FINITE ELEMENT SOLUTION FOR ELLIPTIC PARTIAL DIFFERENTIAL EQUATIONS

BY

KHALID ABD AL-RHMAN HOSSEN, B.Sc.,M.Sc.

A Doctoral Thesis

Submitted in partial fulfilment of the requirements

for the award of Doctor of Philosophy

of the Loughborough University of Technology

1984

Supervisor: Professor D.J. Evans, Ph.D.,D.Sc.

Department of Computer Studies

ACKNOWLEDGEMENTS

I wish to express my sincere gratitude to my supervisor, Professor D.J. Evans, for his considerable guidance, advice and willingness to assist and advise at any time throughout the programme of this work.

I would like also to thank the University of Mosul, Iraq, for giving me the opportunity to continue my studies, and for their financial support.

A sincere gratitude and thanks to my parents, for their patience and moral encouragement.

Finally, my thanks are due to Miss Judith M. Briers for her excellent typing.

ABSTRACT

The contents of this thesis are a detailed study of the implementation of Finite Element method for solving linear and non-linear elliptic partial differential equations. It commences with a description and classification of partial differential equations, the related matrix and eigenvalue theory and the related matrix methods to solve the linear and non-linear systems of equations.

In Chapter Three, we discuss the development of the finite element method and its application with a full description of an orderly step-by-step process. In Chapter Four, we discuss the implementation of developing an efficient easy-to-use finite element program for the general two-dimensional problem along with the capability of handling problems for different domains and boundary conditions and with a fully automated mesh generation and refinement technique along with a description of generalised pre- and post-processors for the Finite Element Method. In Chapter Five, we consider the solution of a free boundary problem whose boundary position is initially unknown and must be determined as part of the solution to the problem, i.e. a sluice gate flow problem is considered.

In Chapter Six, we consider the finite element method for the numerical solution of a class of two-dimensional elliptic boundary value problems which contain boundary singularities and where a number of different strategies are also considered. The numerical results compare favourably with those obtained by other techniques.

Chapters Seven and Eight present the results obtained when solving a useful population of complex linear and non-linear partial differential

problems by the finite element method using different order polynomials basis function such as quadratic, cubic and quartic. The results of different solution plots are presented as output.

The thesis concludes with some general conclusions and recommendations for further study.

CONTENTS

CHAPTER ONE

INTRODUCTION

1.1 INTRODUCTION

It is not possible to identify the exact starting point of the finite element method, because the method makes use of many theories and techniques drawn from mathematics and continuum mechanics, and no single view of its origins can cover all facts of the development process. Moreover, as more individuals and organizations began working with this method, the advances become increasingly more diffuse. However one of the early developments of the finite element method started in the middle of the twentieth century (some thirty years ago), with the analysis of aircraft structural engineering problems and over the years the finite element technique has been so well established that today it is considered to be one of the powerful methods for solving a wide variety of practical problems efficiently. In fact the method has become one of the active research areas for applied mathematics and engineers in which the development has reached the stage where there are very few problems which the method cannot tackle.

Various types of boundary conditions, curved boundaries or complex geometries present no great difficulties for the method, and there are further techniques for dealing with problems which have crack, singularities, and many more difficult problems.

Often this flexibility and the general applicability of this method is a great advantage over various other numerical techniques of solving problems.

Today the finite element method is considered to be one of the more

established and convenient analysis tools by applied scientists and engineers and with the help and the power of the computer the finite element method has much to contribute in applied research.

1.2 THE BASIC IDEAS

The finite element method (FEM) is a numerical discretisation technique for obtaining the approximate solution to problems mainly governed by partial or ordinary differential equations on specified domains. The given specified domain is divided into a finite number of small non-overlapping regions which are called elements. These elements are considered to be interconnected at specified joints which are called nodes or nodal points.

Generally, straight line segments are used for the one dimensional case, *triangles* or *rectangles* elements with algebraic curves as boundaries in the space of two dimensions and tetrahedrans or hexahedrons in the space of three dimensions.

In each element the solution is approximated by a simple function in the form of polynomials where parameters can be adjusted to ensure the existence of continuity of the functions in adjacent elements. Our attention will be devoted almost exclusively to two dimensional triangular elements in this thesis, primarily because arbitrary regions in two dimensions can be approximated by polygons, which can always be divided up into a finite number of triangles more easily than the other element shapes like rectangles. The approximate solution of the general problem by the finite element method always follows an orderly step by step process. These finite element analysis steps are:

(i) Discretization of the domain or solution region,

(ii) Selection of an interpolation model to represent the variation of the field variable,

(iii) Derivation of the discrete approximation of the problem consisting of a finite set of algebraic equations

(iv) Solution of the set of algebraic equations derived in step (iii) by an accurate method

(v) Display and interpret the results (post processing).

These various stages of the finite element method will be discussed later in Chapter 3.

1.3 REMARKS ON THE CLASSIFICATION OF PARTIAL DIFFERENTIAL EQUATIONS

The partial differential equations which arise in many practical problems are equations that express a relationship between an unknown function of several variables (two or more), and its partial derivatives.

The order of a partial differential equation is the order of the highest derivative contained in the equation.

A partial differential equation is *linear*, if it is of first degree in the unknown function and its derivatives, otherwise it is *non-linear*.

For example,

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0,$$

is a *linear* partial differential equation of second order, while the equation,

$$\frac{\partial^3 u^2}{\partial x^3} + 3\frac{\partial^2 u}{\partial x \partial y} + f(x,y,u) = 0 \ ,$$

is a *non-linear* partial differential equation of third order.

A convenient and frequently used method for classifying the basic partial differential equations that characterize field problems follows from a consideration of the mathematical character of the solutions. This method of classification is briefly outlined next to provide a link with a more formal mathematical treatment.

The majority of problems of practical importance are special cases of the general second order partial differential equation

$$Lu = 0 \ , \qquad (1.1)$$

where L is a differential operator defined by,

$$Lu \equiv A\frac{\partial^2 u}{\partial x^2} + 2B\frac{\partial^2 u}{\partial x \partial y} + C\frac{\partial^2 u}{\partial y^2} + D\frac{\partial u}{\partial x} + E\frac{\partial u}{\partial y} + Fu + G = 0 \ . \tag{1.2}$$

If G=0 the partial differential equation is termed as *homogeneous*, otherwise it is called *inhomogeneous*.

Equation (1.2) is classified as *elliptic*, *parabolic*, *hyperbolic*, when the discriminant B^2-4AC is *negative*, *zero* or *positive*, respectively. Because the coefficients A,B and C are, in general, functions of the independent variables x,y, the classification of an equation may change at different positions in space. However, if A,B and C are constants then the equation is of one type throughout the x,y plane.

Well known examples of the three types are:

Heat flow equation

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} \ , \tag{1.3}$$

which is of parabolic type.

Wave equation

$$\frac{\partial^2 u}{\partial t^2} = \frac{\partial^2 u}{\partial x^2} \ , \tag{1.4}$$

which is of hyperbolic type.

Laplace equation

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \ , \tag{1.5}$$

which is of elliptic type.

Boundary-value problems are naturally associated with elliptic equations, while initial-value problems and mixed (initial/boundary value) problems arise in connection with hyperbolic and parabolic differential

equations. The boundary conditions can be one of the following three types:

(i) Boundary-value problem of the first kind. Also called the *Dirichlet problem*. Here the function u(x,y) is prescribed along the boundary, i.e. u is given on the boundary ∂R. If the function takes zero values along the boundary, then the condition is called a homogeneous Dirichlet, otherwise it is an inhomogeneous Dirichlet condition.

(ii) Boundary-value problem of second kind. Often called the *Neumann problem*. Here the normal derivative of the function u(x,y) is specified along the boundary, i.e. $\frac{\partial u}{\partial n}$ given on the boundary. We may also have homogeneous or inhomogeneous Neumann boundary conditions as before.

(iii) Boundary-value problem of the third kind. Here the function u(x,y) and its normal derivatives are prescribed along the boundary i.e. u and $\frac{\partial u}{\partial n}$ are given along ∂R, we may also have homogeneous or inhomogeneous mixed boundary conditions. It is often the case that an elliptic problem is specified by boundary conditions that are of different kinds along different parts ∂R.

We assume throughout our discussion that our mathematical problem is *well posed*, i.e. if the solution exists, it is unique and depends continuously on the given data.

We would expect that small variations in the data should result in correspondingly small variations in the solution. If this does not turn out to be true, we would be inclined to believe that the mathematical model has been badly formulated.

CHAPTER TWO

BASIC LINEAR ALGEBRAIC THEORY AND APPROXIMATION METHODS FOR SOLVING P.D.E.S

2.1 INTRODUCTION

The numerical solution of partial differential equations by the finite element method or other numerical approaches like the finite difference method in all cases generates an associated algebraic problem.

In general this algebraic problem involves the solution of a large set of equations of the form,

$$\sum_{j=1}^{n} a_{ij}x_j = b_i \ , \quad (i=1,2,\ldots,n) \ , \qquad (2.1)$$

which may be written as the matrix system

$$A\underline{x} = \underline{b} \ , \qquad (2.2)$$

where the matrix A is usually square with real elements, and has n rows and columns and the elements a_{ij} $(i,j=1,2,\ldots,n)$ are real numbers. The vectors $\underline{x}$ and $\underline{b}$ have n components.

The usual solution of the problem (2.2) is to find $\underline{x}$ when A and $\underline{b}$ are given. A unique solution of equation (2.2) which may be written in the form $\underline{x} = A^{-1}\underline{b}$, exists for equation (2.2), when A is non-singular which is equivalent to A having a non-singular determinant. Since equation (2.2) is a matrix representation of the differential equation after applying the proper numerical approach, the matrix A is usually sparse (many of its elements are zero), and possesses a definite structure (determined by its non-zero elements).

The method of finding the solution for (2.2) particularly when the order n of the matrix A is large, depends very much on the structure of A.

In this Chapter, an introduction to matrix techniques that are useful

for the solution of (2.2) is given along with very important definitions and theorems associated with the theoretical developments of the finite element method. We will consider also several alternative algorithmic methods for the solution of a large system of equations. We give particular prominence to those methods applicable to the solution of equations arising from finite element calculations.

2.2 BASIC MATRIX ALGEBRA

A review of notation and properties for a square matrix A of order n with real elements, which is relevant to the solution of the equation (2.2) is now given.

2.2.1 USEFUL NOTATIONS

A	Square matrix of order n
a_{ij}	real number, which is the element in the ith row and jth column of the matrix A
A^T	transpose of A
A^{-1}	inverse of A
I	unit matrix of order n
O	null matrix
$\|A\|$	determinant of A
$\rho(A)$	spectral radius of A
$\underline{x}$	column vector with elements x_i, (i=1,2,...,n)
$\underline{x}^T$	row vector with elements x_j, (j=1,2,...,n)
$\|\|A\|\|$	norm of A
$\|\|\underline{x}\|\|$	the norm of $\underline{x}$
P	permutation matrix which has entries of zeros and ones only, with one non-zero entry in each row and column.

2.2.2 DEFINITIONS

The matrix A is:

"non-singular" if $|A| \neq 0$

"symmetric" if $A=A^T$

"orthogonal" if $A^{-1}=A^T$

"null" if $a_{ij}=0$ $(i,j=1,2,\ldots,n)$

"diagonally dominant" if $|a_{ii}| \geq \sum_{j\neq i} |a_{ij}|$ for all i

"tridiagonal" if $a_{ij}=0$ for $|i-j|>1$

"block diagonal" if

$$\begin{bmatrix} B_1 & & & O \\ & B_2 & & \\ & & \ddots & \\ & & & B_k \\ & & & & \ddots \\ O & & & & & B_s \end{bmatrix}$$

where each B_k, $(k=1,2,\ldots,s)$ is a square matrix, not necessarily of the same order.

"upper triangular" if $a_{ij}=0$, $i>j$

"lower triangular" if $a_{ij}=0$, $j>i$

"irreducible" if there exists no permutation transformation PAP^{-1}, which reduces A to the form

$$\begin{bmatrix} P & O \\ R & Q \end{bmatrix}$$

where P and Q are square submatrices of order p and q respectively (p+q=n) and O is a $(p\times q)$ null matrix.

"sparse" if most of the elements a_{ij} are zero

"dense" if most of the elements a_{ij} are non-zero.

"The conjugate" of a matrix A whose elements a_{ij} which are not necessarily real numbers will be denoted by $\bar{A}$.

"The Hermitian transpose" (or conjugate transpose) of a matrix A will be denoted by A^H.

The matrix A is said to be a 'Hermitian' matrix if,

$A^H = A$, i.e. if $\bar{a}_{j,i} = a_{i,j}$, for all i and j, $(i,j=1,2,\ldots,n)$.

The definition of a Hermitian matrix implies that the diagonal elements of the matrix are real.

A real symmetric matrix is always Hermitian, but a Hermitian matrix is symmetric only if it is real.

2.2.3 PARTITIONING OF A MATRIX

A matrix A can be partitioned into submatrices, for example,

$$A = \left[\begin{array}{ccc:cc} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ \hdashline a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \end{array}\right] , \qquad (2.3)$$

is shown partitioned into four submatrices by the dotted lines.

We may write,

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} , \qquad (2.4)$$

where $A_{11}, A_{12}, A_{21}, A_{22}$ themselves are submatrices. In performing any matrix operation, all the rules can first be applied as if each of the submatrices were scalar elements and then carrying out any further operation in the usual way. For example, if we have A as given above in (2.3) and,

$$B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \\ b_{41} & b_{42} \end{bmatrix} , \qquad (2.5)$$

we may write again,

$$B = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} , \qquad (2.6)$$

Then,

$$AB = \begin{bmatrix} A_{11}B_1 + A_{12}B_2 \\ A_{21}B_1 + A_{22}B_2 \end{bmatrix} , \qquad (2.7)$$

can be verified as representing the complete product by further multiplication. The essential feature of partitioning is that the size of subdivisions has to be such as to make the products of type $A_{11}B_1$ meaningful, i.e., the number of columns in A_{11} must be equal to the number of rows in B_1, etc. If the above definition holds, then all further operations can be conducted on partitioned matrices treating each portion as if it were a scalar. It should be noted that a matrix can be multiplied by a scalar (number) here, obviously, the requirement of equality of appropriate rows and columns no longer apply.

If a symmetric matrix is divided into an equal number of submatrices A_{ij} rows and columns then,

$$A_{ij} = A_{ji}^T . \qquad (2.8)$$

2.2.4 QUADRATIC FORMS

Matrix notation is most often employed to deal with sets of linear equations. It is also useful in symbolizing special nonlinear expressions called quadratic forms.

For a function of n variables $x_1, x_2, \ldots, x_n$, a quadratic form is defined as,

$$G(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} x_i x_j$$

$$= a_{11}x_1^2 + a_{12}x_1x_2 + \ldots + a_{1n}x_1x_n + \ldots + a_{21}x_2x_1$$

$$+ \ldots + a_{2n}x_2x_n + \ldots + a_{n1}x_nx_1 + \ldots + a_{nn}x_n^2 \qquad (2.9)$$

A quadratic form in one variable, say, is simply ax_1^2.

In two variablex x_1 and x_2, the most general quadratic form is

$$G(x_1,x_2) = a_{11}x_1^2 + 2a_{12}x_1x_2 + a_{22}x_2^2 .$$

Using matrix notation, we can write this as,

$$G(x_1,x_2) = [x_1,x_2] \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} ,$$

or $$G(x_1,x_2) = \underline{x}^T A \underline{x} . \qquad (2.10)$$

Since equation (2.10) represents a quadratic form of two variables, then the same matrix symbolism also holds for a quadratic form of n variables

2.3 EIGENVALUES AND EIGENVECTORS

The eigenproblem for a given matrix A of order n is to find the eigenvalues λ and the eigenvectors $\underline{x}$ ($\underline{x} \neq 0$) such that,

$$A\underline{x} = \lambda\underline{x} \ . \tag{2.11}$$

The *characteristic equation* of the matrix A is given by

$$|A-\lambda I| = 0 \ . \tag{2.12}$$

The eigenvalues of A are the roots λ_i (i=1,2,...,n) of the characteristic equation.

Two matrices A and B are "similar" if they have the same eigenvalues A and $C^{-1}AC$ are similar if C is a non-singular matrix.

$C^{-1}AC$ is then called a *similarity transformation* of A.

The *spectral radius* of a matrix A is defined as,

$$\rho(A) = \max_{1 \leq i \leq n} |\lambda_i| . \tag{2.13}$$

Given a vector $\underline{x}$ and a Hermitian matrix A then the Hermitian form is,

$$\underline{x}^H A\underline{x} = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{i,j} \overline{x}_i x_j \ , \tag{2.14}$$

where x_i is the complex conjugate of x_i.

Given a real vector $\underline{x}$ and a real symmetric matrix A then the "quadratic form" is,

$$\underline{x}^T A\underline{x} = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{i,j} x_i x_j \ . \tag{2.15}$$

Definition (2.3.1)

A Hermitian matrix is *positive definite* if its Hermitian form is

positive for all $\underline{x} \neq \underline{0}$, i.e.,

$$\underline{x}^H A \underline{x} > 0 \quad , \quad \forall \underline{x} \neq \underline{0} \; . \tag{2.16}$$

Definition (2.3.2)

A real symmetric matrix is positive definite if its quadratic form is positive for all $\underline{x} \neq \underline{0}$, i.e.,

$$\underline{x}^T A \underline{x} > 0 \; , \quad \forall \underline{x} \neq \underline{0} \; , \tag{2.17}$$

[JENNINGS, 1977].

The following theorem is sometimes used as a definition of positive definiteness.

Theorem (2.1)

A real matrix is positive definite if and only if it is symmetric and all its eigenvalues are positive. [YOUNG 1971].

Theorem (2.2)

A real positive definite matrix A has a unique real positive definite square root B, such that $B^2 = A$. B is written $A^{\frac{1}{2}}$. [YOUNG, 1971].

Theorem (2.3)

A real symmetric matrix A is positive definite if and only if it can be written in the form $Q^T Q = A$, where Q is a non-singular matrix of the same order.

Proof:

(i) If $A = Q^T Q$, with $(|Q| \neq 0)$,

then for any vector $\underline{x} \neq \underline{0}$,

$$\underline{x}^T A \underline{x} = \underline{x}^T Q^T Q \underline{x}$$

$= (Q\underline{x})^T(Q\underline{x}) > 0$

$\Rightarrow$ A is positive definite.

(ii) If A is real and positive definite, since $A=A^{\frac{1}{2}}A^{\frac{1}{2}}$ and $A^{\frac{1}{2}}$ is symmetric, therefore $A=(A^{\frac{1}{2}})^T A^{\frac{1}{2}}$.

As $A^{\frac{1}{2}}$ is also positive definite $|A^{\frac{1}{2}}|\neq 0$.

Thus, putting $Q=A^{\frac{1}{2}}$ gives the required condition.

Theorem (2.4)

Let λ be an eigenvalue of A with eigenvector $\underline{x}$. Then,

(i) $\alpha\lambda$ is an eigenvalue of A with eigenvector $\underline{x}$

(ii) $\lambda-\mu$ is an eigenvalue of $A-\mu I$ with eigenvector $\underline{x}$

(iii) If A is non-singular, then $\lambda\neq 0$ and λ^{-1} is an eigenvalue of A^{-1} with eigenvector $\underline{x}$. [STEWART, 1973].

2.4 VECTOR AND MATRIX NORMS

The concept of a norm is very important for analysing the errors in the later chapters, in which the approximate methods are usually associated with some vectors and matrices of which their magnitude are measurable as a non-negative scalar.

2.4.1 VECTOR NORM

Let the vector $\underline{x}$ be given by $\underline{x}^T=[x_1,x_2,\ldots,x_n]$. A norm of the vector $\underline{x}$ is a real number $||\underline{x}||$ satisfying the following requirements:

(i) $||\underline{x}|| \geq 0$, for $\underline{x} \neq \underline{0}$;

(ii) $||\alpha\underline{x}|| = |\alpha| \; ||\underline{x}||$, for any scalar α

(iii) $||\underline{x}+\underline{y}|| \leq ||\underline{x}|| + ||\underline{y}||$ (triangle inequality)

The most frequently used vector norms are:

$$||\underline{x}||_1 = \sum_{i=1}^{n} |x_i| \quad , \qquad \text{(1 norm)} \qquad (2.18)$$

$$||\underline{x}||_2 = \{ \sum_{i=1}^{n} |x_i|^2 \}^{\frac{1}{2}} , \qquad \text{(Euclidean norm)} \qquad (2.19)$$

$$||\underline{x}||_\infty = \max_{1 \leq i \leq n} |x_i| , \qquad (\infty \text{ norm}) \qquad (2.20)$$

Equations (2.18),(2.19) are particular cases of the general L_p-norms

$$||\underline{x}||_p = \{ \sum_{i=1}^{n} |x_i|^p \}^{1/p} \quad , \quad 1 \leq p \leq \infty \qquad (2.21)$$

2.4.2 MATRIX NORM

In a similar manner, the norm of a square matrix A is a non-negative number denoted by $||A||$ satisfying the following conditions:

(i) $||A|| > 0$, if $A \neq 0$,

(ii) $||\alpha A|| = |\alpha| \; ||A||$, for any scalar α

(iii) $||A+B|| \leq ||A|| + ||B||$ and

(iv) $||AB|| \leq ||A||\ ||B||$.

Since matrices and vectors appear simultaneously, it is convenient to introduce the norm of a matrix in such a way that it is compatible with a given vector norm.

A matrix norm is said to be *compatible* with a given vector norm if

$$||A\underline{x}|| \leq ||A||\ ||\underline{x}|| \quad , \tag{2.22}$$

for all non-zero $\underline{x}$.

To convert the matrix norm compatible with the vector norm, it is necessary that,

$$||A|| = \max_{\underline{x}\neq\underline{0}} \frac{||A\underline{x}||}{||\underline{x}||} \tag{2.23}$$

$$= \max_{||\underline{x}||=1} ||A\underline{x}|| \ . \tag{2.24}$$

The matrix norm which is defined by (2.24) is said to subordinate to the corresponding vector norm.

The *matrix norm* subordinate to $||\underline{x}||_p$ is denoted by $||A||_p$, and these norms satisfy the relations,

(i) $||A||_1 = \max_j \sum_i |a_{ij}|$, (maximum absolute column sum)

(ii) $||A||_2 =$ (maximum eigenvalue of $A^TA)^{\frac{1}{2}}$

$$= \sqrt{\rho(A^TA)}$$

(iii) $||A||_\infty = \max_i \sum_j |a_{ij}|$, (maximum absolute row sum)

Theorem (2.5)

If A is a matrix of order n, then,

$$\rho(A) \leq ||A|| \tag{2.25}$$

Proof:

If λ is any eigenvalue of A and $\underline{x}$ is an eigenvector associated with the eigenvalue λ, then $A\underline{x}=\lambda\underline{x}$.

Thus, $$||\lambda\underline{x}|| = |\lambda|\,||\underline{x}|| = ||A\underline{x}||$$
$$\leq ||A||\;||\underline{x}||$$

from which we conclude that

$$|\lambda| \leq ||A||, \text{ for all eigenvalues of A.}$$

Theorem (2.6)

For any real symmetric matrix A of order n,

$$||A||_2 = \rho(A).$$

Proof:

Since A is symmetric

$$||A||_2^2 = \rho(A^T A) = \rho(A^2) = \rho^2(A),$$

and hence the result follows.

2.5 CONVERGENCE OF SEQUENCE OF MATRICES

Definition (2.5.1)

The matrix A converges to zero if the sequence of matrices $A, A^2, A^3, \ldots$ converges to the null matrix O.

Definition (2.5.2)

A Jordan submatrix of A is a matrix of the form,

$$\begin{bmatrix} \lambda_i & & & \\ 1 & \lambda_i & & O \\ & 1 & \ddots & \\ & & \ddots & \lambda_i \\ O & & & 1 \end{bmatrix} , \qquad (2.26)$$

where λ_i is an eigenvalue of A. The order of the Jordan submatrix corresponds to the number of coincident eigenvalues λ_i of A. Each Jordan submatrix has only one eigenvector.

The Jordan canonical form of A is a block diagonal matrix composed of Jordan submatrices and is unique. Any matrix A can be reduced to a Jordan canonical form by a similarity transformation,

$$J = Q^{-1}AQ \ .$$

The diagonal elements of J are the eigenvalues of A.

If A has n distinct eigenvalues, its Jordan canonical form is diagonal and its n associated eigenvectors are unique and linearly independent. They form a complete system of eigenvectors and span the whole n-dimensional space. If A does not have n distinct eigenvalues, it may or may not possess n independent eigenvectors.

Theorem (2.7)

$$\lim_{r\to\infty} A^r = 0 \text{ , if } ||A|| < 1 \text{ .} \tag{2.27}$$

Proof:

$$\begin{aligned} ||A^r|| &= ||A\, A^{r-1}|| \\ &\leq ||A||\ ||A^{r-1}|| \\ &\leq ||A||^2 ||A^{r-2}|| \\ &\vdots \\ &\leq ||A||^r \end{aligned}$$

and so the results follow.

Theorem (2.8)

$$\lim_{r\to\infty} A^r = 0 \text{ if and only if } |\lambda_i| < 1$$

for all eigenvalues λ_i, (i=1,2,...,n) of A.

Proof:

Consider the Jordan canonical form of A. A Jordan submatrix of A is of the form,

$$\begin{bmatrix} \lambda_i & & & \\ 1 & \lambda_i & & \\ & \ddots & \ddots & \\ & & 1 & \lambda_i \end{bmatrix}$$

where λ_i is an eigenvalue of A. If this matrix is raised to the power r, then the result tends to the null matrix as $r\to\infty$, if and only if $|\lambda_i|<1$. i.e. $\rho(A)<1$. (This proof is given in more detail in (Varga (1962), p. 13-15).

2.6 FUNDAMENTAL ANALYSIS

Definition (2.6.1)

A linear space (or linear vector space) is a non-empty set X of elements, in which any two elements $\underline{x},\underline{y} \in X$ can be combined by a process called addition to give some element in X denoted by $\underline{x}+\underline{y}$, provided the process of addition satisfies the following conditions:

(i) $\underline{x}+\underline{y} = \underline{y} + \underline{x}$,

(ii) $\underline{x}+(\underline{y}+\underline{z}) = (\underline{x}+\underline{y})+\underline{z}$,

(iii) there exists a unique element $\underline{0} \in X$ such that $\underline{0}+\underline{x} = \underline{x}+\underline{0}$ for all $\underline{x} \in X$,

(iv) for each $\underline{x}$, there exists a negative $-\underline{x}$ such that $\underline{x}+(-\underline{x}) = \underline{0}$.

It is also a necessary condition of a linear space that an element $\underline{x} \in X$ can be combined with any real number or scalar α by scalar multiplication to give an element $\alpha\underline{x}$.

The process of scalar multiplication must satisfy the following conditions:

(v) $\alpha(\underline{x}+\underline{y}) = \alpha\underline{x}+\alpha\underline{y}$,

(vi) $(\alpha+\beta)\underline{x} = \alpha\underline{x}+\beta\underline{x}$,

(vii) $(\alpha\beta)\underline{x} = \alpha(\beta\underline{x})$,

(viii) $1\underline{x} = \underline{x}$.

Definition (2.6.2)

An expression of the form

$$\alpha_1\underline{x}^{(1)}+\alpha_2\underline{x}^{(2)}+\ldots+\alpha_n\underline{x}^{(n)} \quad , \quad \text{for all } \underline{x}^{(i)} \in X$$

is called a linear combination of the $\underline{x}$'s.

Definition (2.6.3)

A finite set of vectors $\underline{x}^{(1)},\underline{x}^{(2)},\ldots,\underline{x}^{(n)}$ is *linearly dependent* if there are scalars $\alpha_1,\alpha_2,\ldots,\alpha_n$ not all zero, such that

$$\alpha_1\underline{x}^{(1)} + \alpha_2\underline{x}^{(2)}+\ldots+\alpha_n\underline{x}^{(n)} = \underline{0} .$$

If this is not the case, the vectors are called *linearly independent.*

Definition (2.6.4)

Let n be a positive integer. Suppose that we can find a set of n vectors $\underline{x}^{(1)},\underline{x}^{(2)},\ldots,\underline{x}^{(n)} \in X$ which are independent while every set of n+1 vectors are dependent, then X is said to be *a linear space of dimension n.* If no such n exists, then X is called an *infinite dimensional space.* A system of linearly independent vectors is said to constitute a *basis* for a space, if any vector of the space is a linear combination of vectors of the system.

The number of vectors forming a basis is equivalent to the dimension of the space.

The n linearly independent vectors form a complete system and are said to span the whole n space.

The *"inner (or scalar) product"* of two members $\underline{x}$ and $\underline{y}$ of the vector space is defined by $(\underline{x},\underline{y}) = \sum_{i=1}^{n} x_i y_i$.

The *"length"* of a vector $\underline{x}$ is given by,

$$\sqrt{(\underline{x},\underline{x})} = \sqrt{\sum_{i}^{n} x_i^2} \quad .$$

The non-zero vectors $\underline{x}$ and $\underline{y}$ are said to be *"orthogonal"* if $(\underline{x},\underline{y})=0$.

A system of vectors is orthogonal, if and only if, any two vectors of the system are orthogonal to one another.

Theorem (2.9)

The vectors forming an orthogonal system are linear independent.

Proof:

Let $\underline{x}^{(1)},\underline{x}^{(2)},\ldots,\underline{x}^{(n)}$ form an orthogonal system and suppose that,

$$C_1\underline{x}^{(1)} + C_2\underline{x}^{(2)}+\ldots+C_n\underline{x}^{(n)} = \underline{0} .$$

If by taking the scalar product with $\underline{x}^{(i)}$, we obtain

$$C_i(\underline{x}^{(i)},\underline{x}^{(i)}) = 0 ,$$

for any i=1,2,...,n. Since by definition $(\underline{x}^{(i)},\underline{x}^{(i)})\neq 0$, it follows that,

$$C_i = 0 , \quad (i=1,2,\ldots,n) .$$

Thus, the vectors $\underline{x}^{(1)},\underline{x}^{(2)},\ldots,\underline{x}^{(n)}$ are linear independent.

A vector is said to be *normalised* if it is multiplied by a scalar in order to produce the size of components to numbers of values less than or equal to 1 without changing the direction of the vector.

Two common ways of normalising a vector $\underline{x}$ is by selecting a scalar β such that, either:

(1) $\beta = \sqrt{\sum_{i=1}^{n} x_i^2}$

(2) $\beta = \max_i (x_i), \quad (i=1,2,\ldots,n)$

to obtain the normalised vector $(\frac{x_1}{\beta_1} , \frac{x_2}{\beta_2} , \ldots , \frac{x_n}{\beta_n})^T$.

Definition (2.6.5)

A normed linear space (n.l.s.) is a linear space on which there is defined a norm $||\underline{x}||$ such that:

(i) $||\underline{x}|| \geq 0$,

(ii) $||\underline{x}|| = 0$, iff $\underline{x} = 0$,

(iii) $||\underline{x}+\underline{y}|| \leq ||\underline{x}|| + ||\underline{y}||$,

(iv) $||\alpha\underline{x}|| = |\alpha| \, ||\underline{x}||$.

Thus, we have the concept of the length of an element in the linear space. A *semi-norm*, satisfies (i),(iii),(iv) but not (ii).

Definition (2.6.6)

An inner product space (i.p.s.) or scalar product space is a linear space in which there is defined a real-valued function $(\underline{x}^{(1)},\underline{x}^{(2)})$ for which,

$$(i)\quad (\underline{x}^{(1)}+\underline{x}^{(2)},\underline{x}^{(3)}) = (\underline{x}^{(1)},\underline{x}^{(3)})+(\underline{x}^{(2)},\underline{x}^{(3)}) \quad \text{(linearity)}$$

$$(ii)\quad (\underline{x}^{(1)},\underline{x}^{(2)}) = (\underline{x}^{(2)},\underline{x}^{(1)}) \quad \text{(symmetry)}$$

$$(iii)\quad (\alpha\underline{x}^{(1)},\underline{x}^{(2)}) = \alpha(\underline{x}^{(1)},\underline{x}^{(2)}) \quad \alpha \text{ real} \quad \text{(homogenity)}$$

Definition (2.6.7)

(a) A sequence of elements of the linear space X, $\{\underline{x}^{(n)}\}$ is called a Cauchy sequence, if for every $\varepsilon > 0$, there is an integer $N(\varepsilon)$ such that for all $n,m \geq N$,

$$||\underline{x}^{(n)}-\underline{x}^{(m)}|| < \varepsilon ,$$

(b) $\{\underline{x}^{(n)}\}$ is *convergent sequence* if there exists a point $\underline{x}$ in the i.p.s. such that for each $\varepsilon > 0$, there exists some $N=N(\varepsilon)$ such that for all $n \geq N$

$$||\underline{x}^{(n)}-\underline{x}|| < \varepsilon .$$

If every Cauchy sequence in a normed linear space X converges to a point in the space, the space is said to be complete.

A complete normed linear space is called a *Banach space*.

An inner product space which is complete and in which all Cauchy sequences are convergent sequences is called *a Hilbert space* H.

Thus a Hilbert space is an infinite-dimensional Banach space in which an inner product is defined and which is complete with respect to the norm $||\underline{x}|| = \sqrt{(\underline{x},\underline{x})}$. In analysis, a generalization of the integral considered by Lebesgue overcomes the limitation of the Riemann integral.

Consider an integral $\int f(x)dx$ respresenting the area under the curve $y=f(x)$. The Riemann integral can be approximated by the sum,

$$S = y_1(x_1-x_0) + y_2(x_2-x_1)+\ldots+y_n(x_n-x_{n-1}) .$$

Clearly the Riemann integral does not exist if $f(x)$ oscillates too violently. The decisive idea in the Lebesgue integral is the notion of measure. The measure of an open interval $a<x<b$ is simply the length $(a-b)$. If a set consists of a finite collection of such intervals, the measure is the sum of the lengths. The Lebesgue integral is then approximated by the sum,

$$S = y_1 m(e_1) + y_2 m(e_2)+\ldots+y_n m(e_n) , \qquad (2.28)$$

where $m(e_i)$ denotes the measure of the sets e_i, $i=1,2,\ldots,n$.

Riemann's definition breaks down if $f(x)$ remains close to y_k whereas Lebesgue's definition cannot break down because $f(x)$ is automatically close to y_k throughout the set e_k.

So far, the spaces introduced have been such that a point in the space has represented a point on the real line, a vector or a matrix. In order to provide a Hilbert space which is readily applicable to the development of finite element methods, it is necessary to introduce a space in which the points represent functions. We can introduce one function $L_2(R)$ where R is an interval $[a,b]$ along the real line, then functions $f(\underline{x})$ are points in this space, if and only if,

$$(f(\underline{x}),f(\underline{x})) = \int_a^b f^2(x)d\underline{x} < \infty .$$

Such functions are said to be measurable.

For any two points u and v $\in$ H the inner product is given by

$$(u,v) = \int_a^b u(\underline{x})v(\underline{x})d\underline{x} ,$$

where integration in the Lebesgue sense is implied; u and v are orthogonal if $(u,v)=0$, and the norm given by,

$$||u|| = \int_a^b u^2(x)dx < \infty , \qquad \forall u \in H.$$

The space of all equivalence classes of real-valued (or complex-valued) Lebesgue-measurable functions u such that $|u|^p$, , $1 \leq p \leq \infty$ is a Banach space denoted $L_p(R)$ and with the norm

$$||u||_{L_p(R)} = \left(\int_R |u|^p \, dx \right)^{1/p} . \tag{2.29}$$

Linear Differential Operator

Consider a linear boundary value problem,

$$\left.\begin{aligned} Du &= f , \quad \text{in } R , \\ Lu &= g , \quad \text{on } \partial R, \end{aligned}\right\} \tag{2.30}$$

with

where D and L are linear differential operators in the domain R and on the boundaries ∂R.

Consider the 2mth order operator in the form,

$$D = \alpha_1 \frac{d^{2m}}{dx^{2m}} + \alpha_2 \frac{d^{2m-1}}{dx^{2m-1}} + \ldots + \alpha_n . \tag{2.31}$$

Let us construct an inner product of Du with another function v within a domain $x_1 < x < x_2$, and with $\alpha_1=1, \alpha_2=\alpha_3 = \ldots=\alpha_n=0$, then,

$$(Du,v) = \int_{x_1}^{x_2} (Du)v\,dx$$

Integrating by parts yields,

$$\int_{x_1}^{x_2} (Du)v\,dx = \left[\frac{d^{2m-1}u}{dx^{2m-1}} v\right]_{x_1}^{x_2} - \int_{x_1}^{x_2} \frac{d^{2m-1}u}{dx^{2m-1}} \frac{dv}{dx} dx$$

$$= \left[\frac{d^{2m-1}u}{dx^{2m-1}}v\right]_{x_1}^{x_2} - \left[\frac{d^{2m-2}u}{dx^{2m-2}}\frac{dv}{dx}\right]_{x_1}^{x_2} + \int_{x_1}^{x_2}\frac{d^{2m-2}u}{dx^{2m-2}}\frac{d^2v}{dx^2}\,dx$$

$$= \left[\frac{d^{2m-1}u}{dx^{2m-1}}v\right]_{x_1}^{x_2} - \ldots + \ldots - \left[u\frac{d^{2m-1}v}{dx^{2m-1}}\right]_{x_1}^{x_2} + \int_{x_1}^{x_2} u\frac{d^{2m}v}{dx^{2m}}\,dx \qquad (2.32)$$

Equation (2.32) may be put in the form,

$$(Du,v) = (u,D^*v) + [L^{(E)}(u,v) + L^{(N)}(u,v)]_{x_1}^{x_2} , \qquad (2.33)$$

with $$L^{(E)}(u,v) = \sum_{r=0}^{m-1}(-1)^{r+1}G_r u F_r v$$

and $$L^{(N)}(u,v) = \sum_{r=m}^{2m-1}(-1)^{r+1}F_r^* u G_r^* v$$

where $G_r u = g^{(E)}$ and $F_r^* u = g^{(N)}$ are called Dirchlet (essential) and Neumann (natural) boundary conditions, respectively and $G_r u$ and $F_r^* u$ and defined as,

$$\left.\begin{aligned} G_r u &= \left[\frac{d^0}{dx^0}, \frac{d^1}{dx^1}, \frac{d^2}{dx^2}, \ldots, \frac{d^{m-1}}{dx^{m-1}}\right]u \\ F_r^* u &= \left[\frac{d^m}{dx^m}, \frac{d^{m+1}}{dx^{m+1}}, \frac{d^{m+2}}{dx^{m+2}}, \ldots, \frac{d^{2m-1}}{dx^{2m-1}}\right]u \end{aligned}\right\} \qquad (2.34)$$

Here G_r and F_r are the boundary operators. The expression (2.33) is known as Green's formula. In two or three-dimensional problems, the Green's formula takes the form,

$$(Du,v) = \int_R uD^*v\,dR + \int_{\partial R}[L_r^{(E)}(u,v) + L_r^{(N)}(u,v)]\,ds \qquad (2.35)$$

It should be noted that for the 2mth equation, we have D*=D,

$$G_r^*(r=2m-1,2m-2,\ldots,m) = G_r\ (r=0,1,\ldots,m)$$

and $$F_r^*(r=2m-1,2m-2,\ldots,m) = F_r\ (r=0,1,\ldots,m) .$$

Equations with these conditions are referred to as *self-adjoint*; and the linear differential operator D is known as a self-adjoint operator.

