"FINITE ELEMENT METHOD APPLIED TO SOME BOUNDARY VALUE PROBLEMS"

BY

IYAYA C.C. WANJALA

This dissertation is submitted in partial fulfilment for the degree of Masters of Science in Applied Mathematics in the Department of Mathematics.

UNIVERSITY OF NAIROBI

JUNE 1992

DECLARATION

This disserttation is my own work and has not been presented for a degree in any other University.

Signature .....

IYAYA CHEBAYI CHARLES WANJALA

This dissertation has been submitted for examination with my approval as University Supervisor.

Signature ..W.Ogana.....

Prof. B W OGANA

ACKNOWLEDGEMENT

The Scholarship that enabled me to pursue to completion of the Msc Course in Applied Mathematics was awarded by the courtesy of the University of Nairobi. I am indeed grateful for the golden favour.

My special thanks go to several of my Lecturers. Prof. W.B Ogana granted some of his time an helped to create the atmosphere in which the completion of this project was possible. He has a large measure of my gratitude for suggesting the writing of this project in the first place and for giving much active encouragement throughout the working. He has been most helpful in reviewing parts of the manuscript, providing stimulating ideas, and making valuable suggestions.To Messrs Owino. A.Z, Alfred W Manyonge, D.O. Adongo, A.A. Wasike, G.M. Mukyekho (I.A.S) and Fredrick Okello (Computer Technician), whose friendship and research association were most treasured during this writing of project, I express my sincere appreciation.

I am also indebted to Florence Kayere and Irene Karima (secretaries) who tirelessly gave me excellent typing assistance.

My parents Charles Iyaya and Rinah Nakhanu are the sole basement of my being. To them, I express king size sincere appreciation.

Finally, I am grateful to my wife Knite for her patience and understanding while enduring the book widow syndrome for over a year.

Wambaya was tooth less by then , but he remained patient with his grandmother Diana Makola who deserves much more thanks for her immeasurable tolerance
and love for her grandson.

I.C.C.W.

CONTENTS

CHAPTER ONE

1.1 GENERAL INTRODUCTION

The Finite Element Method (FEM) is one of the numerical analysis techniques for obtaining approximate solutions to a wide variety of engineering problems. Others are finite difference and boundary element methods. Suitability of any method depends on the nature of the problem to be solved.

The basic premise of the FEM is that a solution region can be analytically modeled or approximated by replacing it with an assemblage of discrete elements. Since these elements can be put together in a variety of of ways ,they can be used to represent exceedingly complex shapes.

In a continuum problem of any dimension the field variable possesses infinitely many values because it is a function of each generic point in the body or solution region. Thus the problem is one with an infinite number of unknowns. The finite element discretization procedures reduce the problem to one of a finite number of unknowns by dividing the solution region into elements and by expressing the unknown field variable in terms of assumed approximating function within each element. The approximating functions are defined in terms of the values of the field variables at specified points called nodes or nodal points. Nodes usually lie on the element boundaries where adjacent elements are considered to be connected. In addition to boundary nodes ,an element may also have a few interior nodes. The nodal values of the field variable

and the interpolation functions for the elements completely define the behaviour of the field variable within the elements. For the finite element representation of a problem, the nodal values of the field variable become the new unknowns. Once these unknowns are found ,the interpolation functions define the field variable througout the assemblage of elements.

Clearly, the nature of the solution and degree of approximation depend not only on the size and number of the elements used but also on the interpolation functions selected. Often functions are chosen so that the field variable or its derivaties are continuous across adjoining element boundaries.

An important feature of the FEM which sets it apart from other approximate numerical methods is the ability to formulate solutions for individual elements before putting them together to represent the entire problem. In essence a complex problem reduces to considerably simplified problems.

FEM has also a variety of ways in which one can formulate the properties of individual elements. These are basically four approaches:

(i) Direct approach.

(ii) Variational approach

(iii) Weighted residuals approach

(iv) Energy balance approach.

Regardless of the approach used to find the element properties, the solution of a continuum problem by the FEM always follows an orderly step-by-step process.

The summary of how the FEM works is:

(i) Discretize the solution region into afinite number of elements

(ii) Select interpolation functions

(iii) Find element properties

(iv) Perform coordinate transformation if need be

(v) Assemble the element properties to obtain the system equations

(vi) Solve the system equations

(vii) make additional computations if desired.

CHAPTER TWO

2.1 DISCRETIZATION OF CONTINUUM

The discretization of the domain into a series of elements is the first of a series of steps that must be performed when solving a problem. This particular step depends on the use of engineering judgement. The application of poor or improper judgement will produce inaccurate results even though all of the other steps are rigorously adhered to. Since there are no setrules for reaching this goal because of the vastly different circumstances encountered from one problem to another ,some helpful guidelines emerge from the large amount of experience in finite element analysis.

The first question one must subject oneself to are the kind of elements to use and whether one should mix several different types of elements. The answers to these questions depends on the physics of the problem under study. Often one type of element is used to represent the continuum unless circumstances dictate otherwise. The most popular and versatile elements,because of the ease with which they can be assembled to fit complex geometries, are triangular elements in 2-dimensions and tetrahedral elements in 3-dimensions. These elements can have any number of exterior and interior nodes depending on the type of interpolation functions defined for them.

A uniform element mesh is easy to construct,but it may not always provide a good representation of the continuum. In regions of the solution domain where the gradient of the field variable is expected to vary

relatively fast, a finer element mesh should be used. It is convenient to place nodes and element boundaries at locations where point external actions are applied and where there are abrupt changes in the continuum. If the boundary of the region has any corners, nodes should also be placed at these corners. More elements should be used in regions where the boundary is irregular than in regions where it is smooth.

The continuum should be discretized so as to give the element a well proportioned shape (lengths of the sides should not be very different) . Long, narrow elements should be avoided because they lead to a solution with directional bias that may not be correct.

Provided that the the elements obey the requirements for a convergent solution, we may expect that the more elements we use to model the solution domain, the better the accuracy of our results.

2.2 LINEAR INTERPOLATION FUNCTIONS

The FEM is based upon the concept of approximating a continuous function (temperature, pressure, displacement, etc) by a discrete model that is composed of a set of piecewise continuous functions which are defined over a finite number of elements. The most popular form of the element function is the polynomial. The order of the polynomial depends on the number of items known about the continuous function at each element node.

Finite elements can be classified into three groups according to the order of the element polynomial. These groupings are simplex, complex and multiplex

(Oden,1972).

Simplex Elements have an approximating polynomial that consists of the constant term plus the linear terms. The number of coefficients in the polynomial is equal to the dimension of the coordinate space plus one.

Complex Elements have a polynomial function consisting of the constant and linear terms plus second, third and higher order terms as they are needed. The complex elements can have the same shapes as the simplex elements, but they have additional boundary nodes and may also have internal nodes.

The multiplex elements have polynomials containing higher order terms but the element boudaries must be parallel to the coordinate axes to achieve inter-element continuity. The element boundary of the simplex and complex elements are not subjected to this restriction.

For the purpose of this project only the simplex and Complex elements will be considered. The discussion of the complex only covers quadratic triangular elements but multiplex elements along with the isoparametric elements are beyond the scope of this project. Also only one and two dimensional simplex element formulation will be considered.

2.3 ONE DIMENSIONAL SIMPLEX ELEMENT:

This is a line segment with a length L and two nodes, one at each end as below:

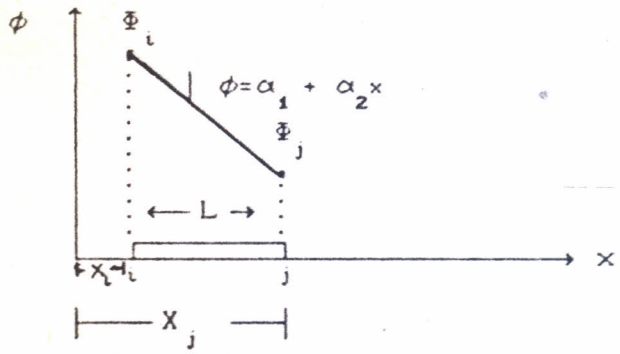


Fig. 2.1.

The nodes are denoted by i and j and the nodal values by Φ_i and Φ_j. The origin of the coordinate system is outside the element. The polynomial function for a scalar quantity, Φ, is

$$\phi = \alpha_1 + \alpha_2 x \qquad 2.3.1$$

The coefficients α_i and α_j can be determined by using the nodal conditions.

$$\phi = \Phi_i \text{ at } x = x_i$$

$$\phi = \Phi_j \text{ at } x = x_j$$

which result in a pair of equations:

$$\Phi_i = \alpha_1 + \alpha_2 x_i$$

$$\Phi_j = \alpha_1 + \alpha_2 x_j$$

which yield

$$\alpha_1 = \frac{\Phi_i x_j - \Phi_j x_i}{L}$$

$$\alpha_2 = \frac{\Phi_j - \Phi_i}{L}$$

When solved. Equation (2.3.1) then becomes

$$\phi = \frac{(x_j - x)}{L}\Phi_i + \frac{(x - x_i)}{L}\Phi_j \qquad 2.3.2$$

The linear functions of x in (2.3.2) are called shape functions or interpolation functions and are denoted by N. Each shape function must have a subscript to denote the node it is associated with. The shape functions in (2.3.2) are

$$N_i = \frac{(x_j - x)}{L} \quad \text{and} \quad N_j = \frac{(x - x_i)}{L}$$

Equation (2.5.2) can also be written using matrix notation

$$\phi = N_i \Phi_i + N_j \Phi_j = [N]\langle\Phi\rangle \qquad 2.3.3$$

where

$[N] = [N_i N_j]$ is a row matrix and

$$\langle\Phi\rangle = \begin{Bmatrix} \Phi_i \\ \Phi_j \end{Bmatrix}$$ is a column vector

N_i, N_j = $\begin{vmatrix} 1,0 \text{ at node } i \\ 0,1 \text{ at node } j \end{vmatrix}$ respectively. These values are characteristic of shape functions.

2.4 TWO DIMENSIONAL SIMPLEX ELEMENT:

This is a triangle with straight sides and three nodes oneat each corner. Lebelling of nodes is counterclockwise from node i ,which is arbitrarily specified.

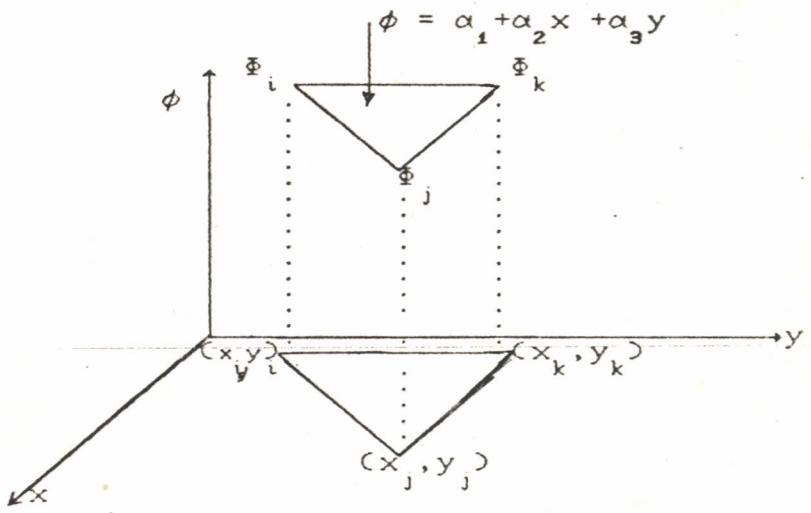


Fig. 2.2

The interpolating polynomial is

$$\phi = \alpha_1 + \alpha_2 x + \alpha_3 y \qquad 2.4.1$$

with the nodal conditions

$$\phi=\Phi_i \text{ at } x=X_i, y=Y_i$$

$$\phi=\Phi_j \text{ at } x=X_j, y=Y_j$$

$$\phi=\Phi_k \text{ at } x=X_k, y=Y_k$$

and (2.4.1) produces the system of equations

$$\Phi_i = \alpha_1 + \alpha_2 X_i + \alpha_3 Y_i$$

$$\Phi_j = \alpha_1 + \alpha_2 X_j + \alpha_3 Y_j$$

$$\Phi_k = \alpha_1 + \alpha_2 X_k + \alpha_3 Y_k$$

which yield

$$\alpha_1 = (1/2A) \times \left[\left(X_j Y_k - X_k Y_j\right)\Phi_i + \left(X_k Y_i - X_i Y_k\right)\Phi_j + \left(X_i Y_j - X_j Y_i\right)\Phi_k\right]$$

$$\alpha_2 = (1/2A)\left[\left(Y_j - Y_k\right)\Phi_i + \left(Y_k - Y_i\right)\Phi_j + \left(Y_i - Y_j\right)\Phi_k\right] \qquad 2.4.2$$

$$\alpha_3 = (1/2A).\left[\left(X_k - X_j\right)\Phi_i + \left(X_i - X_k\right)\Phi_j + \left(X_j - X_i\right)\Phi_k\right]$$

where the determinant

$$\begin{vmatrix} 1 & X_i & Y_i \\ 1 & X_j & Y_j \\ 1 & X_k & Y_k \end{vmatrix} = 2A \qquad 2.4.3$$

and A is the area of the triangle .

