An adaptive grid generation technique for the boundary element method.

by

Dun Leng

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Engineering Mechanics

Program of Study Committee:
Ambar.K.Mitra, Major Professor
Vinay Dayal
Fouad Fanous

Iowa State University

Ames, Iowa

2001

Graduate College
Iowa State University

This is to certify that the master's thesis of

Dun Leng

has met the thesis requirements of Iowa State University

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

The boundary element method is applied to elasticity problems in two-dimensions. The boundary is discretized into straight boundary elements. Second order polynomials are used to represent displacement and traction on each element. Length of each element is determined by utilizing a spline-based grid optimization scheme. Various error norms are controlled based on the equilibrium conditions. The performance of the optimization scheme is assessed and convergence patterns are demonstrated.

INTRODUCTION

The introductory work on the boundary element method (BEM) was done by Fredholm in 1905 when he was investigating the solution of integral equations by using a discretization approach. A more formal study of BEM was conducted by Mikhlin[1]. He studied integral equations with both scalar and vector integrals and in particular those with singular kernels.

BEM requires more elaborate computational tools compared to the finite element method (FEM). The first successful use of BEM in solving plane, isotropic elasticity problems was demonstrated by Rizzo[2].

With the development of powerful computers and better understanding for numerical and computational techniques, BEM code developers continue their steps to solve various engineering problems in the area of fluid flow, plasticity, scattering, and in recent years, in the area of nano-mechanics.

The principal advantage of BEM is that, the surface integrals in two dimensions are reduced to line integrals and volume integrals in three dimensions are reduced to surface integrals. This reduction in dimensionality reduces the effort in grid generation and in solving problems posed on an infinite domain.

In the present work, the BEM is employed for the solution of problems in two-dimensional elasticity. The partial differential equations are converted into integral equations where the integrations are performed along the boundary of the domain. The integrands are

products of two physical variables, such as displacement and traction, and two kernel functions that are the fundamental solutions for displacement and traction. The integral equations are discretized by partitioning the boundary into elements.

When curved elements are used in this discretization, a Jacobian for the transformation from curved elements to straight elements appears in the integrand. In such a situation, the integrands can only be evaluated by numerical quadrature. In order to eliminate the error in the quadrature, straight elements are used in the present work. However, the lengths of these straight elements are to be determined in such a fashion that the original curved boundaries of the domain are represented within a certain specified tolerance. The determination of the lengths of the elements is a part of the adaptive grid generation technique described in this work.

The physical variables, e.g. displacement and traction, are represented along each element in terms of the nodal values of the variables and polynomial shape functions. In general, the accuracy of such representation improves as the element size is reduced or as the order of the polynomial shape functions is increased. Naturally, higher order polynomials involve larger computational effort. In the present work, linear shape functions are employed. The accuracy of the solution is enhanced by determining the lengths of the elements that yields a solution with certain specified tolerance.

The strength of the present work is in the development of an adaptive grid generation scheme that encloses two aspects of grid improvement, the representation of geometry and the representation of the functions, within the same framework.

In Chapter 1, the basic equations of BEM are developed. In Chapter 2, the numerical implementation of the BEM is described. In Chapters 3 and 4, the grid optimization and its controls are developed, respectively. In Chapter 5, solutions of three example problems are shown to demonstrate the performance of the technique.

CHAPTER ONE. BASIC BOUNDARY ELEMENT METHOD

Basic BEM theory

The theory is first demonstrated for the solution of $\nabla^2 u(p)=0$ where the scalar u can be pressure, potential, or temperature.

Figure 1.1 shows a two-dimensional domain R and function f(x,y) with dR as a surface element and dB as a boundary element. Function f(x,y) is analytic in the region R with boundary B. One can show

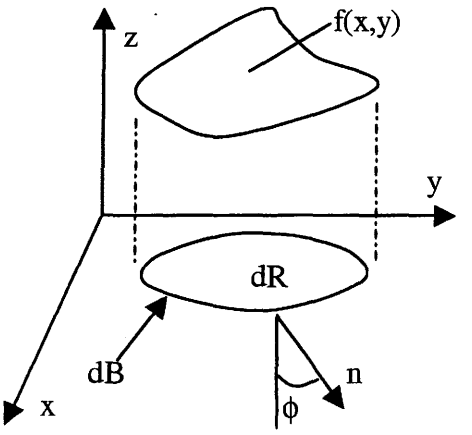

Figure 1.1: General 2-D problem model

$$\int_R \frac{\partial f}{\partial x}\, dR = \oint_B f \cos\phi \, dB \tag{1.1a}$$

$$\int_R \frac{\partial f}{\partial y}\, dR = \oint_B f \sin\phi \, dB \tag{1.1b}$$

where ϕ is the angle between the normal n drawn on dB and the x-axis. When f(x,y)=g(x,y)h(x,y), then

$$\int_R (g\frac{\partial h}{\partial x} + h\frac{\partial g}{\partial x})\, dR = \oint_B gh \cos\phi \, dB \tag{1.2 a}$$

$$\int_R (g\frac{\partial h}{\partial y} + h\frac{\partial g}{\partial y})\, dR = \oint_B gh \sin\phi \, dB \tag{1.2 b}$$

After substituting $h=\frac{\partial u}{\partial x}$, Eq.(1.2a) becomes

$$\int_R (g\frac{\partial^2 u}{\partial x^2} + \frac{\partial u}{\partial x}\frac{\partial g}{\partial x})\, dR = \oint_B g\frac{\partial u}{\partial x} \cos\phi \, dB \tag{1.3a}$$

With $h=\frac{\partial u}{\partial y}$, one obtains from Eq.(1.2b)

$$\int_R (g\frac{\partial^2 u}{\partial y^2} + \frac{\partial u}{\partial y}\frac{\partial g}{\partial y})\, dR = \oint_B g\frac{\partial u}{\partial y} \sin\phi \, dB \tag{1.3b}$$

Adding Eqs.(1.3a) and (1.3b), we get

$$\int_R g\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right) dR+\int_R \left(\frac{\partial u}{\partial x}\frac{\partial g}{\partial x}+\frac{\partial u}{\partial y}\frac{\partial g}{\partial y}\right) dR=\oint_B g\left(\frac{\partial u}{\partial x}\cos\phi+\frac{\partial u}{\partial y}\sin\phi\right) dB \tag{1.4}$$

Introducing the symbols $\nabla^2 u$ and $\nabla u \bullet n = \frac{\partial u}{\partial n}$, one can rewrite (1.4) as

$$\int_R g\nabla^2 u\, dR+\int_R (\nabla u)\bullet(\nabla g)\, dR=\oint_B g\frac{\partial u}{\partial n} dB \tag{1.5}$$

In equation (1.5), there are two variables g and u, and there are two integrals over the whole region. The goal for BEM is to integrate only on the boundary. To eliminate the second integral over R, g and u can be interchanged in Eq.(1.5) to yield

$$\int_R u\nabla^2 g\, dR+\int_R (\nabla u)\bullet(\nabla g)\, dR=\oint_B u\frac{\partial g}{\partial n} dB \tag{1.6}$$

By subtracting Eq.(1.6) from Eq.(1.5), we get

$$\int_R (g\nabla^2 u - u\nabla^2 g)\, dR = \oint_B \left(g\frac{\partial u}{\partial n} - u\frac{\partial g}{\partial n}\right) dB \tag{1.7}$$

As u satisfies the Laplace equation $\nabla^2 u=0$, so $g\nabla^2 u=0$. To eliminate the integral $\int_R -u\nabla^2 g\, dR$ in the two-dimensional region, the function g could be arbitrarily defined.

Consider two points $p(x_0,y_0)$ and $q(x,y)$ and define $r(p,q)=|p-q|$. Then g can be defined as $g=\frac{1}{2\pi}\ln r(p,q)$ and $\nabla^2 g = \frac{1}{2\pi}\nabla^2 \ln r(p,q)$. Since $\nabla^2 \ln r(p,q)=2\pi\delta(q-p)$, and $\nabla^2 g = \delta(q-p)$, Eq.(1.7) becomes

$$c(p)u(p)= \frac{1}{2\pi} \oint_{B(q)} [u(q)\frac{\partial}{\partial n(q)} \log r(p,q) - \frac{\partial u(q)}{\partial n(q)} \log r(p,q)] \; dB(q) \tag{1.8}$$

To calculate the coefficient c(p), substitute u(q)=1 in Eq.(1.8) to obtain

$$c(p)= \frac{1}{2\pi} \oint_{B(q)} \frac{\partial}{\partial n(q)} \log r(p,q) \; dB(q) \tag{1.9}$$

or

$$c(p)= \frac{1}{2\pi} \oint_{B(q)} d\theta \tag{1.10}$$

If p is on a smooth boundary shown in Figure 1.2,

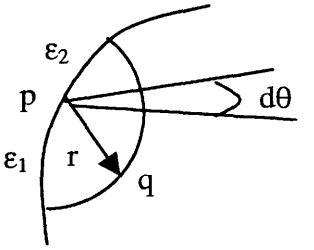

Figure 1.2: c(p) integration on boundary

$$\oint_{B(q)} d\theta = \lim_{\substack{\varepsilon 1 \to 0 \\ \varepsilon 2 \to 0}} \int_{\theta A(\varepsilon 1)}^{\theta B(\varepsilon 2)} d\theta = \int_0^{\pi} d\theta = \pi$$

then $c(p)=\frac{1}{2}$.

If the boundary has a corner, and the corner has an included angle ϕ, $c(p)=\frac{\phi}{2\pi}$ at that corner.

If p is located in the interior of the domain,

$$\oint_{B(q)} d\theta = \int_0^{2\pi} d\theta = 2\pi$$

then c(p)=1.

When p is outside the domain,

$$\oint_{B(q)} d\theta = \int_0^{0} d\theta = 0$$

then c(p)=0.

The Eq.(1.8) is solved by using the method of collocation. The u(q) and $\partial u(q)/\partial n(q)$ along the boundary are expressed in terms of N number of nodal values of these quantities and appropriate shape functions. At each of these nodes, either u(q) or $\partial u(q)/\partial n(q)$ is given as boundary condition. Thus, Eq.(1.8) has N unknowns at N number of nodes. By collocating Eq.(1.8) at the N nodes, N equations are obtained. These equations are solved to obtain the unknowns.

When both of the quantities, u(q) and $\partial u(q)/\partial n(q)$, are known on the boundary, Eq.(1.8) can be utilized to obtain u(p), anywhere within the domain.

BEM in elastostatics

Consider a function u that satisfies the general partial differential equation L(u)= -b in the domain B. The Green's reciprocal identity for L can be written as[3]

$$\int_R [u\bar{L}(v) - vL(u)]\, dR = \oint_B M_n(u,v)\, dB \tag{1.11}$$

where $\bar{L}$ is the adjoint differential operator corresponding to the operator L. Also, $M_n(u,v)=uD_n(v)-vD_n(u)$, where the operator D_n is obtained after integrating the left-hand-side of Eq.(1.11) by parts.

