutionary structural optimization*. Structural and Multidisciplinary Optimization 18: 67-73.

[74]   Li, Q., Steven, G.P., Xie, Y.M. (2001). *Evolutionary thickness design with stiffness maximization and stress minimization criteria*. International Journal for Numerical Methods in Engineering 52: 979-995.

[75]   Li, Q., Steven, G.P., Xie, Y.M. (2001). *A simple checkerboard suppression algorithm for evolutionary structural optimization*. Structural and Multidisciplinary Optimization 22: 230-239.

[76]   Luenberger, D.G. (1973). *Introduction to linear and nonlinear programming*. Addison-Wesley Publishing Company, Inc.

[77]   Mackie, R.I. (1998). *An object-oriented approach to fully interactive finite element software*. Adv. Engng. Software 29(2): 139-149.

[78]   Mattheck, C., Burkhardt, S. (1990). *A New Method of Structural Shape Optimisation Based on Biological Growth*. International Journal of Fatigue 12(3): 185-190.

[79]   Maute, K., Ramm, E., (1995). *Adaptive topology optimization*. Structural optimisation 10: 100-112.

[80]   Maxwell, J.C., (1869). *Scientific Papers*. Volume 2, 175-177.

[81]   Messac, A. (2000). *From Dubious Construction of Objective Functions to the Application of Physical Programming*. AIAA Journal 38(1): 155-163.

[82]   Messac, A., Ismail-Yahaya, A. (2002). *Multiobjective robust design using physical programming*. Struct Multidisc Optim 23: 357-371.

257

[83]   Michell, A.G.M. (1904). *The Limits of Economy of Material in Frame Structures*. Philosophical magazine 8(6): 589-597.

[84]   Mikhlin, S.G. (1957). *Integral Equations*. Pergamon Press. New York.

[85]   Mota Soares, C.A., Rodrigues, H.C., Oliveira Faria, L.M. (1984). *Optimization of the geometry of shafts using boundary elements*. ASME Journal of Mechanisms, Transmissions and Automation in Design 106: 199-202.

[86]   Muskhelishvili, N.I. (1953). *Some Basic Problems of the Mathematical Theory of Elasticity*. Noordhoff, Holland.

[87]   Na, M., Kikuchi, N., Taylor, J.E. (1984). *Optimal shape remodelling of linearly elastic plates using finite element methods*. International Journal for Numerical Methods in Engineering 20: 1823-1840.

[88]   Nardini, D., Brebbia, C.A. (1982). *A New Approach to Free Vibration Analysis Using Boundary Elements*. Boundary Element Methods in Engineering. Computational Mechanics Publications, Southampton.

[89]   Oda, J., Yamazaki, K. (1977). *On a technique to obtain an optimum strength shape of axisymmetric body by the finite element method*. JSME 20(150): 1524-1532.

[90]   Osyczka, A. (2002). *Evolutionary Algorithms for Single and Multicriteria Design Optimization*, Physica-Verlag, Heidelberg New York.

[91]   Patnaik, S.N., Guptill, J.D., Berke, L. (1995). *Merits and limitations of optimality criteria method for structural optimization*. International Journal for Numerical Methods in Engineering 38: 3087-3120.

[92]   Partridge, P.W., Brebbia, C.A., Wrobel, L.C. (1992). *The Dual Reciprocity Boundary Element Method*. Computational Mechanics Publications. Elsevier Applied Science.

[93]   Parvizian, J. and R. T. Fenner (1997). *Shape optimisation by the boundary element method: a comparison between mathematical programming and normal movement approaches*. Engineering Analysis with Boundary Elements 19: 137-145.

[94]   Pedersen, P., Laursen, C.L. (1983). *Design for minimum stress concentration by finite elements and linear programming*. J. Struct. Mech. 10(4): 375-391.

[95]   Perrey-Debain, E., Trevelyan, J., Bettes, P. (2003). *Numerical aspects of single wave basis boundary elements for acoustic scattering*. Proceedings of the fourth UK conference on boundary integral methods. Ed. Sia Amini. The University of Salford. Salford. UK.