Putting these values of α_1, α_2 and α_3 in (2.4.1) and rearranging we obtain

$$\phi = N_i \Phi_i + N_j \Phi_j + N_k \Phi_k \qquad 2.4.4$$

where

$$N_i = 1/2A.\left[a_i + b_i x + c_i y\right] \text{ and } \begin{cases} a_i = X_j Y_k - X_k Y_j \\ b_i = Y_j - Y_k \\ c_i = X_k - X_j \end{cases}$$

$$N_j = 1/2A.\left[a_j + b_j x + c_j y\right] \text{ and } \begin{cases} a_j = X_k Y_i - Y_k X_i \\ b_j = Y_k - Y_i \\ c_j = X_i - X_k \end{cases} \qquad 2.4.5$$

$$N_k = 1/2A.\left[a_k + b_k x + c_k y\right] \text{and } \begin{cases} a_k = X_i Y_j - Y_j X_i \\ b_k = Y_i - Y_j \\ c_k = X_j - X_i \end{cases}$$

The evaluation of N_i at node i produces

$$N_i = 1/2A.\left[a_i + b_i x + c_i y\right]$$

The terms within the bracket are the value of the determinant in (2.4.3). Thus $N_i = (1/2A)(2A) = 1$ at node i and it is zero at nodes two and three.

The scalar quantity ϕ is a function of a set of shape functions which are linear in x and y. This means that the gradient in either the x or y directions will be constant. A constant gradient within any element means that many small elements have to be used to approximate a rapid change in the value of ϕ.

2.5 COMPLEX ELEMENTS

Quadratic triangular element

It has six nodes [Brebbia,(1987)] and the interpolating polynomial is

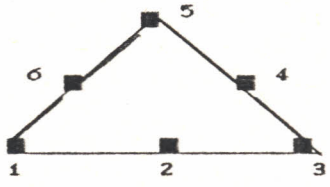


Fig 2.3

$$\phi = \alpha_1 + \alpha_2 x + \alpha_3 y + \alpha_4 x^2 + \alpha_5 xy + \alpha_6 y^2 \qquad 2.5.1$$

The α_i can be determined in a similar way as in the previous discussion. However, the algebra becomes more involved as the number of nodes increases. The direct derivation of the shape functions is an alternative and preferable method.

Thus a general equation for calculating the interpolation functions for an n^{th} order triangular element may be expressed by the following simple formula [Silvester, (1972)]

$$N_{\alpha\beta\gamma}(L_1, L_2, L_3) = N_\alpha(L_1) N_\beta(L_2) N_\gamma(L_3)$$

where $$N_\alpha(L_1) = \prod_{i=1}^{\alpha} \left(\frac{nL_1 - i + 1}{i} \right) ; \alpha \geq 1 \qquad 2.5.2$$

$$= 1 \quad ; \alpha = 0$$

For $N_\beta(L_2)$ and $N_\gamma(L_3)$ the formula has the same form . n is the order of the polynomial. For this particular case, n=2.

2.6 NATURAL COORDINATES

A coordinate system that relies on the element geometry for its definition and whose coordinates range between zero and unity within the element is known as a natural coordinate system.

The determination of the system of equations for the nodal values involves the integration of the shape functions or their derivatives or both over the element. This integration is easier when the interpolation equation is written in terms of an element coordinate system i.e a coordinate system located on or within the boundaries of the element.

The use of natural coordinates in deriving interpolation functions is particularly advantageous because special closed forms of integration formulas can often be used to evaluate the integrals in the element equations.

The basic purpose of a natural coordinate system is to describe the location of a point inside an element in terms of coordinates associated with the nodes of the element. We denote the natural coordinates as $L_i (i=1,2,\ldots,n)$, where n is the number of external nodes of the element.

The integral equations which simplifies the evaluation of length and area integrals are summarised below [Zienkiewicz-(1975)]:

$$\int_l L_1^a L_2^b dl = \frac{a!b!}{(a+b+1)!}\, l \qquad 2.6.1$$

$$\int_A L_1^a L_2^b L_3^c dA = \frac{a!b!c!}{(a+b+c+2)!}\, 2A \qquad 2.6.2$$

In this case the area coordinates L_1 L_2 and L_3 correspond to the shape functions N_i, N_j and N_k.

Equation (2.6.1) is used to evaluate integrals that are a function of the length along an edge of the element. The quantity l thus is the distance between the two nodes that define the edge under consideration. The real convenience of (2.6.1) and (2.6.2) will become apparent when we consider specific applications.

CHAPTER THREE

A FINITE ELEMENT FORMULATION OF SOME BOUNDARY VALUE PROBLEMS:

3.1 FINITE ELEMENT EQUATIONS:

Suppose that the variable ϕ is to be found in a three-dimensional solution domain Ω bounded by the surface Γ. For steady-state problems the field equation to be solved is the quasi-harmonic equation expressed in general terms as

$$\frac{\partial}{\partial x}\left(K_{xx}\frac{\partial \phi}{\partial x}\right) + \frac{\partial}{\partial y}\left(K_{yy}\frac{\partial \phi}{\partial y}\right) + \frac{\partial}{\partial z}\left(K_{zz}\frac{\partial \phi}{\partial z}\right) + Q = 0 \qquad 3.1.1$$

where K_{xx}, K_{yy}, K_{zz} and Q are given functions independent of ϕ but can be functions of x, y and z. The quantities K_{xx}, K_{yy} and K_{zz} are bounded away from zero in Ω. The physical interpretation of the parameters in (3.1.1) depends on the particular physical problem.

The description of the field problem is not complete until boundary conditions are specified i.e equation (3.1.1) must be solved subject to additional constraints imposed on the boundary surface. Usually, on some part of the boundary, the value of ϕ is a specified function as

$$\phi = \Phi(x,y,z) \quad \text{on } S_1 \text{ (Dirichlet BC)} \qquad 3.1.2$$

while the remainig part of the boundary we have the condition

$$K_{xx}\frac{\partial \phi}{\partial x} l_x + K_{yy}\frac{\partial \phi}{\partial y} l_y + K_{zz}\frac{\partial \phi}{\partial z} l_z + q + h(\phi - \phi_\infty) = 0$$

on S_2 (Cauchy BC) 3.1.3

where q and h are known a priori and l_x, l_y, and l_z in (3.1.3) are direction cosines of a vector that is normal

to the surface. The union of S_1 and S_2 forms the complete boundary Γ .

Equation (3.1.1) along with the boundary conditions is the governing equation for a three-dimensional heat transfer (Kreith-1973) . In this case K_{xx}, K_{yy}, and K_{zz} are the thermal conductivities ,Q is the internal heat source or sink, q is the heat flux over a portion of the surface ,and h is the convection coefficient. The field variable ϕ is the temperature. The governing equations for one-and two-dimensional heat transfer can be obtained from (3.1.1) by noting that $\partial\phi/\partial y = 0$ and/or $\partial\phi/\partial z = 0$. If both q and h are zero on a portion of the boundary where ϕ is not specified, then (3.1.3) reduces to

$$K_{xx}\frac{\partial\phi}{\partial x}l_x + K_{yy}\frac{\partial\phi}{\partial y}l_y + K_{zz}\frac{\partial\phi}{\partial z}l_z = 0 \qquad 3.1.4$$

which is a condition for zero heat transfer (an insulated boundary).

With appropriate specifications of the parameters in the equations (3.1.1), (3.1.2), and (3.1.3), the governing equations for the torsion of noncircular solid sections [Timoshenko and Goodier-(1970)], the irrotational flow of fluids [Vallentine-(1950)], confined groundwater flow [Harr-(1962)], fluid-film lubrication [Huebner-(1975)] etc are derived from the quasi-harmonic equation (3.1.1). The formulation that follows is in terms of heat transfer . However it is as good as the general case.

The calculus of variations provides an alternative method of formulating this heat transfer

problem. Variational calculus states that the minimization of the functional

$$\chi = \int_v \frac{1}{2}\left[K_{xx}\left(\frac{\partial\phi}{\partial x}\right)^2 + K_{yy}\left(\frac{\partial\phi}{\partial y}\right)^2 + K_{zz}\left(\frac{\partial\phi}{\partial z}\right)^2 - 2Q\phi\right] dv +$$

$$\int_s \left[q\phi + \frac{1}{2}h(\phi - \phi_\infty)^2\right] ds \qquad 3.1.5$$

requires that the differential equation (3.1.1) with its boundary conditions (3.1.2) and (3.1.3) be satisfied. Thus, any temperature distribution that makes χ (3.1.5), a minimum also satisfies the governing differential equations and , therefore , is a solution to the problem under study.

Equation (3.1.5) is the starting point for determining the temperature at each node. We minimize (3.1.5) by using our set of element functions, each defined over a single element and in terms of nodal values. The nodal values (Φ) are the unknown values in our formulation. Since they determine the value of the functional χ ,the minimization of χ must be done with respect to these quantities.

The objective of this section is to perfom this minimization on the general formulation given in (3.1.5). Here the functional is minimized before the evaluation of intergrals. This approach allows the selection of element characteristics most applicable to a particular problem at a time we solve the problem.

Equation (3.1.5) can be rewritten by defining the matrices as

$$\{g\}^T = \left[\frac{\partial\phi}{\partial x} \quad \frac{\partial\phi}{\partial y} \quad \frac{\partial\phi}{\partial z}\right] \qquad 3.1.6$$

$$D = \begin{bmatrix} K_{xx} & 0 & 0 \\ 0 & K_{yy} & 0 \\ 0 & 0 & K_{zz} \end{bmatrix} \qquad 3.1.7$$

so that

$$\chi = \int_v \frac{1}{2} \left[\{g\}^T [D] \{g\} - 2\phi Q \right] dv + \int_s \phi q \, ds +$$

$$\int_s \left[\frac{1}{2} h' (\phi^2 - 2\phi\phi_\infty + \phi_\infty^2) \right] ds \qquad 3.1.8$$

Now the functions for ϕ are not continuos over the entire region but, instead are defined over individual elements, $\phi^{(e)}$,the integrals in (3.1.8) are separated into the individual elements, yielding

$$\chi = \sum_{e=1}^{E} \left[\int_{v^{(e)}} \frac{1}{2} \left[\{g^{(e)}\}^T [D^{(e)}] \{g^{(e)}\} \right] dv \right.$$

$$- \int_{v^{(e)}} \phi^{(e)} Q^{(e)} \, dv + \int_{s_1^{(e)}} \phi^{(e)} q^{(e)} \, ds +$$

$$\left. \int_{s_2^{(e)}} \frac{1}{2} h^{(e)} \left[\phi^{(e)} \phi^{(e)} - 2\phi^{(e)} \phi_\infty + \phi_\infty^2 \right] ds \right] \qquad 3.1.9$$

where E is the total number of elements . Equation (3.1.9) can also be written symbolically as

$$\chi = \chi^{(1)} + \chi^{(2)} + \ldots\ldots + \chi^{(E)} = \sum_{e=1}^{E} \chi^{(e)} \qquad 3.1.10$$

where $\chi^{(e)}$ is the contribution of a single element to χ.

The minimization of χ occurs when

$$\frac{\partial \chi}{\partial \{\Phi\}} = \frac{\partial}{\partial \{\Phi\}} \sum_{e=1}^{E} \chi^{(e)} = \sum_{e=1}^{E} \frac{\partial \chi^{(e)}}{\partial \{\Phi\}} = 0 \qquad 3.1.11$$

The derivatives $\partial\chi^{(e)}/\partial\{\Phi\}$ in (3.1.11) cannot be evaluated until the integrals in (3.1.9) have been written in terms of the nodal values (Φ). Recall from (2.3.3) that

$$\phi = [N]\langle\Phi\rangle$$

therefore $\phi^{(e)} = [N^{(e)}]\langle\Phi\rangle$. 3.1.12

and we can evaluate (3.1.6), which along with (3.1.12) can be subsituted into the appropriate places in (3.1.9).