Define v=K(p,q) such that $\bar{L}$ (K(p,q)) = -δ(p-q) then

$$\int_R u(q)L(K(p-q))\, dR_q = u(p)$$

For such a choice of v, Eq.(1.11) becomes[3]

$$c(p)u(p)= \oint_B [K(p,q)D_n(u(q))-u(q)D_n(K(p,q))]\, dB(q)+ \int_R b(q)\, K(p,q)\, dR(q) \tag{1.12}$$

The coefficient c(p) can be obtained by setting b(q)=0 and n(q)=1 in Eq.(1.12).

In elastic problems, the equations of equilibrium is

$$\frac{\partial \sigma_{ij}}{\partial xj} = -\, b_i \tag{1.13}$$

where σ_{ij} are the components for the stress tensor and b_i are the components of the body force. According to Hook's law for isotropic materials,

$$\sigma_{ij}=\lambda\, u_{k,k}\,\delta_{ij} + \mu\,[u_{i,j}+u_{j,i}] \tag{1.14}$$

where u_i are the displacements in the material. Combining Eqns.(1.13) and (1.14) one obtains

$$L(u_i) = (\lambda+\mu)u_{k,ki}+\mu\, u_{i,kk} = -\, b_i \tag{1.15}$$

The traction vector t_i is defined as follows

$$t_i=\sigma_{ij}\, n_j=\lambda\, u_{k,k}\, n_j + \mu\,[u_{i,j}+u_{j,i}]\, n_j,$$

where n_j are the components of the normal drawn on the boundary of the domain. It can be shown that the operator D_n converts displacements into tractions

$$D_n(u)=\lambda\, u_{k,k}\, n_j + \mu\,[u_{i,j}+u_{j,i}]\, n_j \tag{1.16}$$

The kernel function K(p,q) for this problem is the fundamental solution U_{ij} for displacement. Thus, for the self-adjust operator L, one can write

$$LU_{ij}(p,q) = \bar{L}\, U_{ij}(p,q) = -\,\delta(\bar{p}\, -\bar{q})\;\delta_{ij} \tag{1.17}$$

Recalling that D_n operator transforms displacements into traction, the traction fundamental solution is defined as

$$T_{ij}= D_n\, U_{ij} \tag{1.18}$$

With such definition for the kernel function, for this elastostatics problem, Eq.(1.12) becomes[3]

$$c_{ij}(p)u_i(p)= \oint_B [t_i(q)\, U_{ij}(p,q) - u_i(q)\, T_{ij}(p,q)]\, dB(q) + \int_R b_i(q)\, U_{ij}(p,q)\, dR(q) \qquad (1.19)$$

As in the case of the Laplace problem, the constant $c_{ij}(p)$ takes different values depending on whether p is located in the interior of the domain, or on a smooth boundary, or at a corner. The $c_{ij}(p)$ can be determined by taking $u_i(q)=1$ and $b_i(q)=0$.

The surface integral involving $b_i(q)$, in Eq.(9), can also be transformed into a contour integral over boundary B by defining a scalar potential Ψ for the body force as follows $b_i = -\Psi_{,i}$. The Eq.(1.19) can be solved by using the method of collocation, described in the following chapter.

The fundamental solution U_{ij} and T_{ij} can be obtained as

$$U_{ij}= -\frac{1}{8\pi(1-\nu)G}[(3-4\nu)\delta_{ij} \ln r - \frac{y_i y_j}{r^2}] \qquad (1.20)$$

$$T_{ij}= -\frac{1}{4\pi(1-\nu)r^2}[(1-2\nu)(y_i n_j - y_j n_i) + \{(1-2\nu)\delta_{ij} + \frac{2y_i y_j}{r^2}\} y_j n_j] \qquad (1.21)$$

where

$y_1 = x\text{-}x_0$

$y_2 = y\text{-}y_0$

$r = | p - q |$

and G and ν are shear modules and Poisson's ratio. For plane stress problems, one has to replace ν by $\nu/(1+\nu)$.

After all the unknown boundary values are determined and both $t_i(q)$ and $u_i(q)$ are known at all points on the boundary, Eq.(1.19) can then be utilized to obtain the displacement, anywhere in the interior. By combining the appropriate derivatives of Eq.(1.19) one can obtain the interior strain, and by using the stress-strain relationships the interior stresses can also be determined.

CHAPTER TWO. NUMERICAL IMPLEMENTTATION FOR BEM

Introduction

The discretization of the boundary integral equations is a three-step process.

1. The boundary is broken up into elements.
2. The variables, such as displacement and traction, are expressed in terms of nodal values and polynomial shape functions.
3. The product of the shape function and the kernel functions are integrated over each boundary element.

The boundary can be partitioned into straight or curved elements. Curved elements represent the boundary of the domain more accurately. However, the integration over these curved elements can only be done numerically.

When the kernel functions are near singular, as in the case in BEM, the numerical quadrature has to be performed with extreme care. The error in quadrature affects the accuracy in the final solution. Furthermore, the accuracy in the quadrature cannot be assessed very well.

In view of this, in the present work, straight elements are used. A good representation of curved boundaries may need a grid optimization. The optimization technique is discussed in a latter chapter.

Elements and integration

The curved boundary is broken up into straight elements as shown in Figure 2.1. On each element the variation of any quantity is assumed to be linear. Thus, all variables are expressed as a linear combination of two linear interpolating functions and two nodal values.

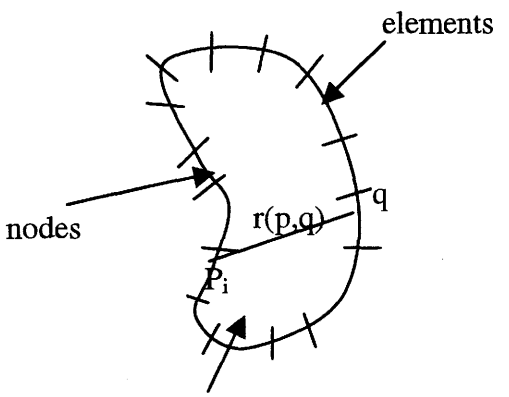

Figure 2.1: Elements and nodes on boundary

Each element contains two nodes at locations $\xi_0=0$ and $\xi_2=1$, where ξ is a local coordinate with its origin at one end of the element. In general, variable U is expressed as

$$U = C_1(1-\xi) + C_2(\xi) \tag{2.1}$$

where C_1 and C_2 are the nodal values of U.

The product of the expression of Eq.(2.1) and the kernels of Eqns.(1.20) and (1.21) leads to integrals of the three following forms.

$$\int_0^1 \xi^m \ln r d\xi$$

$$\int_0^1 \xi^m \frac{(x-x_0)^\alpha (y-y_0)^\beta}{r^2} d\xi$$

$$\int_0^1 \xi^m \frac{(x-x_0)^\alpha (y-y_0)^\beta}{r^4} d\xi$$

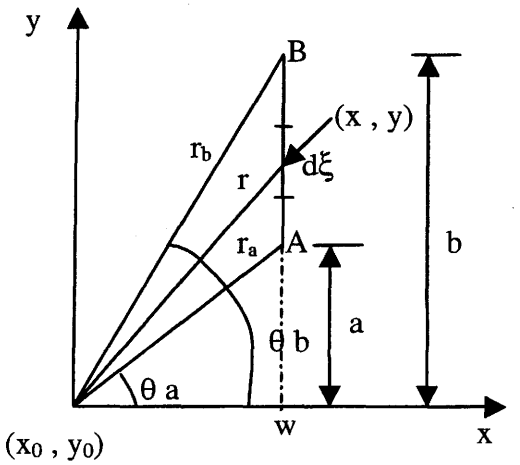

Figure 2.2: Polynomial integration

Definitions of all the variables are shown in Figure 2.2. As ξ = y-a, then

$$\xi^{m} = (\text{y-a})^{m} = \sum_{s=0}^{m} (-1)^{m+s} C_s^m u^{m-s} y^s \quad \text{and} \quad C_s^m = \frac{m!}{s!(s-m)!}$$

The above integrals can then be written as liner combinations of the following integrals.

$$\int_a^b y^s \ln r dy$$

$$\int_a^b \frac{x^t y^s}{r^2} dy$$

$$\int_a^b \frac{x^t y^s}{r^4} dy$$

Also, y=w tanθ, r=w secθ, dy=wsec^{2}θdθ then

$$\int_a^b y^s \ln r dx_2 = \mathrm{w}^{s+1} \int_{\theta a}^{\theta b} \tan^s \theta \ln(w \sec\theta) \sec^2 \theta \; \mathrm{d}\theta$$

$$= \mathrm{w}^{s+1} \left[\frac{\tan^{s+1}\theta}{s+1} \ln(w \sec\theta) \right]_{\theta a}^{\theta b} - \frac{w^{s+1}}{s+1} \int_{\theta a}^{\theta b} \tan^{s+2} \theta \; \mathrm{d}\theta \qquad (2.2)$$

Let $\Theta_s = \int_{\theta a}^{\theta b} \tan^s \theta d\theta$, then $\Theta_{s+2} = \int_{\theta a}^{\theta b} \tan^{s+2} \theta \; \mathrm{d}\theta$. Thus,

$$\int_a^b y^s \ln r dx_2 = \frac{1}{s+1}[b^{s+1}\ln r_b - a^{s+1}\ln r_a] - \frac{w^{s+1}}{s+1}\Theta_{s+2} \tag{2.3}$$

Similarly

$$\int_a^b \frac{x^t y^s}{r^2} dy = w^{t+s-1}\int_{\theta a}^{\theta b} \tan^s\theta \; d\theta = w^{t+s-1}\Theta_s \tag{2.4}$$

Let $\Psi_s = \int_{\theta a}^{\theta b} \frac{\tan^s\theta}{\sec^2\theta}\, d\theta$, then

$$\int_a^b \frac{x^t y^s}{r^4} dy = w^{t+s-3}\int_{\theta a}^{\theta b} \frac{\tan^s\theta}{\sec^2\theta}\, d\theta = w^{t+s-3}\Psi_s \tag{2.5}$$

To evaluate Θ_s, when s is even s=2l, then

$$\Theta_{2l} = \int_{\theta a}^{\theta b} \tan^{2l}\theta d\theta = \int_{\theta a}^{\theta b} (-1)^l [1+\sec^2\theta \sum_{i=1}^{l} (-1)^i \tan^{2i-2}\theta] d\theta$$

$$= (-1)^l[(\theta_b - \theta_a) + \sum_{i=1}^{l} \frac{(-1)^i}{2i-1}\{\tan^{2i-1}\theta_b - \tan^{2i-1}\theta_a\}] \tag{2.6}$$

when s=2l+1, then

$$\Theta_{2l+1} = \int_{\theta a}^{\theta b} \tan^{2l+1}\theta d\theta$$

$$= (-1)^l[(Inr_b - Inr_a) + \sum_{i=1}^{l} \frac{(-1)^i}{2i}\{\tan^{2i}\theta_b - \tan^{2i}\theta_a\}] \tag{2.7}$$

To evaluate Ψ_s, when s is even, s=2l, then

$$\Psi_{2l} = \int_{\theta a}^{\theta b} \frac{\tan^{2l}}{\sec^2 \theta} \, d\theta = \int_{\theta a}^{\theta b} (-1)^l [\cos^2 \theta + \sum_{i=1}^{l} (-1)^i \tan^{2i-2} \theta] d\theta$$

$$= (-1)^l [\frac{1}{4}(\sin 2\theta_b - \sin 2\theta_a) + \frac{1}{2}(\theta_b - \theta_a) + \sum_{i=1}^{l} (-1)^i \Theta_{2i-2}] \tag{2.8}$$

when s=2l+1, then

$$\Psi_{2l+1} = \int_{\theta a}^{\theta b} \frac{\tan^{2l+1} \theta}{\sec^2 \theta} \, d\theta = \int_{\theta a}^{\theta b} (-1)^l [\sin\theta \cos\theta + \sum_{i=1}^{l} (-1)^i \tan^{2i-2} \theta] d\theta$$

$$= (-1)^l [\frac{1}{4}(\sin 2\theta_b - \sin 2\theta_a) + \frac{1}{2}(\theta_b - \theta_a) + \sum_{i=1}^{l} (-1)^i \Theta_{2i-1}] \tag{2.9}$$

By utilizing Eqns. (2.3) through (2.9), all the boundary integrals in Eq.(1.19) can be exactly evaluated.