Moreover, the condition D*=D and v=u result in *symmetric positive*

definite properties for the inner product (Du,v). If $D^* \neq D$, then D is a non-self-adjoint operator, resulting in a nonself-adjoint equation.

The partial differential equation (2.30) is seen to be equivalent to the integral relation (2.35). We then say that u is a solution in the weak sense of the original equation if it satisfies this integral relation for all functions v of the class considered.

To show how we apply Green's theorem in the derivation of the finite element equations for two dimensional p.d.e. problems, we consider the given problem, let

$$D = \alpha\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right) + \beta \ , \quad x_1 < x < x_2 \ ,$$

where α and β are constants.

Then,

$$\begin{aligned}
(Du,v) &= \int_{x_1}^{x_2} \left[\alpha\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) + \beta u\right] v \, dxdy \\
&= \alpha\left(\frac{\partial u}{\partial x} + \frac{\partial u}{\partial y}\right) v \Big]_{x_1}^{x_2} - \int_{x_1}^{x_2} \alpha\left(\frac{\partial u}{\partial x}\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\frac{\partial v}{\partial y}\right) dxdy + \int_{x_1}^{x_2} \beta uv \, dxdy \\
&= \alpha\left[\left(\frac{\partial u}{\partial x} + \frac{\partial u}{\partial y}\right)\right] v \Big]_{x_1}^{x_2} - \alpha\left[u\left(\frac{\partial v}{\partial x} + \frac{\partial v}{\partial y}\right)\right] \Big]_{x_1}^{x_2} \\
&\quad + \int_{x_1}^{x_2} u\left(\alpha\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right)\right) dxdy + \int_{x_1}^{x_2} \beta uv \, dxdy
\end{aligned}$$

Note that m=1 here, and the results above can be written in the form (2.33),

$$(Du,v) = (u,D^*v) - \alpha\left(G_o u F_o v - F_1^* u G_1^* v\right) \Big]_{x_1}^{x_2}$$

where,

$$D = \alpha\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right) + \beta \ , \quad G_o = 1, \quad F_o = \left(\frac{\partial}{\partial x} + \frac{\partial}{\partial y}\right)$$

$$F_1^* = \left(\frac{\partial}{\partial x} + \frac{\partial}{\partial y}\right), \quad G_1^* = 1.$$

Thus, we have $D^*=D$ and $G_1^*=G_o$, and the operator D is the self-adjoint,

operator, with $G_0 u=u$ and $F_1 u=(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y})$ are the Dirichlet and Neumann boundary conditions, respectively.

Sobolev Spaces

Let the set of ordered n-tuples of non-negative integers be given by,

$$\alpha = (\alpha_1,\alpha_2,\ldots,\alpha_n)$$

Also we define,

$$|\alpha| = \alpha_1+\alpha_2+\ldots+\alpha_n$$

$\underline{x} = (x_1,x_2,\ldots,x_n) \in R^n$, where R^n is an n-dimensional space.

$$|\underline{x}| = \Big(\sum_{i=1}^{n} |x_i|^2\Big)^{\frac{1}{2}}$$

$$\underline{x}^{\alpha} = x_1^{\alpha_1} . x_2^{\alpha_2} \ldots\ldots x_n^{\alpha_n}$$

and

$$D^{\alpha}u(\underline{x}) = \frac{\partial^{|\alpha|}u(\underline{x})}{\partial x_1^{\alpha_1}.\partial x_2^{\alpha_2}\ldots\ldots\partial x_n^{\alpha_n}} = \Big(\Big(\frac{\partial^{\alpha_1}}{\partial x_1^{\alpha_1}}\Big)\Big(\frac{\partial^{\alpha_2}}{\partial x_2^{\alpha_2}}\Big)\ldots\Big(\frac{\partial^{\alpha_n}}{\partial x_n^{\alpha_n}}\Big)\Big)u(\underline{x}) \tag{2.36}$$

together with the duality pairing,

$$(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\ldots\int_{-\infty}^{\infty} u(\underline{x})v(\underline{x})\, dx_1.dx_2\ldots\ldots dx_n$$

$$= \int_{R^n} uv d\underline{x} \, . \tag{2.37}$$

Sobolev spaces are the generalization of L_p spaces so that all the weak derivatives of functions $u(\underline{x})$ are included in L_p whose norm is defined by (2.29). If all partial derivatives of $u(\underline{x})$ of order $\leq m$, m being an integer ≥ 0, are in L_p, then $u(\underline{x})$ belongs to a Sololev space denoted as $W_p^m(R)$, of order m,p on R, i.e.,

$$W_p^m(R) = \{u: D^{\alpha}u \in L_p(R) \quad \forall\alpha \text{ such that } |\alpha| \leq m\} \tag{2.38}$$

Clearly, for m=O $W_p^O(R) = L_p(R)$.

Since, for each α such that $0 \leq |\alpha| \leq m$, $D^\alpha u$ is in $L_p(R)$, the sum of the L_p norms of all the weak derivatives of u of order $\leq m$ satisfies the norm axioms and suggested naturally by the definition (2.37).

Thus, we may introduce for each $u \in W_p^m(R)$ the norm,

$$||u||_{W_p^m(R)} = \left(\int_R \sum_{|\underline{\alpha}| \leq m} |D^\alpha u|^p d\underline{x} \right)^{1/p}$$

$$= \left(\sum_{|\alpha| \leq m} ||D^\alpha u||^p_{L_p(R)} \right)^{1/p} \qquad (2.39)$$

Hereafter, whenever we refer to the Sobolev space $W_p^m(R)$, we mean the normed space consisting of linear space of functions given in (2.38) together with the Sobolev norm (2.39).

Consider the spaces L_2, W_2^1, W_3^2 with an open interval on the real line $R=(x_1,x_2)$. The associated Sobolev norms of u is given by,

for $u(x) \in L_2(x_1,x_2)$

$$||u||_{L_2(x_1,x_2)} = \left[\int_{x_1}^{x_2} u^2 dx \right]^{\frac{1}{2}} < \infty,$$

for $u(x) \in W_2^1$

$$||u||_{W_2^1(x_1,x_2)} = \left\{ \int_{x_1}^{x_2} [u^2 + (\frac{du}{dx})^2] dx \right\}^{\frac{1}{2}} < \infty,$$

and for $u(x) \in W_3^2$

$$||u||_{W_3^2(x_1,x_2)} = [\int_{x_1}^{x_2} (|u|^3 + |\frac{du}{dx}|^3 + |\frac{d^2u}{dx^2}|^3) dx]^{1/3} < \infty,$$

If the domain $R=(x_1,x_2)\times(y_1,y_2) \in R^2$ and $u(x,y) \in W_p^2$, $p \geq 1$, then the Sobolev norm of u is,

$$||u||_{W_p^2(R)} = \left[\int_{x_1}^{x_2} \int_{y_1}^{y_2} (|u|^p + |\frac{\partial u}{\partial x}|^p + |\frac{\partial u}{\partial y}|^p + |\frac{\partial^2 u}{\partial x^2}|^p + |\frac{\partial^2 u}{\partial x \partial y}|^p + |\frac{\partial^2 u}{\partial y^2}|^p) dx dy \right]^{1/p}.$$

In the study of most linear boundary value problems, we encounter $W_2^m(R)$

spaces. These spaces are Hilbert spaces $H^m(R)$.

Consider the space $H^m(R)$ of function $u(x)$ on R with m, an integer ≥ 1 defined by,

$$H^m(R) = \{u: D^\alpha u \in L_2(R) \ ; \ \forall \alpha \text{ such that } |\alpha| \leq m\}, \tag{2.40}$$

where R is a bounded open set in R^n, and $D^\alpha u$ denotes the weak αth derivative of u.

We provide $H^m(R)$ with the inner product,

$$\begin{aligned}(u,v)_{H^m(R)} &= \sum_{|\alpha| \leq m} (D^\alpha u, D^\alpha v)_{L_2(R)} \\ &= \sum_{|\alpha| \leq m} \int_R D^\alpha u D^\alpha v \, dx \end{aligned} \tag{2.41}$$

and the associated norm,

$$\begin{aligned}||u||_{H^m(R)} &= \Big(\sum_{|\alpha| \leq m} ||D^\alpha u||^2_{L_2(R)} \Big)^{\frac{1}{2}} \\ &= [(u,u)_{H^m(R)}]^{\frac{1}{2}} < \infty \end{aligned} \tag{2.42}$$

In view of the definitions (2.38) and (2.39), we see that

$$H^m_{(R)} = W^m_2(R) \ . \tag{2.43}$$

The Hilbert space $H^m_{(R)}$ is thus a Sobolev space of order m,2. $H^m(R)$ is a complete Sobolev space of order m defined by (2.42) and with respect to the norm (2.40).

Also, if $H^{m_1}_{(R)}$ and $H^{m_2}_{(R)}$ are two Sobolev spaces, $m_1 \geq m_2 \geq 0$ then,

$$H^{m_1}_{(R)} \subseteq H^{m_2}_{(R)} \subseteq H^0_{(R)} = L_2(R) \ . \tag{2.44}$$

Note that it also follows from the definitions that, whenever $m_1 \geq m_2$, and $u \in H^m_{(R)}$,

$$||u||_{H^{m_1}_{(R)}} \geq ||u||_{H^{m_2}_{(R)}} \ .$$

Our aim here is to present only the essential features of the Sobolev and Hilbert spaces, for additional details on these subjects, the references listed at the end of the thesis can be consulted. In particular see ODEN[1976].

2.7 SOLUTION OF FINITE ELEMENT EQUATIONS

As we mentioned previously, when the finite element method is used for solving a linear problem then an associated set of simultaneous algebraic equations is generated which can be stated in the form given by equation (2.1). Equation (2.1) can be expressed in matrix form as,

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}, \qquad (2.45)$$

where the coefficients a_{ij} and the constants b_i are either given or can be generated. The problem is to find the values x_i, (i=1,2,...,n) if they exist, which satisfy equation (2.45).

A comparison of equations (2.1) and (2.45) shows that,

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}, \quad \underline{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad \underline{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

In finite element analysis, the order of the matrix A will be very large and A is non-singular and sparse (in many cases A is symmetric and positive definite).

The feasibility of the application of the finite element method

hinges on, how fast can the equations be solved? And how accurate is the computed solution? The first question can be answered by counting the operations involved in the solution algorithm used. The second question is more fundamental since its answer determines whether it is meaningful to solve the equations numerically, the major consideration here is that of round-off error, which may lead to poor or even useless results.

There are usually two kinds of methods used for solving a system of equations of large order: (1) direct methods and (2) iterative methods.

A direct method is one which, after a finite number of operations, if all computations were carried out without round-off error, would lead to the exact solution of the algebraic system. An iterative method usually requires an infinite number of iterations to converge to the true solution. Within a tolerable error, there is no clear-cut answer as to which of these methods is best for a system such as (2.1). In practice, the round-off error is usually the controlling factor in determining whether a direct method of solution can be used. Either solution scheme will be seen to have certain advantages, however direct methods appear to hold an advantage in solving 2-dimensional systems arising out of the finite element methods, and iterative methods appear to have the edge in solving systems arising from finite difference equations.

2.7.1 DIRECT SOLUTION METHODS USING TECHNIQUES BASED ON THE GAUSS ELIMINATION PROCESS

The most effective direct solution techniques currently used are basically the applications of the Gauss elimination process, however,

although the basic Gauss solution scheme can be applied to almost any set of simultaneous linear equations, the effectiveness in finite element analysis depends on the specific properties of the finite element master matrix: symmetry, positive definiteness, and bandedness The details of the method are as follows. Starting from the given system (2.1), $A\underline{x}=\underline{b}$, or,

$$A^{(1)}\underline{x} = \underline{b}^{(1)} \quad , \text{ where } A = A^{(1)}, \quad \underline{b} = \underline{b}^{(1)} \quad . \tag{2.46}$$

Step 1: The essence of this method is to reduce the matrix in the preceding equation into a lower triangular form by elimination. Toward this end we define n-1 multipliers,

$$m_{i1} = \frac{a_{i1}}{a_{11}} \quad , \; i=2,3,\ldots,n, \qquad m_1 < 1 \; ,$$

and subtract m_{i1} times the first row from the ith row.

If we define,

$$a'_{ij} = a_{ij} - m_{i1}a_{1j} \quad , \quad i=2,\ldots,n$$

$$b'_i = b_i - m_{i1}b_1 \quad , \quad j=1,\ldots,n,$$

it is easy to see that,

$$a'_{i1} = 0 \quad , \quad i=2,\ldots,n.$$

The resulting equations are,

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ 0 & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ 0 & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \tag{2.47}$$

or

$$A^{(2)}\underline{x} = \underline{b}^{(2)} \quad , \tag{2.48}$$

where in equation (2.47) we have *renamed* the a'_{ij} and b'_i to be a_{ij} and b_i to simplify the notation. We re-emphasize that these a_{ij} and b_i are not the same coefficients appearing in (2.45). We also stress that this

last system (2.47) has the same solution as does (2.45). It is easy to recognize that the preceding step is equivalent to the operation defined as $A^{(2)}=M_1^{-1}A^{(1)}$, where,

$$M_1^{-1} = \begin{bmatrix} 1 & & & \\ -m_{21} & 1 & & \\ -m_{21} & 0 & \ddots & \\ \vdots & \vdots & & \\ -m_{n1} & 0 & \cdots\cdots & 0 \quad 1 \end{bmatrix} \quad , \quad m_{i1} = \frac{a_{i1}^{(1)}}{a_{1,1}^{(1)}} \tag{2.49}$$

Step 2: We continue in a similar way such that at the kth stage we eliminate x_k by defining the multipliers

$$m_{i,k} = \frac{a_{ik}}{a_{k,k}} \quad , \; i=k+1,\ldots,n \tag{2.50}$$

where $a_{k,k}\neq 0$. Then,

$$\hat{a}_{ij} = a_{ij} - m_{ik}a_{kj} \; , \tag{2.51}$$

$$\hat{b}_i = b_i - m_{ik}b_k \; , \tag{2.52}$$

for i=k+1,...,n and j=k,...,n. The index k takes on consecutive integer values from 1 to n-1.

At the point where k=n-1, we are eliminating x_{n-1} from the last equation. The final triangular set of equations is thus given by,

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & a_{22} & a_{23} & \cdots & a_{2n} \\ 0 & 0 & a_{33} & \cdots & a_{3n} \\ - & - & - & - & - \\ - & - & - & - & - \\ - & - & - & - & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix} \quad , \tag{2.53}$$

or $$A^{(n)}\underline{x} = b^{(n)} = S\underline{x} \; , \tag{2.54}$$

where $$A^{(n)} = M_{n-1}^{-1}.M_{n-2}^{-1}.\ldots.M_1^{-1}A^{(1)} \triangleq S \; . \tag{2.55}$$

Step 3: A back-substituion process then produces the solution as follows:-

$$x_n = \frac{b_n}{a_{nn}}$$

$$\vdots$$

$$x_j = \frac{b_j-(a_{j,j+1}+\ldots+a_{jn}x_n)}{a_{jj}} \qquad (2.56)$$

$$\vdots$$

and finally,

$$x_1 = \frac{b_1-(a_{12}x_2+\ldots+a_{1n}x_n)}{a_{11}} \quad .$$

The operations performed in the preceding elimination procedure can be compactly written, for any n×n matrix A, as follows, starting from,

$$A^{(1)} = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} \\ a_{21}^{(1)} & a_{22}^{(1)} & \cdots & a_{2n}^{(1)} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1}^{(1)} & a_{n2}^{(1)} & \cdots & a_{nn}^{(1)} \end{bmatrix} \qquad (2.57)$$

we have,

$$M_1^{-1} = \begin{bmatrix} 1 & & & \\ -m_{21} & 1 & & \\ -m_{31} & 0 & \ddots & \\ \cdots & \cdots & \cdots & \\ -m_{n1} & 0 & \cdots & 1 \end{bmatrix} \quad , \quad m_{21} = \frac{a_{21}^{(1)}}{a_{1,1}^{(1)}} \text{ , etc.}$$

$$\vdots$$

and

$$M_k^{-1} = \begin{bmatrix} 1 & \ddots & & & \\ 0 & \ddots & & & \\ \cdots & \cdots & 1 & & \\ 0 & \cdots & -m_{k+1,k} & \ddots & \\ 0 & \cdots & -m_{k+2,k} & & \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & & -m_{n,k} & & 1 \end{bmatrix} \quad , \quad m_{k+1,k} = \frac{a_{k+1,k}^{(k)}}{a_{k,k}^{(k)}} \text{ , etc.} \qquad (2.58)$$

Now

$$U = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\ 0 & 0 & a_{33}^{(3)} \cdots & a_{3n}^{(3)} \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 \cdots & a_{nn}^{(n)} \end{bmatrix} = M_{n-1}^{-1} M_{n-2}^{-1} \ldots M_1^{-1} A^{(1)} \qquad (2.59)$$

Equation (2.59) can be equivalently written as,

$$A^{(1)} = LU \quad , \qquad (2.60)$$

where

$$L = M_1 . M_2 \ldots . M_{n-1} \quad , \qquad (2.61)$$

$$= \begin{bmatrix} 1 & & & \\ m_{21} & 1 & & \\ m_{31} & m_{32} & 1 & \\ \cdots & \cdots & \cdots & \cdots \\ m_{n1} & m_{n2} \cdots & m_{n,n-1} & 1 \end{bmatrix} \qquad (2.62)$$

Thus, the solution process can be described by the pair of coupled equations,

$$L\underline{y} = \underline{b} \ , \quad U\underline{x} = \underline{y} \quad , \qquad (2.63)$$

If D denotes the diagonal entries of S, then it is evident that $U=D\ L^T$ when A is symmetric. Now equation (2.63) can be written as,

$$(LDL^T)\underline{x} = \underline{b} \ , \qquad (2.64)$$

For this reason, the procedure described above is called a *LDL^T decomposition*. Notice that $\underline{y}$ and $\underline{b}$ are related by,

$$\underline{y} = L^{-1}\underline{b} = M_{n-1}^{-1} M_{n-2}^{-1} \ldots M_2^{-1} M_1^{-1} \underline{b} \ . \qquad (2.65)$$

Thus, Gaussian elimination is nothing but the factorization of A into

the product A=LU, of a lower triangular matrix L times an upper triangular matrix U. Thus $\underline{x}=A^{-1}\underline{b}$ is identical with $\underline{x}=U^{-1}L^{-1}\underline{b}$, and the two triangular matrices are easy to invert.

A very important requirement for successful Gaussian elimination is to guard against dividing by zero during the computations. Even if there are no zeros on the main diagonal at the start of the computations, zeros may be created in subsequent steps. A useful strategy to avoid such zero divisors is to rearrange the equations so as to place the coefficient of largest magnitude on the diagonal at each step. This is called *pivoting*. Complete pivoting may require both row and column interchanges. Partial pivoting, which places the coefficient of largest magnitude per column on the diagonal by row interchanges only is usually adequate in many cases.

Let us summarize the operation of the Gaussian elimination procedure in algorithm form that will facilitate writing a computer program.

Gaussian Elimination Procedure

To solve a system of linear equations we proceed as follows:-

1. Augment the (n×n) coefficient matrix with the vector $\underline{b}$ to form an n×(n+1) matrix.
2. Interchange rows if necessary to make the value of a_{11} the largest magnitude of any coefficient in the first column.
3. Subtract a_{i1}/a_{11} times the first row from the ith row ($2\leq i\leq n$). This should leave a column of zeros below the pivot element in the first column.
4. Repeat steps (2) and (3) for the second through the (n-1)st rows, as follows:

(i) Implement the partial pivoting strategy by considering only rows j to n.

(ii) Subtract a_{ij}/a_{jj} times the jth row from the ith row so as to create zeros in all positions in the jth column below the main diagonal. At this stage, the system is upper-triangular.

5. Solve for x_n from the nth equation by

$$x_n = \frac{a_{n,n+1}}{a_{nn}} .$$

6. Solve for $x_{n-1}, x_{n-2}, \ldots, x_2, x_1$ from the (n-1)st to the first equation in turn, by using the back substitution process,

$$x_i = \frac{a_{i,n+1} - \sum_{j=i+1}^{n} a_{ij} x_j}{a_{i,i}} .$$

It is important to reflect upon the elimination process if A is symmetric, and positive definite. First, the elimination process succeeds because the factorization A=LU exists. The condition for success is that each of the submatrices in the upper left corner of A, that is

$$A_1 = [a_{11}], \quad A_2 = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \quad A_3 = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}, \ldots$$

$$a_{12} = a_{21} \qquad a_{ij} = a_{ji}, \ j \neq i.$$

should have a non-zero determinant. For a positive-definite matrix, these determinants are all positive, and the elimination process can be carried out with no exchanges of rows. Another important requirement is that the pivot elements be not only non-zero but also sufficiently large. Otherwise, round-off errors will dominate the solution. This type of

sensitivity of A to small perturbations is measured by the "condition number" of A. This condition number is the ratio of the largest eigenvalue to the smallest eigenvalue of A.

For the computer implementation of the Gauss elimination solution a minimum solution time is achieved, in addition, the high-speed storage requirements should be as small as possible to avoid the use of back-up storage. However, for large systems it will nevertheless be necessary to use back-up storage, and for this reason it should also be possible to modify the solution algorithm of the finite element analysis so that the master matrix of the elements assemblage is not only symmetric and positive definite but is also banded, i.e. $a_{ij}=0$ for $j>i+m_k$, where m_k is the half-bandwidth of the system. The fact that in finite element analysis all non-zero elements are clustered around the diagonal of the system matrix greatly reduces the total number of operations and the high speed storage required in the equation solution.

However, this property depends on the nodal point numbering of the finite element mesh points, and the programmer must take care to obtain an effective nodal numbering scheme, in order to estimate the number of operations that are necessary for the solution, because this enables the analyst to estimate the computer cost for a specific problem. In addition to the LDL^T decomposition, various other schemes are used that are closely related, all these methods are applications of the basic Gauss elimination procedure.

In the Choleski factorization, the matrix A is decomposed as follows,

$$A = \bar{L}\, L^T \tag{2.66}$$

where $\bar{L} = LD^{\frac{1}{2}}$.

Therefore, the Choleski factors could be calculated from the D and L factors.

An operation count shows that slightly more operations are required in the equation solution if the Choleski factorization is used rather than the LDL^T decomposition. In addition, the Choleski factorization is only suitable for the solution of positive definite systems.

The other algorithm that can effectively be used is the Frontal solution method which will be discussed now.

2.7.2 FRONTAL SOLUTION METHOD

The frontal solution technique first devised by B.M. Irons, is a variation of Gaussian elimination that makes the utilization of external storage easy. It is customary to think of the process of assembling the master matrix and the master vector of imposing boundary conditions, and solving the system of equations,

$$A\underline{x} = \underline{b} \ ,$$

as distinct phases occurring one after the other. However, these processes can be performed in parallel in the Gaussian-elimination method. The frontal routine starts by assembling each of the element matrices in turn into the core storage, until the core area allocated to the solution routine by the programmer is filled. Then, from within this assembled part of the complete matrix, a pivotal search is made to determine the largest entry from amongst those rows and columns which are fully summed, i.e. rows and columns to which no further contributions will arise in the subsequent assembly of the element matrices. The pivotal row is then used to eliminate all the coefficients in the pivotal column, after which it is

placed on the backing storage disc. When sufficient coefficients have been eliminated it is possible to assemble the next element matrix, after which further elimination may take place. Finally after all the coefficients have been eliminated the solution is obtained by a back-substitution.

To demonstrate the procedure, consider the example shown in Figure (2.1) with one degree of freedom per node. The element numbers are circled. All the information contained in A and $\underline{b}$ for the degrees of freedom 1 to 3

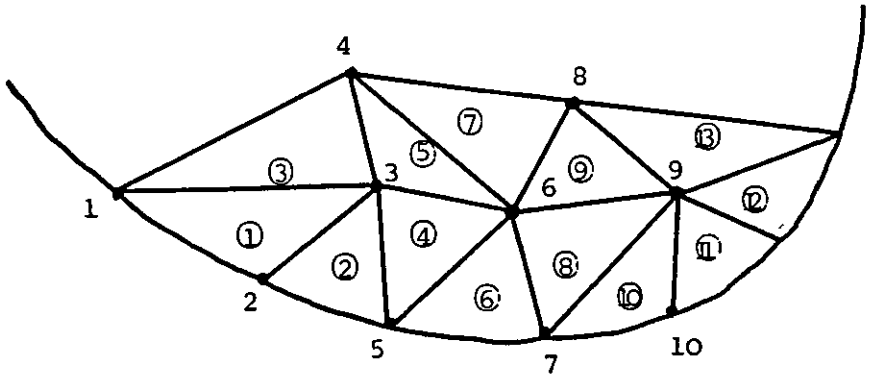


FIGURE 2.1: Finite-element divisions for a frontal solution

has been assembled after the data for element 1 to 5 have been generated. Thus, with the Gaussian technique, it is possible to impose any constraint conditions which may occur at 1 to 3 and eliminate these degrees of freedom, that is, the unconstrained degrees of freedom amongst 1 to 3 can be expressed in terms of the degrees of freedom 4 to 6 before the data for

elements 6,7, etc., are generated. If this is done, then if each condensed row required for the back-substitution phase is saved in external storage, and if their core locations in the computer are used to store the new information being generated from elements 6,7, etc. the core storage requirement for a large problem may be reduced considerably.

A reference to Figure (2.1) shows that some care must be taken in programming to realize fully the potential savings of Gaussian elimination. For example information does not begin to appear for degrees of freedom 9 and 10 until the data for element 10 are generated, and these degrees of freedom may be eliminated after the data for elements 11,12,13 have been assembled. Thus, a requirement exists, not only for a table of the degrees of freedom to which each element connects, but also for some flags to mark the first and last appearance of each degree. The flags serve two purposes. They permit a calculation of the maximum storage requirement, to be made in terms of the maximum number of degrees of freedom for which the information must be held in core simultaneously, and they are also used to reserve subareas of storage as the information associated with various degrees of freedom is shuffled in and out of core. Each of these phases (assembly-constraint-forward elimination and back substitution) propagates through the region from node to node like a wave, hence, the frontral solution is also referred to as the *wave front* technique (Irons, 1970).

To illustrate the difference between the frontal and the regular method, consider, the three elements five nodes mesh shown in Figure (2.2).

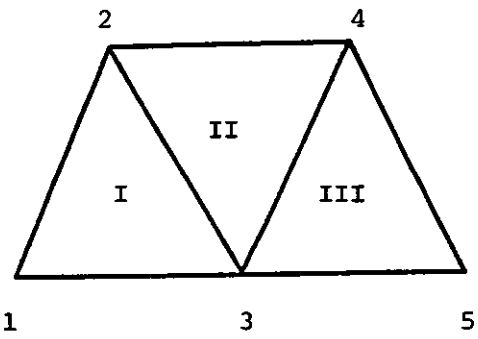


FIGURE 2.2: Three element mesh

Then the totality of the finite element equations can be written as $A\underline{x} = \underline{b}$.

After the assembly of the first element equations the state of this matrix equation is as follows:

$$\begin{bmatrix} a_{11}^{I} & a_{12}^{I} & a_{13}^{I} \\ a_{21}^{I} & a_{22}^{I} & a_{23}^{I} \\ a_{31}^{I} & a_{32}^{I} & a_{33}^{I} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1^{I} \\ b_2^{I} \\ b_3^{I} \end{bmatrix}, \qquad (2.67)$$

where superscript I denotes the element number from which the matrix entry was derived. The difference between the frontal and band routines is that in the frontal routines each equation can be eliminated at an earlier stage than the band routines - as soon as it is complete - due to the superior accounting process. Consequently core requirements are in general less for the frontal routines.

Another effect of the frontal accounting process is that it allows both column and row pivoting without excessive non-zero entry growth. This may be illustrated by carrying out the elimination of x_1 which leads to the following:

$$\begin{bmatrix} a^{I}_{22} - \frac{a^{I}_{21}}{a^{I}_{11}} a^{I}_{12} & a^{I}_{23} - \frac{a^{I}_{21}}{a^{I}_{11}} a^{I}_{13} \\ a^{I}_{32} - \frac{a^{I}_{31}}{a^{I}_{11}} a^{I}_{12} & a^{I}_{33} - \frac{a^{I}_{31}}{a^{I}_{11}} a^{I}_{13} \end{bmatrix} \begin{bmatrix} x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b^{I}_{2} - \frac{a^{I}_{21}}{a^{I}_{11}} b^{I}_{1} \\ b^{I}_{3} - \frac{a^{I}_{31}}{a^{I}_{11}} b^{I}_{1} \end{bmatrix} . \quad (2.68)$$

A subtraction of the Gaussian products does not increase the storage required, since no terms are involved other than those found in equation (2.68).

In order to minimize core requirements, the element numbering is chosen in such a way as to keep the "width" as small as possible. In addition further core is required for the assembly of the next element's equations.

Suppose now that the equations from the next element are assembled, so that the matrix equation becomes:

$$\begin{bmatrix} a^{I}_{22} - \frac{a^{I}_{21}a^{I}_{12}}{a^{I}_{11}} + a^{II}_{22} & a^{I}_{23} - \frac{a^{I}_{21}a^{I}_{13}}{a^{I}_{11}} + a^{II}_{23} & a^{II}_{24} \\ a^{I}_{32} - \frac{a^{I}_{31}a^{I}_{12}}{a^{I}_{11}} + a^{II}_{32} & a^{I}_{33} - \frac{a^{I}_{31}a^{I}_{13}}{a^{I}_{11}} + a^{II}_{32} & a^{II}_{34} \\ a^{II}_{42} & a^{II}_{43} & a^{II}_{44} \end{bmatrix} \begin{bmatrix} x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

$$= \begin{bmatrix} b^{I}_{2} - \frac{a_{21}b^{I}_{1}}{a^{I}_{11}} + b^{II}_{2} \\ b^{I}_{3} - a_{31}b^{I}_{1} + b^{II}_{3} \\ b^{II}_{4} \end{bmatrix} \quad (2.69)$$

From this it may be observed that although equations (2.67) and (2.68)

were altered before they were fully assembled, the terms subtracted involved only those components involving node 1 which were complete.

For this some advantage, of course, there is a price to pay because the wave front method requires many shuffles in and out of core, which means a longer execution time, and a table for tracing the degrees of freedom currently in core, etc. This means more complicated programming.

2.8 ITERATIVE METHODS

The procedure described in Section (2.7) for the solution of the system of linear equations $A\underline{x} = \underline{b}$ were direct methods, involving a fixed number of operations. Alternatively, iterative methods (to be described here) start from a first approximation, which is successively improved until a sufficiently accurate solution is obtained.

The majority of iterative methods for linear systems are stationary linear iteration methods: that is they can be written as,

$$\underline{x}^{(k+1)} = M\underline{x}^{(k)} + \underline{c} \quad , \; k=0,1,2,... \tag{2.70}$$

where M is a matrix depending upon A and $\underline{c}$ is a column vector depending on $\underline{b}$. Most of the well-known iterative methods are built around a partition (or splitting) of the matrix A in the form,

$$A = (D+U+L) \; ,$$

where,

$$D = \text{diag}(a_{11}, a_{22}, \ldots, a_{nn}) \; ,$$

$$U = \begin{cases} a_{ij} & i<j \\ 0 & i \geq j \end{cases} \qquad \text{(strictly upper triangular matrix)},$$

$$L = \begin{cases} a_{ij} & i>j \\ 0 & i \leq j \end{cases} \qquad \text{(strictly lower triangular matrix)}.$$

2.8.1 JACOBI METHOD

The simplest of the iterative methods is widely known as Jacobi's method or as the method of simultaneous displacements. In this scheme, the approximate solution obtained at the end of the kth iteration cycle, starting from some initial estimate $\underline{x}^{(k)}$ (initially $\underline{x}^{(0)}$), we construct

$\underline{x}^{(k+1)}$ as follows,

Jacobi iteration,

$$x_i^{(k+1)} = \frac{b_i - \sum_{\substack{j=1 \\ i\neq j}}^{n} a_{ij}x_j^{(k)}}{a_{1,1}} \quad , \; i=1,2,\ldots,n. \tag{2.71}$$

or

$$\underline{x}^{(k+1)} = -D^{-1}(L+U)\underline{x}^{(k)} + D^{-1}\underline{b} \; . \tag{2.72}$$

Comparing this with the general linear iterative scheme (2.70), we see that the choice,

$$M_J = -D^{-1}(L+U) \; , \quad \underline{c}_J = D^{-1}\underline{b} \; , \tag{2.73}$$

characterize the Jacobi method.

The procedure (2.72) is repeated (for k=0,1,2,...) until it converges to a stationary solution for which,

$$\underline{x}^{(k+1)} - \underline{x}^{(k)} = \underline{0} \; ,$$

according to some chosen norm.

In spite of its simplicity, it is seldom used as it is very slow to converge.

2.8.2 GAUSS-SEIDEL METHOD

The Gauss-Seidel method, also known as the method of successive displacements, represents a refinement of the Jacobi method. In the Jacobi method, one does not use the new values $x_r^{(k+1)}$ until every component of the vector $\underline{x}$ has been evaluated. If new values $x_r^{(k+1)}$, r=1,2,...,i-1, are used in evaluating $x_1^{(k+1)}$, then we have, instead of (2.71):

Gauss-Seidel iteration,

$$x_i^{(k+1)} = \frac{b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}}{a_{i,i}} \tag{2.74}$$

$$i=1,2,\ldots,n.$$

or in matrix notation,

$$\underline{x}^{(k+1)} = -(L+D)^{-1}U\underline{x}^{(k)} + (L+D)^{-1}\underline{b} \ . \qquad (2.75)$$

This is similar to equation (2.70) with,

$$M_{GS} = -(D+L)^{-1}U \ ,$$

and $$\underline{C}_{GS} = (D+L)^{-1}\underline{b} \ .$$

Here, as the values of x_i are successively updated and overwritten, i.e. only one approximation for each x_i needs to be stored at a time, thus saving vital computer memory.

2.8.3 AN ACCELERATION OR (OVER OR UNDER) RELAXATION METHOD

Starting from the basic Jacobi and Gauss-Seidel iteration methods; one may generate families of iterative procedures by inserting an additional parameter in the calculation with the intent of accelerating the rate of convergence, the corresponding methods are called over- or under-relaxation methods.

Jacobi Method with Acceleration

If we rewrite equation (2.72) of the Jacobi method as,

$$\underline{x}^{(k+1)} = \underline{x}^{(k)} + \{D^{-1}[\underline{b}-(L+U)\underline{x}^{(k)}]-\underline{x}^{(k)}\}$$

$$= \underline{x}^{(k)} + \underline{r}^{(k)} \ , \qquad (2.76)$$

where $\underline{r}^{(k)}$, the term in brackets, {.} is seen to be the correction in $\underline{x}$ in the (k+1)st iteration cycle.

To generate a family of accelerated iterative procedures, we multiply this correction by the scalar quantity ω, called the relaxation factor: Hence we have,

$$\underline{x}^{(k+1)} = \underline{x}^{(k)} + \omega\underline{r}^{(k)}$$

$$= \underline{x}^{(k)} + \omega\{D^{-1}[\underline{b}-(L+U)\underline{x}^{(k)}]-\underline{x}^{(k)}\} \ , \qquad (2.77)$$

which may also be written as,

$$\underline{x}^{(k+1)} = (1-\omega)\underline{x}^{(k)} + \omega D^{-1}[\underline{b}-(L+U)\underline{x}^{(k)}] \ , \tag{2.78}$$

When $\omega=1$, this expression becomes the Jacobi iteration, when $\omega>1$, equation (2.78) is called *overrelaxation* and when $\omega<1$, it is called *underrelaxation*.