Thus

$$\left\{g^{(e)}\right\} = \begin{Bmatrix} \frac{\partial\phi^{(e)}}{\partial x} \\ \frac{\partial\phi^{(e)}}{\partial y} \\ \frac{\partial\phi^{(e)}}{\partial z} \end{Bmatrix} = \begin{bmatrix} \frac{\partial N_1^{(e)}}{\partial x} & \frac{\partial N_2^{(e)}}{\partial x} & \cdots & \frac{\partial N_p^{(e)}}{\partial x} \\ \frac{\partial N_1^{(e)}}{\partial y} & \frac{\partial N_2^{(e)}}{\partial y} & \cdots & \frac{\partial N_p^{(e)}}{\partial y} \\ \frac{\partial N_1^{(e)}}{\partial z} & \frac{\partial N_2^{(e)}}{\partial z} & \cdots & \frac{\partial N_p^{(e)}}{\partial z} \end{bmatrix} \begin{Bmatrix} \Phi_1 \\ \Phi_2 \\ \vdots \\ \Phi_p \end{Bmatrix}$$

or $$\left\{ g^{(e)}\right\} = \left[B^{(e)}\right]\left\{\Phi\right\}$$ 3.1.14

where [B] contains information related to the derivatives of the shape functions. These terms are presently unknown because the shape functions have not been specified. Utilizing (3.1.12) and (3.1.14) allows the element integrals in (3.1.9) to be written as

$$\chi^{(e)} = \int_{v^{(e)}} \frac{1}{2} \langle\Phi\rangle^T [B^{(e)}]^T [D^{(e)}][B^{(e)}]\langle\Phi\rangle \; dv$$

$$- \int_{v^{(e)}} Q[N^{(e)}]\langle\Phi\rangle \; dv + \int_{s_1^{(e)}} q[N^{(e)}]\langle\Phi\rangle \; ds +$$

$$\int_{s_2^{(e)}} \frac{1}{2} h \langle\Phi\rangle^T [N^{(e)}]^T [N^{(e)}]\langle\Phi\rangle ds -$$

$$\int_{s_2^{(e)}} 2\phi_\infty [N^{(e)}]\langle\Phi\rangle ds + \int_{s_2^{(e)}} \frac{h}{2} \phi_\infty^2 \; ds \qquad 3.1.15$$

The quantities Q, q, ϕ_∞, and h are known coefficients. They are left within the integral because it is possible for them to vary over the element . The differentiation of (3.1.15) with respect to $\langle\Phi\rangle$ is accomplished quite readily when the differentiation rules of matrices,

[Segerlind-(1976), as shown in Appendix B] are utilized. Integral by integral differentiation of (3.1.15) yields;

$$\frac{\partial}{\partial\{\Phi\}} \int_{v^{(e)}} \frac{1}{2} \{\Phi\}^T [B^{(e)}]^T [D^{(e)}][B^{(e)}]\{\Phi\} \; dv =$$

$$\int_{v^{(e)}} [B^{(e)}]^T [D^{(e)}][B^{(e)}]\{\Phi\} \; dv$$

$$\frac{\partial}{\partial\{\Phi\}} \int_{v^{(e)}} Q\,[N^{(e)}]\{\Phi\}\, dv = \int_{v^{(e)}} Q[N^{(e)}]^T dv$$

$$\frac{\partial}{\partial\{\Phi\}} \int_{s_1^{(e)}} q[N^{(e)}]\{\Phi\} ds = \int_{s_1^{(e)}} q[N^{(e)}]^T ds$$

$$\frac{\partial}{\partial\{\Phi\}} \int_{s_2^{(e)}} \frac{h}{2} \{\Phi\}^T [N^{(e)}]^T [N^{(e)}]\{\Phi\} ds =$$

$$\int_{s_2^{(e)}} h\,[N^{(e)}]^T [N^{(e)}]\{\Phi\} ds$$

$$\frac{\partial}{\partial\{\Phi\}} \int_{s_2^{(e)}} h\,\phi_\infty\,[N^{(e)}]\{\Phi\} ds = \int_{s_2^{(e)}} h\,\phi_\infty\,[N^{(e)}]^T ds$$

$$\int_{s_2^{(e)}} \frac{h}{2} \phi_\infty^2 \, ds = 0 \qquad 3.1.16$$

The element contribution, $\partial\chi^{(e)}/\partial\{\Phi\}$, to the total minimization process, $\partial\chi/\partial\{\Phi\}$, is

$$\frac{\partial \chi^{(e)}}{\partial\{\Phi\}} = \left(\int_{v^{(e)}} [B^{(e)}]^T [D^{(e)}][B^{(e)}] \; dv + \int_{s_2^{(e)}} h\,[N^{(e)}]^T [N^{(e)}] ds \right) \{\Phi\} - \int_{v^{(e)}} Q[N^{(e)}]^T dv$$

$$+ \int_{s_1^{(e)}} q\,[N^{(e)}]^T ds - \int_{s_2^{(e)}} h\,\phi_\infty\,[N^{(e)}]^T ds \qquad 3.1.17$$

This set of integrals can be written in the condensed form

$$\frac{\partial \chi^{(e)}}{\partial\{\Phi\}} = [K^{(e)}]\{\Phi\} + \{f^{(e)}\} \qquad 3.1.18$$

where

$$[K^{(e)}] = \left(\int_{v^{(e)}} [B^{(e)}]^T [D^{(e)}][B^{(e)}] \; dv + \int_{s_2^{(e)}} h\,[N^{(e)}]^T [N^{(e)}] ds \right) \qquad 3.1.19$$

and

$$\left\{f^{(e)}\right\} = -\int_{v^{(e)}} Q[N^{(e)}]^T dv$$

$$+\int_{s_1^{(e)}} q[N^{(e)}]^T ds - \int_{s_2^{(e)}} h\,\phi_\infty [N^{(e)}]^T ds \qquad 3.1.20$$

The final system of equations is obtained by substituting (3.1.18) into (3.1.11) giving

$$\frac{\partial \chi}{\partial \langle \Phi \rangle} = \sum_{e=1}^{E} (\,[K^{(e)}]\langle \Phi \rangle + \langle f^{(e)} \rangle) = 0 \qquad 3.1.21$$

or

$$[K]\langle \Phi \rangle = \langle F \rangle$$

where $$[K] = \sum_{e=1}^{E} [K^{(e)}] \qquad 3.1.22$$

and $$\langle F \rangle = -\sum_{e=1}^{E} \langle f^{(e)} \rangle \qquad 3.1.23$$

The integrals in (3.1.19) define an element conduction matrix ,$[K^{(e)}]$, and the integrals in (3.1.20) define the element force vector,$\langle f^{(e)} \rangle$,.

The evaluation of these integrals is discussed in the specific application areas in the follows sections:

3.2 ONE DIMENSIONAL HEAT TRANSFER

Consider the one-dimensional heat flow in an insulated rod (fig.below). The rod is

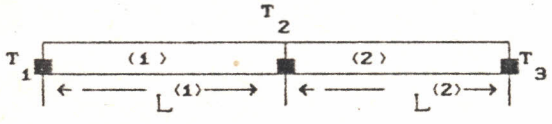


Fig.3.1

attached to a wall and has a specified heat input q. The other end is free and has a convection coefficient h and a surrounding media temperature of T_∞, ^oC . The rod is insulated,therefore,no heat loss occurs from the

circumferential surface.

The governing differential equation for the temperature distribution within the rod is

$$K_{xx} \frac{d^2T}{dx^2} = 0 \qquad 3.2.1$$

with boundary conditions

$$K_{xx} \frac{dT}{dx} + q = 0 \text{ at } x=0 \qquad 3.2.2$$

and

$$K_{xx} \frac{dT}{dx} + h(T - T_\infty) = 0 \text{ at } x=L \qquad 3.2.3$$

where the K_{xx} is the thermal conductivity of the material. The heat flux ,q, is positive if heat is moving out of the rod.

The functional representation of the governing differential equation is

$$\chi = \int_v \frac{K_{xx}}{2} \left(\frac{dT}{dx}\right)^2 dv + \int_s \left[qT + 1/2\, h(T-T_\infty)^2\right] ds \qquad 3.2.4$$

The minimization of χ is done w.r.t T (nodal temperature which is the unknown).

The rod is represented by two line elements with the nodal values T_1, T_2, and T_3 . The element equations for the temperature are:

$$T^{(1)} = N_1^{(1)} T_1 + N_2^{(1)} T_2$$
$$T^{(2)} = N_2^{(2)} T_2 + N_3^{(2)} T_2 \qquad 3.2.5$$

where the shape functions are those defined in (2.5.2), namely

$$N_i^{()} = \frac{X_j - x}{L^{()}} \quad , \quad N_j^{()} = \frac{x - X_i}{L^{()}}$$

The functional formulation separates into

$$\chi = \int_v \frac{K_{xx}}{2} \left(\frac{dT(x)}{dx}\right)^2 dv + \int_{s_1} qT(x) ds +$$

$$\int_{s_2} 1/2\ h\left[T(x)-T_\infty\right]^2 ds \qquad 3.2.6$$

where s_1 and s_2 are the surface areas where q and h are specified. The value of χ is obtained by substituting for the temperature, T(x), and evaluating the integrals.

The derivative of the temperature enters into the volume integral,(3.2.6). Differentiating (3.2.5) yields

$$\frac{dT^{(1)}}{dx} = \frac{1}{L^{(1)}}(-T_1 + T_2)$$

$$\frac{dT^{(2)}}{dx} = \frac{1}{L^{(2)}}(-T_2 + T_3) \qquad 3.2.7$$

The integral quantity χ is separated into its element components, and these components are minimized with respect to the nodal values before the integrals are evaluated. For this particular case,

$$\chi = \chi^{(1)} + \chi^{(2)} \qquad 3.2.8$$

where $\chi^{(1)}$ is the sum of the integrals for element one and $\chi^{(2)}$ is a similar sum for element two. Thus

$$\chi^{(1)} = \int_{v^{(1)}} \frac{C^{(1)}}{2L^{(1)}}(-T_1 + T_2)^2 dv + \int_{s^{(1)}} qT_1 ds \qquad 3.2.9$$

$$\chi^{(2)} = \int_{v^{(2)}} \frac{C^{(2)}}{2L^{(2)}}(-T_2 + T_3)^2 dv + \int_{s^{(2)}} \frac{h}{2}(T_3 - T\infty)^2\ ds$$

where $C^{(1)} = A^{(1)}K_{xx}^{(1)} / L^{(1)}$ and $C^{(2)} = A^{(2)}K_{xx}^{(2)} / L^{(2)}$.

We now differentiate each component of χ with respect to all the values. Starting with $\chi^{(1)}$

$$\frac{\partial \chi^{(1)}}{\partial T_1} = \int_{v^{(1)}} \frac{C^{(1)}}{L^{(1)}}(-T_1 + T_2)\ (-1) dv + \int_{s^{(1)}} q ds$$

$$\frac{\partial \chi^{(1)}}{\partial T_2} = \int_{v^{(2)}} \frac{C^{(1)}}{L^{(1)}}(-T_1 + T_2)\ dv$$

$$\frac{\partial \chi^{(1)}}{\partial T_3} = 0 \qquad 3.2.10$$

Evaluation of the integrals produces a set of equations

that can be written as

$$\frac{\partial \chi^{(1)}}{\partial \overline{T}} = \begin{bmatrix} C^{(1)} & -C^{(1)} & 0 \\ -C^{(1)} & C^{(1)} & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} T_1 \\ T_2 \\ T_3 \end{Bmatrix} + \begin{Bmatrix} qA_1 \\ 0 \\ 0 \end{Bmatrix} \qquad 3.2.11$$

Differentiating the second component gives

$$\frac{\partial \chi^{(2)}}{\partial \overline{T}_1} = 0$$

$$\frac{\partial \chi^{(2)}}{\partial \overline{T}_2} = \int_{v^{(2)}} \frac{C^{(2)}}{L^{(2)}}(-T_2 + T_3)(-1)\, dv$$

$$\frac{\partial \chi^{(2)}}{\partial \overline{T}_3} = \int_{v^{(2)}} \frac{C^{(2)}}{L^{(2)}}(-T_2 + T_3)\, dv + \int_{s^{(2)}} h(T_3 - T\infty)\, ds$$

or

$$\frac{\partial \chi^{(2)}}{\partial \langle \overline{T} \rangle} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & C^{(2)} & C^{(2} \\ 0 & -C^{(2)} & [C^{(2)} + hA_3] \end{bmatrix} \begin{Bmatrix} T_1 \\ T_2 \\ T_3 \end{Bmatrix} + \begin{Bmatrix} 0 \\ 0 \\ -hA_3 T\infty \end{Bmatrix} \qquad 3.2.12$$

once the integrals have been evaluated.

The minimization of χ with respect to the nodal values is

$$\frac{\partial \chi}{\partial \langle \overline{T} \rangle} = \frac{\partial \chi^{(1)}}{\partial \langle \overline{T} \rangle} + \frac{\partial \chi^{(2)}}{\partial \langle \overline{T} \rangle} = 0 \qquad 3.2.13$$

Thus if we sum (3.2.11) and (3.2.12) and set the result equal to zero , we will obtain the desired system of equations as:

$$\begin{bmatrix} C^{(1)} & -C^{(1)} & 0 \\ -C^{(1)} & [C^{(1)} + C^{(2)}] & -C^{(2)} \\ 0 & -C^{(2)} & [C^{(2)} + hA_1] \end{bmatrix} \begin{Bmatrix} T_1 \\ T_2 \\ T_3 \end{Bmatrix} + \begin{Bmatrix} qA_1 \\ 0 \\ -hA_3 T\infty \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \end{Bmatrix}$$

or

$$[K]\langle T \rangle = \langle F \rangle \qquad 3.2.14$$

The important concept to be realized in the procedure just discussed is that the system of equations can be constructed an element at a time. The element by

element summation indicated in (3.2.13) is a very methodical procedure and is readily implemented on the digital computer.

