DESIGN OPTIMIZATION USING THE BOUNDARY INTEGRAL EQUATION METHOD

BY

AZAMOLSADAT MOGHADDASI-TAFRESHI
B.Sc. (Eng), M.Sc. (Eng)

Thesis submitted for the degree of
Doctor of Philosophy
in the Faculty of Engineering of the University of London
and for the
Diploma of Membership of the Imperial College

December 1989

Department of Mechanical Engineering
Imperial College of Science, Technology & Medicine
London SW7 2BX

ABSTRACT

This thesis presents the application of an efficient numerical tool, namely, the Boundary Integral Equation (BIE) method, to design optimization for two-dimensional elastic structures.

Different optimization methods in mathematical programming, especially those numerical optimization methods which are applicable to engineering design are discussed. Following a brief review of the analytic basis of the direct formulation of the BIE method, analytical differentiation of the BIE is carried out for the design sensitivity analysis in order to compute directly the rates of change of stresses with respect to changes of the design variables, which are the positions of the boundary nodes. Then results are compared with those derivatives obtained by finite difference schemes. Several steps in the shape optimization process, such as model description, selection of the objective function and shape variables, representation of boundary shape, boundary element mesh re-generation during the optimization process and the solution procedure for two different categories of shape design are described, which are as follows; I) stress minimization, and II) weight (or mass) minimization with stress constraints.

Results for test cases are given and compared with exact or existing approximate solutions. The numerical formulation employs three-node line elements for the boundary.

To my parents

ACKNOWLEDGEMENTS

The author wishes to express her sincere appreciation and deepest gratitude to Dr. R. T. Fenner, for his invaluable guidance, kind advice and constructive criticism throughout this work. His encouragement and continuous interest in the project led me to the completion of this thesis.

The generous help of Dr. F. Nadiri, the CAD Teaching Unit Manager, in using the CAD system is gratefully acknowledged.

My greatest gratitude is due to my parents for their encouragement and sacrifices through all stages of my life and my husband, Dr. P. Behrouzi, for his support and understanding during this work.

This work was made possible with the financial support of the United Kingdom Atomic Energy Authority, to whom the author is grateful.

CONTENTS

	Page
TITLE PAGE	1
ABSTRACT	2
DEDICATION	3
ACKNOWLEDGEMENTS	4
CONTENTS	5
NOMENCLATURE	11
ABBREVIATIONS	20
CHAPTER 1 INTRODUCTION	21
CHAPTER 2 THEORY OF OPTIMIZATION	
2.1 Introduction	26
2.2 Literature survey	26
2.3 Definition of optimization problems	27
2.4 Optimality conditions	28

2.5 Classification of optimization methods 30

2.6 Unconstrained optimization methods 30

2.6.1 Zero-order methods 32

2.6.2 First-order methods 32

2.6.3 Second-order methods 36

2.6.4 Choice of the method 38

2.6.5 Steplength procedures 39

I) Unconstrained function of one variable 40

II) Constrained function of one variable 42

2.7 Termination criteria 42

2.7.1 Termination criteria for unconstrained optimization 43

2.8 Constrained optimization : Linear and nonlinear programming 45

2.8.1 Linear programming 45

2.8.1.1 Simplex method 47

2.8.2 Nonlinear programming 48

2.8.2.1 Sequential unconstrained minimization techniques (SUMT) 49

2.8.2.2 Sequential linear programming (SLP) 55

2.8.2.3 Feasible direction method (FDM) 56

2.9 Scaling 62

I) Scaling of the constraints 62

II) Scaling of the objective function 63

III) Normalization of search direction 63

CHAPTER 3 THEORY OF THE BOUNDARY INTEGRAL EQUATION METHOD

3.1 Introduction 66

3.2 Formulation of the boundary integral equation 68

3.2.1 Governing equation of elasticity 68

3.2.2 The fundamental solution in two dimensions 70

3.2.3 Betti's reciprocal work theorem 73

3.2.4 Integral equation for displacements and stresses at interior points 74

3.2.5 The boundary integral equation 76

3.3 The numerical solution of the integral equations 77

3.3.1 Introduction 77

3.3.2 Discretization of the boundary 77

3.3.3 Discretization of the integral equations and construction of the system of equations 78

3.3.4 Surface stresses 79

3.4 Weight calculation by the BIE method 81

CHAPTER 4 SHAPE DESIGN SENSITIVITY ANALYSIS USING THE BOUNDARY ELEMENT METHOD

4.1 Introduction 86

4.2 Analytical differentiation of the structural response with respect to the design variables 88

4.2.1 Derivatives of displacement 88

4.2.2 Derivatives of jacobian of transformation 92

4.2.3 Derivatives of Kernel products 93

4.2.4 Derivatives of stresses 94

4.3 Analytic differentiation of weight with respect to the design variables 95

4.4 Comparing derivatives calculated by finite difference and direct analytic differentiation 97

CHAPTER 5 SHAPE OPTIMIZATION

5.1 Introduction 107

5.2 Literature survey 107

5.3 Problem statement 110

5.4 Selection of objective function 110

5.5 Shape representation 112

5.5.1 Cubic and Hermitian cubic splines 113

5.5.2 Advantages of Hermitian cubic spline 118

5.6 Choice of the design variables 119

5.7 Boundary element mesh generation 121

5.7.1 Boundary element mesh redefinition 121

5.8 Optimization methods for structural shape optimization 122

I) Sequential Linear Programming

II) Penalty Function Methods

III) Feasible Direction Method

IV) Calculus of Variations and Optimality Criteria Methods

V) Pattern Transformation Method

VI) Photo-Elasticity Technique

5.9 Optimization of shape to minimize stress peaks 125

5.10 Weight minimization 127

CHAPTER 6 NUMERICAL RESULTS

6.1 Introduction 135

6.2 Smoothing stress peaks

6.2.1 Optimum shape of a hole in an infinite plate 136

6.2.2 Optimum profile design for a connecting rod 138

6.2.3 Tank with internal pressure 139

6.2.4 Cantilever beam 140

6.3 Minimum weight design

6.3.1 Cantilever beam under lateral load 141

6.3.2 Plate under biaxial stresses 141

6.3.3 Torque arm 142

CHAPTER 7 CONCLUSIONS 172

APPENDICES

A: Linear programming : Simplex method 178

B: Two dimensional quadratic boundary element mesh generation of a curve defined with specified continuous function 179

REFERENCES 181

NOMENCLATURE

The notation used in the main text and appendices are briefly described in alphabetical order in the following list. Where there are two or more definitions for the same symbol, the relevant chapter or section is indicated in parentheses. When no such indication is provided, the definition for the symbol applies everywhere other than for the alternative definitions. Where appropriate, the number of the equation in which the symbol first appears is given.

Symbol	Definition	Equation
A	Area of a component in two-dimensional case	(3.61)
A	Matrix A contains the coefficients of the variables in equality constraint equations (a_{ji} : j=1,m ; i=1,n)	(2.47)
$A(x,\lambda,r_p)$	Augmented Lagrangian function	(2.68)
b (b_j : j=1,m)	Vector of constants in linear equality constraint equations	(2.47)
c (c_i : i=1,n)	Vector contains the cost coefficients	(2.46)
c	Boundary of the closed bounded region R	(3.58)
C (>1)	Predefined number which increases the multiplier r_p after each unconstrained minimization in (SUMT)	(2.61)
C' (<1)	Predefined number which decreases the multiplier r'_p after each unconstrained minimization in (SUMT)	(2.61)
C_{ij}	Limiting value of the surface integral $T_{ij}(P,Q)$; i=1,2 ; j=1,2	(3.40)

d(b,c)	Number of the cth node of the bth surface element ; c=1,3 ; b=1,M	(3.51)
D^q	Symmetric update matrix in variable metric methods at iteration q	(2.18)
$D_{ijk}(P,Q)$	Third order tensor for obtaining stresses at P($\underset{\sim}{x}$) ; i=1,2 ; j=1,2 ; k=1,2	(3.36)
e_i	Unit vectors in the x_i th direction	(3.18)
E	Young's modulus	(3.11)
E'	Material property = E for plane strain = E (1-υ'^2) for plane stress	(3.11)
F(X)	Objective function	(2.1)
F_1,F_2	Function values of intermediate points in Golden Section Method such that $F_1<F_2$	(2.6.5)
$g_i(\underset{\sim}{x})$	Prescribed boundary displacement function ; i=1,2	(3.13)
$g_j(X)$	Inequality constraints ; j=1,m	(2.2)
G	Shear modulus [$\equiv\mu$=E/2(1+υ)]	
G_i	Galerkin vector ; i=1,2	(3.16)
$h_i(\underset{\sim}{x})$	Prescribed boundary traction function ; i=1,2	(3.14)
$h_k(X)$	Equality constraints; k=1,l	(2.3)
H(X)	Hessian matrix	(2.6)
$H_i(b,c)$	The gradients ($dx_i/d\zeta$) of the cth node of the bth Hermitian cubic element ; c=1,2 ; b=1,n	(5.15)
I	Identity matrix	(2.23)
$J_i(\zeta)$	Jacobian of transformation from global cartesian coordinates to intrinsic coordinate of element ; i=1,2	(3.49)

$K_j(j=1,n)$	Slopes of the cubic spline at the end of each cubic curve	(5.10)
l	Number of equality constraints	(2.3)
l_k	The gradients of $(dx_1/d\zeta)$ of kth point of the Hermitian cubic spline (Chapter 5)	
L	Length of a curve which is defined by a continuous differentiable function	(B.2)
m	Number of inequality constraints	(2.2)
m_k	The gradients of $(dx_2/d\zeta)$ of kth point of the Hermitian cubic spline (Chapter 5)	
$m_{\beta j}$	Unit vectors in the local coordinate directions, ζ_β ; $\beta=1,2$; $j=1,2$	(3.53)
M	Total number of surface quadratic elements	(3.51)
$M^c(\zeta)$	Cubic shape functions of intrinsic coordinate ζ corresponding to the cth node of the Hermitian cubic element ; c=1,2	(5.11)
n	Number of design variables	(2.4)
n	Number of cubic curves on a Hermitian cubic spline (Chapter 5)	
n_j	Direction cosines of the unit outward normal vector to the surface of the elastic body; j=1,2	(3.6)
N	Number of iterations in Golden Section Method	(2.37)
$N^c(\zeta)$	Cubic shape functions of intrinsic coordinate ζ corresponding to the cth node of the Hermitian cubic element ; c=1,2	(5.12)
$N^c(\zeta)$	Quadratic shape function corresponding to the	(3.43)

	c th node of the quadratic element ; c=1,3	
$P(\underline{x})$	Load point within elastic domain, R (Chapter 3)	
$P(\underline{x})$	Load point at the surface of the elastic domain, R (Chapter 3)	
P(X)	Imposed penalty function on constraints in (SUMT)	(2.56)
P^a	ath node at the surface of solution domain ;a=1,q	(3.51)
q	Iteration number	(2.13)
q	Total number of distinct nodal points on the surface of the solution domain (Chapter 3)	
$q(\underline{x})$	Field point within an elastic domain, R (Chapter 3)	
Q	Large positive number added to design variables make them unrestricted in sign	(2.51)
$Q(\underline{x})$	Field point at the surface of an elastic domain, R (Chapter 3)	
r_p	Multiplier determines the magnitude of penalty function term and is held constant for a complete unconstrained minimization	(2.56)
r_p	Multiplier of equality constraints determines the magnitude of penalty term and is held constant for a complete unconstrained minimization	(2.58)
r'_p	Multiplier of inequality constraints determines the magnitude of penalty term and is held constant for a complete unconstrained minimization	(2.57)
r(p,q)	Distance between the field and load points	(3.18)
R	Elastic domain (Chapter 3)	
S	Surface or boundary of elastic domain (Chapter 3)	
S_b	bth element of the discretized surface; b=1,M (Chapter 3)	
S_b	bth element of the Hermitian cubic spline ; b=1,n	

(Chapter 5)

S^q	Search direction at iteration q	(2.13)
S_t	The part of the surface of the elastic domain where tractions are prescribed	(3.14)
S_u	The part of the surface of the elastic domain where displacements are prescribed	(3.13)
S_ε	Surface of the region of exclusion, R_ε	(3.32)
$S_{ijk}(P,Q)$	Third order tensor for obtaining stresses at $P(\underset{\sim}{x})$; i=1,2 ; j=1,2 ; k=1,2	(3.35)
t_i	Traction vector ; i=1,2	(3.6)
t_i^*	Fundamental traction vector solution to Navier's equations	(3.31)
$\bar{t}_i$	Traction vector in the orthogonal local coordinate system defined by m_{ij} ; i=1,2 ; j=1,2	(3.55)
$T_{ij}(p,q)$	ith component of the traction vector at point $q(\underset{\sim}{x})$ due to a unit point load in the jth direction at $P(\underset{\sim}{x})$; i=1,2 ; j=1,2	(3.25)
		(3.25)
u_i	Displacement vector ; i=1,2	(3.4)
u_i^*	Fundamental displacement vector solution to Navier's equations	(3.31)
$\bar{u}_j$	Displacement vector in the orthogonal local coordinate system defined by m_{ij} ; i=1,2 ; j=1,2	(3.53)
U	Vector of Lagrange multipliers	(2.97)
$U_{ij}(p,q)$	ith component of the displacement vector at point $q(\underset{\sim}{x})$ due to a unit point load in the jth direction at $P(\underset{\sim}{x})$; i=1,2 ; j=1,2	(3.24)

x_i	Rectangular cartesian coordinates ; i=1,2	(3.19)
x^c_i	Cartesian coordinates of the cth node of the quadratic element ; c=1,3 ; i=1,2	(3.43)
$x_i(b,c)$	Cartesian coordinates of the cth node of the bth Hermitian cubic element ; c=1,2 ; b=1,n	(5.15)
$x_k(k=1,2)$	Coordinates of the movable design variable node	(4.2)
X	$X_i=[X_1,X_2,.....,X_n]^T$; design variable vector	(2.1)
X*	Optimum solution (Chapter 2)	
X'	Slack variable vector to convert inequality constraint equations to equality constraint equations	(2.50)
X''	Unrestricted in sign design variables	(2.51)
X_1,X_2	Intermediate points in Golden Section Method such that $X_1<X_2$	(2.34)
X_L,X_U	Lower and upper bounds on X in Golden Section Method	(2.34)
X_i	Body force vector ; i=1,2	(3.1)
X_i^L,X_i^U	Lower and upper side constraints	(2.4)
W	Weight of a component in two-dimensional case with uniform unit thickness	(3.69)

Greek Characters

α	Scalar multiplier in steplength procedure	(2.30)
α^*	Optimum scalar multiplier in steplength procedure	(2.32)

α_q	Scalar multiplier in steplength procedure at iteration q	(2.13)
β	Scalar quantity ensures that the search direction will point into the feasible zone	(2.82)
β_q	Scalar multiplier in conjugate direction method at iteration q	(2.15)
δ_{ij}	Kronecker delta = 0 when i≠j = 1 when i=j	(3.2)
ΔX	Some predefined magnitude for absolute tolerance between intermediate points in Golden Section Method	(2.36)
ε	Some predefined magnitude for relative tolerance between intermediate points in Golden Section Method	(2.36)
ε	Small negative number which marks transition from the interior penalty to the extended penalty function	(2.62)
ε	radius of circular region of exclusion, R_ε (Chapter 3)	
ε_A	Some given small number for checking the convergence of the optimization process for absolute change of objective function between two successive iterations	(2.41)
ε_R	Some given small number for checking the convergence of the optimization process for relative change of objective function between two successive iterations	(2.42)
ε_{ij}	Strain tensor ; i=1,2 , j=1,2	(3.2)
$\overline{\varepsilon_{ij}}$	Strain tensor in the orthogonal local coordinate system defined by m_{ij} ; i=1,2 ; j=1,2	(3.54)
$\phi(X,r_p)$	Pseudo-objective function in (SUMT)	(2.56)

Φ	Weighting factor determines the relative importance of the objective and the constraints	(2.103)
γ_q	Difference between the gradient vectors of objective function in two successive iterations (q,q+1)	(2.19)
λ	Lame's constant $=[2\mu\upsilon/(1-2\upsilon)]$	
λ_j	Lagrange multipliers ; j=1,m	(2.8)
μ	Shear modulus	(3.2)
υ	Poisson's ratio	(3.2)
υ'	Material property $=\upsilon$ for plane strain $=\upsilon/(1+\upsilon)$ for plane stress	(3.9)
θ_j	Push-off scalar factors which push the design away from the active constraints	(2.83)
θ_0	Nominal value of θ_j	(2.86)
ρ	Density	(3.69)
σ_a	Mean stress	(5.6)
σ_e	Von-Mises stress	(4.33)
σ_q	Difference between design variable vectors in two successive iterations (q,q+1)	(2.19)
σ_θ	Tangential stress	(5.5)
σ_{ij}	Stress tensor ; i=1,2 ; j=1,2	(3.1)
$\overline{\sigma_{ij}}$	Stress tensor in the orthogonal local coordinate system defined by m_{ij} ; i=1,2 ; j=1,2	(3.55)
$\sigma_1,\sigma_2,\sigma_3$	Principal stresses (Chapter 4)	
τ=0.38197	Predefined factor in Golden Section Method	(2.34)
ψ	Scalar Lagrange multiplier	(2.99)

ζ	Intrinsic coordinates of isoparametric quadrilateral element ; $\zeta\varepsilon[-1,1]$	(3.43)
ζ	Intrinsic coordintes of Hemitian cubic element ; $\zeta\varepsilon[-1,1]$	(5.11)
ζ_i	Coordinates of the load point ; i=1,2	(3.19)

LIST OF ABBREVIATIONS

AN	Analytic
BEM	Boundary Element Method
BFGS	Broydon-Fletcher-Goldfarb-Shanno
BIE	Boundary Integral Equation
CAD	Computer Aided Design
DFP	Davidon-Fletcher-Powell
FD	Finite Difference
FDM	Feasible Direction Method
FE	Finite Element
I-DEAS	Integrated Design Engineering Analysis Software
LP	Linear Programming
NLP	Nonlinear Programming
SLP	Sequential Linear Programming
SUMT	Sequential Unconstrained Minimization Technique

CHAPTER 1

INTRODUCTION

The job of engineering is to put together useful systems by applying knowledge and methods derived from the pure sciences such as physics, chemistry, biology, etc.. The engineering design function is to develop the best possible system for the given application, consistent with the resources allocated to the project. It is technical problem of optimizing designs toward which this text is directed.

In recent years, the simultaneous advance and maturing of different disciplines has provided an opportunity to create new tools for engineering design, especially structural design. A dominant influence in making design optimization possible is the high-speed digital computer. The use of computers for engineering analysis or design is commonplace in even the smallest design office.

In the past century, outstanding advances in engineering analysis have been made. The computer has allowed the engineer to quantitatively analyze the behaviour of systems that were examined only qualitatively in the past. The mechanical and structural sciences have been improved significantly from this analysis capability.

Finite element analysis has rapidly established itself as an important engineering technique in the last two decades or so for carring out stress analysis on many type of structures. The method divides the whole body under consideration into a number of discrete elements and then characterizes the behaviour of the body by an overall matrix which is constructed from the summation of the behaviour of the individual elements. It can be used for problems with material and geometric non-linearities because of the nature of its formulation, although the method has its drawbacks. The discretization of all the domain is needed which implies difficulty of mesh data preparation, especially for three-dimensional problems. The finite element method provides a large volume of unnecessary data concerning internal nodal points and elements and often is never used. The system of equations which needs to be solved is often very large, even though the

symmetric and banded nature of the coefficient matrix helps to reduce the computational effort. In stress analysis when the problem concerned involves rapidly varying stress fields of the type encountered near stress concentrations and cracks, the FEM is inefficient.

In the past ten years or so, the Boundary Integral Equation (BIE) method has become increasingly successful in many branches of engineering sciences. It is replacing the FE technique for a wide range of applications in stress analysis and is particularly well established in linear elastic problems. The method instead of attempting to find an approximate solution for the governing differential equation throughout the solution domain, converts the equation into an integral form, often involving only integrals over the boundary of the domain. Consequently, the dimensionality of the problem is thereby effectively reduced by one, so only the boundary has to be discretized in order to carry out the integrations. In comparison with the finite element method, the mesh data preparation becomes greatly simplified and is more economical of computing time and storage. The boundary element approach offers continuous interior modelling within the solution domain of the problem, giving high resolution of stresses and displacements. This is particularly useful for stress concentration and fracture mechanics problems. The BIE method is not without its drawbacks. Its extension to non-linear applications has limitations. This is because the nature of the non-linearities no longer allows the partial differential equations to be fully transformable to the surface of the body. Consequently, the integral constraint equation is not strictly of the boundary-type due to the presence of domain integrals over the regions with non-linear behaviour. It is unsuitable if information is required at a large number of internal points. It is also unsuitable for problems having small surface-to-volume ratios.

In addition to advances in computer technology and analysis capability a third discipline has improved dramatically in recent years. This discipline is referred to as numerical optimization. The concept of this methodology is rather new and virtually unknown to many practicing engineers. This is unfortunate because the techniques can provide significant reductions in design time, and are a logical progression from computer analysis. The main attraction of these techniques is that they provide a general, computer-based, design tool for solution of a wide variety of design tasks.

It has not been until the last decades that the improvement of mechanical design has changed from an intuition and experience basis to a general and rigorous application of the concepts of the theory of optimization. The use of numerical optimization techniques is to devise a rational, directed design procedure. These methods provide a computational tool for design and indeed numerical optimization provides us with a new design philosophy. The designs will not be based on intuition and experience as before. It does not mean that the design process can be reduced to a few computer runs and there is no need for experience and intuition. But it prevents repetitive calculations and gives more time to the designer for thinking about the aspects of engineering design.

One of the subjects with a most important development nowadays concerning the general topic of mechanical and structural optimization is, no doubt, the shape optimization problem. The mechanical and structural optimization has generally always been concerned with varying only the transverse size (cross-sectional area or thickness), namely design variables, of bars, membranes, beams, shells, plates, etc. while the shape of the structure remains unchanged. There is an important class of structural design problems in which the shape of a two- or three-dimensional structural component is to be determined, subject to constraints on displacements, stresses and natural frequencies in the structure. For such problems, it is the shape of the physical domain of the structural component that must be treated as the design variable. Of course, shape optimal design, in which the whole geometry changes during the process, has more difficulties, such as how to define general shapes which meet a large number of geometrical requirements, how to impose realistic constraints on the manufacturing feasibility, etc. However, for many problems shape design is more effective than sizing design. A typical example is that of a stress concentration at a hole boundary in a panel. Structural optimization with sizing design variable would increase or decrease the thickness of the panel, while shape optimization would change the shape of the hole boundary. The effect of shape changes can improve the structural performance, so the shape optimization needs sophisticated structural analysis and optimization tools.

The most important part of the shape optimal design involves the shape design sensitivity analysis. Structural design sensitivity analysis concerns the relationship between design variables available to the engineer and structural response that are determined by the laws of mechanics. This measures the rates of change of

displacements, stresses and natural frequency with respect to the change of design variables which in shape optimization define the shape of the boundary and are implicitly defined through the state equations of structural mechanics.

Two different approaches have been used in recent years in order to solve shape optimization problems. The first method uses the variational method. Optimality criteria are derived mathematically and according to these relations the optimum shape will be determined. The material derivative of continuum mechanics is used to predict the effect of a change in shape on functionals that define structural response. The adjoint variable method is used to develop a formulation for representing response variations in terms of variations in design. Usually the finite element method is used for the numerical analysis. This method of approach is not considered here.

The second one couples numerical optimization methods with numerical analysis techniques like the finite element method, and the sensitivity analysis is carried out by implicit differentiation of the discretized finite element system with respect to design variables, and the derivatives are calculated by a finite difference scheme.

The boundary element method requires only the modelling of the boundary and seems to be ideally suited for shape optimization problems. Using the finite element method in shape optimization has two major drawbacks. First, finite element mesh regeneration during the optimization process is very costly and time-consuming while only the boundary of the geometry is changing and the design variables are restricted to the boundary. Second, the computation of the derivatives in the sensitivity analysis, an essential step in the optimization process, which usually has been carried out by using finite difference scheme applied to the results of finite element analysis, is not accurate and is very costly. The BIE method can overcome these problems. However, the BIE analysis involves only information on the boundary, but it should be mentioned that in the vast majority of problems, the maximum stresses occur on the boundary [Fenner, 1989]. Although experience shows that this is very often the case, known exceptions are when contact boundary conditions are applied, and when certain types of body forces exist.

In this thesis, a new approach to shape optimization by using the BIE method for

two-dimensional elastic problems is introduced. Chapter 2 provides a brief discussion of the basic principles of numerical optimization methods, especially those which are more applicable to the design optimization. Unconstrained and constrained optimization methods are reviewed and the choice of the method used in the programs of this thesis is discussed and the appropriate algorithms are presented. This is followed, in chapter 3, by a description of the analytical basis of the BIE method and numerical formulations.

Chapter 4 describes a differentiated form of the boundary integral equation in which the derivatives of displacements and stresses can be calculated analytically for design sensitivity analysis. Then in a test case, these derivatives are compared with the derivatives obtained by using a finite difference scheme applied to the boundary element analysis results.

In chapter 5, different steps in the shape optimization process, such as model description, selection of the objective function and shape variables, representation of boundary shape, boundary element mesh regeneration and solution methods are discussed. The details of two computer programs which are developed for stress and weight minimization are described.

The accuracy of the computer code is assessed in chapter 6 by applying them to some test cases for which exact solutions exit or to examples which have been solved by finite element method. For the stress minimization four test cases are as follows: optimum shape of a hole in an infinite plate under biaxial stresses; optimal profile design for a connecting rod; tank with internal pressure and a cantilever beam. For the weight minimization three test cases are as follows: cantilever beam under lateral load; plate with a hole under biaxial stresses and a torque arm.

CHAPTER 2

THEORY OF OPTIMIZATION

2.1 - INTRODUCTION

Optimization always has been a goal in the field of human activities. It takes different forms in different situations. Here the field of engineering is concerned. Engineering designers are often interested in achieving optimal solutions to their problems. Different designs, all satisfying the specified conditions, are compared, and one is chosen that is also the best in terms of an optimality criterion. Optimization techniques, if properly applied, will examine the cost or profit associated with a large number of different designs and select an optimum.

Classical optimization methods of differential calculus and calculus of variations can be applied to certain optimization problems in the physical or mathematical sciences. There are , however, many problems for which classical methods are not suitable. Use of numerical optimization techniques is the way to devise a rational, directed design procedure. These are iterative techniques that provide a computational tool for design. They are attractive because the result is not based on experiments or intuition. This chapter deals with these numerical optimization methods which are applicable to engineering design.

2.2 - LITERATURE SURVEY

The early works on optimization methods belong to Newton, Lagrange and Cauchy. Newton and Leibnitz developed differential calculus methods of optimization.

The foundations of calculus of variations were laid by Bernoulli, Euler, Lagrange and Weirstrass. The first idea of using multipliers for the constrained optimization was due to Lagrange [Rao, 1978]. Cauchy for the first time used steepest descent method for unconstrained optimization problems [Rao, 1978]. In spite of these early contributions, very little progress was made until the advent of the computer as a powerful tool for implementation of the optimization procedures and then began the application of optimization theory to new areas.

Since 1968 the volume of research in using numerical methods of optimization has increased very rapidly, but there are some significant works published before that time. For example; the development of the Simplex method by Dantzig in 1947 for linear programming, the principle of optimality by Bellman in 1957 for dynamic programming and the necessary and sufficient conditions for optimality by Kuhn and Tucker in 1951 for non-linear programming.

2.3 - DEFINITION OF OPTIMIZATION PROBLEMS

A general optimization problem, sometimes referred to as a mathematical or nonlinear programming problem, can be formulated as follows:

Minimize		$F(X)$	Objective Function	(2.1)
Subject to	:	$g_j(X) \leq 0 \quad j=1,m$	Inequality Constraints	(2.2)
		$h_K(X)=0 \quad k=1,l$	Equality Constraints	(2.3)
		$X_i^L \leq X_i \leq X_i^U$	Side Constraints	(2.4)
Where		$X^T=[X_1, X_2, \ldots\ldots, X_n]$	Design Variables	

The vector X is referred to as the vector of design variables. The objective function F(X) and constraint functions, Eqs.(2.2) and (2.3), may be linear or nonlinear

functions of the design variables X. These functions may be explicit or implicit in X and may be evaluated by any analytical or numerical methods. It is important that these functions be continuous and have continuous first derivatives in X. Sometimes side constraints Eq.(2.4) are included in the inequality constraints. The problem of maximizing a given function can be transformed into a minimization problem simply by reversing the sign of the function.

2.4 - OPTIMALITY CONDITIONS

For practical problems in design, it may be impossible to have a single (absolute) optimum design, because sometimes multiple solutions to the optimization problem exist. The best approach is usually to start the optimization process from several initial vectors and if at the end the results are the same, it is the global minimum. Another possibility is to define necessary conditions for an optimum and if they are satisfied, the problem has at least a relative minimum.

Consider an unconstrained problem, where F(X) should be minimized with no constraints imposed. The necessary condition for it to be a minimum is that the gradient of F(X) must vanish

$$\nabla F(X)=\begin{bmatrix} \frac{\partial F(X)}{\partial X_1} \\ \frac{\partial F(X)}{\partial X_2} \\ \vdots \\ \frac{\partial F(X)}{\partial X_n} \end{bmatrix} = 0 \qquad (2.5)$$

and the sufficient condition is that the Hessian matrix, matrix of second partial derivatives with respect to the design variables, be positive definite [Vanderplaats,

1984]

$$H=\begin{bmatrix} \frac{\partial^2 F(X)}{\partial X_1^2} & \frac{\partial^2 F(X)}{\partial X_1 \partial X_2} & \cdots & \frac{\partial^2 F(X)}{\partial X_1 \partial X_n} \\ \frac{\partial^2 F(X)}{\partial X_2 \partial X_1} & \frac{\partial^2 F(X)}{\partial X_2^2} & \cdots & \frac{\partial^2 F(X)}{\partial X_2 \partial X_n} \\ \frac{\partial^2 F(X)}{\partial X_n \partial X_1} & \frac{\partial^2 F(X)}{\partial X_n \partial X_2} & \cdots & \frac{\partial^2 F(X)}{\partial X_n^2} \end{bmatrix} \qquad (2.6)$$

Positive definiteness means that this matrix has all positive eigenvalues. These two conditions indicate that the solution is a relative minimum (local minimum) for a given X. But sufficient condition for being minimum is that the Hessian matrix should be positive definite for all possible values of the design variables X.

For a constrained optimization, since the constraints may be active (being near the allowable range) at the optimum design, it is no longer necessary that the gradient of the objective function vanish at the optimum. Assume a design is specified. In order to improve on this design, it will be necessary to determine a direction vector S (search direction) which will rapidly reduce the objective function (usable direction) and yet not violate the active constraints and maintaining a feasible design (feasible direction). It means that the search direction should be usable and feasible.

An important advance in optimization theory took place in 1951, when Kuhn and Tucker extended the classical method of Lagrange Multipliers to constrained optimization problems. The Kuhn-Tucker necessary conditions state that for an optimum design, X*, the following three conditions must be satisfied:

$$1)\ X^* \text{ is feasible} \qquad (2.7)$$

$$2)\ \lambda_j g_j(X^*)=0 \qquad j=1,m\ ,\ \lambda_j \geq 0 \qquad (2.8)$$

$$3)\ \nabla F(X^*) + \sum_{j=1}^{m} \lambda_j \nabla g_j(X^*) + \sum_{k=1}^{L} \lambda_{k+m} \nabla h_k(X^*) = 0 \qquad (2.9)$$

$$4)\ \lambda_j \geq 0 \qquad (2.10)$$

$$5)\ \lambda_{k+m} \quad \text{Unrestricted in sign} \qquad (2.11)$$

where $(\lambda_1, \lambda_2, \ldots\ldots, \lambda_m)$ are Lagrange Multipliers. For more detail refer to [Rao, 1978] and [Vanderplaats, 1984]. The first one is obvious, because the optimum design must satisfy all constraints. The second one indicates that, if the constraint is not satisfied then the corresponding Lagrange Multiplier must be zero. The third equation imposes the requirement of usability and feasibility of the design. It should be mentioned that the Kuhn-Tucker conditions are necessary conditions to be satisfied at a relative minimum of F(X). For convex problems, when the Hessian matrix of the objective function is positive definite for all possible designs, the Kuhn-Tucker conditions are necessary and sufficient for a global minimum. For more details about Kuhn-Tucker conditions refer to [Zangwill, 1969].

2.5 - CLASSIFICATION OF OPTIMIZATION METHODS

Optimization problems have been classified based on the existence of constraints as unconstrained or constrained ones, depending upon whether constraints exist or not in the problem.

2.6 - UNCONSTRAINED OPTIMIZATION METHODS

The special characteristic of these methods is that the solution vector X does not need to satisfy any constraint. This problem can be considered as a particular case of the general constrained optimization problem and can be stated mathematically as

follows:

$$\text{Minimize} \qquad F(X) \tag{2.12}$$

$$\text{Where} \quad X=(X_1, X_2, \ldots\ldots, X_n)^T$$

The unconstrained minimization technique is important because there are some design problems that can be treated as unconstrained minimization problems and most of the powerful and convenient methods of solving constrained minimization problems involve the transformation of the problem into one of unconstrained minimization, which will be discussed later.

Most methods of unconstrained minimization have been treated iteratively. Probably the most common form of this iterative procedure is given by

$$X^q = X^{q-1} + \alpha_q \, . \, S^q \tag{2.13}$$

where q is the iteration number, X is the vector of design variables and S^q is the vector of search direction and α_q is a scalar multiplier determining the amount of change in X for this iteration and is called the *steplength.* There are two major steps in the use of Eq.(2.13), first determination of the search direction S^q and second is interpolation for the scalar multiplier which minimizes F(X) as much as possible in direction S^q and will be discussed later.

The distinctions among methods arise from the need to use varying levels of information about derivatives of F(X) in defining the search direction. There are three basic approaches to unconstrained problems which may be extended to other problem categories and they are as follows:

I) Zero-order methods
II) First-order methods
III) Second-order methods

2.6.1 - Zero-Order Methods

Zero-order methods are often termed direct search techniques. These methods rely only on evaluating F(X) at a sequence of points and comparing values in order to reach the optimal point X^*. They do not require the evaluation of function derivatives. These are used when the function F(X) is not differentiable or the derivatives are discontinuous or their evaluation is much more difficult than the evaluation of the function F(X) itself. The problem with them is that often they require thousands of function evaluations to achieve the optimum even for a simple problem, therefore they are not useful when the function evaluation is expensive.

There is a variety of algorithms in this category; for example Powell's method [1964], Hook and Jeeves method [1961] and Rosenbrock[1960].

2.6.2 - First-Order Methods

First-order methods or gradient methods require the evaluation of function derivatives which makes them more reliable and efficient [Walsh, 1975]. There are three major groups in this category:

I) Steepest descent method
II) Conjugate direction methods
III) Variable metric methods

The*Steepest descent method* which is the best known method but usually is not recommended. The search direction S is the negative of the derivative of the objective function

$$S^q = - \nabla F(X^q) \qquad (2.14)$$

Since this method does not use information from previous iterations in order to accelerate the convergence, the convergence rate is very poor but in more powerful methods steepest descent is very good for the initial search direction.

The *Conjugate direction method* may be said to be a modified form of steepest descent method. Let G be a positive definite symmetric matrix. Then the directions represented by the vectors $P \neq 0$ and $Q \neq 0$ are said to be conjugate with respect to G, if $P'GP \neq 0$ [Walsh, 1975].

In 1964, Fletcher and Reeves derived a simple recurrence formula which produces a sequence of conjugate directions. This is accomplished by specifying an initial search vector as the steepest descent defined by Eq.(2.14). On subsequent iterations a conjugate direction is defined as

$$S^q = -\nabla F(X^q) + \beta_q S^{q-1} \tag{2.15}$$

where the scalar β_q is defined as

$$\beta_q = \frac{\left[\text{Abs}\left[\nabla F(X^q)\right]\right]^2}{\left[\text{Abs}\left[\nabla F(X^{q-1})\right]\right]^2} \tag{2.16}$$

Since this method uses some information about the previous iterations by scalar β_q, it dramatically improves the convergence of the optimization process.

Variable metric methods or *Quasi-Newton methods* approximate the inverse of the Hessian matrix by a similar matrix which is modified at each iteration to include information obtained about the curvature of F along the latest direction. Probably, the most popular of the variable metric methods is Davidon-Fletcher-Powell (DFP) method. It was originally devised by Davidon [1959], and was later improved by Fletcher and Powell [1963]. In these methods, the search direction at iteration q is defined as

$$S^q=-H.\nabla F(X^q) \tag{2.17}$$

where H is not the Hessian matrix but it approaches the inverse of the Hessian matrix for quadratic functions. These methods have convergence characteristics similar to second-order methods because an approximation to the inverse of Hessian matrix is created during the process. At the initial design point, the H matrix is taken as the identity matrix H=I, therefore the initial step is the steepest descent. At the end of iteration q, a new matrix is updated by

$$H^{q+1}=H^q+D^q \tag{2.18}$$

for DFP method

$$D^q=A^q-B^q \tag{2.19}$$

Where $A^q=\dfrac{\sigma_q.\sigma_q^T}{\sigma_q^T.\gamma_q}$ and $B^q=\dfrac{H_q.\gamma_q.\gamma_q^T.H_q}{\gamma_q^T.H_q.\gamma_q}$

and $\sigma_q=X_{q+1}-X_q$, $\gamma_q=\nabla F(X^{q+1})-\nabla F(X^q)$

which replaces the inverse of the Hessian matrix by an approximate H_q. The method of computing this metric completely eliminates the need for evaluating second-derivatives and performing matrix inversions, and the sequence of iterations converges quadratically to the minimum point X^*. It should be noted that at the end of each iteration the one-dimensional search is performed according to Eq.(2.13).

Another variable metric method is the Broydon-Fletcher-Goldfarb-Shanno (BFGS) which sometimes is called the complementary DFP formula. It is an alternative formula for updating H_q and was discovered independently and in different ways by Broydon [1970], Fletcher [1970], Goldfarb [1970] and Shanno [1970]. The updating matrix is as follows

$$H_{q+1}=H_q+\frac{1}{\gamma_q^T\sigma_q}(\beta_q\sigma_q\sigma_q^T-\sigma_q\gamma_q^TH_q-H_q\gamma_q\sigma_q^T) \qquad (2.20)$$

where $\beta_q=1+\dfrac{\gamma_q^T H_q\,\gamma_q}{\gamma_q^T\sigma_q}$

The BFGS method appears to be superior to the DFP method because it avoids the tendency which is present in the DFP method for the matrices H_q to become singular [Murray, 1972].

The advantage of the variable metric methods is that it converges quadratically and possessing very good stability. In highly distorted and eccentric functions, it continues to progress, because at each iteration the Hessian matrix is modified and it contains the information about the previous iterations. It can minimize a quadratic function in N iterations and there is no need for second-derivative calculations. Fig.(2.1) shows the algorithm for the variable metric methods.

In the method described in this thesis for the shape optimization (Sec.5.9), the BFGS method has been used for the updating formula. For a better global convergence, some modifications have been made to the original method which are as follows [Skrobanski, 1986].

Let X^q denote the starting point of the qth iteration. Let H_q be the current positive definite variable metric approximation to the inverse Hessian. Let S^q denote the usual variable-metric search direction, i.e.

$$S^q = -H^q\,\nabla F(X^q) \qquad (2.21)$$

Let $\varepsilon>0$ be some given tolerance. It is assumed that the algorithm is terminated at X^q if $\|\nabla F(x^q)\|_2<\varepsilon$. So suppose $\|\nabla F(x^q)\|_2\geq\varepsilon$.The vector S^q is defined by

$$S^q = \begin{bmatrix} S^q & \text{If } -(S^q)^T \nabla F(X^q) > Z_{Tol} \\ -\nabla F(X^q) & \text{Otherwise} \end{bmatrix} \qquad (2.22)$$

where Z_{Tol} is any given tolerance satisfying $0 \leq Z_{tol} \leq \varepsilon^2$.

At the end of each successful one-dimensional search along the search direction S^q, the matrix H_q is updated to H_{q+1}, where

$$H_{q+1} = \begin{bmatrix} H_{BFGS} & \text{If } \sigma^T \gamma > h_{Tol} \\ I & \text{Otherwise} \end{bmatrix} \qquad (2.23)$$

where I denotes the identity matrix of appropriate order and h_{Tol} is a small positive number say 10^{-6}.

2.6.3 - Second-Order Methods

These are also called Newton-type methods. They use the Hessian matrix (H), or a finite difference approximation to H, to define the search direction. Newton-type methods are the most powerful methods available for general problems and will find the minimum of a quadratic function in one iteration.

If a function is extended by second-order Taylor series at iteration q, then

$$F(X) \approx F(X^q) + \nabla(F(X^q))\, \delta X + \frac{1}{2} \delta X\, H(X^q)\, \delta X \qquad (2.24)$$

where $\delta X = X^{q+1} - X^q$

$H(X^q)$ is the Hessian matrix of F(X). As mentioned in Sec. (2.4), the solution of the unconstrained minimization problem requires finding the vector X^* such that $\nabla F(X^*)=0$. If X^q is the qth value of X during the iteration then

$$\nabla F(X^{q+1}) \approx \nabla F(X^q) + \delta X\, H(X^q) \tag{2.25}$$

If X^{q+1} is the required optimum, then

$$\nabla F(X^{q+1}) \approx 0 \tag{2.26}$$

and Eq.(2.25) gives

$$\delta X = - H^{-1}(X^q)\, \nabla F(X^q) \tag{2.27}$$

or

$$X^{q+1}=X^q - \left[H(X^q)\right]^{-1} \nabla F(X^q) \tag{2.28}$$

Comparing the last term in Eq.(2.28) to the S vector in Eq.(2.13) when $\alpha=1$ then

$$S^q = - \left[H(X^q)\right]^{-1} \nabla F(X^q) \tag{2.29}$$

This process can converge extremely rapidly in many problems and is one of the preferred methods of solving systems of simultaneous nonlinear equations [Fox, 1971]. Indeed, if F is a quadratic, the solution is obtained in a single step, but in some problems it will converge to saddle points and maxima. The principal difficulty in using Newton-type methods is that not only function values and gradient information, but also the second-derivatives must be calculated, while in most engineering problems the calculation of the second derivatives is not easy. Matrix H may be singular, or at least not positive definite as is required to guarantee a solution for a minimum of F(X).

2.6.4 - Choice Of Method

The question of the choice of method begins with whether it is possible to evaluate the derivatives without difficulty or expense.

In zero-order methods there is no need for calculation of derivatives and they are very useful with nonconvex and discontinuous functions. The problem with them is that they need too many function evaluations and they are unsuitable for nonlinear functions when function evaluation is expensive.

First-order methods use the first-derivatives for optimization and it makes them more reliable and efficient than zero-order methods. The method of steepest descent is not recommended for general use because of its poor convergence properties. Conjugate direction methods offer a significant efficiency improvement over steepest descent methods because information about the previous iterations is carried forward in the optimization process, and they do not need large storage, and are suitable for large-sized problem which have a lot of design variables. Variable metric methods are the most popular methods for handling unconstrained minimization problems. They gather information about previous iterations and instead of carring that information through a single scalar parameter like the conjugate direction method, this information is stored in an n dimensional array. The BFGS method is preferable to the DFP method as discussed before.

Second-order methods are the classic methods of optimization. They are the most powerful methods for general problems and will find the minimum of a quadratic function in one iteration. Newton-type methods have serious drawbacks in optimization. For them not only function values and gradient information must be provided, but the second-derivatives, matrix H, as well. The problem with them is the calculation of the inverse of Hessian matrix which sometimes is impossible; the H matrix may be singular or at least not positive definite for a non-quadratic function.

For more details about unconstrained optimization refer to: Fletcher (1980), Fox (1971), Rao (1978), Murray (1972), Walsh (1975), Gill et al (1981), Vanderplaats

(1984) and Jacoby et al (1972).

2.6.5 - Steplength Procedures

It is convenient to describe steplength procedures, Eq.(2.13), in terms of the univariate function defined by

$$F(\alpha) = f(X^q + \alpha S^q) \tag{2.30}$$

Strictly speaking, F also depends on q but this is omitted here since q remains constant throughout the present subsection. Since S^q is a descent direction it can be assumed that

$$F'(0) = (S^q)^T \nabla F(X^q) < 0 \tag{2.31}$$

The ideal choice for the steplength α is given by

$$\alpha^* \approx \arg \underset{\alpha \geq 0}{\text{Min}}\ F(\alpha) \tag{2.32}$$

then the steplength procedure is a minimization of a function with one variable and no constraints.

Sometimes the steplength procedure should be carried out, while there are some constraints upon the variable α; for example in the feasible direction method which will be discussed later, the $F(\alpha)$ in Eq.(2.30) should be minimized while the design is in feasible region and no constraint is violated. This problem is the minimization of a constrained function of one variable.

Here unconstrained and constrained functions of one variable are discussed.

I) Unconstrained Function of One Variable

If the single variable is X then mathematically the problem is

$$\text{Minimize} \quad : \quad F(X) \tag{2.33}$$

while there is no constraint upon the variable. Several methods are available for numerically solving the one-dimensional minimization problem. These can be classified as follows :

I- Interpolation Methods
II- Elimination Methods

Historically *interpolation* has been the foundation of classical numerical analysis. The procedure is to evaluate the function at several points (and possibly the derivatives of the function) and often fit a polynomial to those known points. The minimum of this polynomial is then found and is considered to be a good estimate to the minimum of the true function. The advantage with this method is that, it requires only a few function evaluations. However for highly nonlinear functions the approximation may be quite poor and it can not be used for discontinuous functions.