After collocating the integral Eq.(1.19) at all the boundary nodes, one obtains

$$A\,u = B\,t \tag{2.10}$$

where the u and t vectors contains the nodal values of displacement and tractions, respectively. After inserting the boundary conditions and re-arranging, one obtains

$$F\,x = b \tag{2.11}$$

where the vector x contains the unknown displacements and tractions.

Special boundary conditions and multiple nodes

While inserting the boundary conditions, special care has to be taken at points of discontinuities. The discontinuity can occur due to two reasons – at corner nodes where the normal to the boundary abruptly changes direction, and on smooth boundary where the boundary condition changes type. The change in boundary condition can again be of two kinds—displacement boundary condition changing to traction boundary condition or where the traction itself has a jump discontinuity. The discontinuous boundary conditions are schematically shown in Figure 2.3[4].

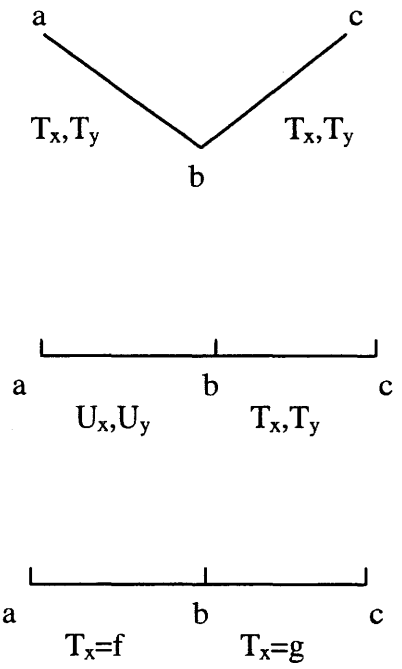

Figure 2.3: Examples of corner nodes and discontinuous boundary condition

The discontinuity in the boundary condition causes a shortage of equations in the final BIE systems. In other words, there are more than one unknown at that node, but only one equation is available.

The scheme to solve this problem is to generate additional equations. The additional equations can be derived from other laws, theorem, differentiations and finite differencing. There is also the method of adding collocation point outside the region, but the condition number for the coefficient matrix is always very large and this affects the accuracy. The method used here derives the extra equations from within the framework of the collocation. We use double functional nodes at the same geometric location that is at the junction of the two elements where the discontinuity is present. We denote the two elements meeting at the discontinuity by '+' and '-'. The '-' element is right before the discontinuity and the '+' node is right after the discontinuity. Among the double functional nodes, one belongs to the '-' element and the other belongs to the '+' element. The collocation scheme employed at the '-' and the '+' nodes depends on the nature of the boundary condition specified on the '-' and '+' elements. Four variables, two displacements and two tractions, are associated with each node. Thus, eight variables are associated with the double functional node. Out of these eight variables, four variables are prescribed as boundary conditions. Thus, one needs four equations to obtain the four unknowns at the double functional node. The scheme for obtaining these four equations is shown in Table 2.1[4].

Table 2.1: Collocation scheme for elastostatics problem

		BC on +					
		T_x,T_y	T_x,U_x	T_x,U_y	T_y,U_x	T_y,U_y	U_x,U_y
BC On –	T_x,T_y	Fc-(x,y) F±(x) F±(y)	Fc-(x,y) F+(x) F±(y)	Fc-(x,y) F±(x) F+(y)	Fc-(x,y) F+(x) F±(y)	Fc-(x,y) F±(x) F+(y)	Fc-(x,y) F+(x) F+(y)
	T_x,U_x	Fc-(x,y) F-(x) F±(y)	Fc-(x,y) Fc+(x) F±(y)	Fc-(x,y) F-(x) F+(y)	Fc-(x,y) Fc+(x) F±(y)	Fc-(x,y) F-(x) F+(y)	Fc-(x,y) Fc+(x) F±(y)
	T_x,U_y	Fc-(x,y) F±(x) F-(y)	Fc-(x,y) F+(x) F-(y)	Fc-(x,y) F±(x) Fc+(y)	Fc-(x,y) F+(x) F-(y)	Fc-(x,y) F±(x) Fc+(y)	Fc-(x,y) F+(x) Fc+(y)
	T_y,U_x	Fc-(x,y) F-(x) F±(y)	Fc-(x,y) Fc+(x) F±(y)	Fc-(x,y) F-(x) F+(y)	Fc-(x,y) Fc+(x) F±(y)	Fc-(x,y) F-(x) F+(y)	Fc-(x,y) Fc+(x) F±(y)
	T_y,U_y	Fc-(x,y) F±(x) F-(y)	Fc-(x,y) F+(x) F-(y)	Fc-(x,y) F±(x) Fc+(y)	Fc-(x,y) F+(x) F-(y)	Fc-(x,y) F±(x) Fc+(y)	Fc-(x,y) F+(x) Fc+(y)
	U_x,U_y	Fc-(x,y) F-(x) F-(y)	Fc-(x,y) Fc+(x) F-(y)	Fc-(x,y) F-(x), Fc+(y)	Fc-(x,y) Fc+(x) F-(y)	Fc-(x,y) F-(x) Fc+(y)	Fc-(x,y) Fc+(x,y)

Notes:
Fc+(i,j): Collocation at an off-functional node on + (the next element).
Fc- (i,j): Collocation at the functional node on – (the previous element).
F- (i): Use $U^+=U^-$, where U^- is known.
F+ (i): Use $U^-=U^+$, where U^+ is known.
F± (i): Use $U^- - U^+ =0$ where U^- and U^+ both are unknown.

CHAPTER THREE. GRID OPTIMIZATION FOR BEM

Introduction

The purpose of grid optimization is to obtain higher accuracy at a low cost. It is also desirable to make the optimization process automatic so that user inputs are minimized.

Initial data preparation for input is a time consuming part. Thus, it is necessary that the BEM be kicked–off on a very crude grid. The program then automatically improves the grid. Smaller boundary elements naturally yield better accuracy. However, using small elements everywhere will result in a large system of equations. The execution time to solve N number of equations is proportional to N^3. Thus, it is necessary that smaller elements be used only at the locations they are essential. In this chapter, a spline-based grid optimization scheme is described that achieves these goals.

The purpose of optimization is two-fold. Firstly, the straight elements must describe the curved boundary appropriately. Secondly, the interpolating function for displacement and traction must mimic the actual variation of these quantities along the boundary. The spline-based grid optimization technique operates on the geometry and the functional parts simultaneously.

Cubic spline curves

The cubic spline curves represent the geometry (the Cartesian coordinates of the nodes) and the boundary functions (displacements and tractions). Altogether, six spline functions are generated for six relevant variables – two for the Cartesian coordinates, two for the two components of displacements on the boundary, and two for the two components of tractions on the boundary. Denoting a variable by S, the spline for S is written in a parametric as S= S(t) , where t is the arc length along the boundary. For a cubic spline, S(t) has the form

$$S(t)=at^3+bt^2+ct+d \tag{3.1}$$

Since the position of the geometric nodes and also the nodal values of displacement and traction for the BEM input are all in finite domain, the parameter t can be scaled into [0,1] interval. Also, all the six variables are scaled into [-1,1] interval.

Consider that the nodal values of x, y, and the displacement and tractions are known. These nodes are the control or the definition points in the spline and say that there are N_d definition points. Next, a boundary fitted arc length variable t is defined.[5] The direction of travel on the boundary while defining the arc length t is clockwise for inner boundaries and counter-clockwise for outer boundaries.

The value of t at the definition point is calculated from

$$t_{i+1}=t_i+\{(x_{i+1}-x_i)^2+(y_{i+1}-y_i)^2\}^2,\ \text{i=1,2,...}N_d\text{-1} \tag{3.2}$$

where t_1=0 and (x_i,y_i) is the position for the ith control point. The spline in the N_d number of segments between the N_d number of definition points is symbolically given by $S_i(t)$. The whole spline is then an assembly of individual splines S_i defined for each interval $\{t_i \le t \le t_{i+1}, i = 1,2,..., N_d - 1\}$.

The Eq.(3.1) can now be written in an alternative forma as

$$S_i(t) = F_i + S'(t_i)(t - t_i) + C_2(t - t_i)^2 + C_3(t - t_i)^3 \tag{3.3}$$

where it can be seen that $S_i(t_i) = F_i$, i = 1,2,....., N_d. As the nodal values of S are known, the F_i's are also known. Thus, in this spline expression, there are three quantities, namely, $S_i(t_i)$, C_2, and C_3 that are to be determined. Figure 3.1 shows what is such kind of spline curve looks like.

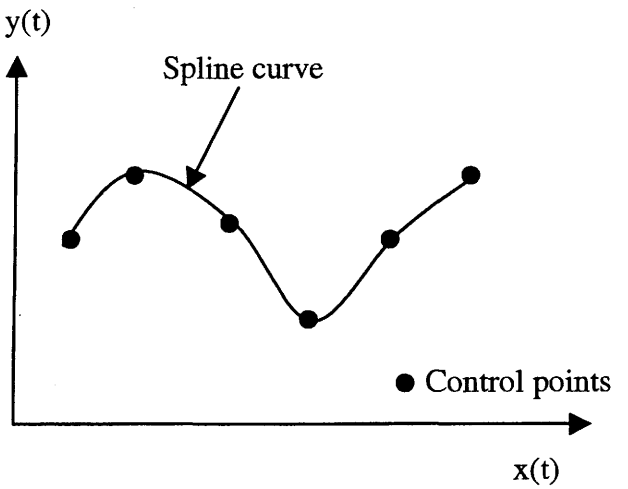

Figure 3.1: Spline curve

These quantities are obtained by demanding that the spline function maintains the continuity in C^0, C^1, and C^2 (C^0, C^1, and C^2 means continuity of 0^{th},1^{st} and 2^{nd} derivative) at all the nodes. Recall that the two splines are joined at each node and what we are demanding is that this joint is smooth up to the second order derivative of the spline. The conditions used for the determination of the three quantities are

$$S_i(t_{i+1}) = F_{i+1} \tag{3.4}$$

$$S_i'(t_{i+1}) = S'(t_{i+1}) \tag{3.5}$$

$$S_i''(t_{i+1}) = S_{i+1}''(t_{i+1}) \tag{3.6}$$

Using the first two conditions, the constants C_2 and C_3 are obtained as[5]

$$C_2 = \frac{3(F_{i+1} - F_i)}{(t_{i+1} - t_i)^2} - \frac{2S'(t_i) + S'(t_{i+1})}{t_{i+1} - t_i} \tag{3.7}$$

$$C_3 = \frac{2(F_i - F_{i+1})}{(t_{i+1} - t_i)^3} - \frac{S'(t_i) + S'(t_{i+1})}{(t_{i+1} - t_i)^2} \tag{3.8}$$

To calculate the unknown $S'(t_i)$, the third condition I written as follows.

$$\frac{S'(t_{i-1}) + 2S'(t_i)}{t_i - t_{i-1}} + \frac{2S'(t_i) + S'(t_{i+1})}{t_{i+1} - t_i} = \frac{3(F_{i+1} - F_i)}{(t_{i+1} - t_i)^2} + \frac{3(F_i - F_{i-1})}{(t_i - t_{i-1})^2} \tag{3.9}$$

One may note that the Eq.(3.9) will yield a set of simultaneous equations that are solved to obtain $S'(t_i)$ in all the segments.