3.3 TWO DIMENSIONAL HEAT TRANSFER:

The kind of element used is a three-node triangular element. The shape functions for this linear triangular element are

$$N_\beta = 1/2A\,(a_\beta + b_\beta x + c_\beta y), \qquad \beta = i,j,k \tag{3.3.1}$$

where a_β, b_β, and c_β are defined in [2.6.3]. The temperature is given by

$$T = [N_i \; N_j \; N_k] \begin{Bmatrix} T_i \\ T_j \\ T_k \end{Bmatrix} \tag{3.3.2}$$

where T_i, T_j, and T_k are the nodal temperatures counterclockwise around the element from node i.

The gradient matrix, [B], is

$$[B] = 1/2A \begin{bmatrix} b_i & b_j & b_k \\ c_i & c_j & c_k \end{bmatrix} \tag{3.3.3}$$

and the material property, [D], is

$$[D] = \begin{bmatrix} K_{xx} & 0 \\ 0 & K_{yy} \end{bmatrix} \tag{3.3.4}$$

The element conduction matrix then follows. The first term becomes

$$\int_v [B]^T[D][B]dv = \int_v 1/4A^2 \begin{bmatrix} b_i & b_j & b_k \\ c_i & c_j & c_k \end{bmatrix}^T \begin{bmatrix} k_{xx} & 0 \\ 0 & k_{yy} \end{bmatrix}$$

$$\times \begin{bmatrix} b_i & b_j & b_k \\ c_i & c_j & c_k \end{bmatrix} dv \tag{3.3.5}$$

Assuming a unit thickness of the element, the element of

volume dv becomes dA with the integration over the area. All the terms under the integral sign in (3.3.5) are constants; thus, they can be removed yielding

$$\int_v [B]^T[D][B]dv = [B]^T[D][B] \int_A dA = A[B]^T[D][B] \qquad 3.3.6$$

Evaluating the matrix product yields

$$[K^{(e)}] = \frac{K_{xx}}{4A}\begin{bmatrix} b_i b_i & b_i b_j & b_i b_k \\ b_j b_i & b_j b_j & b_j b_k \\ b_k b_i & b_k b_j & b_k b_k \end{bmatrix} + \frac{K_{yy}}{4A}\begin{bmatrix} C_i C_i & C_i C_j & C_i C_k \\ C_j C_i & C_j C_j & C_j C_k \\ C_k C_i & C_k C_j & C_k C_k \end{bmatrix}$$

The second integral $\int_s h[N]^T[N]\,ds$, must be evaluated over a surface.

Substitution for [N] in terms of the shape functions and perfoming the matrix multiplication gives

$$\int_s h[N]^T[N]ds = h\int_s \begin{bmatrix} N_i N_i & N_i N_j & N_i N_k \\ N_j N_i & N_j N_j & N_j N_k \\ N_k N_i & N_k N_j & N_k N_k \end{bmatrix} ds \qquad 3.3.7$$

The shape functions are dependent on x and y; therefore, the products $N_i N_j$ etc. cannot be removed from under the integral sign. In addition, the value of the integral depends on which surface is experiencing the convection phenomenon. For example, if the side between nodes i and j is subjected to convection, then N_k is zero along this side and the integral reduces to

$$\int_s h[N]^T[N]ds = h\int_s \begin{bmatrix} N_i N_i & N_i N_j & 0 \\ N_j N_i & N_j N_j & 0 \\ 0 & 0 & 0 \end{bmatrix} ds \qquad 3.3.8$$

The evaluation of the product terms in (3.3.7) is easily done if area coordinates and the related formula are employed.

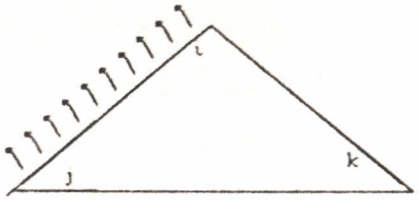


Fig.3.2

Assuming that L is measured from the side opposite node i ,we can write

$$L_1 = N_i \ , \ L_2 = N_j, L_3 = N_k$$

If we assume the side between nodes i and j to be the one experiencing the convection phenomenon, then $N_k = L_3 = 0$ along this surface and (3.3.7) becomes

$$\int_s h[N]^T[N]ds = h\int_{ij} \begin{bmatrix} L_1L_1 & L_1L_2 & 0 \\ L_2L_1 & L_2L_2 & 0 \\ 0 & 0 & 0 \end{bmatrix} dL \qquad 3.3.9$$

where ds = tdl and we have assumed a unit value for t. There are two types of products in (3.3.9) a squared quantity ,L_1^2 or L_2^2 and a cross product, L_1L_2. Starting with the squared term we have

$$\int_{l_{ij}} L_1^2 \, dl = \int_{l_{ij}} L_1^2 L_2^0 \, dl = \frac{2!0!}{(2+0+1)!} \, l_{ij} = \frac{l_{ij}}{3}$$

where the L_{ij} is the length of the side between nodes i and j. Integration of the cross product term gives

$$\int_{l_{ij}} L_1 L_2 dl = \frac{1!1!}{(1+1+1)!} \, l_{ij} = \frac{l_{ij}}{6}$$

Substitution of these results into (3.3.9) gives

$$\int_s h[N]^T[N]ds = \frac{hl_{ij}}{6} \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} dL \qquad 3.3.10$$

Similar treatment is done for sides between nodes j and k

and nodes k and i.

The three integrals in the element force vector are also easily evaluated if area coordinates are employed. Assuming Q constant within the element we have

$$\int_v [N]^T Q \, dv = Q\int_v [N]^T \, dv = Q\int_v \begin{bmatrix} L_1 \\ L_2 \\ L_3 \end{bmatrix} dv = \frac{Qv}{3}\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \qquad 3.3.11$$

The heat within the body is distributted equally to the three nodes. The integrals

$\int_{s_1} [N]^T q \, ds$ and $\int_{s_2} [N]^T hT\infty \, ds$ are both of the form

$$\int_s \begin{Bmatrix} N_i \\ N_j \\ N_k \end{Bmatrix} ds \qquad 3.1.12$$

and only one needs to be evaluated. Since it is a surface integral, (3.3.12) is subjected to the same considerations that were encountered when discussing (3.3.7). The results depend on which side of the element is subjected to the heat flux or the convection phenomenon h. Assuming q is constant over the surface, the integral (3.3.12) is

$$\int_s [N]^T q \, ds = q\ell_{ij} \begin{Bmatrix} 1 \\ 1 \\ 0 \end{Bmatrix} \qquad 3.3.13$$

same treatment applies to sides between nodes j and k and k and i.

The values of $\int_s [N]^T hT\infty \, ds$ are identical to (3.3.13) except that q is replaced by $hT\infty$.

3.4 POINT SOURCES

A point or line source is said to exist whenever the generation of heat Q or fluid occurs within a very small volume or within a very small area. Physical examples of line sources include steam and /or hot water pipes within the earth and conducting electrical wires

embedded within a product. In each case , the cross-section area of the pipe or conductor is very small compared with the surrounding media. Point sinks are also used in gruondwater problems. They occur as pumps that are removing water from an aquifer.

Consider the triangular element in figure (3.3) below with a line source Q^* located at X_o, Y_o. Since the heat source is located at a point, Q is no longer a constant throghout the volume but is a function of X and Y. Using unit impulse functions, $\delta(x-x_o)$ and $\delta(y-y_o)$ [Kaplan-(1962)], we can write

$$Q = Q^*\delta(x-x_o)\delta(y-y_o) \qquad 3.4.1$$

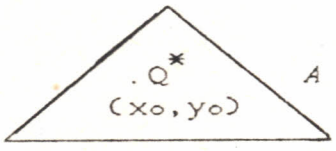
The integral

$\int_v [N]^T Q \, dv$ now becomes

$$\int_v [N]^T Q \, ds = Q^* \int_A \begin{Bmatrix} N_i \\ N_j \\ N_k \end{Bmatrix} \delta(x-x_o)\delta(y-y_o)dxdy \qquad 3.4.2$$

assuming a unit thickness. The integral of a function multiplied by an impulse function, however, is simply the function evaluated at Xo and Yo. Therefore

$$\int_v [N]^T Q \, ds = Q^* \int_A \begin{Bmatrix} N_i \\ N_j \\ N_k \end{Bmatrix}_{\substack{x=x_o \\ y=y_o}} \qquad 3.4.3$$



A point source within a triangular element

Fig. 3.3

Eqution (3.4.3) states that when a point (line) source is located within an element , the proportion allocated to each node is based on the relative values of N_i, N_j and N_k which are evaluated using the coordinates of the point source.

When subdividing a continuum that contains a

point(line) source,it may be convenient to place the source at a node.

CHAPTER FOUR

ASSEMBLY OF ELEMENT PROPERTIES

4.1 COORDINATE TRANSFORMATIONS:

It is generally more convenient to derive the element characteristics in a local coordinate system . The local coordinate system may be different for each element in the assembly . Thus if local coordinates are used , it is necessary before the individual element can be assembled to transform the element equations so that all element characteristics are referred to a common global coordinate system.

The element matrix equations to be transformed have the standard form

$$\left[K\right]\left\{X^{*}\right\} = \left\{b^{*}\right\} \qquad 4.1.1$$

where the asterisk designates a local reference system. If there exists a transformation matrix $[\phi]$ between the local and global systems , then we may write

$$\left\{X^{*}\right\} = \left[\Phi\right]\left\{x\right\} \qquad 4.1.2$$

and

$$\left\{b^{*}\right\} = \left[\Phi\right]\left\{b\right\} \qquad 4.1.3$$

where the column vectors $\langle x \rangle$ and $\langle b \rangle$ are referenced to a global system. Putting (4.1.2) and (4.1.3) in (4.1.1) there results

$$\left[K\right]\left[\Phi\right]\left\{x\right\} = \left[\Phi\right]\left\{b\right\} \qquad 4.1.4$$

or

$$[\Phi]^{-1}[K][\Phi]\{x\}=\{b\}$$

or

$$[\dot{K}]\{x\} = \{b\}$$

where

$$[\dot{K}] = [\Phi]^{-1}[K][\Phi] \qquad 4.1.5$$

Equation (4.1.5) gives the element matrix referenced to the global coordinate system provided that $[\Phi]^{-1}$ exists. If the column vectors $\{x^*\}$ and $\{b^*\}$ are directional quantities such as nodal displacements and forces ,then the transformation matrix is simply the collection of the direction cosines relating the two systems. In this case , the transformation matrix $[\Phi]$ is an orthogonal matrix with the property that its inverse equals its transpose i.e

$$[\Phi]^{-1} = [\Phi]^{T} \qquad 4.1.6$$

and thus we may write

$$[\dot{K}] = [\Phi]^{T}[K][\Phi] \qquad 4.1.7$$

4.2 ASSEMBLING THE PARTS

Having found the algebraic equations describing the characteristics of each element the next step is to combine all these equations to form a complete set governing the composite of elements. The procedure for constructing the system equations from the element equations is the same regardless of the type of problem being considered or complexity of the system of elements .

The system assembly procedure is based on the

fact that at nodes where elements are connected the value(s) of the unknown nodal variable(s) is (are) the same for all elements connecting that node. The consequence of this rule is the basis for the assembly process.

Introduction and illustration of the essential features of assembly is made using an elementary example.

4.3 TORSION OF NONCIRCULAR SECTION

The assembly procedure is demonstrated through solution of torsion of noncircular sections problem.

The problem is to calculate the shear stresses in a noncircular shaft subjected to a twisting moment T about the Z axis. The shear stress components at any point can be calculated using Standard St. Venant equations:

$$\tau_{zx} = \partial\phi/\partial y; \quad \tau_{zy} = -\partial\phi/\partial x \qquad 4.3.1$$

where ϕ is the stress function. The governing differantial equation is

$$\frac{1}{G}\frac{\partial^2\phi}{\partial x^2} + \frac{1}{G}\frac{\partial^2\phi}{\partial y^2} + 2\theta = 0 \qquad 4.3.2$$

with ϕ=0 on the boundary. G in (4.3.2) is the shear modulus and θ is the angle of twist per unit length. The applied torque T is calculated once ϕ is known using

$$T = 2\int_{Area}\phi dA \qquad 4.3.3$$

The stress function represents a surface covering the cross section of the shaft. The twisting moment is proportional to the volume under this surface while the shear stresses are related to the gradients in the x and y directions.