The *elimination* methods can be used for the minimization of even discontinuous functions. It assumes the function is unimodal, so there is no need for continuous derivatives (by definition a unimodal function is one that has only one peak in a given interval). If a function is known to be a unimodal in a given range, the interval in which the minimum lies can be narrowed down, provided the function values are known at two different points in the range. Probably, the most efficient form of the elimination method is the *Golden Section method.* This method needs a lot of function evaluations but it is easy to program. The rate of convergence is known and is very reliable for poorly conditioned problems. In this method, it is assumed that lower and upper bounds on X, X_L and X_U, respectively and their function values F_L and F_U are known. Therefore the initial interval of uncertainty is X_U-X_L. During the process, this interval reduces to some fraction ε of the initial interval or some specified magnitude. If

X_1 and X_2 are two intermediate points such that $X_1<X_2$ and F_1 and F_2 are function values then the interval of uncertainty is reduced in each iteration by the equations :

$$X_1= (1-\tau) X_L +\tau X_U \tag{2.34}$$

or

$$X_2=\tau X_L +(1-\tau) X_U \tag{2.35}$$

where τ=0.38197 is a predefined factor in Golden Section method [Vanderplaats, 1984]. The bounds X_L and X_U and two interior points X_1 and X_2 are specified, then if $F_1>F_2$, X_L is replaced by X_1 and X_1 is replaced by X_2 and X_2 is calculated by Eq.(2.35) and goes to the next iteration. If $F_1<F_2$, X_U is replaced by X_2 and X_2 is replaced by X_1 and X_1 is updated by Eq. (2.34) and the process begins another iteration. Since the initial interval is being reduced then it is convenient to specify the absolute tolerance Δx but work with a relative tolerance ε. The tolerance may be defined by

$$\varepsilon =\frac{\Delta X}{X_U - X_L} \tag{2.36}$$

where ΔX is some specified magnitude. Thus for a specified tolerance

$$\varepsilon = (1-\tau)^{N-3} \tag{2.37}$$

where N is the total number of function evaluations, and with including the initial three

$$N= -2.078 Ln\, \varepsilon +3 \tag{2.38}$$

so N is the convergence criterion for terminating the iteration process. Sometimes the initial bound is so wide, the process needs a lot of function evaluations for good accuracy, so it is better to shorten this bound before beginning the process. For more details refer to [Vanderplaats, 1984].

II) Constrained Function of One Variable

If the single variable is X, mathematically the problem is

$$\text{Minimize} \quad : \quad F(X) \tag{2.39}$$

$$\text{Subject to} \quad : \quad g_j(X) \leq 0 \qquad j=1,m \tag{2.40}$$

This constrained problem can be solved indirectly by the penalty function method which has been discussed in Sec.(2.8.2.1), or can be solved directly by using Golden Section method.

Two different types may occur, when the design is initially feasible or initially infeasible. Whichever the problem is, the basic approach is the same. First an arbitrary initial step is estimated by using the slope of the function. Next the bounds on the solution should be found and then reduced by using Golden Section method. Finally the last four function values can be interpolated with a cubic polynomial approximation or they may be compared and the best can be picked.

The algorithms of these methods are shown in [Vanderplaats, 1984] for unconstrained and constrained functions of one variable and in the method of this thesis those have been programmed for one-dimensional searches.

2.7 - TERMINATION CRITERIA

Since almost all unconstrained optimization process is carried out iteratively, the termination criterion has a significant effect on the process. Termination criteria are necessary for several reasons. First, to decide when the current estimate for design is acceptable, and to stop the process when the progress is unreasonably slow; for example when an acceptable solution does not exist, or when the iterative process is

oscillating.

Termination criteria should therefore be designed to ensure that an adequate approximation to the solution has been found, to minimize the risk of a false declaration of convergence, and to avoid unwanted effort. There are two natural choices, the desired accuracy of the design variable and the desired accuracy of the objective function.

Here the termination criteria in the case of unconstrained minimization have been introduced. Termination criteria in the case of constrained optimization depends on the method used. Therefore, in the next sections the termination criteria for the constrained optimization are discussed individually.

2.7.1 - Termination Criteria for Unconstrained Optimization

A critical part of the overall process is determining when to stop the search for the optimum. A sensible choice of termination criteria can reduce the amount of computation required by most optimization algorithms.

In practice, the decision concerning whether the iterate X_q (design variable) is an adequate approximation to the solution is based on two tests. Whether X_q almost satisfies the sufficient conditions for optimality, and whether the sequence (X_q) appears to have converged. Unfortunately, for many problems only a few of the optimality conditions can be tested. Therefore, considerable reliance must be placed on a decision that the sequence of iterations is converging.

Perhaps the most important termination criteria for unconstrained minimization are as follows:

1) Maximum number of iterations
2) Absolute or relative change in the objective function

3) The Kuhn-Tucker conditions

Criterion No.1 is that if the number of iterations exceeds a predefined number, the process will stop. This prevents unnecessary computations because of numerical or algorithmic difficulties or simply because of some programming error, and the program will not continue to iterate indefinitely.

Criterion No.2 is for checking the progress and convergence of the process. It includes checking whether the absolute or relative change in the objective function, which depends on the magnitude of the objective function is very large or very small. The absolute change is the difference of F(X) (objective function) between successive iterations. In this case, convergence is identified if

$$\left|F(X^q) - F(X^{q-1})\right| \leq \varepsilon_A \tag{2.41}$$

where ε_A is a specified tolerance. The relative change is the relative difference of F(X) between successive iterations. Here, convergence is identified if

$$\frac{|F(X^q) - F(X^{q-1})|}{\text{Max}\left[|F(X^q)|, 10^{-10}\right]} \leq \varepsilon_R \tag{2.42}$$

where ε_R is a specified fractional change.

Criteria No.3 has been discussed in Sec.(2.4). In the case of unconstrained minimization, this simply requires that the gradient of F(X) vanishes or that each component is sufficiently close to zero or is less in magnitude than a specified constant ε.

2.8 - CONSTRAINED OPTIMIZATION :

LINEAR AND NONLINEAR PROGRAMMING

Constrained optimization deals with the more general design optimization problem where the design variables should satisfy some linear or nonlinear constraints as well as side constraints. This general problem is defined by Eqs.(2.1) to (2.4), which are repeated here for convergence, except now the side constraints are included in the inequality constraint set.

$$\text{Minimize} \quad : \quad F(X) \tag{2.43}$$

$$\text{Subject to} \quad : \quad g_j(X) \leq 0 \qquad j=1,m \tag{2.44}$$

$$h_k(X) = 0 \qquad k=1,l \tag{2.45}$$

There are different methods for solving constrained optimization problems but those which are more applicable in design engineering can be classified into two categories:

I) Linear programming
II) Nonlinear programming

2.8.1 - Linear Programming

There are some optimization problems in which the objective function and constraints are linear functions of the design variables. Linear programming is a good approach for this sort of problem. Most practical problems are not linear, but it is often possible to linearize a complex problem and then solve it using the linear programming method.

The linear programming type of optimization problem was first recognized in the 1930s by economists while developing methods for the optimal allocation of resources. The most common method for the solution of linear programming(LP) is referred to as the *Simplex* method and was introduced and developed by Dantzig in the late 1940s. Afterwards, much progress has been made in the theoretical development and in the practical applications of linear programming.

Two simplex algorithms are available, the standard and revised simplex. The revised technique is a modernized version of standard simplex algorithm [Dantzig, 1963] which uses numerical techniques designed to be especially efficient on the digital computer [Wolfe, 1965].

The standard form of the LP problem is defined by

$$\text{Minimize} : \quad f=c.X \tag{2.46}$$

$$\text{Subject to} : \quad A\,X=b \qquad b_j \ (j=1,m) \tag{2.47}$$

$$X_i \geq 0 \qquad (i=1,n) \tag{2.48}$$

where vector c contains the cost coefficients c_i (i=1,n), X contains design variables X_i (i=1,n), matrix A contains the coefficients of constraint equations and b_j (j=1,m) contains the constants.

This is an equality-constrained problem with non-negative design variables. Since most problems of interest are not of this form, they should be converted to this form in linear programming method. For example, by adding an extra design variable to each inequality constraint, they convert to equality constraints. For instance if the problem requires

$$AX \leq b \tag{2.49}$$

by adding the extra design variable X' we have

$$AX+X'=b' \tag{2.50}$$

These extra design variables are also referred to as *slack variables*. When there is a major restriction with Eq.(2.48), in some problems a large positive number, Q in Eq.(2.51), can be added to each design variable so

$$X=X''-Q \tag{2.51}$$

and this equation can be substituted into Eqs.(2.46) to (2.48).

It is often necessary to convert a LP problem to standard form before solving on the computer but it is also possible to use a pre-processor to convert this to standard form before solution, so the general problem can be:

$$\text{Minimize} : \quad f=c.X \tag{2.52}$$

$$\text{Subject to} : \quad AX \leq b \tag{2.53}$$

$$X \text{ unrestricted in sign} \tag{2.54}$$

2.8.1.1 - Simplex Method

The simplex method proceeds in two phases. The purpose of phase I is to produce a system of equations in canonical form. It means that m slack variables are added to convert inequality-constrained problem to equality form. The equations (2.52) to (2.54) can be converted to the canonical form of Eq.(2.55).

$$\begin{bmatrix} 1 & 0 & 0 & 0 & a_{1,m+1} \ldots\ldots a_{1,k} \ldots\ldots a_{1,n} \\ 0 & 1 & 0 & 0 & a_{2,m+1} \ldots\ldots a_{2,k} \ldots\ldots a_{2,n} \\ \cdot & \cdot & \cdot & \cdot & \cdot \quad\quad \cdot \quad\quad \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \quad\quad \cdot \quad\quad \cdot \\ 0 & 0 & 1 & 0 & a_{r,m+1} \ldots\ldots a_{r,k} \ldots\ldots a_{r,n} \\ 0 & 0 & 0 & 1 & a_{m,m+1} \ldots\ldots a_{m,k} \ldots\ldots a_{m,n} \\ 0 & 0 & 0 & 0 & c_{m+1} \ldots\ldots c_k \ldots\ldots c_n \end{bmatrix} \cdot \begin{bmatrix} X_1 \\ X_2 \\ \cdot \\ \cdot \\ X_{m+1} \\ X_k \\ X_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \cdot \\ \cdot \\ b_r \\ b_m \\ f-f_0 \end{bmatrix} \tag{2.55}$$

The first m variables $X_j(j=1,m)$ are basic variables or slack variables and X_j (j=m+1,n) are design variables. f_0 is the current value of the objective function and all b_j (j=1,m) are non-negative. If the problem is already in canonical form the solution begins from phase II.

In phase II, trial subsets of the variables X are selected iteratively and checked whether they are optimal. If they are optimal, the computations are terminated; if they are not, the method drops one variable form the current subset and adds a new one in such a manner that the objective function value is improved. For more details refer to Appendix A.

2.8.2 - Nonlinear Programming

The fundamental theoretical paper on nonlinear programming by Kuhn and Tucker appeared in 1951. Indeed, it was not until 1960 that the first efficient nonlinear programming method (for the linear constraint case) was produced by Rosen [1961] and subsequently many different algorithms have been introduced for handling nonlinear programming.

Solution techniques for the nonlinear programming problem (NLP) play a central role in mathematical programming and practical problem solving and decision making. The techniques available for the solution of a constrained nonlinear programming problem can be classified into two broad categories, namely, the *direct* methods and *indirect* methods.

In the *direct* methods, the constraints are handled in an explicit manner whereas in most of the indirect methods, the constrained problem is solved as a sequence of unconstrained minimization problems.

The most popular methods for shape optimization are three non-linear mathematical programming techniques which are more reliable and powerful for highly non-linear

functions and constraints. These methods are, Sequential Unconstrained Minimization Techniques (SUMT), Sequential Linear Programming (SLP) and Feasible Direction Method (FDM). Of these, SLP and FDM are direct methods while SUMT is an indirect method. Here all three methods are introduced but since in Chapter 5 for shape optimization, SUMT and FDM are used, these two methods are of greatest interest.

For more details about other methods for non-linear programming refer to [Fletcher, 1981], [Gill et al, 1981], [Rao, 1978], [Zoutedijk, 1976], [Gill & Murray, 1974].

2.8.2.1- Sequential Unconstrained Minimization Techniques (SUMT)

SUMT transform the basic optimization problem, Eqs.(2.43) to (2.45), into solving a sequence of unconstrained minimization problems which have the same optimum solution as the constrained problem.

The classical approach to using SUMT is to create a pseudo-objective function of the form

$$\phi(X, r_p) = F(X) + r_p P(X) \tag{2.56}$$

where F(X) is the original objective function. P(X) is an imposed penalty function, the form of which depends on the SUMT being employed. The scalar r_p is a multiplier which is held constant for a complete unconstrained minimization. The subscript p is the unconstrained minimization number.

Probably the most extensive discussion of SUMT is provided by Fiacco & McCormick [1968]. They discuss a complete and rigorous account of SUMT and include many theorems on the theoretical aspects of these methods.

Penalty function methods can be classified into three groups:

I- Interior penalty function method

II- Exterior penalty function method

III- Extended penalty function method

In the *interior penalty function* method, a new pseudo-objective function is constructed by adding a penalty term to the objective function such that its value will be small at points away from the constraint boundaries and will tend to infinity as the constraint boundaries are approached. Thus, once the unconstrained minimization of $\phi(X,r_p)$ is started from any feasible point, the subsequent points generated will always lie within the feasible domain. Since the boundaries act as barriers during the minimization process, the interior penalty function methods are also known as "barrier" methods. Generally the interior penalty function penalizes the objective function as the design approaches a constraint, but constraint violations are never allowed.

The pseudo-objective function defined originally by Carroll [1961] is

$$\phi(X,r_p)=F(X)+r'_p\sum_{j=1}^{m}\frac{-1}{g_j(X)} \tag{2.57}$$

Including equality constraints then

$$\phi(X, r'_p, r_p)=F(X) + r'_p\sum_{j=1}^{m}\frac{-1}{g_j(X)} + r_p\sum_{k=1}^{l}\left[h_k(X)\right]^2 \tag{2.58}$$

The penalty parameter r'_p is used here to distinguish it from r_p, because r'_p begins as a relatively large positive number and decreases, while r_p begins as a small positive number and increases during the optimization process. An alternative form of penalty function for the interior penalty function method can be

$$P(X)=\sum_{j=1}^{m}-\left[\text{Log}\,(-\,g_j(X))\right] \tag{2.59}$$

In exterior penalty function method, the pseudo-objective function is generally given by

$$F(X)=P(X)+r_p\left[\sum_{j=1}^{m}\text{Max}\left[0, g_j(X)\right]^2+\sum_{k=1}^{L}\left[h_K(X)\right]^2\right] \qquad (2.60)$$

The iterative process is exactly the same as for the interior penalty function method, the only difference is that the value of the penalty parameter increases during the optimization process. It should be noted that the exterior penalty function penalizes the objective function only when constraints are violated and during the process, the design can be in an infeasible region as well as in the feasible region.

The iteration procedure of the interior and exterior penalty function methods can be summarized as follows:

1) Start with an initial design (for the interior penalty function method it should be an initial feasible design satisfying all constraints) and initial suitable values for r_p (for the interior penalty, r_p and r'_p), C, C' and P=1

2) Minimize $\phi(X,r_p,r'_p)$ by using any of the unconstrained minimization methods discussed in Sec.(2.6) and obtain X^*_p

3) Test for convergence, stop or continue.

4) Find the value of the next penalty parameter

$$r_{P+1}=C.\,r_P \quad (C>1) \qquad \text{and} \qquad r'_{P+1}=C'\,r'_P \quad (C'<1) \qquad (2.61)$$

and go to step 1 and continue.

From a practical viewpoint, the interior penalty function is suitable for solving inequality constrained minimization when the designs should always be in the feasible

region . However, most of the algorithms for unconstrained minimization are carried out by numerical methods which may well generate trial design points outside the feasible region. This difficulty can be overcome by using an extended interior penalty function which is defined outside the feasible region.

Kavlie and Moe [1971] proposed a formula for the extended function, which at the transition point with the interior penalty function satisfies continuity of both function value and first derivative. They defined the individual constraint contributions to the penalty term of the compound interior-extended penalty function as follows:

$$\tilde{g}_j(X) = \begin{cases} \dfrac{-1}{g_j(X)} & \text{for } g_j(X) \le \varepsilon \qquad (2.62) \\ -\dfrac{2\varepsilon - g_j(X)}{\varepsilon^2} & \text{for } g_j(X) > \varepsilon \qquad (2.63) \end{cases}$$

where the pseudo-objective function is given by

$$\phi(X)=F(X)+ r'_P \left[\sum_{j=1}^{m} \tilde{g}_j(X) \right] \qquad (2.64)$$

In Eqs.(2.62) and (2.63), the parameter ε is a small negative number which marks the transition from the interior penalty given by Eq.(2.62) to the extended penalty given by Eq.(2.63).

If a sequence of improving feasible designs is to be produced, it is necessary to choose ε so that the pseudo-objective function $\phi(X,r'p)$ has a positive slope at the constraint boundary [Vanderplaats, 1984]. Haftka and Starnes [1976] recommended that ε be defined by

$$\varepsilon= -C(r'_p)^a \qquad \frac{1}{3} \le a \le \frac{1}{2} \qquad (2.65)$$

where C is a constant. When the optimization process commences they choose ε in the range $-0.3 \leq \varepsilon \leq -0.1$ and at the beginning of each unconstrained minimization, ε is defined by Eq.(2.65) and is kept constant throughout that unconstrained minimization. Casis and Schmit [1976] had a similar approach for the definition of ε.

One of the most important tasks when using SUMT is the choice of the initial penalty parameters, because it has a major effect on overall optimization efficiency. Cassis and Schmit [1976] have given useful information about the choice of the ε and r'_p, in Eqs.(2.62) and (2.63). Here in the method of this thesis, used in chapter 5

$$\varepsilon = \frac{r'_P}{\phi(X^0)} \tag{2.66}$$

in which X^0 is the starting point. The first r'_p is chosen to satisfy the relation

$$\phi(X_0) = F(X_0) + r'_P \left[\sum_{j=1}^{m} \tilde{g}_j (X) \right] = 1.5\, F(X_0) \tag{2.67}$$

Queau and Trompette [1980] have used Eqs.(2.66) and (2.67) and have shown good results.

The extended penalty term given by Eqs.(2.62) and (2.63) is a linear extended penalty function. It is continuous and has continuous first derivatives. Haftka and Starnes [1976] introduced a quadratic extended penalty function which is continuous and has continuous first and second derivatives. It is suitable when a second-order method is being used for unconstrained minimization which is very costly, as described in Sec.(2.6.3).

The Augmented Lagrange Multiplier method is one of the SUMT type. It uses Lagrange multipliers of Eqs. (2.7) to (2.9) into the SUMT algorithm. The principal motivation of that is to reduce the dependence of the algorithm on the choice of the penalty parameters and the way by which they are updated during the optimization process. The ALM method has been studied by many authors [Pierre & Lowe, 1975],

[Imai, 1978], [Imai & Schmit, 1981] and [Rockafellar, 1973].

The general augmented lagrangian is

$$A(X, \lambda, r_p) = F(X) + \sum_{j=1}^{m}\left[\lambda_j \psi_j + r_p \psi_j^2\right] + \sum_{k=1}^{l}\left[\lambda_{k+m} h_k(X) + r_p\left[h_k(X)\right]^2\right] \tag{2.68}$$

$$\text{where} \quad \psi_j = \text{Max}\left[g_j(X), \frac{-\lambda_j}{2r_p}\right] \tag{2.69}$$

$$\lambda_j^{P+1} = \lambda_j^P + 2r_p\left[\text{Max}\left[g_j(X), \frac{-\lambda_j^P}{2r_p}\right]\right] \qquad j=1,m \tag{2.70}$$

$$\lambda_{k+m}^{P+1} = \lambda_{K+m}^P + 2r_p\, h_K(X) \qquad k=1,l \tag{2.71}$$

Using Lagrange multipliers makes this method very powerful and reliable but the only difficulty is that the current design can be infeasible and is not suitable when only feasible designs are allowed.

Convergence Criteria For Sequential Unconstrained Minimization Techniques

Since SUMT includes several unconstrained minimization iterations, the convergence criteria are that the original function F(X) does not change significantly during one unconstrained minimization, and that all $g_j(X) \leq 0$ and $h_K(X) = 0$ within a specified tolerance.

2.8.2.2 - Sequential Linear Programming (SLP) Technique

This is also referred to as Kelly's cutting plane method[1960]. This method involves linearizing the objective and constraint functions given in Eqs.(2.43) to (2.45). The problem may be approximated via a first-order Taylor series expansion so the problem will be

$$\text{Minimize : } F(X_{i+1}) \approx F(X_i)+\nabla F(X_i)\,\Delta X_i \tag{2.72}$$

$$\text{Subject to : } g_j(X_{i+1}) \approx g_j(X_i) + \nabla g_j(X_i).\,\Delta X_i \le 0 \quad j=1,m \tag{2.73}$$

$$h_K(X_{i+1}) \approx h_k(X_i)+\nabla h_k(X_i).\Delta X_i = 0 \qquad k=1,l \tag{2.74}$$

$$\text{Where } X_i^L \le X_i + \Delta X_i \le X_i^U$$

Therefore, the problem is the determination of a new value of $\{X\}_{i+1}$ where the change from the previous $\{X\}_i$ is given by the vector $\{\Delta X_i\}$. The objective becomes the maximization of the reduction in the function F(X). The problem may be rewritten as

$$\text{Minimize : } F(X_i)+\{\nabla F\}_i\ \{\Delta X\}_i \tag{2.75}$$

$$\text{Subject to : } g_j(X_i)+\{\nabla g_j(X_i)\}\ \{\Delta X_i\} \le 0 \quad j=1,m \tag{2.76}$$

$$h_k(X_i)+\{\nabla h_K(X_i)\}\ \{\Delta X_i\} = 0 \quad k=1,l \tag{2.77}$$

The SLP method first finds an initial feasible solution {X} which satisfies the above conditions. Then the problem is solved by the simplex method, which is discussed in Sec.(2.8.1.1), and the solution used as an improved starting point for a new linear programming problem. After several iterations a solution to the non-linear problem will be achieved.

For better convergence, it is necessary to include adaptive move limits [Reinschmidt et al, 1966], [Wills, 1973] to restrict the values of the variable vector {ΔX}. These

move limits may be specified thus:

$$\Delta X^{Min} \leq \{\Delta X\} \leq \Delta X^{Max} \tag{2.78}$$

where ΔX^{Min} and ΔX^{Max} are the lower and upper limits on $\{\Delta X\}$ respectively. Generally, the limits must be reduced as the iteration proceeds and the solution is approached.

2.8.2.3 - Feasible Direction Methods

In contrast to SLP, FDM deals directly with the nonlinearity of the problem. Methods of feasible direction are constructed to deal with inequality constraints, and for use in the case of equality-constrained problems, those equality constraints can be easily converted to inequality constraints.

This method determines a usable-feasible direction in which the design point may be moved in the design space. This direction is 'feasible' because it does not violate, at least over an infinitely small step, any of the constraints; and 'usable' because it results in a reduction of the objective function.

The basic idea is to choose a starting point satisfying all the constraints and to move to a better point according to the iterative scheme

$$X_{i+1} = X_i + \alpha S_i \tag{2.79}$$

which is exactly the same as Eq.(2.13).

For S to be usable, $F(X)_{i+1} < F(X)_i$ and clearly, such a search direction make an angle greater than 90^o with the gradient vector of the objective function. Mathematically, the usability requirement becomes

$$\nabla F(X).S \leq 0 \tag{2.80}$$

and S is feasible if, for some small move in that direction, the active constraints will not be violated. It means that the vector S must make an obtuse angle with all the constraints normals so

$$\nabla g_j (X) . S \leq 0 \tag{2.81}$$

The greatest reduction in F(X) can be achieved by finding S which minimizes the quantity in Eq. (2.80) while Eq.(2.81) meets with precise equality. By assuming the constraint is nonlinear and convex, a small move in this direction would violate the constraint. With these objectives, the following direction-finding problem is posed [Zoutendijk, 1960], [Fox, 1971].

$$\text{Maximize} \quad : \beta \tag{2.82}$$

$$\text{Subject to} \quad : S^T \nabla g_j (X) + \theta_j \beta \leq 0 \tag{2.83}$$

$$: S^T \nabla F + \beta \leq 0 \tag{2.84}$$

$$\text{and} \quad -C \leq S \leq C \text{ or S bounded} \tag{2.85}$$

The scalars, θ_j, are referred to as 'push-off' factors which effectively 'push' the design away form the active constraints. The final relationship gives simple bounds on the vector S. This algorithm is a linear programming formulation which calculates S and is discussed in Sec.(2.8.1), such that it maximizes the decrease in the objective function. The scalar β ensures that S will point into the feasible zone. If, at a design point, the solution to the above linear programming problem is $\beta_{max}=0$, then the design satisfies the Kuhn-Tucker conditions and the point is an optimum.

The choice of θ_j and the method of imposing bounds on S have significant effects on the efficiency of the method. Vanderplaats and Moses [1973] have found that θ_j is better defined by the following expression:

$$\theta_j = \left[1 - \frac{g_j(X)}{\varepsilon} \right]^2 \theta_0 \qquad (2.86)$$

This equation has the feature that, as constraint just becomes active the push-off factor is zero (where a constraint is considered active it its value is more positive than some parameter ε_{min}). As $g_j(X)$ approaches zero (from the negative side), θ_j increases quadratically until $\theta_j=\theta_0$ at $g_j(X)=0$. Usually $\theta_0=1$ is adequate for the nominal value of θ_j.

The other difficulty is the way of imposing bounds on the direction S. Probably the most common method is to impose bounds on the individual components of the form

$$-1 \le S_i^q \le 1 \qquad i=1,2,...,n \qquad (2.87)$$

then substituting

$$S_i = S'_i - 1 \qquad (2.88)$$

Eq. (2.87) changes to

$$0 \le S'_i \le 2 \qquad (2.89)$$

By substituting Eq. (2.89) into Eqs. (2.87) and (2.83) to (2.85) give the problem in standard linear programming form like Eqs. (2.52) to (2.54) for solution by the Simplex method, where the design variables are S'_i (i=1,2,...,n) and β. Solving this problem, final search 'S' can be found by Eq. (2.88). While this problem is easily solved, the resulting direction tends to point towards the corners of a hypercube in the design space. Although this approach has been found to yield satisfactory results [Moses & Onoda, 1969] a more desirable direction may be obtained by requiring that

$$y^q . y^q = S^q . S^q + \beta \leq 1 \tag{2.90}$$

$$\text{where} \quad y^q = \begin{bmatrix} S_1^q \\ S_2^q \\ \cdot \\ \cdot \\ S_n^q \\ \beta \end{bmatrix} \tag{2.91}$$

Using Eq. (2.90) with Eqs. (2.82) to (2.85) results in a linear problem with one quadratic constraint. While this problem may be solved by conversion to a quadratic programming problem with linear constraints, Zoutendijk [1960] has shown that a linear problem with one quadratic constraint may be readily solved by applying the conditions of Kuhn-Tucker to yield a strictly linear problem.

Eqs. (2.82) to (2.85) may be converted to matrix form as follows:

$$\text{Maximize} \quad : \quad P.y \tag{2.92}$$

$$\text{Subject to} \quad : \quad A.y \leq 0 \tag{2.93}$$

$$y . y \leq 1 \tag{2.94}$$

$$\text{where} \quad y = \begin{bmatrix} S_1 \\ S_2 \\ \cdot \\ \cdot \\ S_n \\ \beta \end{bmatrix} \qquad P = \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ 0 \\ 1 \end{bmatrix} \qquad A = \begin{bmatrix} \nabla^T g_1(X) & \theta_1 \\ \nabla^T g_2(X) & \theta_2 \\ \cdot & \cdot \\ \cdot & \cdot \\ \nabla^T g_j(X) & \theta_j \\ \nabla^T F(X) & 1 \end{bmatrix}$$

matrix B and vector C may be defined as follows :

$$B = -A\,A^T \tag{2.95}$$

$$C = -A\,P \tag{2.96}$$

Applying the Kuhn-Tucker conditions, the direction finding problem now requires the solution of the linear system [Vanderplaats, 1984]

$$BU + V = C \tag{2.97}$$

$$U \geq 0 \quad V \geq 0 \quad U.V = 0 \tag{2.98}$$

where 'U' is a vector of Lagrange multipliers and V is defined by

$$V = -\psi A\, y \tag{2.99}$$

where ψ is a scalar Lagrange multiplier. Eq. (2.97) can be written by the form

$$\begin{bmatrix} B & I \end{bmatrix} \begin{bmatrix} U \\ V \end{bmatrix} = C \tag{2.100}$$

where I is the identity matrix. The third requirement of Eq.(2.98) is that, $U_i=0$ when $V_i \neq 0$ and $V_i=0$ when $U_i \neq 0$. An initial solution can be $V_i=C_i$ and $U_i=0$, then the V_i are basic variables and U_i are non-basic variables. Now if some V_i is negative, the second condition of Eq. (2.98) is violated. The solution then proceeds by searching for the C_i with the least value. In this case, let C_{ii} be the diagonal element of the ith nonbasic variable. Initially, $C_{ii}=B_{ii}$ and it can be shown that if $C_i<0$ then $C_{ii}<0$ and

$$\text{Max} \left. \frac{C_i}{C_{ii}} \right|_{C_i<0} \tag{2.101}$$

should be found. Let K=i, which satisfies Eq. (2.101), and replace V_k in the basis by U_k. This is done by pivoting on C_{kk},just as in the standard Simplex algorithm. This process will be repeated until all $C_i \geq 0$. Then Eq. (2.97) and (2.98) are solved and the desired usable-feasible direction vector S^q is found from

$$\begin{bmatrix} S^q \\ \beta \end{bmatrix} = y^q = P - A^T U \tag{2.102}$$

For more details refer to [Vanderplaats & Moses, 1973], [Vanderplaats, 1984]. It should be mentioned that for an initially infeasible design, the above direction-finding problem needs a simple modification. It means the search direction will simultaneously minimize the objective function while overcoming the constraint violations. These considerations lead to the following statement of the direction-finding problem :

$$\text{Maximize} : \quad -\nabla F(X).S + \Phi \beta \tag{2.103}$$

$$\text{Subject to} : \quad \nabla g_j(X).S + \theta_j \beta \leq 0 \quad j \in J \tag{2.104}$$

$$S \,.\, S \leq 1 \tag{2.105}$$

Now the set of J contains all violated constraints. The scalar Φ is a weighting factor determining the relative importance of the objective and the constraints. According to Eqs.(2.103) to (2.105) the vector P in Eq.(2.92) and the matrix A of Eq.(2.93) should be modified as follows :

$$P = \begin{bmatrix} -\nabla F(X) \\ \Phi \end{bmatrix} \quad \text{and} \quad A = \begin{bmatrix} \nabla g_1^T(X) & \theta_1 \\ \nabla g_2^T(X) & \theta_2 \\ \vdots & \vdots \\ \nabla g_j^T(X) & \theta_j \end{bmatrix}_{j \times (n+1)} \tag{2.106}$$

and the procedure is exactly the same as before.

The algorithm for the feasible-direction method can be found in Fig.(2.2). The choice of convergence criterion depends on the nature of the design problem and the required precision. The best way is to begin with an initial constraint thickness (for example ε=-0.1). If any currently feasible design failed to reduce the objective function

by one percent, the $|\varepsilon|$ is reduced by a factor less than 1 and the design process continued. If two consecutive iterations failed to reduce the objective function by one percent and the latest value of ε was less than -0.005 the search is terminated. The reason is that at the beginning this numerically large "constraint boundary" has the effect of "trapping" constraints quickly, and it reduces the "zig-zagging" problem where a constraint alternately becomes active and inactive on successive iterations, which reduces the efficiency of the method.

2.9 - SCALING

Scaling and normalization have significant influence on the performance of optimization methods. To normalize constraints so that the order of each of them is unity improves the conditioning of the optimization, as does scaling each constraint so that the gradient of the constraint is of the same order of magnitude as the gradient of the objective function. Scaling the design variables causes each variable to be of similar weight during optimization. Therefore scaling makes the optimization more reliable. The scaling used in the methods of this thesis have been discussed, and are as follows :

I) Scaling Of The Constraints

Constraint functions should be balanced with respect to each other, i.e. all the constraints should have "equal weight" in the solution process. For example, if the stress at some points in the structure is required not to exceed a specified value, σ_0, the corresponding constraint function would be

$$g_j(X^q) = \frac{\sigma_j}{\sigma_0} - 1 \leq 0 \qquad (2.107)$$

where σ_j is the calculated stress at this point. In this manner, the constraint functions will have values between -1 and 0 for all strictly feasible designs.

$$g_j (X^q) \leq 0 \qquad (2.108)$$

II) Scaling The Objective Function

It is generally considered desirable for the magnitude of the objective function to be of an order no greater than unity in the region of interest. This can be achieved easily, by simply choosing the initial value of the objective function, F_0, to be divided into F and all its derivatives.

III) Normalization Of The Search Direction

Computationally, it is desirable to normalize the vector S. The simplest approach is to find the maximum absolute component $|S_i|_{i=1,n}$ .Then normalize S by dividing all components of S by this scaling factor, so the components of S change between -1 and +1. Therefore, the normalization provides a simple means of identifying the important parameters in the design.

Another possibility is to normalize S so that the Euclidian norm

$$\| S^k \| = 1 \qquad (2.109)$$

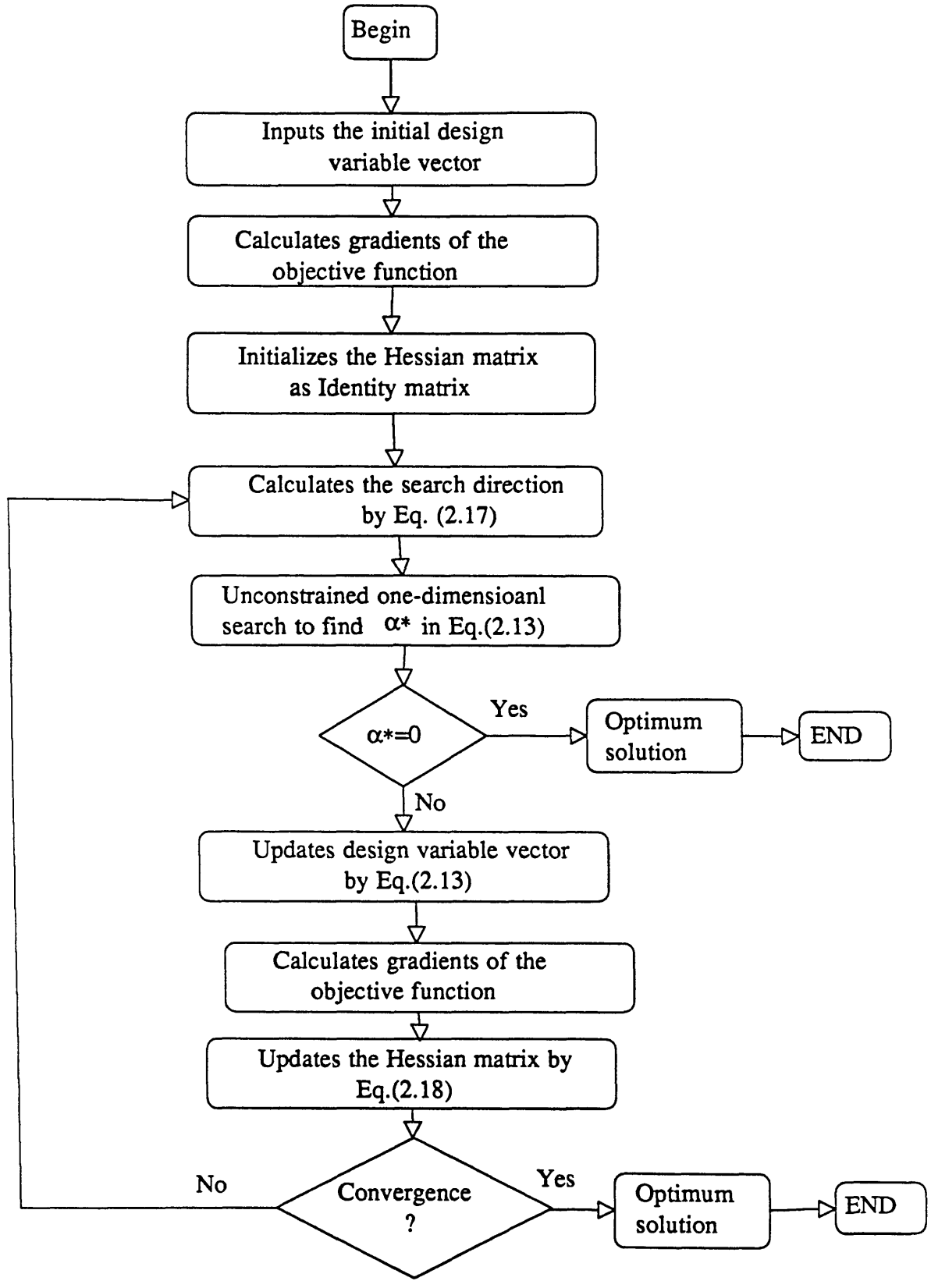


Fig. 2.1 Algorithm for the variable metric method

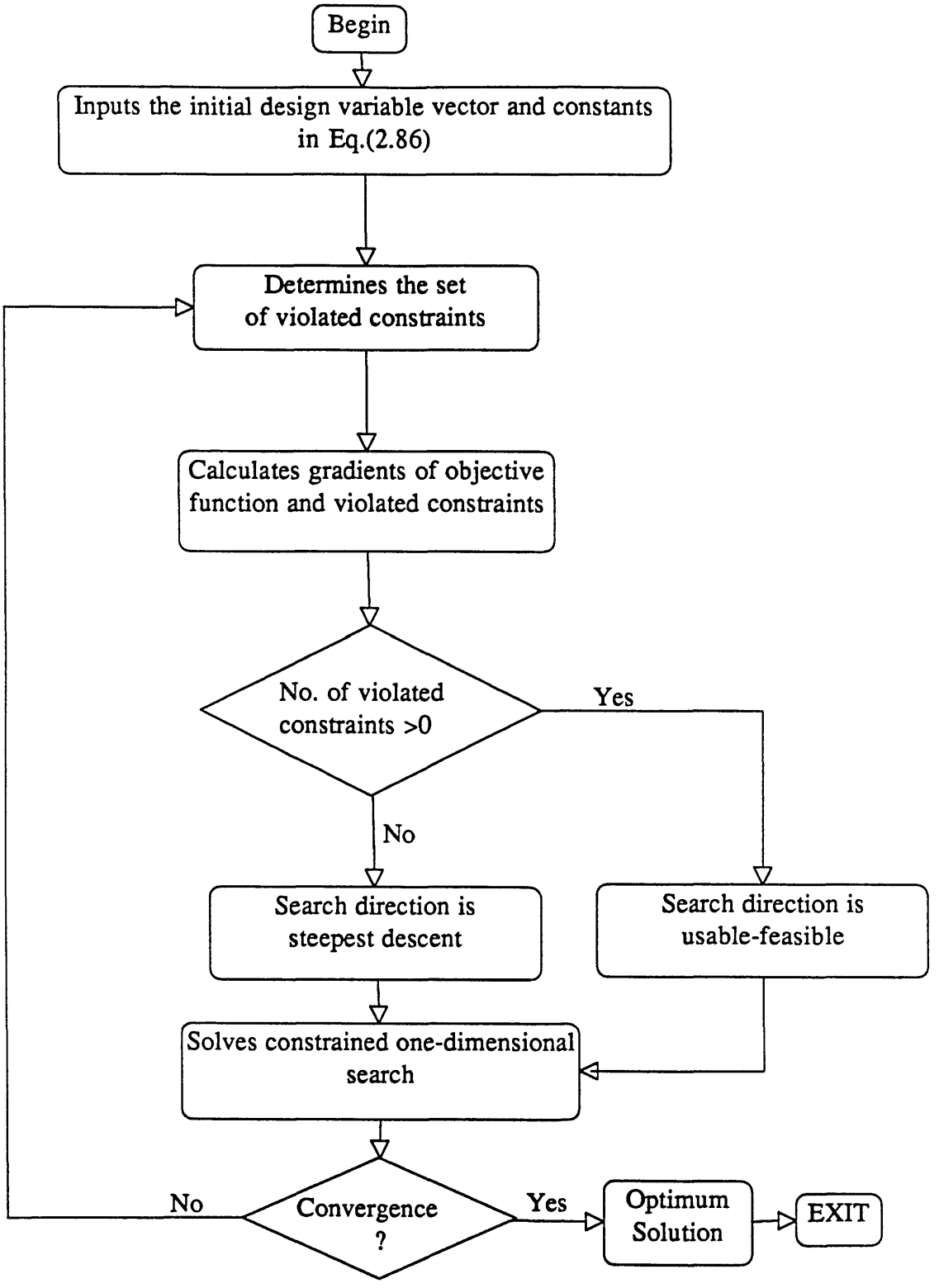


Fig. 2.2 Algorithm for the method of feasible directions

CHAPTER 3

THEORY OF THE BOUNDARY INTEGRAL EQUATION METHOD

3.1 - INTRODUCTION

Most practical engineering problems are impossible to solve by analytical methods. Finite element analysis has rapidly established itself as an important engineering technique in the last two decades or so for carrying out stress analysis on many type of structures. The boundary integral equation (BIE) method is an attractive alternative to the finite element technique for a wide range of applications in stress analysis and is particularly well established for linear elastic problems [Cruse, 1969, 1973, 1979], [Lachat & Watson, 1976] and [Tan & Fenner, 1979].

It is easier for mesh data preparation because only the surface of the component or structure to be analyzed needs to be discretized. It needs less computing time and storage for the same level of accuracy because within the solution domain the governing differential equations are satisfied exactly, rather than approximated via functional minimization as in the case of FEM [Fenner, 1983]. In using the BIE method less unwanted information is obtained when there is no need to have information about internal points. Since it offers continuous interior modelling within the solution domain and gives high resolution of stresses and displacements, it is particularly well suited for shape optimization problems.

However, while the BIE method has only relatively recently been applied to stress analysis, its mathematical basis is classical and similar to the methods in potential theory. In fact the fundamental theorems of BIE method, Green's theorem and Green's function were established as early as the 19th century by the mathematician George Green.

The analogue between classical elasticity and potential theory led to development of integral equation [Betti, 1872], [Somigliana, 1885] and [Lauricella, 1909], for boundary value problems of elasticity. The first numerical treatments of the BIE method were formulated by Jaswon [1963], Symm [1963], Rizzo [1967] and since then this method has been extended to a wide range of applications in elastostatics, elastoplasticity, elastodynamics, in potential flow, in heat conduction, in contact problems etc.. In this thesis, the BIE method has been shown to be a very good tool for the designer solving shape optimization problems.

The formulations of the BIE method may be classified into three different categories, namely, *indirect, semi-direct* and *direct* methods.

The *indirect* method, often known as the source potential or fictitious density method, formulates the problem in terms of unknown fictitious density functions which have little physical significance. However, once these density functions have been determined, the displacements and stresses can be easily obtained by simple integration [Mendleson, 1973].

In the *semi-direct* formulation, the problem is formulated in terms of unknown functions which are related to the stress functions. Then, the stresses and displacements can be determined from the solutions of the integral equations by simple differentiations [Jaswon & Symm, 1977] and [Mendleson, 1975].

In the *direct* method, the method used in this thesis, the integral equations are formed in terms of displacements and tractions on the boundary. The analytical formulation involves the transformation of the governing differential equations applicable to the whole domain into an integral equation over the boundary. The basic idea is the use of Betti's reciprocal work theorem together with Kelvin's fundamental solution.