The condition of continuity for the second derivative cannot be applied at the two ends of the region over which the spline is being fitted, namely at the nodes numbered 1 and numbered N_d. Thus we fall short by two equations. The extra conditions are obtained as follows.

For the close boundary that the two ends joined together and joined smoothly, the continuity for the two ends themselves can be used. Thus,

$$S_1''(t_1) = S_{Nd-1}''(t_{Nd})$$

$$S'(t_1) = S'(t_{Nd})$$

For open curves and curves that are not joined smoothly, third derivative of the two end points can be used, and the continuity of the third derivative can give the two extra equations as follows.

$$S_1'''(t_2) = S_2'''(t_2)$$

$$S_{Nd-1}'''(t_{Nd-1}) = S_{Nd}'''(t_{Nd-1})$$

Now the system of equations can be solved and the spline curve can be generated. In the two-dimensional case, the nodal coordinates x and y are given the parametric form x(t) and y(t) and can be approximated by the corresponding splines of x and y. In the elastostatics problem, the functions U_x, U_y, T_x, T_y are also described by similar splines.

Adaptive grid generation

When the all the spline curves are generated, (two for the geometry x and y, four for the functions U_x, U_y, T_x and T_y), the next question is whether these spline curves are close enough to the original linear element approximation. In other words, we now have two representations for nodal coordinates, nodal displacements, and nodal tractions. One is the spline representation of these variables. The other is the linear geometric discretization and linear shape functions for displacements and tractions. For improved accuracy the difference between these two representations has to be reduced to desirable level. This reduction of the difference is the basis for the adaptive grid generation.

To obtain the difference between the spline and the piecewise linear approximation, we consider only the quadratic part of the spline within the interval [t_i t_{i+1}].

$$S(t) = S(t_i) + S'(t_i)(t - t_i) + \frac{1}{2} S''(t_i)(t - t_i)^2 \tag{3.10}$$

For the linear approximation, the equation of the line in the same interval is

$$L(t) = S(t_i) + \frac{t - t_i}{t_{i+1} - t_i}[S(t_{i+1}) - S(t_i)] \tag{3.11}$$

Then the difference between the linear approximation and the spline curve is maximum at the midpoint. This maximum value can be obtained as[5]

$$\varepsilon_{\max} = \frac{1}{8}(t_{i+1} - t_i)^2 S''(t_i) \tag{3.12}$$

The tolerance of the approximation is the maximum separation allowed between the linear and spline representations. The condition for the optimization is that the maximum difference given in Eq.(3.12) is less than a tolerance specified by the user. With the given tolerance ε_{max}, the length of an element $t_{i+1} - t_i = h$ can be obtained from Eq.(3.12) as

$$t_{i+1} - t_i = h = \sqrt{\frac{8\varepsilon_{max}}{S''(t_i)}} \tag{3.13}$$

With the new step length h, the original interval can be broken into the new intervals with the new lengths. This process starts at one end of the interval. The second derivative and step h are calculated. This length is added to the first nodal position to get the other end point of this interval. A marching process continues until the end of the spline curve is reached. After that, the error between the new linear approximation and spline curve will satisfy the specified tolerance.

In the present elastostatics problem, there are six spline curves at one point – two for the Cartesian coordinates, two for the displacements, and two for the tractions. Then six step sizes are obtained for the six variables by utilizing Eq.(3.13). The program then picks the smallest of these element lengths. The final expression of h could be written as[5]

$$h = \sqrt{\frac{8\varepsilon}{\max[x''(t_i), y''(t_i), U_x''(t_i), U_y''(t_i), T_x''(t_i), T_y''(t_i)]}} \tag{3.14}$$

The above grid optimization scheme works for most of the problem, here are two special cautions for this algorithm. A very small tolerance number always means a good

approximation, but if the number is too small, lots of new nodes will be generated. Sometimes, the spline curve with too many nodes changes the curve into a totally different spline which is shown is Figure 3.2.

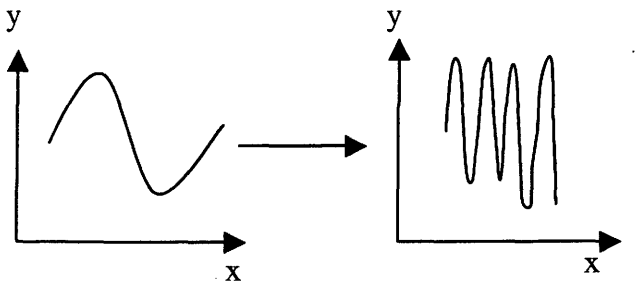

Figure 3.2: Tiny tolerance number problem

Another problem is the peak crossover which is shown in Figure 3.3. The step length is calculated at the starting point of the interval, and it is controlled by the second derivative at that point. When the curves inside this interval have peaks (first derivative crosses zero), and if the new step length is too long, the new ending point of the generated spline interval will crossover the peak. This will lose some significant characters of the curve, and sometimes makes the curve collapse into very simple spline.

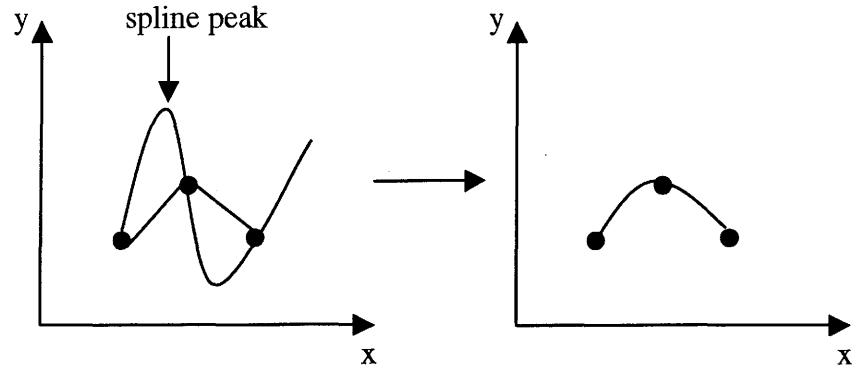

Figure 3.3: Cross peak problem

CAPTER FOUR. CONTROLS FOR THE OPTIMIZATION

Introduction

In this chapter, we discuss further details of the implementation of the grid optimization. The optimization is performed over several loops to generate new grids and it is essential to check accuracy at the end of each loop. There are some controls designed to evaluate the efficiency of the optimization, and to make the decision to terminate the loop. Those controls include the tolerance number mentioned in the previous chapter, the condition number of the resulting matrix, the global equilibrium equations, and the area difference between the spline representation and the straight element approximation.

Optimization loop

The BEM input grid can be very crude that requires very little input labor. Then this original input goes into the BEM solver. From the first solution with the original input, the spline curves are fitted and the accuracy of the initial solution is gauged. If it does not satisfy the tolerance requirement, the optimization scheme is employed. After the optimization, the result is checked again with additional control parameters. This check decides whether the

optimization has to be carried to the next loop. Then the new BEM grid is used to recalculate the solution. The accuracy of the new solution is tested and the loop continues until termination.

The controls in the loop decide its termination. There are two possibilities. One is the result is good enough, and the loop stops to finish the optimization successfully. Another one is the conclusion that the results have become divergent, and the loop will stop with no good optimization. So the control for the loop becomes very critical for the optimization process.

Controls in the optimization loop

There are several kinds of controls in the loops. They will cover different areas of the BEM implementation. The first control is the tolerance number mentioned in the previous chapter. The tolerance number roughly determines the closeness between the linear approximation and the spline representation. Figure 4.1 shows the tolerance number. This tolerance number can be easily implemented in the optimization program and this evaluation is very helpful to decide the loop process. But as mentioned before, a tiny tolerance number may lead to unrealistic splines and a large tolerance number may lead to crossover. Thus, the tolerance number alone cannot guarantee highly accurate optimization.

The next control is on the condition of the coefficient matrix. When the elements get too small, the collocation nodes become too close to each other. In this situation, any two rows may become nearly identical, resulting in a nearly singular coefficient matrix. For

nearly singular matrix, the condition number of the matrix becomes too large. For such matrices, the linear solver yields inaccurate solutions.

The body force equilibriums control is to check the accuracy of the result. This control is independent of the BEM solution. The independency of this control makes it a useful one, and it indicates the quality of the result. The calculated traction values along the boundary are summed to check the force and moment equilibrium in the domain.

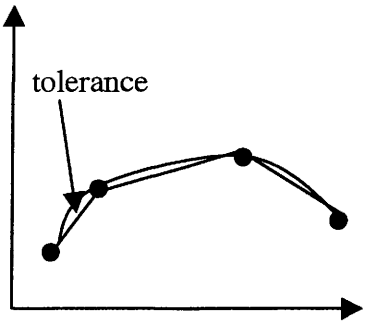

Figure 4.1: Tolerance number control

A more efficient control in the loop is to check the difference in the areas enclosed under the spline representation and the linear approximation. To calculate the area enclosed by the linear approximation, the trapezoidal rule is used. With the known nodal values of the variables, the area which is shown in Figure 4.2 under the linear interpolation is obtained as

$$A_t=1/2\ [F_i+F_{i+1}] \tag{4.1}$$

The area under the spline approximation can be evaluated analytically. Since the constants coefficients in the spline approximation are already generated for the optimization, the analytical evaluation of the integrals is a simple matter.

Defining the area under the spline which is shown in Figure 4.3 as

$$A_s=\int S(t)\ dt \tag{4.2}$$

the difference between A_t and A_s can be found. The relative difference between A_t and A_s shown in Figure 4.4 is used as the third control parameter.