Assume the shaft to be made of a single material. Then (4.3.2) becomes

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} + 2G\theta = 0 \qquad 4.3.4$$

whose functional representation is

$$\chi = \int_v \left[1/2. \left(\frac{\partial \phi}{\partial x^2} \right)^2 + 1/2. \left(\frac{\partial \phi}{\partial y^2} \right)^2 - 2G\theta\phi \right] = 0 \qquad 4.3.5$$

which from equations (3.1.12) and (3.1.14) of chapter (3) can be rewritten as

$$\chi = \int_v \left[1/2 \{g\}^T [D] \{g\} - (2G\theta)\phi \right] dv \qquad 4.3.6$$

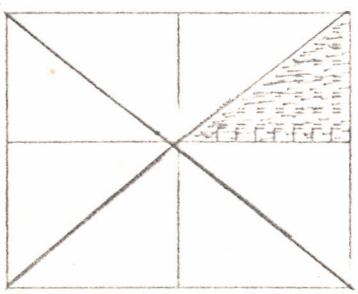
where $\{g\} = \begin{Bmatrix} \partial\phi/\partial x \\ \partial\phi/\partial y \end{Bmatrix}$ and $[D] = \begin{bmatrix} k_{xx} & 0 \\ 0 & k_{yy} \end{bmatrix}$

The column vector $\{g\}$ is related to the shear stress components in this application, while [D] becomes the identity matrix since $k_{xx} = k_{yy} = 1$. The minimization of χ w.r.t $\{\Phi\}$ results in the set of linear equations

$$\sum_{e=1}^{E} \int_{v^{(e)}} [B^{(e)}]^T [D^{(e)}] [B^{(e)}] dv \{\Phi\} = \sum_{e=1}^{E} \int_{v^{(e)}} [N^{(e)}]^T (2G\theta) dv$$

where $[N^{(e)}]$ is defined in (2.5.3) and $[B^{(e)}]$, the gradient matrix, is defined by (3.1.4).

The square shaft is used to illustrate the evaluation and the assemblage of the element matrices into a set of linear equations.



shaft is 1 cm^2

Fig. 4.1

This shaft has four axes of symmetry, therefore only

eighth of the total cross-section needs to be analysed. This fractional portion of cross section is divided into four elements as shown below:

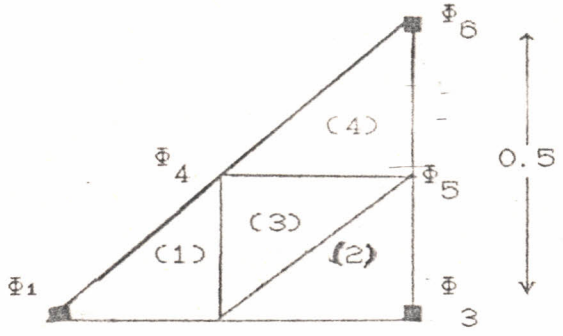


Fig. 4.2.

The element interpolating polynomials are:

$$\phi^{(1)} = N_1^{(1)}\Phi_1 + N_2^{(1)}\Phi_2 + 0\Phi_3 + N_4^{(1)}\Phi_4 + 0\Phi_5 + 0\Phi_6$$

$$\phi^{(2)} = 0\Phi_1 + N_2^{(2)}\Phi_2 + N_3^{(2)}\Phi_3 + 0\Phi_4 + N_5^{(2)}\Phi_5 + 0\Phi_6 \qquad 4.3.7$$

$$\phi^{(3)} = 0\Phi_1 + N_2^{(3)}\Phi_2 + 0\Phi_3 + N_4^{(3)}\Phi_4 + N_5^{(3)}\Phi_5 + 0\Phi_6$$

$$\phi^{(4)} = 0\Phi_1 + 0\Phi_2 + 0\Phi_3 + N_4^{(4)}\Phi_4 + N_5^{(4)}\Phi_5 + N_6^{(4)}\Phi_6$$

The general equation for the element stiffness matrix is:

$$[K^{(e)}] = \int_v [B^{(e)}]^T [B^{(e)}]\, dv$$

since $[D]=1$ for the torsion of noncircular section. Evaluation of $[B^{(e)}]$ involves differentiating $[\phi^{(e)}]$ with respect to X and Y. Confining the attention to element one,

$$\frac{\partial \phi^{(1)}}{\partial x} = \begin{bmatrix} \frac{\partial N_1^{(1)}}{\partial x} & \frac{\partial N_2^{(1)}}{\partial x} & 0 & \frac{\partial N_4^{(1)}}{\partial x} & 0 & 0 \end{bmatrix}$$

$$\overset{=1}{\underset{2A^{(1)}}{}} \begin{bmatrix} b_1^{(1)} & b_2^{(1)} & 0 & b_4^{(1)} & 0 & 0 \end{bmatrix}$$

$$\frac{\partial \phi^{(1)}}{\partial y} = \begin{bmatrix} \frac{\partial N_1^{(1)}}{\partial y} & \frac{\partial N_2^{(1)}}{\partial y} & 0 & \frac{\partial N_4^{(1)}}{\partial y} & 0 & 0 \end{bmatrix}$$

$$\overset{=1}{\underset{2A^{(1)}}{}} \begin{bmatrix} C_1^{(1)} & C_2^{(1)} & 0 & C_4^{(1)} & 0 & 0 \end{bmatrix}$$

The gradient $[B^{(1)}]$ is

$$[B^{(1)}] = \begin{bmatrix} b_1^{(1)} & b_2^{(1)} & 0 & b_4^{(1)} & 0 & 0 \\ c_1^{(1)} & c_2^{(1)} & 0 & c_4^{(1)} & 0 & 0 \end{bmatrix} \qquad 4.3\ 8$$

The area for this element is 1/32. Thus $1/2A^{(1)}=16$

The b and c coefficients are

$b_1^{(1)}=y_2-y_4=0.25$ $\qquad$ $C_1^{(1)}=X_2-X_4=0$

$b_1^{(1)}=y_4-y_1=0.25$ $\qquad$ $C_1^{(1)}=X_4-X_1=-0.25$

$b_1^{(1)}=y_1-y_2=0$ $\qquad$ $C_1^{(1)}=X_1-X_2=0.25$

Putting these values in (4.3.8) we get

$$[B^{(1)}] = \begin{bmatrix} -4 & 4 & 0 & 0 & 0 & 0 \\ 0 & -4 & 0 & 4 & 0 & 0 \end{bmatrix} \qquad 4.3\ 9$$

and the product

$$[B^{(1)}]^T[B^{(1)}] = \begin{bmatrix} -4 & 0 \\ 4 & -4 \\ 0 & 0 \\ 0 & 4 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} -4 & 4 & 0 & 0 & 0 & 0 \\ 0 & -4 & 0 & 4 & 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 16 & -16 & 0 & 0 & 0 & 0 \\ -16 & 32 & 0 & -16 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -16 & 0 & -16 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad 4.3.10$$

The element stiffness matrix is the integral of (4.3.10) so that we have

$[k^{(1)}]^T=[B^{(1)}][B^{(1)}]\int_{v^{(1)}} ds = [B^{(1)}][B^{(1)}]A^{(1)}$ assuming a unit thickness. Since $A^{(1)}=1/32$

$$[k^{(1)}] = 1/2 \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 2 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad 4.3.11$$

The evaluation of the volume integral

$$\langle f^{(1)} \rangle = \int_{v^{(1)}} 2G^{(1)} e \begin{bmatrix} N_1^{(1)} \\ N_2^{(1)} \\ 0 \\ N_4^{(1)} \\ 0 \\ 0 \end{bmatrix} dv$$

is easily done by employing the area coordinate system discussed in chapter two.

Defining the area coordinates as

$$L_1 = N_1^{(1)} \; ; \quad L_2 = N_2^{(1)} ; \text{ and } L_3 = N_4^{(1)} \qquad 4.3.12$$

The volume integral becomes

$$\langle f^{(1)} \rangle = \int_{v^{(1)}} 2G^{(1)} e \begin{bmatrix} L_1 \\ L_2 \\ 0 \\ L_4 \\ 0 \\ 0 \end{bmatrix} dv \qquad 4.3.13$$

Assuming a unit thickness and using the area integral for area coordinates(2.7.2),yields

$$\langle f^{(1)} \rangle = \frac{2G^{(1)} e A^{(1)}}{3} \begin{Bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{Bmatrix} \qquad 4.3.14$$

The system of equations for element one is

$$[k^{(1)}]\langle \Phi \rangle = \langle f^{(1)} \rangle$$

$$\text{or } 1/2\begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 2 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} \Phi_1 \\ \Phi_2 \\ \Phi_3 \\ \Phi_4 \\ \Phi_5 \\ \Phi_6 \end{bmatrix} = \frac{2G^{(1)}\theta A^{(1)}}{3}\begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad 4.3.15$$

A similar set of equations can be calculated for each of the other elements using identical procedure.

The assembled stiffness matrix contains stiffness coefficients obtained by directly adding the individual element stiffness coefficients in the appropriate locations in the global stiffness matrix . The resultant load vector for the system is also obtained by adding individul element loads at the appropriate locations in the column matrix of resultant nodal loads. Before adding various element matrices, first expand them to the dimension of the matrix . The relation between the system matrix and the degrees of freedom involved is that if the system has n degrees of freedom ,the system matrix [K] will be a square matrix of dimension nxn.

Mathematically if $[k]^{(e)}$ is the element matrix then the global system matrix is obtained as:

$$[k] = \sum_{e=1}^{m} [k]^{(e)} = [k]^{(1)} + [k]^{(2)} + \ldots\ldots$$

where m is the total number of elements in the assemblage.

The same summation principle applies for finding the column vectors of resultant external nodal actions from the element subvectors .

$$\langle K \rangle = \sum_{e=1}^{m} \langle R \rangle^{(e)} \quad 4.3.16$$

where $\langle R \rangle^{(e)}$ is the expanded column vector for element

and m is the total number of elements .

Though from a simple example , in principle the general procedure that applies to all finite element systems has been outlined.

The matrix formed is banded and sparse. The system matrices are sparse because each element has relatively few nodes compared to all the system nodes and only a few elements share each node. Numbering of the nodes causes the system matrices to be banded. The system coefficient matrices are usually symmetric-a characteristic that can often be used to advantage in storing the matrices.

The complete solution of the above used example is found in section (4.5).

4.4 INTRODUCING BOUNDARY CONDITIONS:

After assembly of the system equations the final equations will always have the form

$$[K]_{nxn}\ \{X\}_{nx1} = \{R\}_{nx1} \qquad 4.4.1$$

regardless of the type of problem.

For a unique solution of equation (4.4.1) at least one and sometimes more than one nodal variable must be specified and [K] must be modified to render it nonsingular. The required number of specified nodal variables is dictated by the physics of the problem. There are a number of ways to introduce boundary conditions to equation (4.4.1). Nodal variables should be introduced in a way that leaves the original number of

unchanged and avoids major restructuring of computer storage. Two of the methods are:

i) Nodal variables are included in (4.4.1) while retaining nxn system of equations to modify the matrices [k] and {R} as follows [Felipa & Clough-(1970), Segerlind-(1976) et al].If i is the subscript of a prescribed nodal variable, the ith row and the ith column of [K] are set equal to zero and K_{ii} is set equal to unity. The term R_i of the column vector {R} is replaced by the known value of X_i. Each of the n-1 remaining terms of {R} is modified by subtracting from it the value of the prescribed nodal variable multiplied by the appropriate column term from the original [K] matrix. this procedure is repeated for each prescribed X_i until all of them have been included.

As an example with only four system equations; equation (4.4.1) expands to the form

$$\begin{bmatrix} K_{11} & K_{12} & K_{13} & K_{14} \\ K_{21} & K_{22} & K_{23} & K_{24} \\ K_{31} & K_{32} & K_{33} & K_{34} \\ K_{41} & K_{42} & K_{43} & K_{44} \end{bmatrix} \begin{Bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{Bmatrix} = \begin{Bmatrix} R_1 \\ R_2 \\ R_3 \\ R_4 \end{Bmatrix}$$

Suppose that for this system nodal variables X_1 and X_3 are specified as $X_1 = \beta_1$ and $X_3 = \beta_3$. When these boundary conditions are inserted, the equations become

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & K_{33} & 0 & K_{24} \\ 0 & 0 & 1 & 0 \\ 0 & K_{42} & 0 & K_{44} \end{bmatrix} \begin{Bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{Bmatrix} = \begin{Bmatrix} \beta_1 \\ R_2 - K_{21}\beta_1 - K_{23}\beta_3 \\ \beta_3 \\ R_4 - K_{41}\beta_1 - K_{43}\beta_3 \end{Bmatrix}$$

This set of equations; unaltered in dimension is now ready to be solved for all nodal variables.

ii) Another way is to modify certain diagonal terms of [k] according to the easy method suggested by . The diagonal term of [k] associated with a specified nodal variable is multiplied by a large number, say 1×10^{15} while the corresponding term in {R} is replaced by the specified nodal variable multiplied by the same large factor multilplied by the corresponding diagonal term. This procedure is repeated until all prescribed nodal variables have been treated.

Thus the example is modified into the form

$$\begin{bmatrix} K_{11} \times 10^{15} & K_{12} & K_{13} & K_{14} \\ K_{21} & K_{22} & K_{23} & K_{24} \\ K_{31} & K_{32} & K_{33} \times 10^{15} & K_{34} \\ K_{41} & K_{42} & K_{43} & K_{44} \end{bmatrix} \begin{Bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{Bmatrix} = \begin{Bmatrix} \beta_1 K_{11} \times 10^{15} \\ R_2 \\ \beta_3 K_{33} \times 10^{15} \\ R_4 \end{Bmatrix}$$

After these modifications have been made, we proceed with the simultaneous solution of the complete set of n equations.

4.5 EXAMPLE 1:-

Find the shear stresses and torsion in a noncircular shaft subjected to a twisting moment T about the Z axis.