3.2 - FORMULATION OF THE BOUNDARY INTEGRAL EQUATION METHOD

3.2.1 - Governing Equations Of Elasticity

The equilibrium equations for a given point in an elastic continuum body is given by

$$\sigma_{ij,j} + X_i = 0 \tag{3.1}$$

where X_i is the body force vector and σ_{ij} is the stress tensor. Hooke's law for a homogeneous isotropic material is :

$$\sigma_{ij} = \frac{2\mu\upsilon}{1-2\upsilon}\delta_{ij}\,\varepsilon_{mm} + 2\mu\varepsilon_{ij} \tag{3.2}$$

where ε_{ij} is the strain tensor, μ and υ respectively, are the strain shear modulus and Poisson's ratio and δ_{ij} is the Kronecker delta and

$$\delta_{ij} = \begin{cases} 1 & \text{for } i=j \\ 0 & \text{for } i\neq j \end{cases} \tag{3.3}$$

Substituting the strain tensor

$$\varepsilon_{ij} = \frac{1}{2}(u_{i,j} + u_{j,i}) \tag{3.4}$$

into Eq.(3.2) gives

$$\sigma_{ij} = \frac{2\mu\upsilon}{1-2\upsilon}\delta_{ij}\, u_{m,m} + \mu\,(u_{i,j} + u_{j,i}) \tag{3.5}$$

and since the traction is defined as the limiting vector of force per unit area then

$$t_i = \sigma_{ij}\, n_j \tag{3.6}$$

Substitution of Eq.(3.5) into Eq.(3.6)

$$t_i = \left[\frac{2\mu\upsilon}{1-2\upsilon}u_{k,k}\,\delta_{ij} + \mu\,(u_{i,j} + u_{j,i})\right] n_j \tag{3.7}$$

In order to obtain the Navier's equation involving the displacements u_i, Eq.(3.5) is differentiated and substituted into the stress equilibrium equations to give

$$\mu\nabla^2 u_i + \frac{\mu}{1-2\upsilon}u_{j,ji} + X_i = 0 \tag{3.8}$$

For two-dimensional problems, Eqs.(3.2) and (3.8) are as follows [Remzi, 1981]:

$$\sigma_{ij} = 2\mu\varepsilon_{ij} + \delta_{ij}\frac{2\mu\upsilon'}{1-2\upsilon'}\varepsilon_{kk} \tag{3.9}$$

and

$$\mu\nabla^2 u_i + \frac{\mu}{1-2\upsilon'}u_{j,ji} + X_i = 0 \tag{3.10}$$

where i,j=1,2 and

$$\upsilon' = \frac{\upsilon}{1+\upsilon} \quad , \quad E' = E(1-\upsilon^2) \qquad \text{Plane Stress} \qquad (3.11)$$

$$\upsilon' = \upsilon \quad , \quad E' = E \qquad \text{Plane Strain} \qquad (3.12)$$

The solution to Eq.(3.10) should satisfy boundary conditions for displacements and tractions on surface S (Fig.3.1), given respectively as

$$u_i(\underset{\sim}{x}) = g_i(\underset{\sim}{x}) \qquad \text{on } S_U \qquad (3.13)$$

$$t_i(\underset{\sim}{x}) = \sigma_{ij} n_j = h_i(\underset{\sim}{x}) \qquad \text{on } S_t \qquad (3.14)$$

where $g_i(\underset{\sim}{x})$ and $t_i(\underset{\sim}{x})$ are prescribed functions, and n_j is the unit outward normal to the surface S.

Substitution of Eqs.(3.2) and.(3.4) into Eq.(3.8) results in:

$$\left[\frac{2\mu\upsilon}{(1-2\upsilon)} u_{k,k}\delta_{ij} + \mu(u_{i,j} + u_{j,i})\right] n_j = h_i(\underset{\sim}{x}) \qquad (3.15)$$

Thus for a well-posed boundary value problem, the boundary conditions are either u_i, or a combination of the first derivatives of u_i over S.

3.2.2 - The Fundamental Solution In Two Dimensions

The fundamental solution to Eq.(3.8) can be conveniently obtained using a Galerkin vector $G_i(\underset{\sim}{x})$, which is a displacement function, assumed to have continuous derivatives through the fourth order [Fung, 1965], where

$$G_{iijjmm} = -\frac{X_i}{\mu} \tag{3.16}$$

Hence, G is biharmonic when the body force is zero (X_i=0) and the basic equations of elasticity are satisfied when

$$u_i = G_{,ijj} - \frac{1}{2(1-\upsilon')} G_{,jij} \tag{3.17}$$

The problem of a single concentrated force acting in the interior of an infinite body is known as Kelvin's problem. Consider the coordinate axes to be chosen such that the concentrated unit load, D(P,Q) (Dirac delta function), acts at the origin ζ_i=0 in the x_i direction. It can be verified that there is a Galerkin vector in two-dimensional case of the type [Remzi, 1981]

$$G_i = \frac{1}{8\pi\mu} \left(1+\log \frac{1}{r} \right) r^2 e_i \tag{3.18}$$

which will satisfy the boundary conditions that all the stresses vanish at infinity and the stress singularity is equivalent to a concentrated unit load in the x_i direction at the origin. In Eq.(3.18), e_i is a set of unit vectors in the x_i directions. r(P,Q) is defined as the distance between the field point Q($\underline{x}$) with coordinates $x(x_1, x_2)$ and the point load P($\underline{\zeta}$) with coordinates $\zeta(\zeta_1, \zeta_2)$:

$$r = \left[(x_i - \zeta_i)(x_i - \zeta_i) \right]^{1/2} \tag{3.19}$$

Differentiation of Eq.(3.18) and substitution into Eq.(3.17) results in the fundamental solution for the particular displacement solution

$$u_i (P,Q) = \frac{1}{8\pi\mu(1-\upsilon')} \left[(3-4\upsilon') \delta_{ij} \log \frac{1}{r} + r_{,i} r_{,j} \right] e_j \tag{3.20}$$

It should be mentioned that, here all the differentiations of r(P,Q) are with respect to the coordinates of the field point Q($\underset{\sim}{x}$), that is

$$r_{,i} = \frac{\partial r}{\partial x_i} = \frac{1}{r}(x_i - \zeta_i) = -\frac{\partial r}{\partial \zeta_i} \tag{3.21}$$

The particular traction vector at point P($\underset{\sim}{\zeta}$) is obtained by differentiating the displacement solution, Eq.(3.20) and using Eq.(3.7), then

$$t_i(P,Q) = -\frac{1}{4\pi(1-\upsilon')}\frac{1}{r}\left[\frac{dr}{dn}\left[(1-2\upsilon')\delta_{ij} + 2r_{,i}\, r_{,j}\right] - (1-2\upsilon')\left[n_j r_{,i} - n_i\, r_{,j}\right]\right] \tag{3.22}$$

where the normal, n_i, is taken at the field point Q($\underset{\sim}{x}$) and

$$\frac{dr}{dn} = \frac{\partial r}{\partial x_i}\, n_i \tag{3.23}$$

The displacement and the traction vectors corresponding to a unit load in each of the x_i directions can be represented by second order tensors

$$u_i(P,Q) = U_{ji}(P,Q)\, e_j \tag{3.24}$$

$$t_i(P,Q) = T_{ji}(P,Q)\, e_j \tag{3.25}$$

The first index in the Kernels $U_{ji}(P,Q)$ and $T_{ji}(P,Q)$ refers to the direction of the unit load at P($\underset{\sim}{\zeta}$) and the second index corresponds to the directions of the respective displacements and tractions at point Q($\underset{\sim}{x}$). Hence:

$$U_{ij}(P,Q) = \frac{1}{8\pi\mu(1-\upsilon')}\left[(3-4\upsilon')\,\delta_{ij}\log\left(\frac{1}{r}\right) + r_{,i}\, r_{,j}\right] \tag{3.26}$$

$$T_{ij}(P,Q) = -\frac{1}{4\pi(1-\upsilon')}\left(\frac{1}{r}\right)\left[\frac{dr}{dn}\left[(1-2\upsilon')\,\delta_{ij} + 2r_{,i}\,r_{,j}\right] - (1-2\upsilon')\left[n_j r_{,i} - n_i\, r_{,j}\right]\right] \tag{3.27}$$

3.2.3 - Betti's Reciprocal Work Theorem

Betti's general method of integrating the equations of elasticity is one of the classical methods of solving Navier's equations and it may be regarded as a direct extension of the divergence theorem which states that :

$$\int_R f_{i,i}\, dV = \int_S f_i\, n_i\, ds \tag{3.28}$$

where f_i is a vector field defined in a continuous region with continuous first partial derivatives in R and on its surface, S.

Consider two equilibrium states of an elastic body : one with displacements u^1_i due to the body forces X^1_i and surface tractions t^1_i, and the other with displacements u_i^2 due to body forces X_i^2 and surface tractions t^2_i. Betti's reciprocal theorem states that:

$$\int_S t_i^1 u_i^2\, ds + \int_R X_i^1 u_i^2\, dv = \int_S t_i^2 u_i^1\, ds + \int_R X_i^2 u_i^1\, dv \tag{3.29}$$

The proof of this theorem can be found in [Tan, 1979]. In the absence of body forces, Betti's reciprocal theorem may be written in the form:

$$\int_S t_i^1 u_i^2\, ds = \int_S t_i^2 u_i^1\, ds \tag{3.30}$$

In what follows, only the case of zero body force will be treated.

3.2.4 - Integral Equations For Displacements And Stresses At Interior Points

Consider Betti's reciprocal theorem Eq.(3.30) in a two dimensions for a domain R with boundary S. Assume that one stress state is unknown with no singularity, and that the other state corresponds to the fundamental solution to Navier's equation and is starred, then

$$\int_S t_i^* u_i \, ds = \int_S t_i u_i^* \, ds \tag{3.31}$$

By utilizing Kelvin's point load solution as the starred state, and excluding a small circle of radius ε with boundary S_ε, centred at point P, due to the singular nature of the solution (See Fig. 3.2), Eq.(3.31) becomes

$$\int_{S+S\varepsilon} T_{ij}(P,Q)\, u_j(Q)\, ds = \int_{S+S\varepsilon} U_{ij}(P,Q) t_j(Q)\, ds \tag{3.32}$$

Note that the variable point P and the field point Q, refer to points on the boundary and in the interior, respectively. When ε approaches zero, an integral equation is obtained, which relates the displacements at an interior point P, to the displacements and tractions on the surface

$$u_i(P) = -\int_S T_{ij}(P,Q) u_j(Q) ds + \int_S U_{ij}(P,Q) t_j(Q)\, ds \tag{3.33}$$

This equation is known as Somigliana's identity for displacements at point $P(\zeta)$ within the domain R due to known surface tractions and displacements. By differentiating the

Eq.(3.33) and using Hooke's law with respect to coordinates at P, the stress state at interior points is

$$\sigma_{ij}(P)= \frac{2\mu\upsilon}{1-2\upsilon}\delta_{ij}\, u_{k,k}+ \mu(u_{i,j}+ u_{j,i}) \tag{3.34}$$

Substituting Eq.(3.33) into Eq.(3.34) yields :

$$\sigma_{ij}(P) = -\int_S S_{kij}(P,Q)\, u_k(Q)\, ds + \int_S D_{kij}(P,Q)\, t_k(Q)\, ds \tag{3.35}$$

The third-order displacement and traction tensors, respectively, are given by :

$$D_{kij}(P,Q) = \frac{1}{4\pi(1-\acute{\upsilon})}\frac{1}{r}\left[(1-2\acute{\upsilon})(\delta_{jk}r_{,i}+\delta_{ik}r_{,j}-\delta_{ij}r_{,k})+2r_{,i}\,r_{,j}\,r_{,k}\right] \tag{3.36}$$

and

$$\begin{aligned} S_{kij}(P,Q) =& \frac{\mu}{2\pi(1-\acute{\upsilon})}\frac{1}{r^2}\Bigg[2\left[\acute{\upsilon}(\delta_{jk}r_{,i}+\delta_{ik}r_{,j})+(1-2\acute{\upsilon})\delta_{ij}-4r_{,i}\,r_{,j}\,r_{,k}\right]\frac{dr}{dn} \\ &+\left[(1-2\acute{\upsilon})\delta_{jk}+2\upsilon r_{,j}\,r_{,k}\right]n_i+\left[(1-2\acute{\upsilon})\delta_{ik}+2\acute{\upsilon}r_{,i}\,r_{,k}\right]n_j \\ &-\left[(1-4\acute{\upsilon})\delta_{ij}-2(1-2\acute{\upsilon})\,r_{,i}\,r_{,j}\right]n_k\Bigg] \end{aligned} \tag{3.37}$$

where $$\frac{\partial r}{\partial \zeta_i} = -\frac{\partial r}{\partial x_i} \tag{3.38}$$

where x_i are the coordinates of the point $Q(\underset{\sim}{x})$ and all the differentiation in Eqs.(3.36) & (3.37) are with respect to these coordinates and the normal is evaluated there too.

3.2.5 - The Boundary Integral Equation

Eq.(3.33), the Somigliana identity, represents the displacements of an interior point in the body in terms of integrals involving tractions and displacements of the boundary. If, in these integrals, the interior point is allowed to approach a boundary point, a limiting form of the Somigliana is obtained and the integral relations will be between boundary tractions and displacements.

As P approaches the boundary point P, the integrals need to be determined in the limit over the main boundary of the body and the boundary of a small circular region of radius ε with boundary S_ε centred at point P which is not contained inside the domain of the body , see Fig.(3.3)

$$u_i(P) = -\int_{S+S\varepsilon} T_{ij}(P,Q)\, u_j(Q)\, ds + \int_{S+S\varepsilon} U_{ij}(P,Q)\, t_j(Q)\, ds \tag{3.39}$$

By the limiting procedure of $\varepsilon \to 0$ [Tan, 1979], the boundary equation is derived:

$$C_{ij}(P)u_j(P) + \int_S T_{ij}(P,Q)\, u_j(Q)\, ds = \int_S U_{ij}(P,Q)\, t_j(Q)\, ds \tag{3.40}$$

where the integral on the left side of Eq.(3.39) only exists in the sense of the Cauchy principal value and so

$$C_{ij} = \delta_{ij} + \lim_{\varepsilon \to 0} \int_{S\varepsilon} T_{ij}(P,Q)\, ds \tag{3.41}$$

If the tangent plane at P is continuous, it can be shown [Remzi, 1981]

$$\lim_{\varepsilon \to 0} \int_{S\varepsilon} T_{ij}(P, Q)\, ds = -\frac{1}{2}\delta_{ij} \tag{3.42}$$

Eq.(3.40) is the boundary integral equation (BIE) which is an integral constraint relating boundary tractions to boundary displacements for an elastic body.

3.3 - THE NUMERICAL SOLUTION OF THE INTEGRAL EQUATIONS

3.3.1 - Introduction

The boundary integral equations, which are obtained in the previous section, are highly nonlinear and should be solved numerically. Numerical solutions to BIE have been found by discretizing the boundary into elements, over which the distributions of the unknowns are expressed in terms of suitable algebraic functions involving values at certain nodal points associated with the element. These unknowns, namely, tractions and displacements are to be determined by numerical integration of the kernel products over the elements. This reduces the BIE to a set of linear algebraic equations for the unknowns at the nodal points which can be solved to find boundary data. Once these boundary data are determined, the displacements and stresses at internal points may be calculated by using Somigliana's identities, Eqs.(3.33), (3.34).

3.3.2 - Discretization Of The Boundary

The variation of displacements and tractions may be defined by shape functions for example; linear, quadratic, cubic, etc.. It has been found that isoparametric variation is usually the most efficient [Lachat, 1975]. Therefore, in the present work, the surface is represented by a sequence of three-noded curved line elements and the tractions and displacements are considered to have quadratic variation in terms of shape functions of the intrinsic coordinate $\zeta \in [-1,1]$, proposed by Ergatoudis [1968] (Fig. 3.4). Thus, the coordinates, displacements and tractions at nodal points may then be expressed as:

$$x_i(\zeta) = N^c(\zeta)\, x_i^c \tag{3.43}$$

$$u_i(\zeta) = N^c(\zeta)\, u_i^c \tag{3.44}$$

$$t_i(\zeta) = N^c(\zeta)\, t_i^c \tag{3.45}$$

where $N^c(\zeta)$, c=1,2,3, are quadratic shape functions of the intrinsic coordinate ζ (ζ=-1,0,1 at the first, second, and third nodes along the element) given by

$$N^1(\zeta) = \frac{1}{2}\zeta\,(\zeta-1) \tag{3.46}$$

$$N^2(\zeta) = 1-\zeta^2 \tag{3.47}$$

$$N^3(\zeta) = \frac{1}{2}\zeta\,(\zeta+1) \tag{3.48}$$

where the Jacobian of the transformation from x_i to ζ is

$$J(\zeta) = |\, J_1^2(\zeta) + J_2^2(\zeta)|^{1/2} \tag{3.49}$$

where

$$J_i(\zeta) = \frac{dx_i}{d\zeta} = \frac{dN^c(\zeta)}{d\zeta}\, x_i^c \quad (i=1,2) \tag{3.50}$$

3.3.3 - Discretization Of The Integral Equations And Construction Of The System Of Equations

If the boundary is represented by M quadratic elements and N nodes, where

N=2M, Eq.(3.40) becomes

$$C_{ij}(P^a)\,u_j(p^a)+\sum_{b=1}^{M}\sum_{c=1}^{3}u_j(P^{d(b,c)})\int_{S_b}T_{ij}(P^a,Q(\zeta))\,N^c(\zeta)\,J(\zeta)\,d\zeta=$$

$$\sum_{b=1}^{M}\sum_{c=1}^{3}t_j(P^{d(b,c)})\int_{S_b}U_{ij}(P^a,Q(\zeta))\,N^c(\zeta)\,J(\zeta)\,d\zeta \qquad (3.51)$$

where P^a is the ath node, and d(b,c) is the number of cth node of the bth element and $J(\zeta)$ is the Jacobian of transformation.

Eq.(3.51) represents a set of 2N linear algebraic equations with 4N nodal point values of displacements and tractions, some of which are unknown. The coefficients of the equations are the values of the free term and the integrals of kernel products which may be evaluated by using Gaussian quadrature formulae and there should be some precautions due to singularity of r=0 when the field point and the load point coincide.

The problem reduces to the solution of the system of simultaneous equations of the form

$$A\,\overline{U}=B \qquad (3.52)$$

with 2N or 4M linear algebraic equations to be solved for the unknown boundary tractions and displacements.

3.3.4 - Surface Stresses

To calculate surface stresses from the already calculated surface tractions and strains, it is necessary to consider a local system of coordinates (ζ_1,ζ_2). Let m_{1j} be the

unit vector in the tangential and m_{2j} in the normal directions (j=1,2). Let $\overline{u_i}$, $\overline{t_i}$, $\overline{\varepsilon_{ij}}$ and $\overline{\sigma_{ij}}$ be the displacement, traction, strain and stress, respectively in the local coordinates. The displacement in the tangential direction is

$$\overline{u_1}(\zeta) = N^c(\zeta) u_j^c m_{1j} \tag{3.53}$$

Strain in the tangential direction is obtained by differentiating Eq.(3.53)

$$\overline{\varepsilon_{11}}(\zeta) = \frac{dN^c(\zeta)}{d\zeta} u_j^c m_{1j} \frac{1}{J(\zeta)} \tag{3.54}$$

Then by Hooke's law the stress tensor in the local coordinate system is

$$\overline{\sigma_{11}} = \left[\frac{E'}{1+\upsilon'} \overline{\varepsilon_{11}} + \upsilon' \overline{t_2} \right] /(1-\upsilon') \tag{3.55}$$

$$\overline{\sigma_{12}} = \overline{\sigma_{21}} = \overline{t_1} \tag{3.56}$$

$$\overline{\sigma_{22}} = \overline{t_2} \tag{3.57}$$

The definition of E' and υ' have been described by Eqs.(3.11) and (3.12). Then a simple transformation gives the stress tensor in the global system.

The computer program BIEQD is written by Remzi [1981] in Fortran in order to solve two-dimensional elastic problems which has been used in this thesis for displacement and stress analysis.

3.4 - WEIGHT CALCULATION BY THE BIE METHOD

During the weight minimization process, since the shape of the boundary is changing and it does not have a regular shape its weight should be calculated numerically. In the two-dimensional case there is a linear proportion between weight and area, so here the method of calculating area by using BIE method is discussed.

Let R be a closed bounded region in the XY- plane whose boundary c consists of a finite number of smooth curves. Let f(x,y) and g(x,y) be functions which are continuous and have continuous partial derivatives $\partial f/\partial y$ and $\partial g/\partial x$ everywhere in some domain containing R, Fig.(3.5). According to Green's theorem in the plane (Kreyszig, 1979)

$$\iint_R \left(\frac{\partial g}{\partial x} - \frac{\partial f}{\partial y}\right) dx\, dy = \int_c (f\, dx + g\, dy) \tag{3.58}$$

In Eq.(3.58) let f=0 and g=x then

$$\iint_R dxdy = \int_c x\, dy \tag{3.59}$$

The integral on the left- side is the area A of R, let f=-y and g=0 then

$$\iint_R dxdy = -\int_c y\, dx \tag{3.60}$$

By adding Eqs.(3.59) and (3.60) we obtain

$$A = \frac{1}{2}\int_c (x\, dy - y\, dx) \tag{3.61}$$

This formula expresses the area of R in terms of a line integral over the boundary.

The Eq.(3.61) may be written as

$$A = \frac{1}{2}\int_{c} \left(x \frac{dy}{ds} - y \frac{dx}{ds} \right) ds \tag{3.62}$$

where s is the arc length of c and (dy/ds)and (-dx/ds) are components of the unit normal vector to c so

$$\bar{n} = \frac{dy}{ds}\,\bar{i} - \frac{dx}{ds}\,\bar{j} \tag{3.63}$$

If the boundary is divided into quadratic elements, the cartesian coordinates $x_i (i=1,2)$ can be expressed as

$$x_i(\zeta) = N^c(\zeta)\, x_i^c \tag{3.64}$$

By changing the variables in the integral of Eq.(3.62) and using Jacobian of transformation we have

$$A = \frac{1}{2}\sum_{b=1}^{M}\int_{-1}^{1} \left[x(\zeta)\, n_1 + y(\zeta)\, n_2 \right] J(\zeta)\, d(\zeta) \tag{3.65}$$

where n_1 and n_2 are

$$n_1 = \frac{dy}{ds} = \frac{dy/d\zeta}{ds/d\zeta} = \frac{dy/d\zeta}{J(\zeta)} \tag{3.66}$$

$$n_2 = \frac{-dx}{ds} = \frac{-dx/d\zeta}{ds/d\zeta} = -\frac{dx/d\zeta}{J(\zeta)} \tag{3.67}$$

so Eq.(3.65) can be written as

$$A=\frac{1}{2}\sum_{b=1}^{M}\int_{-1}^{1}\left[\, x(\zeta)\,\frac{dy}{d\zeta} - y(\zeta)\,\frac{dx}{d\zeta}\,\right] d\zeta \qquad (3.68)$$

where $\dfrac{dy}{d\zeta}=\dfrac{dN^c(\zeta)}{d\zeta}\,y_i^c$, $\dfrac{dx}{d\zeta}=\dfrac{dN^c(\zeta)}{d\zeta}\,x_i^c$

and the area can be easily calculated by this equation when boundary has been divided into M quadratic elements and the weight of the component with uniform thickness and density ρ can be written as

$$W = \rho \,.\, A \qquad (3.69)$$

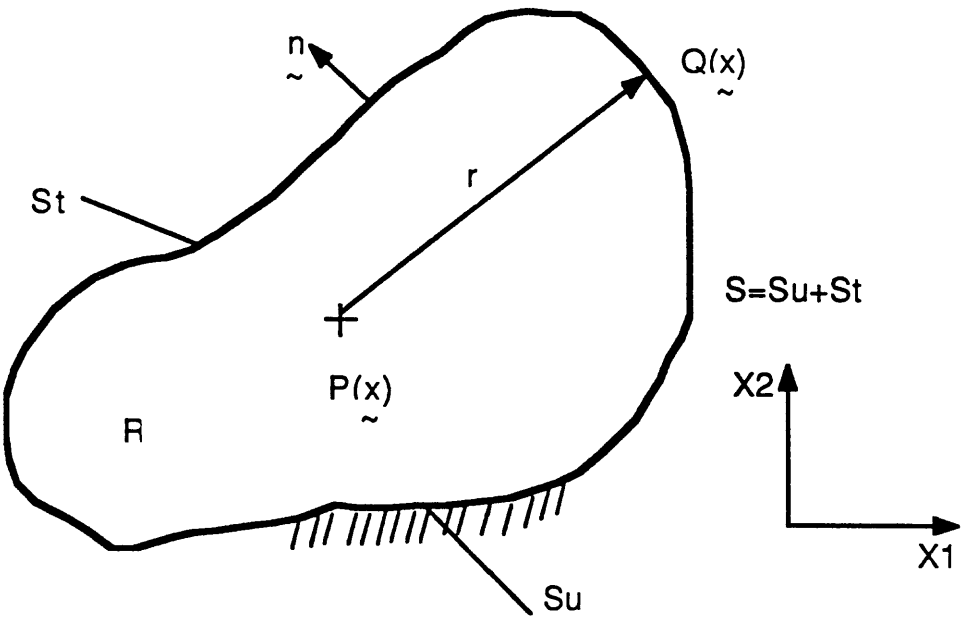


Fig. 3.1 The physical domain of a mixed boundary value problem

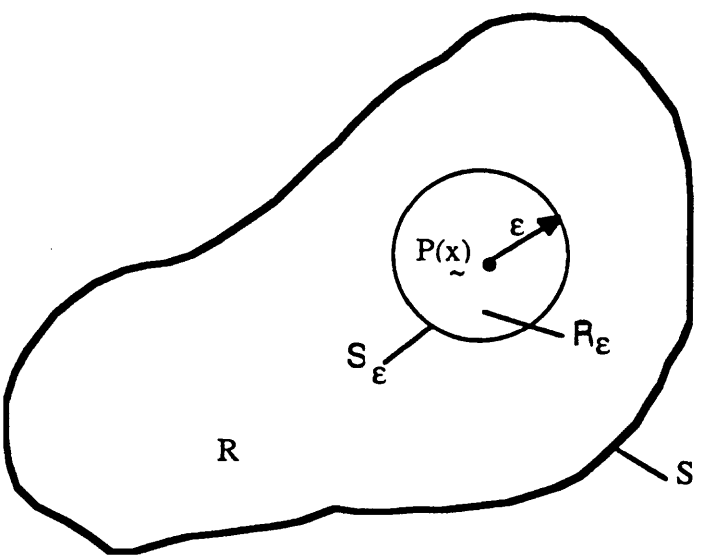


Fig. 3.2 The domain R_ε isolating $P(x)$

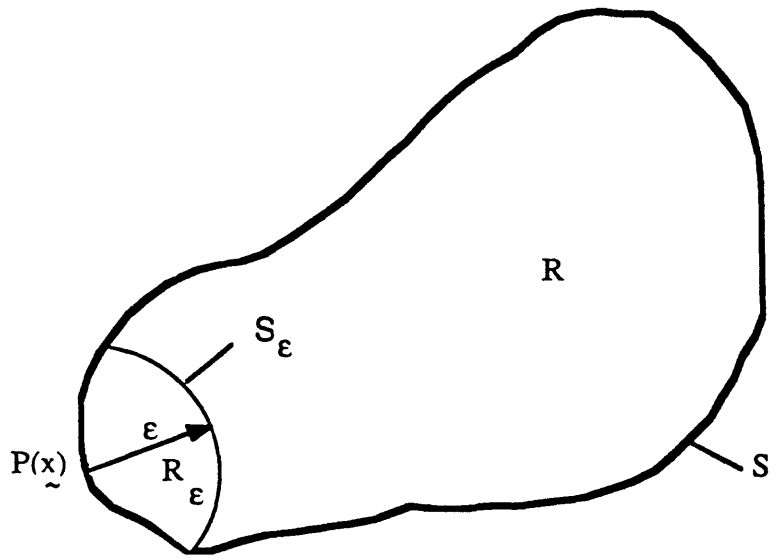


Fig. 3.3 The domain R_ε isolating $P(x)$

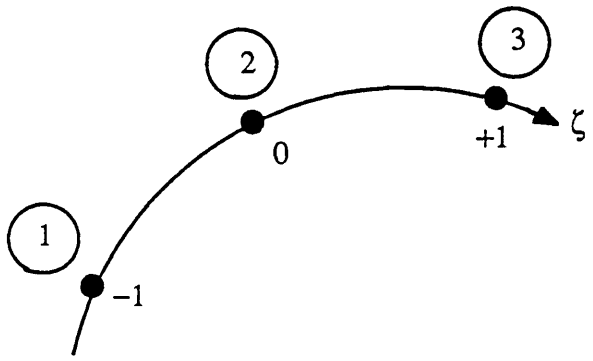


Fig. 3.4 Nodes for the quadratic representation of functions

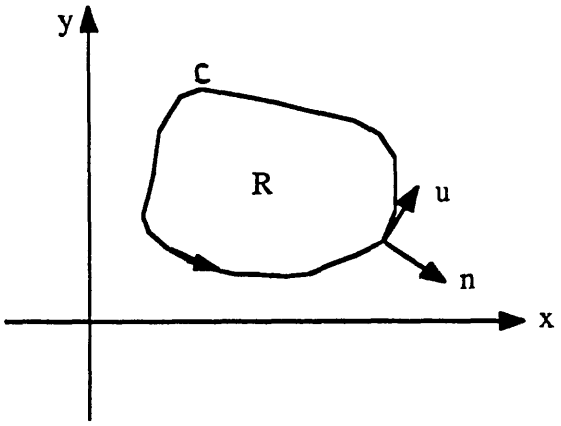


Fig.3.5 Region R with boundary c

CHAPTER 4

SHAPE DESIGN SENSITIVITY ANALYSIS USING THE BOUNDARY ELEMENT METHOD

4.1 - INTRODUCTION

As mentioned in chapter 2, the vital part of the optimization algorithm is the calculation of the gradients of the objective and constraint functions with respect to the design variables. During the optimization process, considerable economy can be achieved by calculating the gradients as part of the analysis process. The speed of convergence to an optimum design depends on the accuracy of these gradients. It is possible to calculate gradients using finite difference schemes. However, these are generally less accurate and take longer to calculate than the explicit gradient calculation methods used by optimization.

In shape optimization, these gradients of the objective function or constraints are investigated as shape design sensitivity analysis. Technically speaking, the effect of a shape change on performance of the structure is called " shape design sensitivity analysis". Design sensitivity analysis, is a fundamental requirement for the shape optimization. The first partial derivatives of structural response with respect to design variables provide the essential information required to couple mathematical optimization methods and structural analysis procedures, in order to reach the optimum in a reasonable number of solutions of the analysis model.

A substantial literature on the technical aspects of structural design sensitivity analysis exists. Some is devoted directly to the subject, but mostly are parts of the papers devoted to structural optimization.

There are two basic approaches to the calculation of sensitivity derivatives. The first

method uses the variational method. Rigorous optimality criteria can be derived mathematically and according to these relations the optimum shape will be determined. Haug & Rousselet [1980], Rousselet & Haug [1981], Haug et al [1984], and Choi & Haug [1983] have used the variational method to develop a unified theory for structural design sensitivity analysis for linear elastic structures and a full discussion can be found in Haug et al [1986]. It allows one to take the material derivative of the variational state equation and use the adjoint variable technique for design sensitivity analysis. It provides a unified method for calculation of design sensitivity for conventional and shape optimal design problems and also it can detect errors which may occur in the numerical method [Yang & Choi, 1985]. Usually the finite element method is used for the numerical analysis. The disadvantage of this method is the complexity of the approach.

The other alternative is to use the finite element method, and is based on direct differentiation of a discretized finite element system with respect to design variables and then calculating the derivatives by a finite difference scheme. Much work has been done by many researchers, in order to use the finite element method for design sensitivity analysis, such as [Botkin, 1981], [Braibant & Fleury, 1984], [Wassermann, 1983-84], [Kristensen & Madsen, 1976], [Pedersen & Laursen, 1982-83], [Wang et al, 1985]. In comparison with the variational method, this is more simple and general but the disadvantage with this method is that it is time consuming and, due to using the finite difference to approximate the derivatives, it is less accurate. It should be mentioned that in the implicit differential scheme, the derivatives are taken with respect to design variables after discretizing the problem, while in the variational approach, the derivatives are carried out before the discretization [Haug et al, 1986].

Using the boundary element method in sensitivity analysis overcomes these problems. From a mathematical point of view, it is easier than the variational method. It is general and problem independent. There is no need to change the solver for different problems to calculate derivatives. In comparison with the finite element method, it is less time-consuming and more accurate because the derivatives can be calculated analytically rather than by a finite difference scheme.

Here, the design sensitivity analysis is carried out by implicit differentiation of the

structural response. This means differentiating the basic equation of the BIE, Eq.(3.52) directly, and then calculating derivatives analytically.

It should be mentioned here that, since for the analysis, quadratic boundary elements have been used, the design variables are coordinates of some nodes on the movable part of the boundary, which will be discussed later in chapter 5. Throughout this thesis the design variables are only the coordinates of the end nodes, and the middle nodes are not included because the middle node is on one element while the first and end nodes are shared between two elements.

4.2 - ANALYTICAL DIFFERENTIATION OF THE STRUCTURAL RESPONSE WITH RESPECT TO THE DESIGN VARIABLES

4.2.1 - Derivatives Of Displacement

The gradients of stresses usually require the intermediate calculation of the gradient of displacements with respect to design variables, which are the coordinates of some nodes of the movable part of the boundary.

When analysis is carried out by the boundary element formulation, the simultaneous equations of the form of Eq.(3.52)

$$A\,\overline{U} = B \tag{4.1}$$

should be solved, where $\overline{U}$ is the vector of unknown displacements, and A and B are the matrices which contain the integrals of the traction and displacement kernels, respectively. Usually both [A] and [B] will be dependent on the design variables $x_k(k=1,2)$ which determine coordinates of the nodes, x and y respectively.

Consider the derivative of the above equation with respect to a design variable x_k

$$\frac{\partial}{\partial x_k}(A\,\bar{U}) = B \tag{4.2a}$$

$$\frac{\partial}{\partial x_k} A\,\bar{U} + A\frac{\partial \bar{U}}{\partial x_k} = \frac{\partial B}{\partial x_k} \tag{4.2b}$$

$$A\frac{\partial \bar{U}}{\partial x_k} = \left[\frac{\partial B}{\partial x_k} - \frac{\partial}{\partial x_k} A\,\bar{U}\right] \tag{4.2c}$$

Eq.(4.2c) contains n equations for the unknown gradients $\partial\bar{U}/\partial x_k$. This is equivalent to solving the same equation as Eq.(4.1), thus if the quantity in parentheses in Eq.(4.2c) is separately assembled, then the displacement derivative vector $\partial\bar{U}/\partial x_k$ can be computed in one pass by re-entering the equation solver.

By differentiating both sides of Eq.(3.51)

$$C_{ij}(P^a)u_j(P^a) + \sum_{b=1}^{M}\sum_{c=1}^{3} u_j(P^{d(b,c)}) \int_{S_b} T_{ij}(P^a, Q(\zeta)) N^c(\zeta)\, J(\zeta)\, d\zeta$$

$$= \sum_{b=1}^{M}\sum_{c=1}^{3} t_j(P^{d(b,c)}) \int_{S_b} U_{ij}(P^a, Q(\zeta))\, N^c(\zeta) J(\zeta)\, d\zeta \tag{4.3}$$

we have

$$C_{ij}(P^a)\frac{\partial u_j(P^a)}{\partial x_k} + \sum_{b=1}^{M}\sum_{C=1}^{3} \frac{\partial u_j(P^{d(b,c)})}{\partial x_k} \int_{S_b} T_{ij}(P^a, Q(\zeta))\, N^c(\zeta) J(\zeta) d\zeta =$$

$$\sum_{b=1}^{M}\sum_{c=1}^{3} \frac{\partial t_j(P^{d(b,c)})}{\partial x_k} \int_{S_b} U_{ij}(P^a, Q(\zeta)) N^c(\zeta) J(\zeta) d\zeta + \sum_{b=1}^{M}\sum_{c=1}^{3} t_j(P^{d(b,c)}) \frac{\partial}{\partial x_k} \int_{S_b} U_{ij}(P^a, Q(\zeta)) N^c(\zeta) J(\zeta) d\zeta$$

$$-\sum_{b=1}^{M}\sum_{c=1}^{3} u_j(P^{d(b,c)})\frac{\partial}{\partial x_k}\int_{S_b} T_{ij}(P^a,Q(\zeta))\,N^c(\zeta)J(\zeta)d\zeta \qquad (4.4)$$

Since for calculating displacement gradients, only the calculation of the right-hand side of the Eq.(4.2c) is necessary, then the second member matrix B in Eq.(4.1), should be replaced by a new matrix, B', which contains the derivatives with respect to the design variables where

$$B'=\sum_{b=1}^{M}\sum_{c=1}^{3}\frac{\partial t_j(P^{d(b,c)})}{\partial x_k}\int_{S_b} U_{ij}(P^a,Q(\zeta))\,N^c(\zeta)J(\zeta)d\zeta+\sum_{b=1}^{M}\sum_{c=1}^{3} t_j(P^{d(b,c)})\frac{\partial}{\partial x_k}$$

$$\int_{S_b} U_{ij}(P^a,Q(\zeta))N^c(\zeta)J(\zeta)d\zeta-\sum_{b=1}^{M}\sum_{c=1}^{3} u_j(P^{d(b,c)})\frac{\partial}{\partial x_k}\int_{S_b} T_{ij}(P^a,Q(\zeta))N^c(\zeta)J(\zeta)d\zeta \qquad (4.5)$$

or

$$B'=\sum_{b=1}^{M}\sum_{c=1}^{3}\frac{\partial t_j(P^{d(b,c)})}{\partial x_k}\int_{S_b} U_{ij}(P^a,Q(\zeta))N^c(\zeta)J(\zeta)d\zeta+\sum_{b=1}^{M}\sum_{c=1}^{3} t_j(P^{d(b,c)})\int_{S_b}\frac{\partial U_{ij}(P^a,Q(\zeta))}{\partial x_k}N^c(\zeta)J(\zeta)d\zeta$$

$$+\sum_{b=1}^{M}\sum_{c=1}^{3} t_j(P^{d(b,c)})\int_{S_b} U_{ij}(P^a,Q(\zeta))N^c(\zeta)\frac{\partial J(\zeta)}{\partial x_k}d\zeta-\sum_{b=1}^{M}\sum_{c=1}^{3} u_j(P^{d(b,c)})\int_{S_b}\frac{\partial T_{ij}(P^a,Q(\zeta))}{\partial x_k}N^c(\zeta)J(\zeta)d\zeta$$

$$-\sum_{b=1}^{M}\sum_{c=1}^{3} u_j(P^{d(b,c)})\int_{S_b} T_{ij}(P^a,Q(\zeta))N^c(\zeta)\frac{\partial J(\zeta)}{\partial\zeta}d\zeta \qquad (4.6)$$

while the kernel products are

$$U_{ij}(P,Q)=\frac{1}{8\pi\mu(1-\upsilon')}\left[(3-4\upsilon')\delta_{ij}\log\left(\frac{1}{r}\right)+r_{,i}r_{,j}\right] \qquad (4.7)$$

$$T_{ij}(P,Q)= - \frac{1}{4\pi(1-\upsilon')} (\frac{1}{r})\left[\frac{dr}{dn}\left[(1-2\upsilon')\delta_{ij} +2r_{,i}\, r_{,j} \right] - (1-2\upsilon')\left[n_j r_{,i} - n_i\, r_{,j} \right]\right] \tag{4.8}$$

and in Eqs.(4.7) and (4.8), the differentiations of r(P,Q) are with respect to the coordinates of the field point Q($\underset{\sim}{x}$), that is

$$r_{,i} = \frac{\partial r}{\partial x_i} = \frac{1}{r}(x_i - \zeta_i) = - \frac{\partial r}{\partial \zeta_i} \tag{4.9}$$

In order to calculate the derivative of any function like $\phi(r)$ with respect to design variables $x_k(k=1,2)$ we have

$$\frac{\partial \phi}{\partial x_k} = \frac{\partial \phi}{\partial \zeta_k} \cdot \frac{\partial \zeta_k}{\partial r} \cdot \frac{\partial r}{\partial x_k} \tag{4.10}$$

Since from Eq.(3.21)

$$\frac{\partial r}{\partial x_i} = - \frac{\partial r}{\partial \zeta_i} \tag{4.11}$$

then

$$\frac{\partial \phi}{\partial x_k} = - \frac{\partial \phi}{\partial \zeta_k} \tag{4.12}$$

Since the integrals will be calculated numerically over the elements, the field point is a gauss point on the element. When the design variable is a load point, the derivatives are calculated with respect to the corresponding field points on every element and then signs are reversed. When the variable node is on the field element, the derivatives are calculated for all the Gauss points over that element and multiplied by the appropriate shape function, so the derivative of any function like $\phi(r)$ with respect to the design variable x_k will be

$$\frac{\partial \phi}{\partial x_k} = \frac{\partial \phi}{\partial x_G} N^c(\zeta_G) \tag{4.13}$$

where c is the local element numbering (c=1,3) and the G is the appropriate Gauss point during integration.

In order to calculate Eq.(4.6), the derivatives of kernel products and the Jacobian of transformation should be obtained and here they are discussed in detail.

4.2.2 - Derivatives Of Jacobian Of Transformation

Consider the design variable node coordinates x_k to be on the mth element, and the Jacobian is defined as

$$J = \left[\left(\frac{dx}{d\zeta}\right)^2 + \left(\frac{dy}{d\zeta}\right)^2 \right]^{1/2} \tag{4.14}$$

where $x(\zeta)=x^cN^c(\zeta)$ and $y(\zeta)=y^cN^c(\zeta)$

$$\frac{\partial J}{\partial x_k} = \left[J\right]^{-1} \left[\frac{dx}{d\zeta} \frac{\partial}{\partial x_k}\left(\frac{dx}{d\zeta}\right) + \frac{dy}{d\zeta} \frac{\partial}{\partial x_k}\left(\frac{dy}{d\zeta}\right) \right] \tag{4.15}$$

If the variable node is not on the element, the gradient of the Jacobian is zero. If the x-coordinate of node is variable then

$$\frac{\partial}{\partial x_k}\left(\frac{dy}{d\zeta}\right) = 0 \tag{4.16}$$

and then

$$\frac{\partial J}{\partial x_1} = [J]^{-1}\left[\frac{dx}{d\zeta}\frac{dN^c(\zeta)}{d\zeta}\right] \tag{4.17}$$

When the y-coordinate of node is variable

$$\frac{\partial}{\partial x_k}\left(\frac{dx}{d\zeta}\right) = 0 \tag{4.18}$$

and then

$$\frac{\partial J}{\partial x_2} = [J]^{-1}\left[\frac{dy}{d\zeta}\frac{dN^c(\zeta)}{d\zeta}\right] \tag{4.19}$$

4.2.3 - Derivatives Of Kernel Products

$$U_{ij,k} = \frac{1}{8\pi\mu(1-\upsilon')}\left[(3-4\upsilon')\delta_{ij}\frac{-1}{r}r_{,k} + r_{,i}r_{,jk} + r_{,ik}r_{,j}\right] \tag{4.20}$$

$$T_{ij,k} = \frac{1}{4\pi(1-\upsilon')}\frac{1}{r}r_{,k}\left[\frac{dr}{dn}\left[(1-2\upsilon')\delta_{ij} + 2r_{,i}r_{,j}\right] - (1-2\upsilon')\left[n_j r_{,i} - n_i r_{,j}\right]\right]$$

$$- \frac{1}{4\pi(1-\upsilon')}\frac{1}{r}\left[\left(\frac{dr}{dn}\right)_{,k}\left[(1-2\upsilon')\delta_{ij} + 2r_{,i}r_{,j}\right] + \frac{dr}{dn}\left[2r_{,ik}r_{,j} + 2r_{,i}r_{,jk}\right]\right.$$

$$\left. - (1-2\upsilon')\left[n_j r_{,ik} - n_{j,k}r_{,i} - n_i r_{,jk} - n_{i,k}r_{,j}\right]\right] \tag{4.21}$$

where

$$\left(\frac{dr}{dn}\right)_{,k} = (r_{,i}n_i)_{,k} = (r_{,ik}n_i + r_{,i}n_{i,k}) \tag{4.22}$$

and k=1,2.

4.2.4 - Derivatives Of Stresses

With differentiating both sides of Eq.(3.54), the gradient of strain in the tangential direction is obtained from

$$\overline{\varepsilon_{11,k}(\zeta)} = \frac{dN^C(\zeta)}{d\zeta} u^C_{j,k} m_{1j} \frac{1}{J(\zeta)} + \frac{dN^C(\zeta)}{d\zeta} u^C_j m_{1j,k} \frac{1}{J(\zeta)} + \frac{dN^C(\zeta)}{d\zeta} u^C_j m_{1j} \frac{-J(\zeta)_{,k}}{[J(\zeta)]^2} \tag{4.23}$$

which is a function of gradients of displacements which are calculated above. Differentiating Eqs.(3.55) to (3.57) which have been obtained by Hooke's law

$$\left[\begin{array}{ll} \overline{\sigma_{11,k}} = \left[\frac{E'}{1+\upsilon'} \overline{\varepsilon_{11,k}} + \upsilon' \overline{t_{2,k}} \right] /(1-\upsilon') & (4.24) \\ \overline{\sigma_{22,k}} = \overline{t_{2,k}} & (4.25) \\ \overline{\sigma_{12,k}} = \overline{\sigma_{21,k}} = \overline{t_{1,k}} & (4.26) \end{array}\right.$$

On the other hand, the transformation of the stress tensor from the local system to the global system is as follows [Timoshenko, 1970]

$$\left[\begin{array}{ll} \sigma_{xx} = \overline{\sigma_{11}} n_2^2 + \overline{\sigma_{22}} n_1^2 - 2\overline{\sigma_{12}} n_1 n_2 & (4.27) \\ \sigma_{yy} = \overline{\sigma_{11}} n_1^2 + \overline{\sigma_{22}} n_2^2 + 2\overline{\sigma_{12}} n_1 n_2 & (4.28) \\ \sigma_{xy} = (-\overline{\sigma_{11}} + \overline{\sigma_{22}}) n_1 n_2 + \overline{\sigma_{12}} (n_1^2 - n_2^2) & (4.29) \end{array}\right.$$

In order to calculate the gradients of stress tensor in the global system, the Eqs.(4.27) to (4.29) can be differentiated, and then using the quantities of Eq.(4.23) and Eqs (4.24) to (4.26) to give

$$\sigma_{xx,k} = \overline{\sigma_{11,k}}\, n_2^2 + \overline{\sigma_{22,k}}\, n_1^2 - 2\,\overline{\sigma_{12.k}}\, n_1 n_2 + 2\overline{\sigma_{11}} n_2 n_{2,k} + 2\overline{\sigma_{22}}\, n_1 n_{1,k}$$

$$-2\overline{\sigma_{12}}\, n_{1,k} n_2 - 2\,\overline{\sigma_{12}}\, n_1 n_{2,k} \qquad (4.30)$$

$$\sigma_{yy,k} = \overline{\sigma_{11,k}}\, n_1^2 + \overline{\sigma_{22,k}}\, n_2^2 + 2\,\overline{\sigma_{12.k}}\, n_1 n_2 + 2\overline{\sigma_{11}} n_1\, n_{1,k} + 2\overline{\sigma_{22}}\, n_2\, n_{2,k}$$

$$+2\overline{\sigma_{12}}\, n_{1,k} n_2 + 2\,\overline{\sigma_{12}}\, n_1 n_{2,k} \qquad (4.31)$$

$$\sigma_{xy,k} = (-\overline{\sigma_{11,k}} + \overline{\sigma_{22,k}})\, n_1 n_2 + (-\overline{\sigma_{11}} + \overline{\sigma_{22}})(n_{1,k} n_2 + n_1 n_{2,k}) +$$

$$\overline{\sigma_{12}}\,(2n_1 n_{1,k} - 2n_2 n_{2,k}) + \overline{\sigma_{12,k}}\,(n_1^2 - n_2^2) \qquad (4.32)$$

Since during shape optimization in this thesis the Von-Mises stress criterion has been used, then here the derivatives of equivalent stresses will be determined.