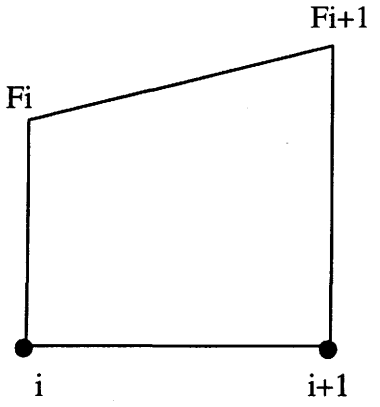

Figure 4.2: Trapezoid integration

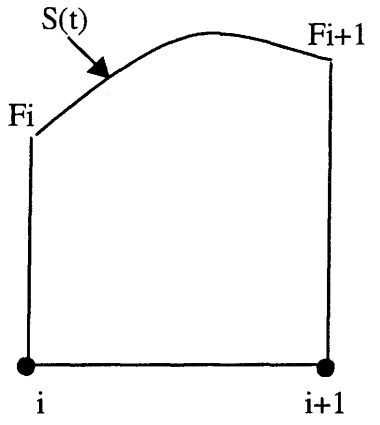

Figure 4.3: Spline curve integration

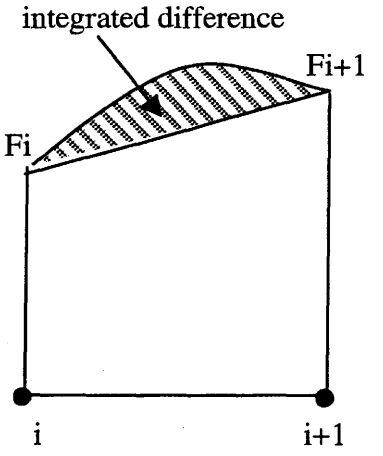

Figure 4.4: Area difference for integration

The ratio of the area difference is a scaled measure of the error as opposed to the absolute error controlled by the tolerance. To implement this ratio scheme, the ratio is calculated before the optimization, and then the optimization program is used. After that, the area difference ratio is recalculated. If the ratio decreases after the optimization, it means that the optimization is progressing in the correct direction, and the loop can continue. If the ratio increases, solution is worse than the result in the previous step, so the loop is terminated.

CHAPTER FIVE. SAMPLE PROBLEMS

Introduction

In this chapter, some examples for the optimization schemes are presented. The examples include the bending beam model, the elastic plate with a model and the clamped elastic plate model. The BEM program is written in FORTRAN 77, the output of the optimization result is written in Matlab 5.0. All the programs are executed on the SGI Indy or SGI O2 machine.

Beam bending model

We first consider a pure bending problem. The schematic for the problem is given in Figure 5.1. The bar is four units long and one unit wide.

The specified boundary conditions are as follows. In addition, the midpoint of the side BC is fixed to eliminate rigid body motion. In Figures 5.2 and 5.3, the beginning grids and the improved grids at the end of each loop are shown for various tolerance values.

On AB: $t_x = t_y = 0$

On BC: $u_x = 0$, $t_y = 0$

On CD: $t_x = t_y = 0$

On AD: $t_y = 0$, $t_x = 0.5 - (l-h)/l$

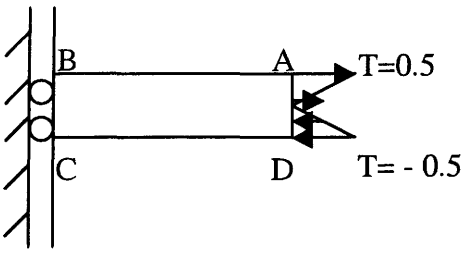

Figure 5.1: Pure bending model

As the optimization process continues, more nodes get clustered on the side BC. Also, on side BC the concentration of the nodes is highest at the two corners and at the mid point where the zero displacement conditions are imposed. On side BC, the grid is controlled by the vertical displacement. The clustering of the grid is a result of the inflexion point that exists in variation of the vertical displacement.

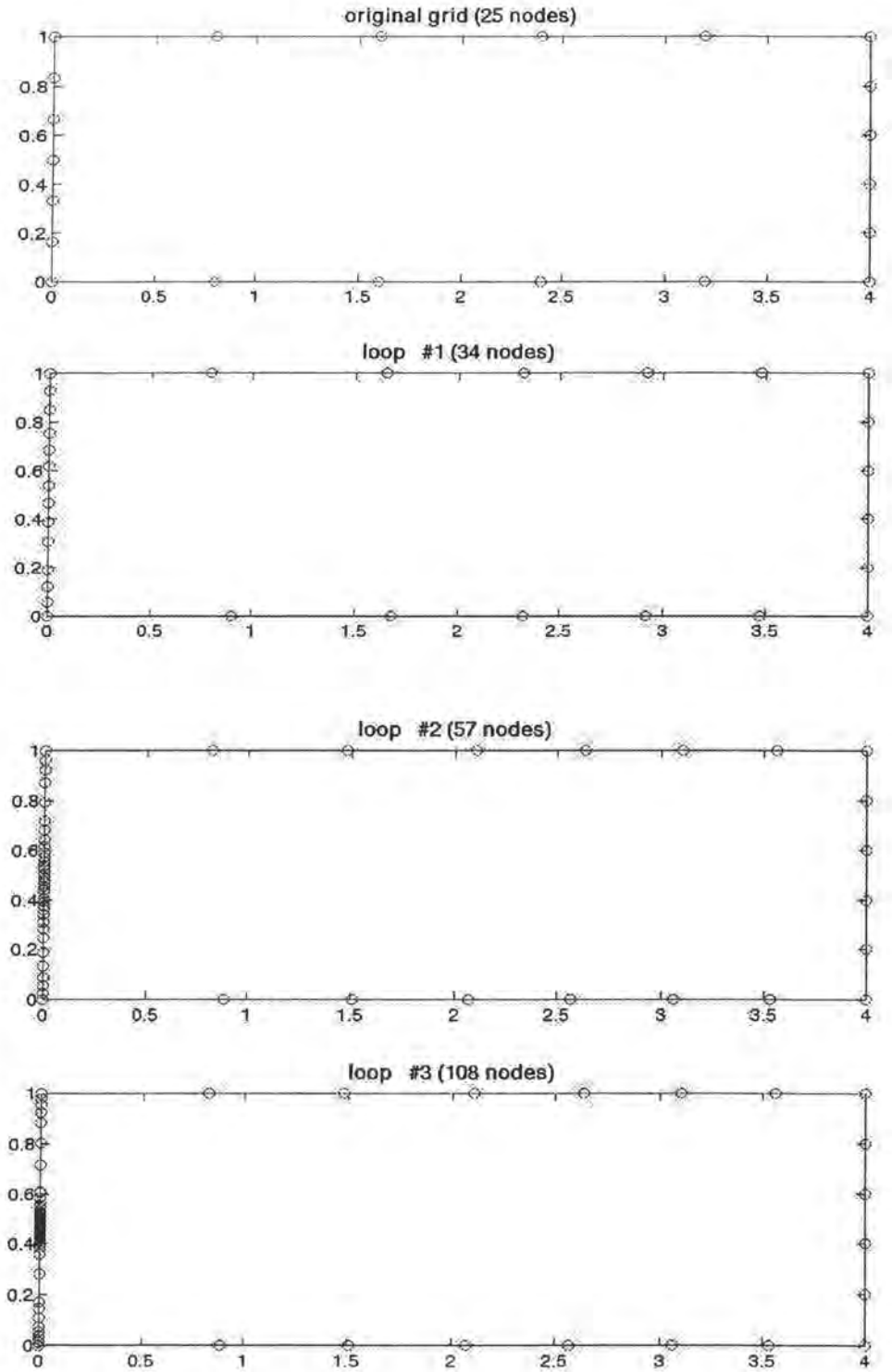

Figure 5.2: Grid optimization in pure bending (tolerance=0.01)

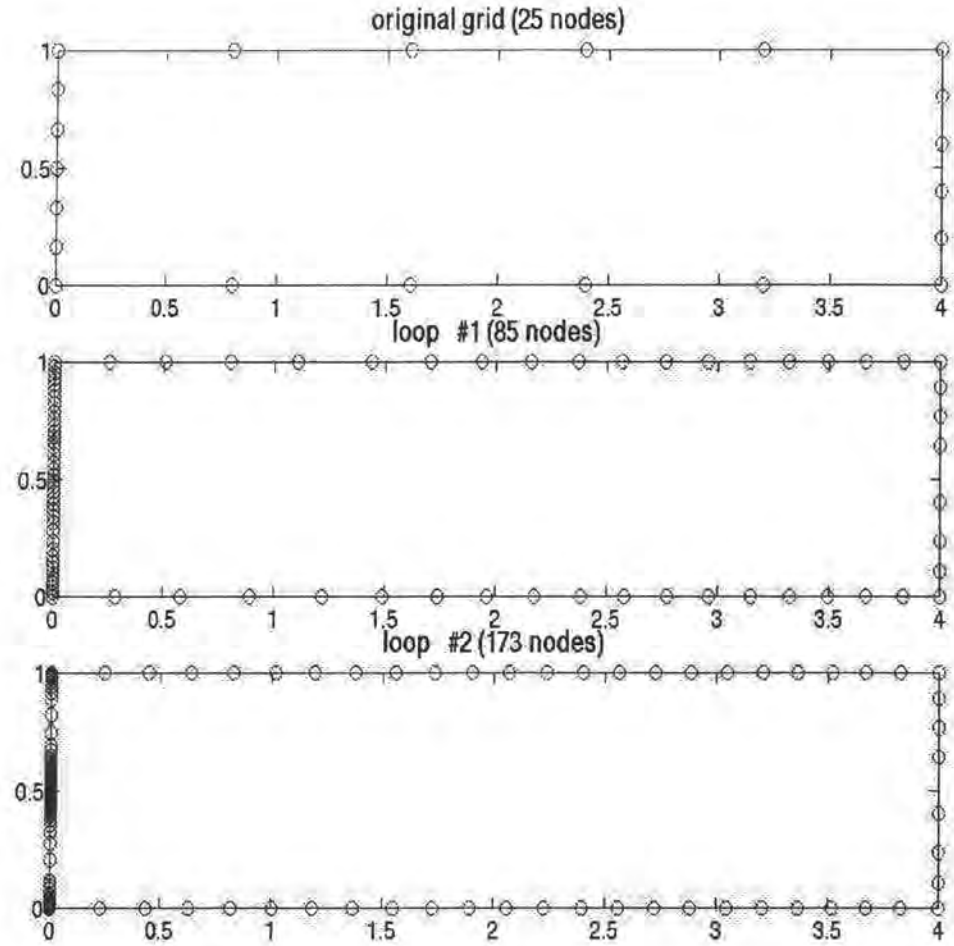

Figure 5.3: Grid optimization in pure bending (tolerance=0.001)

Figures 5.4 and 5.5 show the displacement U_y along the left and top side of the bar. Both of these variations are quadratic, and it turns out to be U_y that controls the size of the elements on these two sides.