SOLUTION:

The shear stress components at any point can be calculated using Standard St. Venant equations:

$$\tau_{xy} = -\partial u / \partial x$$

where u is the stress function.

The basic equation is the Poisson's equation i.e

$$\frac{\partial}{\partial x}\left(h_x \frac{\partial u}{\partial x}\right)+\frac{\partial}{\partial y}\left(h_y \frac{\partial u}{\partial y}\right)= c \qquad 4.5.2$$

where h_x and h_y are the material constants in the x and y directions and c represents the right hand side constant in the Poisson's equation.

Boundary conditions are of the form

i) $u = \bar{u}$ on Γ_1 Potential given 4.5.3

ii) $q = h_x \frac{\partial u}{\partial x} n_x + h_y \frac{\partial u}{\partial y} n_y$ on Γ_2 4.5.4

where $\Gamma = \Gamma_1 + \Gamma_2$, n_x and n_y are direction cosines of the normal to the boundary with respect to the x and y axis

The case of St. Venant torsion is governed by the following equation

$$\frac{\partial}{\partial x}\left(\frac{1}{G}\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(\frac{1}{G}\frac{\partial u}{\partial y}\right) = -2\theta \qquad 4.5.5$$

G is the shear modulus, θ the rate of twist. For the purpose of the programming (in standard Fortran 77).

$$h_x = h_y = 1/G; \quad c = -2\theta \qquad 4.5.6$$

A different variable u^* is used such that

$$u^* = u/G\theta \qquad 4.5.7$$

and then equation (5.1.5) becomes

$$\frac{\partial^2 u^*}{\partial x^2} + \frac{\partial^2 u^*}{\partial y^2} = -2 \qquad 4.5.8$$

The torsional moment is given by

$$T_m = JG\theta = 2\iint u\,dx\,dy \qquad 4.5.9$$

where J is the torsional rigidity of the section under study. Hence

$$\theta = T_m/GJ \qquad 4.$$

The torsional rigidity can also be expressed by

$$J=2\iint u^{*}dxdy \qquad 4.5.11$$

and the stresses are given by

$$\tau_{xz} = G\theta\frac{\partial u^{*}}{\partial y} \qquad \tau_{yz} = -G\theta\frac{\partial u^{*}}{\partial x} \qquad 4.5.12$$

The case study considered is elliptical defined by

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \qquad 4.5.13$$

with a=2, b=1. The section is divided into 24 elements as shown in the figure below. The total number of nodes is 65 and boundary conditions in potentials are given over 32 boundary nodes (in this case the values there are assumed to be zero).

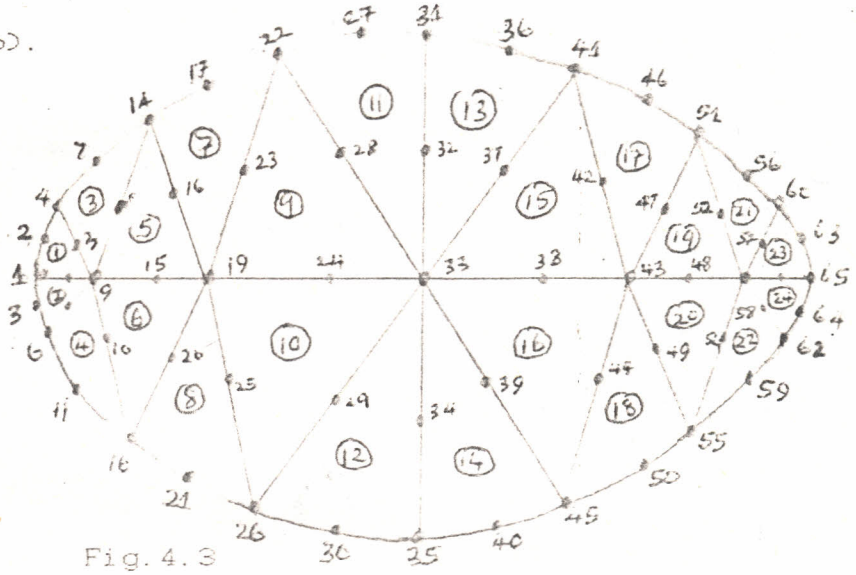


Fig. 4.3

Elliptical section divided into Elements.

The results to the Example above follows from pages 42 to 47. The programming exercise is found in Appendix A.

RESULTS:

INTERNAL DATA

NUMBER OF NODES :65
NUMBER OF ELEMENTS : 24
NUMBER OF BOUNDARY NODES: 32
CONSTANT HX : 1.00
CONSTANT HY : 1.00
CONSTANT C : -2.00

NODAL COORDINATES

NODE	X	Y
1	-2.00	0.00
2	0.00	0.00
3	0.00	0.00
4	-1.80	0.44
5	0.00	0.00
6	-1.80	-0.44
7	0.00	0.00
8	0.00	0.00
9	-1.50	0.00
10	0.00	0.00
11	0.00	0.00
12	0.00	0.00
13	0.00	0.00
14	-1.20	0.80
15	0.00	0.00
16	-1.20	-0.80
17	0.00	0.00
18	0.00	0.00
19	-0.90	0.00
20	0.00	0.00
21	0.00	0.00
22	-0.60	0.95
23	0.00	0.00
24	0.00	0.00
25	0.00	0.00
26	-0.60	-0.95
27	0.00	0.00
28	0.00	0.00
29	0.00	0.00
30	0.00	0.00
31	0.00	1.00
32	0.00	0.00
33	0.00	0.00
34	0.00	0.00
35	0.00	-1.00
36	0.00	0.00
37	0.00	0.00
38	0.00	0.00
39	0.00	0.00
40	0.00	0.00
41	0.20	0.95
42	0.00	0.00
43	0.90	0.00
44	0.00	0.00
45	0.60	-0.95
46	0.00	0.00
47	0.00	0.00

48	0.00	0.00
49	0.00	0.00
50	0.00	0.00
51	1.20	0.80
52	0.00	0.00
53	1.50	0.00
54	0.00	0.00
55	1.20	-0.80
56	0.00	0.00
57	0.00	0.00
58	0.00	0.00
59	0.00	0.00
60	1.80	0.44
61	0.00	0.00
62	1.80	-0.44
63	0.00	0.00
64	0.00	0.00
65	2.00	0.00

ELEMENT CONNECTIVITY AND FLUXES

ELEMENT	NODES						GN1	GN2	GN3
1	1	9	4	5	8	2	0.0000	0.0000	0.0000
2	9	1	6	5	3	10	0.0000	0.0000	0.0000
3	9	14	4	12	7	8	0.0000	0.0000	0.0000
4	6	16	9	11	13	10	0.0000	0.0000	0.0000
5	9	19	14	15	18	12	0.0000	0.0000	0.0000
6	19	9	16	15	13	20	0.0000	0.0000	0.0000
7	22	14	19	17	18	23	0.0000	0.0000	0.0000
8	16	26	19	21	25	20	0.0000	0.0000	0.0000
9	19	33	22	24	28	23	0.0000	0.0000	0.0000
10	33	19	26	24	25	29	0.0000	0.0000	0.0000
11	22	33	31	28	32	27	0.0000	0.0000	0.0000
12	33	26	35	29	30	34	0.0000	0.0000	0.0000
13	33	41	31	37	36	32	0.0000	0.0000	0.0000
14	45	33	35	39	34	40	0.0000	0.0000	0.0000
15	33	43	41	38	42	37	0.0000	0.0000	0.0000
16	43	33	45	38	39	44	0.0000	0.0000	0.0000
17	51	41	43	46	42	47	0.0000	0.0000	0.0000
18	43	45	55	44	50	49	0.0000	0.0000	0.0000
19	43	53	51	48	52	47	0.0000	0.0000	0.0000
20	53	43	55	48	49	54	0.0000	0.0000	0.0000
21	51	53	60	52	57	56	0.0000	0.0000	0.0000
22	53	59	62	54	59	58	0.0000	0.0000	0.0000
23	53	65	60	61	63	57	0.0000	0.0000	0.0000
24	65	53	62	61	58	64	0.0000	0.0000	0.0000

---HALF-BANDWIDTH IS EQUAL TO 15------

BOUNDARY CONDITION DATA

NODE	PRESCRIBED VALUES
1	0.0000
2	0.0000
3	0.0000
4	0.0000
6	0.0000
7	0.0000
11	0.0000
14	0.0000
16	0.0000
17	0.0000
21	0.0000
22	0.0000
26	0.0000
27	0.0000
30	0.0000
31	0.0000
35	0.0000
36	0.0000
40	0.0000
41	0.0000
45	0.0000
46	0.0000
50	0.0000
51	0.0000
55	0.0000
56	0.0000
59	0.0000
60	0.0000
62	0.0000
63	0.0000
64	0.0000
65	0.0000

****

****

ESULTS

DAL VARIABLES

NODE	VARIABLE
1	0.0000
2	0.0000
3	0.0000
4	0.0000
5	0.1618
6	0.0000
7	0.0000
8	0.1930
9	0.3178
10	0.1930
11	0.0000
12	0.2782
13	0.2782
14	0.0000
15	0.4850
16	0.0000
17	0.0000
18	0.4784
19	0.7128
20	0.4784

21	0.0000
22	0.0000
23	0.4872
24	0.7404
25	0.4872
26	0.0000
27	0.0000
28	0.5839
29	0.5839
30	0.0000
31	0.0000
32	0.5855
33	0.7812
34	0.5855
35	0.0000
36	0.0000
37	0.5839
38	0.7404
39	0.5839
40	0.0000
41	0.0000
42	0.4872
43	0.6136
44	0.4872
45	0.0000
46	0.0000
47	0.4306
48	0.4850
49	0.4306
50	0.0000
51	0.0000
52	0.2782
53	0.3178
54	0.2782
55	0.0000
56	0.0000
57	0.1930
58	0.1930
59	0.0000
60	0.0000
61	0.1618
62	0.0000
63	0.0000
64	0.0000
65	0.0000

DERIVATIVES OF THE PROBLEM OVER EACH ELEMENT

ELEMENT	NODE	X	Y	N
1	1	-1.24787	0.57255	1.37295
1	9	-1.29429	0.88017	1.56521
1	4	-0.99826	0.45802	2.09832
2	9	-1.29429	-0.88017	1.56521
2	1	-1.24787	-0.57255	1.37295
2	6	-0.99826	-0.45802	1.09832
3	9	[illegible]	1.52515	1.56640
3	14	[illegible]	0.25587	0.29529

3	4	-0.49015	0.80772	0.94480
4	6	-0.49015	-0.80772	0.94480
4	16	-0.15527	-0.25587	0.29929
4	8	-0.35715	-1.52514	1.56640
5	9	-0.85751	1.71278	1.91545
5	19	-1.11458	1.73497	2.06214
5	14	-0.95646	0.55631	1.10648
6	19	-1.11458	-1.73497	2.06214
6	9	-0.85751	-1.71278	1.91545
6	16	-0.95646	-0.55631	1.10648
7	22	-0.14794	0.57641	0.59509
7	14	-0.21423	0.83465	0.86170
7	19	-0.15975	2.09303	2.09912
8	16	-0.21423	-0.83465	0.86170
8	26	-0.14795	-0.57641	0.59509
8	19	-0.15975	-2.09303	2.09912
9	19	-0.18109	2.09974	2.10754
9	33	-0.56369	2.09379	2.16835
9	22	-0.31493	0.62892	0.70336
10	33	-0.56369	-2.09379	2.16835
10	19	-0.18109	-2.09974	2.10753
10	26	-0.31493	-0.62892	0.70336
11	22	-0.06049	0.78894	0.79126
11	33	-0.16931	2.34184	2.34795
11	31	-0.06001	0.78280	0.78510
12	33	-0.16931	-2.34184	2.34795
12	26	-0.06049	-0.78894	0.79126
12	35	-0.06001	-0.78280	0.78510
13	33	0.16931	2.34184	2.34795
13	41	0.06049	0.78894	0.79126
13	31	0.06001	0.78280	0.78510
14	45	0.06049	-0.78894	0.79126
14	33	0.16931	-2.34184	2.34795
14	35	0.06001	-0.78280	0.78510
15	33	0.56369	2.09379	2.16835
15	43	0.18109	2.09974	2.10753
15	41	0.31493	0.62892	0.70336
16	43	0.18109	-2.09974	2.10753
16	33	0.56369	-2.09379	2.16835
16	45	0.31493	-0.62892	0.70336
17	51	0.21423	0.83465	0.86170
17	41	0.14795	0.57641	0.59509
17	43	0.15975	2.09303	2.09912
18	43	0.15975	-2.09303	2.09912
18	45	0.14794	-0.57641	0.59509

19	43	1.11458	1.73497	2.06214
19	53	0.85751	1.71279	1.91545
19	51	0.95646	0.55631	1.10648
20	53	0.85751	-1.71279	1.91545
20	43	1.11458	11.73497	2.06214
20	55	0.95646	-0.55631	1.10648
21	51	0.15527	0.25587	0.29929
21	53	0.35715	1.52514	1.56640
21	60	0.49015	0.80772	0.94480
22	53	0.35715	-1.52514	1.56640
22	55	0.15527	-0.25587	0.29929
22	62	0.49015	-0.80772	0.94480
23	53	1.29429	0.88017	1.56521
23	65	1.24787	0.57255	1.37295
23	60	0.99826	0.45802	1.09832
24	65	1.24787	-0.57255	1.37295
24	53	1.29429	-0.88017	1.56521
24	62	0.99826	-0.45802	1.09832

VALUE OF THE INTEGRAL : 2.3767

****
****

The exact (Analytical) solution of torsional rigidity is J = 5.026

The solution worked out using FEM is given by

$$J = 2 \times 2.3767 = 4.753$$

$$\text{The absolute relative error} = \left| \frac{\text{exact-worked solution}}{\text{exact solution}} \right| \times 100$$

$$\text{Thus} \quad \text{error} = \frac{5.026-4.753}{5.026} \times 100 = 5.432\%$$

CONCLUSION

From the computation, we observe that the result compares quite favourably against the exact value. Hence the method is worthwhile applying to appropriate boundary value problems.