On a single equation basis, (2.78) may be written as

$$x_i^{(k+1)} = (1-\omega)x_i^{(k)} + \omega \frac{b_i - \sum_{i\neq j}^{n} a_{ij}x_j^{(k)}}{a_{ii}} \ . \tag{2.79}$$

Then equation (2.78) is equivalent to the following two steps:

$$\begin{aligned} &\text{(i)} \quad \hat{\underline{x}}^{(k+1)} = D^{-1}[\underline{b}-(L+U)\underline{x}^{(k)}] \ , \\ &\text{(ii)} \quad \underline{x}^{(k+1)} = (1-\omega)\underline{x}^{(k)} + \omega\hat{\underline{x}}^{(k+1)} \ , \end{aligned} \tag{2.80}$$

where $\hat{\underline{x}}^{(k+1)}$ is the normal Jacobi result. Similarly, on a single equation basis, we may write, (2.80),

$$\begin{aligned} &\text{(i)} \quad \hat{x}_i^{(k+1)} = \frac{(b_i - \sum_{i\neq j}^{n} a_{ij}x_j^{(k)})}{a_{ii}} \ , \ i=1,2,\ldots,n \\ &\text{(ii)} \quad \underline{x}_i^{(k+1)} = (1-\omega)\underline{x}_i^{(k)} + \omega\hat{\underline{x}}_i^{(k+1)} \ , \quad i=1,2,\ldots,n. \end{aligned} \tag{2.81}$$

The Successive Overrelaxation (SOR) Method

This method is closely related to the point Gauss-Seidel method. Instead of equation (2.74), an acceleration is effected after each line as,

$$\begin{aligned} x_i^{(k+1)} &= x_1^{(k)} + \omega\Big(\frac{(b_i - \sum_{j=1}^{1-1} a_{ij}x_j^{(k+1)} - \sum_{j=1+1}^{n} a_{ij}x_j^{(k)}}{a_{ii}} - x_1^{(k)}\Big) \\ &= (1-\omega)x_i^{(k)} + \omega \frac{(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)})}{a_{ii}} \end{aligned} \tag{2.82}$$

which may be expressed in matrix form as,

$$\underline{x}^{(k+1)} = (1-\omega)\underline{x}^{(k)} + \omega D^{-1}(\underline{b}-L\underline{x}^{(k+1)}-U\underline{x}^{(k)}) \quad , \tag{2.83}$$

or

$$\underline{x}^{(k+1)} = (I+\omega D^{-1}L)^{-1}[(1-\omega)I+\omega D^{-1}U]\underline{x}^{(k)}+\omega(I+\omega D^{-1}L)^{-1}D^{-1}\underline{b}$$

i.e. $$\underline{x}^{(k+1)} = M_{SOR}\,\underline{x}^{(k)} + \underline{c}_{SOR} \quad .$$

Here $$M_{SOR} = (I+\omega D^{-1}L)^{-1}[(1-\omega)I+\omega D^{-1}U]$$

and $$\underline{c}_{SOR} = (I+\omega D^{-1}L)^{-1}\omega D^{-1}\underline{b} \quad , \tag{2.84}$$

where M_{SOR} is called the successive overrelaxation (SOR) iteration matrix and k is the iteration index.

2.8.4 CONVERGENCE OF POINT ITERATIVE METHODS

We shall now discuss the convergence rates for these iterative methods. Converence is the property that the error,

$$\underline{\varepsilon}^{(k)} = \underline{x} - \underline{x}^{(k)} \quad , \tag{2.85}$$

possesses (where $\underline{x}$ is the exact solution of $A\underline{x}=\underline{b}$) as it tends to zero as $k\to\infty$. The analysis of convergence is an important concern, because there is no a priori indication that any of these methods should converge at all. Moreover, we shall see that the rate of convergence for methods with acceleration depends (as expected) on the acceleration factor ω.

A relation between the error in successive approximations can be derived by subtracting from equation (2.70), the equation,

$$\underline{x} = M\underline{x} + \underline{c} \quad , \tag{2.86}$$

from which we obtain the result,

$$\underline{\varepsilon}^{(k+1)} = M\underline{\varepsilon}^{(k)} \quad . \tag{2.87}$$

Therefore, using (2.87) successively we obtain,

$$\begin{aligned} \underline{\varepsilon}^{(k)} &= M\underline{\varepsilon}^{(k-1)} \\ &= M^2\underline{\varepsilon}^{(k-2)} \\ &\vdots \\ &= M^k\underline{\varepsilon}^{(0)} \quad , \end{aligned} \tag{2.88}$$

where $\underline{\varepsilon}^{(0)} = \underline{x}-\underline{x}^{(0)}$, and $\underline{x}^{(0)}$ is an arbitrary known set of initial values. The sequence of iterative values $\underline{x}^{(1)},\underline{x}^{(2)},\ldots,\underline{x}^{(k)}$, will converge to $\underline{x}$ as k tends to infinity if,

$$\lim_{k\to\infty} \underline{\varepsilon}^{(k)} = \underline{0} \ .$$

Since $\underline{x}^{(0)}$ and therefore $\underline{\varepsilon}^{(0)}$ is arbitrary it follows that the iteration will converge if and only if

$$\lim_{k\to\infty} M^k = \underline{0} \ . \tag{2.89}$$

Now, let the matrix M of order (n×n) have eigenvalues $\lambda_1,\lambda_2,\ldots,\lambda_n$ and assume that the corresponding eigenvectors $\underline{v}_1,\underline{v}_2,\ldots,\underline{v}_n$, are linearly independent. Then, we can expand the initial error as,

$$\underline{\varepsilon}^{(0)} = \alpha_1\underline{v}_1 + \alpha_2\underline{v}_2 + \ldots + \alpha_n\underline{v}_n = \sum_{i=1}^{n} \alpha_i\underline{v}_i \ , \tag{2.90}$$

where α_i; i=1,2,...,n are scalars, and thus,

$$\underline{\varepsilon}^{(1)} = M\underline{\varepsilon}^{(0)} = \sum_{i=1}^{n} \alpha_i M\underline{v}_i \quad ,$$

but $M\underline{v}_i = \lambda_i\underline{v}_i$, where λ_i is the eigenvalue corresponding to $\underline{v}_i$, therefore,

$$\underline{\varepsilon}^{(1)} = \sum_{i=1}^{n} \alpha_i\lambda_i\underline{v}_i \ .$$

Similarly, we have,

$$\underline{\varepsilon}^{(k)} = \sum_{i=1}^{n} \alpha_i\lambda_i^k\underline{v}_i \ . \tag{2.91}$$

From this it follows that the iteration will converge from an arbitrary

initial vector $\underline{x}^{(0)}$ if and only if the eigenvalues of M satisfy,

$$|\lambda_i| < 1 \ , \quad i=1,2,\ldots,n.$$

Theroem (2.10)

A sufficient condition for the iterative method $\underline{x}^{(k+1)} = M\underline{x}^{(k)}+\underline{c}$ to converge is that,

$$||M|| < 1 \ .$$

Proof:

Since $M\underline{v}_i = \lambda_i \underline{v}_i$,

we have $$||M\underline{v}_i|| = ||\lambda_i \underline{v}_i|| = |\lambda_i| \ ||\underline{v}_i||.$$

But for any matrix norm that is compatible with a vector norm $||\underline{v}_i||$ we have,

$$||M\underline{v}_i|| \leq ||M|| \ ||\underline{v}_i|| \ .$$

Therefore,

$$|\lambda_i| \ ||\underline{v}_i|| \leq ||M|| \ ||\underline{v}_i||$$

so,

$$|\lambda_i| \leq ||M|| \ . \tag{2.92}$$

It follows from (2.92) that a sufficient condition for convergence is that $||M||<1$. It is not a necessary condition because the norm of M can exceed unity, even when $\rho(M)<1$.

Theorem (2.11) [VARGA, 1962]

Let A be a strictly or irreducibly diagonally dominant complex matrix of order n. Then, both the associated point Jacobi and point Gauss-Seidel iteration matrices are convergent, and the Jacobi iterative method and Gauss-Seidel iterative methods for the problem $A\underline{x}=\underline{b}$ are convergent for any arbitrary initial approximation vector $\underline{x}^{(0)}$.

2.8.5 RATE OF CONVERGENCE

We now study the rate of convergence of a convergent linear point iterative method. Since even if the iterative method converges it may converge too slowly to be of any practical value. Therefore, it is essential to determine the effectiveness of each method. To accomplish this, assume that the eigenvalues of the iteration matrix M are of decreasing order as follows:

$$|\lambda_1| > |\lambda_2| \geq \ldots \geq |\lambda_n| , \tag{2.93}$$

and that the matrix M has n linearly independent eigenvectors, $\underline{v}_1, \underline{v}_2, \ldots, \underline{v}_n$, namely.

Now equation (2.91) i.e.,

$$\underline{\varepsilon}^{(k)} = \sum_{i=1}^{n} \alpha_i \lambda_i^k \underline{v}_i ,$$

can be rewritten as,

$$\underline{\varepsilon}^{(k)} = \lambda_1^k (\alpha_1 \underline{v}_1 + (\frac{\lambda_2}{\lambda_1})^k \alpha_2 \underline{v}_2 + \ldots + (\frac{\lambda_n}{\lambda_1})^k \alpha_n \underline{v}_n) . \tag{2.94}$$

For large values of k we have that,

$$\underline{\varepsilon}^{(k)} \cong \lambda_1^k \alpha_1 \underline{v}_1 , \tag{2.95}$$

similarly,

$$\underline{\varepsilon}^{(k+1)} \cong \lambda_1^{(k+1)} \alpha_1 \underline{v}_1 , \tag{2.96}$$

so,

$$\underline{\varepsilon}^{(k+1)} \cong \lambda_1 \underline{\varepsilon}^{(k)} . \tag{2.97}$$

If the ith component of $\underline{\varepsilon}^{(k)}$ is denoted by $\varepsilon_i^{(k)}$, it is seen that

$$\frac{|\varepsilon_i^{(k)}|}{|\varepsilon_i^{(k+1)}|} \cong \frac{1}{|\lambda_1|} = \frac{1}{\rho(M)} .$$

Hence, $\ell n \frac{1}{\rho(M)} = - \ell n \rho(M)$ gives an indication of the number of decimal digits by which the error is eventually decreased by each convergent iteration. Since, for convergence, $0 < \rho(M) < 1$, the number of decimal

digits of accuracy gained per iteration increases as $\rho(M)$ decreases. Alternatively, for large k, $\underline{\varepsilon}^{(k)} \cong \lambda_1 \underline{\varepsilon}^{(k-1)}$, therefore,

$$\underline{\varepsilon}^{(k+p)} \cong \lambda_1 \underline{\varepsilon}^{(k+p-1)} \cong \ldots \cong \lambda_1^p \underline{\varepsilon}^{(k)} \quad , \; p=1,2,\ldots \; .$$

Hence, if we want to reduce the size of the error by a factor 10^{-q}, say, then the number of iterations needed to do this will be the least value of p for which,

$$|\lambda_1^p| = (\rho(M))^p \leq 10^{-q} \quad .$$

Hence,

$$p \geq q/-\ell n(\rho(M)) \quad , \tag{2.98}$$

which shows that p decreases as $-\ell n(\rho(M))$ increases. Clearly, the number $-\ell n(\rho(M))$ provides a measure for the comparison of the rates of convergence of different iterative methods when k is sufficiently large. For this reason $-\ell n(\rho(M))$ is defined to be the asymptotic rate of convergence and is denoted by $R_\infty(M)$.

The average rate of convergence $R_k(M)$ after k iterations is defined by the quantity,

$$R_k(M) = -\frac{1}{k} \ell n||M^k|| \quad . \tag{2.99}$$

It can be proved that [VARGA, 1962] the asymptotic rate of convergence,

$$R_\infty(M) = \lim_{k\to\infty} R_k(M) \quad , \tag{2.100}$$

and that the number of iterations required to reduce the error $||\underline{\varepsilon}^{(k)}||$, to $||\underline{\varepsilon}^{(0)}||/\alpha$, for sufficiently large k is $\geq(-\ell n\alpha)/R_n(M)$, [YOUNG, 1971].

Theorem (2.12) [YOUNG, 1971]

Let the matrix A be irreducible with weak diagonal dominance, then,

(i) The Jacobi method converges, and the JOR method converges for $0 < \omega \leq 1$.

(ii) Both the Gauss-Seidel and the SOR methods converges for $0 < \omega \leq 1$.

Theorem (2.13) [YOUNG, 1971]

Let A be a symmetric matrix with positive diagonal elements then the SOR method converges if and only if A is positive definite and $0<\omega<2$.

Theorem (2.14)

For the iteration matrix in the SOR method, we have

$$\rho(M_{SOR}) \geq |\omega-1|$$

so the method only converges for $0<\omega<2$.

Proof:

Since the determinant of a triangular matrix is the product of its diagonal elements, and $(I+\omega D^{-1}L)^{-1}$ and $[(1-\omega)I+\omega D^{-1}U]$ are both triangular matrices, hence we obtain,

$$\det(M_{SOR}) = \det(I+\omega D^{-1}L)^{-1}\det[(1-\omega)I+\omega D^{-1}U]$$

if we use the standard result from matrix theory that the product of the eigenvalues of a matrix is equal to its determinant; if the eigenvalues of M_{SOR} are denoted by $\lambda_1,\lambda_2,\ldots,\lambda_n$, then,

$$\begin{aligned}\lambda_1\cdot\lambda_2\cdot\ldots\cdot\lambda_n &= \det(M_{SOR}) \\ &= \det(I+\omega D^{-1}L)^{-1}\det[(1-\omega)I+\omega D^{-1}U] \qquad (2.101)\end{aligned}$$

Both matrices appearing in the above are triangular (the inverse of a triangular matrix is also a triangular matrix); and the determinant of a triangular matrix is equal to the product of its diagonal elements. Thus,

$$\lambda_1\cdot\lambda_2\cdot\ldots\cdot\lambda_n = (1-\omega)^n ,$$

whence,

$$\max_i |\lambda_i| \geq |1-\omega| , \quad 0<\omega<2 , \qquad (2.102)$$

which proves the theorem.

In practice ω usually lies between 1 and 2, and the optimum ω denoted by ω_{opt}, for the maximum rate of convergence is given by [YOUNG, 1971],

$$\omega_{opt} = \frac{2}{1+\sqrt{1-\rho^2(M)}} \quad , \tag{2.103}$$

where $\rho(M)$ is the eigenvalue of largest modulus of the Jacobi matrix M.

2.9 SOLUTION OF THE EIGENVALUE PROBLEM

When the finite element method is applied to the solution of the eigenvalue problem, we obtain the following algebraic eigenvalue problem,

$$\underline{Ax} = \lambda \underline{x} \quad , \tag{2.104}$$

where A is a given (n×n) matrix, λ is a scalar (called the eigenvalue or characteristic value) of the matrix A and $\underline{x}$ is a column vector with n components (called the eigenvector).

There are in general two types of methods for solving eigenvalue problems,

(i) methods which make use of similarity transformations which are commonly referred to as direct methods or transformation methods, such as Jacobi, Given's and Householder's.

(ii) iterative methods where, an arbitrary initial approximation to the eigenvector corresponding to the dominant eigenvalue (eigenvalue which is largest in modulus) or the smallest eigenvalue is successively improved until some required precision is reached. The iterative methods are most useful in the treatment of large sparse matrices when good estimates of the eigenvectors are available.

We shall concentrate our attention on the iterative methods.

(1) The Power Method

The Power method is a well-known iterative procedure for finding the largest eigenvalue (λ_1), along with the corresponding eigenvector.

Let us consider an (n×n) matrix A, whose eigenvalues are ordered so that,

$$|\lambda_1| = |\lambda_2| = \ldots = |\lambda_r| > |\lambda_{r+1}| \geq \ldots \geq |\lambda_n| . \tag{2.105}$$

By assuming there exists n linearly independent eigenvectors $\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_n$ and any arbitrary vector $\underline{z}^{(0)}$ can be expressed in the form,

$$\underline{z}^{(0)} = \sum_{i=1}^{n} \alpha_i \underline{x}_i \quad , \tag{2.106}$$

where α_i are scalars, not all zero.

Let us define the iterative scheme given by,

$$\underline{z}^{(k)} = A\underline{z}^{(k-1)} \quad , \quad k=1,2,\ldots \tag{2.107}$$

where $\underline{z}^{(0)}$ is an arbitrary vector. Then,

$$\begin{aligned} \underline{z}^{(k)} &= A\underline{z}^{(k-1)} \\ &= A^2\underline{z}^{(k-2)} \\ &\vdots \\ &= A^k\underline{z}^{(0)} \\ &= \sum_{i=1}^{n} \alpha_i \lambda_i^k \underline{x}_i \quad , \end{aligned} \tag{2.108}$$

where we have used equation (2.106).

Now since $\alpha_1, \alpha_2, \ldots, \alpha_n$ are not all zero, the right-hand side of equation (2.108) will be dominated by the terms,

$$\sum_{i=1}^{r} \alpha_i \lambda_i^k \underline{x}_i \ .$$

If r=1, and we assume that $\alpha_1 \neq 0$ we have,

$$\underline{z}^{(k)} = \lambda_1^k \{\alpha_1 \underline{x}_1 + \sum_{i=2}^{n} \alpha_i (\lambda_i/\lambda_1)^k \underline{x}_i\} \tag{2.109}$$

$$= \lambda_1^k \{\alpha_1 \underline{x}_1 + \underline{\varepsilon}_k\} \tag{2.110}$$

for sufficiently large k, where $\underline{\varepsilon}_k$ is a vector with very small components when k is so large that $\underline{\varepsilon}$ is negligible to the required precision, it follows that $\underline{z}^{(k)}$ is an approximation to the un-normalized vector $\underline{x}_1$.

This forms the basis for the simple power method for computing the

dominant eigenvalue.

Now since $\underline{z}^{(k+1)} = \lambda_1^{k+1}\{\alpha_1\underline{x}_1 + \underline{\varepsilon}^{(k+1)}\}$, then for the ith component of $\underline{z}^{(k)}$, we have,

$$\frac{(\underline{z}^{(k+1)})_i}{(\underline{z}^{(k)})_1} = \lambda_1 \left\{\frac{\alpha_1(\underline{x}_1)_1 + \underline{\varepsilon}^{(k+1)}}{\alpha_1(\underline{x}_1)_i + \underline{\varepsilon}^{(k)}}\right\} \to \lambda_1 \text{ as } k\to\infty. \qquad (2.111)$$

The rate of convergence will depend on the constant α_i, but more essentially on the ratios,

$$\left|\frac{\lambda_2}{\lambda_1}\right|, \left|\frac{\lambda_3}{\lambda_1}\right| , \ldots, \left|\frac{\lambda_n}{\lambda_1}\right| ,$$

from which it follows that the smaller these ratios are, the faster will be convergence. In particular, if $\left|\frac{\lambda_2}{\lambda_1}\right|$ is close to unity then the convergence is likely to be very slow.

In order to keep the elements of $\underline{z}^{(k)}$ within reasonable bounds during computation to prevent overflow, it is usual to normalise the vector at each iteration by dividing all its elements by the element of largest modulus, the sequence of normalising factors then converges to λ_1. That is, the elements $\underline{x}^{(k)}$ are scaled at each step, and equation (2.107) is replaced by the pair of equations

$$\underline{y}^{(k)} = A\underline{z}^{(k-1)}$$

$$\underline{z}^{(k)} = \frac{\underline{y}^{(k)}}{||\underline{y}^{(k)}||_\infty} .$$

In this case,

$$\underline{z}^{(k)} \to \frac{\underline{x}_1}{||\underline{x}_1||_\infty} ,$$

and

$$||\underline{y}^{(k)}||_\infty \to \lambda_1 , \text{ as } k\to\infty .$$

Now suppose that r>1, and that equation (2.105) is satisfied with

$$\lambda_1 = \lambda_2 = \ldots = \lambda_r .$$

Then we have,

$$\underline{z}^{(k)} = \lambda_1^k \{ \sum_{i=1}^{r} \alpha_i \underline{x}_i + \sum_{i=r+1}^{n} \alpha_i (\lambda_i/\lambda_1)^k \underline{x}_1 \} \qquad (2.112)$$

$$= \lambda_1^k \{ \sum_{i=1}^{r} \alpha_i \underline{x}_i + \underline{\varepsilon}^{(k)} \} \quad , \qquad (2.113)$$

for sufficiently large k, where again $\underline{\varepsilon}^{(k)}$ is a vector with very small elements.

Thus, the convergence of the power method is not affected, and the iterates $\underline{z}^{(k)}$ tend to a vector which is some linear combination of the eigenvectors corresponding to λ_1. Thus, the power method will only supply one eigenvector corresponding to a multiple dominant eigenvalue, for each $\underline{z}^{(0)}$.

However, the iterative procedure breaks down, if there are a number of unequal eigenvalues of the same modulus.

This breakdown is characterized by the failure of the iterates to converge, and by the changes in sign of the approximation to λ_1, (see A.R. GOURLAY, 1973).

(2) The Inverse Power Method

The other most powerful methods available in connection with solving matrix eigenproblems is the technique, known as *inverse iteration*. This method is not only of general use in that it may be applied to the computation of an eigenvalue and/or an eigenvector, but it also possesses a fast rate of convergence.

A direct iteration of the form,

$$\underline{y}^{(k+1)} = B\underline{z}^{(k)}$$

$$\underline{z}^{(k+1)} = \frac{\underline{y}^{(k+1)}}{||\underline{y}^{(k+1)}||_\infty} \qquad (2.114)$$

gives under suitable conditions a convergent sequence of values approximating to the dominant eigenvalue of B, and its associated eigenvector. The process defined by,

$$A\underline{y}^{(k+1)} = \underline{z}^{(k)}$$

$$\underline{z}^{(k+1)} = \underline{y}^{(k+1)} / ||y^{(k+1)}||_{\infty} , \tag{2.115}$$

is equivalent to (2.114), but with the matrix $B = A^{-1}$. Thus, the sequence (2.115) will converge to the eigenvalues of A of smallest modulus.

To show this, assume,

$$\underline{z}^{(0)} = \sum_{i=1}^{n} \hat{\alpha}_i \underline{x}_i \quad . \tag{2.116}$$

Then equation (2.115) gives,

$$\underline{z}^{(k)} = T_k \sum_{i=1}^{n} \hat{\alpha}_i \lambda_i^{-k} \underline{x}_i \quad , \tag{2.117}$$

where T_k is a scaling factor introduced by the rescaling part of equation (2.115).

The vector $\underline{z}^{(k)}$ is richest in the vector $\underline{x}_n$ corresponding to the smaller eigenvalue λ_n.

Therefore the sequence $\underline{z}^{(k)}$ will tend to a multiple of $\underline{x}_n$ as $k \to \infty$ and also, for each j in general

$$\frac{z_j^{(k+1)}}{z_j^{(k)}} \to \frac{1}{\lambda_n} \quad , \text{ as } k \to \infty \ .$$

The process (2.115) is known as the inverse iteration in its simplest form.

2.10 THE SOLUTION OF NON-LINEAR EQUATIONS

The finite element analysis of any physical or engineering problem leads to a system of matrix equations, some of the methods available for solving the final system were presented in Section (2.7) and (2.8). It is to be noted that the problem has to be linear in order to apply the solution of Section (2.7) and (2.8).

If the problem is non-linear, the resulting matrix equations will also be non-linear irrespective of the type of the problem (elliptic, eigenvalue or parabolic problem), and some sort of iterative procedure has to be used for finding the solution.

We shall here be brief and mention just a few of the more recent important contributions in this area. Perhaps the most important is the book by Ortega and Rheinboldt (1970) which gives detailed and some practical considerations of the solution of a set of nonlinear equations.

We consider in this section the problem of finding a solution of a fixed point (stationary point) $\underline{x}^* = (x_1^*, x_2^*, \ldots, x_n^*)^T$, of the system of n nonlinear equations,

$$\begin{aligned} f_1(x_1, x_2, \ldots, x_n) &= 0 , \\ f_2(x_1, x_2, \ldots, x_n) &= 0 , \\ &\vdots \\ f_n(x_1, x_2, \ldots, x_n) &= 0 , \end{aligned} \tag{2.118}$$

which can be written as,

$$\underline{f}(\underline{x}) = \underline{0} , \tag{2.119}$$

where $\underline{x}$ is an n-dimensional column vector with component $x_1, x_2, \ldots, x_n$ and

$\underline{f}(\underline{x})$ is an n-dimensional vector valued function, i.e., a column vector with components $f_1(\underline{x}), f_2(\underline{x}), \ldots, f_n(\underline{x})$.

We shall assume the existence of $\underline{x}^*$ and also that some initial supplied approximation $\underline{x}^{(0)}$ to $\underline{x}^*$ is available.

Most of the methods to be described are iterative methods which generate a sequence of points,

$\underline{x}^{(1)}, \underline{x}^{(2)}, \ldots$ say, or $\{\underline{x}^{(k)}\}$ (the superscripts denoting iteration number)

hopefully converging to a fixed point $\underline{x}^*$ which is the solution to the problem. If the problem functions which arise are smooth, that is continuous and continuously differentiable (c^1). Therefore, for a function $f(\underline{x})$ at any point $\underline{x}$ there is a vector of first partial derivatives, or gradient vector,

$$\begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix} = \nabla f(\underline{x}) \ , \qquad (2.120)$$

where ∇ denotes the gradient operator $(\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \ldots, \frac{\partial}{\partial x_n})^T$.

If $f(\underline{x})$ is twice continuously differentiable (c^2) then there exists a matrix of second partial derivatives or Hessian matrix, written $\nabla^2 f(\underline{x})$, for which the i,jth element is $\frac{\partial^2 f}{\partial x_i \partial x_j}$.

This matrix is square and symmetric, since any column (the jth say), is $\nabla(\frac{\partial f}{\partial x_j})$, the matrix can strictly be written as $\nabla(\nabla f^T(\underline{x})$. For example, $f(\underline{x}) = 100(x_2 - x_1^2) + (1-x_1)^2$, gives

$$\nabla f(\underline{x}) = \begin{bmatrix} -400x_1(x_2-x_1^2)-2(1-x_1) \\ 200(x_2-x_1^2) \end{bmatrix} ,$$

$$\nabla^2 f(\underline{x}) = \begin{bmatrix} 1200x_1^2-400x_2+2, & -400x_1 \\ -400x_1 , & 200 \end{bmatrix} ,$$

and this illustrates that ∇f and $\nabla^2 f$ will in general depend upon $\underline{x}$, and vary from point to point.

The iterative methods which will be discussed to solve (2.119) will have the following states:

(a) Initialisation: user supplied approximation $\underline{x}^{(0)}$,

(b) Iteration: $\underline{x}^{(k+1)} = \phi(\underline{x}^{(k)})$, $k=0,1,2,\ldots$,

(c) Termination: convergence criterion for (b).

The system (2.119) arises during the numerical solution of elliptic and parabolic partial differential equations. Such a system of nonlinear equations arises, for example in the finite element method via a minimum variational principle (or via a Galerkin approximation), of the elliptic equation,

$$\nabla^2 u = u^2 .$$

This equation gives the system of n nonlinear equations,

$$\underline{AU} = \underline{F}(\underline{U}) , \tag{2.121}$$

which system can be written as,

$$\underline{f}(\underline{U}) = \underline{AU} - \underline{F}(\underline{U}) = \underline{0} . \tag{2.122}$$

It should be noted that,

(i) $\underline{f}(\underline{U})$ consists of a linear and nonlinear part, as might be expected.

(ii) The matrix A is symmetric, positive definite, banded and sparse.

(iii) Each function $F_i(\underline{U})$ depends on a small number (determined by the index i of the variables U_j.

We shall consider how each property can be used to advantage in the numerical solution of (2.122).

2.10.1 FUNCTIONAL ITERATION

One of the form in which the nonlinear equations may appear is

$$\underline{x} = \underline{g}(\underline{x}) \ , \tag{2.123}$$

where $\underline{g}$ is a nonlinear vector function. The simplest procedure for finding a solution of this system of equations is known as functional iteration or fixed point iteration. It proceeds as follows. From some initial guess $\underline{x}^{(0)}$ at the solution, the sequence of iterates $\{\underline{x}^{(k)}, k=0,2,...\}$ is defined by the relation,

$$\underline{x}^{(k+1)} = \underline{g}(\underline{x}^{(k)}) \ . \tag{2.124}$$

The convergence of this procedure is governed by the contracting mapping theorem.

Theorem (2.14)

If $g(\underline{x})$ satisfies,

$$||g(\underline{x})-g(\underline{y})|| \leq \lambda||\underline{x}-\underline{y}|| \ , \tag{2.125}$$

for all vectors $\underline{x},\underline{y}$ such that $||\underline{x}-\underline{x}^{(0)}|| \leq \rho$, $||\underline{y}-\underline{x}^{(0)}|| \leq \rho$ with the Lipschitz constant, λ, satisfying,

$$0 \leq \lambda < 1 \ .$$

Let the initial iterate, $\underline{x}^{(0)}$ satisfy,

$$||g(\underline{x}^{(0)})-\underline{x}^{(0)}|| \leq (1-\lambda)\rho,$$

then (i) all iterates (2.124) satisfy,

$$||\underline{x}^{(k)}-\underline{x}^{(0)}|| \leq \rho \ ;$$

(ii) the iterates converge to some vector, say

$$\lim_{k\to\infty} \underline{x}^{(k)} = \underline{x}^* ,$$

which is the root of (2.118).

(iii) $\underline{x}^*$ is the only root of (2.123) in $||\underline{x}-\underline{x}^{(0)}|| \leq \rho$

The formal proof may be found in (Isaacson and Keller, 1966).

Now (2.121) can be written in the form (2.123) as,

$$\underline{U} = A^{-1}\underline{F}(\underline{U}) .$$

The method of functional iteration for solving (2.122) then consists of the iteration,

$$\underline{U}^{(k+1)} = A^{-1}\underline{F}(\underline{U}^{(k)}) , \quad k=0,1,\ldots,$$

or

$$A\underline{U}^{(k+1)} = \underline{F}(\underline{U}^{(k)}) , \quad k=0,1,\ldots . \qquad (2.126)$$

The iteration process (2.126) requires the solution of a sequence of $(n\times n)$ linear systems with a constant matrix A and a varying right hand side vector $\underline{F}(\underline{U}^{(k)})$, $k=0,1,\ldots$.

This simple form results from using property (i) of the system (2.122). We can also use property (ii) to factorize A as,

$$A = LDL^T ,$$

where L is a unit diagonal, lower triangular matrix, and D is a diagonal matrix. Then, each iteration of (2.126) consists of the evaluation of $F(U^{(k)})$ and forward and backward substitutions to give $\underline{U}^{(k+1)}$.

2.10.2 NEWTON'S METHOD

A particularly effective procedure for solving (2.119) is known as Newton's method which makes use of the iteration,

$$\underline{x}^{(k+1)} = \underline{x}^{(k)} + \underline{p}^{(k)} , \quad k=0,1,\ldots \qquad (2.127)$$

where $\underline{p}^{(k)} = -J(\underline{x}^{(k)})^{-1}\underline{f}(\underline{x}^{(k)})$, (2.128)

and $J(\underline{x})$ is the $(n \times n)$ Jacobian matrix of $\underline{f}(\underline{x})$ with components

$$J_{ij}(\underline{x}) = \frac{\partial f_i(\underline{x})}{\partial x_j} \quad , \tag{2.129}$$

There are many theorems concerning the convergence and its rate for Newtons Method (Ortega and Rheinboldt, 1970) gives conditions on $\underline{f}(\underline{x})$, $J(\underline{x})$ and $\underline{x}^{(0)}$ which guarantee the convergence of the iteration (2.127). Of course, the computation is not carried out in the form (2.127), but rather by solving the system of linear equations,

$$J(\underline{x}^{(k)})(\underline{x}^{(k+1)} - \underline{x}^{(k)}) = -\underline{f}(\underline{x}^{(k)}) \quad , \tag{2.130}$$

at each step of the iteration.

The justification for Newton's method is taken from Taylor's theorem, where the Taylor expansion of $\underline{f}$ in a point $\underline{x}^{(k)}$ that lies in the neighbourhood of a solution $\underline{x}^*$ may be expressed as,

$$\underline{f}(\underline{x}^*) = \underline{f}(\underline{x}^{(k)}) + J(\underline{x}^{(k)})(\underline{x}^* - \underline{x}^{(k)}) + \text{higher order terms}, \tag{2.131}$$

But since $\underline{x}^*$ is a root to $\underline{f}(\underline{x}^*) = \underline{0}$, whence

$$\underline{0} \cong \underline{f}(\underline{x}^{(k)}) + J(\underline{x}^{(k)})(\underline{x}^* - \underline{x}^{(k)}) \quad , \tag{2.132}$$

This approximation may be solved for the unknown $\underline{x}^*$, giving precisely the formula (2.127).

The basic Newton method as it stands is not suitable for a general purpose algorithm, since the Jacobian $J(x^{(k)})$ may not be positive definite when $\underline{x}^{(k)}$ is remote from the solution, therefore, a good initial estimate must be provided. Furthermore even if $J(\underline{x}^{(k)})$ is positive definite then convergence may not occur, in fact $f(\underline{x}^{(k)})$ may not even decrease. The latter possibility can be eliminated by using *Newton's method with a damping factor* in which correction is used to generate a direction of search,

$$\underline{x}^{(k+1)} = \underline{x}^{(k)} + \alpha^{(k)} \underline{p}^{(k)} \quad , \tag{2.133}$$

where $\underline{p}^{(k)} = -J(\underline{x}^{(k)})^{-1} \underline{f}(\underline{x}^{(k)})$, and $\alpha^{(k)}$ is chosen such that

$$||\underline{f}(\underline{x}^{(k+1)}|| < \quad ||\underline{f}(\underline{x}^{(k)})|| \tag{2.134}$$

with $0<\varepsilon\leq 1$.

There are many ways of choosing $\alpha^{(k)}$; one simple choice,

$$\alpha^{(k)} = \frac{1}{2^m} \quad , \quad m=0,1,\ldots$$

where m is the smallest integer such that,

$$||f(\underline{x}^{(k)}) + \frac{1}{2^m} \underline{p}^{(k)}|| \leq \varepsilon ||\underline{f}(\underline{x}^{(k)})|| \quad . \tag{2.135}$$

Other good choices of $\alpha^{(k)}$ are possible, see [GILL and MURRAY, 1974]. Provided the matrix $J(\underline{x}^{(k)})$ is not too large then Gaussian elimination with partial pivoting could be used to factorise $J(\underline{x}^{(k)})$ as

$$\underline{p}J(\underline{x}^{(k)}) = LU \quad ,$$

where p is a permutation matrix, L is a unit diagonal lower triangular matrix and U is an upper triangular matrix, and forward and backward substitution processes used to find $\underline{p}(\underline{x}^{(k)})$.

We note that each iteration of Newton's method requires the evaluation and factorization of the (n×n) Jacobian matrix $J(\underline{x}^{(k)})$.

Hence, in general, Newton's method requires more operations per iteration than the functional iteration of Section (2.10.2), although in comparison Newton's method converges at a second order rate which may be particularly useful if accurate results are required.

Here, we will give a Theorem which guarantees the convergence of Newton's method.

Theorem (2.15)

If $\underline{x}^{(k)}$ is sufficiently close to $\underline{x}^*$ for some k, and if the Jacobian matrix $J(\underline{x}^*)$ is positive definite, then Newton's method is well defined for all k, and converges at second order.

Proof:

It is assumed that $\underline{f}(\underline{x}) \in C^2$, and the elements of the Jacobian matrix satisfy a Lipschitz condition,

$$|J_{ij}(\underline{x}) - J_{ij}(\underline{y})| \leq \alpha ||\underline{x}-\underline{y}|| \quad . \tag{2.136}$$

Then the Taylor expansion of $\underline{f}(\underline{x}^{(k)}+\underline{h})$ about $\underline{x}^{(k)}$ is

$$\underline{f}(\underline{x}^{(k)}+\underline{h}) = \underline{f}(\underline{x}^{(k)}) + J(\underline{x}^{(k)})\underline{h} + O(||\underline{h}||^2) \quad , \tag{2.137}$$

with $\underline{h} = \underline{x}^{(k)} - \underline{x}^*$ and,

(if we use order notation, i.e. $F(\underline{x}) = O(\underline{h}(\underline{x}))$, means that,

$$|F(\underline{x})| \leq c\underline{h}(\underline{x})$$

$\underline{h} = -\underline{h}^{(k)}$ gives,

$$\underline{0} = f(\underline{x}^*) = f(\underline{x}^{(k)}) - J(\underline{x}^{(k)})\underline{h}^{(k)} + O(||\underline{h}^{(k)}||^2) \tag{2.138}$$

Let $\underline{x}^{(k)}$ be in a neighbourhood of $\underline{x}^*$ for which $J(\underline{x}^{(k)})$ is positive definite and $J(\underline{x}^{(k)})^{-1}$ is bounded above.

Such a neighbourhood exists by a continuity of $J(\underline{x})$. Then, the kth iteration exists and by multiplying through (2.138) by $J(\underline{x}^{(k)})^{-1}$ gives

$$\underline{0} = J(\underline{x}^{(k)})^{-1} f(\underline{x}^{(k)}) - \underline{h}^{(k)} + O(||\underline{h}^{(k)}||^2) \quad , \tag{2.139}$$

$$= -\underline{p}^{(k)} - \underline{h}^{(k)} + O(||\underline{h}^{(k)}||^2) \quad ,$$

$$= -\underline{h}^{(k+1)} + O(||\underline{h}^{(k)}||^2) \tag{2.140}$$

by definition of $\underline{h}^{(k+1)}$.

Hence by definition of O(.) there exists a constant c such that

$$||\underline{h}^{(k+1)}|| \leq c||\underline{h}^{(k)}||^2 \quad . \tag{2.141}$$

If $\underline{x}^{(k)}$ is in a closer neighbourhood for which $||\underline{h}|| \leq \alpha/c$, where $0<\alpha<1$, then it follows that,

$$||\underline{h}^{(k+1)}|| \leq \alpha||\underline{h}^{(k)}|| \ . \qquad (2.142)$$

Thus, $\underline{x}^{(k+1)}$ is in the neighbourhood, and by induction the iteration is well defined for all k and

$$||\underline{h}^{(k)}|| \to 0 \ .$$

Finally, the iteration converges and the rate is shown to be second order by (2.139).