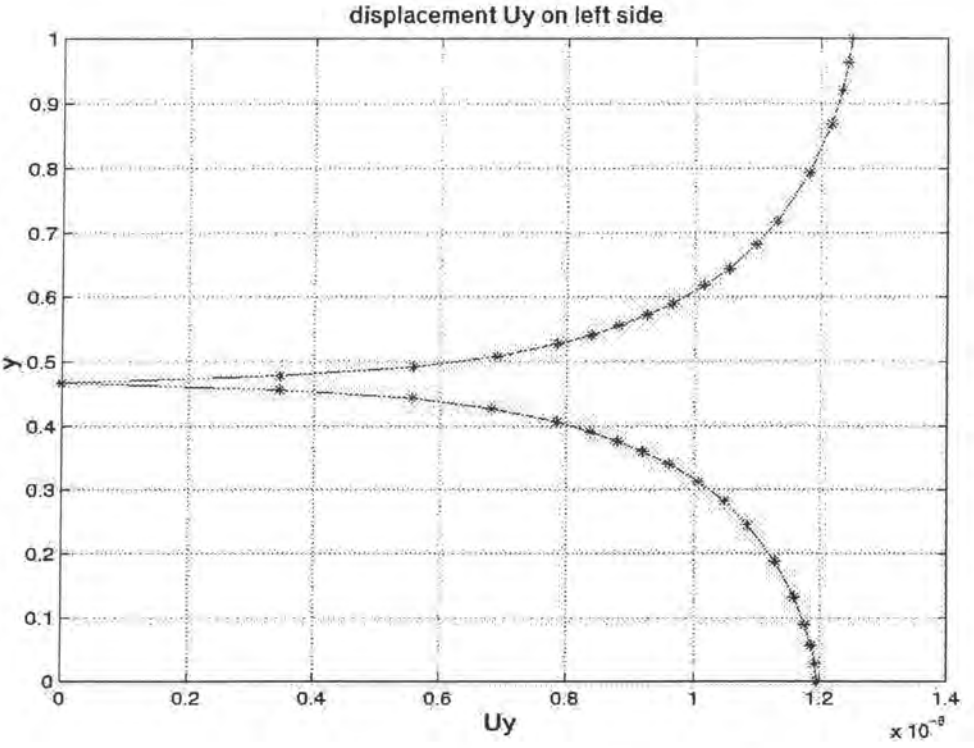

Figure 5.4: Displacement U_y along the left side of the bar (tol=0.01).

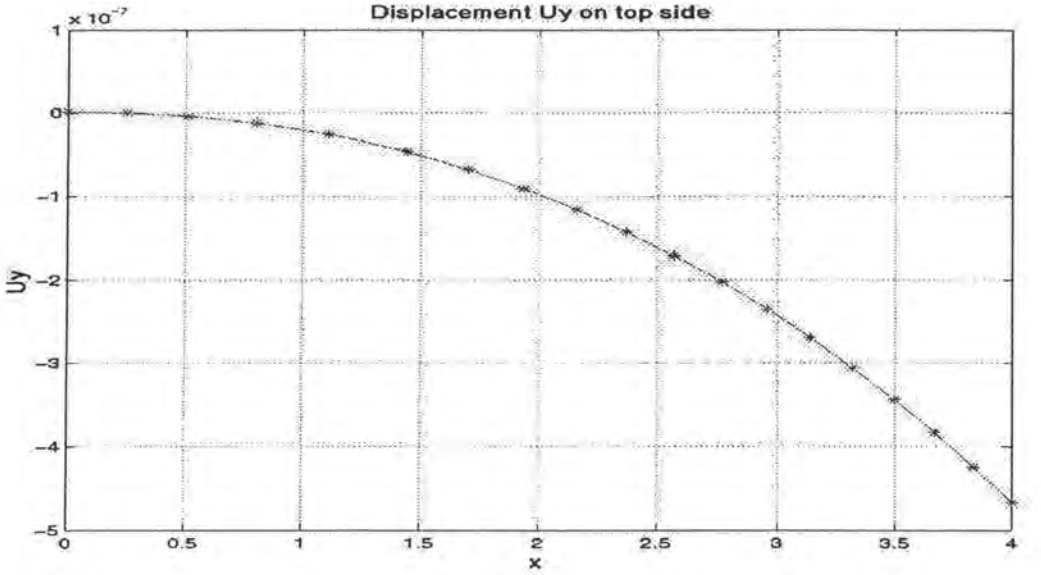

Figure 5.5: Displacement U_y along the top side of the bar model (tol=0.001)

In Tables 5.1 and 5.2, the values of the various controls, discussed in Chapter 4, are presented. The results show that the equilibrium equations are more accurately satisfied as the number of nodes is increased. Also, the difference in the areas under the spline and linear representations reduces as the number of nodes is increased. However, as more nodes are added, the condition of the coefficient matrix in the BEM solver deteriorates.

Table 5.1: Tolerance=0.001 for pure bending

Loop	Nodes	Condition	ΣF_x	ΣF_y	ΣM	Area difference
1	86	2.35E3	-1.37E-4	0	3.63E-2	2.71%
2	175	7.64E4	9.98E-5	0	2.24E-2	1.38%

Table 5.2: Tolerance=0.01 for pure bending

Loop	Nodes	Condition	ΣF_x	ΣF_y	ΣM	Area difference
1	34	8.13E2	7.21E-3	0	9.35E-2	2.477%
2	57	4.26E3	2.82E-3	0	8.36E-2	2.79%
3	109	1.52E5	2.25E-3	0	8.30E-2	2.49%

Plate with a hole

In this example, a plate with a hole is subjected to uniform, unit normal stresses on the two vertical sides, as shown in Figure 5.6.

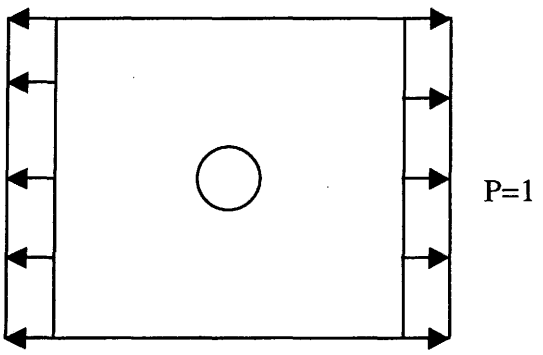

Figure 5.6: Center hole plate model

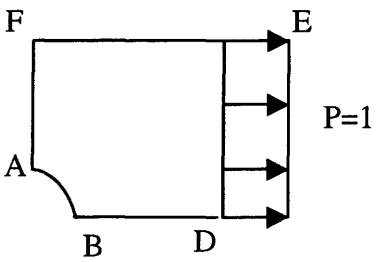

Figure 5.7: Simplified center hole plate

Because of the symmetry, only one quarter of the plate needs to be modeled as shown in Figure 5.7. The arc AB (quarter circler) and sides AF and BD are critical segments in this model. The sides EF and DE are of unit length, and the radius of the hole is 0.1 units. The boundary conditions are specified as follows:

$$\text{On EF: } t_x = t_y = 0$$

$$\text{On FA: } u_x = t_y = 0$$

$$\text{On AB: } t_x = t_y = 0$$

$$\text{On BD: } t_y = u_y = 0$$

$$\text{On DE: } t_y = 0,\ t_x = 1$$

The grids and the end of a number of loops in the adaptive scheme for two tolerance values are shown in Figures 5.8 and 5.9.

The grids in Figures 5.8 and 5.9 show that the straight elements need to be clustered on the surface of the hole to accurately describe the curved boundary. No new nodes are added to the straight edges on the right and top. The reason is that the tractions and displacements are uniform on these two sides and the curvatures of the functions are zero. Hence, very few numbers of straight elements with linear shape functions can correctly describe the variations on these two ides. A large number of nodes are added at two locations where the hole meets the straight edges. The stress near these locations has the most rapid variation.

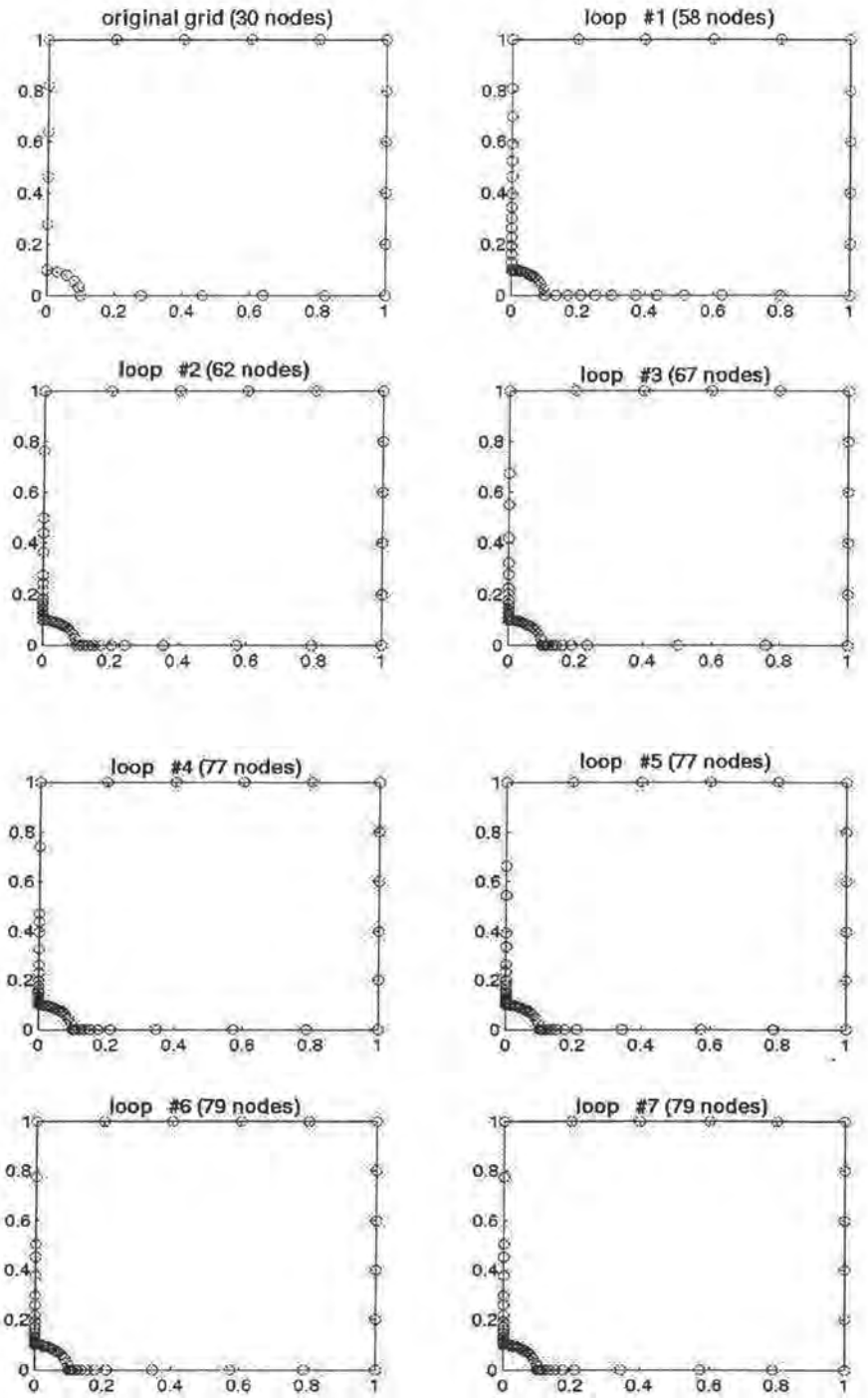

Figure 5.8: Grid optimization for plate with a hole (tolerance=0.01)