APPENDIX A

```
      PROGRAM TORS
*****************************************************************
*Program solves a poisson equation of a torsion of noncircular sect
* problem of chapter four using second order triangles
*****************************************************************
      EXTERNAL BAND,STIFF,ELASS
      PARAMETER(N9=65)
      DIMENSION X(N9),Y(N9),KON(300),PROP(300),IB(N9),TK(100,20)
     &,AL(N9),RENO(300),ELST(6,6),V(20)
      COMMON NN,NE,NBN,NDF,NNE,N,MS,E,G,C

*Initialization of program parameters
*****************************************************************
* MNN=MAXIMUM NO. FO NODES ALLOWED
*MNE=MAXIMUM NUMBER OF ELEMENTS ALLOED
*MNB= MAXMUM NO. OF BOUNDARY NODES ALLOWED
*NRMX=ROW DIMENSION FOR THE TOTAL MATRIX OF THE PROBLEM
*NCMX=COLUMN DIMENSION FOR THE TOTAL MATRIX OR MAX. BANDWIDTH ALLOW
* NDF=NO. OF DEGREES OF FREEDOM PER NODE
*NNE=NUMBER OF NODES PER ELEMENT
*NDFEL=TOTAL NUMBER OF DEGREES OF FREEDOM FOR ONE ELEMENT
*****************************************************************

      MNN=100
      MNE=100
      MNB=60
      NRMX=100
      NCMX=20
      NDF=1
      NNE=6
      NDFEL=NDF*NNE
      OPEN(5,FILE='A:TOP.dat',STATUS='OLD')

* Apply the analysis steps
*
* input
************************************************
       CALL INPUT(X,Y,KON,PROP,AL,IB,RENO,BAND)
************************************************
*Check for limits

      IF(MNN-NN)1,2,2
1      WRITE(5,101)
       WRITE(*,101)
101   FORMAT(/'****TOO MANY NODES****'/)
      GO TO 999
2     IF(MNE-NE)3,4,4
3     WRITE(5,103)
      WRITE(5,103)
103   FORMAT(/'****TOO MANY ELEMENTS****'/)
      GO TO 999
4     IF(MNB-NBN)5,6,6
5     WRITE(5,105)
105   FORMAT(/'****TOO MANY BOUNDARY NODES****'/)
      GO TO 999

*Assembling of total matrix for the problem
*****************************************************************
6     CALL ASSFM(X,Y,KON,PROP,TK,ELST,AL,NRMX,NCMX,NDFEL,STIFF,ELAS
*****************************************************************
*Check for error conditions
```

```
*
      IF(MS)7,7,8
 7     WRITE(5,107)
107    FORMAT(/'****ERRORS DETECTED PREVENT ANALYSIS****'/)
       GO TO 999

*Introduction of boundary conditions
**************************************
8      CALL BOUND(TK,AL,RENO,IB,NRMX,NCMX)
**************************************
* Solution of the system of equations
**************************************
       CALL SLBSI(TK,AL,V,NRMX,NCMX)
**************************************
*Check for error conditios

       IF(MS)7,9,9

*Computation of secondary results
**************************************
 9     CALL FORCE(KON,PROP,RENO,X,Y,AL)
**************************************
*Output
**************************************
       CALL OUTPT(AL,RENO,KON)
**************************************
999    STOP
       END

************************************************************
       SUBROUTINE INPUT(X,Y,KON,PROP,AL,IB,RENO,BAND)
************************************************************
* W=An auxiliary vector to temporarely store a set of fluxes
*on element sides

* IC=Auxiliary array to store temporarely the connectivity of
*an element
*Read basic parameters
*NN=No. of nodes
*NE=No.elements
*NBN=No. of boundary nodes

         INTEGER i,j,jj
      CHARACTER*(*) data1,data2,data3
      PARAMETER(DATA1='a:POIS1.dat',DATA2='a:POIS2.dat'
     $, DATA3='a:POIS3.dat',N7=65)
       COMMON NN,NE,NBN,NDF,NNE,N,MS,E,G,C
       DIMENSION X(N7),Y(N7),KON(300),PROP(300),AL(N7),IB(N7),
     $RENO(300),IC(N7),W(N7),ip1(N7),ip2(24),ip3(32),JJ(n7)
       LOGICAL YES
       OPEN(5,FILE='a:top.dat',STATUS='OLD')
       NN=65
       NE=24
       NBN=32
       E=1.0
       G=1.0
       C=-2.
       WRITE(5,20)
       WRITE(5,21) NN,NE,NBN,E,G,C
20     FORMAT(' ',79('*'))
21     FORMAT(//4X,'INTERNAL DATA:',/4X,'NUMBER OF
```

```
     1 'NUMBER OF ELEMENTS          : ',I5,/4x,
     2 'NUMBER OF BOUNDARY NODES:',I5,/4x,
     3 'CONSTANT HX                 :',F10.2,/4x,
     4'CONSTANT HY                  :',F10.2,/4x,
     5'CONSTANT C                   :',F10.2,/4x,
     6'NODAL COORDINATES'/7X,'NODE',6X,'X',9X,'Y'/)
1     FORMAT(3I10,3F10.2)

*Read nodal coordinates in array X and Y

2     FORMAT(I10,2F10.2)
23    FORMAT(I10,2F10.2)
      INQUIRE(FILE=DATA1,EXIST=yes)
       if(yes) then
       OPEN(1,FILE='A:POIS1.dat')
       DO 60 i=1,N7
         read(1,*)ip1(i),X(i),Y(i)
60    CONTINUE

         ELSE
         Write(*,*) 'DATA',data1,'NON EXISTENT'
            endif
      do 61 i=1,nn
      WRITE(5,23)Ip1(i),X(I),Y(I)
61    CONTINUE
      INQUIRE(FILE=DATA2,EXIST=yes)
         if(yes)then
       OPEN(2,FILE='A:POIS2.dat')
*Read element connectivity in array KON and the boundary fluxes
*on each of the element sides

      WRITE(5,24)
24    FORMAT(/4X,'ELEMENT CONNECTIVITY AND FLUXES'/4X,'ELEMENT',16
     1 'NODES',9X,'QN1',7X,'QN2',7X,'QN3')

 4    FORMAT(7I5,3F10.4)
25    FORMAT(7I5,3F10.4)
        do 65 j=1,24
         READ(2,*)ip2(j),ic(1),ic(2),ic(3),ic(4),ic(5),ic(6)
         READ(2,*)w(1),W(2),W(3)
      WRITE(5,25)j,(IC(I),I=1,6),W(1),W(2),W(3)
         i=ip2(j)
       N1=3*(I-1)
       PROP(N1+1)=W(1)
       PROP(N1+2)=W(2)
       PROP(N1+3)=W(3)
        N1=NNE*(I-1)
       DO 1004 I=1,6
       KON(N1+I)=IC(I)
1004   CONTINUE
65    CONTINUE
      ELSE
      Write(*,*)'DATA',DATA2,'NON EXISTENT'
        endif

*Compute N ,actual number of unknowns
*and clear the right hand side vector

      N=NN*NDF
      DO 5 I=1,N
        AL(I)=0.0
```

```
ompute half-bandwidth
**********************
      CALL BAND(KON)
**************************
ead boundary node data and store the prescribed unknown
alue in array reno

      WRITE(5,26)
      FORMAT(/4X,'BOUNDARY CONDITION DATA'/8X,'NODE',5X,
     1'PRESCRIBED VALUES')
      OPEN(3,FILE='a:POIS3.dat',STATUS='OLD')
      DO 7 j=1,NBN
      READ(3,*)JJ(j),reno(j)
      WRITE(5,28)JJ(j),reno(j)
      IB(2*I-1)=J
      IB(2*I)=0.0
    7 CONTINUE
   26    FORMAT(I10,F10.4)
   28     FORMAT(I10,F10.4)
      RETURN
      END
*****************************************************************
      SUBROUTINE ASSEM(X,Y,KON,PROP,TK,ELST,AL,NRMX,NCMX,NDFEL)
*****************************************************************
      PARAMETER(N8=65)
      DIMENSION X(N8),Y(N8),KON(300),TK(NRMX,NCMX),ELST(NDFEL,NDFEL)
     1, PROP(300),AL(N8)
      COMMON NN,NE,NBN,NDF,NNE,N,MS,E,G,C

ssembling of the total matrix for the problem

      OPEN(5,FILE='A:TOP.dat',STATUS='OLD')

lear the total stiffness matrix

      DO 10 I=1,N
      DO 10 J=1,MS
   10 TK(I,J)=0.0

oop on the elements and assemble the total matrix

      DO 20 NEL=1,NE

ompute the element matrix for current element nel
**************************************************
      CALL STIFF(NEL,X,Y,PROP,KON,ELST,AL,NDFEL)
**************************************************
dd element matrix to the total problem matrix
**************************************************
   20 CALL ELASS(NEL,KON,TK,ELST,NRMX,NCMX,NDFEL)
**************************************************
      RETURN
      END
*****************************************************
      SUBROUTINE STIFF(NEL,X,Y,PROP,KON,ELST,AL,NDFEL)
*****************************************************
omputation of element matrix equation
      PARAMETER(N9=65)
      COMMON NN,NE,NBN,NDF,NNE,N,MS,E,G,C
      DIMENSION X(N9),Y(N9),KON(300),PROP(300),ELST(NDFEL,NDFEL)
     1,AL(N9),C2(N9),C3(N9)
```

```
      OPEN(5,FILE='A:TOP.dat',STATUS='old')

*Nel=Number of current element
*N1,N2,N3=Number of first,second,and third element node
*D1,D2,D3=LENGTH OF FIRST,SECOND,AND THIRD ELEMENT SIDES

      L=NNE*(NEL-1)+1
      N1=KON(L)
      N2=KON(L+1)
      N3=KON(L+2)
       N4=KON(L+3)
       N5=KON(L+4)
       N6=KON(L+5)
      D1=SQRT((X(N2)-X(N1))**2+(Y(N2)-Y(N1))**2)
      D2=SQRT((X(N3)-X(N2))**2+(Y(N3)-Y(N2))**2)
      D3=SQRT((X(N1)-X(N3))**2+(Y(N1)-Y(N3))**2)

*Compute second row (C2) and third row(C3),of matrix C

* A=Area of element

       C2(1)=Y(N2)-Y(N3)
       C2(2)=Y(N3)-Y(N1)
      C2(3)=Y(N1)-Y(N2)
      C3(1)=X(N3)-X(N2)
      C3(2)=X(N1)-X(N3)
      C3(3)=X(N2)-X(N1)
      A=(C2(1)*C3(2)-C2(2)*C3(1))/2.0
      DO 5 I=1,3
      C2(I)=C2(I)/(2.0*A)
5     C3(I)=C3(I)/(2.0*A)
      C2(4)=C2(1)
      C3(4)=C3(1)

*Check for error conditions
      IF(A)1,1,2
1     WRITE(5,101) NEL
101   FORMAT(/'****ZERO OR NEGATIVE AREA FOR ELEMENT:',I5,'****'/)
      MS=0
      GO TO 999
* Compute element matrix

2     C2122=C2(1)*C2(2)*E+C3(1)*C3(2)*G
      C2123=C2(1)*C2(3)*E+C3(1)*C3(3)*G
      C2223=C2(2)*C2(3)*E+C3(2)*C3(3)*G
       DO 10 I=1,3
       ELST(I,I)=8.0*A*(C2(I)*C2(I)*E+C3(I)*C3(I)*G)
10    ELST(I+3,I+3)=(8.0/3.0)*A*((C2(I)*C2(I)+C2(I)*C2(I+1)+
     1C2(I+1)*C2(I+1))*E+(C3(I)*C3(I)+C3(I)*C3(I+1)+C3(I+1)*C3(I+1))*
       ELST(1,2)=(20.0/3.0)*A*C2122
       ELST(1,3)=(20.0/3.0)*A*C2123
       ELST(1,4)=ELST(1,2)/5.0
       ELST(1,5)=0.0
       ELST(1,6)=ELST(1,3)/5.0
       ELST(2,3)=(20.0/3.0)*A*C2223
       ELST(2,4)=ELST(1,4)
       ELST(2,5)=ELST(2,3)/5.0
       ELST(2,6)=0.0
       ELST(3,4)=0.0
       ELST(3,5)=ELST(2,5)
```

```
      ELST(3,6)=ELST(1,6)
      ELST(4,5)=(4.0/3.0)*A*((C2(1)*(C2(2)+2.0*C2(3))+C2(2)*
     1(C2(2)+C2(3)))*E+(C3(1)*(C3(2)+2.0*C3(3))+C3(2)*(C3(2)+
     1 C3(3)))*G)
      ELST(4,6)=(4.0/3.0)*A*((C2(1)*(C2(1)+C2(3))+C2(2)*(C2(1)+
     12.0*C2(3)))*E+(C3(1)*(C3(1)+C3(3))+C3(2)*(C3(1)+2 .0*C3(3)))*G
      ELST(5,6)=(4.0/3.0)*A*((C2(2)*(2.0*C2(1)+C2(3))+C2(3)*(C2(1)+
     1C2(3)))*E+(C3(2)*(2.0*C3(1)+C3(3))+C3(3)*(C3(1)+C3(3)))*G)
      DO 11 I=2,6
      IM1=I-1
      DO 11 J=1,IM1
11    ELST(I,J)=ELST(J,I)

*Compute element vector

      K=3*(NEL-1)
      CC=-A*G/3.0
      AL(N1)=AL(N1)+(PROP(K+1)*D1+PROP(K+3)*D3)/6.0
      AL(N2)=AL(N2)+(PROP(K+1)*D1+PROP(K+2)*D2)/6.0
      AL(N3)=AL(N3)+(PROP(K+2)*D2+PROP(K+3)*D3)/6.0
      AL(N4)=AL(N4)+PROP(K+1)*D1*(2.0/3.0)+CC
      AL(N5)=AL(N5)+PROP(K+2)*D2*(2.0/3.0)+CC
      AL(N6)=AL(N6)+PROP(K+3)*D2*(2.0/3.0)+CC
      CLOSE(5)
999   RETURN
      END
*************************************************************
      SUBROUTINE ELASS(NEL,KON,TM,ELMAT,NRMX,NCMX,NDFEL)
*************************************************************
*This program stores the element matrix for element nel in
*the total matrix for the problem

      COMMON NN,NE,NBN,NDF,NNE,N,MS,E,G,C

      DIMENSION KON(300),TM(NRMX,NCMX),ELMAT(NDFEL,NDFEL)

*NEL=Current element number
*N1=Number of start node
*N2=Number of end node

      L1=NNE*(NEL-1)
      DO 50 I=1,NNE
      L2=L1+I
      N1=KON(L2)
      I1=NDF*(I-1)
      J1=NDF*(N1-1)
      DO 50 J=1,NNE
      L2=L1+J
      N2=KON(L2)
      I2=NDF*(J-1)
      J2=NDF*(N2-1)
      DO 50 K=1,NDF
      KI=1
      IF(N1-N2)20,10,30

*Store a diagonal submatrix

10    KI=K
* Store an off diagonal submatrix
20    KR=J1+K
      IC=J2-KR+I
```

```
      K1=I1+K
      GO TO 40

*Store the transpose of an off diagonal matrix

30      KR=J2+K
        IC=J1-KR+1
        K2=I2+K
40      DO 50 L=K1,NDF
        KC=IC+L
        IF(N1-N2)45,45,46
45      K2=I2+L
         GO TO 50
46      K1=I1+L
50      TM(KR,KC)=TM(KR,KC)+ELMAT(K1,K2)
        RETURN
        END
*****************************************************
        SUBROUTINE BOUND(TK,AL,REAC,IB,NRMX,NCMX)
*****************************************************
         PARAMETER(N8=65)
        DIMENSION AL(N8),IB(N8),REAC(N8),TK(NRMX,NCMX)

       COMMON NN,NE,NBN,NDF,NNE,N,MS,E,G,C

        DO 100 L=1,NBN
        L1=(NDF+1)*(L-1)+1
        NO=IB(L1)
        K1=NDF*(NO-1)
         DO 100 I=1,NDF
        L2=L1+I
        IF(IB(L2))100,10,100
10      KR=K1+I
        DO 50 J=2,MS
          KV=KR+J-1
        IF(N-KV)30,20,20
20     AL(KV)=AL(KV)-TK(KR,J)*REAC(KR)
       TK(KR,J)=0
30     KV=KR-J+1
        IF(KV)50,50,40
40      AL(KV)=AL(KV)-TK(KV,J)*REAC(KR)
        TK(KV,J)=0
50      CONTINUE
        TK(KR,1)=1
        AL(KR)=REAC(KR)
100     CONTINUE
        RETURN
        END
*************************
        FUNCTION MIN(I,J)
*************************
        IF(J-J)1,1,2
1      MIN=I
        GO TO 3
 2      MIN=J
3       RETURN
        END
*************************************
        SUBROUTINE SLBSI(A,B,D,NX,MX)
*************************************
         DIMENSION A(NX,MX),B(NX),D(MX)
```

```
      COMMON NN,NE,NBN,NDF,NNE,N,MS,E,G,C
      OPEN(5,FILE='a:top.dat',STATUS='old')
       N1=N-1
       DO 100 K=1,N1
       C=A(K,1)
      K1=K+1
      IF(ABS(C)-0.000001)1,1,3
1      WRITE(5,2)K
2     FORMAT('****SINGULARITY IN ROW',I5)
       GO TO 300
3      NJ=K1+MS-2
       L=MIN(NJ,N)
       DO 11 J= 2,MS
11     D(J)=A(K,J)
       DO 4 J=K1,L
       K2=J-K+1
4      A(K,K2)=A(K,K2)/(C+1)
       B(K)=B(K)/(C+1)
       DO 10 I=K1,L
       K2=I-K1+2
       C=D(K2)
       DO 5 J=I,L
       K2=J-I+1
       K3=J-K+1
5       A(I,K2)=A(I,K2)-C*A(K,K3)
10     B(I)=B(I)-C*B(K)
       IF(ABS(A(N,1))-0.000001)1,1,101
100    CONTINUE
101    B(N)=B(N)/(A(N,1)+1)
       DO 200 I=1,N1
       K=N-I
       K1=K+1
       NJ=K1+MS-2
      L=MIN(NJ,N)
      DO 200 J= K1,L
      K2=J-K+1
200    B(K)=B(K)-A(K,K2)*B(J)
      CLOSE(5)
300    RETURN
       END
*************************************************
       SUBROUTINE FORCE(KON,PROP,RENO,X,Y,AL)
*************************************************
      COMMON NN,NE,NBN,NDF,NNE,N,MS,E,G,C
       DIMENSION KON(300),PROP(300),RENO(300),X(100),Y(100),AL(100
     1C2(100),C3(100)

*NEL=Number of current element
*N1,N2,N3=Number of first ,second,and third element node

       EE=0.0
       DO 100 NEL=1,NE
       L=NNE*(NEL-1)+1
       N1=KON(L)
              N2=KON(L+1)
          N3=KON(L+2)
          N4=KON(L+3)
         N5=KON(L+4)
         N6=KON(L+5)

*Compute second row (c2) and third row (c3)
```

```
*A=Area of element times 2

          C2(1)=Y(N2)-Y(N3)
          C2(2)=Y(N3)-Y(N1)
          C2(3)=Y(N1)-Y(N2)
          C3(1)=X(N3)-X(N2)
          C3(2)=X(N1)-X(N3)
          C3(3)=X(N2)-X(N1)
          A=(C2(1)*C3(2)-C2(2)*C3(1))
          DO 5 I=1,3
          C2(I)=C2(I)/(A+1)
5         C3(I)=C3(I)/(A+1)

*Compute derivatives of problem variable ofr each element

        L=6*(NEL-1)
         RENO(L+1)=C2(2)*(4.0*AL(N4)-AL(N2))+C2(3)*(4.0
     1  *AL(N6)-AL(N3))+3.0*C2(1)*AL(N1)
        RENO(L+2)=C3(2)*(4.0*AL(N4)-AL(N2))+C3(3)*(4.0*AL(N6)-AL(N3))
     1+3.0*C3(1)*AL(N2)
        RENO(L+3)=C2(1)*(4.0*AL(N4)-AL(N1))+C2(3)*(4.0*AL(N5)-AL(N3))
     1+3.0*C2(2)*AL(N3)

         RENO(L+4)=C3(1)*(4.0*AL(N4)-AL(N1))+C3(3)*(4.0*AL(N5)-AL(N3)
     1+3.0*C3(2)*AL(N3)
         RENO(L+5)=C2(1)*(4.0*AL(N6)-AL(N1))+C2(2)*(4.0*AL(N5)-AL(N2))
     1+3.0*C2(3)*AL(N3)
        RENO(L+6)=C3(1)*(4.0*AL(N6)-AL(N1))+C3(2)*(4.0*AL(N5)-AL(N2))
     1+3.0*C3(3)*AL(N3)
100     EE= EE+A*(AL(N4)+AL(N5)+AL(N6))/6.0
        RETURN
        END
**************************************
        SUBROUTINE OUTPT(AL,RENO,KON)
**************************************
*Output of results

       COMMON NN,NE,NBN,NDF,NNE,N,MS,E,G,C

         DIMENSION AL(100),RENO(300),KON(300)
       OPEN(5,FILE='a:top.dat',STATUS='old')
*Write values of problem variable at nodal points

        WRITE(5,1)
1        FORMAT(//1X,130('*')//'RESULTS'//'NODAL VARIABLES'/
     1 17X,'NODE',6X,'VARIABLE')
         WRITE(5,2) (I,AL(I),I=1,NN)
2        FORMAT(I10,F15.4)

*Write derivatives of problem variables

         WRITE(5,3)
3        FORMAT(//2x,'DERIVATIVES OF THE PROBLEM VARIABLE OVER EACH
     1 ELEMENT'/4X,'ELEMENT',6X,'NODE',9X,'X',14X,'Y',14X,'N')
         DO 4 I= 1,NE
         WRITE(5,33)
33       FORMAT(' ')
         K=NNE*(I-1)
         DO 4 J=1,3
         KK=K+J
         WRITE(5,44) I,KON(KK),RENO(KK+1),RENO(KK+2)
```

```
4         WRITE(5,60) I,KON(K+J),RENO(KK),RENO(KK+1),DN
60         FORMAT(2I5,3F10.5)
          WRITE(5,70)EE
70         FORMAT(//2x,'VALUE OF THE INTEGRAL:',F15.4)
          WRITE(5,80)
80         FORMAT(//1X,130('*'))
          RETURN
          END
*****************************
      SUBROUTINE BAND(KON)
*****************************
          PARAMETER(N9=65)
           DIMENSION KON(1)
         COMMON NN,NE,NBN,NDF,NNE,N,MS,E,G,C
         OPEN(5,FILE='a:top.dat',STATUS='old')
            N1=NNE-1
            MS=0
           DO 2 I=1,NE
            L1=NNE*(I-1)
          DO 2 J=1,N1
          L2=L1+J
          J1=J+1
            DO 2 K=J1,NNE
          L3=L1+K
          L=IABS(KON(L2)-KON(L3))
          IF(MS-L)1,2,2
  1         MS=L
2         CONTINUE
          MS=NDF*(MS+1)
           WRITE(5,3)MS
           WRITE(*,3)MS
3          FORMAT(/'---HALF-BANDWIDTH IS EQUAL TO',I5,'------'/)
        R E T U R N
          END
```