The kth iteration of Newtons Method can be written

(a) calculate $\underline{f}(\underline{x}^{(k)})$ and,

(b) solve for $\underline{p}^{(k)}$ from $J(\underline{x}^{(k)})\underline{f}(\underline{x}^{(k)})$

(c) evaluate $\underline{x}^{(k+1)}$ from $\underline{x}^{(k+1)} = \underline{x}^{(k)} + \underline{p}^{(k)}$

(d) calculate $\underline{f}(\underline{x}^{(k+1)})$.

As the calculation of the Jacobian matrix is very expensive for some nonlinear problems, a variation of Newton's method exists in which the Jacobian matrix is not evaluated on every iteration, but the factors from a previous iteration are used in its place. This saves effort in carrying out the iteration, but slows down the overall rate of convergence.

Many modifications of Newton's method arise, especially when the Jacobian is not positive definite, or when convergence may not occur.

CHAPTER THREE

THE FINITE ELEMENT METHOD

3.1 THE BASIC PROBLEM

The general problem to be solved takes the form of a differential equation,

$$\left.\begin{aligned} \underline{Du} &= \underline{f} \ , \\ &\text{In some region R in the space (x,y),} \\ &\text{and subject to the condition,} \\ \underline{Lu} &= \underline{g} \text{ on the boundary } \partial R, \end{aligned}\right\} \qquad (3.1)$$

where $\underline{D}=[D_1,D_2,\ldots,D_r]^T$ are a set of differential operators which acts on the unknown $\underline{u}=\underline{u}(x,y)$ to generate the function $\underline{f}=[f_1,f_2,\ldots,f_r]^T$, and also $\underline{L}=[L_1,L_2,\ldots,L_r]^T$ are again differential operators which hold on the boundary ∂R of the domain R and $\underline{g}=[g_1,g_2,\ldots,g_r]^T$ is a given function as shown in Figure (3.1).

The unknown $\underline{u}$ may be a scalar or a vector of several quantities and similarly the differential equation (3.1) may be single or a set of simultaneous equations.

The finite element approximation U(x,y) which is made up of a linear combination of suitable functions and satisfies the given boundary conditions is given by,

$$U(x,y) = \sum_{i=1}^{n} N_i(x,y)U_i \ , \qquad (3.2)$$

where $N_i(x,y)$, (i=1,2,...,n) are "basis functions" or "shape functions" prescribed in terms of the independent variables x,y, and U_i (i=1,2,...,n) are known parameters. The aim of the method is

to determine the parameters U_i so that $U(x,y)$ in some sense is a good approximation to the true solution.

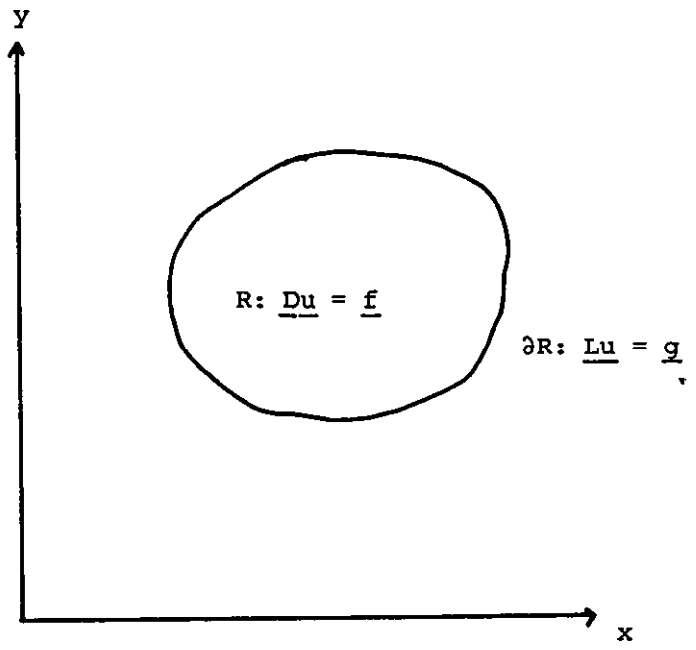


FIGURE 3.1

The general procedure to be adopted in the various stages of the finite element method is outlined in the following sections.

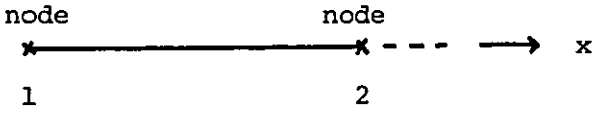
A description of the various alternative forms of the finite element method are given such as the variational principles (method which is based on Calculus of Variations), and also the weighted residuals method which is a more widely used technique and more general in its applications.

3.2 DISCRETIZATION PROCESSES

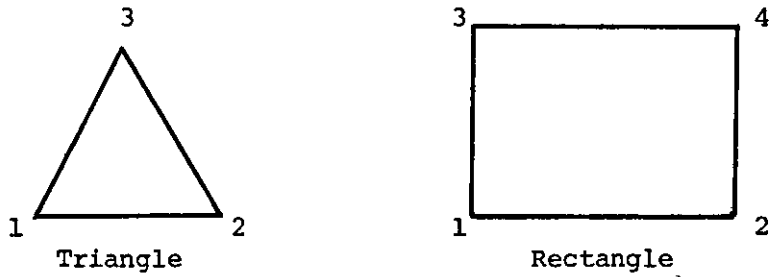
(1) Type of the Element

The discretization of the domain or solution region into a series of finite elements (subregions) is the first step in the finite element method. This is equivalent to replacing the domain having an infinite number of degrees of freedom by a system having a finite number of degrees of freedom. The number, shapes and sizes of the elements have to be chosen carefully such that the original domain is simulated as closely as possible with regard to the computational effort needed for the solution.

Mostly, the choice of the type of the element is dictated by the number of independent spatial coordinates necessary to describe the system. Some of the most popular used elements are one-two-three-dimensional straightside linear elements and are shown in Figures(3.2) and (3.3) below.

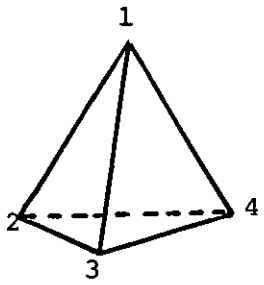


(a) One dimensional element with two nodes

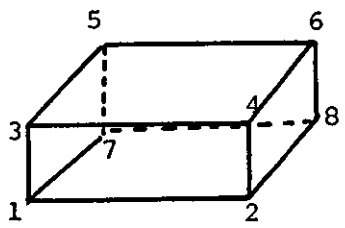


(b) Two dimensional elements

FIGURE 3.2



Tetrahedron

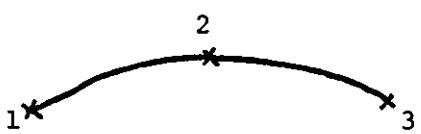


Rectangular Prism

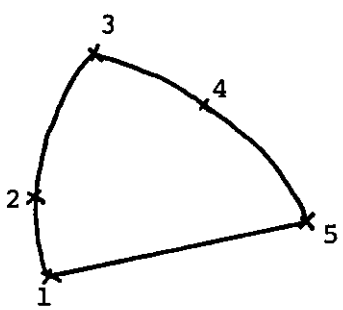
(c) Three dimensional elements

FIGURE 3.3

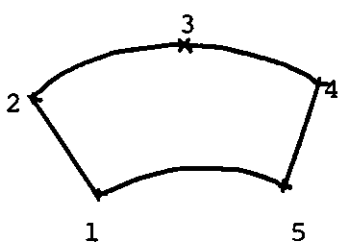
For the discretization of problems involving curved geometries, finite elements with curved sides are used. The ability to model curved boundaries has been made possible by the addition of mid-side nodes. Typical elements having curved boundaries are shown in Figure (3.4).



Curved-line-element



Plane triangle with curved sides



Annular element

FIGURE 3.4: Finite elements with curved boundaries

(2) Size and Number of the Elements

The size of the elements influences the convergence of the solution directly and hence it has to be chosen with care. If the size of the elements is small, the final solution is expected to be more accurate. However, we have to remember that the use of elements of smaller size will also mean more computational time. Sometimes, we may have to use elements of different sizes in the same domain. In general, whenever steep gradients of the solution region are expected, we have to use a finer mesh in those regions. Also the number of elements to be chosen for idealization is related to the accuracy desired, size of elements, and the number of degrees of freedom involved. Provided that 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 accuracy of our results. For any given problem, there will be a certain number of elements beyond which the accuracy cannot be improved by any significant amount. This behaviour is shown graphically in Figure (3.5). Moreover, since increasing the number of elements leads to higher computational expense, we may also have the added difficulty that we may not be able to store the resulting matrices in the available computer memory. When solving a particular type of problem for the first time, it is good practice to obtain several solutions with different numbers of elements. By comparing these results it is then possible to see whether enough elements are being used in the solution. A similar trial-and-error procedure is used for determining satisfactory mesh representation of domains of infinite extent. The procedure is to construct a finite mesh encompassing the regions of the solution domain where the phenomena are occurring. By comparing solutions obtained for meshes of increasing

extent, we can determine the point beyond which the location of the boundary no longer has significant effect on the solution.

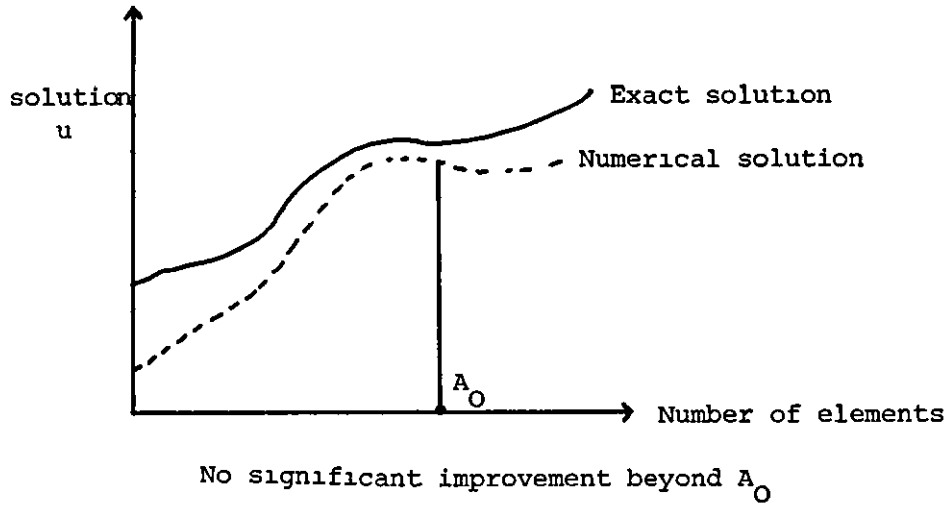


FIGURE 3.5: Effect of varying the number of elements

(3) Node Numbering Scheme

The solution of the finite element problem often leads to matrix equations in which the matrices involved will be banded. The reasons most often presented for reducing the bandwidth of a matrix are to reduce the storage and computation required to solve the system of equations. The advances in the finite element analysis of large systems have been made possible largely due to the banded nature of the matrices. Further, since most of the matrices involved are symmetric, the demands on the computer storage can be substantially reduced by storing only the elements involved in the half band width instead of storing the whole matrix.

The bandwidth of the finite element matrix depends mainly on the

node numbering scheme. If we can minimize the bandwidth, the storage requirements as well as the solution time can also be minimized.

The bandwidth can be minimized by using a proper node numbering scheme. For any finite element network we define the *bandwidth* as the largest difference in the node numbers occurring for all elements of the assembled system.

This indicates that the bandwidth can be minimized by reducing the differences in the node numbering that occur for all elements in the given region of solution.

As an example, consider the rectangular region, with 171 elements as shown in Figure (3.6). There are 200 unknowns in the final finite element system. If the entire matrix is stored in the computer it will require $(200)^2$ = 40,000 locations. The bandwidth overall is 20 and thus the storage required for the upper half bandwidth is only 20×200=4000 locations.

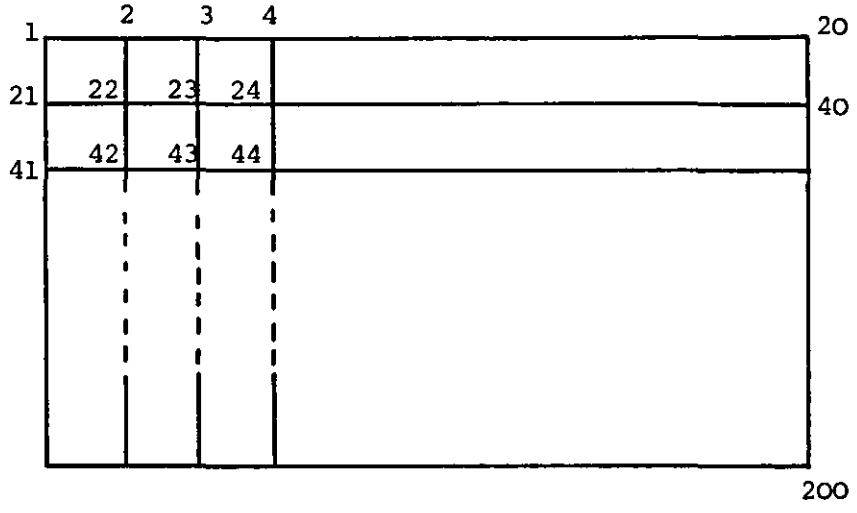


FIGURE (3.6): Numbering of the node of a rectangular region with bandwidth 20

A shorter bandwidth can be obtained simply by numbering the nodes across the shorter dimension of the region.

This is clear from Figure (3.7) where the numbering of the nodes along the shorter dimension produces a bandwidth equal to 10, and hence the storage required for the upper half band is only 10×200=2000 locations.

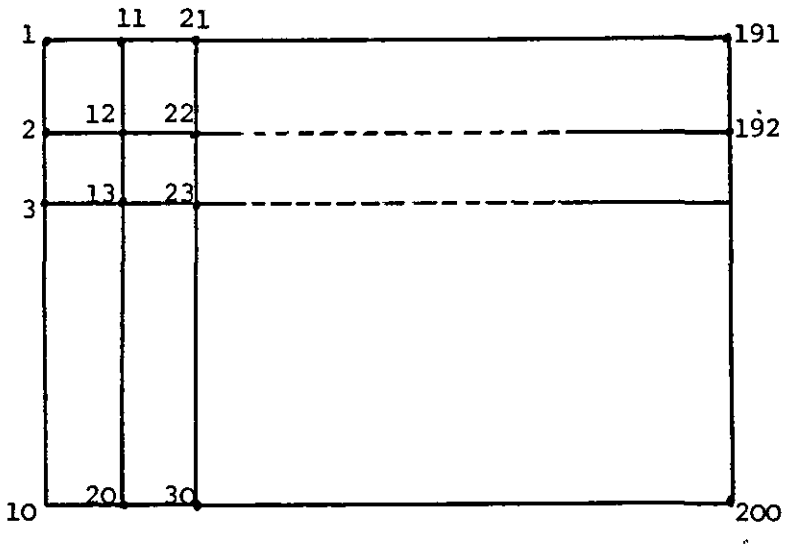


FIGURE (3.7): Numbering of the node along the shorter dimension with bandwidth 10

Several approaches are available for minimzing the bandwidth of the finite element systems of algebraic equations, we will describe here the Cuthill-Mckee algorithm for ordering the unknowns to produce a matrix with reasonably narrow bandwidth.

1. THE CUTHILL-MCKEE ALGORITHM

When the number of nodal variables is sufficiently large for the

storage space and computing time to be important, it is advisable to attempt to arrange the order of the variables so as to give an economical solution. An alternative procedure is to allow the variables to be specified in an arbitrary order within the input data, and to include an initial segment in the program which automatically rearranges the nodes numbering in a way that should give an efficient solution. Cuthill and Mckee's algorithm provides a simple scheme for renumbering the nodes of the finite element problem.

Before describing the algorithm we will summarize some of the notation to be used as well as some definitions of required terms from graph theory.

Consider the system of linear algebraic equations,

$$\underline{Ax} = \underline{b} \ ,$$

where A is an $(n \times n)$ sparse symmetric positive definite matrix. The elements of A will be designated a_{ij}, where i is a row index and j a column index.

Definition (3.1):

Let A be an $(n \times n)$ symmetric or lower triangular matrix with element a_{ij}. For the ith row of A, i=1,2,...,n, we define,

$$f_i(A) = \min\{J: a_{ij} \neq 0\}$$

that is $f_i(A)$ is the column subscript of the first non-zero element of the ith row of A

$$b_i(A) = 1 - f_i(A) \ ,$$

that is, $b_i(A)$ is the *band width* of the ith row of A.

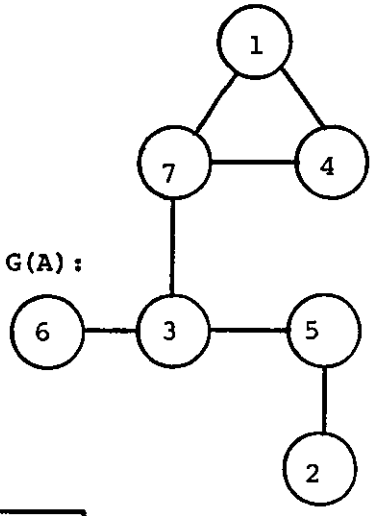
Then the bandwidth of A is given by,

$$B(A) = \max\{|i-j| : a_{ij} \neq 0\}$$

for example if,

$$A = \begin{bmatrix} X & O & O & X & O & O & X \\ O & X & O & O & X & O & O \\ O & O & X & O & X & X & X \\ X & O & O & X & O & O & X \\ O & X & X & O & X & O & O \\ O & O & X & O & O & X & O \\ X & O & X & X & O & O & X \end{bmatrix}$$

, G(A):



i	$f_i(A)$	$b_i(A)$
1	1	0
2	2	0
3	3	0
4	1	3
5	2	3
6	3	3
7	1	6

$$B(A) = 6$$

and note that,

$$B(A) = \max_i b_i(A) .$$

Definition (3.2)

For a graph G(A) corresponding to the matrix A we will have n nodes labelled, i=1,2,...,n. For each non-zero element a_{ij}, i<j of A there will be an edge connecting nodes i and j. From the graph of A we can determine the position of all off-diagonal non-zero elements of A.

Definition (3.3)

Any two nodes of G(A) are said to be connected if there is a

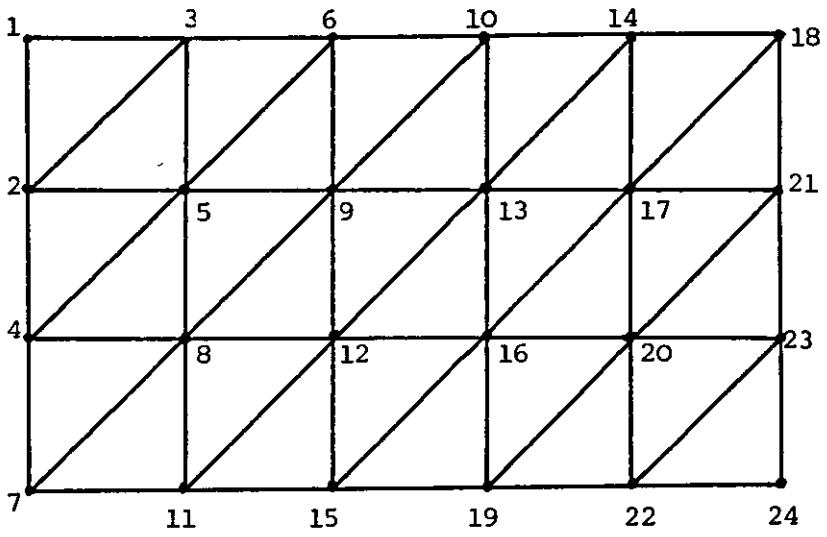
sequence of edges joining them such that consecutive edges have a common end point. Two nodes of G(A) are said to be adjacent if they are connected by an edge.

A graph G(A) is said to be *connected* if every pair of nodes of the graph are connected. If G(A) is connected, the corresponding matrix is irreducible.

A popular ordering strategy is the Cuthill-Mckee algorithm. It attempts to find a permutation matrix P for which PAP^T has a small bandwidth, when we permute the rows and columns of A using the permutation matrix P generating PAP^T, the graph of PAP^T - namely $G(PAP^T)$, is identical to A but the node labels have been permuted according to the permutation matrix P.

The procedure presented here for determining P or equivalently a renumbering scheme for G(A) is given as follows:

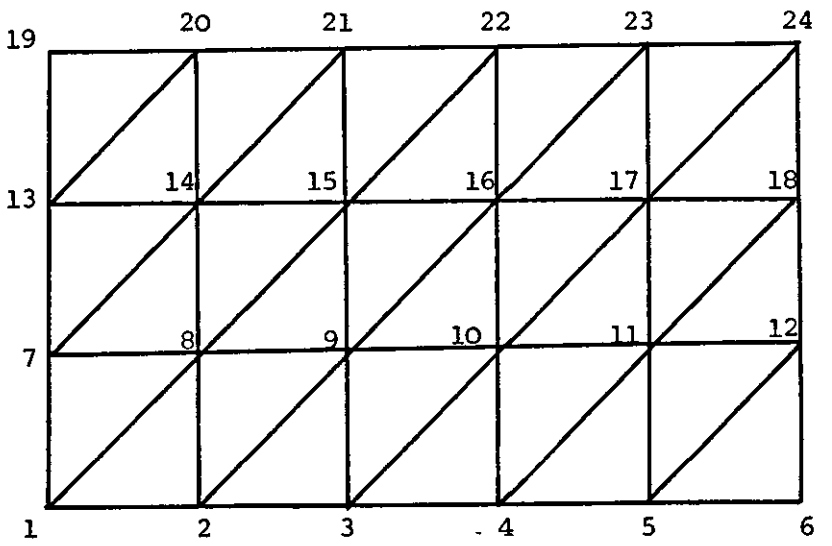
a. select a node to be relabelled 1, this node should be located at an extremity of the graph and should have, if possible a few connections to other sides.
b. the nodes adjacent to this node are numbered in sequence beginning with 2 in the order of their increasing degree (the degree of a node is the number of nodes to which it is connected).
c. The procedure is then extended by relabelling the other nodes which are directly connected to the new node 2, in the order of their increasing degree, and so on until the renumbering is complete. The graph shown in Figure (3.8) is the application of the algorithm to the Triangular Network starting at a corner node which produces a



Bandwidth 4

FIGURE 3.8: Cuthill-Mckee numbering scheme for a triangular network

matrix of bandwidth equal to 4, while another way of numbering the same Triangular Network gives a matrix of bandwidth equal 7 as shown below in Figure (3.9).



Bandwidth 7

FIGURE 3.9

ii. THE REVERSE CUTHILL-MCKEE ORDERING

Mckee considered the reverse Cuthill-Mckee algorithm which re-numbers the Cuthill-Mckee ordering in the reverse way. Surprisingly, this simple modification often yields an ordering superior to the original ordering in terms of efficiency, although the bandwidth remains unchanged but the reverse scheme is always at least as good, as far as storage and operation counts are concerned. Here, reverse is used in the sense that element say (i,j,k,ℓ) moves to $(N-i+1,N-j+1,N-k+1,N-\ell+1)$.

Figures (3.10) and (3.11) show both the Cuthill-Mckee and the reverse numbering algorithms arising in the use of finite element methods for the solution of partial differential equations in a square region with rectangular elements.

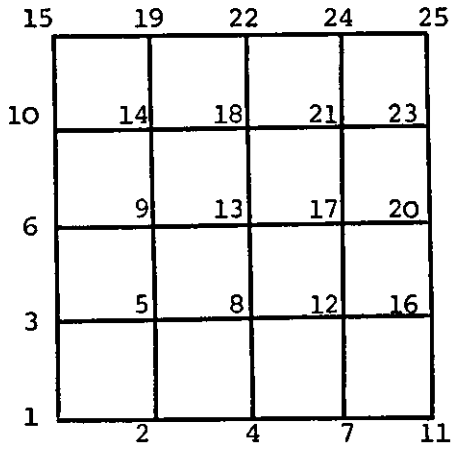


FIGURE 3.10

Cuthill-Mckee numbering with bandwidth 5

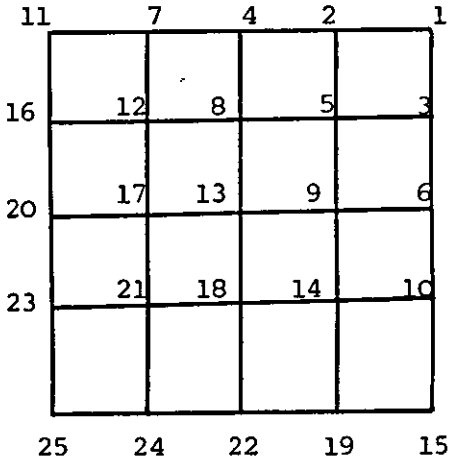


FIGURE 3.11

Reverse Cuthill-Mckee numbering with bandwidth 5

3.3 INTERPOLATION FUNCTIONS

The most crucial step in the finite element analysis of a given problem is the choice of adequate interpolation functions. They must be chosen to meet certain criteria such that the convergence to the true solution of the governing differential equation is achieved. The finite element interpolations are characterized by the shape of the element and on the order of the approximations chosen.

In general, the choice of a finite element depends on the geometry of the solution domain and the degree of accuracy desired in the solution. The functions used to represent the behaviour of the solution in each element are called interpolation functions or approximating functions.

Polynomial type interpolation functions are the most common forms of approximation for the finite element applications because they are easy to handle, specifically, it is easier to perform differentiation or integration with polynomials and because it is possible to improve the accuracy of the results by increasing the order of the polynomial. Theoretically, a polynomial of infinite order corresponds to the exact solution, but in practice we take polynomials of finite order only as an approximation.

While choosing the order of the polynomial in a polynomial type interpolation function, the following considerations have to be taken into account:

i) The interpolation polynomial should satisfy, as far as possible, the convergence requirements, the unknown must be continuous within the elements, for this reason complete polynomials are

often favoured. Complete polynomials are those in which all possible terms up to any given degree are present, the necessary terms for all possible polynomials up to a complete order six are shown in Figure (3.12), which is known as the Pascal triangle.

ii) The polynomial representation within an element should not change with a change in the local coordinate system (when a transformation is made from one cartesian coordinate system to another). This property is called geometric isotropy or geometric invariance.

In order to achieve geometric invariance the polynomial should contain terms which do not violate symmetry in Figure (3.12).

	Name	No. of terms
1	Constant	1
$x \quad y$	linear	3
$x^2 \quad xy \quad y^2$	quadratic	6
$x^3 \quad x^2y \quad xy^2 \quad y^3$	cubic	10
$x^4 \quad x^3y \quad x^2y^2 \quad xy^3 \quad y^4$	quartic	15
$x^5 \quad x^4y \quad x^3y^2 \quad x^2y^3 \quad xy^4 \quad y^5$	quantic	21
$x^6 \quad x^5y \quad x^4y^2 \quad x^3y^3 \quad x^2y^4 \quad xy^5 \quad y^6$	hexadic	28

FIGURE 3.12: Array of terms in complete polynomials of various orders in two dimensions

Thus, in the case of two dimensional linear elements (triangle), the polynomial should include terms containing both x and y in addition to the constant term. In the case of the cubic polynomial if we neglect the term x^2 for any reason, we should not include xy and y^2 also in order to maintain geometric isotropy of the model.

iii) The other consideration in selecting the order of the polynomial is to make the number of terms involved in the polynomial equal to the total number of degrees of freedom associated with the element otherwise the polynomial may not be unique.

The satisfaction of this requirement enables us to express the polynomial coefficients in terms of the nodel unknowns of the element. For some problems, however, choosing interpolation functions that meet all the requirements may be difficult and may involve excessive numerical computation.' For this reason, some investigators have ventured to formulate interpolation functions for elements that do not meet all the requirements. In some instances acceptable convergence has been obtained, whereas in others no convergence or convergence to an incorrect solution has occurred.

3.4 THE TWO DIMENSIONAL TRIANGULAR ELEMENT

The two dimensional triangular element is probably the most widely used finite element. One reason for this is that arbitrary regions in two dimensions can be approximated by polygons, which can always be divided up into a finite number of triangles. In addition the complete mth order polynomial,

$$U = \alpha_1+\alpha_2 x+\alpha_3 y+\alpha_4 xy+\alpha_5 x^2+\alpha_6 xy+ \ldots+\alpha_n y^m , \qquad (3.3)$$

where $\alpha_1,\alpha_2,\ldots,\alpha_n$ are the coefficients of the polynomial, also known as generalized coordinates, m is the degree of the polynomial and $n = \sum_{j=1}^{m+1} j$ can be used to interpolate a function say U, at $\frac{1}{2}(m+1)(m+2)$ symmetrically placed nodes in a triangle.

For example, the value of a linear triangle function may be found at any point if its values at three nodes, typically the vertices are known. For higher degrees of polynomial, we can generate the required nodes by taking (n-1) equally spaced lines parallel to each side and defining the nodes to be the intersections of these lines with each other and with the sides of the triangle as shown in Figure (3.13).

We consider first the *linear case* as indicated in Figure (3.14). Let the nodes be labelled as 1,2 and 3 and let the global coordinates of the nodes 1,2 and 3 be given by $(x_1,y_1),(x_2,y_2)$ and (x_3,y_3) and the nodal values of U(x,y) by U_1,U_2 and U_3 respectively.

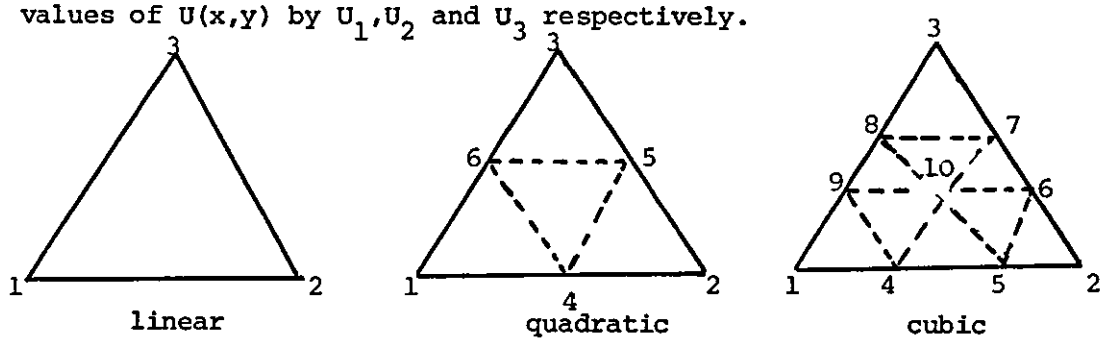


FIGURE 3.13: Nodes for linear, quadratic and cubic approximations on a single triangular element

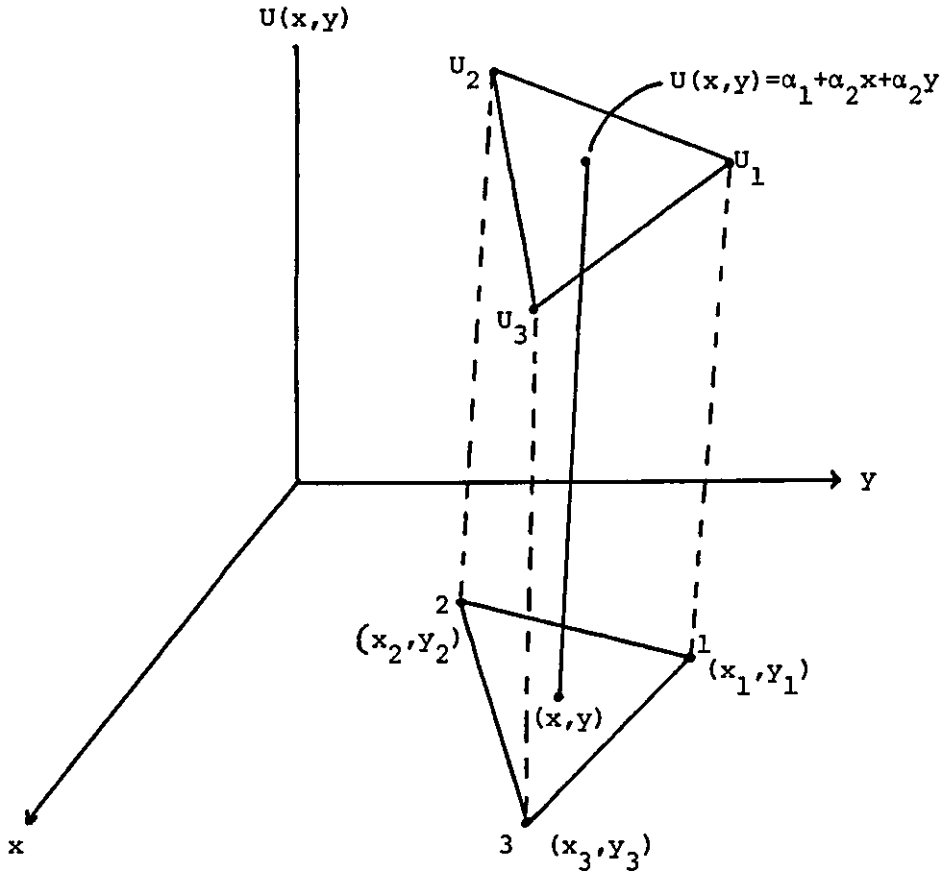


FIGURE 3.14

The variation of U inside the element is assumed to be linear and of the form,

$$U(x,y) = \alpha_1+\alpha_2 x+\alpha_3 y \ , \qquad (3.4)$$

The α_i are uniquely determined when the values of U(x,y) are specified at the nodes.

We now evaluate U at each node of the triangle in Figure (3.14). Thus,

$$U_1 = \alpha_1 + \alpha_2 x_1 + \alpha_3 y_1$$
$$U_2 = \alpha_1 + \alpha_2 x_2 + \alpha_3 y_2 \qquad (3.5)$$
$$U_3 = \alpha_1 + \alpha_2 x_3 + \alpha_3 y_3$$

Solving equations (3.5) for α_1, α_2 and α_3 yields,

$$\alpha_1 = \frac{1}{2\Delta}(a_1 U_1 + a_2 U_2 + a_3 U_3)$$
$$\alpha_2 = \frac{1}{2\Delta}(b_1 U_1 + b_2 U_2 + b_3 U_3) \qquad (3.6)$$
$$\alpha_3 = \frac{1}{2\Delta}(c_1 U_1 + c_2 U_2 + c_3 U_3)$$

where Δ is the area of the triangle 1,2,3 given by,

$$\Delta = \tfrac{1}{2}\begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix}$$

$$= \tfrac{1}{2}(x_1 x_2 + x_2 y_3 + x_3 y_1 - x_1 y_3 - x_2 y_1 - x_3 y_2) \ , \qquad (3.7)$$

$$a_1 = x_2 y_3 - x_3 y_2$$
$$b_1 = y_2 - y_3 \qquad (3.8)$$
$$c_1 = x_3 - x_2$$

with the other a's, b's and c's obtainable by cyclic permutation of the subscript 1,2,3.

The substitution of (3.6) into (3.5) with rearrangement yields the equation,

$$U(x,y) = N_1^{(1)}(x,y)U_1 + N_2^{(1)}(x,y)U_2 + N_3^{(1)}(x,y)U_3,$$

or

$$U(x,y) = \sum_{i=1}^{3} N_i^{(1)}(x,y)U_i \ , \qquad (3.9)$$

where,

$$N_i^{(1)}(x,y) = \frac{1}{2\Delta}(a_i + b_i x + c_i y) \ , \quad i=1,2,3 \qquad (3.10)$$

The function $N_i^{(1)}(x,y)$ is called an interpolation function or 'Shape Function', and has the value 1 at the ith node and the value

O at the other two nodes. Since $N_i(x,y)$ is linear in the variables x and y, it is identically zero on the side between nodes 2 and 3 and the gradient of U in x or y direction will be a constant.

For the *Quadratic* approximation with nodes numbered as shown in Figure (3.15), the complete polynomial is given by

$$U(x,y) = \alpha_1+\alpha_2x+\alpha_3y+\alpha_4xy+\alpha_5x^2+\alpha_6y^2 \tag{3.11}$$

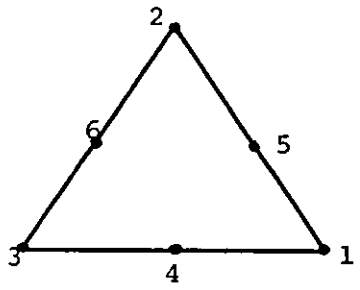


FIGURE 3.15

A similar procedure to that given for the linear case yields the approximation,

$$U(x,y) = \sum_{i=1}^{6} N_i^{(2)}(x,y)U_i \ , \tag{3.12}$$

where U_i (i=1,...,6) are the values of U(x,y) at the vertices. The $N_i^{(2)}(x,y)$ (i=1,2,...,6) are given by,

$$\left.\begin{aligned}
N_1^{(2)}(x,y) &= N_1^{(1)}(2N_1^{(1)}-1) \ , \\
N_2^{(2)}(x,y) &= N_2^{(1)}(2N_2^{(1)}-1) \ , \\
N_3^{(2)}(x,y) &= N_3^{(1)}(2N_3^{(1)}-1) \ , \\
N_4^{(2)}(x,y) &= 4N_1^{(1)}N_2^{(1)} \ , \\
N_5^{(2)}(x,y) &= 4N_2^{(1)}N_3^{(1)} \ , \\
N_6^{(2)}(x,y) &= 4N_1^{(1)}N_3^{(1)} \ .
\end{aligned}\right\} \tag{3.13}$$

Again it follows that,

$$N_i^{(2)} = \begin{cases} 1, & i=j, \\ 0, & i\neq j, \end{cases} \quad 1\leq i\leq j\leq 6. \tag{3.14}$$

It is particularly satisfactory that the shape function $N_i^{(2)}(x,y)$ $(i=1,2,\ldots,6)$ can be expressed in terms of the shape function $N_i^{(1)}(x,y)$ of the linear case and therefore to simplify the formula we shall denote the $N_i^{(1)}(x,y)$ simply by $N_\imath$, $(\imath=1,2,3)$.