The expansion for Von-Mises stress is [Mendleson, 1968]

$$\sigma_e = \left[\, \sigma_{xx}^2 - \sigma_{xx}\,\sigma_{yy} + \sigma_{yy}^2 + 3\sigma_{xy}^2 \right]^{1/2} \qquad (4.33)$$

Differentiating with respect to the design variables

$$\sigma_{e,k} = \left[2\sigma_{xx}\,\sigma_{xx,k} - \sigma_{xx,k}\,\sigma_{yy} + 2\sigma_{yy}\,\sigma_{yy,k} - \sigma_{xx}\,\sigma_{yy,k} + 6\sigma_{xy}\,\sigma_{xy,k}\right] \times \frac{1}{2} \times \sigma_e^{-1}$$

(4.34)

4.3 - ANALYTIC DIFFERENTIATION OF WEIGHT WITH RESPECT TO THE DESIGN VARIABLES

As discussed in Sec.(3.4), the weight of a component in a two-dimensional case with uniform unit thickness and density ρ, which has been discretized to M quadratic

boundary elements and N nodes can be written as

$$W = \rho \, . \, A \tag{4.35}$$

where A, the area of the domain, is

$$A = \frac{1}{2} \sum_{b=1}^{M} \int_{-1}^{1} \left[x(\zeta) \frac{dy}{d\zeta} - y(\zeta) \frac{dx}{d\zeta} \right] d\zeta \tag{4.36}$$

$$\text{where } \frac{dy}{d\zeta} = \frac{dN^c(\zeta)}{d\zeta} y_i^c \quad , \quad \frac{dx}{d\zeta} = \frac{dN^c(\zeta)}{d\zeta} x_i^c \tag{4.37}$$

By differentiating Eq.(4.36) with respect to the design variable x_k

$$\frac{\partial A}{\partial x_k} = \frac{1}{2} \sum_{b=1}^{M} \int_{-1}^{1} \left[\frac{\partial}{\partial x_k} x(\zeta) \frac{dy}{d\zeta} + x(\zeta) \frac{\partial}{\partial x_k} \left(\frac{dy}{d\zeta}\right) - \frac{\partial}{\partial x_k} y(\zeta) \frac{dx}{d\zeta} - y(\zeta) \frac{\partial}{\partial x_k} \left(\frac{dx}{d\zeta}\right) \right] d\zeta \tag{4.38}$$

If x_k is the x-coordinate of movable node then

$$\frac{\partial}{\partial x_k} \left(\frac{dy(\zeta)}{d\zeta}\right) = 0 \qquad \text{and} \frac{\partial}{\partial x_k} (y(\zeta)) = 0 \tag{4.39}$$

and Eq.(4.38) can be written as follows

$$\frac{\partial A}{\partial x_1} = \frac{1}{2} \sum_{b=1}^{M} \int_{-1}^{1} \left[\frac{\partial}{\partial x_1} x(\zeta) \frac{dy}{d\zeta} - y(\zeta) \frac{\partial}{\partial x_1} \left(\frac{dx}{d\zeta}\right) \right] d\zeta \tag{4.40}$$

If the local number of this node is t then

$$\frac{\partial}{\partial x_1}(x(\zeta)) = N^t(\zeta) \qquad \text{and} \quad \frac{\partial}{\partial x_1}\left(\frac{dx}{d\zeta}\right) = \frac{dN^t(\zeta)}{d\zeta} \tag{4.41}$$

when the x_k is y-coordinate of a movable node, then

$$\frac{\partial}{\partial x_2}\left(\frac{dx(\zeta)}{d\zeta}\right) = 0 \qquad \text{and} \quad \frac{\partial}{\partial x_2}(x(\zeta)) = 0 \tag{4.42}$$

$$\frac{\partial A}{\partial x_2} = \frac{1}{2}\sum_{b=1}^{M}\int_{-1}^{1}\left[x(\zeta)\frac{\partial}{\partial x_2}\left(\frac{dy(\zeta)}{d\zeta}\right) - \frac{\partial}{\partial x_2} y(\zeta)\frac{dx(\zeta)}{d\zeta}\right] d\zeta \tag{4.43}$$

where $\frac{\partial}{\partial x_2}\left(\frac{dy(\zeta)}{d\zeta}\right) = \frac{dN^t(\zeta)}{d\zeta}$ and $\frac{\partial}{\partial x_2}(y(\zeta)) = N^t(\zeta)$

In both cases when the design node is not included in an element, the gradient over that element is zero. Therefore summation of Eqs(4.40) and (4.43) for only two elements, which are shared at that node, will be calculated, and for the rest of elements is zero.

4.4 - COMPARING DERIVATIVES CALCULATED BY FINITE DIFFERENCE AND DIRECT ANALYTIC DIFFERENTIATION

A simple plate is studied to demonstrate the analytical sensitivity analysis and the boundary element implementation. The boundary elements, dimensions, material properties and loading are shown in Fig.(4.1). Due to the symmetry only the right half of the plate is considered.

The design variable, x coordinate of node 17, is chosen to represent a change in the

shape of the edge BC. 24 quadratic boundary elements have been used to describe the plate as shown in Fig.(4.2). The derivatives of displacements and stresses are calculated by the boundary element method described in the last sections. The accuracy is compared against the results of the finite difference applied to boundary element analysis. The results in Tables (4.1) and (4.2) show the displacement sensitivity (in the X and Y directions, u and v, respectively). The stress sensitivity results are shown in Table (4.3). The first column in these tables shows the node number, second column is the derivatives calculated by boundary element method with respect to the design variable and the next columns are finite difference results with different movements of the design variable node in the x-direction ($d=10^{-2}$ to 10^{-6}). It seems that the results of finite difference with movements ($d=10^{-4}$ to 10^{-6}) are more similar and reliable. The same results of Tables (4.1) to (4.3), but only for the nodes along the edge BC are plotted graphically in Figs. (4.3) to (4.5), respectively. The agreement between FD and AN is very good. For example, due to the symmetric boundary conditions the edge BC is symmetric with respect to the node 17 or point (M), so the derivatives of the nodes for the upper and lower part of this edge should be symmetric. This can be seen in the results calculated by analytical sensitivity analysis but not by using finite difference method.

It should be mentioned that the computational results of finite difference are obtained using double precision (32 digits) because with single precision the results changes dramatically when the movement (d) of design variable becomes smaller. The other method is carried out with single precision (8 digits), which makes a considerable reduction in computer time.

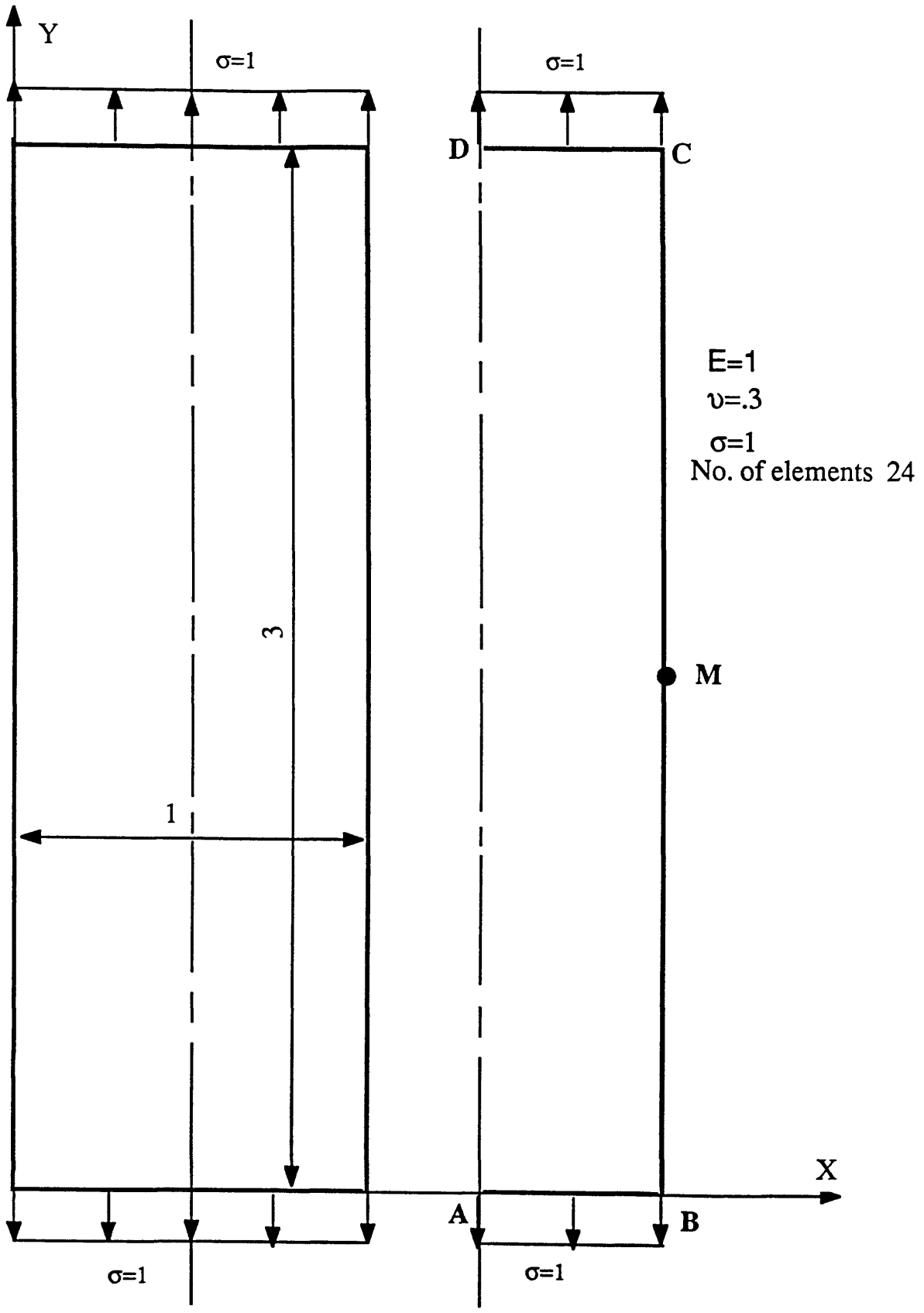


Fig. 4.1 Plate under tension

DATABASE: Y0
VIEW : No stored VIEW
Task: Analysis Cases
Model: 1-FE_MODEL1

UNITS : SI
DISPLAY : No stored OPTION
Associated Workset: 1-WORKING_SET1

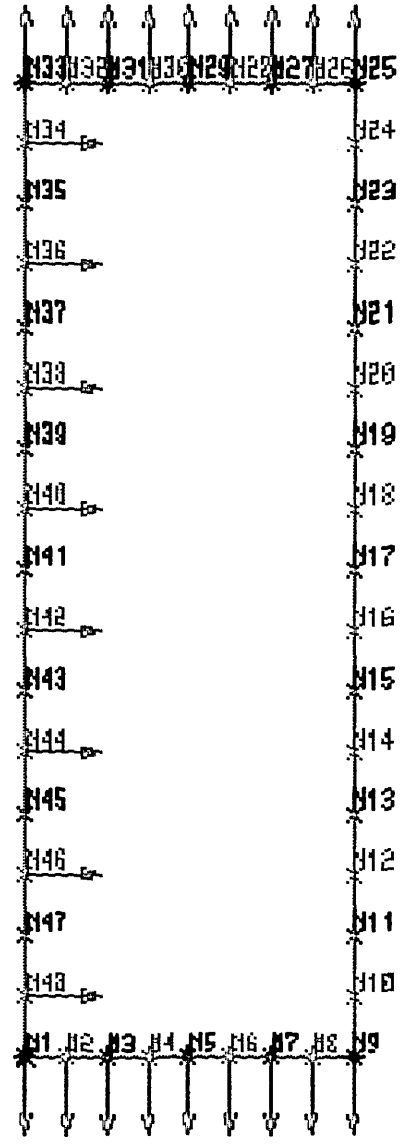


FIG. 4.2 Boundary element mesh of the right half of the plate

ODE NO.	AN	FD d=0.01	FD d=0.001	FD d=0.0001	FD d=0.00001	FD d=0.000001
1	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
2	0.006553	0.022055	0.009173	0.007901	0.007774	0.007761
3	0.012424	0.039421	0.016866	0.014638	0.014416	0.014394
4	0.016967	0.053779	0.022980	0.019939	0.019635	0.019605
5	0.019703	0.065582	0.027230	0.023443	0.023065	0.023027
6	0.020415	0.075094	0.029437	0.024928	0.024478	0.024433
7	0.019233	0.082819	0.029688	0.024441	0.023917	0.023864
8	0.016664	0.089770	0.028432	0.022373	0.021769	0.021708
9	0.013544	0.098531	0.026589	0.019482	0.018773	0.018702
10	0.015921	0.073545	0.025758	0.021037	0.020566	0.020519
11	0.021476	0.062440	0.028117	0.024727	0.024389	0.024355
12	0.030756	0.060620	0.034011	0.031384	0.031122	0.031096
13	0.042105	0.064071	0.042358	0.040216	0.040002	0.039980
14	0.053927	0.044941	0.022766	0.020584	0.020366	0.020344
15	0.160974	0.174436	0.160597	0.159249	0.159115	0.159101
16	-0.101875	-0.106928	-0.101278	-0.100727	-0.100672	-0.100667
17	-0.865344	-0.923339	-0.867158	-0.861614	-0.861061	-0.861005
18	-0.101886	-0.069444	-0.066003	-0.065710	-0.065681	-0.065678
19	0.160974	0.182439	0.167595	0.166121	0.165973	0.165959
20	0.053926	0.077732	0.058837	0.056970	0.056783	0.056765
21	0.042105	0.066473	0.045138	0.043032	0.042822	0.042801
22	0.030756	0.059564	0.033065	0.030450	0.030189	0.030163
23	0.021476	0.058836	0.024409	0.021011	0.020672	0.020638
24	0.015921	0.069246	0.021289	0.016554	0.016082	0.016034
25	0.013544	0.096243	0.024289	0.017183	0.016474	0.016403
26	0.016665	0.086501	0.025081	0.019016	0.018411	0.018351
27	0.019233	0.079238	0.026004	0.020748	0.020224	0.020171
28	0.020415	0.071632	0.025876	0.021359	0.020908	0.020863
29	0.019703	0.062495	0.024055	0.020262	0.019883	0.019845
30	0.016967	0.051186	0.020310	0.017263	0.016959	0.016928
31	0.012424	0.037395	0.014770	0.012537	0.012314	0.012292
32	0.006553	0.020764	0.007826	0.006549	0.006422	0.006409
33	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
34	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
35	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
36	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
37	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
38	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
39	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
40	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
41	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
42	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
43	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
44	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
45	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
46	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
47	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
48	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

Table 4.1 Displacement (u) derivatives for analytic differentiation (AN) and using finite difference (FD) at different movements (d) of the design variable in the x-direction

ODE NO.	AN	FD d=0.01	FD d=0.001	FD d=0.0001	FD d=0.00001	FD d=0.000001
1	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
2	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
3	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
4	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
5	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
6	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
7	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
8	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
9	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
10	0.014167	0.026984	0.015733	0.014624	0.014514	0.014503
11	0.031148	0.045859	0.033384	0.032158	0.032035	0.032023
12	0.054004	0.069804	0.056845	0.055575	0.055448	0.055435
13	0.088474	0.108676	0.095002	0.093665	0.093532	0.093518
14	0.128016	0.145440	0.126869	0.125054	0.124873	0.124854
15	0.142434	-0.023538	-0.076605	-0.081774	-0.082290	-0.082341
16	1.051841	1.093791	1.048186	1.043710	1.043263	1.043218
17	0.000036	-0.222460	-0.225174	-0.225266	-0.225274	-0.225274
18	-1.051841	-1.102356	-1.060586	-1.056468	-1.056057	-1.056016
19	-0.142434	-0.173214	-0.142144	-0.139122	-0.138821	-0.138791
20	-0.128016	-0.147534	-0.130211	-0.128525	-0.128357	-0.128340
21	-0.088474	-0.104738	-0.091075	-0.089741	-0.089608	-0.089595
22	-0.054004	-0.069848	-0.057109	-0.055861	-0.055736	-0.055724
23	-0.031148	-0.046815	-0.034530	-0.033322	-0.033201	-0.033189
24	-0.014167	-0.028408	-0.017319	-0.016226	-0.016117	-0.016106
25	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
26	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
27	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
28	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
29	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
30	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
31	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
32	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
33	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
34	-0.000550	0.002251	-0.003518	-0.004086	-0.004143	-0.004149
35	0.000882	0.003698	-0.004341	-0.005132	-0.005211	-0.005219
36	0.005781	0.007570	-0.001375	-0.002254	-0.002342	-0.002351
37	0.014313	0.014710	0.005707	0.004823	0.004735	0.004726
38	0.023769	0.022847	0.014498	0.013679	0.013597	0.013589
39	0.027730	0.025566	0.018762	0.018096	0.018029	0.018022
40	0.019480	0.015681	0.011543	0.011138	0.011098	0.011094
41	0.000000	-0.006304	-0.006890	-0.006945	-0.006950	-0.006951
42	-0.019479	-0.029023	-0.026035	-0.025739	-0.025709	-0.025706
43	-0.027729	-0.040591	-0.034905	-0.034342	-0.034286	-0.034280
44	-0.023767	-0.039358	-0.032122	-0.031404	-0.031332	-0.031325
45	-0.014311	-0.031472	-0.023572	-0.022787	-0.022709	-0.022701
46	-0.005779	-0.022846	-0.014924	-0.014137	-0.014058	-0.014050
47	-0.000881	-0.015796	-0.008591	-0.007875	-0.007804	-0.007797
48	0.000551	-0.009633	-0.004387	-0.003867	-0.003815	-0.003810

Table 4.2 Displacement (v) derivatives for analytic differentiation (AN) and using finite difference (FD) at different movements (d) of the design variable in the x-direction

ODE NO.	AN	FD d=0.01	FD d=0.001	FD d=0.0001	FD d=0.00001	FD d=0.000001
1	-0.006574	-0.164893	-0.063565	-0.053551	-0.052552	-0.052452
2	-0.003452	-0.080105	-0.032039	-0.027286	-0.026811	-0.026764
3	0.007357	-0.058729	-0.020682	-0.016915	-0.016539	-0.016502
4	0.028076	-0.028187	0.001590	0.004541	0.004836	0.004865
5	0.050616	0.000137	0.023019	0.025291	0.025518	0.025540
6	0.072585	0.024041	0.044998	0.047082	0.047290	0.047311
7	0.090491	0.060582	0.064952	0.065399	0.065444	0.065448
8	0.094108	0.054397	0.072833	0.074668	0.074851	0.074869
9	0.079820	0.189740	0.076727	0.065582	0.064470	0.064358
10	0.095962	0.122291	0.089024	0.085754	0.085427	0.085395
11	0.109061	0.083468	0.092597	0.093514	0.093606	0.093615
12	0.176482	0.167512	0.164315	0.164020	0.163991	0.163988
13	0.284413	0.495925	0.520110	0.522464	0.522699	0.522722
14	0.166126	-0.352570	-0.457618	-0.467839	-0.468858	-0.468960
15	5.824484	5.575272	5.348753	5.325556	5.323231	5.322999
16	-0.438391	-0.357349	-0.379387	-0.380963	-0.381114	-0.381129
17	-12.514730	-8.752272	-10.899513	-11.098676	-11.118413	-11.120385
18	-0.438611	0.304531	0.238256	0.231404	0.230717	0.230648
19	5.824589	5.016939	4.791798	4.769189	4.766928	4.766702
20	0.166125	0.182601	0.136183	0.131683	0.131234	0.131189
21	0.284414	0.245792	0.246393	0.246477	0.246485	0.246486
22	0.176483	0.154463	0.150787	0.150451	0.150418	0.150415
23	0.109060	0.081359	0.090780	0.091730	0.091825	0.091835
24	0.095962	0.124840	0.092079	0.088858	0.088537	0.088505
25	0.079818	0.218356	0.107855	0.096944	0.095855	0.095746
26	0.094108	0.064707	0.084210	0.086146	0.086340	0.086359
27	0.090488	0.081380	0.087593	0.088213	0.088275	0.088281
28	0.072584	0.046801	0.069626	0.071883	0.072109	0.072131
29	0.050611	0.027831	0.052830	0.055297	0.055544	0.055568
30	0.028074	0.002068	0.034041	0.037193	0.037507	0.037539
31	0.007353	-0.021708	0.018969	0.022976	0.023376	0.023415
32	-0.003455	-0.041714	0.009036	0.014035	0.014533	0.014583
33	-0.006582	-0.089749	0.017172	0.027700	0.028751	0.028856
34	-0.017289	-0.041662	-0.003602	0.000143	0.000517	0.000554
35	-0.032809	-0.040397	-0.020096	-0.018103	-0.017904	-0.017884
36	-0.058271	-0.057386	-0.043861	-0.042536	-0.042403	-0.042390
37	-0.084584	-0.077197	-0.069813	-0.069093	-0.069021	-0.069014
38	-0.045981	-0.043147	-0.042403	-0.042330	-0.042323	-0.042322
39	0.018370	0.013706	0.006929	0.006269	0.006203	0.006196
40	0.122107	0.110333	0.096346	0.094979	0.094843	0.094829
41	0.203546	0.188222	0.168641	0.166722	0.166531	0.166512
42	0.122105	0.119907	0.105850	0.104466	0.104328	0.104314
43	0.018368	0.030254	0.023537	0.022867	0.022800	0.022794
44	-0.045983	-0.033217	-0.032264	-0.032178	-0.032170	-0.032169
45	-0.084584	-0.075583	-0.068027	-0.067289	-0.067216	-0.067208
46	-0.058269	-0.067074	-0.054045	-0.052762	-0.052634	-0.052622
47	-0.032807	-0.058999	-0.039747	-0.037846	-0.037656	-0.037637
48	-0.017286	-0.073454	-0.037520	-0.033967	-0.033612	-0.033577

Table 4.3 Stress (σ_e) derivatives for analytic differentiation (AN) and using finite difference (FD) at different movements (d) of the design variable in the x-direction

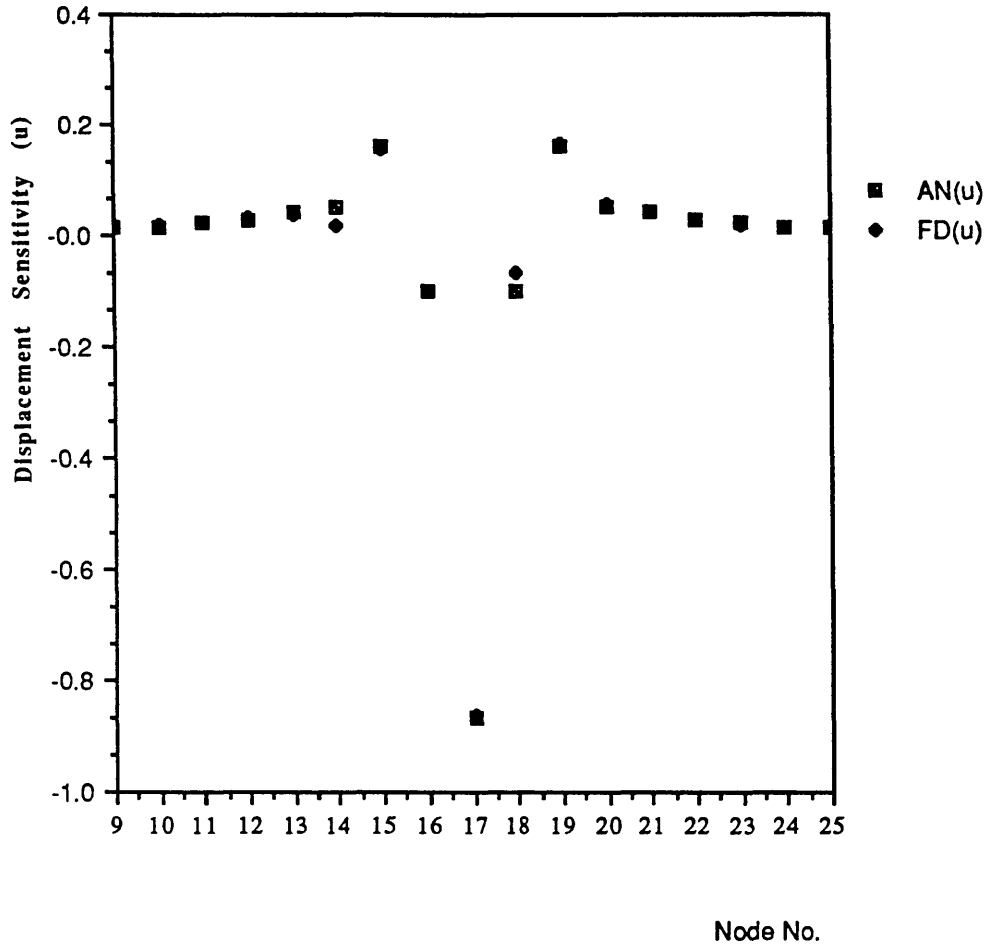


FIG. 4.3 Displacement (u) derivatives at the nodes along the edge BC

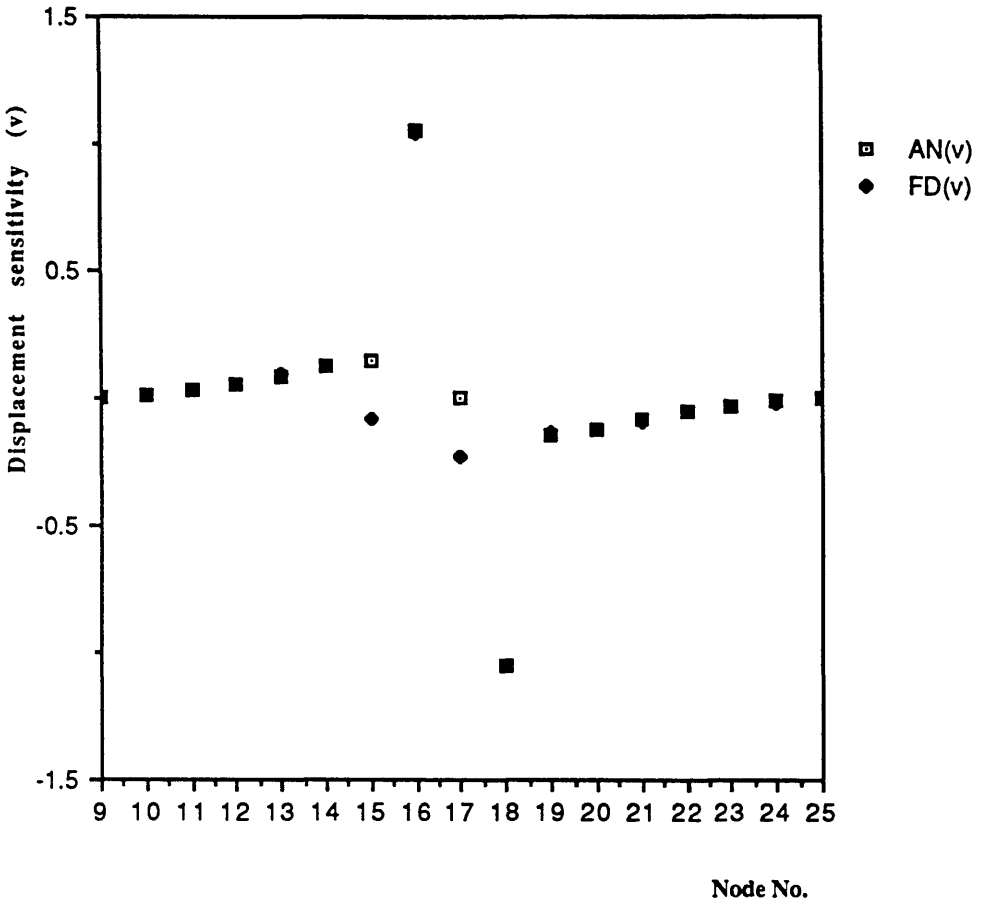


FIG. 4.4 Displacement (v) derivatives at the nodes along the edge BC

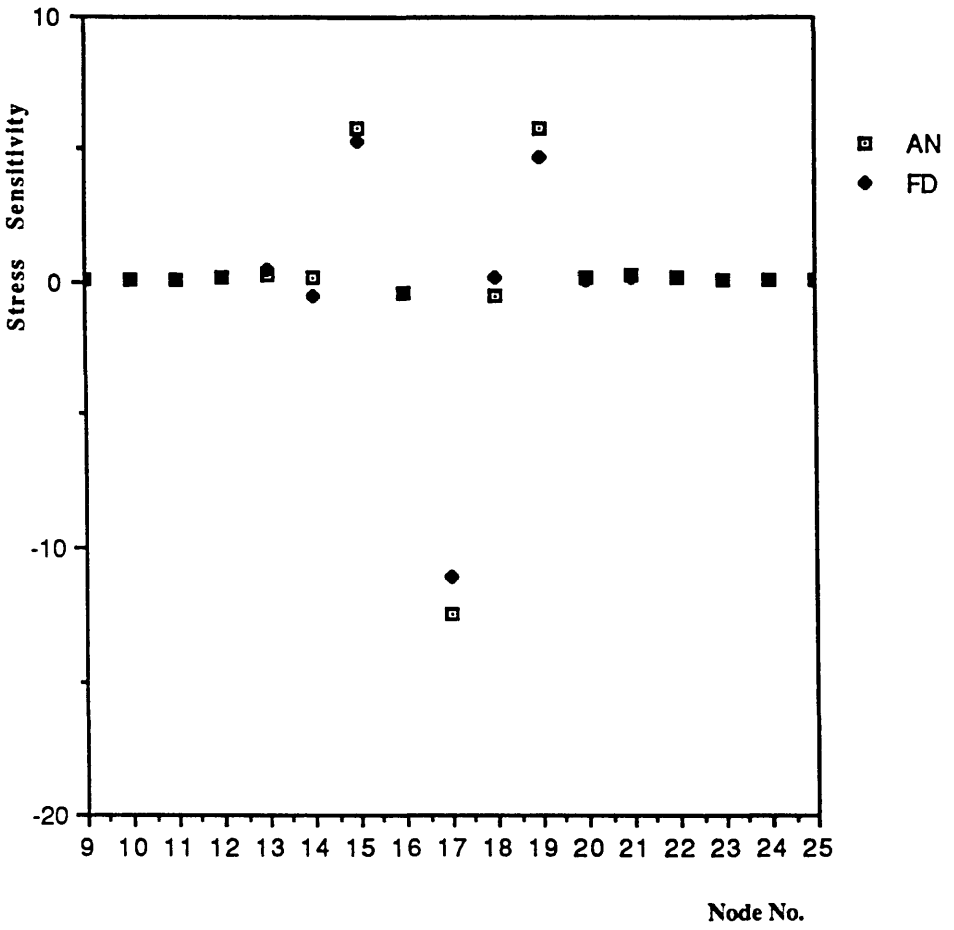


FIG. 4.5 Stress derivatives at the nodes along the edge BC

CHAPTER 5

SHAPE OPTIMIZATION

5.1 - INTRODUCTION

Structural optimization, in the classical sense, has been focused on sizing design variables, such as plate thickness, bar cross-sectional areas, member sizes or width of cross sections of a model in which the geometry remains unchanged.

Shape optimization is more complex than structural optimization with sizing design variables, because it changes the whole shape of the boundary while there are some constraints on displacements, stresses, natural frequencies, etc.. Since the shape of the object is continuously changing, it needs a good consideration during the optimization process, for changing the boundary with a continuous boundary description, maintaining an adequate finite element or boundary element mesh, having an accurate sensitivity analysis for calculating derivatives of objective and constraint functions, imposing necessary constraints to have a realistic shape and implying the optimization methods accurately.

5.2 - LITERATURE SURVEY

Development of optimum structural design, in chronological order, can be categorized in four independent major areas [Zienkiewicz & Campbell, 1973] :

1) Theory of layout
2) Simultaneous mode of failure

3) Optimality criteria method

4) Mathematical programming formulations

Theory of layout was the first approach for volume minimization of structures with uniaxial members. The basic theory was established by Maxwell [1869], but then was developed by Michell [1904]. An analytical method was used for solving the optimization problem. Since the theorem was applied without meaningful constraints, the results were unrealistic. This theory was reconsidered by Cox [1965] and Hemp [1958].

Simultaneous mode of failure assumes when each element of a composite structure is at its limit of strength, the design is optimum. Here 'simultaneous' means single load condition, so it was suitable for simple situations and had limited applications. For more details see [Shanley, 1960], [Gerard, 1956], [Cox, 1965]. The fully stressed design, which these days is being used for optimum structural design, is the more complex form of this method but with more than one load condition. It means, each member of the structure carries the maximum allowable load. However, from mathematical point of view, it is not an optimum solution but sometimes is an efficient and economical method of calculating low-weight design for complex structures.

Optimality criteria method, was first used for structural volume minimization in the early 1960s. Some criteria can be derived mathematically, for example by using some assumptions based on variational methods, the optimum shape can be achieved. Prager [1971] and Taylor [1969] have been the pioneers in the development of much of this work.

Mathematical programming, is concerned with the determination of a maximum or minimum of a function of several variables, which also has to satisfy a number of constraints and is fully described in chapter 2. It was first applied to structural optimization in the late 1950s, by Livesley [1956], Pearson [1958] and Schmit [1960]. Since then, many authors have published papers, in which they have used numerical optimization methods for optimum structural design.

Due to the needs of engineering design, shape optimization employing mathematical

programming techniques is of relatively recent interest. The early work was on arch dam design [Sharpe, 1969] and the optimum shape of a structure by Zienkiewicz and Campbell [1973]. Since then many authors have published papers , in which, mostly two- or three-dimensional finite element methods have been coupled with mathematical programming techniques. The design sensitivity analysis has been performed by finite difference schemes to design the optimum shape of a variety of structures or mechanical components, for example; dams, plates, shells, etc..

There are some difficulties in using the finite element method, which are mostly encountered in three broad areas. First, because of the continuously changing finite element model, the accuracy of the finite element analysis may be inadequate during the optimization process. Second, since during the optimization process the shape of the boundary is changing, then the mesh regeneration during the process is very costly. Finally, and the most important one is the sensitivity analysis in calculation of the derivatives by approximate methods which are time-consuming and not very accurate.

The boundary integral equation (BIE) or boundary element method, which is fully discussed in chapter 3, has been established in recent years as a powerful technique in linear elastic stress analysis. The BIE approach offers continuous interior modelling within the solution domain of a problem, giving high resolution of stresses and displacements. It appears to be more convenient for handling shape optimization problems, because the BIE method is a boundary oriented technique which can overcome the above problems associated with finite elements, and will be discussed in more detail in the next sections.

Recently, the boundary element method has been used in the shape optimization technique for example; optimization of shafts [Soares et al, 1984] and bidimensional structures [Eizadian and Trompette, 1985].

Here shape optimal design of two-dimensional elastic components and structures is formulated using mathematical programming coupled with boundary element method.

5.3 - PROBLEM STATEMENT

The shape optimization problem can be stated mathematically as

$$\text{Min } (F(X_K)) \qquad K=1,2,....,n \qquad (5.1)$$

$$\text{Subject to } g_i(X_K) \leq 0 \qquad i=1,2,..... m \qquad (5.2)$$

$$X_k^L \leq X_K \leq X_K^U \qquad (5.3)$$

in which F is the objective function and g_i the constraint function describing the ith structural response. X is a vector of design variables which defines the shape of the variable part of the structure. X_k^L and X_k^U are the lower and the upper limits of the shape variables which are linear constraints and could be included in the inequality constraint set given by Eq. (5.2). It is usually convenient to treat them separately because they define the region of search for the optimum. n is the number of design variables and m is the number of behaviour constraints. In [Braibant & Fleury, 1984], [Queau & Trompette, 1980], [Kritensen & Madsen, 1976] and [Bennet and Botkin, 1984] some geometric constraints have also been introduced to deal with the geometrical requirements such as keeping slope continuity on the boundary during the optimization process

5.4 - SELECTION OF THE OBJECTIVE FUNCTION

In shape optimization problems, there are different ways of choosing the objective function. Usually weight (mass) is selected as the objective function and this is in general a nonlinear function of the design variables. In [Ramakrishnan & Francavilla, 1975], [Epsing, 1985] the objective function is weight while the constraints are "Side Constraints" to control geometry during optimization and "Behaviour Constraints" to keep stresses below some limiting value. In [Botkin, 1981], [Bennet and Botkin, 1984]

and [Braibant and Fleury, 1984] the objective function is weight while satisfying certain constraints upon stresses, displacements and frequencies besides the geometric constraints.

The maximum equivalent stress along a given or the whole part of the boundary can be considered as an objective function while the constraints are on geometry; for example in [Queau & Trompette, 1980], [Kristensen & Madsen, 1976], [Pedersen & Laursen, 1982-3], [Spoerl, 1986] and [Francavilla et al, 1975].

$$F(X) = \text{Max} [\sigma_e] \tag{5.4}$$

The objective function can be the difference between maximum and minimum tangential stresses [Bhavikatti & Ramakrishnan, 1980] where maximum and minimum stresses are for all the sampling points.

$$F(X) = \sigma_{\theta_{MAX}} - \sigma_{\theta_{MIN}} \tag{5.5}$$

An alternative objective function has been considered, the square of the deviation of the equivalent stresses from a desired uniform mean stress (σ_a), sometimes called stress levelling.

$$F(X) = \oint_A (\sigma_e - \sigma_a)^2 \, dA \tag{5.6}$$

where A is that part of the boundary which is optimized [Francavilla et al, 1975], [Bhavikatti & Ramakrishnan, 1980] , [Weck & Steinke, 1983-84] and [Yang & Choi, 1985].

Another possibility is the weighted objective function [Bhavikatti & Ramakrishnan, 1980]

$$F(X) = 0.5 \frac{W}{W_o} + \frac{\int_A (\sigma_e - \sigma_a)^2 \, dA}{\int_A (\sigma_0 - \sigma_a)^2 \, dA} \qquad (5.7)$$

where W is the weight of the structure, W_0 is the weight for the initial shape, σ_0 is the equivalent stress for the initial shape and σa is the mean stress. In this case, stresses and weight together have been considered as the objective function.

The purpose of using different type of stresses in the objective function is to decrease the peaks of stress concentration at the modified boundary.

The objective function and constraint functions (except side constraints) are in most cases highly nonlinear with respect to the design variables, and must be handled numerically.

5.5 - SHAPE REPRESENTATION

The key step in the process of shape optimization is to describe the shape of the structure by a suitable function. The use of boundary nodes in the finite element model as shape variables is the earliest method [Haftka & Grandhi, 1986], [Braibant & Fleury, 1984], [Parasad & Haftka, 1979]. The advantages with this method are that the approach is simple, no mesh generation during the optimization procedure is needed because the coordinates of all the nodes of the variable part of the boundary are considered as the design variables. The disadvantages are the increase of design variables which causes high costs and difficulties in the solution of optimization problem, and discontinuity between boundary nodes which leads to unrealistic designs. Also it would be difficult to have an adequate mesh during the optimization process.

Another approach is the use of polynomials to represent the shape of the boundary [Botkin, 1981], [Imam, 1982], [Bhavikatti & Ramakrishnan, 1980], [Wassermann,

1983-4], [Kristensen & Madsen, 1976], [Ramakrishnan & Francavilla, 1975], [Frankavilla et al, 1975] and [Pedersen & Laursen, 1982-3] which reduces the number of shape variables but results in oscillatory boundary shapes for sophisticated boundaries with higher order polynomials. Thus they are an inappropriate tool for the designer.

Using of spline or spline blending functions is another method of boundary representation. It eliminates the difficulty found with complex boundaries defined by higher order polynomials, because splines are piecewise polynomial functions which maximize smoothness. A variety of splines have been used for shape representation, for example; cubic spline [Weck & Steinke, 1983-84], [Yang & Choi, 1985], [Luchi et al, 1980] and Ferguson & B-Spline [Braibant & Fleury, 1984], [Epsing, 1985] and [Haftka & Grandhi, 1986]. Details of various kinds of splines can be found in [Faux & Pratt, 1979], [Rogers & Adams, 1976], [de Boor, 1978], [Schumaker, 1981], [Prenter, 1975], [Powell, 1981] and [Riesenfeld, 1973].

In this section, the implementation of cubic and Hermitian cubic splines is described and discussed in detail and the advantages in using them are explained.

5.5.1 -Cubic and Hermitian Cubic Splines

A cubic spline function is a combination of piecewise cubic curves whose the first and second derivatives are continuous at the ends of each curve. Let f(x) be defined on the interval $a \leq x \leq b$ and be subdivided into subintervals with common end points $a=x_0<x_1<x_2.....<x_n=b$ and let k_0 and k_n be any two given numbers which define the slopes at first and end points of the spline. Then there exists one and only one cubic spline g(x) which satisfies

$$g(x_0) = f(x_0) \quad , \quad g(x_1)=f(x_1) \quad ,\ldots\ldots\ldots, g(x_n)=f(x_n) \tag{5.8}$$

$$g'(x_0) =k_0 \quad , \quad g'(x_n)=k_n \tag{5.9}$$

and has continuous second derivatives with unique slopes $(k_1,k_2,\ldots..k_{n-1})$ at the end of each subinterval. These slopes can be obtained by solving the system of (n-1) linear equations of the form [Kreyszig, 1979]

$$C_{j-1} K_{j-1} + 2(C_{j-1} +C_j) K_j + C_j K_{j+1} = 3 \left[C_{j-1}^2 \nabla f_j + C_j^2 \nabla f_{j+1}\right] \tag{5.10}$$

Where $\nabla f_j = f(x_j) - f(x_{j-1})$, $\nabla f_{j+1} =f(x_{j+1}) - f(x_j)$

and $C_j = \dfrac{1}{x_{j+1} - x_j}$

Once all the K-values have been calculated, the particular cubic spline which fits each individual span of the spline is obtained.

The main problem with the cubic spline is that since they may be defined as functions of x or y, they are axis-dependent and since the slopes at both ends of the spline should be finite then they can not cope with a vertical tangent.

One way of avoiding this problem is to use Hermite interpolation for the piecewise curves (elements). Each Hermitian cubic element has two nodes and geometry is defined in terms of the cubic shape functions of intrinsic coordinates of the element where

$$M^1(\zeta) = \frac{1}{4}(1+\zeta)^2(2 - \zeta) \tag{5.11}$$

$$N^1(\zeta) = -\frac{1}{4}(1+\zeta)^2(1 - \zeta) \tag{5.12}$$

$$M^2(\zeta) = \frac{1}{4}(1-\zeta)^2(2+\zeta) \tag{5.13}$$

$$N^2(\zeta) = \frac{1}{4}(1-\zeta)^2(1+\zeta) \tag{5.14}$$

the intrinsic coordinate ζ varies form 1 at node 1 to -1 at node 2, Fig.(5.1). The cartesian coordinates $x_i(\zeta)$ of any point of the element S_b are defined in terms of $x_i(b,c)$ the coordinates, and $H_i(b,c)$ the gradients $dx_i/d\zeta$ of the cth node, Eq.(5.15), [Faux & Pratt, 1979] and [Watson, 1982].

$$x_i(\zeta)=\sum_{c=1}^{2}\left[M^c(\zeta)x_i(b,c) + N^c(\zeta)H_i(b,c)\right] \tag{5.15}$$

Consider a continuous function f(x,y) passes through n points (1,2,...,n) with coordinates (x_1, y_1), (x_2, y_2),.....,(x_n, y_n). There exists one and only one Hermitian cubic spline $g(x(\zeta), y(\zeta))$, where $H_1(l_1, m_1)$, $H_n(l_n, m_n)$ are any given numbers for derivatives of the first and end points of the spline with respect to the intrinsic coordinate ζ, by requiring that

$$g_1(x(\zeta), y(\zeta))=f_1(x, y) \quad , \quad g_2(x(\zeta), y(\zeta))=f_2(x, y),.....,g_n(x(\zeta), y(\zeta))=f_n(x,y) \tag{5.16}$$

and

$$\left[\left(\frac{dx}{d\zeta}\right)_1=l_1 , \left(\frac{dy}{d\zeta}\right)_1=m_1\right], \left[\left(\frac{dx}{d\zeta}\right)_n=l_n, \left(\frac{dy}{d\zeta}\right)_n=m_n\right] \tag{5.17}$$

and has continuous second derivatives with respect to the intrinsic coordinate ζ with unique derivatives (H_2 (l_2,m_2) , H_3 (l_3,m_3) , ...,H_{n-1} (l_{n-1},m_{n-1})) at the end of each subinterval.