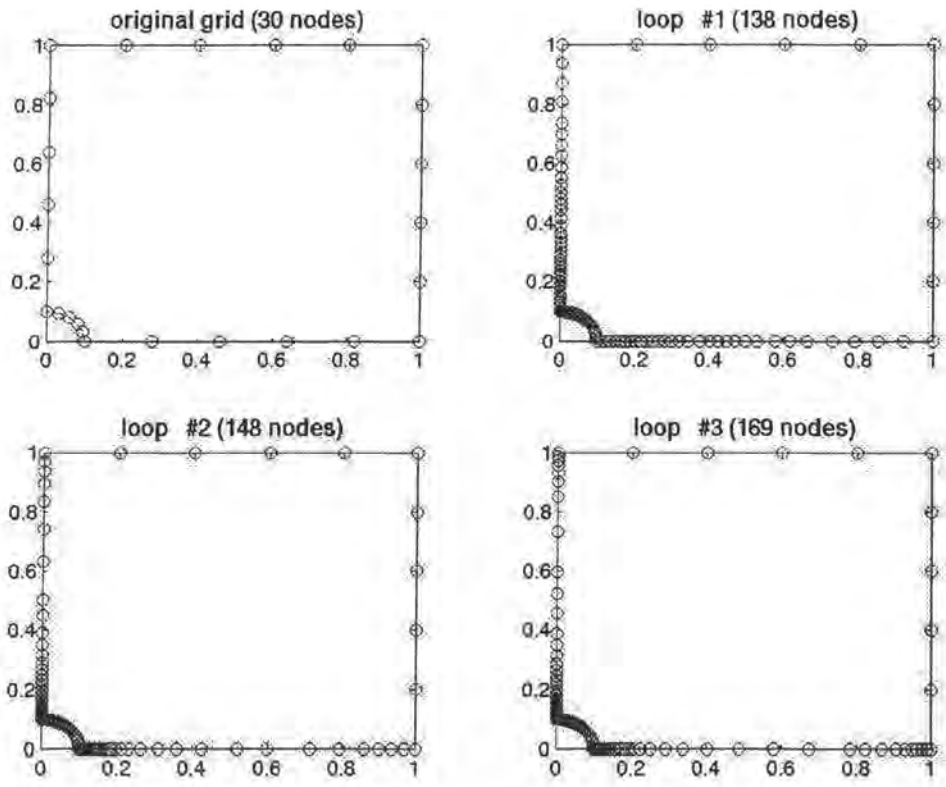

Figure 5.9: Grid optimization for plate with a hole (tolerance=0.001)

In Tables 5.3 and 5.4, the various controls for the adaptive scheme are given for two tolerance values and or each step of the adaptive grid generation scheme. The pattern is satisfactory in the sense that the control values reduced as additional nodes are placed on the boundary. In Figure 5.10, the variation of the horizontal traction along the side AF is shown. The theoretical value of the stress concentration t A for a hole in an infinite plate is 3; for the present finite plate case, a value very close to 3 is obtained.

Table 5.3: Tolerance =0.001 for plate with a hole model

Loop	Nodes	Condition	ΣF_x	ΣF_y	ΣM	Area difference	Max. T_x
1	138	8.57E3	8.49E-5	-4.99E-5	-2.38E-2	2.46%	-3.02
2	148	5.46E4	4.09E-5	-2.02E-5	-2.38E-2	0.104%	-3.04
3	169	3.40E5	3.99E-5	-1.70E-5	-2.38E-2	0.0626%	-3.03
4	169	3.40E5	3.99E-5	-1.70E-5	-2.38E-2	0.0579%	-3.03

Table 5.4: Tolerance =0.01 for plate with a hole model

Loop	Nodes	Condition	ΣF_x	ΣF_y	ΣM	Area difference	Max. T_x
1	58	1.10E3	4.41E-5	-6.83E-5	-2.40E-2	2.497%	-2.91
2	62	6.92E3	-1.32E-4	-5.41E-5	-2.42E-2	0.6033%	-3.09
3	67	2.68E4	-5.32E-4	-4.25E-5	-2.45E-2	0.3095%	-3.11
4	77	3.05E5	-2.36E-4	-4.78E-5	-2.43E-2	0.4237%	-3.13
5	77	3.04E5	-1.93E-4	-5.99E-5	-2.44E-2	0.2922%	-3.15
6	79	3.04E5	-2.45E-4	-5.93E-5	-2.43E-2	0.2963%	-3.14
7	79	3.04E5	-2.45E-4	-5.93E-5	-2.43E-2	0.2918%	-3.14

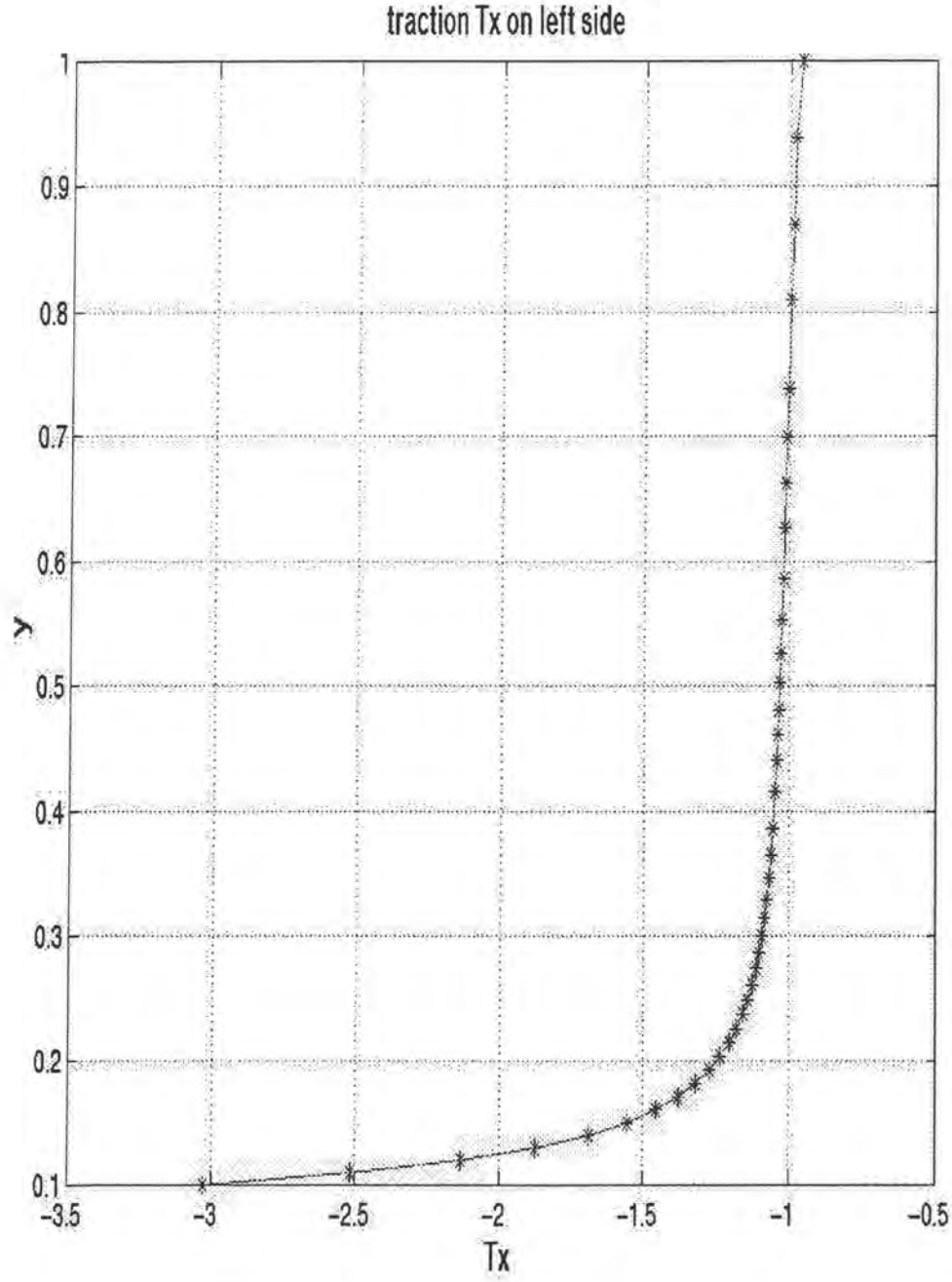

Figure 5.10: Traction T_x along the side AF of the plate with a hole.

Clamped plate model

The clamped plate model has two sides fixed to the wall, and the traction is on the top edge. This problem also includes a discontinuity in traction at point D as shown in Figure 5.11. The plate is a 1-units by 1-unit square and the segment DC is 0.3 units. The following boundary conditions are specified for this problem.