Finally, for the *Cubic* case $(m=3)$ with nodes numbered as shown in Figure (3.16) below,

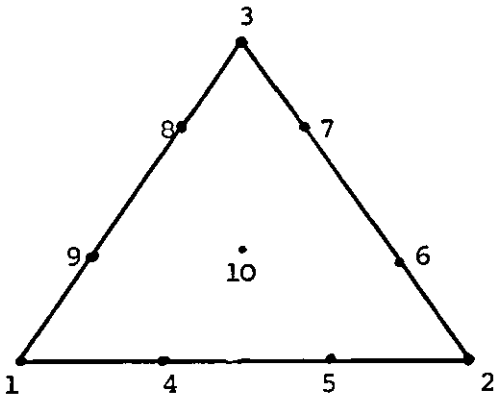


FIGURE 3.16

The complete cubic polynomial is given by:

$$U(x,y) = \alpha_1+\alpha_2x+\alpha_3y+\alpha_4xy+\alpha_5x^2+\alpha_6y^2+\alpha_7x^2y+\alpha_8xy^2+\alpha_9x^3+\alpha_{10}y^3 \qquad (3.15)$$

As before the approximate polynomial is given by,

$$U(x,y) = \sum_{i=1}^{10} N_\imath^{(3)}(x,y)U_i \ , \qquad (3.16)$$

where U_i, $(i=1,2,3)$ are the values of $U(x,y)$ at the vertices $(1,2,3)$, U_i $(i=4,5,\ldots,9)$ are values at the points of trisection of the sides and U_{10} is the value of $U(x,y)$ at the centroid of the triangle as shown in Figure (3.16).

The shape functions are given by,

$$N_1^{(3)}(x,y) = \tfrac{1}{2}N_1(3N_1-1)(3N_1-2) \ ,$$

with $N_2^{(3)}(x,y)$ and $N_3^{(3)}(x,y)$ similarly,

$$
\left.
\begin{aligned}
N_4^{(3)}(x,y) &= \frac{9}{2}N_1N_2(3N_1-1) \ , \\
N_5^{(3)}(x,y) &= \frac{9}{2}N_1N_2(3N_2-1) \ , \\
\text{with } (N_6^{(3)}(x,y),\ldots,N_9^{(3)}(x,y)) &\text{ similarly,} \\
N_{10}^{(3)} &= 27N_1N_2N_3 \ .
\end{aligned}
\right\} \qquad (3.17)
$$

The tenth parameter can be eliminated by using the linear relation,

$$N_{10}^{(3)} = \frac{1}{4}(N_4+N_5+N_6+N_7+N_8+N_9) - \frac{1}{6}(N_1+N_2+N_3) \ , \qquad (3.18)$$

to yield a function that will still interpolate the quadratic exactly. This procedure is called the elimination of internal parameters.

Again, it follows that,

$$N_i^{(3)}(x_j,y_j) = \begin{cases} 1 & i=j \\ 0 & i\neq j \end{cases} \ , \ 1\leq i\leq j\leq 10. \qquad (3.19)$$

In a similar manner the shape function can be generated for any order of parameters.

We have created a set of shape functions $N_i(x,y)$ which form a basis for all functions which are linear on each element and continuous within the element and so on for quadratic, cubic and other higher order elements.

Another common element shape is a "rectangle", on which in a similar manner a family of shape functions can be developed, details are given in ZIENKIEWICZ (1977).

3.5 CURVED BOUNDARIES

So far shape functions have been constructed for straight sides only. To solve a problem with a curved boundary the mesh must be refined until the boundary is sufficiently closely approximated by a series of straight-line segments.

Another technique which is introduced into structured analysis by ERGATOUDIS, IRONS, and ZIENKIEWICZ [1968], is to use a curved finite element which is based on geometrical considerations, whereby interpolating functions are obtained directly in terms of x and y for the triangle and quadrilateral with arbitrarily placed side points. These local functions can be used to construct piecewise smooth global interpolating functions over regions possessing curved boundaries and composed of elements which are triangles and parallelograms with arbitrarily positioned side points.

This approach is called the "isoparametric formulation". The simplest member of the isoparametric family is the "linear" element and, by definition, this may not have curved sides. A more useful isoparametric element is the "quadratic" element because it may have curved sides and therefore provides a better fit to the curved shape of the region. The essential ideal underlying the development of elements with curved sides centres on transforming simple geometric shapes in some local coordinate system into distorted shapes in the global system.

For the case of a triangular element with straight sides and no side points, the linear transformation from the local (p,q) system to the global (x,y) system is given by,

$$x = px_1 + qx_2 + rx_3 \quad ,$$
$$y = py_1 + qy_2 + ry_3 \tag{3.20}$$

In addition to these equations a third condition requiring that the sum of p,q and r are unit, that is,

$$p + q + r = 1 \tag{3.21}$$

From equation (3.21) it is clear that only two of the local systems p,q can be independent, just as the original coordinate system, where there are only two independent coordinates. Thus equation (3.20) can be written as,

$$x = (x_1-x_3)p + (x_2-x_3)q + x_3$$
$$y = (y_1-y_3)p + (y_2-y_3)q + y_3 \quad , \tag{3.22}$$

where the various quantities are explained in Figure (3.17).

Inversion of equation (3.20) and (3.21) give the local coordinates in terms of the global coordinates. Thus,

$$p = \frac{1}{2\Delta}[(y_2-y_3)x + (x_2-x_1)y + (x_2x_3-x_3y_2)]$$
$$q = \frac{1}{2\Delta}[(y_3-y_1)x + (x_1-x_3)y + (x_3y_1-x_1y_3)] \tag{3.23}$$

where Δ is the area of the triangle.

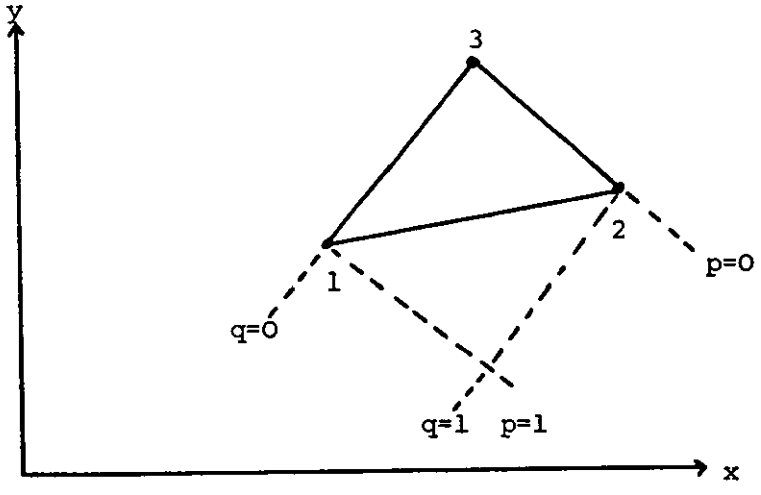


FIGURE 3.17

In the more general case of a triangle A.R. MITCHELL [1971] illustrates the nature of the computations involved by considering an example consisting of a triangular element with two straight sides and one curved side. To maintain generality, first a triangle with three curvilinear sides as shown in Figure (3.18) is considered. Mitchell proceeds to transform this triangle into the standard triangle in the (p,q) plane by using the transformation formulae,

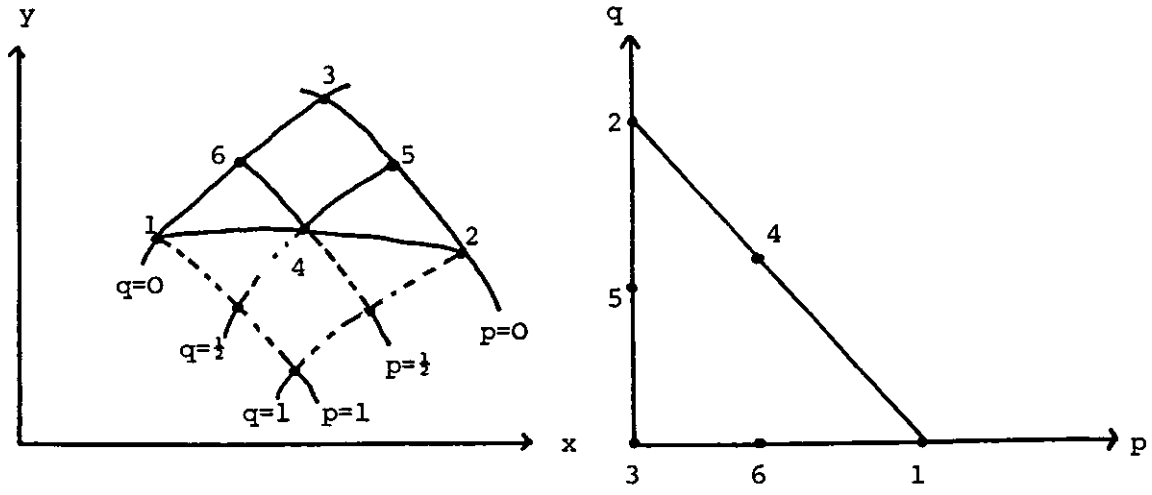


FIGURE 3.18: Treatment of curved boundaries via isoparametric transformations

$$x = p(2p-1)x_1+q(2q-1)x_2+r(2r-1)x_3+4pqx_4+4qrx_5+4rpx_6$$
$$y = p(2p-1)y_1+q(2q-1)y_2+r(2r-1)y_3+4pqy_4+4qry_5+4rpy_6 \qquad (3.24)$$

where r=1-p-q, which can be rewritten in the form,

$$x = 2(x_1+x_3-2x_6)p^2+2(x_2+x_3-2x_5)q^2+4(x_3+x_4-x_5-x_6)pq+$$
$$(4x_6-x_1-3x_3)p+(4x_5-x_2-3x_3)q + x_3 \qquad (3.25)$$

and $$y = 2(y_1+y_3-2y_6)p_2+2(y_2+y_3-2y_5)q^2+4(y_3+y_4-y_5-y_6)pq+$$

$$(4y_6-y_1-3y_3)p+(4y_5-y_2-3y_3)q + y_3 \ .$$

This time it is not an easy matter to solve (3.25) for the curvilinear coordinates p and q in terms of x,y and the coefficients of the six points. It is sufficient to say that the desired expressions for p and q are, in general of quadratic form in the x and y. Hence, the local curvilinear (p,q) system is uniquely determined in terms of the fixed (x,y) system and the location of the six points.

If the sides 2,5,3 and 3,6,1 are straight sides with 5 and 6 the mid-points respectively, the transformation formulae reduce to

$$\tilde{x} = X\tilde{p}\tilde{q} + \tilde{x}_1\tilde{p} + \tilde{x}_2\tilde{q}$$

$$\tilde{y} = Y\tilde{p}\tilde{q} + \tilde{y}_1\tilde{p} + \tilde{y}_2\tilde{q} \ ,$$

where,

$$\tilde{x} = x-x_3 \ , \qquad \tilde{y} = y-y_3$$

$$X = 2[2\tilde{x}_4-(\tilde{x}_1+\tilde{x}_2)], \quad Y = 2[2\tilde{y}_4-(\tilde{y}_1+\tilde{y}_2)] \ .$$

After some considerable manipulation it can be shown that the line p+q=1 for this case in the (p,q) plane corresponds to the quadratic curve,

$$[(\bar{y}_1\bar{x}-\bar{x}_1\bar{y}) + (\bar{y}_2\bar{x}-\bar{x}_2\bar{y})]^2 = (\bar{y}_1\bar{x}_2-\bar{x}_1\bar{y}_2)[(\bar{y}_1\bar{x}-\bar{x}_1\bar{y})-(\bar{y}_2\bar{x}-\bar{x}_2\bar{y})], \quad (3.26)$$

where $\bar{x}=x-\bar{x}_4$, and $\bar{y}=y-\bar{y}_4$. In the special case, where the points are given by $1\equiv(1,0)$ $2\equiv(0,1)$, $3\equiv(0,0)$ and $4\equiv(\ell,\ell)$. Equation (3.26) reduces to,

$$(x-y)^2 = \frac{x+y-2\ell}{1-2\ell} \ . \qquad (3.27)$$

The quadratic curve given by Equation (3.26) is, of course, only an approximation to the original curvilinear side of the triangle in Figure (3.17), this example illustrates a method of handling curvilinear sides. For a more thorough discussion on the methods of treating isoparametric elements, details are given by A.R. MITCHELL [1973].

3.6 VARIATIONAL PRINCIPLES AND WEIGHTED RESIDUALS

3.6.1 VARIATIONAL FORMULATION OF THE FINITE ELEMENT METHOD

Variational principles occur naturally in many physical and other engineering problems and the approximate methods of solution of such problems are often based on associated variational principles.

We will discuss first a finite element approximate method which is directly based on the variational principle. A general analysis of the variational principles is given by L. EISGOLTS [1973].

Briefly, the mathematical formulation of a variational principle is that the integral of some typical function has a *minimum* or a *maximum* value for the actual performance of the system than for any virtual performance subject to the general conditions of the system.

A functional $J(\underline{u})$ can be defined as a function of several functions which has a value dependent on a function $\underline{u}$ and is defined by an integral of the form,

$$J(\underline{u}) = \int_R F(\underline{u}, \frac{\partial \underline{u}}{\partial x}, \ldots) dR + \int_{\partial R} E\ (\underline{u}, \frac{\partial \underline{u}}{\partial x}, \ldots\ )\ ds, \quad (3.28)$$

where F and E are specified operators and in general the unknown function $\underline{u}$ is a vector.

The main idea in variational principle theory is to find the function $\underline{u}$ which minimizes the value of $J(\underline{u})$. A necessary condition for this is that the first variation in $J(\underline{u})$: $\delta J(\underline{u})$, must be zero when $\underline{u}$ is varied by an arbitrary small amount $\delta \underline{u}$:

$$\begin{aligned} \delta J(\underline{u}) &= J(\underline{u} + \delta \underline{u}) - J(\underline{u}) \\ &= 0 + O(\delta \underline{u}^2) \ . \end{aligned} \quad (3.29)$$

Given a differential equation problem such as that specified by (3.1) we say that there is a variational principle for the problem if the task of finding the solution $\underline{u}$ of the original problem can be reformulated as the problem of minimizing a particular functional $J(\underline{v})$ over a set of admissible functions $\underline{v}$, which satisfy certain conditions at the boundaries of the domain of the problem.

The finite element method makes use of this idea, and in particular it involves a careful analysis of the set of admissible functions which must satisfy the essential boundary conditions. In general, if the functional (3.28) contains derivatives up to and including the pth, the set of admissible functions in which we look for the solution has to be the space H^p, defined as the space of all functions $\underline{v}$ which has finite energy in all derivatives up to and including the pth derivatives, i.e. if $v \in H^p$ then,

$$\int_R (v^2 + v'^2 + \ldots + v^{(p)^2}) dR < \infty. \qquad (3.30)$$

In particular this means that H^p contains all functions with continuous $(p-1)^{th}$ derivatives.

We restrict the choice of $\underline{v}$ to those functions in H^p which satisfy the boundary conditions, i.e. to a subspace which we label H^p_β.

The finite element method makes use of the "weak form" of the variational principle which is obtained by integrating (3.28) by parts to reduce the pth derivatives. In general, if $p=2m$, say, this may be done m times, so that the maximum order derivative occurring in the variational principle is m; this has some important consequences:

The new form of the functional $J(\underline{u})$ contains lower order derivatives

of the unknown function $\underline{u}$ compared to the governing differential equation, so the set of admissible functions can be enlarged and hence an approximate solution can be obtained using a larger class of functions.

Regarding the boundary conditions, the variational formulation permits us to treat complicated boundary conditions as natural or free boundary conditions:

(a) Natural Boundary Conditions

These are typically conditions on the higher derivatives, which are absorbed into the new form of the functional when we integrate by parts.

(b) Essential Boundary Conditions (or forced boundary conditions)

These have to be satisfied by the new space of admissable functions. If the finite element equations are derived on the basis of the new variational principle, the natural boundary conditions will be automatically incorporated in the formulation and hence conditions are to be enforced on the solution in order to obtain a unique solution, we denote the new space of admissable function H_E^m, where m is the order of the new functional derivatives and E refers to the fact that the function need only satisfy the essential boundary conditions.

3.6.2 DERIVATION OF FINITE ELEMENT EQUATIONS USING VARIATIONAL APPROACH

Let the general problem be defined as (3.28),

$$J(\underline{u}) = \int_R E_1(\underline{u},\underline{u}_x,\ldots)dR + \int_{\partial R} E_2(\underline{u},\underline{u}_x,\ldots)ds \ .$$

The finite element procedure for solving this problem can be stated by the following steps:

1. The solution domain R is divided into n smaller parts called elements, the commonly used element shapes are given in Figures (3.2), (3.3).

2. The unknown variable is assumed to vary in each element in a suitable manner similar to those given in Equation (3.2), i.e.,

$$U(x,y) = \sum_{i=1}^{n} N_i(x,y)U_i \ ,$$

where N_i is the shape function, and U_i is the nodal values

3. The solution of U(x,y) is obtained from the minimum of J(U) with respect to all unknown nodal values U_i. This is equivalent to having,

$$\delta J(\underline{U}(x,y)) = \underline{0} \quad \text{or,} \tag{3.31}$$

$$\frac{\partial J}{\partial \underline{U}} = \begin{bmatrix} \frac{\partial J}{\partial U_1} \\ \frac{\partial J}{\partial U_2} \\ \vdots \\ \frac{\partial J}{\partial U_N} \end{bmatrix} = \underline{0} \ , \tag{3.32}$$

where N denotes the total number of nodal unknowns in the problem. If the functional J can be expressed as a summation of elemental contributions as:

$$J = \sum_{e=1}^{E} J^{(e)} \ , \tag{3.33}$$

where e indicates the element number, then equation (3.33) can be expressed as,

$$\frac{\partial J}{\partial U_i} = \sum_{e=1}^{E} \frac{\partial J^{(e)}}{\partial U_i} = 0, \quad i=1,2,\ldots,N. \tag{3.34}$$

In the special case, where J is a quadratic function of U and its derivatives, we can obtain the element equations as,

$$\frac{\partial J^{(e)}}{\partial U^{(e)}} = K^{(e)} U^{(e)} - f^{(e)} , \tag{3.35}$$

where $K^{(e)}$ is the element characteristic matrix and $f^{(e)}$ is the element characteristic vector.

4. To obtain the overall equations of the system, we rewrite equation (3.35) as,

$$\frac{\partial J}{\partial \underline{U}} = \underline{K}\underline{U} - \underline{f} = \underline{0} , \tag{3.36}$$

where

$$\underline{K} = \sum_{i=1}^{E} K^{(e)}$$

$$\underline{f} = \sum_{e=1}^{E} f^{(e)}$$

and the summation sign indicates the assembly over all finite elements in the region.

5. The linear simultaneous equations (3.36) can be solved after applying the boundary conditions to find the unknowns $\underline{U}$.
If J is not quadratic in $\underline{U}$ then we obtain a set of simultaneous non-linear equations. These may be solved for $\underline{U}$ by using a standard iterative method.
The main difficulty with this form of finite element method is that it relies on reformulating the original problem as a variational

principle. The governing differential equations have to be the Euler equations of the functional, and while every functional has a set of Euler equations, the reverse is not always true: not every set of differential equations can be expressed as the Euler equations of some functional. Thus, the range of application of variational principles is somewhat limited, and we now look at another method of solution based on weighted residuals.

3.6.3 THE METHOD OF WEIGHTED RESIDUALS

The method of weighted residuals which includes the Galerkin method as a special case is an approximate method which seeks a solution that is a good approximation to the exact solution over the whole domain of the given problem.

To introduce the method, we consider the set of differential equations (3.1).

The solution of (3.1) is equivalent to determining $\underline{u}$ so that,

$$((\underline{Du} - \underline{f}), \underline{w}) = \underline{0} , \tag{3.37}$$

or

$$\int_R (\underline{Du}-\underline{f})\underline{w}^T dxdy = \int_R [D_1\underline{u}w_1 + D_2\underline{u}w_2 + \ldots + D_r\underline{u}w_r]dxdy - \int_R [f_1w_1 + f_2w_2 + \ldots + f_rw_r]dxdy = \underline{0} \tag{3.38}$$

where $\underline{w}=[w_1,w_2,\ldots,w_r]^T$, are a set of arbitrary weighting functions.

The converse is also true: if (3.37) is satisfied for all $\underline{w}$ then (3.1) must be satisfied at all points of the region R. The solution $\underline{u}$ must also satisfy the boundary conditions (3.2), and these are incorporated either by considering only those functions which satisfy (3.2), on ∂R, or by specifying that,

$$\int_{\partial R} \underline{Lu}\,\underline{\bar{w}}^T\,ds = \underline{0} \quad , \tag{3.39}$$

in the process of solution, where again $\underline{\bar{w}}$ is a vector of arbitrary weighting functions. The two methods give the same results, but sometimes it is easier to incorporate the boundary conditions á priori, and sometimes easier to use them later in the solution process.

It is clear that if,

$$\int_R (\underline{Du}-\underline{f})\,\underline{w}^T dxdy + \int_{\partial R} \underline{Lu}.\underline{\bar{w}}^T\,ds = 0 \quad , \tag{3.40}$$

is satisfies for arbitrary $\underline{w}$ and $\underline{\bar{w}}$ then (3.1) and (3.2) are satisfied, and the converse is also true. Thus, any solution of (3.40) is a solution of (3.1) and (3.2), and conversely.

As with the method of variational principles, we integrate by parts and replace (3.40) by a form,

$$\int_R (\underline{\bar{D}}\underline{u}-\underline{f})\,\underline{c}(\underline{w})^T dxdy + \int_{\partial R} \underline{\bar{L}}\underline{u}\;\underline{c}(\underline{\bar{w}})^T ds = 0 \quad , \tag{3.41}$$

where $\underline{\bar{D}}$ and $\underline{\bar{L}}$ usually contain lower-order derivatives than those in $\underline{D}$ and $\underline{L}$, so a lower order of continuity is required in $\underline{u}$ but $\underline{w}$ and $\underline{\bar{w}}$ usually have to be more continuous. The same points about continuity that were made for the variational principle form also apply here. The next step in the application of the method of weighted residual to the finite element formulation is to introduce a trial solution,

$$\underline{u} \cong U = \sum_{i=1}^{n} N_i(x,y)U_i \quad , \tag{3.42}$$

which it is hoped, is close to $\underline{u}$ in some sense or can be made so if n is large enough. The trial solution is chosen to satisfy the boundary condition, and since the approximate solution should be capable of

converging to the exact solution as n approaches infinity it is important that the functions N_i are linearly independent and chosen from a set of functions which is complete in the domain of interest.

Clearly, it will not generally be possible to make such an approximation and also satisfy the differential equations (3.1) and (3.2) exactly, but the integral form allows an approximation to be made if we put a finite set of prescribed functions in place of the arbitrary functions $\underline{w}$ and $\underline{w}$,

$$\underline{w} = \overline{\underline{w}} = \underline{\phi}_i \quad , \ (i=1,2,\ldots,n) \ . \qquad (3.43)$$

and if we proceed by substituting the approximate solution (3.42) into equation (3.40) and (3.41) we get,

$$\int_R \underline{r}(x,y,\underline{U})\ \underline{\phi}_i^T dxdy + \int_{\partial R} \underline{Lu}\ .\ \underline{\phi}_i^T\ ds = \underline{0} \ , \ (i=1,\ldots,n) \qquad (3.44)$$

and,

$$\int_R \overline{\underline{r}}(x,y,\underline{U}).\underline{F}(\underline{\phi}_i)^T dxdy + \int_{\partial R} \overline{\underline{Lu}}.\ \underline{F}(\underline{\phi}_i)^T\ ds \ , \ (i=1,2,\ldots,n), \qquad (3.45)$$

where $\underline{r}(x,y,\underline{U})=(\underline{DU}-\underline{f})$, represents the errors, or "residuals". It is to be expected that $\underline{r}$ will be small, in some sense, but not zero throughout, the domain in which the solution is sought.

Since $\underline{r}=\underline{0}$ throughout the domain when the exact solution is obtained, $\underline{r}$ will be considered as a measure of error and since the exact solution is not available in general, the size and the distribution of $\underline{r}$ in the domain can be used to assess the accuracy of the solution. Thus, if a solution for a particular value of n has been obtained, $\underline{r}$ can be evaluated. The effect of obtaining a new solution with increased n should cause a reduction in $\underline{r}$ in some average sense.

A different choice of the set of functions (ϕ_i, i=1,...,n) give rise to different methods which collectively are known as the methods of weighted residuals. A common choice (which we have used throughout the work) is to take the shape function N_i as ϕ_i i.e. the functions ϕ_i are chosen from the same family as the trial functions in equation (3.42), since the trial functions are chosen from a linearly independent set of functions, complete in the domain of interest equations. This choice leads to the "Galerkin Method", and the effect is to make the error vector orthogonal to each of the shape functions and hence to any linear combination of them.

It is known (see ZIENKIEWICZ [1977]) that if a variational principle exists for a linear problem, then the Galerkin method gives rise to precisely the same equations as the variational principle, when the finite element method is applied. The advantage of the Galerkin method is that it is valid for problems which have no variational formulation, and so it is more widely applicable.

The description of the method of weighted residuals given above requires that the boundary conditions are satisfied exactly and that the differential equation is satisfied approximately. This is called an interior method. The converse is also possible. In the boundary method the differential equation is satisfied exactly but the boundary conditions are satisfied approximately.

To demonstrate the connection between the variational method (Ritz) and the Galerkin method a problem governed by Poisson's equation is considered. That is,

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f \ , \tag{3.46}$$

subject to the condition u=0 on the boundary.

The equivalent variational problem requires that,

$$J(u) = \iint [(\frac{\partial u}{\partial x})^2 + (\frac{\partial u}{\partial y})^2 + 2fu] \, dxdy \ , \tag{3.47}$$

has a minimum corresponding to the exact solution, subject to the same condition u=0 on the boundary.

Substituting equation (3.42) into equation (3.47) gives,

$$J(U) = \int \int [\{ \sum_{i=1}^{n} U_i \frac{\partial N_i}{\partial x} \}^2 + \{ \sum_{i=1}^{n} U_i \frac{\partial N_i}{\partial y} \}^2 + 2f \sum_{i=1}^{n} U_i N_i] dxdy \tag{3.48}$$

Imposing the conditions given by equation (3.34) which require that $\frac{\partial J}{\partial U_j} = 0$ (j=1,...,n) produces the result,

$$\frac{\partial J}{\partial U_j} = \iint [2 \frac{\partial N_j}{\partial x} . \{ \sum_{i=1}^{n} U_i \frac{\partial N_i}{\partial x} \} + 2 \frac{\partial N_j}{\partial y} \{ \sum_{i=1}^{n} U_i \frac{\partial N_i}{\partial y} \} + 2f.N_j] \, dxdy = 0 \tag{3.49}$$

or,

$$\iint \frac{\partial N_j}{\partial x} . \frac{\partial U}{\partial x} + \frac{\partial N_j}{\partial y} \frac{\partial U}{\partial y} + f N_j] \, dxdy = 0 \tag{3.50}$$

The application of the Galerkin method to equation (3.46) gives,

$$\iint N_i [\frac{\partial^2 U}{\partial x^2} + \frac{\partial U^2}{\partial y^2} - f] \, dxdy = 0 \ , \tag{3.51}$$

and by applying Green's theorem to the first two terms leads to the result,

$$\int N_i [\frac{\partial U}{\partial x} \ell_x + \frac{\partial U}{\partial y} \ell_y] ds - \iint [\frac{\partial N_i}{\partial x} \frac{\partial U}{\partial x} + \frac{\partial N_i}{\partial y} \frac{\partial U}{\partial y} + f N_i] dxdy = 0 \tag{3.52}$$

Since N_1 is chosen to satisfy homogeneous boundary conditions the first term in equation (3.52) disappears and equation (3.52) reduces to equation (3.50).

Thus the two techniques are equivalent for this problem.

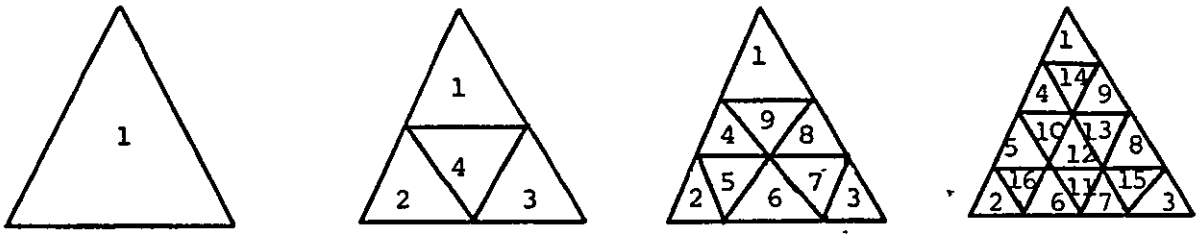
3.7 ERROR ESTIMATES

Expressed in its simplest terms the finite element method is a procedure for finding a piecewise smooth approximation to the solution of some underlying differential equation or system of differential equations. In most applications the polynomials defined on a partition (element) of the given domain are used to form the trial and test function spaces.

The finite element technique and computer implementation of the method has been to arbitrarily set the polynomial degree p at a fixed low value (typically, p=1,2,3 or 4) and to decrease the size of the element subdomains in order to reduce the error in the approximate solution. Error estimates showing the dependence of the rate of convergence on the mesh are well known. In fact, since the mesh size is usually denoted by the letter h, we refer to this standard approach as the h-version of the finite element method, see Figure (3.19).

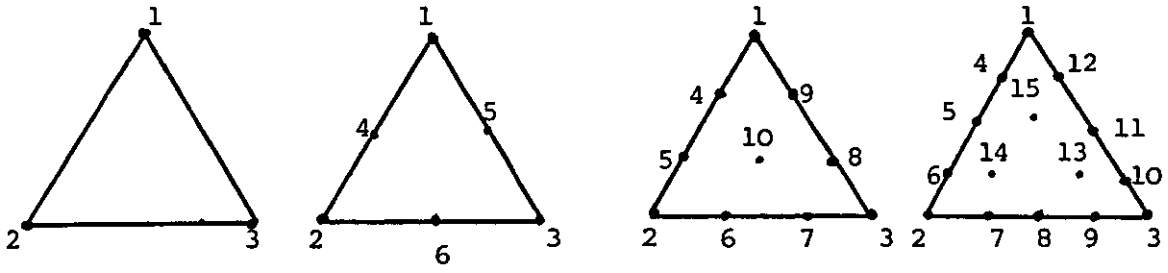
There is also another approach that has arisen recently by BABUSKA and DOOR [1981], in which they refer to the p-version of the finite element method. Here, one fixes the mesh size and increases the degree p of the piecewise polynomials in order to obtain the convergence of the approximation solution to the exact solution. This method is analyzed where the error estimates, in terms of the polynomial degree p are obtained. In particular, it is shown that, if the rates of convergence for the h-version using uniform refinement and the p-version are expressed in terms of the number of degrees of freedom, the p-version cannot have a slower rate of convergence than the h-version. Furthermore,

when corner singularities are present, the rate of convergence of the p-version is exactly twice that of the h-version. However, this is an interesting theoretical method but it is difficult to see it ever being widely used.



1-linear element 4-linear element 9-linear element 16-linear element

h-version finite elements



linear element quadratic element cubic element quartics element

p-version finite element

FIGURE 3.19: h- and p-version of the finite element method

Given a basic triangular element grid a display of options for obtaining a better solution may be set out as shown in Figure 3.19.

To write a program to increase the order p is likely to be substantially more difficult than writing one which is capable of decreasing h. In addition, the h-version will produce a matrix with substantially the same sparseness matrix, while the p-version will become comparatively less sparse, thus requiring more storage and more work to solve the system of equations. This is offset by the better convergence of the p-version, but on the whole the advantage seems to lie with the h-version.

STRANG and FIX [1974] and MITCHELL and WAIT [1977] give a detailed analysis and proof of convergence. Here we will give only a statement of the error bounds which is relevant to the present work.

We consider the finite element subspace $S^h \in H_E^m$ (m=0,1,2,...,), in the finite element method, an approximate solution is sought amongst functions which belong to the closed subspace S^h. The questions that arise then are "does the method converge as the mesh size decreases (i.e. as h→0) and can the error bounds be obtained in terms of h.

Consider the discretization of some two-dimensional region R by means of triangles.

The form used here is given by,

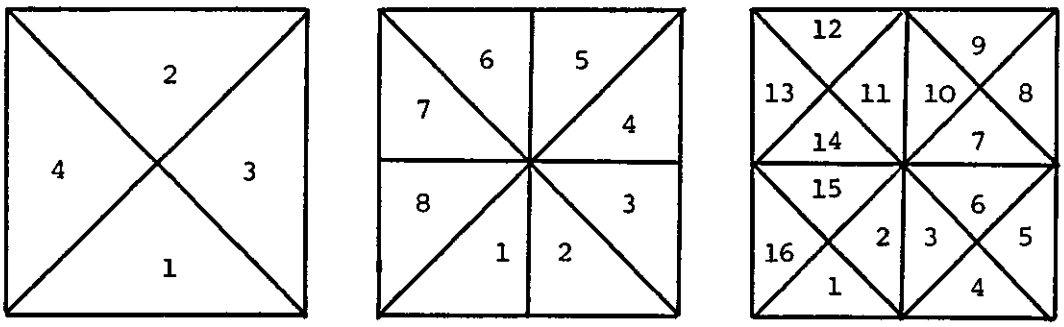
$$||u||_p = \left[\left\{ \int\int_R u^2 + \left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial^2 u}{\partial x^2}\right)^2 + \ldots + \left(\frac{\partial^p u}{\partial y^p}\right)^2 \right\} dxdy \right]^{\frac{1}{2}} .$$

The error e=u-U may be shown to satisfy an inequality of the form,

$$||e||_1 \leq c^2 h^2 \max(|u_{xx}|, |u_{xy}|, |u_{yy}|) , \qquad (3.53)$$

i.e., the norm of the error behaves like h^2 as $h \to 0$, for some constant C. Although the bounds on the error show that the method converges as $h \to 0$, the manner in which convergence occurs is not apparent. MELOSH [1963] gives the following sufficient condition *"If each subdivision of the finite element mesh contains the previous one as a subset, then the convergence will be monotonic"*.

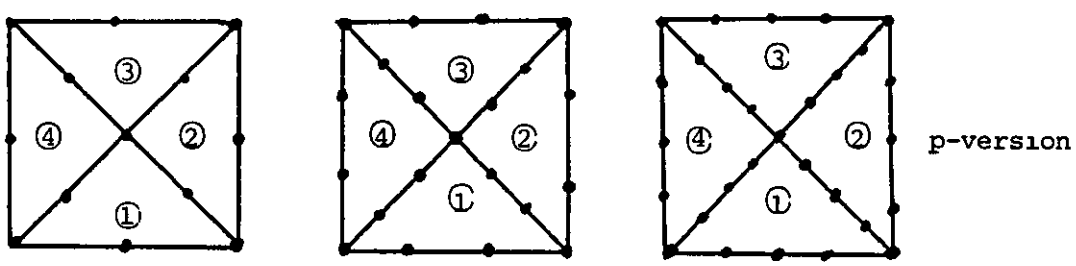
Table (3.1) shows such convergence for Poisson's equation (3.54) in a square, where the mesh is obtained by halving the dimensions of the triangles (h-version) in Figure (3.20a) below:



4 Elements 8 Elements 16 Elements

FIGURE 3.20(a)

and also we consider the convergence of the same problem (3.54) by increasing the order of the triangles, and fixing the number of elements (p-version) as shown in Figure (3.20b) below:



Quadratic (4 element) Cubic (4 element) Quartic (4 element)

FIGURE 3.20(b)

Example 1

The governing equation is:

$$\left.\begin{aligned} &\nabla^2 u = 0 \ , \\ &\text{with the boundary conditions,} \\ &u(0,y) = (1-e^{-\pi/2}) - \cos\frac{\pi}{2} y \\ &u(1,y) = 0 \\ &u(x,\pm 1) = 0 \end{aligned}\right\} \qquad (3.54)$$

which has the exact solution,

$$u = e^{-\pi/2\, x} \cdot \cos\frac{\pi}{2} y \ .$$

The following provide a good comparison of how the accuracy of the problem given in equation (3.54), increases for both h-version, and p-version finite element solutions.

No. of Elements (Triangles)	L_2 Error Norm		
	Quadratic	Cubic	Quartic
12	5.3967×10^{-3}	4.9248×10^{-4}	3.1466×10^{-5}
25	1.07761×10^{-3}	5.20561×10^{-5}	2.68135×10^{-6}
36	8.2776×10^{-4}	3.3992×10^{-5}	2.1310×10^{-6}
50	6.1016×10^{-4}	2.23794×10^{-5}	2.04479×10^{-6}
60	5.6662×10^{-4}	1.8876×10^{-5}	1.9830×10^{-6}
75	1.38908×10^{-4}	7.00277×10^{-6}	1.8915×10^{-6}

TABLE 3.1

Also the results given in Table (3.1) are plotted in Figure (3.21), which clearly show that the quartic element is more accurate than both cubic and quadratic elements.