The proof is by definition $g(x(\zeta), y(\zeta))$ has continuous second derivatives with respect to intrinsic coordinate ζ. This gives the conditions

$$\left[\left(\frac{d^2x_i}{d\zeta^2}\right)_k\right]_{S_b^{K-1}} = \left[\left(\frac{d^2x_i}{d\zeta^2}\right)_k\right]_{S_b^{K}} \tag{5.18}$$

where S_b^{k-1} and S_b^{k} are two consecutive cubic curves of the spline. Using Eq.(5.15) with the conditions (5.17) and (5.18) , the system of 2(n-1) linear equations take the form

$$l_{k-1} + 4l_k + l_{k+1} = \frac{3}{2}(x_{k-1} - x_{k+1}) \tag{5.19}$$

$$m_{k-1} + 4m_k + m_{k+1} = \frac{3}{2}(y_{k-1} - y_{k+1}) \tag{5.20}$$

where k=2,...,n-1. The system of 2(n-1) linear equations has a unique solution (l_2, m_2), (l_3,m_3),.....,(l_{n-1},m_{n-1}) because the matrix of coefficients of the system is tridiagonal (having non-zero elements only on the leading diagonal and to either side of it), the coefficient determinant cannot be zero [Goult et al, 1974]. Hence using these unique values of (l_2, m_2), (l_3, m_3), ...,(l_{n-1}, m_{n-1}), Eq.(5.15) gives the particular cubic Hermite polynomials which fits each individual element of the spline.

By virtue of the continuity condition imposed, in the construction of a Hermitian cubic spline over n points, the only data requirements are the function values (x_i, y_i) at the points together with extra information about the derivatives of the first and final points.

Sometimes derivatives of one or both of the end points of the spline are not specified then There is a possibility to consider the first and/or the last element of the spline as quadratic. Since quadratic function has a constant second derivative, then in this case two extra equations are

$$\left[\left(\frac{d^2x_i}{d\zeta^2}\right)_1\right]_{s_b^1} = \left[\left(\frac{d^2x_i}{d\zeta^2}\right)_2\right]_{s_b^1} \tag{5.21}$$

$$\left[\left(\frac{d^2x_i}{d\zeta^2}\right)_{n-1}\right]_{s_b^{n-1}} = \left[\left(\frac{d^2x_i}{d\zeta^2}\right)_n\right]_{s_b^{n-1}} \tag{5.22}$$

Using Eq.(5.15) with the conditions (5.21) and (5.22),

$$\begin{cases} l_1 + l_2 = x_1 - x_2 \\ m_1 + m_2 = y_1 - y_2 \end{cases} \tag{5.23}$$

and/or

$$\begin{cases} l_{n-1} + l_n = x_{n-1} - x_n \\ m_{n-1} + m_n = y_{n-1} - y_n \end{cases} \tag{5.24}$$

Eqs. (5.23) and (5.24) with Eqs. (5.19) and (5.20) make 2n linear equations with unique solutions for $(l_1,m_1),(l_2, m_2),.....,(l_n,m_n)$.

Fitting the spline for a closed curve (Fig.5.3), since the first and end points of the spline are coincident then $l_1=l_n$, $m_1=m_n$ and

$$\left[\left(\frac{d^2x_i}{d\zeta^2}\right)_1\right]_{s_b^1} = \left[\left(\frac{d^2x_i}{d\zeta^2}\right)_n\right]_{s_b^{n-1}} \tag{5.25}$$

Using Eq.(5.15) with this last condition

$$l_{n-1}+4\,l_1+l_2=\frac{3}{2}(x_{n-1}-x_2) \tag{5.26}$$

$$m_{n-1}+4\,m_1+m_2=\frac{3}{2}(y_{n-1}-y_2) \tag{5.27}$$

These equations with Eqs(5.19) and (5.20) make 2n linear equations to calculate unique derivatives at each point. The only information required from the user in order to fit Hermitian cubic spline through n points for a closed curve, is the coordinates of the points.

Figs.(5.4) and (5.5) show 4 points and 8 points, respectively, on a circle (red) which are approximated by a Hermitian cubic spline (green) which shows a very good agreement.

5.5.2 - Advantages of Hermitian Cubic Spline

The Hermitian cubic spline has two continuous derivatives everywhere, minimum mean curvature and is axis-independent. It also possesses globally controlled properties; i.e. moving any design variable (control point) will change the shape globally not locally. Unlike the B-spline the number of points and the order of that do not change the resulting curve, so it makes them problem-independent and easy to use. Complex geometries can be described by a small set of design variables with good accuracy. It is no longer necessary to use any optimization constraint to avoid unrealistic designs during the optimization procedure.

5.6 - CHOICE OF THE DESIGN VARIABLES

In contrast to structural optimization with sizing design variables, shape optimization uses design variables which are relevant to the boundary shape. In early works, the coordinates of the boundary nodes of the finite element model were used as design variables [Zienkiewicz & Campbell, 1973] and [Braibant & Fleury, 1984] which often resulted in undesirable shapes. Polynomials have been used to describe the boundary shape, with the polynomial coefficients being the design variables [Bhavikatti & Ramakrishnan, 1977], [Pedersen & Laursen, 1983], [Parasad & Emerson, 1984].

A more general approach is to define the boundary as a linear combination of shape functions with the coefficients being the design variables. [Kristensen & Madsen, 1976] defined the boundary using a linear combination of orthogonal functions added to the initial design by treating the coefficients of the functions as design parameters. Dems [1980] also used a formulation describing the boundary using a set of prescribed shape functions and applied it to the simple case of piecewise linear boundaries.

Another way which has been used in conjunction with finite element methods is to use the "design element" concept introduced by Imam [1982]. In this approach, the structure is divided into a few regions. These regions or design elements, are described by a set of key nodes (or master nodes) that control the geometry. Associated with the design element is a set of design variables that specify the location of the key nodes that are allowed to move in the design, for example; [Braibant & Fleury, 1984], [Botkin, 1981]. Bennet and Botkin [1984] describe the drawbacks of this method. First, consuming a significant amount of time building the design element model for each structure. Secondly, any moderate changes in the design required complete rebuilding of the design element model and last, the need for automatic adaptive mesh generation with finite elements. These drawbacks have led researchers to describe structures with a series of both key nodes and boundary design elements, but still they have used the finite element method for sensitivity analysis and solution.

Here, since the boundary element method, as described in Chapters 2 and 3, has been chosen for solution and sensitivity analysis; and Hermitian cubic splines, as described in Sec.(5.5), are selected for shape representation, then the obvious choices

of design variables for shape design optimization are the coordinates of some points on the movable part of the boundary which is going to be optimized. Since a small number of Hermitian cubic elements gives a very good approximation for the boundary description then a set of points of that part of the boundary are selected as design variables. Every two points define a Hermitian cubic element and all of them, together, build a Hermitian cubic spline. For the analysis method as described in chapter 3, two-dimensional quadratic boundary elements are used, therefore during the optimization procedure automatic mesh generation includes the redefinition of two-dimensional quadratic boundary elements of the movable part of the boundary which has been defined by Hermitian cubic spline [Appendix B].

In the programs used in this thesis, the design variables are considered as the cartesian coordinates of some selected nodes on the moving part of the boundary. In some cases, it is desired to use polar coordinates to have better results. This can be obtained by application of the chain rule,

$$\frac{\partial F}{\partial r} = \frac{\partial F}{\partial x}\frac{\partial x}{\partial r} + \frac{\partial F}{\partial y}\frac{\partial y}{\partial r} \tag{5.28}$$

and

$$\frac{\partial F}{\partial \theta} = \frac{\partial F}{\partial x}\frac{\partial x}{\partial \theta} + \frac{\partial F}{\partial y}\frac{\partial y}{\partial \theta} \tag{5.29}$$

For example, when the design variable is the radius of a node on the movable part of the boundary with cartesian coordinates (x,y),at a constant angle (θ), we have

$$\frac{\partial F}{\partial r} = \frac{\partial F}{\partial x}\cos(\theta) + \frac{\partial F}{\partial y}\sin(\theta) \tag{5.30}$$

In this case, the derivatives will be calculated with respect to the cartesian coordinates and then by Eq.(5.30) the derivatives of the objective and constraint functions with respect to the radius can be obtained.

5.7 - BOUNDARY ELEMENT MESH GENERATION

The information required to use the BIE method for stress analysis is input data for the analysis program, which consists of the geometry idealization, the material properties and the loading and boundary conditions. The major point of this is the geometric representation of the structure by a suitable mesh. If this task is to be done manually, it is obviously time-consuming and is liable to errors. Using a Computer Aided Design (CAD) system in the data preparation process can reduce these difficulties. The CAD system can then be used after the analysis to present data graphically so that the results can be quickly assessed.

In recent years, a wide variety of CAD packages have been developed for finite element data preparation, but at present the boundary element method is rarely used in CAD systems. Here an interactive program file in conjunction with the available CAD software I-DEAS (Integrated Design Engineering Software) [SDRC, 1986-88] has been developed as a pre-processor to generate boundary element meshes, create boundary conditions and tabulate physical and material properties, or in general to prepare input data to interface any two-dimensional boundary element analysis programs. Another program file has been also developed as post-processor to display and tabulate analysis results.

For a three-dimensional BIE problem the user is able to use the same software which generates finite elements but instead of generating the whole domain, generates two-dimensional boundary elements only on the surface boundaries.

5.7.1 - Boundary Element Mesh Redefinition

A problem within the shape optimization is the boundary element mesh re-generation during the process. In contrast to the FEM, the discretization of all the domain is not needed and the design variables are restricted to the boundary, which implies an important decrease in time for data preparation and analysis, also avoiding

obtaining a large amount of unnecessary results. A fixed boundary element is not valid, because during the optimization procedure the shape of the boundary changes. One way of handling it is to use an automated mesh generator during the optimization process.

In the two-dimensional case, since the boundary is defined by curves then for boundary element mesh regeneration the length of that part of the boundary which is going to be moved, will be calculated numerically and defined by the same number of elements and nodes in each iteration of optimization process. If there is any concentration of elements at corners, it will be kept the same during the optimization process. This task is done automatically.

For more details about two-dimensional quadratic boundary element mesh generation of a curve with a specified function refer to Appendix B.

5.8- OPTIMIZATION METHODS FOR STRUCTURAL SHAPE OPTIMIZATION

The optimization problem can be solved in principle by any of the non-linear programming methods discussed in chapter 2. Most of the work in recent years for solving the shape design problems is based on employing mathematical programming coupled with finite element analysis of the structure. The optimization methods which have been applied to solve the shape optimization problems in the last two decades can be categorized as follows:

I) Sequential Linear Programming
II) Penalty Function Methods
III) Feasible Direction Methods
IV) Calculus of Variations and Optimality Criteria Methods
V) Pattern Transformation Methods
VI) Photo-Elasticity Technique

I - Sequential Linear Programming

Most shape optimization problems are nonlinear with respect to the design variables. This method involves linearizing the objective and constraint functions and then obtaining the solution to this linear approximation using the linear programming methods of sec.(2.8.1). Using this approximate solution, again linearize about this point and solve the new linear programming problem; repeating the process until a precise solution is achieved which is discussed in Sec.(2.8.2.2).

With these approximations, the shape optimization problem is reduced to sequence of linear programming problems; see [Zienkiewicz & Campbell, 1973], [Braibant & Fleury, 1984], [Wassermann, 1983-84], [Kristensen & Madsen, 1976], [Pedersen & Laursen, 1982-83], [Bhavikatti & Ramakrishnan, 1980] and [Ricketts & Zienkiewicz, 1984].

II - Penalty Function Methods

The penalty function method or sequential unconstrained minimization technique (SUMT), transforms the constrained problem into a sequential unconstrained optimization problem and has the same optimum solution as the constrained problem, as discussed in Sec.(2.8.2.1). The particular technique considered for use in the shape optimization has been the interior penalty (Barrier) function method, because the search is restricted to the feasible region and the design would be realistic and the unconstrained optimization usually is carried out by the variable metric method; see [Queau & Trompette, 1980], [Ramakrishnan & Francavilla, 1975], [Riccketts & Zienkiewicz, 1984] and[Francavilla et al, 1975].

III - Feasible Direction Method

This method determines a usable-feasible direction in which the design point may be moved in the design space. This direction is 'feasible' because it does not violate, at lest over an infinitely small step, any of the constraints; and 'usable' because it results

in a reduction of the objective function. Using gradient information, the direction is found and along this direction, a one-dimensional search is next performed. This technique is very suitable for highly non-linear shape optimization problems because the design point is always feasible; see [Botkin, 1981], [Braibant & Fleury, 1984], [Bennet & Botkin, 1984] and [Spoerl, 1986].

IV - Calculus of Variations and Optimality Criteria Methods

This method is another way of handling shape optimization problems. By using a variational method, rigorous optimality criteria are derived mathematically and according to these relations the optimum shape will be determined. For example, it can be assumed that the tangential stress or the strain energy density should be uniform along the boundary; see [Dems & Mroz, 1978], [Na et al, 1983,1984], [Dems, 1980], [Cherpanov, 1974] and [Weck & Steinke, 1983-84]. This method can be either rigorous or intuitive [Haftka & Grandhi, 1986].

The next two methods, the pattern transformation method and the photo-elasticity technique are included in the optimality criteria methods but usually they are categorized separately because of their unique resizing approaches.

V - Pattern Transformation Method

An alternative for solving shape optimization problems is the pattern transformation method, which has been introduced by [Oda, 1977], [Oda and Yamazaki, 1977, 1979, 1981]. This is an iterative method for transforming the shape of the boundary based on the stress ratio in the finite elements at the boundary. It consists of two steps. First, the stress ratio in the relevant elements is calculated and then the size of the elements is changed to approach an optimum shape by the proportional transformation of the finite elements constituting the boundary. In some cases instead of stress ratio, stress energy ratio has been considered, see [Seguchi & Tada, 1978], [Tada & Seguchi, 1981] and

[Hemada et al, 1980].

VI - Photo-Elasticity Technique

Stress concentrations due to the presence of holes and notches always have been a major problem for the designer. Recently, these sort of problems have been considered experimentally for the purposes of shape optimization. A step-by-step procedure may be used to modify the boundary with photo-elastic experiments; for example by removing material from low stress regions, in order to obtain uniform stress conditions; see [Durelli & Rajiah, 1979], [Durelli et al, 1981], [Rajaiah & Durelli, 1981] and [Rajaiah & Kumar, 1982]. Schnack[1979] and [1983] developed a procedure for the numerical equivalent of the experimental approach.

5.9 - OPTIMIZATION OF SHAPE TO MINIMIZE STRESS PEAKS

All engineering structures and components have to be strong and sufficiently rigid. When designing such structures and components, various factors have to be taken into consideration. One of these is the localized high stresses around holes, notches, etc.. This localization of high stresses is generally known as the stress concentration measured by the stress concentration factor

$$\eta = \frac{\sigma e_{max}}{\sigma_m} \tag{5.31}$$

where σe_{max} is the maximum equivalent stress (depends on failure criteria) at a point in the region and σ_m is a standard stress with respect to which the stress-concentration factor is defined.

One way of optimizing stress peaks is to minimize the maximum equivalent stress (Eq.5.4) and another alternative is to try to make uniform the equivalent stresses along a given part of the boundary. The second one can be treated as a mathematical programming problem where the objective function of the optimization problem is the square of the deviation of the equivalent stresses from a desired stress (Eq. 5.6) while there are some side constraints upon the design variables. In this case the objective function is highly nonlinear and the constraints are linear.

As the optimum is not likely to be at the intersection of fixed boundaries, sequential linear programming (SLP), although economical, is not really suited to the problem The penalty function method appears to be more desirable, because it changes the problem into an unconstrained problem. Since the optimum is not near a constraint; it is not necessary to repeat the process with different penalty parameters. As discussed in Sec.(2.8.2.1), the exterior penalty function method is very suitable here because the penalty term is continuous and also exhibits continuity of slopes at the point of transition. For unconstrained minimization, the Broydon-Fletcher-Goldfarb-Shanno technique, which has proved well able to deal with non-linear objective functions, is used, followed by an unconstrained one-dimensional search. It should be mentioned here that since the constraint violation has no meaning for the geometric side constraints, then if the minimization approaches a constraint, implying its exact satisfaction, the constraint may be incorporated in the objective function and the problem can be solved as one of unconstrained optimization with (n-1) design variables.

In order to illustrate the applications of the boundary integral equation method for shape optimization to smooth stress peaks for two-dimensional linear elastic problems, the STRESOPT analysis computer program has been developed to study a number of problems. This program is an iterative procedure. Each iteration involves three steps :

I) An analysis of the stresses for a design
II) Sensitivity analysis corresponding to possible changes in this design
III) The decision of redesign (optimization step)

The BIE method is particularly well suited not only to the stress analysis step, but

also for the sensitivity analysis. The flow chart of this program can be seen in Fig. (5.6) and some test problems are described in chapter 6. These problems have been solved by finite element method in different papers and the results are compared.

The optimizer part of the program which uses exterior penalty function method has been written generally, so with small changes can be used for any other problems.

It should be noted that the Hermitian cubic splines are used for shape representation while the design variables are first and end nodes of each Hermitian cubic curve and the BIE method uses quadratic boundary elements for its analysis. The program has an automatic mesh generator and the iterative optimization process is executed automatically without user intervention.

5.10 - WEIGHT MINIMIZATION

Minimum weight design of structures or mechanical components, as long as the stresses are within some allowable range, always has been a major desire for engineers. The structures are strong enough while a minimum quantity of materials for their manufacture is used.

The minimization of weight to meet imposed constraints such as maximum allowable stresses, displacements, frequencies,etc. is a mathematical programming exercise. The general problem to be dealt with in this thesis is the minimization of structural weight while satisfying certain constraints upon stresses and geometry. The problem can be stated mathematically as follows :

$$\text{Minimize} \quad : \quad W=f(X_j) \qquad j=1,....,n \tag{5.32}$$

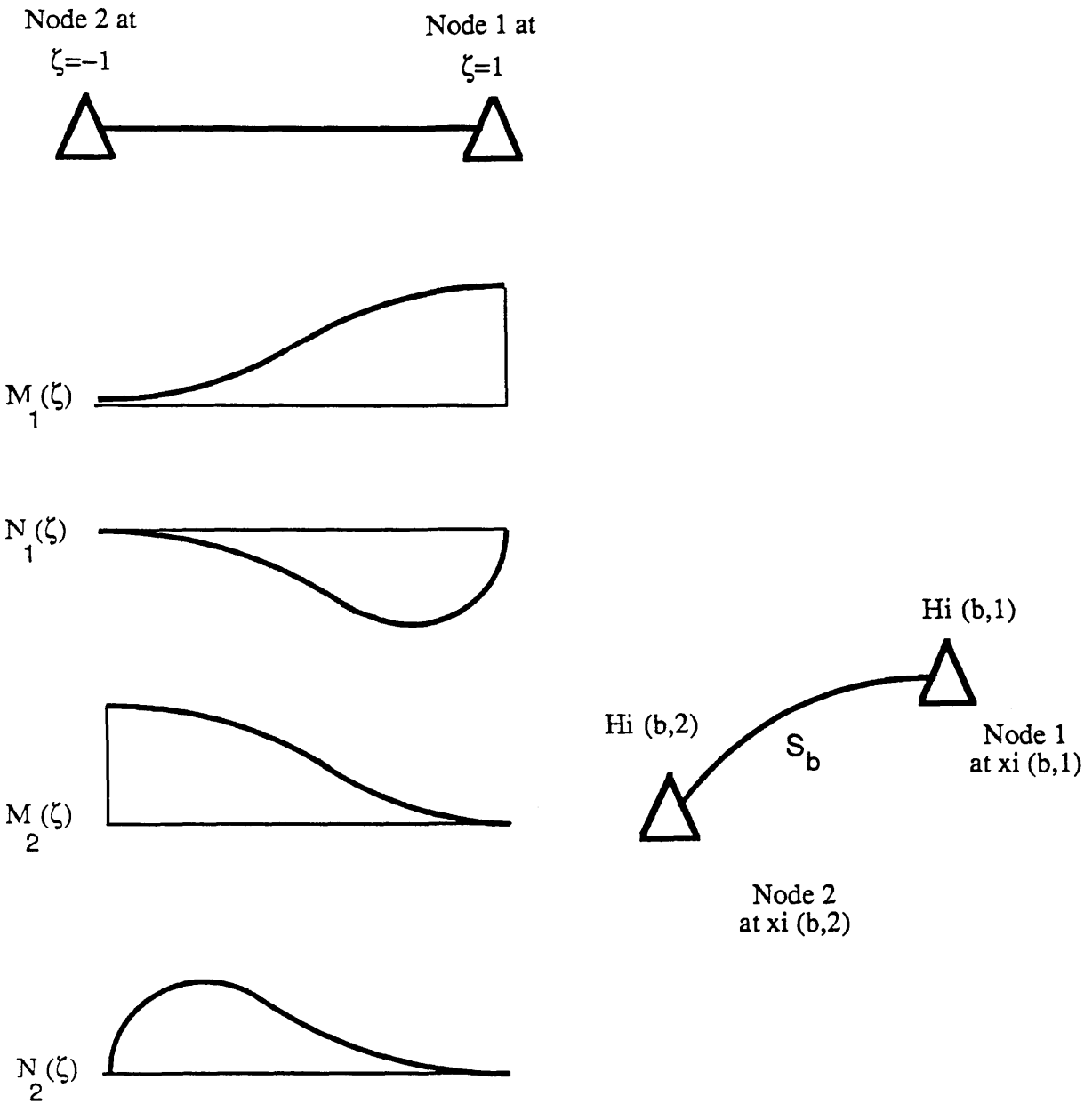
$$\text{Subject to} \quad : \quad g_k(X_j) \leq G_k \qquad k=1,....,m \tag{5.33}$$

in which n is the number of design variables, X_j is the design variables vector (co-

ordinates of some nodes on the moving part of the boundary which define the shape). m is the number of behaviour constraints, include stress constraints on whole the boundary or some part of it. Limitations are also placed upon the range of the design variables and are referred to as side constraints. Weight (W) is a nonlinear function and is calculated numerically by the BIE method during the process and is discussed in Sec.(3.4). Since both weight and stresses are nonlinear functions of design variables, then feasible direction method, in comparison with the Penalty Function Method and Sequential Linear Programming, seems to be more reliable because it deals directly with the nonlinearity of the problem. The first aim was to use the program of Sec.(5.9) which uses Penalty Function Method, but the results were not very accurate. Here the general WEIGHTOPT analysis program is written to handle the problem in Eqs (5.32), (5.33). The program couples the feasible direction method with a boundary element program. The optimizer with a small change can be used for any other optimization purposes.

This program like the program in the last section has three steps: Analysis of the stresses, sensitivity analysis and decision of redesign. The BIE method has shown itself to be a powerful method for analyzing specially for sensitivity analysis, which is discussed in chapter 4.

It has the same automatic mesh generator during the process and uses Hermitian cubic splines for its boundary shape representation with the same design variables. Usually after each iteration in the optimization process the stresses are active (meaning that they are near the allowable range), then for the one-dimensional search, the constrained one-dimensional search is followed which has been discussed in sec.(2.6.5). During the one-dimensional search, if a side constraint becomes critical, its value is fixed at its bound. The flow chart of this program can be seen in Fig.(5.7).



(a) Shape functions

(b) Geometry

Fig. 5.1 Hermitian cubic element S_b

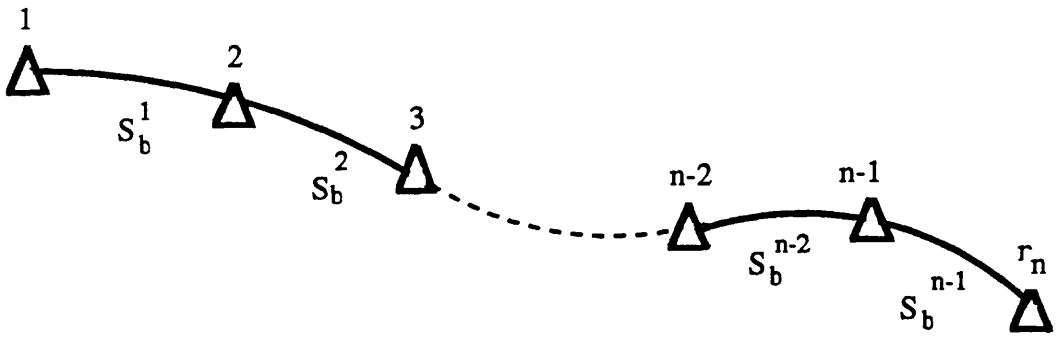


Fig. 5.2 Hermitian cubic spline through n points

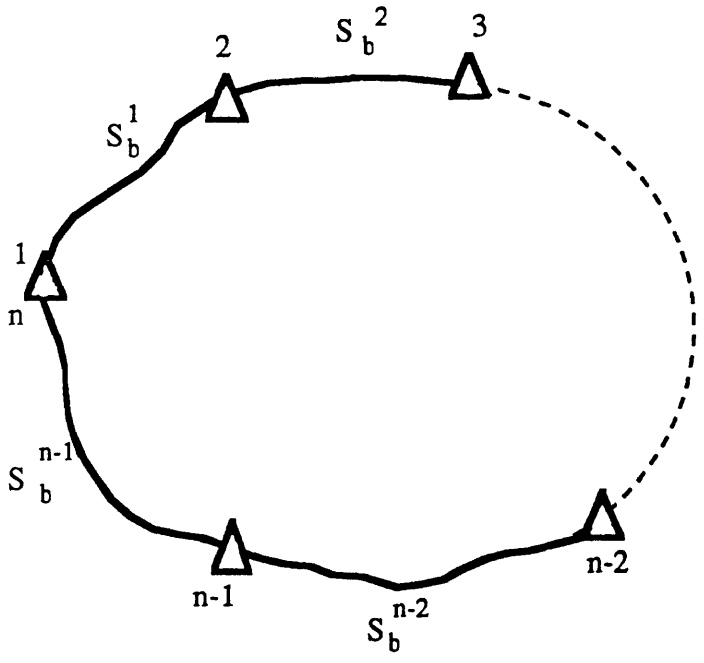


Fig. 5.3 Hermitian cubic spline through n points on a closed curve

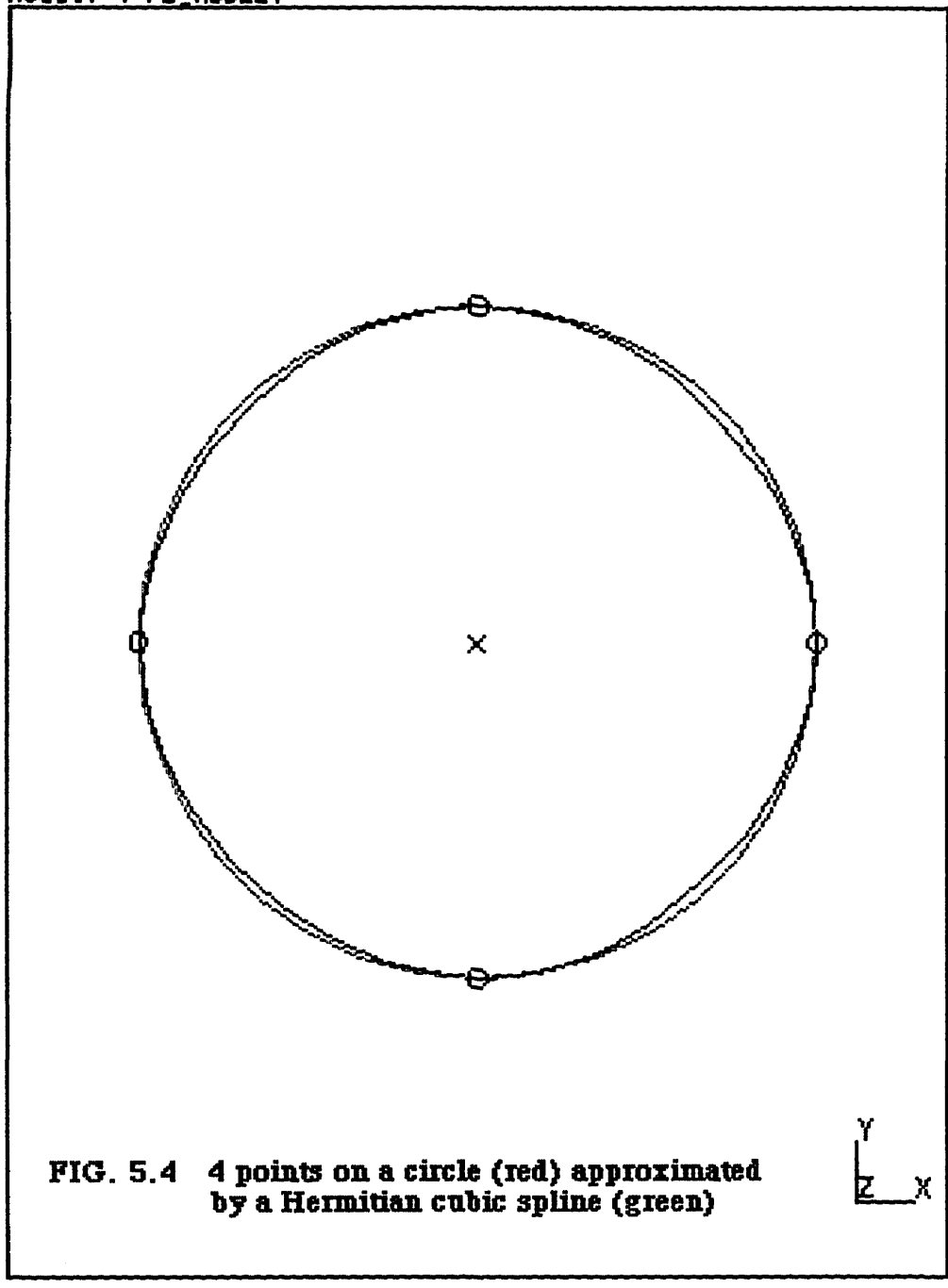


FIG. 5.4 4 points on a circle (red) approximated by a Hermitian cubic spline (green)

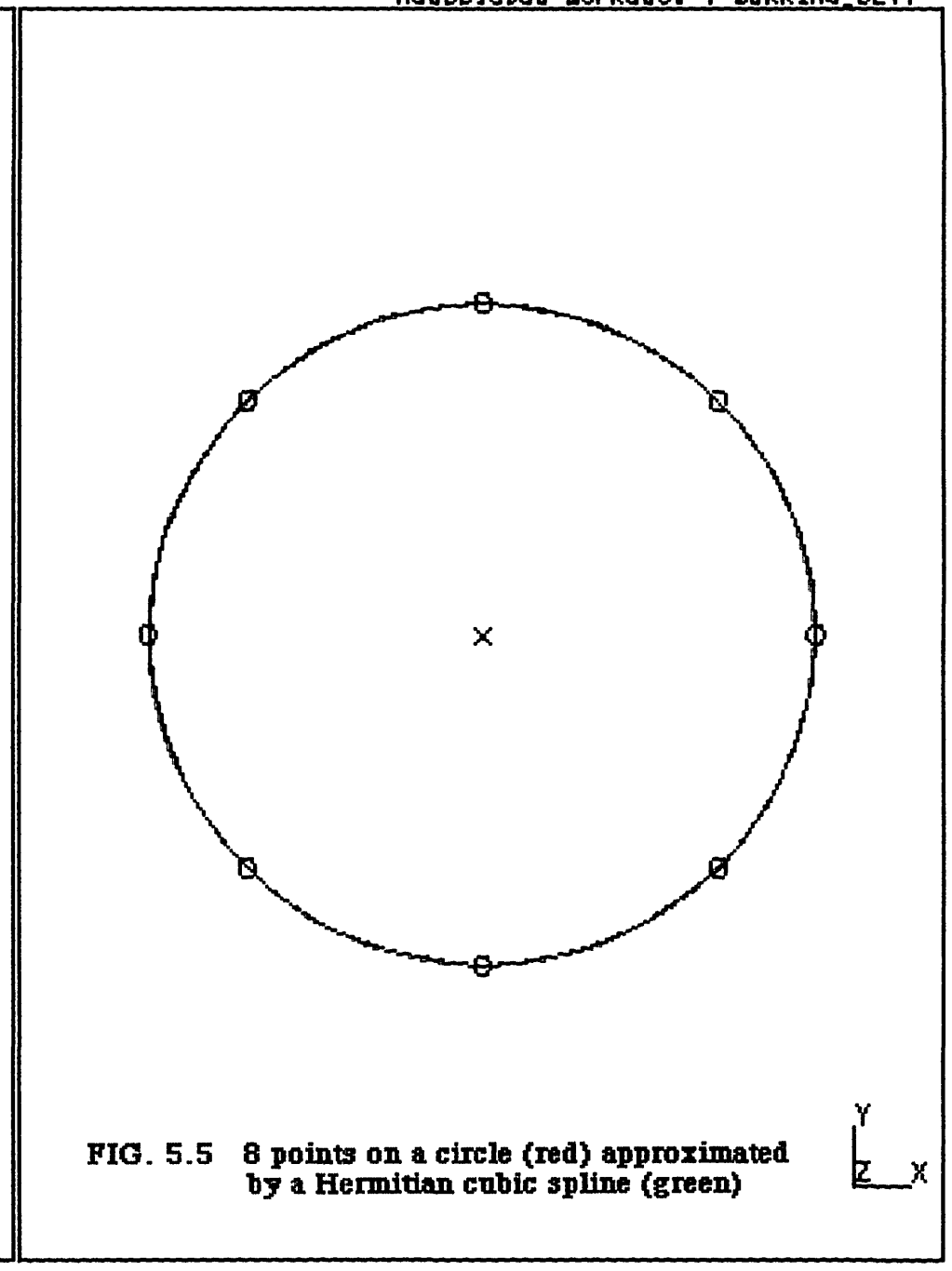


FIG. 5.5 8 points on a circle (red) approximated by a Hermitian cubic spline (green)

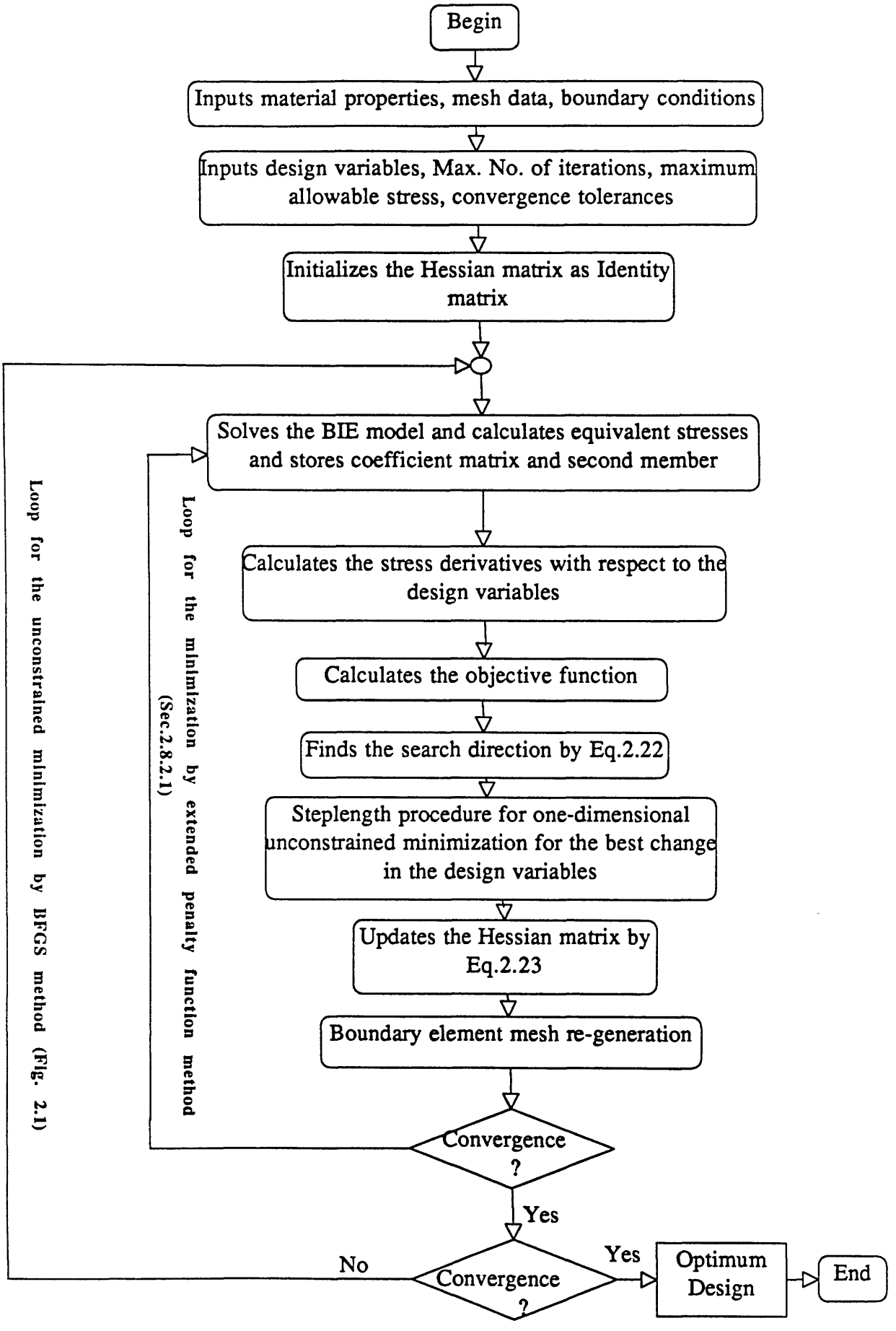
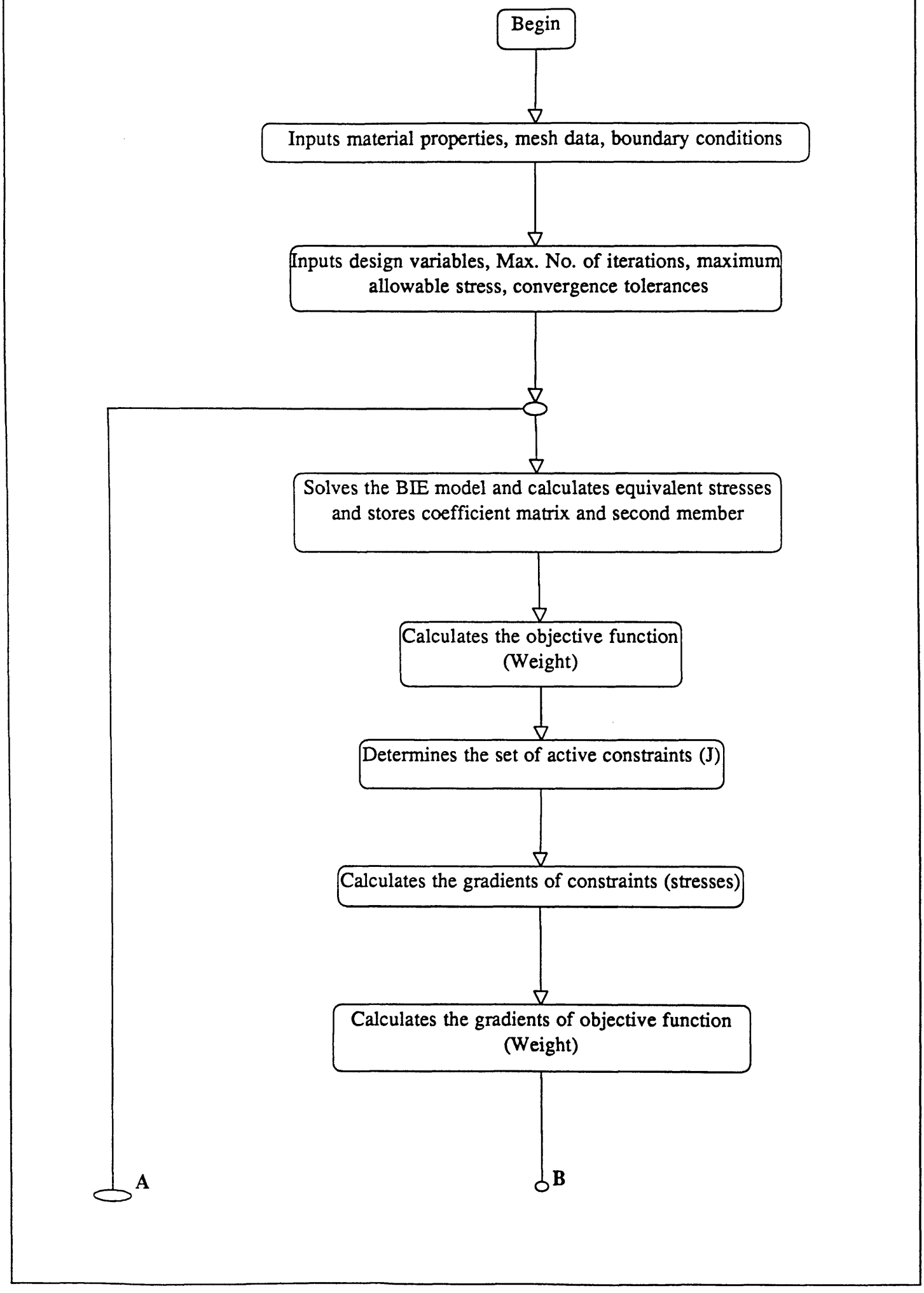


Fig. 5.6 Flow diagram for the peak stress minimization


Begin
Inputs material properties, mesh data, boundary conditions
Inputs design variables, Max. No. of iterations, maximum allowable stress, convergence tolerances
Solves the BIE model and calculates equivalent stresses and stores coefficient matrix and second member
Calculates the objective function (Weight)
Determines the set of active constraints (J)
Calculates the gradients of constraints (stresses)
Calculates the gradients of objective function (Weight)
A
B

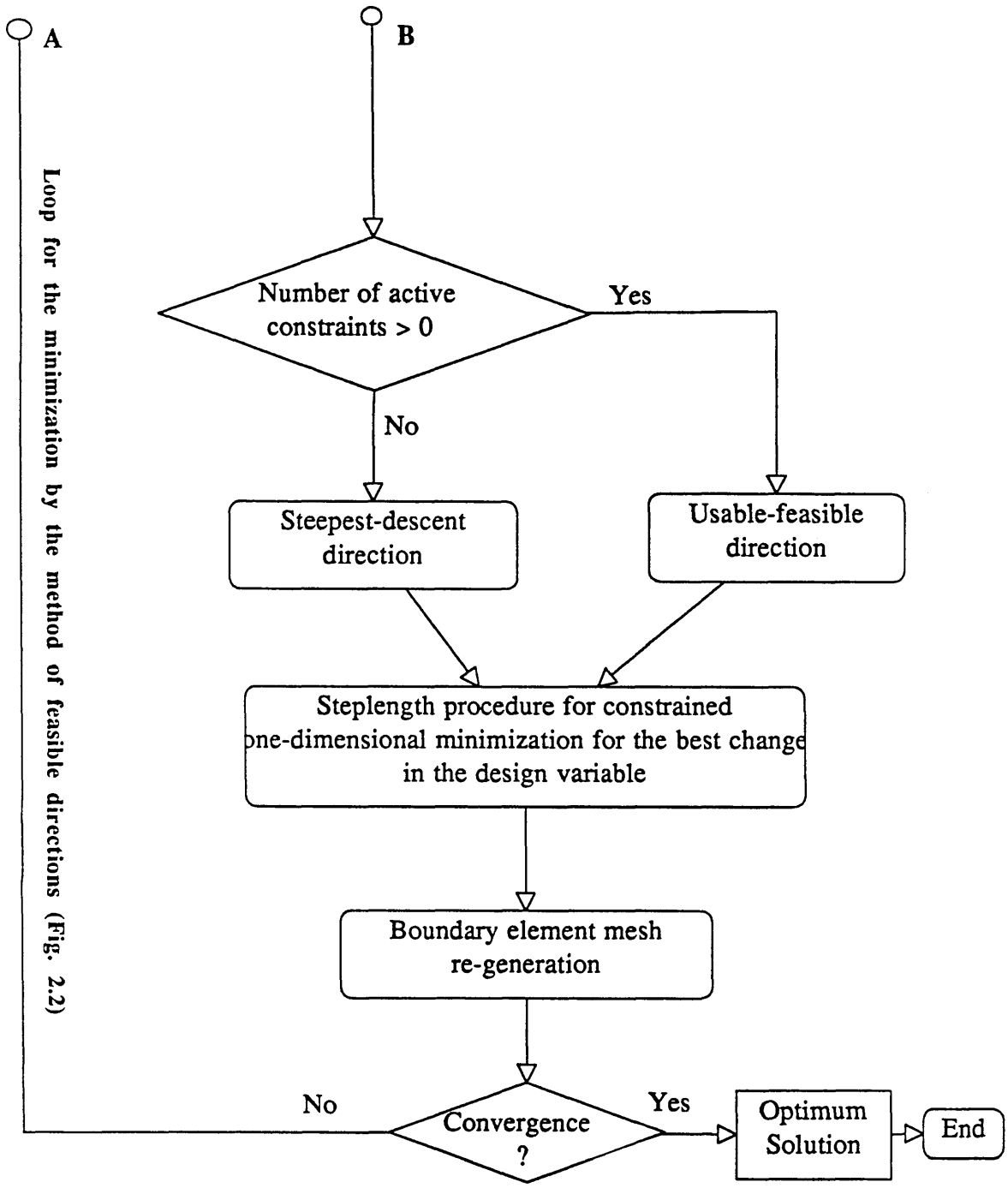


Fig. 5.7 Flow diagram for the weight minimization

CHAPTER 6

NUMERICAL RESULTS

6.1 - INTRODUCTION

The accuracy of the present solution algorithms may be determined by applying them to some standard test cases for which exact solutions exist, or to examples which have been solved by finite element method for comparison purposes.