On OA: $u_x = u_y = 0$

On OB: $u_x = u_y = 0$

On AC: $t_x = t_y = 0$

On BD: $t_y = t_y = 0$

On DC: $t_y = 1$, $t_x = 0$

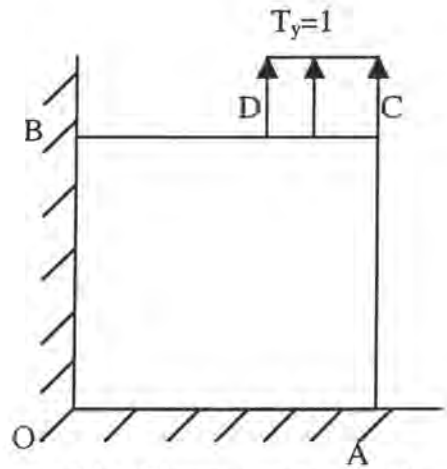

Figure 5.11: Clamped plate model

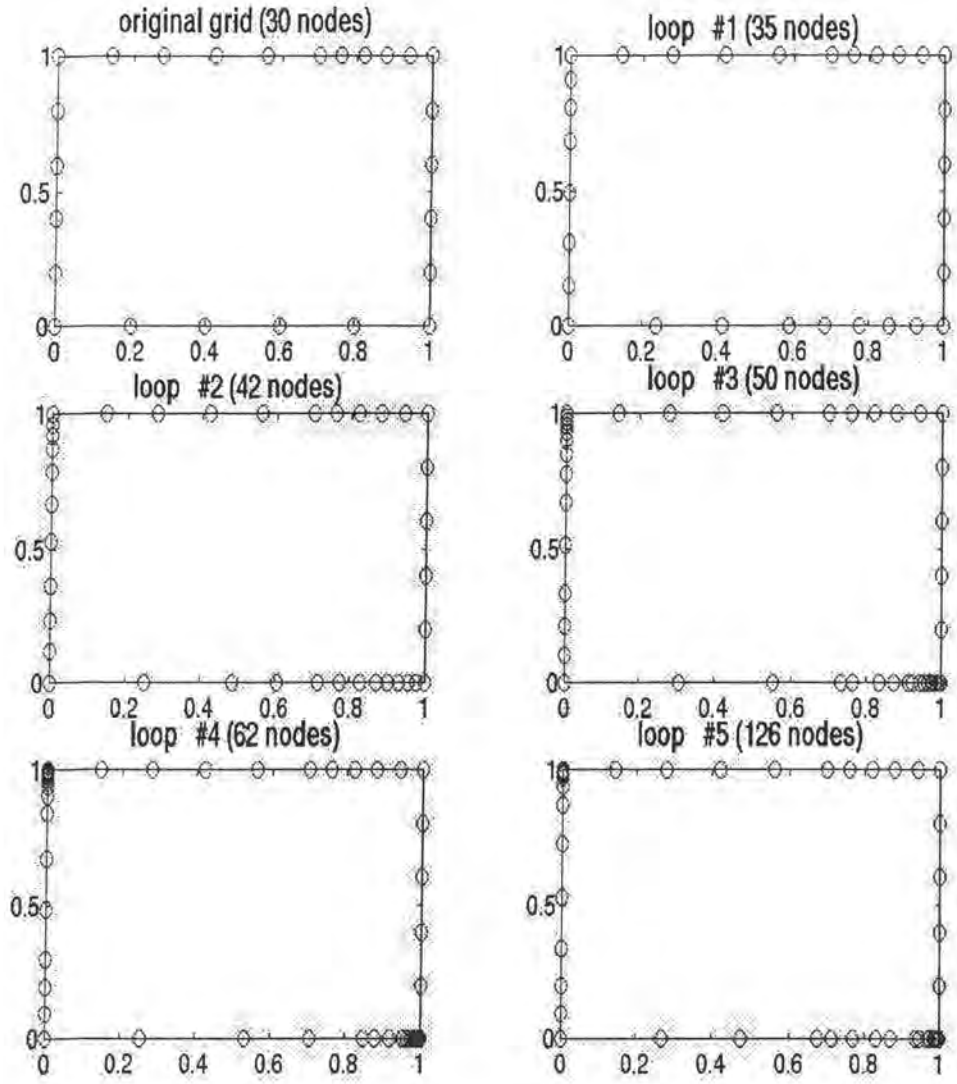

Figure 5.12: Grid optimization in clamped plate (tolerance=0.01)

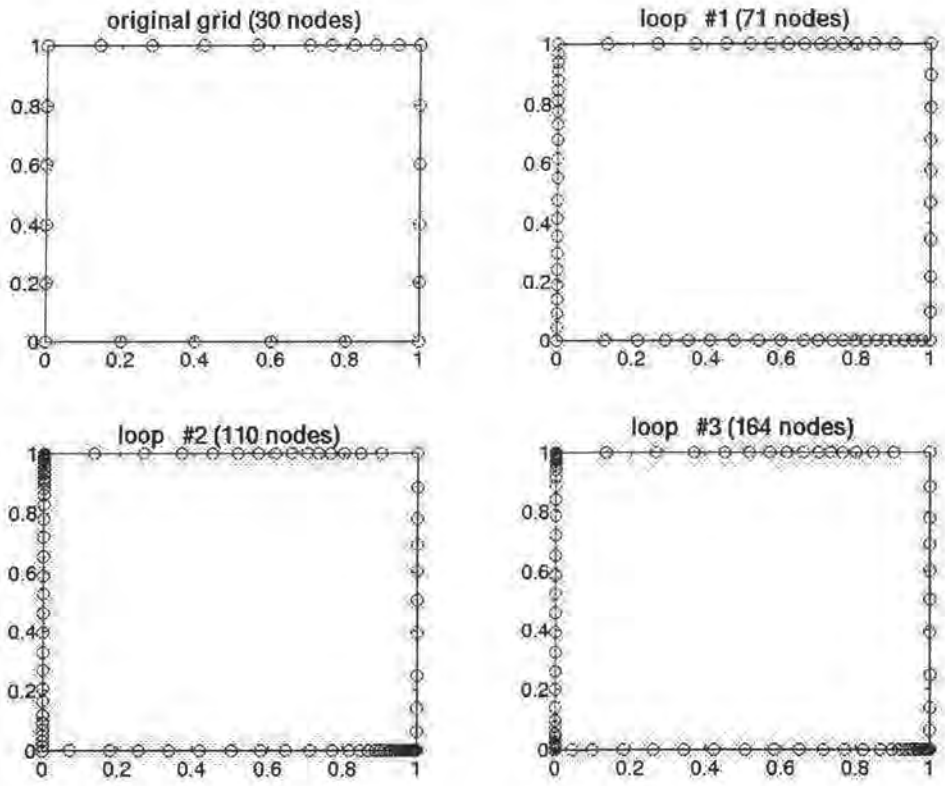

Figure 5.13: Grid optimization in clamped plate (tolerance=0.001)

The distributions of the nodes at various stages of the grid improvement are shown in Figures 5.12 and 5.13 for two values of the tolerance. In this problem, the stresses are singular at points A and B. These are the two near which the clustering of nodes takes place. The nodes are also somewhat clustered near the discontinuity in stress t point D.

In Tables 5.5 and 5.6, the various controls for the adaptive technique are shown. The solution consistently improves as more and more nodes are added. However, the moment

equation was not very well satisfied. The reason being that the numerical solution cannot exactly simulate the singularities at A and B. The numerical computations yield only a large value for the stresses instead of an infinite value. Because of heavy clustering of the nodes, the condition number of the coefficient matrix from BEM increases rapidly showing the further clustering will not enhance the accuracy. We believe that this is a typical situation when the BEM is attempting to simulate the singularity by clustering more and more nodes.

In Figure 5.14, the rapid rise in stresses near the singularities is shown. One very encouraging fact is that the solution does not show any spurious oscillation near the singularity. Such spurious oscillations have been observed in many published work on singular problems.

Table5.5: Tolerance=0.01 for clamped plate

Loop	Nodes	Condition	ΣF_x	ΣF_y	ΣM	Area difference	Max. T_x	Max. T_y
1	35	2.98E2	3.55E-2	4.72E-3	0.11	2.84%	0.52	0.83
2	42	8.53E2	3.53E-2	4.71E-3	0.11	2.05%	0.72	1.05
3	50	5.98E3	3.53E-2	4.99E-3	0.11	1.66%	1.47	1.55
4	62	1.02E5	3.52E-2	4.60E-3	0.11	1.39%	4.01	3.25
5	126	2.61E7	3.54E-2	4.83E-3	0.11	1.64%	26.5	18.4

Table5.6: Tolerance=0.001 for clamped plate

Loop	Nodes	Condition	ΣF_x	ΣF_y	ΣM	Area difference	Max. T_x	Max. T_y
1	71	1.02E3	3.43E-2	6.47E-3	0.11	2.48%	0.65	1.12
2	110	2.98E4	3.42E-2	6.66E-3	0.11	0.705%	1.82	2.52
3	164	1.21E6	3.42E-2	6.73E-3	0.11	0.304%	6.65	6.73

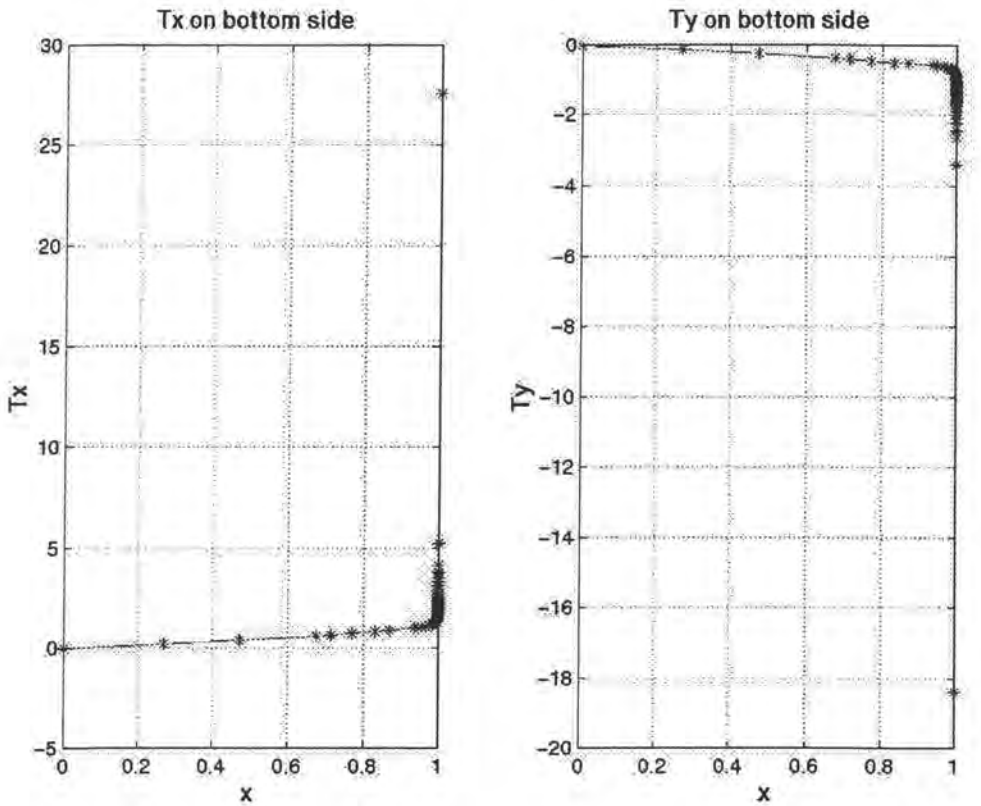

Figure 5.14: Traction T_x and T_y on bottom side OA in clamped plate model

CONCLUSION

A new spline based, adaptive grid generation technique is developed and employed in some benchmark problems in elasticity posed on a two-dimensional domain. The resulting grids and the solutions are very satisfactory. The main features of the present adaptive technique are as follows:

- The user can begin the BEM calculation on a very crude grid. This reduces the amount of user input and saves data preparation time.
- The size of the elements to accurately describe curved boundaries and the size of the elements to accurately describe the rapid variation of functions are determined from the same framework of spline interpolation. This is a major advantage because the representations of the geometric and the functional variations are improved simultaneously.
- The adaptive process is fully automatic in the sense that the errors are calculated at the end of each cycle of grid improvement and method proceeds to the next cycle of improvement, when necessary, without any user intervention.
- The adaptive technique adds new nodes on the boundary only at locations of large curvature in the geometry or rapid variations in traction and displacement. This keeps the number of simultaneous equations to be solved to a low value.

REFERENCE

[1] Milkhlin S.G., *Approximate Solutions of Differential and Integral Equations,* Oxford, Pergamon Press, 1965

[2] Rizzo F.J., *An Integral equation approach to boundary value problems of classical elastostatics* Q. Appl. Math. 25 1967, pp.93-116

[3] Kane J.H., *Boundary element analysis in engineering continuum mechanics,* Englewood Cliffs, New Jersey, Prentice Hall, 1994, pp.21-25

[4] Mitra A.K., Ingber M.S., *A multiple-node method to resolve the difficulties in the boundary integral equation method caused by corners and discontinuous boundary condition,* International journal for numerical methods in engineering, Vol.36, 1993, pp.1735-1746

[5] Mitra A.K., Salazar L.R., Sayyer M.L., *Spline assisted grid optimization scheme for the boundary element method.* Boundary elements XV, New York, Elsevier Science Publishers, 1993, pp.643-656