[96]   Piegl, L, Tiller, W. (1997). *The NURBS book.* 2nd ed. Springer-Verlag.

[97]   Pineda, E., Aliabadi, M.H. (2003). *Advanced boundary element analysis of elastoplastic problems*. Proceedings of the fourth UK conference on boundary integral methods. Ed. Sia Amini. The University of Salford. Salford. UK.

[98]   Portela, A., Aliabadi, M.H., Rooke, D.P. (1992). *The dual boundary element method: efficient implementation for crack problems*. International Journal for Numerical Methods in Engineering 33: 1269-1287.

[99]   Prager, W., Shield, R.T. (1967). *A general theory of optimal design.* J. Appl. Mech. 34: 184-186.

[100]   Press, W.H., Flannery, B.P., Teukolsky, S.A., Wetterling, W.T. *Numerical Recipes*. Cambridge: Cambridge University Press, 1986.

[101]   Proos, K.A., Steven, G.P., Querin, O.M., Xie, Y.M. (2001). *Stiffness and inertia multicriteria evolutionary structural optimisation*. Engineering Computations 18(7): 1031-1054.

[102]   Queau, J.P., Trompette, P.H. (1980). *Two-dimensional shape optimal design by the finite element method*. International Journal for Numerical Methods in Engineering 15: 1603-1612.

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

[104] 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.

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

[106] Reynolds, D., Christie, W.C., Bettess, P., McConnachie, J., Bull, J.W. (2001). *Evolutionary material translation: a tool for the automatic design of low weight, low stress structures.* International Journal for numerical methods in engineering 50: 147-167.

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

[108] Rogers, D.F. (2001). *An introduction to NURBS: with historical perspective.* Morgan Kaufmann publishers. Academic Press.

[109] Rozvany, G.I.N., Hill, R.H. (1976). *General theory of optimal force transmission by flexure.* Advances in Applied Mechanics 16: 184-308.

[110] Rozvany, G.I.N., Gollub, W. Zhou, M. (1989). *Optimal design of large discretized systems by iterative optimality criteria methods.* Proceedings of the NATO Advanced Study Institute on Optimization and Decision Support Systems in Civil Engineering, Edinburgh, UK. ISBN 0-7923-1955-9 (Volume I).

[111] Rozvany, G.I.N., Zhou, M., Birker, T. (1992). *Generalized shape optimization without homogenization.* Structural Optimization 4: 250-252.

[112] Rozvany, G.I.N., Bendsøe, M. P., Kirsch, U. (1995). *Layout optimization of structures.* Applied Mechanics Reviews 48(2): 41-119.

[113] Saad, Y., Schultz, M.H. (1986). *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*. SIAM, J. Sci. Stat. Comput. 7(3): 856-869.

[114] Sanchez-Palencia, E. (1980). *Non-homogeneous media and vibration theory.* Lecture notes in physics, 127, Springer, Berlin.

[115] Schmit, L.A., Farshi B. (1974). *Some approximation concepts for structural synthesis*, AIAA Journal, 12(5): 692-699.

[116] Schmit, L.A., Miura, H. (1976). *A new structural analysis/synthesis capability-ACCES1.* AIAA Journal 14(5): 661-671.

[117] Schmit, L.A. (1981). *Structural synthesis-its genesis and development*, AIAA Journal 19(10): 1249-1263.

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

[119] Schramm, U., Pilkey, W.D. (1995). *The application of rational B-splines for shape optimization.* Proceedings of the First World Congress of Structural Multidisciplinary Optimization (WCSMO-1), Germany.

[120] Sienz, J. (1994). *Integrated structural modelling, adaptive analysis and shape optimization.* PhD Thesis. Dept. of Civil Engineering. Swansea, University of Wales.

[121] Sienz, J., Hinton, E. (1997). *Reliable structural optimization with error estimation, adaptivity and robust sensitivity analysis.* Computers and Structures 64: 31-63.

[122] Sigmund, O., Petersson, J. (1998). *Numerical instabilities in topology optimization: A survey on procedures dealing with checkerboards, mesh-dependencies and local minima.* Structural Optimization 16: 68-75.