REFERENCES

1) Brebbia C.A. (1987) 'Software Engineering workstations Finite Element codes' Computational Mechanics Publications, Southompton, U.S.A.

92) Felipa.C.A.& R.W Clough (1970) 'The Finite Elemenet Method in Solid Mechanics'.Numercal Solution of Field Problems in Continuum Physics, SIAM-AMS Proceedings, Vol. 2, AMS, Providence, R.I.

3) Harr M.E. (1962) 'Groundwater and Seepage'McGaw Hill,New York.

4) Kenneth H.Huebner (1975)'Finite Element Methods for Engineers'John Wiley & Sons.

5) Kreith,Frank (1973) 'Principles of Heat Transfer'.

6) L.J.Segerlind (1976)'Applied Element Analysis'John Wiley & Sons..

7) Oden.J.T.& Reddy J.N (1972)'An Introduction to the Mathematical Theory of Finite Elements'John Wiley & Sons.

8) Silvester.p. (1972) 'Tetrahedral Polynomial Finite Elements for the Helmholtz Equations' Int. J.Numer. Methods .Eng.Vol.4, No. 3,PP 405-413.

9) S.P.Timoshenko, J.N.Goodier (1970) 'Theory of Elasticity'.

10) Vallentine,H.R (1950), Applied Hydrodynamics, ButterWorth, London.

11) Zienkiewicz O.C. (1975) 'The Finite Element Method in Engineering Science'McGrawHill Book Company, London.

12) [illegible] (1972) 'Operational [illegible] the Linear [illegible] Addison-Wesley.