Two different programs have been utilized for the shape optimization analysis. The first program is for the smoothing stress peaks under geometric constraints. Four examples are presented in this category as follows : optimization of the shape of a hole in an infinite plate under biaxial stresses (for which there is an analytical solution); optimal profile design for a connecting rod; optimizing the internal boundary of a tank with internal pressure; and optimum shape of a cantilever beam.

The second program is for weight minimization with constraints upon stresses, while keeping them under some allowable magnitude and also some geometric constraints. Three examples discussed in this part are as follows : a plate with a hole under biaxial stresses; a cantilever beam; and a torque arm.

The evaluation of integrals is performed using six Gauss points for integrals over boundary elements.

In the methods of *this* thesis, Hermitian cubic splines and in some cases cubic splines are chosen for the boundary description. As mentioned in Sec.(5.5), for definition of these sorts of splines through n defined points, besides the coordinates of the points, it is necessary to define values of the gradients of the path with respect to the intrinsic coordinate ζ for each end of the curve (for the cubic spline only the slopes at

each end). By having the coordinates of the n points , the coefficients of 2(n-2) linear equations of the gradients of the path with respect to ζ at each point are known, so the only data required by the user is to define coefficients of four extra equations related to the gradients of the first and end points of the spline. In the following examples, these assumptions are briefly described in each case.

6.2 - SMOOTHING STRESS PEAKS

6.2.1 - Optimum Shape Of A Hole In An Infinite Plate

This example (Fig. 6.1a) comes from the paper of Kristensen and Madsen [1976] where they used the finite element method. The optimum shape of the hole is required. The plate is under a biaxial stress field with the ratio σ_2/σ_1 =1/2. The thickness of the plate is assumed to be small enough so that a state of plane stress may be assumed. The initial shape for the hole is considered as a circle. The problem is symmetric about both coordinate axes, so only the quadrant region ABCDE needs to be modelled. The faces AB and DE are the two faces of symmetry. The boundary conditions are v=0 along AB and u=0 along ED, where u and v are the displacement components. The model contains 36 quadratic boundary elements. Each side ED and AB has 10 elements with element concentration factor 1.5, each side CD and BC has 2 elements and along the curved edge, AE, 12 elements with no element concentration are employed, Fig.(6.2).

During the optimization process, the point E is fixed while the point A is allowed to change its position horizontally. The tangents of the edge curve should remain perpendicular to the axes at A and E. The radii of points P_1-P_4 at the fixed angles shown in Fig.(6.1b) are considered as the design variables. Since in Chapter 4, the derivatives of the objective and constraint functions are considered with respect to the cartesian coordinates of the design variable points, so in these cases the programs are capable of converting the derivatives with respect to the radii of these points by using

Eq.(5.30).

The Hermitian cubic spline, the boundary description for the curved edge, should pass through the points P_1-P_5. The tangents of the edge AE is perpendicular to the axes at A and E then

$$l_{P1}=0 \quad \text{and} \quad m_{P5}=0 \tag{6.1}$$

where l_{P1} and m_{P5} are the gradients $(dx_i/d\zeta)$ of the path at points P_1 and P_5 with respect to the intrinsic coordinate ζ, defined in Sec.(5.5), and L_{P5} and m_{P1} are unknown. The model is symmetric about both coordinate axes. If points P'_2 and P'_4 (Fig.6.1b) are the symmetry of points P_2 and P_4 relative to the edges AB and DE, respectively, then the aforementioned spline should pass through points P'_2 and P'_4 as well. Due to the symmetry $x_{P2}=x_{P'2}$, $y_{P2}=-y_{P'2}$ and $x_{P4}=-x_{P'4}$, $y_{P4}=y_{P'4}$ and

$$l_{P4} = l_{P'4} \quad \text{and} \quad m_{P2} = m_{P'2} \tag{6.2}$$

By adding Eqs.(6.1) and (6.2) to 2(N-1) linear equations of (5.19) and (5.20), the gradients $(dx_i/d\zeta)$ of the path at each design variable point will be calculated during the process and the unique spline will be defined at each iteration.

Fig.(6.3) shows the shape of the quadrant of the hole in some successive iterations. Fig.(6.4) shows the equivalent stresses around the curve edge at the initial and optimum shape and also for the analytical solution. The process has been completed in 25 iterations and the defined mean stress is the average of the stresses around the hole at the initial step.

There is an analytic solution for this problem [Kristensen & Madsen, 1976] which shows the optimum shape of the hole should be an ellipse with the ratio of $b/a=\sigma_2/\sigma_1=1/2$ where the equivalent stresses are uniform around the hole. In Fig.(6.5), the optimum shape has been compared with the analytical solution which shows that the optimum shape has a good agreement with the exact solution. Table (6.1) shows the analytical and numerical values of σ_{max}/σ_1 for the initial and optimum design.

Circular Hole	Anal.	2.5
	Num.	2.509
Optimum Shape	Anal.	1.59
	Num.	1.5

Table 6.1 Analytical and numerical values of σ_{max}/σ_1

6.2.2 - Optimum Profile Design For A Connecting Rod

This example was presented by Francavilla et al [1975]. the purpose is the design of a connecting rod profile in the transition region of its connection with the eye, (Fig.6.6). The loading assumed is a uniform internal pressure of 1 unit over the semi-circular region and a uniform tension at the end of the rod. Due to the symmetry only the upper half of the model is considered.

In this example, instead of Hermitian cubic spline, the profile assumed is a cubic spline (y=f(x)), because there is no possibility for dy/dx along the profile to be infinity. The three design variables chosen are the x-coordinate of point D and y-coordinates of points B and C. Point A has a fixed position. The cubic spline passes through these four points and has zero slopes at points A and D. The geometric constraint in this model is that none of these coordinates can be negative. The boundary element mesh, shown in Fig.(6.7), consists of 36 quadratic isoparametric elements.

The initial shape assumed was chosen to give a quite severe stress concentration in the region. The successive shapes taken by the optimization at different stages of convergence and the optimum shape are shown in Fig.(6.8). Fig.(6.9) shows the stress-distribution along the profile before and after optimization. Technically speaking, the flat region of uniform stress is very satisfactory. 7 iterations were necessary to

reach the solution.

6.2.3 - Tank With Internal Pressure

This example is concerned with the optimal shape of a thick-walled tank under an internal pressure of 1 unit, Fig.(6.10a). The model is symmetric about both coordinate axes, so only the quadrant region ABCDEF is considered. The boundary conditions are v=0 along AB and u=0 along DE.

The purpose is to find the optimum shape for the inner boundary, EFA. Design variables are the radii of the points P_1-P_7 while the edge FA is allowed to move horizontally with the same amount of movement of point F in the x-direction and the boundary BCD is fixed. The Hermitian spline should pass through points P_1-P_7. The tangents of the edge EF should remain perpendicular to the axes at F and E then

$$l_{P1} = 0 \qquad \text{and} \qquad m_{P7} = 0 \tag{6.3}$$

The curved edge EF is symmetric about the y-coordinate, so if P'_6 is the symmetry of P_6 relative to the y-axis, the spline passes through this point as well, then

$$l_{P6} = l_{P'6} \tag{6.4}$$

while $x_{P6} = -x_{P'6}$ and $y_{P6} = y_{P'6}$. Consider the point P'_1 is on the edge FA and very close to point P_1 and it may be assumed the spline passes through this point then

$$m_{P1} = m_{P'1} \tag{6.5}$$

while the coordinates of P'_1 are known.

The geometric constraints in this example are, the x-coordinate of A and y-coordinate of E can not be negative. The boundary element mesh, shown in Fig.(6.11), consists of 42 quadratic elements. The initial shape assumed for the inner boundary is a semi-circle. The optimum and initial shapes are shown in Fig.(6.12). Fig.(6.13) shows the stress-distribution before and after optimization which, with the stresses being uniform around the edge EF, is very satisfactory. 13 iterations were necessary to reach the optimum solution.

6.2.4 - Cantilever beam

The cantilever beam is defined in Fig.(6.14). The beam is under a uniform shear load of 1 unit at one end and clamped at the other end. With symmetry, only the upper half of the beam is considered (ABCD). The purpose is to find the optimum shape of the edge CD. The point C is allowed to move vertically while the point D is fixed. 6 design variables chosen are the y-coordinates of points P_1-P_6 which are equally spaced along the edge DC. Since the slopes at the beginning and end of this edge are not defined, the first and last elements of the spline are assumed to be quadratic. The boundary element mesh shown in Fig.(6.15) consists of 40 quadratic elements. The optimization process is completed in 21 iterations. The same problem was solved with the slope at the point D assumed to be zero. Fig.(6.16)shows the optimum shape when the slope at point D is zero (green curve) and when the slope at point D is not zero (pink curve) which are approximately the same. The initial shape chosen is a straight line. Distribution of equivalent stresses on the upper edge of the beam before and after optimization for both cases are shown in Fig.(6.17). The initial maximum stress was 35.2 and the final one is about 20.0, which shows a great reduction.

6.3 - MINIMUM WEIGHT DESIGN

6.3.1 - Cantilever Beam Under Lateral Load

Selection of the best shape of a beam with minimum weight so that no yielding occurs has long attracted the attention of engineers. A cantilever beam is defined in Fig.(6.18). There is no symmetry and the whole model is considered. This beam is under a uniform pressure of magnitude 0.1. The purpose is to find the optimum shape of edge AB. The point B is allowed to move vertically and the points A, D and C are fixed. The slopes at the beginning and the end of the edge AB are not zero. 6 design variables, y-coordinates of points P_1-P_6, which have equal distance from each other, are chosen. The Hermitian cubic spline passes through points P_1-P_7 and the first and the end elements of the spline, since the slopes at both ends are not defined, are assumed as quadratic. The boundary element mesh shown in Fig.(6.19) consists of 54 quadratic elements. The initial shape chosen is a straight line, which together with the optimum shape are shown in Fig.(6.20). Fig.(6.21) shows the weight history. This run terminated with a feasible design at 9 steps. There is a reduction of 43.4% in the mass. The maximum allowable stress is σ_y=9.1. The maximum equivalent stress at initial step is 9.067 and for the optimum shape is 9.1064.

6.3.2 - Plate Under Biaxial Stresses

This example was presented by [Esping, 1985]. The model is defined in Fig. (6.22a). It is a plate with uniform thickness with a hole in the centre, under biaxial stresses with ratio σ_2/σ_1.=1/2. The purpose is to remove material around the hole as much as possible but with no yielding occurring. Due to the symmetry, only a quadrant of the plate is considered. Point A is allowed to move horizontally, while the point E is allowed to move vertically. The tangents to the edge EA should remain perpendicular at

points A and E. The 7 design variables chosen are the radii of points P_1-P_7 at constant angles shown in Fig. (6.22b). The Hermitian cubic spline passes through points P_1-P_7 and the definition of the gradients of the path ($dx_i/d\zeta$) at the first and end of the curve is the same as for the infinite plate with hole in Sec.(6.2.1). The boundary element mesh shown in Fig.(6.23) consists of 62 quadratic elements. The initial shape chosen is a circle, which together with the optimum shape are shown in Fig.(6.24). The weight history is shown in Fig.(6.25) which displays 17% reduction in the mass in 35 steps. The maximum allowable stress is σ_y=3.6. The maximum equivalent stress before optimization is 3.5378 and after optimization is 3.5998.

6.3.3 - Torque Arm

An automotive design of a torque arm studied by [Botkin, 1981] is analyzed, with the geometry shown in Fig.(6.26) A single non-symmetric in-plane static load and plane stress are considered. Zero displacement boundary conditions are applied around the larger hole on the right, in order to simulate attachment to a solid rear axle. The boundary element mesh, geometry, loading conditions, and dimensions are shown in Figs.(6.26) and (6.27). Thickness is kept constant at 0.3 cm. The shape of upper and lower portions of the boundary between points A and B is to be determined through the optimization process, with zero slopes at the ends. The other boundary segments are kept fixed.

Design variables for a cubic spline function representation of the boundary are heights of the nodes N7, N15, N23, N31 and N39. A symmetry of geometry is imposed on the torque arm, so that the upper edge has the same movement as the lower edge. The boundary element model (Fig. 6.27) includes 70 quadratic elements.

With the initial shape, no constraint is violated. The weight initially is 1.04679 Kg and after optimization is reduced to 0.70604 Kg and still no constraint is active which shows 32.55% reduction in the weight. The final design shape is shown in Fig.(6.28) compared with the result in reference [Botkin, 1981], the reduction in material is very

close, but the optimum shapes are slightly different. The different design, may result from different number of design variables. 5 design variables are chosen to parametrize the lower boundary while symmetric geometry is imposed on the upper and lower boundaries. In the above reference only two design variables are used for each edge without symmetric geometry.

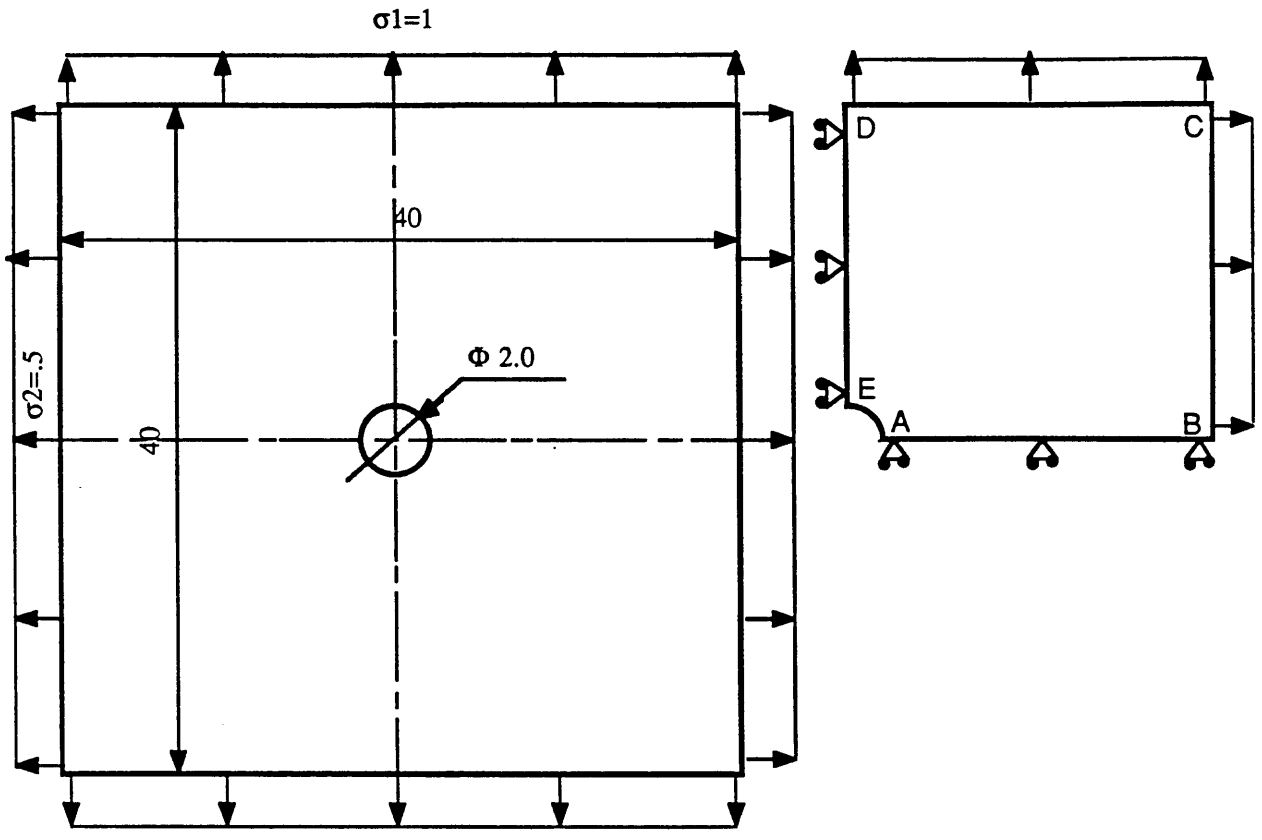


Fig. 6.1a A hole in an infinite plate under biaxial stresses

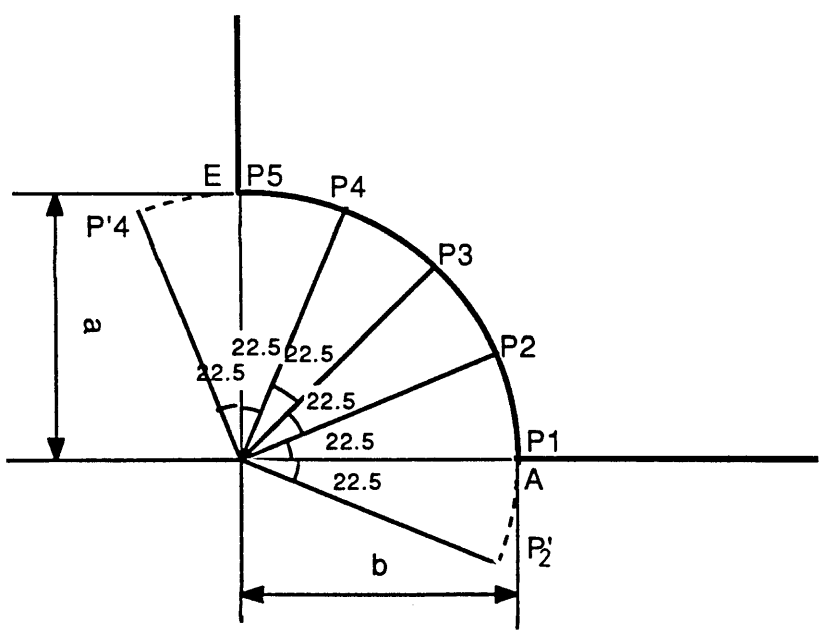


Fig. 6.1b Position of the design variable points on the edge EA

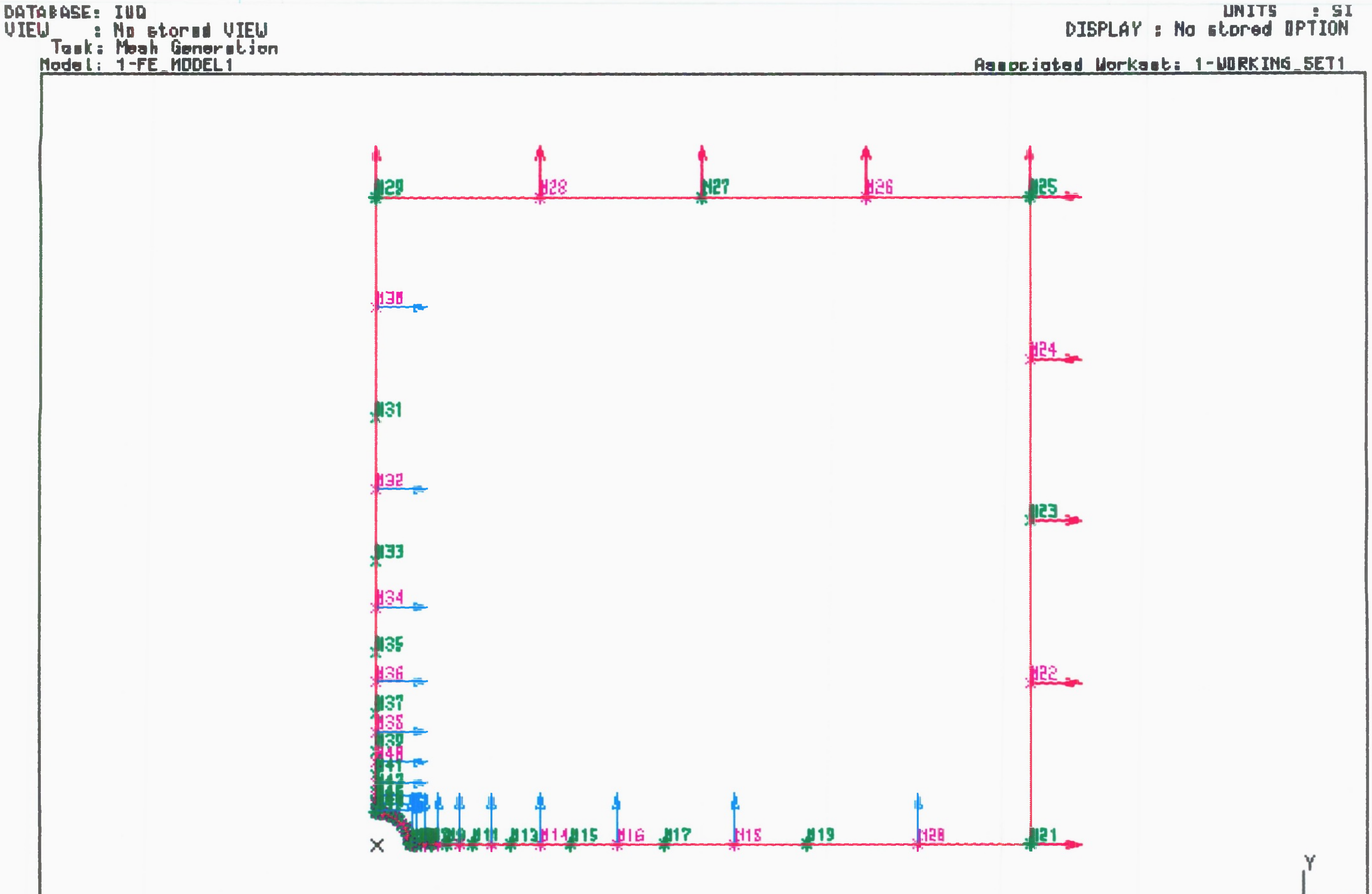


FIG. 6.2 Boundary element mesh of a quarter of the plate with a hole

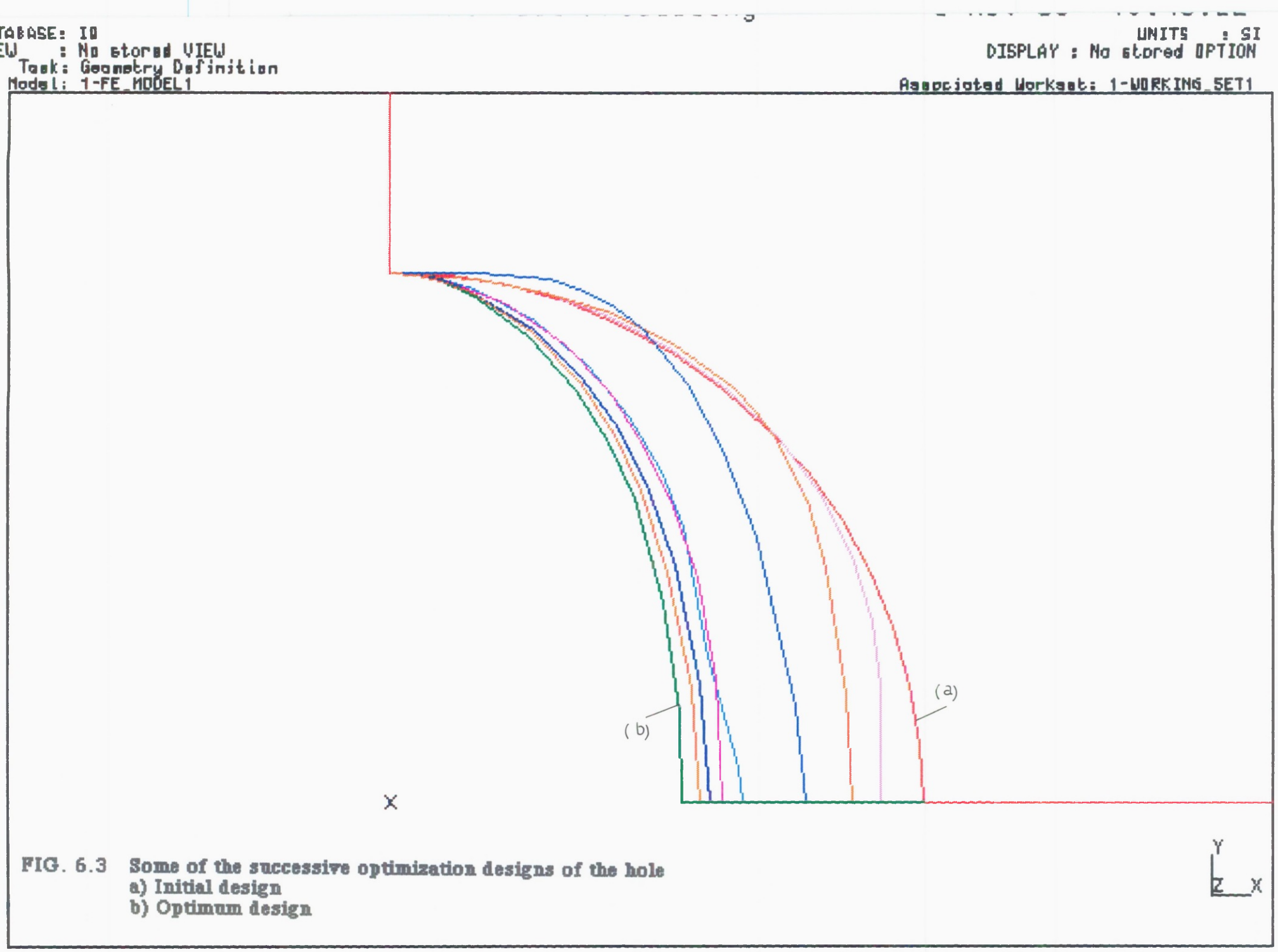


FIG. 6.3 **Some of the successive optimization designs of the hole**
a) Initial design
b) Optimum design

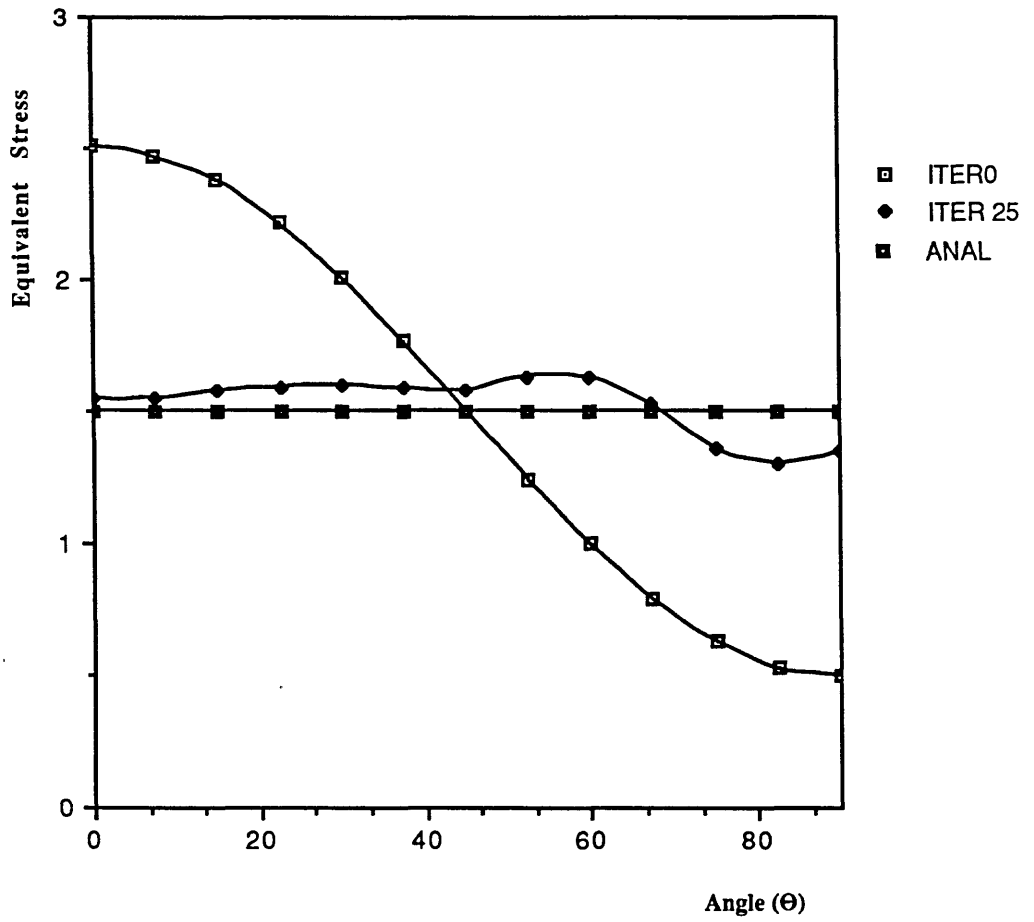


FIG. 6.4 Equivalent stresses around the hole before and after optimization and for the analytical solution

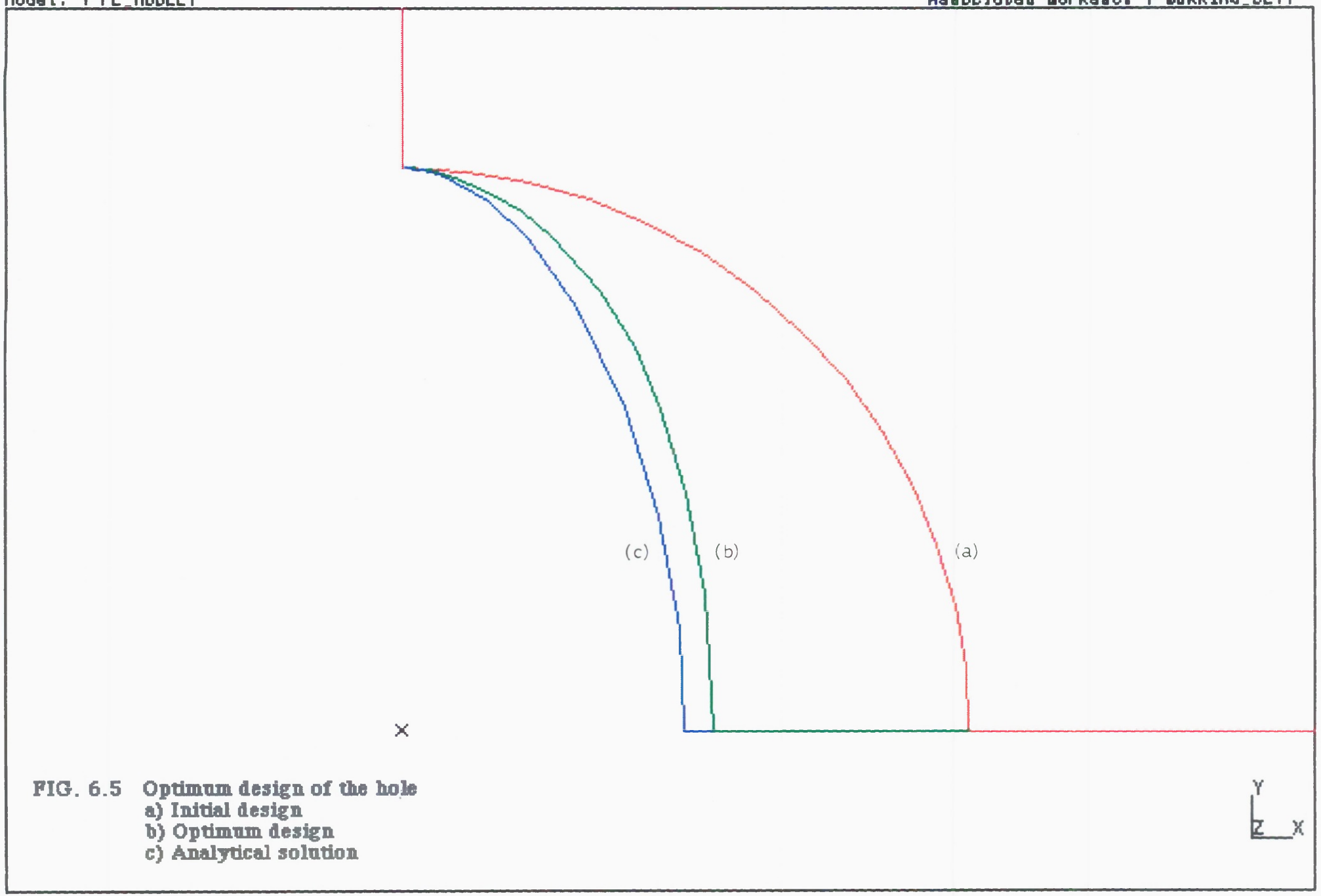


FIG. 6.5 Optimum design of the hole
a) Initial design
b) Optimum design
c) Analytical solution

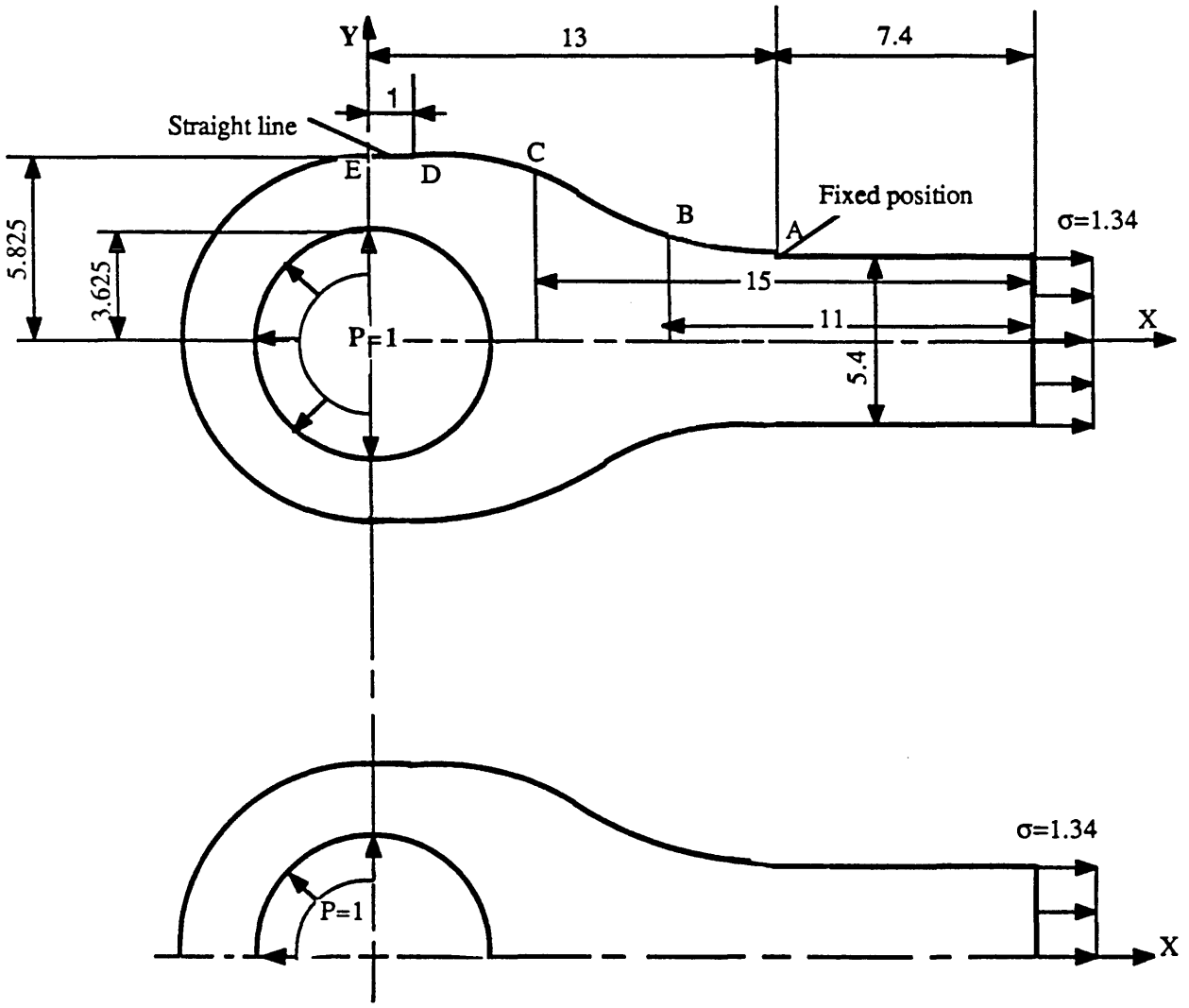


FIG. 6.6 Connecting rod

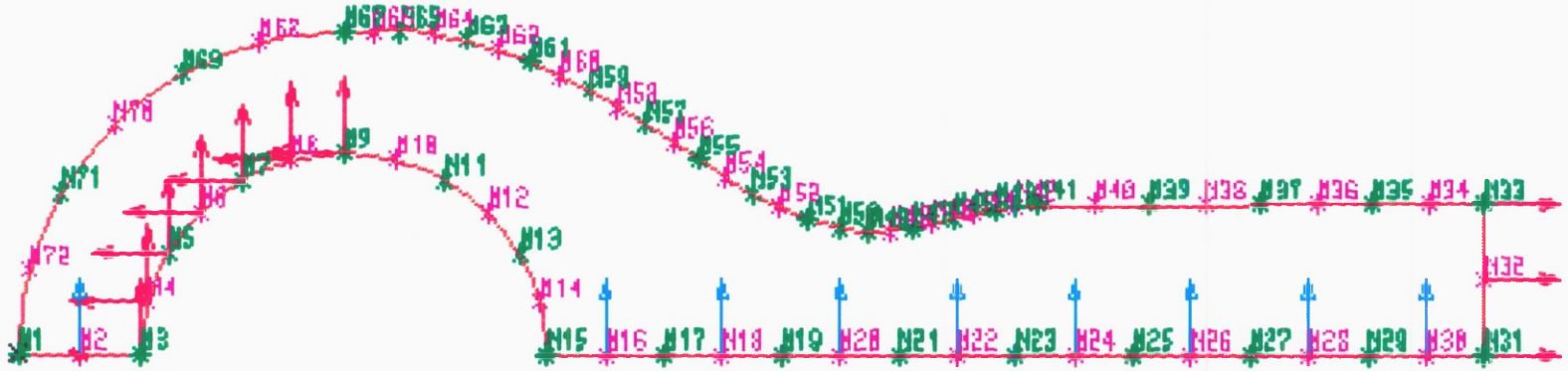


FIG. 6.7 **Boundary element mesh of half of the connecting rod**

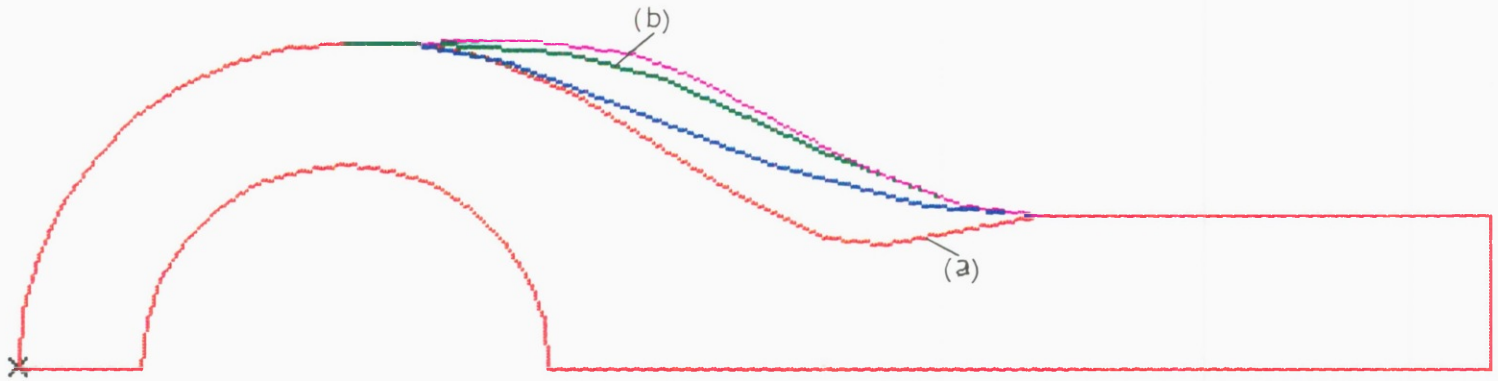


FIG. 6.8 Successive optimization designs of the profile
a) Initial design (red)
b) Optimum design (green)