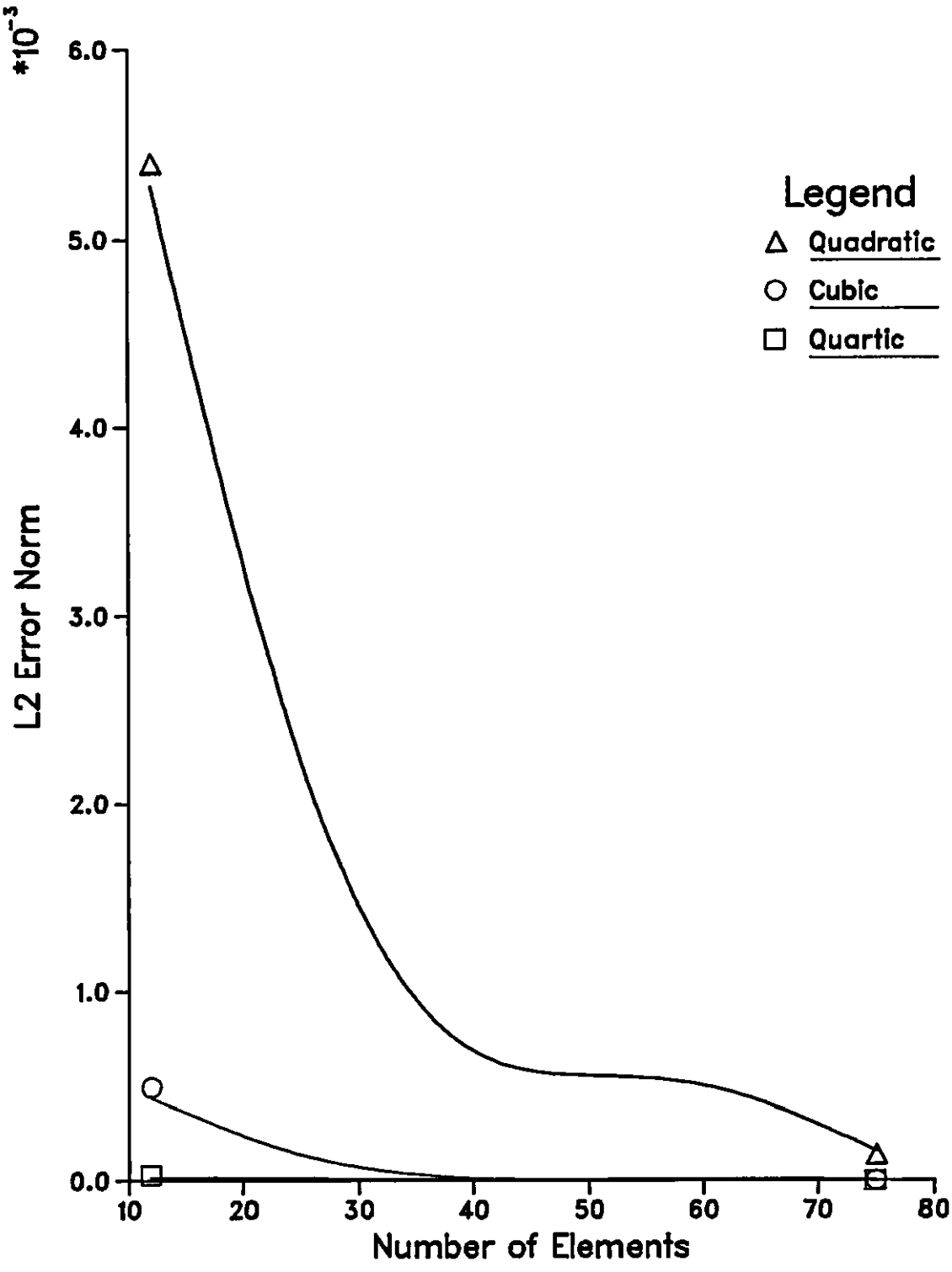

*10^-3
6.0
5.0
4.0
3.0
2.0
1.0
0.0
L2 Error Norm
10
20
30
40
50
60
70
80
Number of Elements
Legend
Quadratic
Cubic
Quartic

Figure (3.21)

3.8 ASSEMBLY OF ELEMENT MATRICES AND VECTORS

Once the element properties, namely, the element matrices and element vectors are determined in a common global coordinate system, the next step is to construct the overall system of equations. The procedure for constructing the system equations from the element equations is the same regardless of the type of the problem and the number and the type of the elements used, i.e., even if the system is modelled with a mixture of several different kinds of elements, the system equations are assembled from the element equations in the same way.

The procedure of assembling the element matrices and vectors is based on the requirement of "compatibility" at the element nodes, by this we mean that at nodes where elements are connected the value (values) of the unknown nodal variable (or variables, if more than one exists at the node) is (are) the same for all the elements connecting at that node. The consequence of this rule is the basis for the assembly process, which is an essential part of every finite element solution. If E and N denote the total number of element and nodal unknowns (degrees of freedom) respectively, $\underline{U}$ denotes the vector of N nodal degrees of freedom, $\underline{K}$ the assembled system characteristic matrix (master matrix) of order $(N^{\times}N)$ and $\underline{f}$ the characteristic vector of order N, then the global characteristic matrix (master matrix) and the global characteristic vector can be obtained by algebraic addition,

$$\underline{K} = \sum_{e=1}^{E} K^{(e)} , \tag{3.55}$$

and,

$$\underline{f} = \sum_{e=1}^{E} f^{(e)} , \tag{3.56}$$

where $K^{(e)}$ and $f^{(e)}$ are the element characteristic matrix and the element characteristic vector respectively.

The procedure is illustrated with reference to the assemblage of the two dimensional problem shown below in Figure (3.22), with the local numbering of each element indicated at the corners within each element. Since there is one degree of freedom for each node, each element has three degrees of freedom. There are 11 degrees of freedom for the entire domain. Thus, the order of $\underline{K}$ and $\underline{f}$ will be (11×11) and (11×1) respectively.

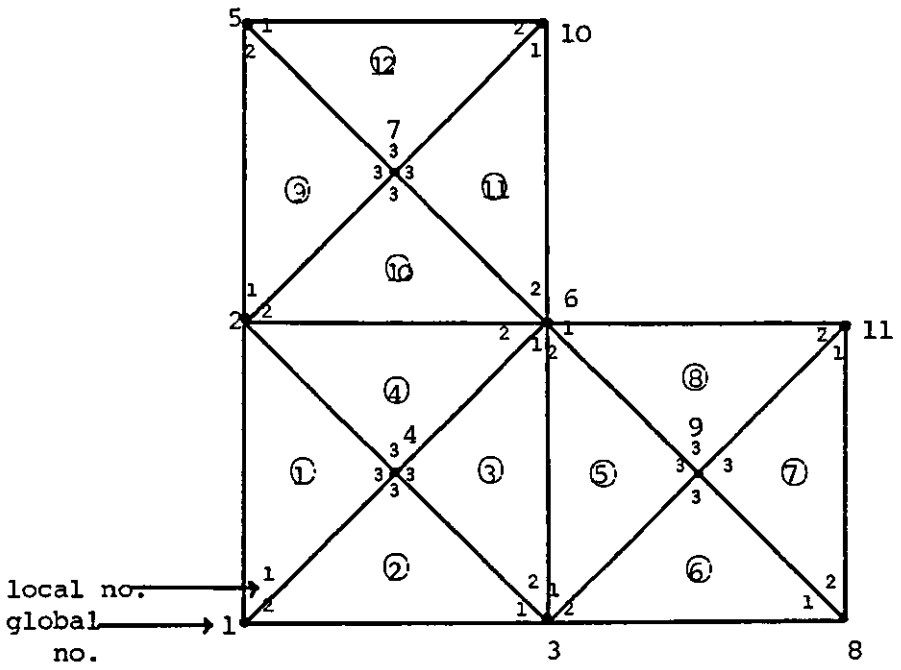


FIGURE 3.22: Local and global numbers in a finite element division of a domain

Table (3.2), emphasizes that the local number of each element is just a way of indicating the ordering of the degrees of freedom in an element while the global numbering scheme as indicated in Figure (3.22) and Table (3.2) which establishes the identification of these nodes and

elements which is an essential part of the solution process.

Once the numbering scheme has been established for the finite element mesh, we must create the record of which nodes belong to each elements. This record given as input to the computer program or generated internally by the program, serves to define the connectivity of the element mesh. In other words, it gives information on how the elements are joined together.

Elements	Local Numbers	Global Numbers
1	1,2,3	1,2,4
2	"	3,1,4
3	"	6,3,4
4	"	2,6,4
5	"	3,6,9
6	"	8,3,9
7	"	11,8,9
8	"	6,11,9
9	"	2,5,7
10	"	6,2,7
11	"	10,6,7
12	"	5,10,7

TABLE 3.2: The Local and the Global numbers for the elements of the problem in Figure (3.22)

Having specified the record of which nodes belong to each elements, which is simply the ordered numbering of the nodes, we can summarize the

general procedure of assembly in the following steps:

1. We set up a (N×N) null matrix and (N×1) null vector (all zero-entries), where N equals the number of system nodal unknowns.
2. Then, starting with one element, transform the element equations from local to global coordinates, if these two coordinate systems do not coincide.
3. Perform any necessary matrix operations on the element matrices, where some times we have one or more nodes which have no connectivity, When this occurs, it is necessary to eliminate the nodal unknowns or degree of freedom associated with these nodes.
4. Using the established correspondence between local and global numbering schemes, change to global indices.
   (i) the subscript indices of the coefficients in the square matrix
   (ii) the single subscript index of the terms in the column matrix.
5. Insert these terms into the corresponding (N×N) and (N×1) master matrices in the locations designated by their indices. Each time that a term is placed in a location where another term has already been placed, it is added to whatever value is there.
6. Return to step 2, and repeat this procedure for one element after another until all the elements have been treated.

The result will be the (N×N) master matrix $\underline{K}$, and (N×1) vector $\underline{f}$. The complete system equations are then,

$$\underline{KU} = \underline{f} \ , \tag{3.57}$$

where $\underline{U}$ is the column vector of nodal unknowns for the assemblage. The generality of this assembly process for the finite element method

offers a definite advantage. Once a computer program for the assembly process has been developed for the solution of one particular class of problems by the finite element method, it may be used again for the finite element solution of other classes of problems.

In fact, the procedure is applicable equally well to all types of problems. We now consider developing the expanded element matrices for our two-dimensional problem in Figure (3.22).

For the first element ① , the coefficients of element matrix $K^{(1)}$ and the element vector $f^{(1)}$ can be written as shown in Table (3.3a) and (3.3b), respectively, the location of any component $k_{ij}^{(1)}$ is identified by the global degrees of freedom $U_{m,n}$ corresponding to the local degrees of freedom $U_{i,j}$, respectively, for i=1,2,3 and j=1,2,3.

Thus, the location of the (N×N) components $K^{(1)}$ in $\underline{K}$ will be shown in Table (3.3b), similarly, the location of the components of the vector $f^{(1)}$ will also be shown in Table (3.3b). By proceeding in a similar way for elements e=2,...,12, the final master matrix $\underline{K}$ and the vector $\underline{f}$ can be obtained as given in Tables (3.16) and (3.17) respectively.

For element ① the corresponding relation between the local and global numbering schemes indicates that the following holds,

The local numbering					The corresponding global numbering				
U_i, U_j →	1	2	3		U_m, U_n →	1	2	4	
↓1	$k_{11}^{(1)}$	$k_{12}^{(1)}$	$k_{13}^{(1)}$		↓1	$k_{11}^{(1)}$	$k_{12}^{(1)}$	$k_{14}^{(1)}$	
2	$k_{21}^{(1)}$	$k_{22}^{(1)}$	$k_{23}^{(1)}$	→	2	$k_{21}^{(1)}$	$k_{22}^{(1)}$	$k_{24}^{(1)}$	
3	$k_{31}^{(1)}$	$k_{32}^{(1)}$	$k_{33}^{(1)}$		4	$k_{41}^{(1)}$	$k_{42}^{(1)}$	$k_{44}^{(1)}$	

for the vector $f^{(1)}$ of the element ① the correspondence relation between the local and global numbering schemes indicates that the following holds,

The local numbering			The corresponding global numbering	
1	$f_1^{(1)}$		1	$f_1^{(1)}$
2	$f_2^{(1)}$	→	2	$f_2^{(1)}$
3	$f_3^{(1)}$		4	$f_4^{(1)}$

TABLE 3.3a The correspondence between the local and global numbering schemes for both coefficient element matrix and element vector of element ①

Hence, when these coefficients are inserted into the expanded matrix $K^{(1)}$ and the expanded vector $f^{(1)}$, we have,

1. The location of $K^{(1)}$ in $\underline{K}$.

$$K^{(1)} = \begin{array}{c} \text{Global} \rightarrow \\ \downarrow \end{array} \begin{array}{c} \begin{array}{ccccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \end{array} \\ \begin{bmatrix} k_{11}^{(1)} & k_{12}^{(1)} & & k_{13}^{(1)} & & & & & & & \\ k_{21}^{(1)} & k_{22}^{(1)} & & k_{23}^{(1)} & & & & & & & \\ & & & & & & & & & & \\ k_{31}^{(1)} & k_{32}^{(1)} & & k_{33}^{(1)} & & & & \bigcirc & & & \\ & & & & & & & & & & \\ & & \bigcirc & & & & & & & & \\ & & & & & & & & & & \\ & & & & & & & & & & \\ & & & & & & & & & & \end{bmatrix} \end{array} \begin{array}{l} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \end{array}$$

2. The location of $f^{(1)}$ in $\underline{f}$.

$$f^{(1)} = \begin{array}{c} \text{Global} \\ \downarrow \end{array} \begin{array}{l} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \end{array} \begin{bmatrix} f_1^{(1)} \\ f_2^{(1)} \\ \\ f_3^{(1)} \\ \\ \bigcirc \\ \\ \\ \\ \\ \end{bmatrix}$$

TABLE 3.3b: The location of both $K^{(1)}$ and $f^{(1)}$ in $\underline{K}$ and $\underline{f}$

For element ②, the correspondence relation between the local and global numbering schemes indicates that the following holds.

The local numbering				The corresponding global numbering			
U_i, $U_j \rightarrow$	1	2	3	U_m, $U_n \rightarrow$	3	1	4
1	$k_{11}^{(1)}$	$k_{12}^{(1)}$	$k_{13}^{(1)}$	3	$k_{33}^{(2)}$	$k_{31}^{(2)}$	$k_{34}^{(2)}$
2	$k_{21}^{(1)}$	$k_{22}^{(1)}$	$k_{23}^{(1)}$	1	$k_{13}^{(2)}$	$k_{11}^{(2)}$	$k_{14}^{(2)}$
3	$k_{31}^{(1)}$	$k_{32}^{(1)}$	$k_{33}^{(1)}$	4	$k_{43}^{(2)}$	$k_{41}^{(2)}$	$k_{44}^{(2)}$

Also, for the vector $f^{(2)}$ of element ②, the correspondence relation between the local and global numbering schemes indicates that the following holds.

The local numbering			The corresponding global numbering	
1	$f_1^{(2)}$		3	$f_3^{(2)}$
2	$f_2^{(2)}$	$\rightarrow$	1	$f_1^{(2)}$
3	$f_3^{(2)}$		4	$f_4^{(2)}$

TABLE 3.4a: The correspondence between the local and the global numbering schemes, for both element matrix and element vector of element number ②

Hence, when these coefficients are inserted into the expanded matrix $K^{(1)}$ and the expanded vector $f^{(1)}$ we have,

1. The location of $K^{(2)}$ in K

$$K^{(2)} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \end{array} \begin{array}{c} \begin{array}{ccccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \end{array} \\ \begin{bmatrix} k_{22}^{(2)} & & k_{21}^{(2)} & k_{23}^{(2)} & & & & & & & \\ & & & & & & & & & & \\ k_{12}^{(2)} & & k_{11}^{(2)} & k_{13}^{(2)} & & & & & \mathrm{O} & & \\ k_{32}^{(2)} & & k_{31}^{(2)} & k_{33}^{(2)} & & & & & & & \\ & & & & & & & & & & \\ & & & \mathrm{O} & & & & & & & \\ & & & & & & & & & & \\ & & & & & & & & & & \end{bmatrix} \end{array}$$

2. The location of $f^{(2)}$ in $\underline{f}$

$$f^{(2)} = \begin{array}{c} \text{Global} \\ \downarrow \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \end{array} \begin{bmatrix} f_2^{(2)} \\ \\ f_1^{(2)} \\ f_3^{(2)} \\ \\ \\ \mathrm{O} \\ \\ \\ \\ \end{bmatrix}$$

TABLE 3.4b: The location of both $K^{(2)}$ and $f^{(2)}$ in $\underline{K}$ and $\underline{f}$.

For element ③ the correspondence between local and global numbering schemes indicates that the following holds,

The local numbering					The corresponding global numbering			
U_i, U_j →	1	2	3		U_m, U_n →	6	3	4
↓					↓			
1	$k_{11}^{(3)}$	$k_{12}^{(3)}$	$k_{13}^{(3)}$		6	$k_{66}^{(3)}$	$k_{63}^{(3)}$	$k_{64}^{(3)}$
2	$k_{21}^{(3)}$	$k_{22}^{(3)}$	$k_{23}^{(3)}$	→	3	$k_{36}^{(3)}$	$k_{33}^{(3)}$	$k_{34}^{(3)}$
3	$k_{31}^{(3)}$	$k_{32}^{(3)}$	$k_{33}^{(3)}$		4	$k_{46}^{(3)}$	$k_{43}^{(3)}$	$k_{44}^{(3)}$

for the vector $f^{(3)}$ of element ③ correspondence between local and global numbering schemes indicates that the following holds,

The local			The corresponding global	
1	$f_1^{(3)}$		6	$f_6^{(3)}$
2	$f_2^{(3)}$	→	3	$f_3^{(3)}$
3	$f_3^{(3)}$		4	$f_4^{(3)}$

TABLE 3.5a: The correspondence between the local and the global numbering schemes, for both coefficient elements matrix and element vector of element number ③

Hence, when these coefficients are inserted into the expanded matrix $K^{(3)}$ and the expanded vector $f^{(3)}$ we have,

1. The location of $K^{(3)}$ in $\underline{K}$

Global →	1	2	3	4	5	6	7	8	9	10	11
1											
2											
3			$k_{22}^{(3)}$	$k_{23}^{(3)}$		$k_{21}^{(3)}$					
4			$k_{32}^{(3)}$	$k_{33}^{(3)}$		$k_{31}^{(3)}$			O		
5											
6			$k_{12}^{(3)}$	$k_{13}^{(3)}$		$k_{11}^{(3)}$					
7											
8											
9											
10		O									
11											

$K^{(3)} =$

2. The location of $f^{(3)}$ in $\underline{f}$

Global	
1	
2	
3	$f_2^{(3)}$
4	$f_3^{(3)}$
5	
6	$f_1^{(3)}$
7	
8	
9	
10	O
11	

$f^{(3)} =$

TABLE 3.5b: The location of both $K^{(3)}$ and $f^{(3)}$ in $\underline{K}$ and $\underline{f}$

For element ④ the correspondence between local and global numbering schemes indicates that the following holds.

The local numbering					The corresponding global numbering			
U_i ↓	U_j → 1	2	3		U_m, U_n → 2	6	4
1	$k_{11}^{(4)}$	$k_{12}^{(4)}$	$k_{13}^{(4)}$		↓ 2	$k_{22}^{(4)}$	$k_{26}^{(4)}$	$k_{24}^{(4)}$
2	$k_{21}^{(4)}$	$k_{22}^{(4)}$	$k_{23}^{(4)}$	→	6	$k_{62}^{(4)}$	$k_{66}^{(4)}$	$k_{64}^{(4)}$
3	$k_{31}^{(4)}$	$k_{32}^{(4)}$	$k_{33}^{(4)}$		4	$k_{42}^{(4)}$	$k_{46}^{(4)}$	$k_{44}^{(4)}$

for the vector $f^{(4)}$ of element ④ the correspondence between local and the global numbering schemes indicates that the following holds.

The local			The corresponding global	
1	$f_1^{(4)}$		2	$f_2^{(4)}$
2	$f_2^{(4)}$	→	6	$f_6^{(4)}$
3	$f_3^{(4)}$		4	$f_4^{(4)}$

TABLE 3.6a: The correspondence between the local and the global numbering schemes for both coefficient element matrix and element vector of element number ④

Hence, when these coefficients are inserted into the expanded matrix $K^{(4)}$, and the expanded vector $f^{(4)}$, we have

1. The location of $K^{(4)}$ in K

$$K^{(4)} = \begin{array}{c} \text{Global} \rightarrow \\ \downarrow \end{array} \begin{array}{c} \begin{array}{ccccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \end{array} \\ \begin{bmatrix} & & & & & & & & & & \\ & k_{11}^{(4)} & & k_{13}^{(4)} & & k_{12}^{(4)} & & & & & \\ & & & & & & & & & & \\ & k_{31}^{(4)} & & k_{33}^{(4)} & & k_{32}^{(4)} & & & & & \\ & & & & & & & & & & \\ & k_{21}^{(4)} & & k_{23}^{(4)} & & k_{22}^{(4)} & & & & & \\ & & & & & & & & & & \\ & & & & & & & & & & \\ & & & & & & & & & & \\ & & & & & & & & & & \end{bmatrix} \end{array} \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \end{array}$$

2. The location of $f^{(4)}$ in $\underline{f}$

$$f^{(4)} = \begin{array}{c} \text{Global} \\ \downarrow \end{array} \begin{array}{c} \begin{bmatrix} \\ f_1^{(4)} \\ \\ f_3^{(4)} \\ \\ f_2^{(4)} \\ \\ \\ \\ \\ \end{bmatrix} \end{array} \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \end{array}$$

TABLE 3.6b: The location of both $K^{(4)}$ and $f^{(4)}$ in $\underline{K}$ and $\underline{f}$

For element ⑤ the correspondence between local and global numbering schemes indicate that the following holds,

The local numbering					The corresponding global numbering			
$U_i, U_j \rightarrow$	1	2	3		$U_m, U_n \rightarrow$	3	6	9
$\downarrow$ 1	$k_{11}^{(5)}$	$k_{12}^{(5)}$	$k_{13}^{(5)}$		$\downarrow$ 3	$k_{33}^{(5)}$	$k_{36}^{(5)}$	$k_{39}^{(5)}$
2	$k_{21}^{(5)}$	$k_{22}^{(5)}$	$k_{23}^{(5)}$	$\rightarrow$	6	$k_{63}^{(5)}$	$k_{66}^{(5)}$	$k_{69}^{(5)}$
3	$f_{31}^{(5)}$	$k_{32}^{(5)}$	$k_{33}^{(5)}$		9	$k_{93}^{(5)}$	$k_{96}^{(5)}$	$k_{99}^{(5)}$

for the vector $f^{(5)}$ of element ⑤ correspondence between local and global numbering schemes indicates that the following holds,

The local			The corresponding global	
1	$f_1^{(5)}$		3	$f_3^{(5)}$
2	$f_2^{(5)}$	$\rightarrow$	6	$f_6^{(5)}$
3	$f_3^{(5)}$		9	$f_9^{(5)}$

TABLE 3.7a: The correspondence between the local and the global numbering schemes for both coefficient element matrix and element vector of element number ⑤

Hence, when these coefficients are inserted into the expanded matrix $K^{(5)}$ and the expanded vector $f^{(5)}$ we have,

1. The location of $K^{(5)}$ in $\underline{K}$

$$K^{(5)} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \end{array} \begin{array}{c} \begin{array}{ccccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \end{array} \\ \begin{bmatrix} & & & & & & & & & & \\ & & & & & & & & & & \\ & & k_{11}^{(5)} & & & k_{12}^{(5)} & & & k_{13}^{(5)} & & \\ & & & & & & & & & \bigcirc & \\ & & & & & & & & & & \\ & & k_{21}^{(5)} & & & k_{22}^{(5)} & & & k_{23}^{(5)} & & \\ & & & & & & & & & & \\ & & & & & & & & & & \\ & & k_{31}^{(5)} & & & k_{32}^{(5)} & & & k_{33}^{(5)} & & \\ & \bigcirc & & & & & & & & & \\ & & & & & & & & & & \end{bmatrix} \end{array}$$

2. The location of $f^{(5)}$ in $\underline{f}$

$$f^{(5)} = \begin{array}{c} \text{Global} \\ \downarrow \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \end{array} \begin{bmatrix} \\ \\ f_1^{(5)} \\ \\ \\ f_2^{(5)} \\ \\ \\ f_3^{(5)} \\ \\ \end{bmatrix}$$

TABLE 3.7b: The location of both $K^{(5)}$ and $f^{(5)}$ in $\underline{K}$ and $\underline{f}$

For element number ⑥ the correspondence between local and global numbering schemes indicate that the following holds

The local numbering					The corresponding global numbering			
U_i, U_j →	1	2	3		U_m, U_n →	8	3	9
1	$k_{11}^{(6)}$	$k_{12}^{(6)}$	$k_{13}^{(6)}$		8	$k_{88}^{(6)}$	$k_{83}^{(6)}$	$k_{89}^{(6)}$
2	$k_{21}^{(6)}$	$k_{22}^{(6)}$	$k_{23}^{(6)}$	→	3	$k_{38}^{(6)}$	$k_{33}^{(6)}$	$k_{39}^{(6)}$
3	$k_{31}^{(6)}$	$k_{32}^{(6)}$	$k_{33}^{(6)}$		9	$k_{98}^{(6)}$	$k_{93}^{(6)}$	$k_{99}^{(6)}$

for the vector $f^{(5)}$ of element ⑥ the correspondence between the local and the global numbering schemes indicates that the following holds

The local			The corresponding global	
1	$f_1^{(6)}$		8	$f_8^{(6)}$
2	$f_2^{(6)}$	→	3	$f_3^{(6)}$
3	$f_3^{(6)}$		9	$f_9^{(6)}$

TABLE 3.8a The correspondence between the local and the global numbering schemes for element both coefficient matrix and element matrix of element number ⑥

Hence, when these coefficients are inserted into the expanded matrix $K^{(6)}$ and the expanded vector $f^{(6)}$, we have,

1. The location of $K^{(6)}$ in $\underline{K}$

$$K^{(6)} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \end{array} \begin{array}{c} \begin{array}{ccccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \end{array} \\ \begin{bmatrix} & & & & & & & & & & \\ & & & & & & & & & & \\ & & k_{22}^{(6)} & & & & & k_{21}^{(6)} & k_{23}^{(6)} & & \\ & & & & & & & & & & \\ & & & & & \bigcirc & & & & & \\ & & & & & & & & & & \\ & & & & & & & & & & \\ & & k_{12}^{(6)} & & & & & k_{11}^{(6)} & k_{13}^{(6)} & & \\ & & k_{32}^{(6)} & & & & & k_{31}^{(6)} & k_{33}^{(6)} & & \\ & & & & & & & & & & \\ & \bigcirc & & & & & & & & & \end{bmatrix} \end{array}$$

2. The location of $f^{(6)}$ in $\underline{f}$,

$$f^{(6)} = \begin{array}{c} \text{Global} \\ \downarrow \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \end{array} \begin{bmatrix} \\ \\ f_2^{(6)} \\ \\ \\ \\ \\ \\ f_1^{(6)} \\ f_3^{(6)} \\ \\ \end{bmatrix}$$

TABLE 3.8b: The location of both $K^{(6)}$ and $f^{(6)}$ in $\underline{K}$ and $\underline{f}$

For element number ⑦ the corresponding local and global numbering schemes indicate that the following holds

The local numbering					The corresponding global numbering			
$U_i, U_j \rightarrow$	1	2	3		$U_m, U_n \rightarrow$	11	8	9
$\downarrow$ 1	$k_{11}^{(7)}$	$k_{12}^{(7)}$	$k_{13}^{(7)}$		$\downarrow$ 11	$k_{11,11}^{(7)}$	$k_{11,8}^{(7)}$	$k_{11,9}^{(7)}$
2	$k_{21}^{(7)}$	$k_{22}^{(7)}$	$k_{23}^{(7)}$	$\rightarrow$	8	$k_{8,11}^{(7)}$	$k_{88}^{(7)}$	$k_{89}^{(7)}$
3	$k_{31}^{(7)}$	$k_{32}^{(7)}$	$k_{33}^{(7)}$		9	$k_{9,11}^{(7)}$	$k_{98}^{(7)}$	$k_{99}^{(7)}$

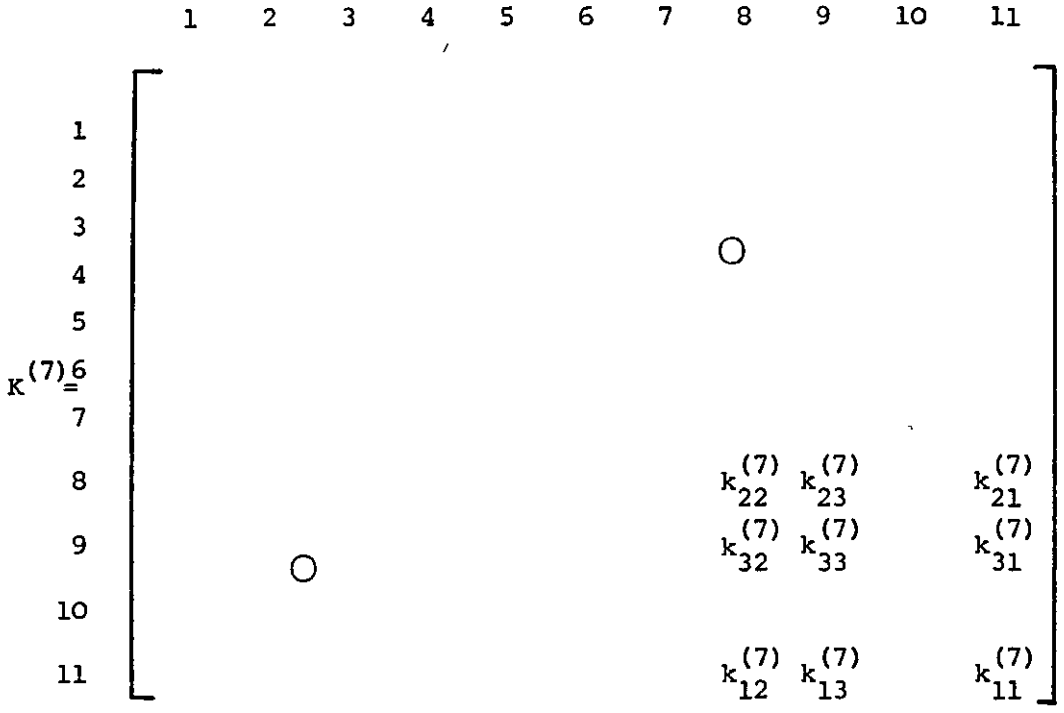
for the vector $f^{(7)}$ of element ⑦ the correspondence between the local and the global numbering schemes indicates that the following holds,

The local			The corresponding global	
1	$f_1^{(7)}$		11	$f_{11}^{(7)}$
2	$f_2^{(7)}$	$\rightarrow$	8	$f_8^{(7)}$
3	$f_3^{(7)}$		9	$f_9^{(7)}$

TABLE 3.9a: The correspondence between the local and the global numbering schemes for both coefficient matrix and element vector of element number ⑦

Hence, when coefficients are inserted into the expanded matrix $K^{(7)}$ and the expanded vector $f^{(7)}$, we have,

1. The location of $K^{(7)}$ in $\underline{K}$



2. The location of $f^{(7)}$ in $\underline{f}$,

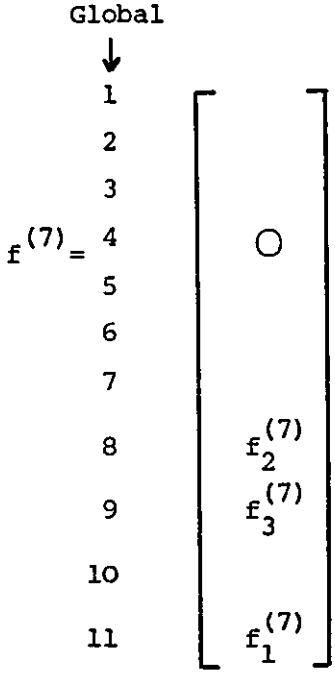


TABLE 3.9b: The location of both $K^{(7)}$ and $f^{(7)}$ in $\underline{K}$ and $\underline{f}$

For element number ⑧ , the correspondence between local and global numbering schemes indicates that the following holds

The local numbering					The corresponding global numbering			
U_i, U_j →	1	2	3		U_m, U_n →	6	11	9
↓					↓			
1	$k_{11}^{(8)}$	$k_{12}^{(8)}$	$k_{13}^{(8)}$		6	$k_{66}^{(8)}$	$k_{6,11}^{(8)}$	$k_{69}^{(8)}$
2	$k_{21}^{(8)}$	$k_{22}^{(8)}$	$k_{23}^{(8)}$	→	11	$k_{11,6}^{(8)}$	$k_{11,11}^{(8)}$	$k_{11,9}^{(8)}$
3	$k_{31}^{(8)}$	$k_{32}^{(8)}$	$k_{33}^{(8)}$		9	$k_{96}^{(8)}$	$k_{9,11}^{(8)}$	$k_{99}^{(8)}$

and for the vector $f^{(8)}$ of element ⑧ , correspondence between local and global numbering schemes indicates that the following holds,

The local			The corresponding global	
1	$f_1^{(8)}$		6	$f_6^{(8)}$
2	$f_2^{(8)}$	→	11	$f_{11}^{(8)}$
3	$f_3^{(8)}$		9	$f_9^{(8)}$

TABLE 3.10a: The correspondence between the local and the global numbering schemes for both coefficient element matrix and element vector of element number ⑧

Hence, when coefficients are inserted into the expanded matrix $K^{(8)}$ and the expanded vector $f^{(8)}$, we have,

1. The location of $K^{(8)}$ in $\underline{K}$

$$K^{(8)} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \end{array} \begin{array}{c} \begin{array}{ccccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \end{array} \\ \begin{bmatrix} & & & & & & & & & & \\ & & & & & & & & \circ & & \\ & & & & & & & & & & \\ & & & & & & & & & & \\ & & & & & & & & & & \\ & & & & & k_{11}^{(8)} & & & k_{13}^{(8)} & & k_{12}^{(8)} \\ & & & & & & & & & & \\ & & & & & & & & & & \\ & & \circ & & & k_{31}^{(8)} & & & k_{33}^{(8)} & & k_{32}^{(8)} \\ & & & & & & & & & & \\ & & & & & k_{21}^{(8)} & & & k_{23}^{(8)} & & k_{22}^{(8)} \end{bmatrix} \end{array}$$

2. The location of $f^{(8)}$ in $\underline{f}$,

$$f^{(8)} = \begin{array}{c} \text{Global} \\ \downarrow \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \end{array} \begin{bmatrix} \\ \\ \circ \\ \\ \\ f_1^{(8)} \\ \\ \\ f_3^{(8)} \\ \\ f_2^{(8)} \end{bmatrix}$$

TABLE 3.10b: The location of both $K^{(8)}$ and $f^{(8)}$ in K and $\underline{f}$

For element number ⑨ , the correspondence between local and global numbering schemes indicates that the following holds,

The local numbering				The corresponding global numbering			
U_i, U_j →	1	2	3	U_m, U_n →	2	5	7
↓ 1	$k_{11}^{(9)}$	$k_{12}^{(9)}$	$k_{13}^{(9)}$	↓ 2	$k_{22}^{(9)}$	$k_{25}^{(9)}$	$k_{27}^{(9)}$
2	$k_{21}^{(9)}$	$k_{22}^{(9)}$	$k_{23}^{(9)}$	5	$k_{52}^{(9)}$	$k_{55}^{(9)}$	$k_{57}^{(9)}$
3	$k_{31}^{(9)}$	$k_{32}^{(9)}$	$k_{33}^{(9)}$	7	$k_{72}^{(9)}$	$k_{75}^{(9)}$	$k_{77}^{(9)}$

and for the vector $f^{(9)}$ of element ⑨ , correspondence between local and global numbering schemes indicates that the following holds,

The local		The corresponding global	
1	$f_1^{(9)}$	2	$f_6^{(9)}$
2	$f_2^{(9)}$	5	$f_{11}^{(9)}$
3	$f_3^{(9)}$	7	$f_9^{(9)}$

TABLE 3.11a: The correspondence between the local and the global numbering schemes for both coefficient element matrix and element vector of element number ⑨

Hence, when coefficients are inserted into the expanded matrix $K^{(9)}$ and the expanded vector $f^{(9)}$, we have,

1. The location of $\underline{K}^{(9)}$ in $\underline{K}$

$$K^{(9)} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \end{array} \begin{array}{c} \begin{array}{ccccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \end{array} \\ \begin{bmatrix} & & & & & & & & & & \\ & k_{11}^{(9)} & & & k_{12}^{(9)} & & k_{13}^{(9)} & & & & \\ & & & & & & & & & & \\ & & & & & & & & & & \\ & k_{21}^{(9)} & & & k_{22}^{(9)} & & k_{23}^{(9)} & & & & \\ & & & & & & & & & & \\ & k_{31}^{(9)} & & & k_{32}^{(9)} & & k_{33}^{(9)} & & & & \\ & & & & & & & & & & \\ & & & & & & & & & & \\ & & & & & & & & & & \end{bmatrix} \end{array}$$

2. The location of $f^{(9)}$ in $\underline{f}$,

$$f^{(9)} = \begin{array}{c} \text{Global} \\ \downarrow \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \end{array} \begin{bmatrix} \\ f_1^{(9)} \\ \\ \\ f_2^{(9)} \\ \\ f_3^{(9)} \\ \\ \\ \\ \end{bmatrix}$$

TABLE 3.11b: The location of both $K^{(9)}$, and $f^{(9)}$ in K and $\underline{f}$

For element (10) , the correspondence between local and global numbering schemes indicates that the following holds,