[123] Steven, G.P., Querin, O.M., Xie, Y.M. (2000). *Evolutionary structural optimisation (ESO) for combined topology and size optimisation of discrete structures*. Comp. Meth. In Appl.Mech. Engng 188: 743-754.

[124] Steven, G.P., Li, Q., Xie, Y.M. (2000). *Evolutionary topology and shape design for general physical field problems*. Computational Mechanics 26: 129-139.

[125] Steven, G.P., Qing, L., Proos, K., Xie, Y.M. (2002).*The Role of Physical Sensitivity in Evolutionary Topology Design Optimisation with Multi-Criteria and Multi-Physics*. Proceedings of the Fifth World Congress on Computational Mechanics (WCCM V), Vienna, Austria.

[126] Su, J., Renaud, J.E. (1997). *Automatic Differentiation in Robust Optimization*. AIAA Journal 35 (6): 1072-1079.

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

[128] Timonshenko, S.P., Goodier, J.N. (1970) *Theory of Elasticity*.

[129] Trevelyan, J. (1994). *Boundary Elements for Engineers*. Computational Mechanics Publications.

[130] Trevelyan, J., Wang, P. (2001). *Interactive re-analysis in mechanical design evolution. Part I. Background and implementation*, Computers & Structures 79(9): 929-938.

[131] Trevelyan J., Wang, P., Walker, S.K. (2002). *A scheme for engineer-driven mechanical design improvement*. Engineering Analysis with Boundary Elements 26(5): 425-433.

[132] Vanderplaats, G.N. (1989). *An Assessment of Current Non-Linear Programming Algorithms for Structural Design. Part I: Basic Algorithms*. NATO Advanced Study Institute on Optimization and Decision Support Systems in Civil Engineering, Edinburgh, UK.

[133] Van der Vorst, H.A. (1992). *BI-CGSTAB: A fast and smoothly converging variant of BI-CG for the solution of nonsymmetric linear systems.* SIAM, J. Sci. Stat. Comput. 13(2): 631-644.

[134] Venkayya, V B., Knot, N.S., Berke, L. (1973). *Application of optimality criteria approaches to automated design of large practical structures.* 2nd Symposium on structural optimization, Milan, Italy, AGARD-CP-123.

[135] Wasiutyński, Z., Brandt, A. (1963). *The present state of knowledge in the field of optimum design of structures.* Appl. Mech. Rev., 16(5): 341-350.

[136] Wilde, D.J., Beightler, C.S. (1967). *Foundations of Optimization.* Prentice-Hall, N.J.

[137] Woon, S.Y. (2002). *Effective GA-Based optimisation of continuum structures.* PhD thesis. University of Sydney.

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

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

[140] Yang, X.Y., Xie, Y.M., Liu, J.S., Parks, G.T., Clarkson, P.J. (2003). *Perimeter control in the bidirectional evolutionary optimization method.* Structural and Multidisciplinary Optimization 24: 430-440.

[141] Young, V.O., Querin, O.M., Steven, G.P. (1999). *3D and multiple load case bi-directional evolutionary structural optimization (BESO).* Structural Optimization 18: 183-192.

[142] Zhao, Z.Y. (1995). *Shape design sensitivity analysis by BEM - A review.* Proceedings of the First World Congress of Structural Multidisciplinary Optimization (WCSMO-1), Germany.

[143] Zhou, M., Rozvany, G.I.N. (1993). *DCOC: an optimality criteria method for large systems. Part II: algorithm.* Structural Optimization 6: 250-262.

[144] Zhou, M. and R. T. Haftka (1995). *A comparison of optimality criteria methods for stress and displacement constraints.* Comp. Meth. In Appl. Mech. Engng 124: 253-271.

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

[146] Zhou, M. Rozvany, G.I.N. (2001). *On the validity of ESO type methods in topology optimization.* Structural Optimization 21: 80-83.

[147] Zienkiewicz, O.C., Campbell, J.S. (1977). Gallagher R.H., Zienkiewicz O.C. (eds.) *Optimum Structural Design,* 109-126, Wiley.

[148] Zienkiewicz, O.C. (1989). *The finite element method. Vol. 1. Basic formulation and linear problems.* McGraw-Hill. New York.