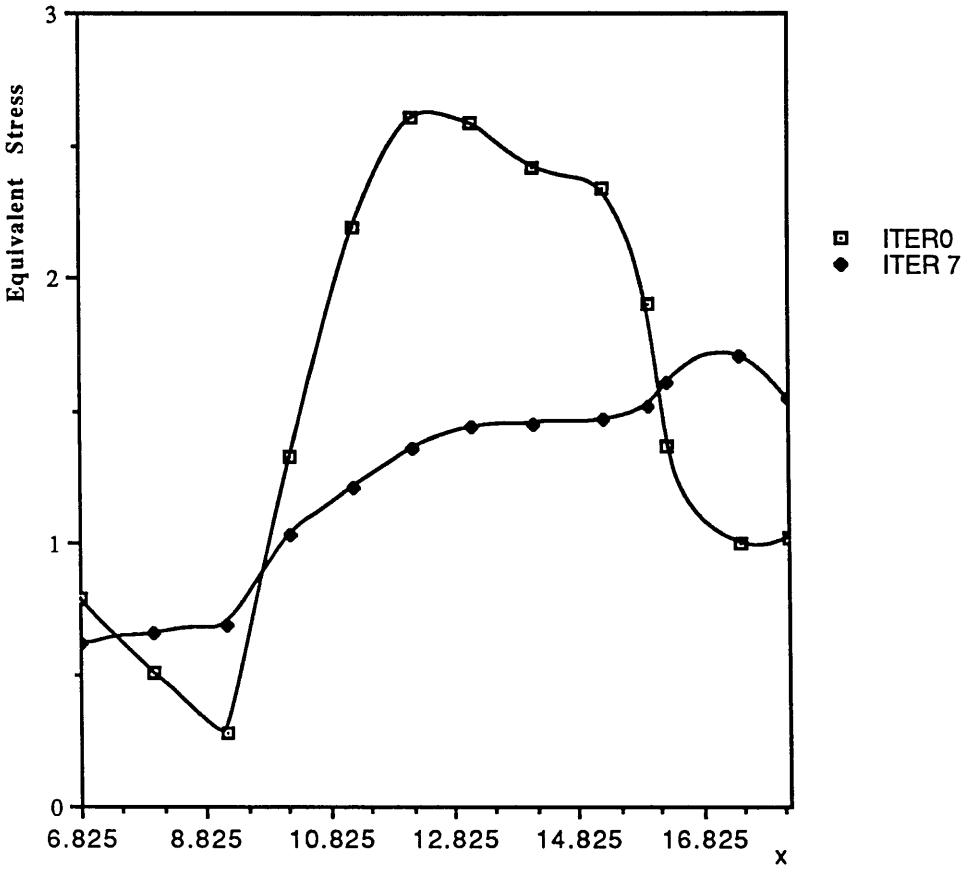


FIG. 6.9 Stress distribution along the profile before and after optimization

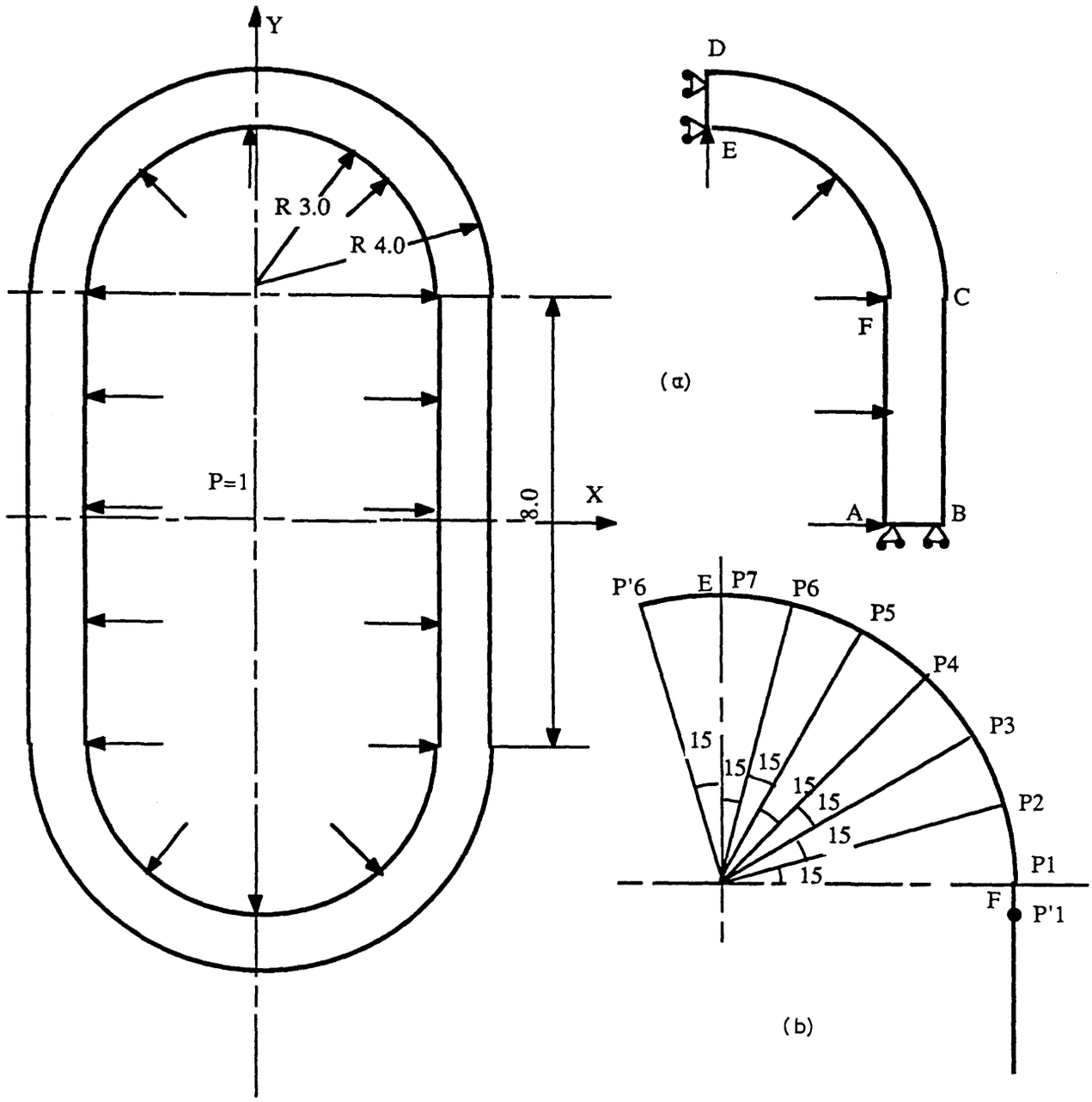


FIG. 6.10 Tank with internal pressure

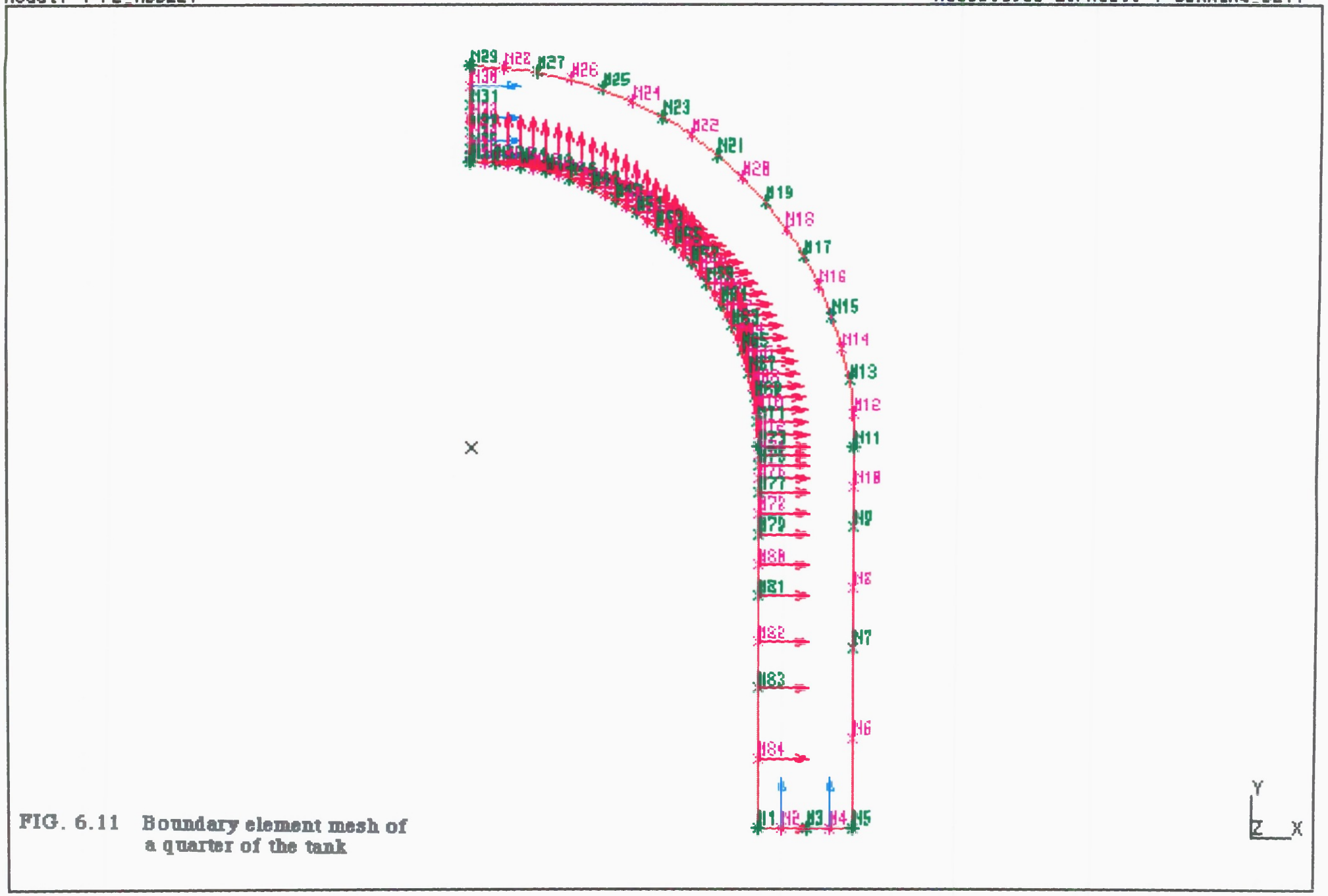


FIG. 6.11 Boundary element mesh of a quarter of the tank

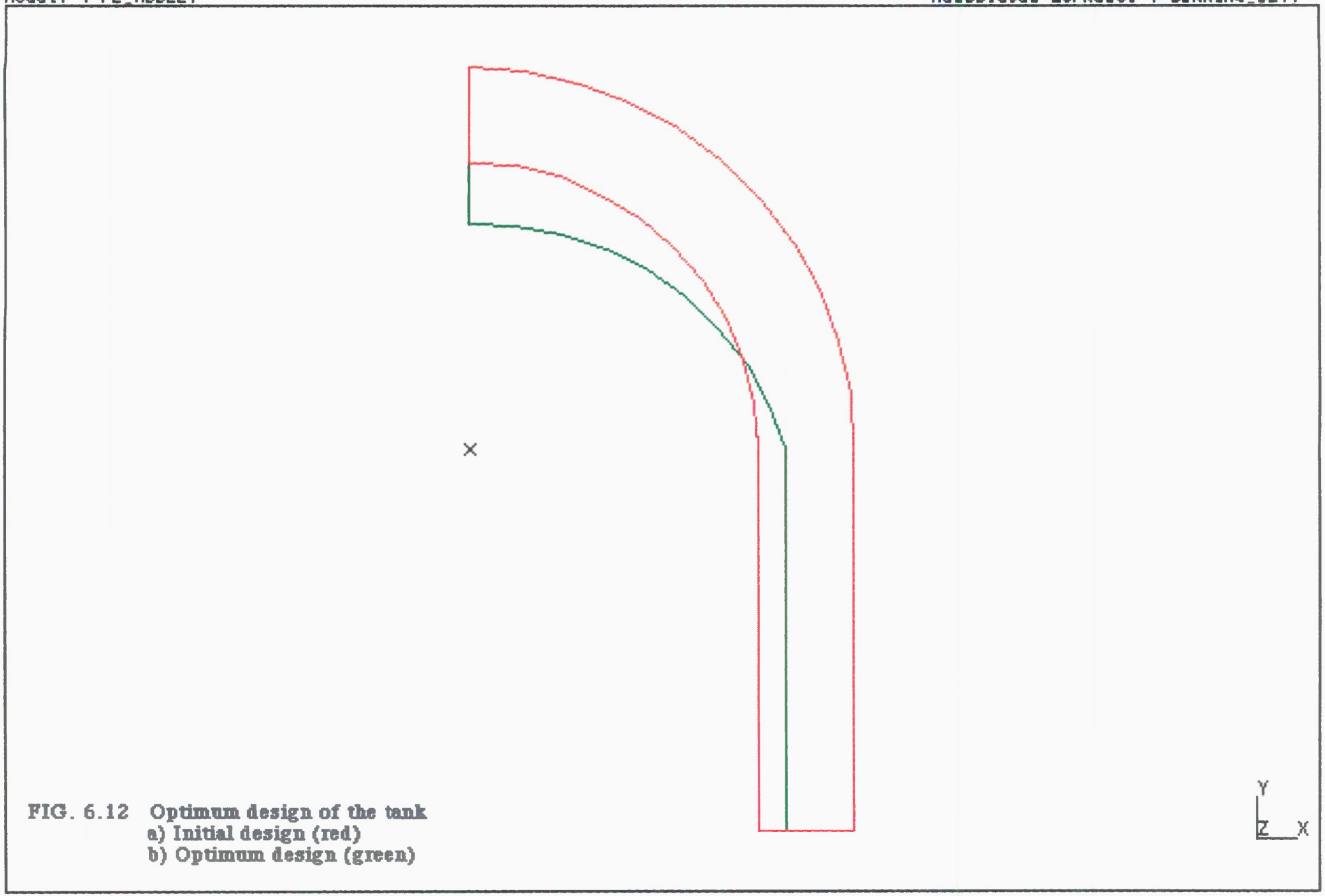


FIG. 6.12 Optimum design of the tank
a) Initial design (red)
b) Optimum design (green)

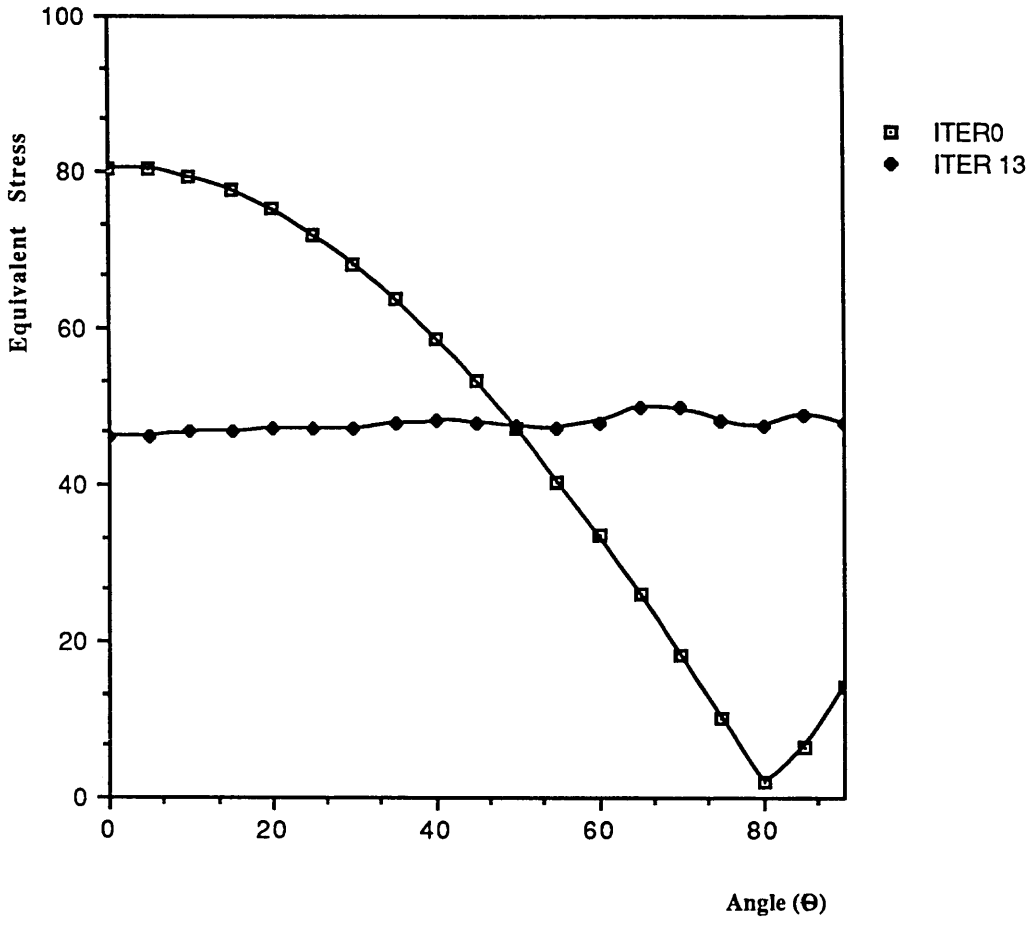


FIG. 6.13 Equivalent stresses around the edge EF before and after optimization

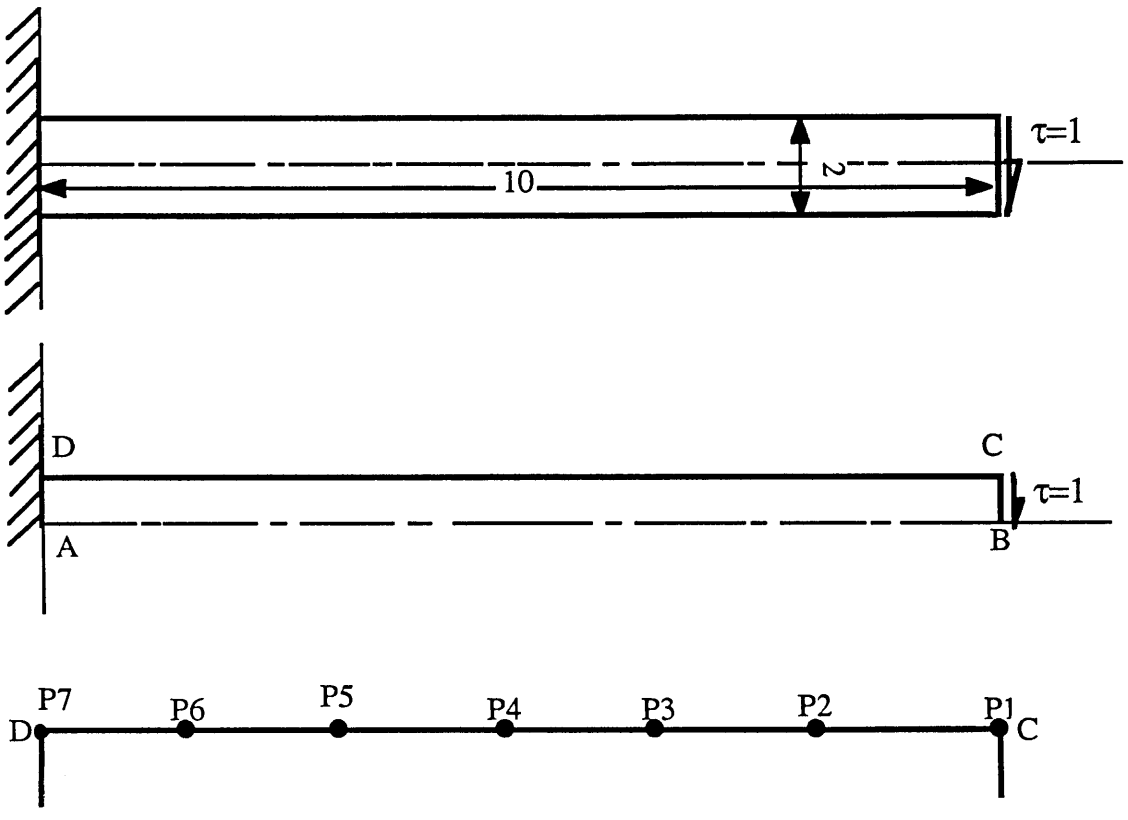


Fig. 6.14 Cantilever beam



FIG. 6.15 **Boundary element mesh of upper half of the cantilever beam**

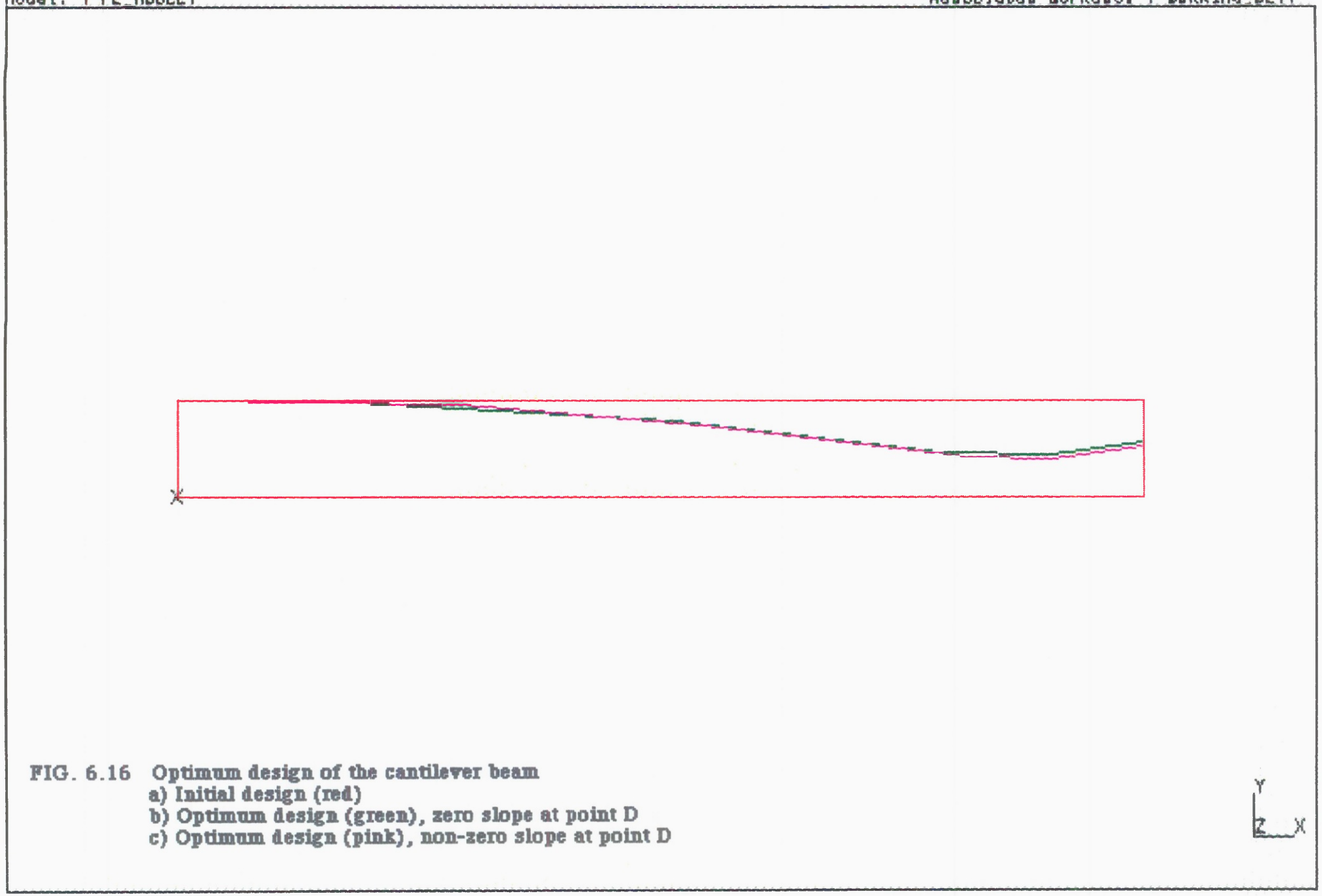


FIG. 6.16 Optimum design of the cantilever beam
a) Initial design (red)
b) Optimum design (green), zero slope at point D
c) Optimum design (pink), non-zero slope at point D

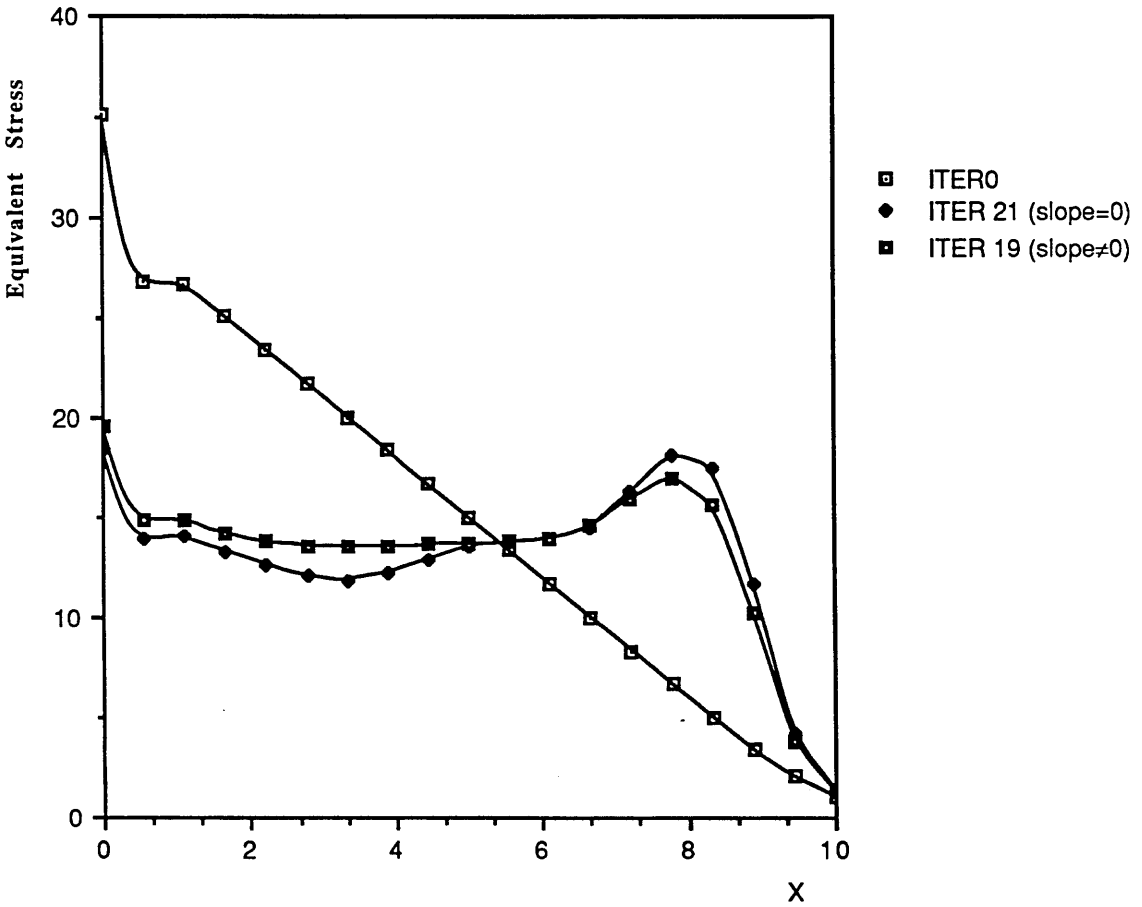


FIG. 6.17 Equivalent stresses along the upper edge of the beam before and after optimization (for zero and non-zero slopes at point D)

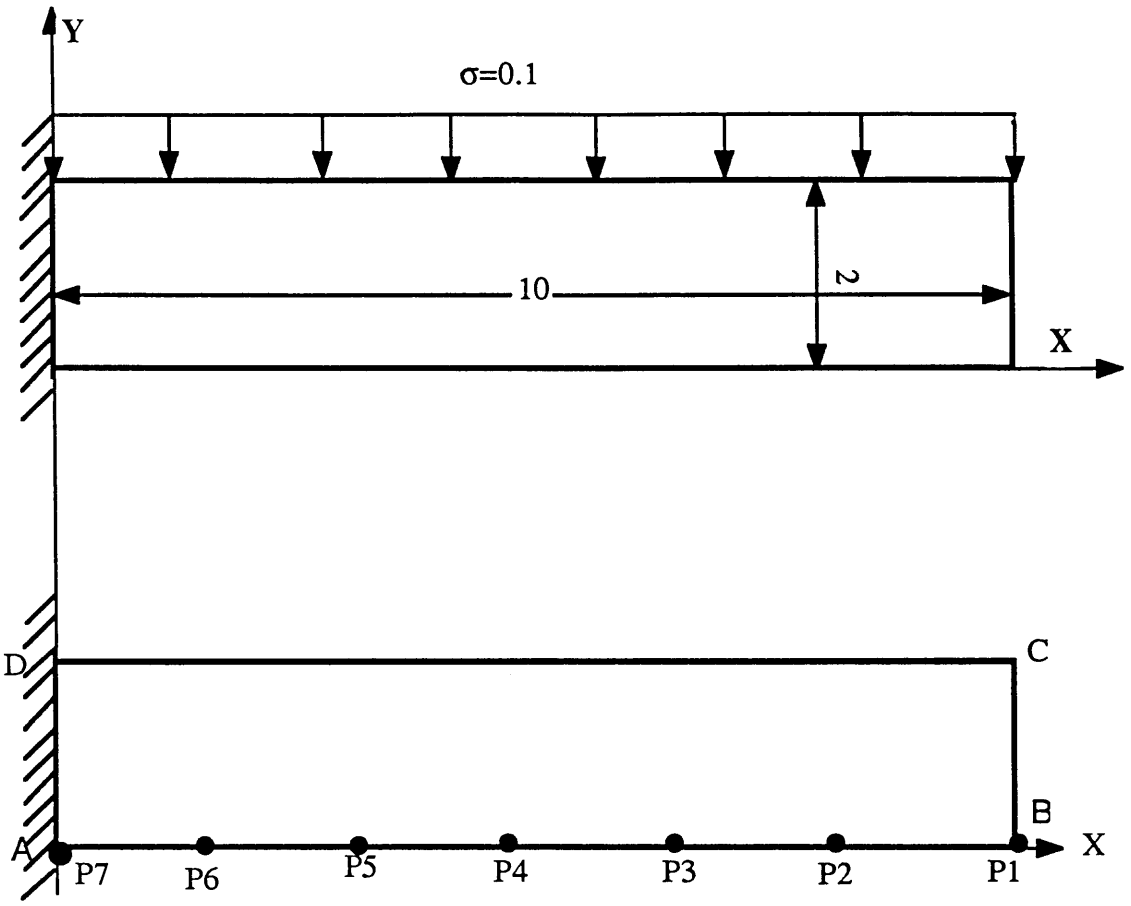


Fig. 6.18 Cantilever beam

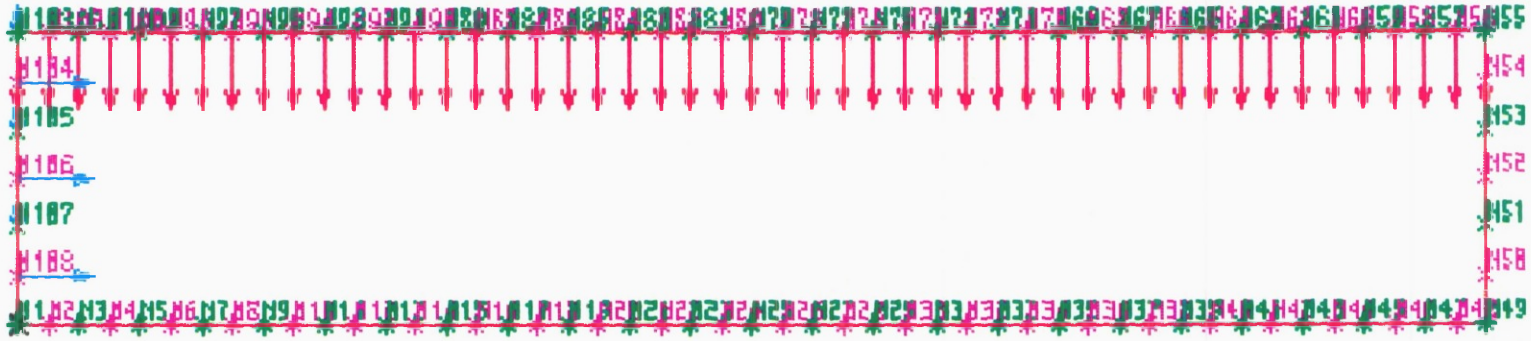


FIG. 6.19 Boundary element mesh of the cantilever beam

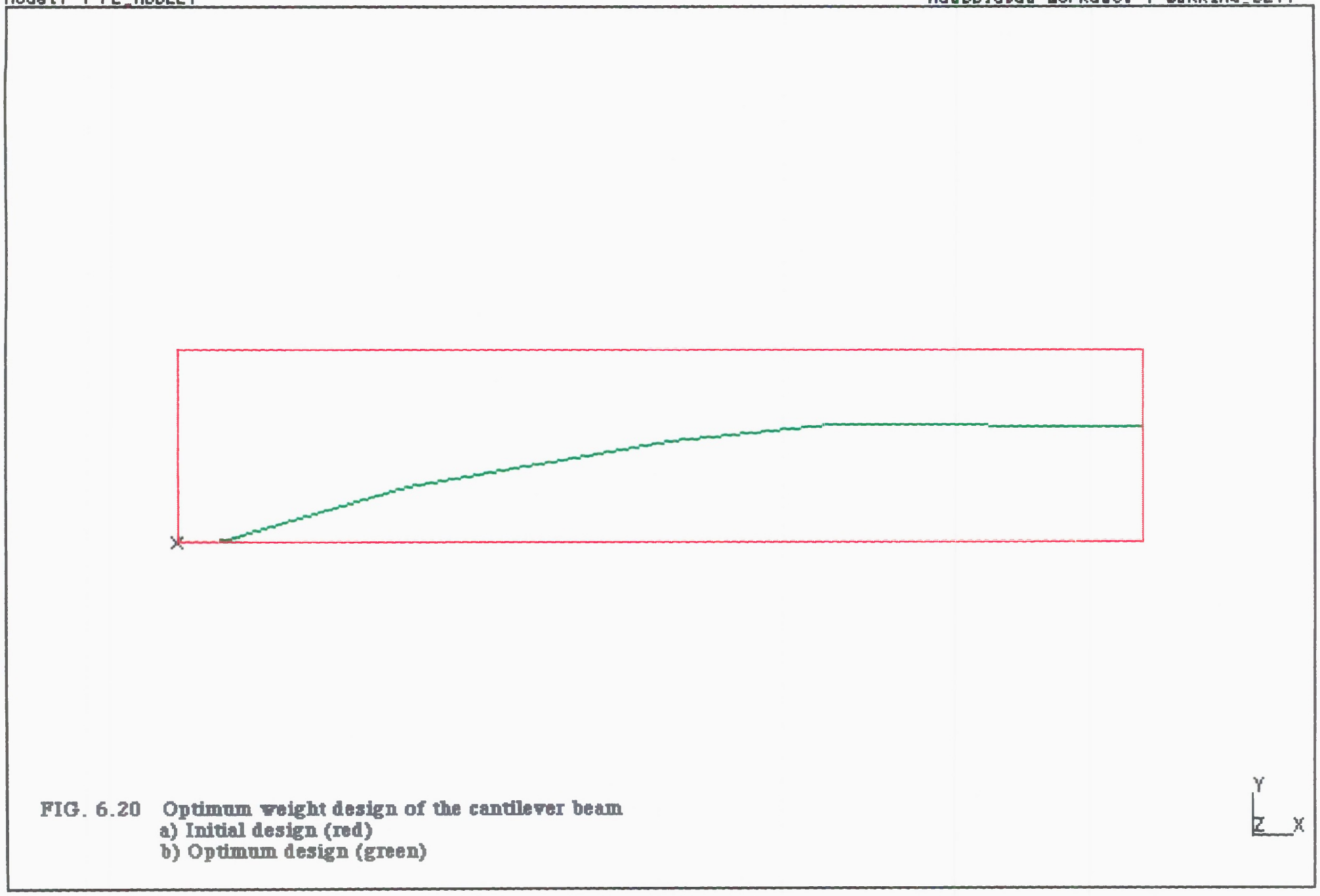


FIG. 6.20 Optimum weight design of the cantilever beam
a) Initial design (red)
b) Optimum design (green)

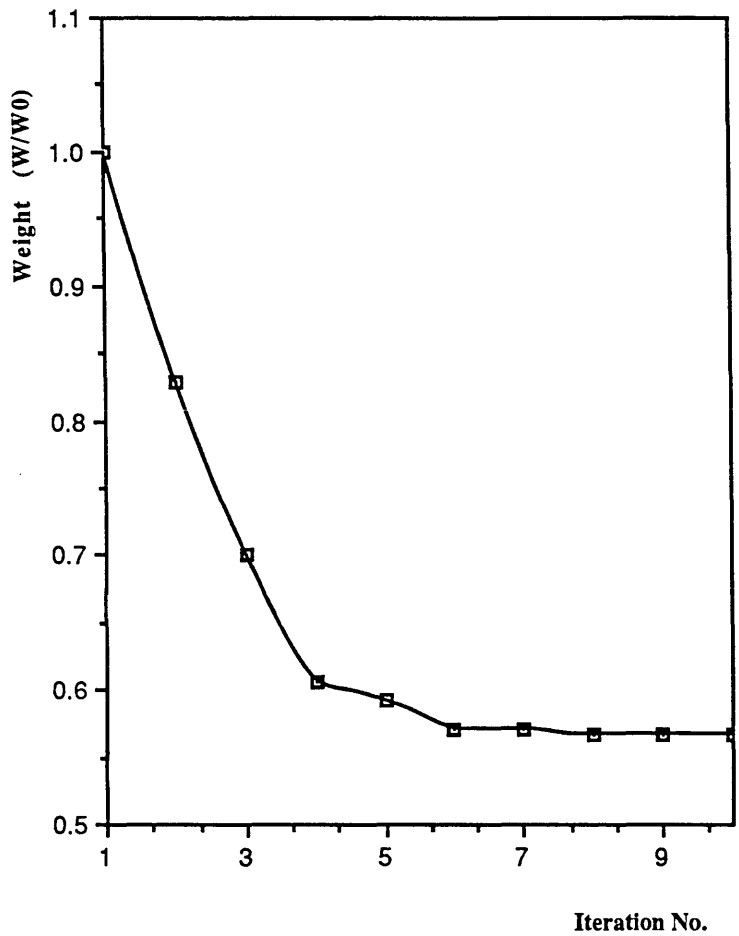


FIG. 6.21 Weight history

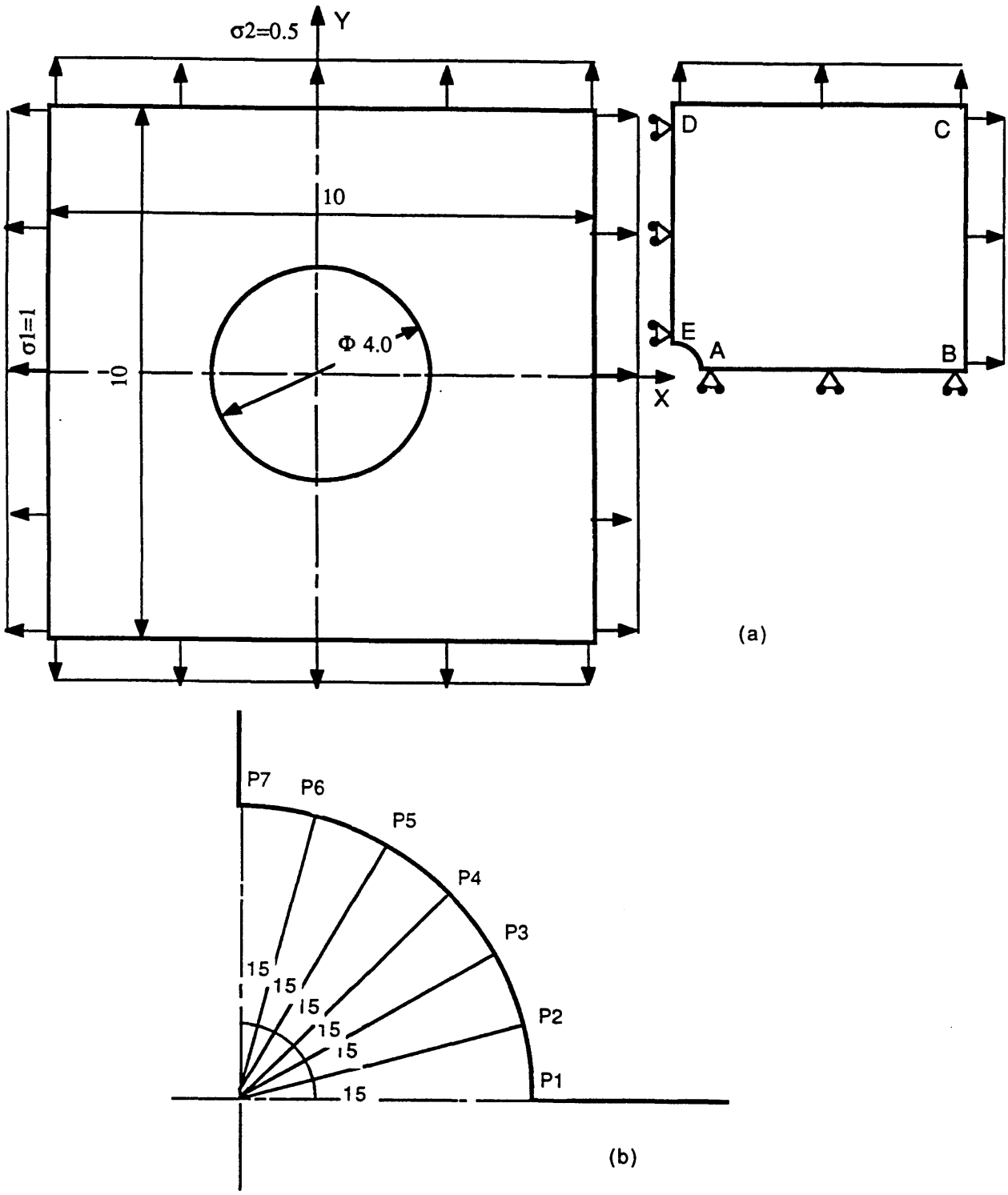


Fig. 6.22 **A hole in a square plate under biaxial stresses**

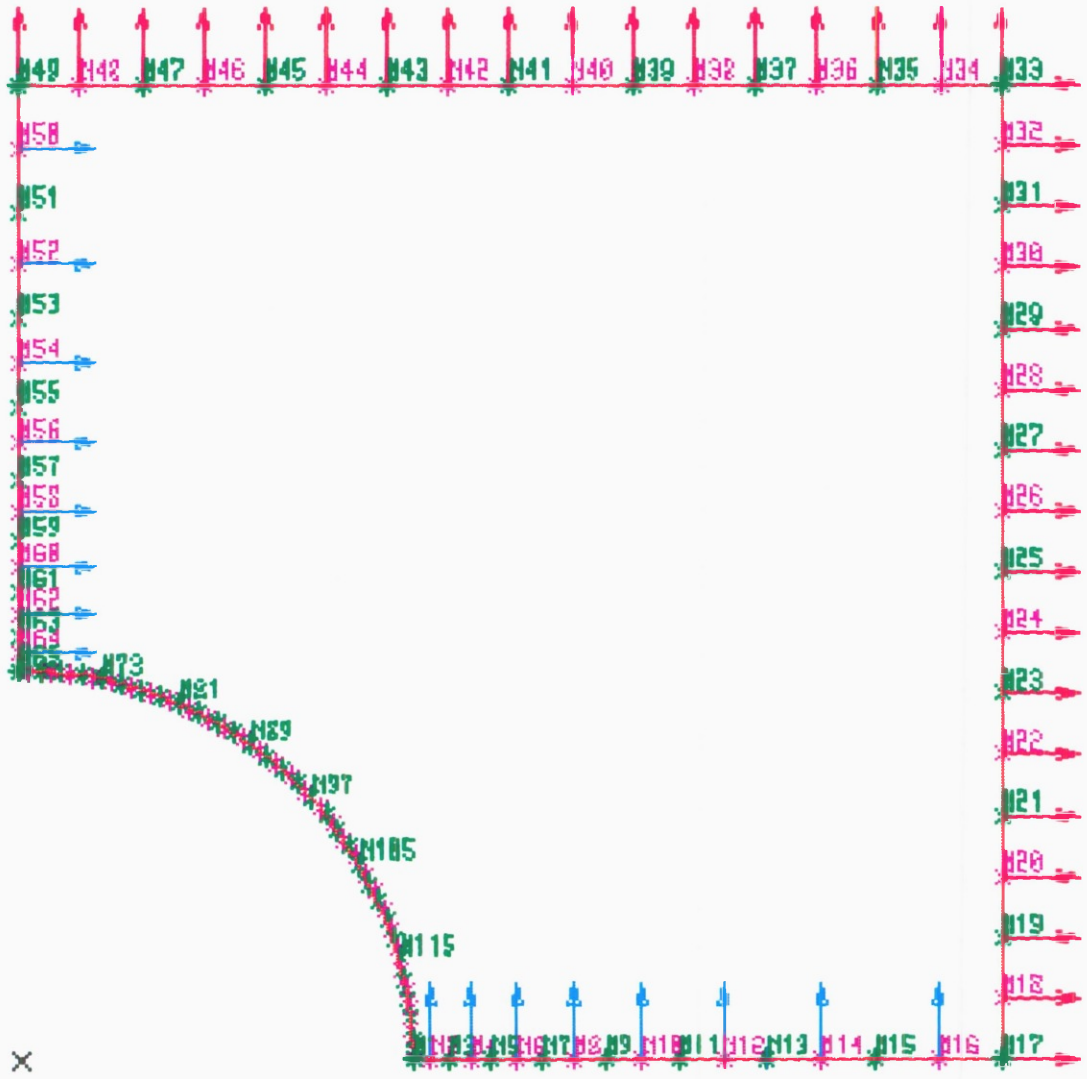


FIG. 6.23 Boundary element mesh of a quarter of the square plate with a hole

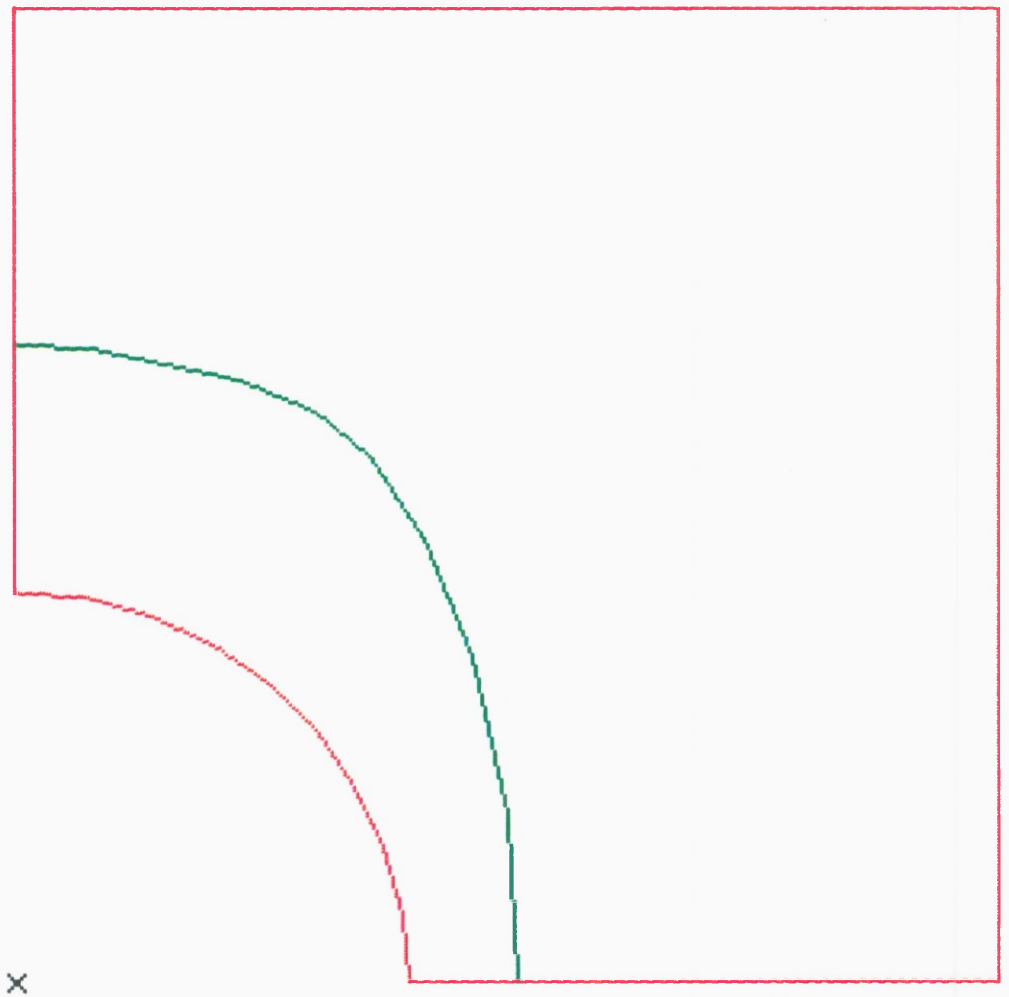


FIG. 6.24 Optimum weight design of the plate with hole
a) Initial design (red)
b) Optimum design (green)