The local numbering					The corresponding global numbering			
U_i, U_j →	1	2	3		U_m, U_n →	6	2	7
↓					↓			
1	$k_{11}^{(10)}$	$k_{12}^{(10)}$	$k_{13}^{(10)}$		6	$k_{66}^{(10)}$	$k_{62}^{(10)}$	$k_{67}^{(10)}$
2	$k_{21}^{(10)}$	$k_{22}^{(10)}$	$k_{23}^{(10)}$	→	2	$k_{26}^{(10)}$	$k_{22}^{(10)}$	$k_{27}^{(10)}$
3	$k_{31}^{(10)}$	$k_{32}^{(10)}$	$k_{33}^{(10)}$		7	$k_{76}^{(10)}$	$k_{72}^{(10)}$	$k_{77}^{(10)}$

and for the element vector $f^{(10)}$ of element (10) , correspondence between local and global numbering schemes indicates that the following holds,

The local			The corresponding global	
1	$f_1^{(10)}$		6	$f_6^{(10)}$
2	$f_2^{(10)}$	→	2	$f_2^{(10)}$
3	$f_3^{(10)}$		7	$f_7^{(10)}$

TABLE 3.12a: The correspondence between the local and the global numbering schemes for both coefficient element matrix and element vector of element number (10)

Hence, when coefficients are inserted into the expanded matrix $K^{(10)}$ and the expanded vector $f^{(10)}$, we have,

1. The location of $K^{(10)}$ in $\underline{K}$

$$K^{(10)} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \end{array} \begin{array}{c} \begin{array}{ccccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \end{array} \\ \begin{bmatrix} & & & & & & & & & & \\ & k_{22}^{(10)} & & & & k_{21}^{(10)} & k_{23}^{(10)} & & & & \\ & & & & & & & & & & \\ & & & & & & & & & & \\ & & & & & & & & & \circ & \\ & k_{12}^{(10)} & & & & k_{11}^{(10)} & k_{13}^{(10)} & & & & \\ & k_{32}^{(10)} & & & & k_{31}^{(10)} & k_{33}^{(10)} & & & & \\ & & & & & & & & & & \\ & & & \circ & & & & & & & \\ & & & & & & & & & & \end{bmatrix} \end{array}$$

2. The location of $f^{(10)}$ in $\underline{f}$,

$$\begin{array}{c} \text{Global} \\ \downarrow \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \end{array} \begin{bmatrix} \\ f_2^{(10)} \\ \\ \\ \\ f_1^{(10)} \\ f_3^{(10)} \\ \\ \circ \\ \\ \end{bmatrix}$$

TABLE 3.12b: The location of both $K^{(10)}$ and $f^{(10)}$ in $\underline{K}$ and $\underline{f}$

For the element (11), the correspondence between local and global numbering schemes indicates that the following holds,

The local numbering					The corresponding global numbering				
U_i, U_j →	1	2	3		U_m, U_n →	10	6	7
↓1	$k_{11}^{(11)}$	$k_{12}^{(11)}$	$k_{13}^{(11)}$		↓10	$k_{10,10}^{(11)}$	$k_{10,6}^{(11)}$	$k_{10,7}^{(11)}$
2	$k_{21}^{(11)}$	$k_{22}^{(11)}$	$k_{23}^{(11)}$	→	6	$k_{6,10}^{(11)}$	$k_{66}^{(11)}$	$k_{67}^{(11)}$
3	$k_{31}^{(11)}$	$k_{32}^{(11)}$	$k_{33}^{(11)}$		7	$k_{7,10}^{(11)}$	$k_{76}^{(11)}$	$k_{77}^{(11)}$

and for the element vector $f^{(11)}$ of element (11), correspondence between local and global schemes indicates that the following holds,

The local			The corresponding global	
1	$f_1^{(11)}$		10	$f_{10}^{(11)}$
2	$f_2^{(11)}$	→	6	$f_6^{(11)}$
3	$f_3^{(11)}$		7	$f_7^{(11)}$

TABLE 3.13a: The correspondence between the local and the global numbering schemes for both coefficient element matrix and element vector of element number (11)

Hence, when coefficients are inserted into the expanded matrix $K^{(11)}$ and the expanded vector $f^{(11)}$, we have,

1. The location of $K^{(11)}$ in $\underline{K}$

$$K^{(11)} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \end{array} \begin{array}{c} \begin{array}{ccccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \end{array} \\ \begin{bmatrix} & & & & & & & & & & \\ & & & & & & & & \circ & & \\ & & & & & & & & & & \\ & & & & & & & & & & \\ & & & & & & & & & & \\ & & & & & k_{22}^{(11)} & k_{23}^{(11)} & & & k_{21}^{(11)} & \\ & & & & & k_{32}^{(11)} & k_{33}^{(11)} & & & k_{31}^{(11)} & \\ & & \circ & & & & & & & & \\ & & & & & & & & & & \\ & & & & & k_{12}^{(11)} & k_{13}^{(11)} & & & k_{11}^{(11)} & \\ & & & & & & & & & & \end{bmatrix} \end{array}$$

2. The location of $f^{(11)}$ in $\underline{f}$,

$$f^{(11)} = \begin{array}{c} \text{Global} \\ \downarrow \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \end{array} \begin{bmatrix} \\ \circ \\ \\ \\ \\ f_2^{(11)} \\ f_3^{(11)} \\ \\ \\ f_1^{(11)} \\ \end{bmatrix}$$

TABLE 3.13b: The location of both $K^{(11)}$ and $f^{(11)}$ in $\underline{K}$ and $\underline{f}$

and for the element (12), the correspondence between local and global numbering schemes indicates that the following holds,

The local numbering					The corresponding global numbering				
U_i, U_j →	1	2	3		U_m, U_n →	5	10	7
↓					↓			
1	$k_{11}^{(12)}$	$k_{12}^{(12)}$	$k_{13}^{(12)}$		5	$k_{55}^{(12)}$	$k_{5,10}^{(12)}$	$k_{57}^{(12)}$
2	$k_{21}^{(12)}$	$k_{22}^{(12)}$	$k_{23}^{(12)}$	→	10	$k_{10,5}^{(12)}$	$k_{10,10}^{(12)}$	$k_{10,7}^{(12)}$
3	$k_{31}^{(12)}$	$k_{32}^{(12)}$	$k_{33}^{(12)}$		7	$k_{75}^{(12)}$	$k_{7,10}^{(12)}$	$k_{77}^{(12)}$

and for the element vector $f^{(12)}$, of element (12), correspondence between local and global schemes indicates that the following holds,

The local			The corresponding global	
1	$f_1^{(12)}$		5	$f_5^{(12)}$
2	$f_2^{(12)}$	→	10	$f_{10}^{(12)}$
3	$f_3^{(12)}$		7	$f_7^{(12)}$

TABLE 3.14a: The correspondence between the local and the global numbering schemes for both coefficient element matrix and element vector of the element number (12)

Hence, when coefficients are inserted into the expanded matrix $K^{(12)}$ and the expanded vector $f^{(12)}$, we have,

1. The location of $K^{(12)}$ in $\underline{K}$

	1	2	3	4	5	6	7	8	9	10	11
1											
2							○				
3											
4											
5					$k_{11}^{(12)}$		$k_{13}^{(12)}$			$k_{12}^{(12)}$	
6											
7					$k_{31}^{(12)}$		$k_{33}^{(12)}$			$k_{32}^{(12)}$	
8											
9											
10					$k_{21}^{(12)}$		$k_{23}^{(12)}$			$k_{22}^{(12)}$	
11		○									

2. The location of $f^{(12)}$ in $\underline{f}$,

Global	
1	
2	○
3	
4	
$f^{(12)}$ = 5	$f_1^{(12)}$
6	
7	$f_3^{(12)}$
8	
9	
10	$f_2^{(12)}$
11	

TABLE 3.14b: The location of both $K^{(12)}$ and $f^{(12)}$ in $\underline{K}$ and $\underline{f}$

After assembling the element characteristic matrices $K^{(e)}$ and the element characteristic vectors $f^{(e)}$, the overall or system equations of the entire domain can be written as, equation (3.57), i.e.,

$$\underline{K}\underline{U} = \underline{f} \ , \tag{3.57}$$

These equations cannot be solved for $\underline{U}$ since the matrix $\underline{K}$ will be singular and hence its inverse does not exist. However, for a unique solution of equation (3.57) some boundary or support conditions have to be applied to the equation (3.57), i.e., at least one and some times more than one nodal variable must be specified and thus $\underline{K}$ must be modified to render it non-singular. The required number of specified nodal variables is dictated by the physics of the problem.

There are a number of ways to apply the boundary conditions to equation (3.57), and when they are applied, the number of nodal unknowns and the number of equations to be solved are effectively reduced. However, it is most convenient to introduce the known nodal variables in a way that leaves the original number of equations unchanged and avoids major restructuring of computer storage.

Method 1

To illustrate this method we partition equation (3.57) in the form,

$$\begin{bmatrix} \underline{K}_{11} & \underline{K}_{12} \\ \underline{K}_{21} & \underline{K}_{22} \end{bmatrix} \begin{bmatrix} \underline{U}_1 \\ \underline{U}_2 \end{bmatrix} = \begin{bmatrix} \underline{f}_1 \\ \underline{f}_2 \end{bmatrix} \ , \tag{3.58}$$

where $\underline{U}_2$ is assumed to be the vector of specified nodal variables, and $\underline{U}_1$ is a vector of unknown nodal variables and $\underline{f}_2$ will be the vector of unknown nodal variables.

Equation (3.58) can be written as,

$$\underline{K}_{11}\underline{U}_1 + \underline{K}_{12}\underline{U}_2 = \underline{f}_1 \ ,$$

i.e.,
$$\underline{K}_{11}\underline{U}_1 = \underline{f}_1 - \underline{K}_{12}\underline{U}_2 \ , \tag{3.59}$$

and
$$\underline{K}_{12}^T\underline{U}_1 + \underline{K}_{22}\underline{U}_2 = \underline{f}_2 \ , \tag{3.60}$$

Here $\underline{K}_{11}$ will not be singular and hence equation (3.59) can be solved to obtain,

$$\underline{U}_1 = \underline{K}_{11}^{-1}(\underline{f}_1 - \underline{K}_{12})\underline{U}_2 \ . \tag{3.61}$$

Once $\underline{U}_1$ is known, the vector of unknown nodal variables $\underline{f}_2$ can be found from equation (3.60). In the special case, where all the prescribed nodal variables are equal to zero, we can delete the rows and columns corresponding to $\underline{b}_2$ and state the equations simply as,

$$\underline{K}_{11}\underline{U}_1 = \underline{f}_1 \ . \tag{3.62}$$

Since all the prescribed nodal degrees of freedom usually do not come at the end of the vector $\underline{U}$, the procedure of method 1 involves an awkward renumbering scheme. Even when the prescribed nodal variables are not zero, it can be seen that the rearrangement of equation (3.58) and the solutions of equation (3.59) and (3.60) are time consuming and tend to destroy the bandedness property of the original matrix.

Hence the following equivalent method can be used for incorporating the prescribed boundary conditions $\underline{U}_2$.

Method 2

Equations (3.59) and (3.60) can be written together as,

$$\begin{bmatrix} \underline{K}_{11} & \underline{O} \\ \underline{O} & I \end{bmatrix} \begin{bmatrix} \underline{U}_1 \\ \underline{U}_2 \end{bmatrix} = \begin{Bmatrix} \underline{f}_1 - \underline{K}_{12}\underline{U}_2 \\ \underline{U}_2 \end{Bmatrix} \qquad (3.63)$$

In actual computations, the process indicated in equations (3.63) can be performed without reordering the equations implied by the partitioning as follows:

(i) If U_j is prescribed as $\bar{U}_j$, the characteristic vector $\underline{b}$ is modified as

$$f_i = f_i - k_{ij}\bar{U}_j, \text{ for } i=1,2,\ldots,N.$$

(ii) The rows and columns of $\underline{K}$ corresponding to U_j are made zero except the diagonal element, which is made unity, that is,

$$k_{ji} = k_{ij} = 0, \quad \text{for } i=1,2,\ldots,N$$
$$k_{jj} = 1$$

(iii) The prescribed value of U_j is inverted in the characteristic vector as,

$$f_j = \bar{f}_j .$$

This procedure (i) to (iii) is repeated for all prescribed nodal variables U_j.

It can be noted that this procedure retains the symmetric property of the equations and the matrix $\underline{K}$ can be stored in the band format with little extra programming effort.

To illustrate this procedure for entering the boundary conditions, we consider a simple example with only for system equations. Thus, equation (3.57) expands to the form,

$$\begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} \\ k_{21} & k_{22} & k_{23} & k_{24} \\ k_{31} & k_{32} & k_{33} & k_{34} \\ k_{41} & k_{42} & k_{43} & k_{44} \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} \qquad (3.64)$$

Suppose that for this system nodal variables U_3 and U_4 are specified as

$$U_3 = a_3 \ , \quad U_4 = a_4 \ .$$

When these boundary conditions are inserted, the equations become,

$$\begin{bmatrix} k_{11} & k_{12} & 0 & 0 \\ k_{21} & k_{22} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} f_1 - k_{13}a_3 - k_{14}a_4 \\ f_2 - k_{23}a_3 - k_{24}a_4 \\ a_3 \\ a_4 \end{bmatrix} \qquad (3.65)$$

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

The Assembly Process

Now we observe that the master matrix of equation (3.55) can be obtained by simply adding the matrices $K^{(1)}, K^{(2)}, \ldots, K^{(12)}$.

The mathematical statement of this assembly procedure is as follows

$$\underline{K} = K^{(1)} + K^{(2)} + \ldots + K^{(12)}$$

$$= \sum_{e=1}^{E} K^{(e)} \ ,$$

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

The master matrix $\underline{K}$ of our problem is given in Table (3.14). The same expansion and summation principle also applies for finding the

column vectors,

$$\underline{f} = \sum_{e=1}^{E} f^{(e)} ,$$

where $f^{(e)}$ is the expanded column vector for element e, and E is the total number of elements. The master vector $\underline{f}$ of our problem is given in Table (3.15).

In our example N=12, but in an actual problem there might be several hundred elements. Even if the assemblage contains many different kinds of elements, equation (3.55) still holds and each individual element matrix is expanded (according to the global numbering scheme) to the dimension of the system matrix, and then these matrices are added.

Global → / ↓

	1	2	3	4	5	6	7	8	9	10	11
1	$k_{11}^{(1)}+k_{22}^{(2)}$	$k_{12}^{(1)}$	$k_{21}^{(2)}$	$k_{13}^{(1)}+k_{23}^{(2)}$							
2	$k_{21}^{(1)}$	$k_{22}^{(1)}+k_{11}^{(4)}+k_{11}^{(9)}+k_{22}^{(10)}$		$k_{23}^{(1)}+k_{13}^{(4)}$	$k_{12}^{(9)}$	$k_{12}^{(4)}+k_{21}^{(10)}$	$k_{13}^{(9)}+k_{23}^{(10)}$				
3	$k_{12}^{(2)}$		$k_{11}^{(2)}+k_{22}^{(3)}+k_{11}^{(5)}+k_{12}^{(6)}$	$k_{13}^{(2)}+k_{23}^{(3)}$		$k_{21}^{(3)}+k_{12}^{(5)}$		$k_{21}^{(6)}$	$k_{13}^{(5)}+k_{23}^{(6)}$		
4	$k_{31}^{(1)}+k_{32}^{(2)}$	$k_{32}^{(1)}+k_{31}^{(4)}$	$k_{31}^{(2)}+k_{32}^{(3)}$	$k_{33}^{(1)}+k_{33}^{(2)}+k_{33}^{(3)}+k_{33}^{(4)}$		$k_{31}^{(3)}+k_{32}^{(4)}$					
5		$k_{21}^{(9)}$			$k_{22}^{(9)}+k_{11}^{(12)}$		$k_{23}^{(9)}+k_{13}^{(12)}$			$k_{12}^{(12)}$	
6		$k_{21}^{(4)}+k_{12}^{(10)}$	$k_{12}^{(3)}+k_{21}^{(5)}$	$k_{13}^{(3)}+k_{23}^{(4)}$		$k_{11}^{(3)}+k_{22}^{(4)}+k_{22}^{(5)}+k_{11}^{(8)}+k_{11}^{(10)}+k_{22}^{(11)}$	$k_{13}^{(10)}+k_{22}^{(11)}$		$k_{23}^{(5)}+k_{13}^{(8)}$	$k_{21}^{(11)}$	$k_{12}^{(8)}$
7		$k_{31}^{(9)}+k_{32}^{(10)}$			$k_{32}^{(9)}+k_{31}^{(12)}$		$k_{33}^{(9)}+k_{33}^{(10)}+k_{33}^{(11)}+k_{33}^{(12)}$			$k_{31}^{(11)}+k_{32}^{(12)}$	
8			$k_{12}^{(6)}$					$k_{11}^{(6)}+k_{22}^{(7)}$	$k_{13}^{(6)}+k_{23}^{(7)}$		$k_{21}^{(7)}$
9			$k_{31}^{(5)}+k_{32}^{(6)}$			$k_{32}^{(5)}+k_{31}^{(8)}$		$k_{31}^{(6)}+k_{32}^{(7)}$	$k_{33}^{(5)}+k_{33}^{(6)}+k_{33}^{(7)}+k_{33}^{(8)}$		$k_{31}^{(7)}+k_{32}^{(8)}$
10					$k_{21}^{(12)}$	$k_{12}^{(11)}$	$k_{13}^{(11)}+k_{23}^{(12)}$			$k_{11}^{(11)}+k_{12}^{(12)}$	
11						$k_{21}^{(8)}$		$k_{12}^{(7)}$	$k_{13}^{(7)}+k_{23}^{(8)}$		$k_{11}^{(7)}+k_{22}^{(8)}$

TABLE 3.15a: Assembled master matrix K

Global ↓

$$\underline{f}=f^{(1)}+f^{(2)}+\ldots+f^{(n)} =$$

1	$f_1^{(1)}+f_2^{(2)}$
2	$f_2^{(1)}+f_1^{(4)}+f_1^{(9)}+f_2^{(10)}$
3	$f_1^{(2)}+f_2^{(3)}+f_1^{(5)}+f_2^{(6)}$
4	$f_3^{(1)}+f_3^{(2)}+f_3^{(3)}+f_3^{(4)}$
5	$f_2^{(9)}+f_1^{(12)}$
6	$f_1^{(3)}+f_2^{(4)}+f_2^{(5)}+f_1^{(8)}+f_1^{(10)}+f_2^{(11)}$
7	$f_3^{(9)}+f_3^{(10)}+f_3^{(11)}+f_3^{(12)}$
8	$f_1^{(6)}+f_2^{(7)}$
9	$f_3^{(5)}+f_3^{(6)}+f_3^{(7)}+f_3^{(8)}$
10	$f_1^{(11)}+f_2^{(12)}$
11	$f_1^{(7)}+f_2^{(8)}$

TABLE 3.15b: Assembled master vector $\underline{f}$

As given in section (3.6.2), the finite element analysis leads to a system of matrix equations. After incorporating the boundary conditions in the assembled system as outlined in the section we obtained the final matrix equation which can be solved by using one of the methods described in Chapter 2.

CHAPTER FOUR

A GENERAL PROGRAMMING SYSTEM FOR THE FINITE ELEMENT METHOD

4.1 INTRODUCTION

The general applicability of the finite element method to a wide variety of different engineering and mathematical fields, makes it a powerful and versatile tool. In fact, the method has become one of the most active research areas for applied mathematicians and engineers.

One of the main reasons for the popularity of the method in different applications is that once a general computer program is written, it can be used for the solution of many problems simply by changing the input data.

Although applications are many and different, a typical finite element program consists of a few well defined operations such as:

- *The input description of the mathematical model*
- *The generation of the element matrices*
- *The assembly of elements to form the Jacobian matrix*
- *A solution of the resulting linear or nonlinear system of equations*
- *The calculation of the element characteristics, and the presentation of the results (post-processor).*

Thus, provided a sufficiently general data problem has been defined, the standard operations need to be programmed only once and organized as modules (subprograms) of *a programming system* or subroutine library.

Such a programming system is not intended to be used by itself to solve the problems. It should be used as a tool for the programmer in the development of an executable, special or general purpose program by organizing the modules or building blocks of a programming system. The

subroutines included in the programming system should cover the main operations associated with the finite element analyses. In addition, service routines for operations like data transfer between central memory and peripheral storage, matrix operations, pre- and post-processing, etc. are necessary modules when developing an executable program.

An executable (or application) program may in the present context be characterized as follows:

The user has to describe the geometry, element, mesh, boundary conditions, etc. of the model of the problem in accordance with the input requirements of the application program, and after the program has performed the finite element analysis, the user has to interpret the results.

Hence a number of computer program packages have been developed for the solution of a variety of engineering problems. Some of the programs have been developed in such a general manner (like TWODEPEP) that the user can use the same program for the solution of problems belonging to different branches of application fields with little modification in the input data. A summary of the more widely used packages and their capabilities can be found in NOOR [1981].

Here we will present the programming system TWODEPEP which copes with all parts of a typical finite element program as listed previously. As the success of the "programming system philosophy" depends on the quality and properties of the programming systems we will list some general requirements and discuss our experiences with (TWODEPEP), referring to the listed requirements. This experience comes from the development of the solution to problems in several applications areas.

4.2 GENERAL INFORMATION OF TWODEPEP

TWODEPEP is a production of IMSL which is a Fortran application software finite element package for solving a large class of partial differential equations. New releases of the program are generated at the rate of about one per year.

TWODEPEP is a general purpose, easy to use, finite element program which solves a large class of elliptic (steady-state), parabolic (time dependent), eigenvalue problems, and other problems defined exclusively by partial differential equations in general two-dimensional regions. Applications include elasticity, diffusion, heat conduction, fluid mechanics, potential energy, time-dependent and time-independent, Schrodinger equations, semi-conductor and shell problems. The program includes a preprocessor and a graphical output package. The design priorities of TWODEPEP are in order: generality, easy to use, storage efficiency, accuracy and speed. Most of the methods employed are general and standard proven techniques.

4.3 PROBLEM DEFINITION OF TWODEPEP

The most general form of the differential equations solved by the finite element program TWODEPEP is:

$$C_1 \frac{\partial u}{\partial t} = \frac{\partial}{\partial x}(\sigma_{xx}) + \frac{\partial}{\partial y}(\sigma_{xy}) + F_1$$
$$C_2 \frac{\partial v}{\partial t} = \frac{\partial}{\partial x}(\sigma_{yx}) + \frac{\partial}{\partial y}(\sigma_{yy}) + F_2 \qquad (4.1)$$

In a two dimensional region R, where C_1, C_2 may be constants or a function of (x,y,u,v,t) and $\sigma_{xx}, \sigma_{xy}, \sigma_{yx}, \sigma_{yy}$, F_1 and F_2 are in general functions of $(x,y,u,v,t,u_x,u_y,v_x,v_y)$, with

$$\left.\begin{aligned} u &= FB_1(s,t) \\ v &= FB_2(s,t) \quad \text{, for s on part of the boundary } \partial R_1 \\ \text{and} \quad \sigma_{xx}\ell_x + \sigma_{xy}\ell_y &= GB_1(s,u,v,t) \\ \sigma_{yx}\ell_x + \sigma_{yy}\ell_y &= GB_2(s,u,v,t) \end{aligned}\right\} \qquad (4.2)$$

for s on the other part of the boundary ∂R_2. ℓ_x, ℓ_y are the unit outward normal to the boundary, and,

$$\left.\begin{aligned} u &= u_o(x,y) \\ v &= v_o(x,y) \end{aligned}\right\} \text{, for } t=T_o \text{ .} \qquad (4.3)$$

It is assumed that the problem is self-adjoint, although TWODEPEP can solve some non-symmetric problems, but with less efficiency and accuracy.

However, TWODEPEP can also solve several simultaneous equations of the above form, Elliptic ($C_1=C_2=0$), and eigenvalue problems can also be solved. In addition, the case of a single equation on one unknown can be solved, and the problems with more than two unknowns can also be solved iteratively, using the program's temporary solution storage capability.

4.4 METHOD OF SOLUTION

We will consider the case of the elliptic problem and describe the techniques which are used by TWODEPEP, since the methods used to solve time-dependent, eigenvalue and non-symmetric problems are extensions of the same techniques used for the elliptic problems.

We seek a solution to the elliptic problems which can be put in the form that permits a solution to be found by minimizing the equivalent integral in the arbitrary two dimensional region R

$$J(\underline{u}) = \iint_R E_1(\underline{u},\underline{u}_x,\underline{u}_y)\, dxdy + \int_{\partial R_2} E_2(\underline{u})ds \ , \qquad (4.4)$$

where $\underline{u}$ is a vector function satisfying the boundary condition $\underline{u}=\underline{FB} \in \partial R_1$. Then the problem becomes one of finding a particular $\underline{u}$ which minimizes the integral (4.4).

Then, we must have, for any $\underline{\phi}$ satisfying $\underline{\phi}=\underline{O}$, on ∂R_1,

$$\frac{\partial J}{\partial \delta}\Big|_{\delta=O}(\underline{u}+\delta\underline{\phi}) = \sum_i^m \left\{ \iint_R \left[\frac{\partial E_1}{\partial u_{ix}}\phi_{ix} + \frac{\partial E_1}{\partial u_{iy}}\phi_{iy} + \frac{\partial E_1}{\partial u_i}\phi_i \right] dxdy + \int_{\partial R_2} \frac{\partial E_2}{\partial u_i}\phi_i \, ds \right\} = O \ , \qquad (4.5)$$

since the problems solved by TWODEPEP are in general two dimensional problems, and the vector $\underline{u}$ is normally a one or two component vector. Thus m=1 or 2.

Let

$$\frac{\partial E_1}{\partial u_{ix}} = \sigma_{(ix)}$$

$$\frac{\partial E_1}{\partial u_{iy}} = \sigma_{(iy)}$$

$$\frac{\partial E_1}{\partial u_i} = F_1$$

and

$$\frac{\partial E_2}{\partial u_i} = G_i B_i \tag{4.6}$$

Now equation (4.5) can be written in the following form,

$$\sum_{i=1}^{m} \left\{ \int\int_R [\sigma_{ix}\phi_{(ix)} + \sigma_{iy}\phi_{(iy)} + F_i\phi_i]dxdy + \int_{\partial R_2} GB_i\phi_i ds \right\} = 0 \tag{4.7}$$

Applying Green's theorem to equation (4.7) leads to,

$$\sum_{i=1}^{m} \left\{ \iint_R [\sigma_{((ix)x}\phi_i + \sigma_{(iy)y}\phi_i + F_i\phi_i]dxdy + \int_{\partial R_2} (GB_1 - \sigma_{(ix)}\ell_x - \sigma_{iy}\ell_y)\phi_i \right\} ds = 0 \ . \tag{4.8}$$

This leads to the general form of the elliptic equations,

$$\frac{\partial}{\partial x}\sigma_{(ix)}(\underline{u},\underline{u}_x,\underline{u}_y) + \frac{\partial}{\partial y}\sigma_{(iy)}(\underline{u},\underline{u}_x,\underline{u}_y) + F_i(\underline{u},\underline{u}_x,\underline{u}_y) = 0 \tag{4.9}$$

in R, and,

$$\underline{u}_i = FB_1 \quad \text{on } \partial R_1$$

$$\sigma_{(ix)}\ell_x + \sigma_{(iy)}\ell_y = GB_i \text{ on } \partial R_2 \ .$$

The finite element method minimizes the integral (4.1) over a class of piecewise polynomials. The idea is to choose a finite number of trial functions $\phi_1,\phi_2,\ldots,\phi_N$, and among all their linear combinations $\sum_1 a_i\phi_i$ to find the one which is the minimum, the unknown a_i are determined not by the differential equation, but by a system of N discrete algebraic equations which the computer can handle. Therefore, the goal is to choose trial functions ϕ_i which are convenient enough for the given integral (4.1) to be computed and minimized, and at the same time general enough to approximate closely the unknown $\underline{u}$.

TWODEPEP starts by a subdivision of the given region into smaller pieces which are triangles with standard six-node with a quadratic basis function, and with one edge curved when adjacent to a curved boundary according to the isoparametric method. It is also optional to use 10-point cubic (3rd degree) or the 15-points quartic (4th degree) isoparametric triangular elements for greater accuracy.

Each time a triangle is partitioned, it is divided by a line from the midpoint of its longest side to the opposite vertex. If this side is not on the boundary, the triangle which shares that side must also be divided to avoid non-conforming elements with discontinuous basis functions.

An initial triangulation with sufficient triangles to define the region is supplied by the user, then the refinement and grading of this triangulation is guided by a user supplied function D3EST which should be largest where the final triangulation is to be most dense. The Cuthill-Mckee algorithm is used to initially number the nodes, and a special bandwidth reduction algorithm is used to decrease the bandwidth of the Jacobian matrix even further.

In all cases the algebraic system is solved by Newton's method. One iteration per time step is done for parabolic problems and one iteration is sufficient for linear elliptic problems. The linear system which must be solved to do an iteration of Newton's method is solved directly by block Gaussian elimination, without row interchanges since pivoting is unnecessary when the matrix is positive definite.

Symmetry is also taken advantage of in the elimination process; if it is present then the storage and computational work are halved. If the Jacobian matrix is too large to keep in core, the frontal method is used to efficiently organize its storage out of core. For time-dependent problems, the right hand side of equation (4.9) is replaced by $c_i u_{it}$, and initial conditions $u_i=u_{iO}$ are given, then the resulting system of equations (4.8), after making the obvious change to account for the extra term, becomes a system of ordinary differential equations, and the unknown coefficients are now functions of t. The implicit or Crank-Nicolson method is used to discretize time steps, and a Richardson extrapolation may also be done to increase the order of convergence in a manner similar to that used to control the mesh grading, a user supplied function of t controls the time step variation. The Newton iteration is handled in the same manner as for elliptic problems.

The eigenvalue problem obtained by adding $\lambda p_i u_i$ to the left hand side of equation (4.9), is solved for the smallest eigenvalue by the inverse power method.

TWODEPEP can also solve non-symmetric problems of the form (4.9) solving the corresponding non-symmetric system directly by block Gaussian elimination.

TWODEPEP was basically designed for a maximum of two partial differential equations. It is assumed that, in applications, systems of several equations can often be divided into sets similar to (4.9), of one or two unknowns, with strong coupling within each set but weak coupling

between sets. Under this assumption, a system of several equations can be handled by solving the different sets alternately, substituting the latest calculated values for the unknowns corresponding to the other sets.

4.5 SUMMARY OF THE SPECIAL FEATURES OF TWODEPEP

- TWODEPEP is a general purpose finite element program which solves a large class of partial differential equations of the form (4.1).

- TWODEPEP has a preprocessor program which allows the user to write the problem definition in a simple and readable format. Hence, nearly all the Fortran programming involved can be eliminated. The preprocessor also controls the dimension sizes so that only storage necessary for that particular problem is utilized.

- TWODEPEP uses a standard quadratic element, and optimal cubic and quartic isoparametric triangular elements for more higher accuracy.

- Solves up to nine simultaneous equations per set.

- Draws a printer plot of the vertices and centres of triangles in the final triangulation.

- Provides automatic and accurate calculation of the user specified function and/or its derivatives.

- Provides a portable 3-dimensional graphical output program which plots scalar, vector and stress fields.

4.6 INPUT SUMMARIES

The TWODEPEP has a preprocessor Fortran program which reads the user input describing the problem in a format designed to minimize user effort, and then outputs some problem-dependent subprograms which must then be compiled and executed with the problem-independent subprograms. To illustrate the simplicity of the input format, we will list below all information necessary to construct the TWODEPEP input data set for the problem which is similar to the two dimensional elliptic Poisson's problem.

Problem

Solve the two dimensional problem,

$$\frac{\partial}{\partial x}[A(x,y)\frac{\partial u}{\partial x}] + \frac{\partial}{\partial y}[A(x,y)\frac{\partial u}{\partial y}] + B(x,y)u + C(x,y) = 0 \in R,$$

with,

$$\left.\begin{aligned} u &= FB_1(s) \quad \in \partial R_1 \\ \frac{\partial u}{\partial n} &= GB_1(s,u) \in \partial R_2 \end{aligned}\right\} \tag{4.10}$$

The TWODEPEP input for this problem is now given:

The boundary of the region R is divided into distinct arcs, each of which possesses smooth boundary conditions. Thus at every point where the boundary conditions have a discontinuity or corner point, a different boundary condition is defined, a new boundary arc must begin. Each arc is given a distinct identifying integer I, must be negative if u is given on the boundary arc, and must be positive if the normal derivative of u is given. Each curved arc is given by a parameter s, varying from 0 to 1; the orientation of the arc being unimportant.

The user creates an initial triangulation of R with only enough triangles to define the region which has the following properties:

1. Each point where two of the boundary arcs meet is included as a vertex in the triangulation.
2. No vertex of any triangulation touches another in a point which is not a vertex of the second triangle (i.e. the triangulation is "conforming").
3. No triangle may have all three vertices on the boundary.
4. Small angles should be avoided wherever possible.

The input data set consists of three parts, as follows:

A. A single line giving the values of certain variables which must be read in integer format.

B. A group of records defining the values of variables and functions, boundary conditions, initial/triangulation and solution method.

C. Functions which are too complicated to define in part B, a user-supplied Fortran function subprogram may be defined at the end of the input preprocessor.

The first line contains 3 integers NEQ, NTF, NDIM in free format at least one blank between numbers, where

NEQ= number of simultaneous PDE's being solved

NTF= number of triangles desired in the final triangulation

NDIM= storage reserved for the Jacobian matrix. Should be about:

(1) $12 \times 2\sqrt{(NTF)^3}$, if only in-core storage to be used

(2) $20 \times NTF$, if out-of-core storage is to be used.

If NDIM is input as 1 or 2, it will default to the first or the second formula respectively.

Each of the following lines has a function or variable name beginning in column 1. In columns 9-72 the function or variable is defined using Fortran syntax. All of the functions or variables below must be defined or defaulted. Except as expressly noted, the order of the lines is unimportant.

If **** is put in columns 1-4, columns 5-72 may contain comments.

If any function definitions are too long to fit into a single line, Fortran functions may be called in their definition. These functions can be defined after all other input by writing the functions subprograms following a line with ADD in columns 1-4. The last line in the input should have END in columns 1-4.

NAME	MEANING
σ_{xx}	$A(x,y)*U_x$
σ_{xy}	$A(x,y)*U_y$
F1	B(x,y)*U+C(x,y)
D3EST	TWODEPEP tries to distribute D3EST(x,y) ×h(j)**3 evenly over the final triangulation, where h(j) is the diameter of triangle j. The user normally will simply make D3EST largest where he wants the triangulation to be most dense. The triangulation may be plotted to see if it is graded properly.
NX NY XA HX YA HY	The solution will be output at the points of the grid: X = AX + i*HX , i=0,...,NX Y = YA + j*HY , j=0,...,NY

Here XA=minimum value of x in R, etc. If output is desired at an arbitrary sequence of points (XA(M),YA(M),M=1...(NX+1)*(NY+1)), then HX=HY=0 and XA and YA are defined as functions of M.

MWR — Output logical unit number. 6=printer, 8,9=disk files for postprocessing.

PLOT — If PLOT=1, printer plots of the initial triangulation and of the centers of the triangles in the final triangulation will be generated, provided NDIM.GE.500.

Cubics, Quartics — if cubic=1 or quartic=1, cubic or quartic isoparametric elements will be used. They are of higher order accuracy than the default quadratic element.

****
**** BOUNDARY FUNCTIONS

**** For each boundary arc (except those on which all
**** boundary functions are defaulted) there is a line
**** with ARC= in columns 1-4 immediately followed
**** (within the next 12 columns) by the arc number.
**** Immediately following this line the appropriate
**** boundary functions (X,Y,FB1,GB1) for that arc are
**** defined. On any arc the functions FB1, GB1 may be
**** described as functions of X and Y. On curved arcs
**** they may alternatively be described as functions
**** of the arc parameters.

X, Y — line (X(S),Y(S)).(G.LE.S.LE.1) are the parametric equations for arc number I (curved arc only).

FBl	O	FBl(S,X,Y) on arc number I (I negative)
GBl	O	GBl(S,X,Y,U) on boundary arc number I (I positive)

**** Initial Trıangulation Arrays

**** The arrays VXY, IABC, I defining the initıal trı-

**** angulation are defined by free format lısts (at

**** least one comma or blank separating entries). If

**** more than one line is needed, the list can be

**** continued on the immediate followıng lines if the

**** array name ıs repeated on the continuation lınes.

VXY	VX(1),VY(1),VX(2),VY(2),...,VX(NV),VY(NV) where VX(I),VY(I) are the coordınates of vertex number I. The vertıces may be listed in any numbers referred to in IABC.
IABC	IA(1),IB(1),...,IA(NT),IB(NT),IC(NT) where IA(K),IB(K), IC(K) are the numbers (as listed in VXY of the vertices A,B,C of triangle k). A,B,C must be order counter-clockwise and such that C is not on the boundary.
I	I(1),I(2),...,I(NT), where I(K) ıs the ıdentıfyıng integer of the boundary arc cut off by the base, AB, of triangle k. I(k)=O if none.
SYMMETRY	1, for thıs application

Since the followıng two matrıces are symmetric,

$$\begin{array}{cccccc}
\sigma_{xx}/u_x & \sigma_{xx}/u_y & \sigma_{xx}/v_x & \sigma_{xx}/v_y & \sigma_{xx}/u & \sigma_{xx}/v \\
\sigma_{xy}/u_x & \sigma_{xy}/u_y & \sigma_{xy}/v_x & \sigma_{xy}/v_y & \sigma_{xy}/u & \sigma_{xy}/v \\
\sigma_{yx}/u_x & \sigma_{yx}/u_y & \sigma_{yx}/v_x & \sigma_{yx}/v_y & \sigma_{yx}/u & \sigma_{yx}/v \\
\sigma_{yy}/u_x & \sigma_{yy}/u_y & \sigma_{yy}/v_x & \sigma_{yy}/v_y & \sigma_{yy}/u & \sigma_{yy}/v \\
-F_1/u_x & -F_1/u_y & -F_1/v_x & -F_1/v_y & -F_1/u & -F_1/v \\
-F_2/u_x & -F_2/u_y & -F_2/v_x & -F_2/v_y & -F_2/u & -F_2/v
\end{array}$$

and

$$\begin{array}{cc}
GB_1/u & GB_1/v \\
GB_2/u & GB_2/v
\end{array}$$

If the problem is symmetric, the elements above the diagonal in these two matrices need not be defined, and the storage required for the Jacobian will be cut in half. A warning message should be issued if SYMMETRY is set to 1 when the problem is non-symmetric on output, the values of u and $(\sigma_{xx},\sigma_{xy})=(A*U_x,A*U_y)$ will be printed.