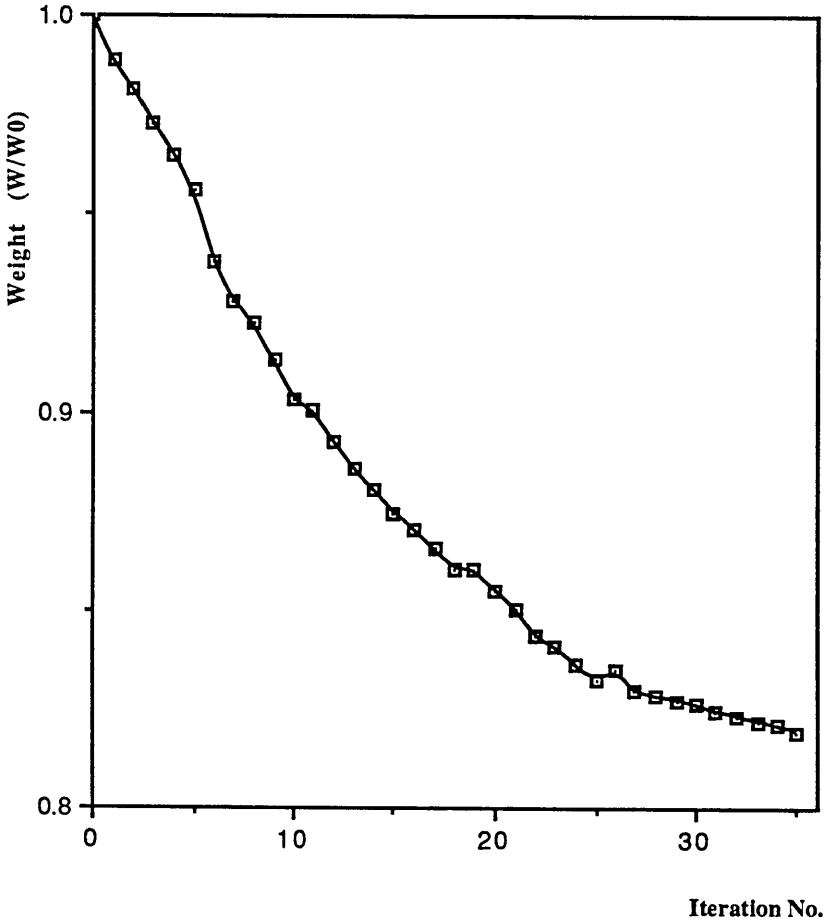


FIG. 6.25 Weight history

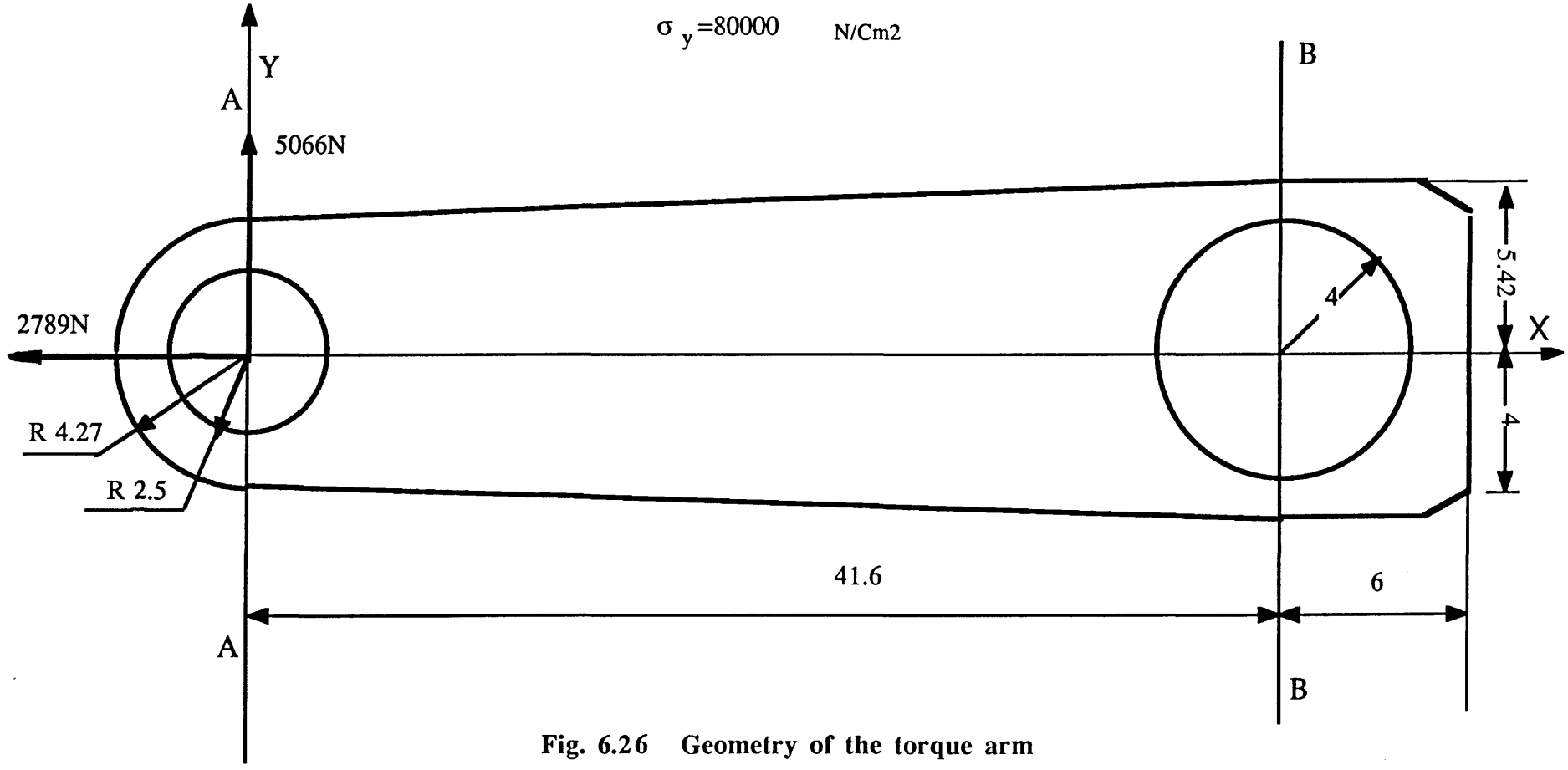


Fig. 6.26 Geometry of the torque arm



FIG. 6.27 Boundary element mesh of the torque arm

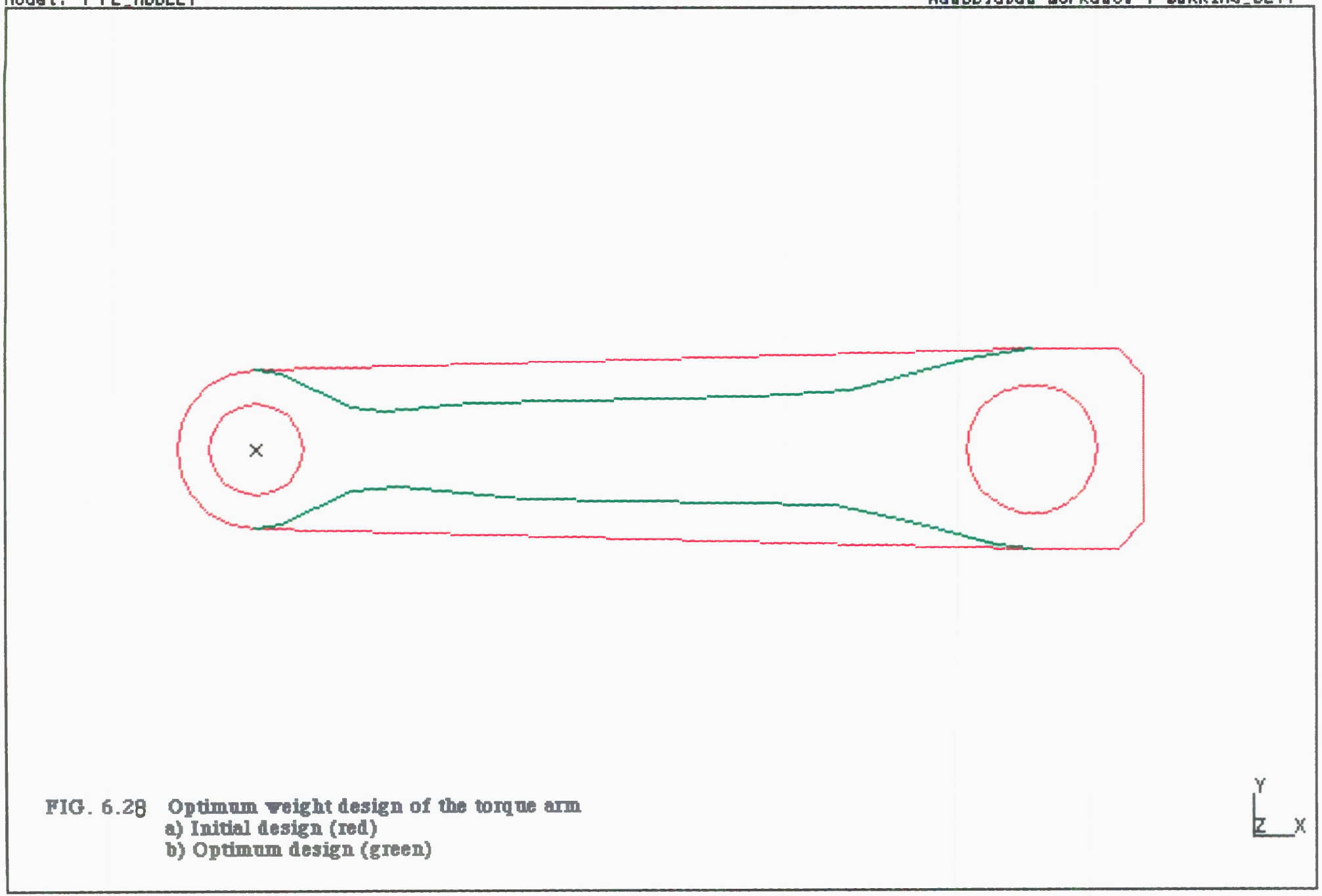


FIG. 6.28 Optimum weight design of the torque arm
a) Initial design (red)
b) Optimum design (green)

CHAPTER 7

CONCLUSIONS

Following a review of basic concepts of numerical optimization and the general mathematical problem statement, the characteristics of unconstrained and constrained optimization problems were described. Optimality conditions, the necessary mathematical conditions for a solution to be the optimum among all possible solutions, were discussed. The minimization of multivariable unconstrained functions were demonstrated. Engineering problems which require the minimization of some function without constraints can be solved by these techniques. An equally compelling reason for studying unconstrained minimization techniques is that they provide a logical way toward the solution of nonlinear constrained problems in design. Unconstrained minimization techniques are separated into zero-, first- and second-order methods. It was seen that each case has a place in automated design, depending on the nature of the problem being solved. The choice of the method which has been used for the programs of this thesis were described. In one of the computer programs, it was more convenient to use the BFGS method. This is a variable metric method and belongs to the first-order group. It converges quadratically and possesses very good stability. Because at each iteration the Hessian matrix which contains the information about the previous iterations is modified and avoids the tendency for the Hessian matrix to be singular, the BFGS method is superior to the DFP method. Some modifications have been made to the original BFGS method to ensure its convergence, which were discussed, and in practice it worked well. First, for finding the search direction at the beginning of each iteration, and second for updating the Hessian matrix. The solution of optimization problems defined by one variable, for unconstrained and constrained functions, were discussed. The Golden Section method which is a powerful technique for one-dimensional search was presented. This method can be used for even discontinuous functions, so it was preferable to Polynomial approximations.

The various techniques of solving a general nonlinear constrained optimization problem, both the direct and indirect methods, especially those which are more

applicable to solving the shape design optimization problems were discussed. Some nonlinear constrained minimization design problems can be simplified by using linear programming techniques. The basic algorithm for the solution is the Simplex method which was discussed. Each approach for solving constrained optimization problems has its own attractions as well as drawbacks, and a good understanding of these techniques aids in determining which is most useful for solving specific design problems. In the category of indirect methods, the Sequential Unconstrained Minimization Techniques (SUMT) which convert the original constrained optimization problem to a sequence of unconstrained problems through the use of penalties for constraints, were discussed in detail. In fact SUMT may be carried out in both the feasible and infeasible regions, the exterior penalty function method provides an elegant way to return to the positive domain of the design variables from design points having negative components (where the stress analysis is meaningless). A rational method for determining the transition between the interior and exterior portions, which provides a powerful algorithm for inequality constrained minimizations, was introduced. In the category of direct methods, the feasible-direction method was presented. The concepts of usable and feasible direction, the method of generating a usable feasible direction by linear programming were discussed. This method is suitable for highly nonlinear design problems when the constraints are nonlinear as well, because it deals directly with the nonlinearity of the problem.

Scaling has a significant effect on performance of the optimization process because it may overcome the numerical difficulties which may arise during the process. Scaling the constraints and making them of the same order or "equal weight" improves the optimization process and the design procedure is not dominated by a single constraint. Scaling the objective function and make it to be of order of unity in the region of interest is important especially when the objective function is very small or very large and the termination criterion is concerned with absolute change of the objective function. Normalization of the search direction specifies the important design variables in the design region. These concepts were discussed in detail.

The mathematical basis of the boundary integral equation method has been presented and the integral equation relating boundary tractions to boundary displacements for an elastic body has been derived. Numerical implementation of the technique for two-

dimensional elasto-statics was reviewed and the numerical formulation was described. The numerical approach considered involves using isoparametric boundary elements, over which both functions and geometry are defined by quadratic shape functions. Numerical formulation for calculating the weight of two-dimensional bodies by using the BIE method were presented.

The evaluation of accurate derivatives of the objective and constraint functions is of paramount importance in the shape design sensitivity analysis, which is a fundamental requirement for the shape optimization. A new formulation was presented in which the design sensitivity analysis was carried out by implicit differentiation of the structural response. the basic equation of the BIE was differentiated directly and then derivatives were calculated analytically without using a finite difference scheme. Therefore the calculation of the gradients of displacements and stresses can be achieved as part of the analysis process. In order to obtain complete generality for the initial and subsequent geometries that appear along the optimization path, it being possible to model any section with meshes as fine as needed, the coordinates of some nodes of the boundary mesh were chosen as design variables. The gradients of weight were presented in a similar way. In a test case, the accuracy of the gradients was demonstrated by comparing them against the results of the finite difference method applied to boundary element analysis. Although finite difference results were obtained using double precision (32 digits), the other results were more accurate.

The basic principles in the shape optimization process, namely shape representation, the choice of design variables and objective function, boundary element mesh generation during the process, and the solution methods were discussed. It was shown that a powerful and rather general approach for shape optimal design of elastic structures consists of using generalized design curves defined by Hermitian cubic splines. It provides a flexible and compact geometric description capability. Complex geometries can be described in a very compact way by a small number of design variables. Additional geometrical optimization constraints that are generally necessary to avoid unrealistic designs are automatically taken into account. A complete generality, for the different shapes is provided and in contrast to the other papers there is no need to change the shape representation and mesh regeneration parts of the program for each individual geometry.

In conjunction with the CAD software, I-DEAS (Integrated Design Engineering Software) [SDRC, 1986-1988], an interactive program file was developed as a pre-processor to generate boundary element meshes, create boundary conditions and tabulate physical and material properties for two-dimensional objects. Another program file was also developed as post-processor to display and tabulate analysis results.

Two computer programs, STRESOPT and WEIGHTOPT, were developed to minimize stress peaks while there are some side constraints upon the design variables and to minimize weight (mass) with stress constraints and some side constraints, respectively. Both programs have three major parts : stress analyzer, sensitivity analyzer and optimizer. The first two parts are similar in both programs which use the BIE method with quadratic boundary elements, but the difference is in the optimizer. The optimization method used in the first program was the extended penalty function and for unconstrained minimization, the BFGS method was used followed by unconstrained one-dimensional search. The method was very suitable because the objective function, the square of the deviation of the equivalent stresses around the movable part of the boundary from a desired stress, was highly nonlinear while the side constraints upon the design variables were linear. In practice the program worked well and the optimization in each case was carried out in few iterations. The optimization method used in the second program was the feasible direction method followed by constrained one-dimensional search, because the objective function, weight, and stress constraints were highly nonlinear and in this method the subsequent designs satisfy the constraints. With this, the process can be stopped after any iteration with the guarantee that the current design is valid (although is not optimum). It should also be noted that the present formulation in the optimizer in both cases can be used for other optimization problems, with different objective function and constraints with a minimum of extra work. In the programs, the whole process is carried out automatically without user intervention. The results involve the geometry of the optimum shape and the shapes obtained during the procedure are stored in a universal file which the CAD package is able to read and then plot.

In contrast to the structural optimization with sizing design variables, for the shape optimization a fixed boundary element is no longer valid because the shape of the boundary is continuously changing. In the programs, an automated mesh generator was developed. Since the movable part of the boundary does not have a regular shape, a

formula was presented to generate a two-dimensional quadratic boundary element mesh for a curve which is defined by any continuous differentiable function and it was used as a part of mesh generator.

The performances of the computer programs were tested with some examples. Four test cases were presented for the stress minimization. The first problem, which concerned the optimum shape of a hole in an infinite plate under biaxial stresses, showed that the result was very close to the analytic solution. The optimization process led to uniformity of the equivalent stresses around the hole, which agreed with the exact solution. The second example was a optimal profile design for a connecting rod in its central transition region. The same problem was solved in a paper by finite element method.They used 36 quadrilateral isoparametric elements and 256 degrees of freedom, and the process converged in 6 iterations. They used a fifth-degree polynomial curve and a horizontal line for the shape representation. Here 36 quadratic boundary elements were used. The process finished in 7 iterations with the last three being approximately coincident. Here, the flat region of the profile after optimization seemed to be more convenient. The third example was a tank with internal pressure where the optimum shape of the internal boundary was obtained. The uniform stress-distribution around the internal boundary after optimization was very satisfactory. The fourth example was a cantilever beam under uniform shear load at the end. The optimum shape of the upper edge was obtained. The comparison of maximum equivalent stress before and after optimization showed a great reduction.

Three test cases were presented for weight minimization. Firstly, a cantilever beam under lateral load with the lower edge movable. The process stopped after 9 iterations with 43.4% reduction in weight, while the maximum equivalent stress around the whole boundary before and after optimization were nearly the same. The second example was a plate under biaxial stresses with the shape of the hole movable. The final design for the hole was nearly an ellipse with reduction of 17% in weight. The last example was a torque arm, for which the optimum shape was very satisfactory and was discussed in detail.

It was shown throughout this thesis that the boundary element method, being a boundary-oriented technique is very appropriate for shape optimization design, where

only the information of the boundary is used and it can overcome the problems associated with the finite element method. The application of the BIE method to two-dimensional shape optimization design has been demonstrated. The natural developments of this work would be its extensions to axisymmetric and three-dimensional problems.

Finally it should be mentioned that the optimization techniques have no stored experience or intuition, but they enhance the possibility of obtaining improved, non-traditional designs.

APPENDIX-A

LINEAR PROGRAMMING : SIMPLEX ALGORITHM

Assume that the problem is in canonical form given by Eq.(2.55), so the solution begins from phase II. This phase consists of three steps:

The first step is to find the minimum c_i ; because the objective function will be reduced if the nonnegative cost coefficients decrease. Assume K is the column number of the minimum c_i. If this is positive, the solution is complete and the process stops, otherwise goes to step 2.

Second step is to find the minimum b_j/a_{jk} (j=1,m) for all positive a_{jk}. If all a_{jk} are negative, the solution is complete and the process stops, otherwise let the subscript r denotes the row corresponding to this minimum.

Third step is pivoting on the element a_{rk}, just as in the solution of simultaneous linear equations by Gaussian elimination. It means, row r is divided by a_{rk} then subtracted a_{jk} times row r from row j (j=1,m+1) and $j \neq r$. At this step the process begins from the first step.

The purpose of second and third steps is to put x_k in the basis and drop a variable from the basic variables, and again produce a system of equations in canonical form.

When the solution is complete these different cases may exist:

I) If all c_i associated with nonbasic variables are positive, the solution is unique.
II) If some c_i associated with a nonbasic variable is zero, the solution is not unique.
III) If some c_i is negative, the solution is unbounded.

APPENDIX-B

TWO-DIMENSIONAL QUADRATIC BOUNDARY ELEMENT MESH GENERATION OF A CURVE DEFINED WITH SPECIFIED CONTINUOUS FUNCTION

In order to generate boundary elements for two-dimensional objects, the length of the boundary is needed. During the shape optimization process, some parts of the boundary do not have a regular shape from which their lengths can be calculated analytically. Therefore, here a general formula is given from which the curve length can be calculated numerically.

Calculation of the length of a line or an arc is straightforward , but for a curve which is defined by a continuously differentiable function (meaning that the first derivative exists and is continuous everywhere), y=f(x) or x=f(y), then the length of that curve between two specified points is given by [Kreyszig, 1979]

$$L=\int_a^b \sqrt{1+\left(\frac{dy}{dx}\right)^2}\, dx = \int_a^b \sqrt{1+\left(\frac{dx}{dy}\right)^2}\, dy \qquad (B.1)$$

Consider a curve has been defined by a composite function [$x=f(\zeta)$, $y=\phi(\zeta)$] and is continuously differentiable between two points $\zeta_1=\alpha$, $\zeta_2=\beta$, then the length of the curve is given by.

$$L=\int_\alpha^\beta \sqrt{\left(\frac{dx}{d\zeta}\right)^2+\left(\frac{dy}{d\zeta}\right)^2}\, d\zeta \qquad (B.2)$$

If this curve is going to be generated to M quadratic boundary elements then the length of half an element is

$$L' = \frac{1}{2M} \int_{\alpha}^{\beta} \sqrt{\left(\frac{dx}{d\zeta}\right)^2 + \left(\frac{dy}{d\zeta}\right)^2}\, d\zeta \qquad (B.3)$$

In order to find the coordinates of each node this equation should be solved numerically (Bisection method) for each of them,

$$L' \times K - \int_{\alpha}^{\zeta_K} \sqrt{\left(\frac{dx}{d\zeta}\right)^2 + \left(\frac{dy}{d\zeta}\right)^2}\, d\zeta = 0 \qquad (B.4)$$

in which ζ_k is the unknown and is correspondent to the kth node.

Since in chapter 5, the Hermitian cubic spline is used for boundary description then the length of the spline will be calculated as follows : Using Eq.(B.2) and Eq.(5.15) the length of a Hermitian cubic element is given by

$$L = \int_{-1}^{1} \sqrt{\left[\sum_{c=1}^{2}\left[\frac{d^2M^c(\zeta)}{d\zeta^2} x_i(b,c) + \frac{d^2N^c(\zeta)}{d\zeta^2} H_i(b,c)\right]\right]^2}\, d\zeta$$

REFERENCES

Bellman, R. & Dreyfus, S. E. [1962]

" Applied Dynamic Programming",

Princeton University Press

Bennet, J. A. and Botkin, M. E. [1984]

" Structural Shape Optimization With Geometric Description and Adaptive Mesh Refinement",

AIAA Journal, Vol. 23, No. 3

Betti, E. [1872]

II NUOVO Cimento, PP 6-10

Bhavikatti, S. S. and Ramakrishnan, C. V. [1977]

" Optimum Design of Fillets in Flat and Round Tension Bars ",

ASME paper 77-DET-45, Presented at the Design Engineering Technical Conference, Chicago, IL

Bhavikatti, S. S. and Ramakrishnan, C. V. [1980]

" Optimum Shape Design of Rotating Disks ",

Comput. and Struct., 11, 397-401

de Boor, C. [1978]

" A Practical Guide to Splines ",

Springer-Verlag

Botkin, M. E. [1981]

" Shape Optimization of Plate and Shell Structures ",

AIAA Journal, Vol. 20, N0.2

Braibant, V. and Fleury, C. [1984]

" Shape Optimal Design Using B-Splines ",

Comput. Meths. Appl. Mech. and Engng, 44, 247-267

Broydon, C. G. [1970]

" The Convergence of a Class of Double Rank Minimization Algorithms",

Parts I and II, J. Inst. Math. Appl., Vol. 6, PP 76-90, 222-231

Carrol, C. W. [1961]

" The Created Response Surface Technique for Optimizing Nonlinear Restrained Systems",

Operations Research, Vol. 9, PP169-184

Casis, J.H. and Schmit, L. A. (JR). [1976]

" On Implementation of the Extended Interior Penalty Function ",

Int. J . Numer. Meths. Engng, Vol. 10, 3-23

Cherpanov, G. P. [1974]

" Inverse Problems of the Plane Theory of Elasticity "

Prikl. Mat. Mekh. 38(6), 963-979

Choi, K. K. and Haug, E. J. [1983]

" Shape Design Sensitivity Analysis of Elastic Structures ",

J. Struct. Mech., 11(2), 231-269

Cox, H. [1965]

"The Design of Structures of Least Weight ",

Pergamon, Oxford

Cruse, T. A. [1969]

" Numerical Solutions in Three Dimensional Elasto-Statics",

Int. J. Solids Struct., 5, 1259-1274

Cruse, T. A. [1973]

" Application of the Boundary Integral Equation Method to Three Dimensional Stress Analysis",

Comp. & Struct., 3, 509-527

Cruse, T. A. [1974]

" An Improved BIE Method for Three Dimensional Elastic Stress Analysis",

Comp. & Struct., 4, 741-754

Cruse, T. A. [1977]

" Mathematical Foundations of the Boundary Integral Equation in Solid Mechanics",

Pratt & Whitney, Aircraft Group Report, No. AFOSR-TR-77-1002

Cruse, T. A. [1978]

" Two-Dimensional BIE Fracture Mechanics Analysis",

App. Math. Modelling, 2, 287-293

Dantzig, G. B. [1948]

" Programming in a Linear Structure",

Comptroller, USAF, Washington, D.C.

Dantzig, G. B. [1963]

"Linear Programming and Extensions",

Princeton University Press

Davidon, W. C. [1959]

" Variable Metric Method for Minimization",

Argone National Laboratory, ANL-5990 Rev., University of Chicago

Dems, K. [1980]

" Multiparameter Shape Optimization of Elastic Bars in Torsion "

Int. J. Numer. Meths. Engineering, Vol.15, PP 1517-1539

Dems, K. and Mroz, Z. [1978]

" Multiparameter Structural Shape Optimization by the Finite Element Method ",

Int. J. Numer. Meths. Engng, Vol. 13, PP247-263

Durelli, A. J. and Rajaiah, K. [1979]

" Optimum Hole Shapes in Finite Plates under Uniaxial Load ",

ASME J. Appl. Mech. ,46, 691-695

Durelli, A. J. ; Erickson, M. and Rajaiah, K. [1981]

" Optimum Shapes of Central Holes in Square Plates Subjected to Uniaxial Uniform Load",

Int. J. Solids and Structures, 17, 787-793

Eizadian, D. and Trompette, P. [1985]

" Shape Optimization of Bidimensional Structures by the Boundary Element Method ",

Paper presented at the University Of Arizona CAD/CAM, Robotics and automation Conference, Tucson, AZ

Epsing, B.J.D. [1985]

"A CAD Approach to the Minimum Weight Design Problem ",

Int. J. Numer. Meths. Engng, Vol. 21, PP 1049-1066

Ergatoudis, J. G [1968]

" Isoparametric Finite Elements in Two- and Three-Dimensional Stress Analysis",

Ph.D Thesis, University of Wales, Swansea

Faux, I. D. and Pratt, M. J. [1979]

" Computational Geometry for Design and Manufacture ",

Ellis Horwood Limited.

Fenner, R. T. [1983]

" The Boundary Integral Equation Method in Engineering Stress Analysis",

J. Strain Analysis, Vol. 18, No. 4

Fenner, R. T. [1989]

" Stress Analysis Using Boundary Elements",

Institute of Physics Conference on Modern Practice in Stress and Vibration Analysis

Fiacco, A. V. & McCormick, G. P. [1968]

" Nonlinear Programming : Sequential Unconstrained Minimization Techniques",

John Wiley & Sons, USA

Fletcher, R. [1970]

" A New Approach to Variable Metric Algorithms ",

Computer J., Vol. 13, No. 3

Fletcher, R. [1980]

" Practical Methods of Optimization, Vol. I : Unconstrained Optimization",

John Wiley & Sons, Chichester

Fletcher, R. [1981]

" Practical Methods of Optimization, Vol. II : Constrained Optimization",

John Wiley & Sons, Chichester

Fletcher, R. & Powell, M. J. D. [1963]

" A rapidly Convergent Method for Minimization",

Computer J. Vol. 6, No. 2, PP 163-168

Fox, R. L. [1971]

" Optimization Methods for Engineering Design",

Addison-Wesley Publishing Comp.

Francavilla, A. ; Ramakrishnan, C. V. and Zienkiewicz, O. C. [1975]

" Optimization of Shape to Minimize Stress Concentration ",

Journal of Strain Analysis, Vol. 10, No. 2

Fung, Y. C. [1965]

" Foundations of Solid Mechanics",

Prentice-Hall, Englewood Cliffs, New Jersey

Gerard, G. [1956]

" Minimum Weight Analysis of Compression Structures ",

New York University Press

Gill, P. E. & Murray, W. (Eds.) [1974]

" Numerical Methods for Constrained Optimization ",

Academic Press, London and New York

Gill, P. E. ; Murray, W. and Wright, M. H. [1981]

" Practical optimization ",

Academic Press

Goldfarb, D. [1970]

" A Family of Variable Metric Methods Derived by Variational Means",

Math. Comput., Vol. 24, PP 23-36

Goult, R. J. ; Hoskins, R. F. ; Milner, J. A. and Pratt, M. J. [1974]

" Computational Methods in Linear Algebra",

Stanley Thornes (Publishers) Ltd.

Haftka, R. T. and Grandhi, R. V. [1986]

" Structural Shape Optimization - A survey",

Comput. Meths. App. Mech. Eng. Vol. 57, No. 1, PP 91-106

Haftka, R. T. & Starnes, J. H. [1976]

" Applications of a Quadratic Extended Interior Penalty Function for Structural Optimization",

AIAA J., Vol. 14, No. 6, PP. 718-724

Haug, E. J. ; Choi, K. K. ; Hou, Y. M. & Yoo, Y. M. [1984]

" A Variational Method for Shape Optimal Design of Elastic Structures, in New Directions in Optimum structural design",

(E. Artek, R. H. Gallagher, K. M. Ragsdell, and O. C. Zienkiewicz, eds), Wiley, Chichester

Haug, E. J. ; Choi, K. K. and Komkov, V. [1986]

"Design sensitivity analysis of structural systems",

Academic Press, Inc.

Haug, E. J. & Rousselet, B. [1980]

" Design Sensitivity Analysis in Structural Mechanics. I. Eigenvalue Variations",

J. Struct. Mech. 8(1) : 17-41

Haug, E. J. & Rousselet, B. [1980]

" Design Sensitivity Analysis in Structural Mechanics. II. Eigenvalue Variations",

J. Struct. Mech. 8(2) : 161-186

Hemada, M. ; Seguchi, Y. and Tada, Y. [1980]

" Shape Determination Problems of Structures by the Inverse Variational Principle",

Bull. JSME 23-184 (10) 1581

Hemp, W. [1958]

" Theory of Structural Design ",

Report 214, AGARD

Hooke, R. & Jeeves, T. A. [1961]

" Direct Search Solution of Numerical and Statistical Problems",

J. Assoc. Computer Mach., Vol. 8, PP 212-229

Imai, K. [1978]

" Configuration Optimization of Trusses by the Multiplier Method",

Ph.D Thesis, University of California, Los Angeles

Imai, K. & Schmit, L. A. [1981]

"Configuration Optimization of Trusses",

ASCE, J. Struct. Div., Vol. 107, No. ST5, PP 745-756

Imam, M. H. [1982]

" Three-Dimensional Shape Optimization ",

Int. J. Numer. Meth. Engng, Vol. 18, 661-673

Jacoby, S. L. S. ; Howalik, J. S. & Pizzo, J. T. [1972]

" Iterative Methods for Nonlinear Optimization Problems"

Prantice-Hall, Inc.

Jaswon, M. A. [1963]

" Integral Equation Methods in Potential Theory I",

Proc. R. Soc. London, A275, PP. 23-32

Jaswon, M. A. & Symm, G. T. [1977]

" Integral Equation Methods in Potential Theory and Elasto-statics",

Academic Press, London

Kavlie, W. C. & Moe, J. [1971]

" Automated Design of Frame Structures",

J. Struct. Div., ASCE, 97, 33-62

Kelly, J. E. [1960]

" The Cutting Plane Method for Solving Convex Programs",

J. of SIAM, Vol.III, No.4, PP. 703-712

Kreyszig, E. [1979]

"Advanced Engineering Mathematics",

John Wiley & Sons, USA

Kristensen, E. S. and Fleming Madsen, N. [1976]

" On the Optimum Shape of Fillets in Plates Subjected to Multiple in-Plane Loading Cases ",

Int. J. Num. Meths in Engng, Vol. 10, 1007-1019

Kuhn, H. W. and Tucker, A. [1951]

" Nonlinear Programming ",

In Proceedings of the Second Berkeley Symposium on Mathematical Statistic and probability, University Of California Press, Berkeley

Lachat, J. C. [1975]

" A Further Development of the Boundary Integral Technique for Elasto-statics",

Ph.D Thesis, University of Southampton

Lachat, J. C. & Watson, J. O. [1976]

" Effective Numerical Treatment of Boundary Integral Equations: A Formulation for Three-Dimensional Elasto-statics",

Int. J. Num. Meths. Engng, 10, 991-1005

Lauricella, G. [1909]

" Sur L'intègration de L'equation Relative à L'equilibre des Plagues Elastiques Encastrèes",

Acta Math. 34

Livesley, R. K. [1956]

" The Automatic Design of Structural Frames",

Quart. J. Mech. Appl. Math., 9, Part 3

Luchi, M. L. ; Poggialini, A. and Persiani, F. [1980]

" An Interactive Optimization Procedure Applied to the Design of Gas Turbine Discs ",

Comput. Struct., 11, 629-637

Maxwell, J. C. [1869]

"Scientific Papers",

Vol. 2, P175

Mendleson, A. [1968]

" Plasticity : Theory and Applications ",

McMillan Comp., U.S.A.

Mendleson, A. [1973]

" Boundary Integral Methods in Elasticity and Plasticity",

NASA, TN D-7418

Mendleson, A. [1975]

" Solution of Elasto-plastic Torsion Problem by Boundary Intergal Method",

NASA, TN D-7872

Michell, A. G. M. [1904]

" The Limits of Economy of Material in Framed Structures",

Phil. Mag. (Series 6), 8, 589-597

Moses, F. & Onoda, S. [1969]

" Minimum Weight Design of Structures with Application to Elastic Grillages",

Int. J. Numer. Meths. Engng.

Murray, W.(Ed.) [1972]

" Numerical Methods for Unconstrained Optimization",

Academic Press, London and New York

Na, M. S. ; Kikuchi, N. and Taylor, J. E. [1983]

" Optimal Modification of Shape for Two-Dimensional Elastic Bodies",

J. Struct. Mech., 11(1), 111-135

Na, M. S. ; Kikuchi, N. and Taylor, J. E. [1984]

"Optimal Shape Remodelling of Linearly Elastic Plates Using Finite Element Methods",

Int. J. Numer. Meths. Engng. 20, 1823-1840

Oda, J. [1977]

" On a Technique to Obtain an Optimum Strength Shape by the Finite Element Method",

Bull. JSME 20, 160-167

Oda, J. and Yamazaki, K. [1977]

" On a Technique to Obtain an Optimum Strength Shape of Axisymmetric Body by the Finite Element Method ",

Bull. JSME 20 (150), 1524-1532

Oda, J. & Yamazaki, K. [1979]

" On a Technique to Obtain an Optimum Strength Shape by the Finite Element Method : Application to the Problem under Body Force"

, Bull. JSME 22, 131-140

Oda, J. and Yamazaki, K. [1981]

" Pattern Transformation Method for Shape Optimization and its Application to Spoked rotatry disks",

International Symposium On Optimum Structural Design, Tucson, AZ, 4.29-4.35

Parasad, B. and Emerson, J. F. [1984]

" Optimal Structural Remodelling of Multi-Objective Systems ",

Comp. & Struct. 18(4), 619-628

Parasad, B. and Haftka, R. T. [1979]

" Optimal Structural Design with Plate Finite Elements",

J. Struct. Div. ASCE ST11, 2367-2382

Pearson, C. [1958]

" Structural Design by High-Speed Computing Machines",

Proc. of ASCE Conf. on Electronic Computation, Kansas City, Missouri

Pedersen, P. and Laursen, C. L. [1982-83]

" Design for Minimum Stress Concentration by Finite Elements and Linear Programming",

J. Struct. Mech., 10(4), 375-391

Pierre, D. A. & Lowe, M. J. [1975]

" Mathematical Programming via Augmented Lagrangians",

Applied Mathematics and Computation Series, Addison-Wesley, Reading Mass.

Powell, M. J. D. [1964]

" An Efficient Method for Finding the Minimum of a Function of Several Variables Without Calculating Derivatives",

Comput. J. Vol. 7, No. 4, PP. 303-307

Powell, M. J . D. [1981]

" Approximation Theory and Methods",

Cambridge University Press

Prager, W. and Marcal, P. [1971]

"Optimality Criteria in Structural Design ",

AFFDL-TR-70-166

Prenter, P. M. [1975]

" Splines and Variational Methods",

A Wiley-Interscience Publication, John Wiley & sons

Queau, J. P. and Trompette, PH. [1980]

" Two Dimensional Shape Optimal Design by the Finite Element Method",

Int. J. Numer. Meths. Engng, Vol. 15, 1603-1612

Rajaiah, K and Durelli, A. J. [1981]

" Optimization of Hole Shapes in Circular Shells under Axial Tension",

Experimental Mechanics, 21, 201-204

Rajaiah, K. and Kumar, R. R. [1982]

" Optimum Hole Shapes in Circular Shells under Tension and Torsion ",

Mech. Res. Comm., 9(2), 85-93

Ramakrishnan, C. V. and Francavilla, A. [1975]

" Structural Shape Optimization Using Penalty Functions",

J. Struct. Mech., 3(4), 403-422

Rao, S. S. [1978]

" Optimization Theory and Applications",

Wiley Eastern Limited, India

Reinschmidt, K. F. ; Cornell, C. A. & Brotchie, J. F. [1966]

" Iterative Design and Structural Optimization",

J. Struct. Div. ASCE 92, (ST6) 281-319

Remzi, E. M. [1981]

" Boundary Integral Stress Analysis of Incompressible and Nearly Incompressible Materials",

Ph.D. Thesis, Imperial College, University of London

Rickettes, R. E. and Zienkiewicz, O. C. [1984]

" Shape Optimization of Continum Structures. In New Direction in Optimum

Structural Design",

Edited by E. Atrek, R. H. Gallagher, K. M. Ragsdell and O. C. Zienkiewicz, 139-166, John Wiley, New York

Riesenfeld, R. F. [1973]

" Applications of B-Spline Approximation to Geometric Problems of Computer-Aided Design",

Ph.D. Dissertation, Syracuse University

Rizzo, F. J. [1967]

" An Integral Equation Approach to Boundary Value Problems of Classical Elasto-statics",

J. Appl. Math., 25, PP. 83-95

Rockafellar, R. T. [1973]

" The Multiplier Method of Hestenes and Powell Applied to Convex Programming",

J. Optim Theory Appl., Vol.12, No.6, PP555-562

Rogers, D. F. and Alan Adams, J. [1976]

" Mathematical Elements for Computer Graphics",

McGraw-Hill Book Company

Rosen, J. B. [1961]

" The Gradient Projection Method for Nonlinear Programming, Part 1 : Linear Constraints",

J. of the Society for Industrial and Applied Mathematics, Vol.9, PP.514-532

Rosenbrock, H. H. [1960]

" An Automated Method for Finding the Greatest or Least Value of a Function",

Computer J., Vol. 3, PP 175-184

Rousselet, B. & Haug, E. J. [1981]

" Design Sensitivity Analysis of Shape Variation",

In Optimization of Distributed Parameter Structures (E. J. Haug and J. Cea, eds),

Sijthoff & Noordhoff, Alphen aan den Rijn, Netherlands

Schmit, L. [1960]

" Structural Design by Systematic Synthesis",

Proc. of ASCE 2nd Conf. on Electronic Computation, Pittsburgh, Pa.

Schnack, E. [1979]

" An Optimization Procedure for Stress Concentration by the Finite Element

Technique ",

Int. J. Numer. Meths. in Engng., Vol. 14, 115-124

Schnack, E. [1983]

" Computer-Simulation of an Experimental Method for Notch Shape Optimization",

In : J. Burger and Y. Jarny, eds., Simulation in Engineering Sciences {Elsevier

Science Publishers, Amsterdam, 1983), 311-316

Schumaker, L. L. [1981]

" Spline Functions Basic Theory ",

A Wiley-Interscience Publication, John Wiley $ Sons

SDRC [1986]

"Geomod, Solid Modelling and Design User's Guide ",

Structural Dynamics Research Corporation, U.S.A.

SDRC [1986]

" Supertab, Engineering Analysis Model Solution and Optimization",

Structural Dynamics Research Corporation, U.S.A.

SDRC [1986]

" Supertab, Engineering Analysis Pre- and Post-Processing ",

Structural Dynamics Research Corporation, U.S.A.

SDRC [1986]

" Technical Overview User's Guide ",

Structural Dynamics Research Corporation, U.S.A.

Seguchi, Y. & Tada, Y. [1977-78]

" Shape Determination Problems of Structures by Inverse Variational Principle / The Finite Element Formulation",

L. C. Wellford, Jr., Ed. , Proceedings Symposium on Application of Computational Methods in Engineering, Vol. II, 899

Shanley, F. [1960]

" Weight-Strength Analysis of Aircraft Structures",

Dover, New York

Shanno, D. F. [1970]

" Conditioning of Quasi-Newton Methods for Function Minimization",

Math. Comput., Vol. 24, PP 647-656

Sharpe, R. [1969]

" The Optimum Design of Arch Dams",

Proceedings, Institute of Civil Engineering, Paper 7200s, Suppl. Vol. 73-98

Skrobanski, J. J. [1986]

" Optimization Subject to Nonlinear Constraints",

Ph.D Thesis, Imperial College, University of London

Soares, C. A. M. ; Rodrigues, H. C. ; Faria, L. M. O. and Haug, E. J. [1984]

" Optimization of the Geometry of Shafts Using Boundary Elements",

ASME J. Mechanics, Transmissions and Automation in Design, 106(2), 199-203

Soares, C. A. M. ; Rodrigues, H. C. and Choi, K. K. [1984]

" Shape Optimal Structural Design Using Boundary Elements and minimum compliance technique"

ASME Paper 84-DET-57 Presented at the Design Engineering Conference, Cambridge, MA

Somigliana, C. [1885]

II NUOVO Cimento, PP 17-20

Spoerl, U. [1986]

"SURFOPT : Shape Optimal Design for Minimum Stress Concentration by Finite Elements with the program system 'SURFOPT' ",

Edited by Lari, Structural Analysis System 1, Pergamon Press, Oxford, Vol. 3, 239-250

Symm, G. T. [1963]

" Integral Equation Methods in Potential Theory II",

Proc. R. Soc. London, A275, PP. 33-46

Tada, Y. & Seguchi, Y. [1981]

" Shape Determination of Structures Based on the Inverse Variational Principle / The Finite Element Approach",

Int. Symposium on Optimum Structural Design, Tucson, AZ, 5.1-5.7

Tan, C. L. [1979]

" Three-Dimensional Boundary Integral Equation Stress Analysis of Cracked Components",

Ph.D Thesis, University of London

Tan, C. L. & Fenner, R. T. [1979]

" Elastic Fracture Mechanics Analysis by the Boundary Integral Equation Method",

Proc. R. Soc. Lond., A369, 243-260

Taylor, J. [1969]

" Optimal Design of Structural Systems : An Energy Formulation",

AIAA J., 7, 1404-1406

Timoshenko, S. P. & Goodier, J. N. [1970]

" Theory of Elasticity ",

McGraw-Hill, U.S.A

Vanderplaats, G. N. [1984]

" Numerical Optimization Techniques for Engineering Design with Applications",

McGraw-Hill Comp.

Vanderplaats, G. N. & Moses, F. [1973]

" Structural Optimization by Methods of Feasible Directions",

Comput. Struct. Vol. 3, PP 739-755, Pergamon Press, Printed in Great Britain

Walsh, G. R. [1975]

" Methods of Optimization",

John Wiley & Sons, U.K.

Wang, S. Y.; Sun, Y. B. & Gallagher, R. H. [1985]

"Sensitivity Analysis in Shape Optimization of Continum Structures",

Comput. Struct. 20, 855-867

Wassermann, K. [1983-84]

"Three-Dimensional Shape Optimization of Arch Dams with Prescribed Shape Functions",

J. Struct. Mech. 11, 465-489

Watson, J. O. [1982]

" Hermitian Cubic Boundary Elements for Plane Problems of Fracture Mechanics",

Int. J. Struct. Mech. and Materials Science, Vol. 4, No.1

Weck, M. and Steinke, P. [1983-84]

"An Efficient Technique in Shape Optimization",

J. Struct. Mech., 11(4), 433-449

Wills, J. A. [1973]

" A Mathematical Optimization Procedure and its Application to the Design of

Bridge Structures",

Transport and Road Research Laboratory, TRRL Report 555

Wolfe, P. [1965]

" The Product Form of the Simplex Method"

Report P-3075, The RAND Corporation, Santa Monica, Cal.

Yang, R. J. & K. K. Choi [1985]

" Accuracy of Finite Element Based Shape Design Sensitivity Analysis",

J. Struct. Mech., 13(2), 223-239

Zangwill, W. I. [1969]

" Nonlinear Programming : A Unified Approach",

Prentice-Hall, Englewood Cliffs, N. J.

Zienkiewicz, O. C. & Campbell, J. S. [1973]

" Shape Optimization and Sequential Linear Programming",

In : R. H. Gallagher and O. C. Zienkiewicz, Eds, Optimum Structural Design

(Wiley,Newyork), 109-126

Zoutendijk, G. [1960]

" Methods of Feasible Directions",

Elsvier, Amsterdam

Zoutendijk, G. [1976]

" Mathematical Programming Methods",

North-Holand Company