4.7 REQUIREMENTS OF THE 2DEPEP PROGRAMMING SYSTEM

The requirements of a general programming system used as a tool by the programmer for special or general purpose finite element analysis programs will of course depend on the type of the problem to be solved and the application of the system.

However, ideally a general programming system for special or general purpose finite element analysis programs should be:

1. Versatile (machine independent)
2. General
3. Capable of handling any reasonable problem size
4. Efficient
5. Reliable
6. Easy to use and maintain
7. Easy to modify and extend (open-ended)

We shall here refer to this list while discussing the experience gained with the programming system (TWODEPEP).

1. VERSATILITY

Although most of the finite element programs are written in standard Fortran IV language, programs developed on one computer system may not be entirely compatible with other systems due to the difference in I/O facilities operating system, precision of the machine, i.e. VAX, PRIME, CDC or many other machines.

2. GENERALITY

This criterion was given a high priority during the development of TWODEPEP. It proved to be a well equipped tool for the advanced programmer for special or general purpose finite element analysis programs during the course of this work.

No restrictions have been found on the number or type of elements. The completely dynamic manner in which the data is stored on peripheral storage also adds to the generality and flexibility of the system.

The TWODEPEP package offers a very large range of applications in linear and non-linear analysis, with effective methods of solution. The programs contain state-of-art finite element procedures together with the implementation of nonlinear models in iteration procedures with accuracy and cost effectiveness.

The programs can be employed effectively in linear analysis, and then, with only a few input changes; several linear elliptic problems which are to be solved on the same triangulation may be solved in one run, and also in many nonlinear analysis.

However, it is very difficult to satisfy all the requirements of generality simultaneously and there are notable limitations in using TWODEPEP, such as,

(i) The partial differential equations solved by TWODEPEP should be restricted to this form,

$$C_1(x,y,u,v,t)\frac{du}{dt} = \frac{\partial}{\partial x}(\sigma_{xx}) + \frac{\partial}{\partial y}(\sigma_{xy}) + F_1$$

$$C_2(x,y,u,v,t)\frac{du}{dt} = \frac{\partial}{\partial x}(\sigma_{yx}) + \frac{\partial}{\partial y}(\sigma_{yy}) + F_2$$

where $\sigma_{xx}, \sigma_{xy}, \sigma_{yx}, \sigma_{yy}$, F_1 and F_2 are functions of $(x,y,u,v,u_x,v_x,u_y,v_y,t)$.

There are many different linear and nonlinear, variable coefficient problems which do not satisfy these above forms.

(ii) Restriction on the boundary conditions. These should be of the form,

$$u = FB_1(s,t) \qquad \in \partial R_1$$

$$v = FB_2(s,t)$$

and

$$\sigma_{xx}\ell_x + \sigma_{xy}\ell_y = GB_1(s,u,v,t)$$

$$\sigma_{yx}\ell_x + \sigma_{yy}\ell_y = GB_2(s,u,v,t) \qquad \in \partial R_2$$

and

$$u = u_o$$

$$v = v_o$$

i.e. TWODEPEP does not solve problems with boundary conditions of different types, all equations must have boundary conditions of the same type on each boundary arc, except for very special cases only. This is indeed a very weak property of TWODEPEP in handling different boundary conditions.

(iii) Round-off error appears to be present in the solution of some problems, which may be diminished with some experience.

3. PROBLEM SIZE

In principle, the programming system TWODEPEP does not impose any limitation on the size of the problem (i.e. number of the unknowns). We can solve any one or two dimensional problem and up to nine equations per set with a maximum of five sets being permitted, the only real limitations with the work on PRIME has proved to be the availability of computing

time and the peripheral storage capacity.

Finally, TWODEPEP has the capability to make a realistic analysis of really large problems.

4. EFFICIENCY

It is difficult to satisfy simultaneously the requirements of both generality and efficiency. Normally, in the case of conflict *generality* has been given the higher priority in the programming systems discussed here. The numerical operations are, however performed efficiently. All key operations are carried out in the Fortran language.

In general, for all types of problems that fit the programming system TWODEPEP format and its boundary conditions, TWODEPEP is very efficient. While estimating efficiency of an application program the cost of man hours is very often neglected. However, in practical applications of the finite element method, this may be decisive for the total cost of the project. Using all the features available in the TWODEPEP programming system, a program can be built to minimize the requirements in man-time for providing input, output data.

5. RELIABILITY

The programming systems (TWODEPEP) consists of a number of well-defined modules (subroutines and functions) each of which has been thoroughly tested, resulting in systems which have proved to be extremely reliable.

The main features of TWODEPEP have been used in application programs,

and the number of program errors which have been found in TWODEPEP over a period of a year is 1.

The detection of an error is always accompanied by a printed message which will help to pinpoint the error. Errors in the hardware or operating system are, of course, not the responsibility of a programming system.

Another aspect of reliability is the numerical precision and the accuracy of the results which may be checked by computing the residuals or error norm. Double precision is also available.

6. MODIFICATIONS AND EXTENSIONS

A general programming system will never, due to its very nature, be complete. New applications may call for modifications and the applicability will depend on the success with which the weaknesses of the system may be improved.

The programming systems (TWODEPEP) are designed to be open-ended, and (up to now) modifications and extensions have proved to be easily accommodated and incorporated.

4.8 GENERALIZED PRE- AND POST-PROCESSORS FOR FINITE ELEMENT PROGRAMS

A crucial factor in all finite element analysis is the large number of input data required and the numerous output results obtained. For nearly all finite element programs in use nowadays a detailed description of the problem to be solved must be fed into the computer in an unfavourable manner to the user which easily promotes errors. Therefore, many pre-processors are designed which allow a short and compact description of the problem to be automatically transformed into the input data. Similarly, a post-processor transforms the output data into graphs, diagrams, tables etc. Therefore, the purpose of a general preprocessor is to:

1. Minimize the amount of input data to be specified by the user
2. Ensure reliability of input data
3. Reduce the total elapsed time for the analysis.

Correspondingly, the post-processing programs should give a simple means to present, interpret and analyse the results. For a typical analysis with an existing program it is reasonable to believe that about 40% of the costs are spent in the model definition and input specification phase, 30% is related to the computer costs for solving the problem, and 30% is used in the presentation and interpretation of analysing the results.

Future trends will lead to steadily decreasing price/performance of computers and increasing man-power costs. It is obviously then good economy to develop tools which reduce the man-power spent on the analysis. It is believed that the input specification task is the most attractive to attack because this is where most of the tedious work time is spent.

The development of efficient pre- and post-processors is not only a matter of good economy, but it compensates for the predicted shortage of development work by increasing the research capability.

(i) PREPROCESSORS

The need for efficient pre-processors including automatic input data generators has been realized from the beginning of the development of finite element programs.

Considerable efforts have been made in developing batch pre-processors which generate all necessary input data from a minimum of input. Input devices for the transfer of previously calculated data (from other programs) are also available. Automatic checking of input data, print and plot of generated data (e.g. geometry and element mesh). For huge and complex problems, it is necessary to have batch and interactive specifications, probably the most efficient use of interactive graphic pre-processors is for the editing of data. In designing such a pre-processor the following requirements are essential and will guide the development of general interactive and batch preprocessors.

1. The preprocessors must be easy to learn and use
2. It must offer possibilities of control
3. The preprocessors should be able to work both in batch mode and in interactive mode handling input data from the keyboard and graphic input devices. The selection of a mode should be controlled by a special command in the input system.
4. Backup generation. If a fatal error is committed during the operation of the program, the information generated up to a

certain stage should be available for a capability.

5. The preprocessor should also contain effective 3 dimensional geometry generators.
6. The interactive routines communicating with the user should supply the user with sufficient instructions on request.

One of the main points in the design of finite element programs is to have a standardization of the data problem between the preprocessor and the analysis programs so that the preprocessor can be used for different types of analysis and even be linked to different finite element programs, by this procedure we are aiming at a standardization of the input data to many commonly used finite element programs.

This is very attractive and important, because the user need only be familiar with one input system from which he can have access to different analysis programs. This is perhaps the most difficult requirement to satisfy. However the idea has been brought forth by finite element software developers worldwide.

As an example of a preprocessor which has been linked to different analysis programs is FEMGEN, however it seems that none of the available systems offer sufficient generality.

We can define now an *ideal preprocessor* as one that allows the user to generate the necessary information with the least effort for as wide a range of problems as possible. The term "user friendly" has been used to describe a preprocessor that can be operated with relative ease.

(ii) POSTPROCESSORS

The aim of the postprocessors described here is to provide users of finite element programs with tools for selection and presentation of analysis and results (velocity, displacements, etc.) in the form of printed tables, and drawings, interactive graphics, etc.
It may be suitable to distinguish between:

1. General postprocessors, i.e. programs which are applicable for many types of problems and for different applications.
2. Application dependent postprocessors, i.e. programs which are unique to a specific problem or specific research.

The general postprocessors should have the properties of:

1. Presentation of the computed quantities or field variables in the form of,
    (a) Diagrams, isoplots, etc.
    (b) Selected printout, e.g. velocity above a given level, displacements at certain nodes, etc.
    (c) Scaling and combination of analysis from different cases and alternative analysis

According to the above requirements, there are many programs which perform, print and plot an analysis of the results (as an example PRE for the TWODEPEP system and NV340 is a general postprocessor to SESAM.69).

In the next generation of general postprocessors it will be possible to select and present analysis results also from interactive graphic terminals. This gives an efficient means of scanning through the analysis results before scaling data for permanent print and plots. Thus, it will

be possible for the researchers to directly access and present analysis results, and hence the corresponding data base may serve as an easily accessible permanent data storage.

Normally it will be advantageous to perform pre- and post-processing on minicomputers, and hence the easy transfer of data between different computers should be provided. Requirement of the postprocessors may be application dependent and may also be unique to specific projects. For this reason special purpose post-processors are frequently developed either separately or by modification of the general programs.

In order to facilitate the development of special post-processors, a thorough documentation of the data analysis and special programs to handle transfer of this data are required.

An interesting problem arises in displaying results from non-linear or time dependent problems which have variations with respect to a given parameter (time, etc.). The easiest and most widely used method is to present the results by a series of separate or "frozen" pictures or graphs corresponding to the step in the solution process. An alternative procedure is to present the results by movies obtained by animation of the results computed at different instants of time. This method which has been demonstrated by CHRISTIANSEN [1981] is very instructive. It will be neither desirable nor possible for one designer to develop all the software of pre- and post-processors that are needed in an institution.

For this reason co-operation with other institutions and companies is needed in order to share the costs of development and try to implement existing software into the system.

CHAPTER FIVE

THE FINITE ELEMENT METHOD FOR FREE SURFACE PROBLEMS

5.1 INTRODUCTION

The application of the finite element method to the solution of some partial differential equations in a region characterised by flows having a boundary which is not known (free surface) *a priori* has grown very rapidly and become an important area for many researchers and scientists.

Wakes constitute an example of such problems. These phenomena are produced in reality by placing an obstacle such as a plate in a moving stream so that the flow separates from the obstacle along the separating streamlines, the fluid between these streamlines constitutes the wake. In high speed motion of a liquid, the wake may become gaseous and thus form a cavity. Jets offer another example in which a free surface is present, a jet may be of water in air, water in water, etc. Porous media flows from another category of physical problems in which there is a free surface, seepage under or through dams, moisture flow through saturated or partially saturated soils and flows to and from drains, ditches or wells.

Conductive heat transfer with change of phase, evaporation of liquid from porous media or precipitation of products in chemical solutions give rise to a class of unsteady free surface (interface) problems.

Open channel flows offer a rich source of real examples of steady-free surface problems with a strong nonlinearity and complicated singularities.

A large number of different flow situations can be considered in open channels. Typical examples are:

flows under a sluice gate, flows over a weir, flows over a spillway, flows over a step and other bed configuration. Many effects on the free surface of the flow such as surface tension and gravity give rise to different approximations to the real problem. The governing differential equations represents an approximation of the phenomena of interest.

In the case of open channel flows it is reasonable to assume that the effects of gravity are predominant over the effects of surface tension. Moreover, the flow may be assumed to be inviscid, incompressible and irrotational. Such approximations represent effects of non-uniqueness and limiting cases of steady flows and standing waves.

Analytical treatment of the governing differential equations is possible in some situations but at the expense of further simplification. The hodograph transformation (see OCKENDON and TAYLER [1979]), is a good technique that can be used when dealing with two-dimensional potential flows with a free surface. An illustration of this analytical approach is provided by BENJAMIN[1956] who computed the flow under a sluice gate.

The numerical approach has become more widely used technique for solving free surface problems in general, one of the advantages over analytical methods is that it can be applied to more general physical problems.

One of the first reported successful attempts to solve some open channel problems numerically is due to SOUTHWELL and VAISEY [1946], where they used the finite difference relaxation technique to solve

problems of flow under a sluice gate, jets, stationary waves of finite amplitude, flows under a planing surface and wakes. The great merit of the work by Southwell and Vaisey is that they treat the full potential problem with free surface. However, the technique used is subject to problems of accuracy for curvilinear field geometries which is precisely an important feature of free surface problems.

In any numerical approach for the analysis of the free surface problems the exact position of the free surface is not known *a priori* and its location forms part of the analysis. We note here the difference between *free* and *moving* boundary problems: a free surface problem is a part of a steady state problem and does not in fact move at all. Generally, extra conditions are specified on a free boundary and this enables its free position to be located, a moving boundary problem is generally a time-dependent problem and an essential feature of these problems is the presence of a sharp boundary surface that moves through the medium, the mathematical formulation of this problem arises in the study of heat flow in a medium that undergoes a phase change.

As a result the finite element method has become a very popular numerical technique in fluid mechanics.

A description of the method is provided in many text books such as ZIENKIEWICZ [1971], and MITCHELL and WAIT [1977]. When the flow is known then the finite element method is directly and easily applicable to solve potential flow problems. However, a major problem is posed when the flow has a free surface.

The variational principles in the finite element method has become of great importance MITCHELL [1972], their principles governing a variety of free surface flows are presented by many researchers; O'CARROLL [1978] who discusses the problem of choosing the appropriate functional associated with the stream function and velocity potential. The fundamental features of the introduced variational principle is that they govern both the internal flow and the free surface position problems.

The method for locating the free surface positions is acknowledged to be the major difficulty in these free surface problems and we list below some of the difficulties which arise in solving free surface problems:

1. a varying domain - where the position of the free surface is not known a priori.
2. the occurrance of non-linear boundary conditions, and
3. a central region where the critical depth is not known.

The prediction of the position of the free surface can be carried out numerically, which was first done in finite difference by SOUTHWELL and VAISEY [1946]. If the finite elements are used to model the flow, three main approaches can be used:

1) To extend the finite element mesh from the bed to the free surface flow, and as iterations are performed, to move the mesh to follow the free surface and satisfy the total energy criteria.
2) To fix the element mesh and to vary the element properties, so as to model the position of the free surface. This method has

only been used for seepage and other similar flows, in which the kinetic energy of the flow is small.

c) To invert the problem using co-ordinates as the dependent variables and using the stream function and velocity potential as independent variables, this method has only been applied using relaxation techniques, not finite elements. Although it appears to be very promising, this method was first suggested in the context of free surface problems by MARKLAND [1965] who applied it to the free flow over an overfall, using a relaxation technique. It was subsequently applied to large amplitude waves by WILLIAMS [1974].

Another important method is that deviced by VAROGLU and FINN [1978], which is a semi-inverse method and thus falls between methods 1 and 3 above. Method 1 and the Varoglu and Finn method have only been applied to date using simple linear triangle elements. Our method is based on strategy 1 and by using quadratic and quartic triangular elements with a dense area of elements near the free surface flow. We will discuss in detail the problem later in this chapter.

A typical problem involving the percolation of a fluid through porous material is illustrated in Figure (5.1), typically an earth or sand construction in which part of the porous medium is wet and the remainder is dry and we have to calculate the position of the dividing line between the wet and dry (the free surface).

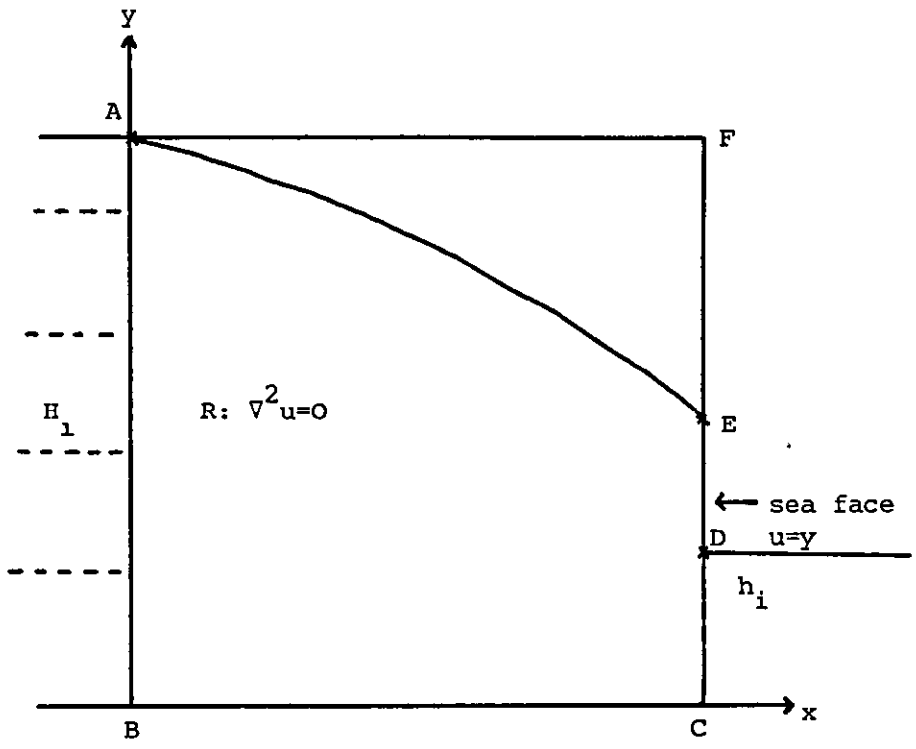


FIGURE 5.1: Water Seepage through an earth dam

Normally the problem involving Laplace's equation

$$\nabla^2 u = 0 \quad \text{in } R,$$

as well as the following boundary conditions:

on AB $u = H_i$

BC $\frac{\partial u}{\partial y} = 0$

CD $u = h_i$

DE $u = y$

and AE free surface $\frac{\partial u}{\partial n} = 0$.

One problem of this type has become a standard test problem in the field of free surface solutions. It is usually known as the classical dam problem and is illustrated in Figure (5.1).

This problem assumed a dramatic new importance when it was shown by Biaocchi [1972] that the region of solution can be extended to the complete rectangle ABCF, i.e. we solve a modified problem that does not involve the position of the free surface explicitly and then locate the free boundary from this extended solution. The original purpose of Biaocchi's work was to provide a proof of the existence and uniqueness of the solution to the original mathematical problem. However, such a formulation is very convenient for a numerical solution and has been shown to be successful on this limited standard problem (see AITCHISON [1977]). In general, we have the free surface problem shown in Figure (5.2), where the differential equation,

$$Du = 0 \text{ in the region } R \ , \tag{5.1}$$

and subject to the condition,

$$Lu = 0 \text{ on } \partial R_1 \text{, the boundary of } R, \tag{5.2}$$

and the free boundary ∂R in ∂R_1:

$$Cu = 0 \ , \tag{5.3}$$

where D,L,C are a set of differential operators,

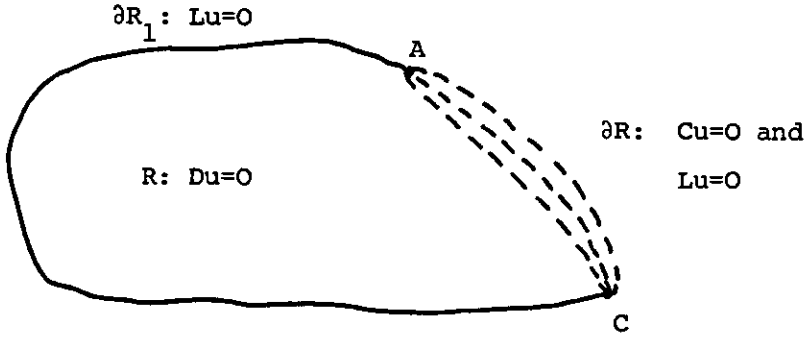


FIGURE 5.2: The general free boundary problem

The problem is to determine the shape and location of the free boundary AC.

The trial free boundary method which involves the solution of a sequence of problems with different fixed boundaries is applicable to all free boundary surface problems and requires no preliminary analysis, although some analysis may often be desirable to make a good initial guess at the position of the free boundary.

Some conclusions can be drawn in regard to using the trial free problem by the finite element technique. These include:

- if the differential operator D is linear in equation (5.1), then the computational effort is to solve only a linear set of equations, while if D is not linear, then it leads to a nonlinear system of equations, which can be solved iteratively by using one of the methods discussed in Chapter 2.
- general or special purpose packages can often be used to solve the differential equations, where special techniques are not needed.
- with the very rapid growth of the finite element method, it is easier to implement the method than it was when finite differences were the only common method of solution.
- a characteristic of free boundary problems is that they generally require the differential equation to be solved on a region with normally a curved free surface. This irregular shaped boundary can be approximated by using elements with straight sides or matched exactly using element curved boundaries. However the finite element method is not limited to regular shapes with easily defined boundaries, whereas programming this with the finite differences may be more difficult.

- the successful application of the trial free boundary method, over a large number of free surface problems has perhaps tended to discourage work on other free boundary methods, and this in turn means that there is a limited work on other methods, with which to compare the results which we obtain.
- despite its many applications, there is remarkably little theoretical understanding of why it works and how it is affected by different boundary conditions and different problems.

There are, of course, certain disadvantages in the use of the trial free boundary method, such as:

- The solution processes involve the computations of a sequence of solutions $\{u^{(k)}\}$, k=1,2,..., for different fixed boundaries which requires a large amount of computer time and storage.
- It seems to be that there is no fixed rules which ensure convergence since generally speaking different techniques are needed for different problems. Often it is not clear which of the available conditions to use for solving the differential equation (i.e. Lu=0, or Cu=0).
- It is difficult to obtain high accuracy, and hard to estimate the error in $u^{(k)}$ and on $\partial R^{(k)}$, particularly since at each stage we only find an approximation $u^{(k)}$ to u. In some problems the shape of the free boundary is very sensitive to small errors in the condition Cu=0, and so it is difficult to achieve high accuracy near points of separation.

Alternative methods for solving free boundary problems have been devised such as the third approach described at the beginning of this

chapter. Aitchison, Baiochi used a method which avoided the outer iteration to find the position of the free boundary. The problem is then reformulated as a quadratic minimization problem on a fixed region, and it works well, but it is only used on porous flow problems, which are not easily applicable to more general problems.

Viscous flow problems are particularly difficult to solve, and for this case only the trial free boundary method is available at present.

5.2 FINITE ELEMENT SOLUTION OF SLUICE GATE FLOW

The first to consider the influence of gravity on flows under a sluice gate appears to have been PAJER [1937]; he assumed that, while a circle in the hodograph plane corresponded to the limiting case of zero gravity, an ellipse could be used to replace the circle when gravity is present. This fixed the shape of the free streamline, but the boundary condition of constant pressure was not verified. The resulting streamline was correct at the end points, and nearly correct at intermediate points.

An improvement of Pajer's method was devised by BENJAMIN [1956]. He assumed the shape of the hodograph as the arc of an ellipse to a region on the streamline where the solution essentially matched that of a solitary wave based on the downstream Froude number. The boundary condition of constant pressure was also not verified for this result either.

Infinite series were used by PERRY [1957] for improving the hodograph method to include gravity. An inverse hodograph of arbitrary shape was mapped onto a circle. By increasing the number of terms in the series, a mapping of the free streamline was made to satisfy the constant pressure condition at an increasing number of points. Changing the number of terms in the series changes the values of the constants, and the method was dropped in favour of one treating flows with gravity as a perturbation of the flow without gravity. The resulting shape of the free streamline in the hodograph plane was essentially that of a shifted circle.

The contraction coefficient was found to be theoretically related to the total head, H, and the gate opening, b, by,

$$C_c = 0.6110 - \frac{0.0170}{\frac{H}{b} - 1}, \qquad (5.4)$$

All the preceding solutions assume an infinite reservoir with no free-surface upstream from the gate. The effect of the upstream free surface was included by Southwell and Vaisey. They solved Laplace's equation for a sluice gate by substituting a finite difference equation for the partial differential equation and applying relaxation procedures. For a gate opening to total head ratio of approximately 0.53, the only configuration reported, the downstream depth was 0.608 of the gate opening.

T.S. Strikoff proposed a general method for solving gravity flows and applied it to the sharp-crested weir. An integral equation resulted, which was then solved by a numerical, iterative procedure. The method is adaptable to other rapidly varied open-channel flows in which the boundaries are horizontal and vertical. The formulation of a boundary-value problem for the sluice gate is based on this method. J.A. MCCORQUODALE [1971], presents a finite element procedure for computing the hydraulic characteristics of sluice gates with two-dimensional irrotational gravity flow.

PROBLEM FORMULATION

This section presents a numerical procedure for computing the hydraulic characteristics of a sluice gate with two dimensional irrotational gravity flow. The procedure for solving free surface potential flow problems involves solving Laplace's equation as usual plus satisfying the conditions that the velocity normal to the free surface be zero and the pressure along the free surface be constant.

The method used to locate the free surface is to select a trial free surface shape, then solve the Laplace equation and calculate the velocity components along the assumed free surface profile.

From the solution obtained the pressure condition is checked at each surface node by means of the equation,

$$\frac{1}{2g}\left[\left(\frac{\partial\psi}{\partial x}\right)^2 + \left(\frac{\partial\psi}{\partial y}\right)^2\right] + y = E \ ,$$

which is then used to correct the surface position in order to obtain a new domain for solving the problem, until the prescribed error criterion is satisfied.

Consider the analysis of a sluice gate flow performed as follows. The formulation is in terms of the stream function ψ, velocities u and v can be obtained from the stream function ψ by

$$u = -\frac{\partial\psi}{\partial y} \ , \quad v = \frac{\partial\psi}{\partial x} \ . \tag{5.5}$$

The pressure energy equation is given by,

$$p = E - \frac{1}{2g}\left[\left(\frac{\partial\psi}{\partial x}\right)^2 + \left(\frac{\partial\psi}{\partial y}\right)^2\right] - y \ , \tag{5.6}$$

where E is the total energy, and y is equal to the potential energy.

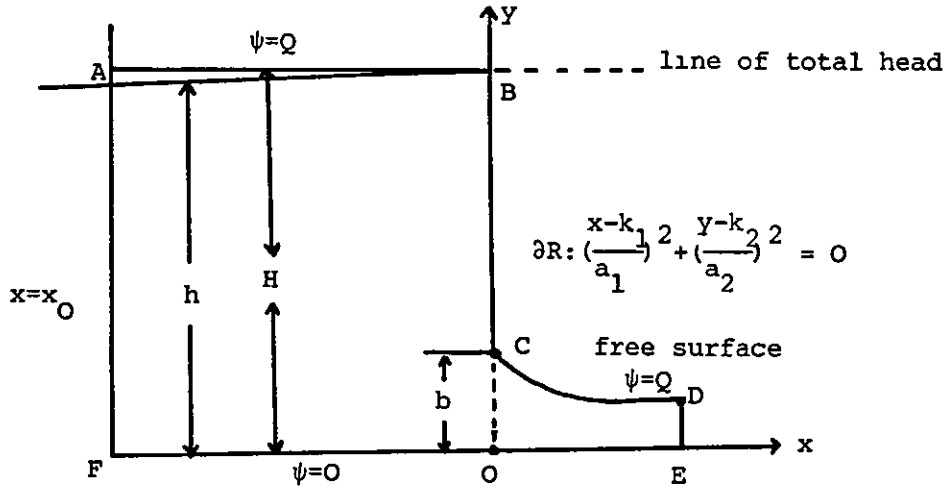


FIGURE 5.3

and thus under the stated physical assumption yields the Laplace equation,

$$\nabla^2\psi = 0 \text{ in } R \ , \tag{5.7}$$

as well as the following boundary conditions,

$$\frac{\partial\psi}{\partial x} = 0 \text{ for } x=x_0 \text{ and } x=h \ , \tag{5.8}$$

and,

$$\frac{1}{2g}\left[\left(\frac{\partial\psi}{\partial x}\right)^2 + \left(\frac{\partial\psi}{\partial y}\right)^2\right] + y = E, \text{ for } y=h \ . \tag{5.9}$$

In addition there are the imposed boundary condition,

$$\psi = Q \text{ for } y=h \ , \tag{5.10}$$

$$\psi = 0 \text{ for } y=0 \ . \tag{5.11}$$

The function h(x) is unknown but it must be located so that the boundary conditions (5.9) and (5.10) are satisfied.

The upstream (subcritical) portion of the free surface AB (as given in Figure (5.3) can initially be taken $h \cong E$, since the velocity head is very small.

Later h can be corrected for velocity head.

The downstream (supercritical) portion h must be treated more carefully as follows. An elliptical curve,

$$\left(\frac{x-k_1}{a_1}\right)^2 + \left(\frac{y-k_2}{a_2}\right)^2 = 0 \ , \tag{5.12}$$

was selected to describe the outflow free surface since this function can be made to satisfy the tangency condition at the gate lip by setting a_1 and a_2 (see Figure 5.3).

5.3 MOVING STRATEGY

From the description of the finite element method in Chapter 3, and its application we will now look at the problem of locating the position of the free surface, a problem described in Section (5.2). If the position of the free boundary CD (which we denote by ∂R) were known then (5.7), and the boundary conditions (5.8),...,(5.11) would suffice to solve the problem for ψ, but since ∂R is not known a priori, then moving the free boundary $\partial R^{(k)}$ to $\partial R^{(k+i)}$, i=1,2,..., often turns out to be the most difficult aspect of the trial free boundary surface. Mostly, authors simply say, the trial free surface was adjusted until both the given boundary conditions were satisfied, and give no further details.

The prediction of the position of the free surface in the early days was first done by hand, such as Southwell and Vaisey [1946], and they did not use any specific rules to move the free surface. Basically it is desirable to have a given scheme for moving the boundary so that this can be done automatically.

To implement any moving strategy on a computer it is convenient to regard the boundaries $\partial R^{(k)}$ as being defined by a number of parameters $a_1^{(k)}, a_2^{(k)}, \ldots, a_n^{(k)}$, say these define, for instances, the vertices of a curved boundary giving the curve the formula,

$$\partial R^{(k)} = \partial R^{(k)}(a_1^{(k)}, a_2^{(k)}, \ldots, a_n^{(k)}) \ . \qquad (5.13)$$

Here we will discuss now the methods of moving the boundary, which may be divided into three categories, *local*, *integral* and *global*.

To describe these methods we consider the free surface part of the free surface problem, which is illustrated in Figure (5.4), and in terms of the velocity potential $u \equiv \phi$, with,

$$Du = \nabla^2 u = 0 \quad , \text{in } R$$

$$Lu = 0 \quad , \text{on } \partial R$$

$$Cu = 0 \quad , \text{on } \partial R$$

together with appropriate boundary conditions on the fixed boundary.

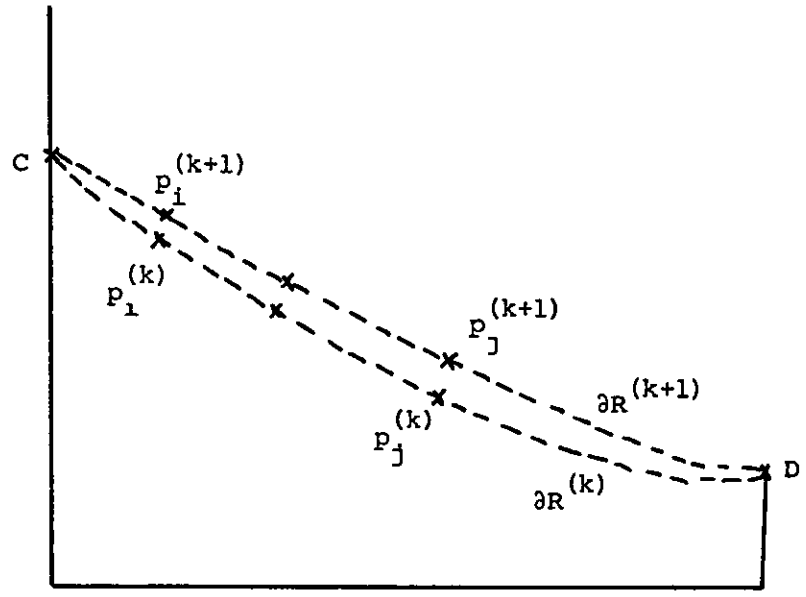


FIGURE 5.4: Movement strategy

5.3.1 LOCAL MOVEMENT STRATEGY

From Figure (5.4) above, in this strategy, the adjustments to $\partial R^{(k)}$ are made at individual points on the basis of the error in $Cu^{(k)}$ at these points will be minimum. Thus, $Cu^{(k)}$ is computed at m points,

$$p_j^{(k)} \in \partial R^{(k)} \quad , \quad 1 \leq j \leq m,$$

If $Cu^{(k)}$ is not zero at the point $p_j^{(k)}$, then another point in the neighbourhood of $p_j^{(k)}$ is found $p_j^{(k+1)}$ where the boundary condition $Cu=0$ is satisfied better.

$\partial R^{(k+1)}$ is then drawn through the points $p_j^{(k+1)}$, $(j=1,2,\ldots,m)$.

Convergence of this strategy, is of course, not guaranteed, nor easily obtained for some problems. Sometimes it has been found that an error over one part of the boundary can only be reduced by moving a different part of the boundary.

With the high-speed computer it has become desirable to automate the movement of points along the boundary curve.

The approach usually is to determine the points $p_j^{(k+1)}$ according to the condition that,

$$Cu^{(k)}(p_j^{(k+1)}) = 0 \ , \tag{5.14}$$

together with the condition that,

$$p_j^{(k+1)} = p_j^{(k)} + \alpha_j^{(k)} \underline{s}_j^{(k)} \ , \tag{5.15}$$

where $\underline{s}_j^{(k)}$ is a specified direction vector, and $\alpha_j^{(k)}$ a constant to be determined which minimizes the error. The different methods correspond to different ways of choosing $\underline{s}^{(k)}$ in (5.15), while possible choices for $\underline{s}_j^{(k)}$ are: the unit outward normal to $\partial R^{(k)}$ at $p_j^{(k)}$.

Thus, some form of line search is performed along the direction $\underline{s}_j^{(k)}$. The computation of $p_j^{(k+1)}$ so as to satisfy (5.14), depends of course upon the structure of C.

The most obvious approach based on Newton's method is to define a function $f_j^{(k)}(\alpha)$ by

$$f_j^{(k)}(\alpha) = Cu^{(k)}(p_j^{(k+1)}) = 0$$

$$= Cu^{(k)}(p_j^{(k)} + \alpha s_j^{(k)}) \ , \tag{5.16}$$

and then find a root of $f_k^{(k)}(\alpha)$. Then compute or estimate,

$$\frac{d}{d\alpha} f_j^{(k)}(\alpha)\Big|_{\alpha=0} \ , \tag{5.17}$$

and then set,

$$\alpha_j^{(k)} = -f_j^{(k)}(0) / \frac{d}{d\alpha} f_j^{(k)}(\alpha)\Big|_{\alpha=0} \ . \tag{5.18}$$

However, various other approaches have been used, [CRYER (1976)]. In a number of instances it has been found desirable to smooth the points $p_j^{(k+1)}$ so as to prevent undesirable oscillations.

FINNEMORE and PERRY [1968] used a standard smoothing subroutine. CRYER [1970] determined m points $p_j^{(k+1)}$, $1 \leq j \leq m$, and then fitted a curve $\partial R(\underline{a})$ of prescribed form through these points to obtain $\partial R^{(k+1)} = \partial R^{(k+1)}(\underline{a})$.

In summary then a local strategy has been used very successfully despite their apparent arbitrariness.

5.3.2 MOVEMENT STRATEGY: INTEGRAL APPROACH

In an integral strategy of moving the free boundary condition:

$$Cu = 0 \ ,$$

can be expressed in an implicit form such as:

$$G(u(s), u_x(s), u_y(s), \underline{x}(s), \dot{\underline{x}}(s)) = 0 \ , \tag{5.19}$$

where $\underline{x}(s)$ is the free boundary curve, and $\dot{\underline{x}}(s)$ denotes the derivatives of $\underline{x}(s)$ with respect to the arc-length s.

Given $\partial R^{(k)}$ and $u^{(k)}$, the curve $\partial R^{(k+1)}$ is obtained by integrating the differential equation for $\underline{x}(s)$, in an approximate form,

$$G(\hat{u}^{(k)}(s), \hat{u}_x^{(k)}(s), \hat{u}_y^{(k)}(s), \underline{x}^{(k+1)}(s), \dot{\underline{x}}^{(k+1)}(s)) = 0 , \quad (5.20)$$

where $\hat{u}^{(k)}(s)$, etc. represents approximations to the value of $u^{(k)}(s)$ etc. at the point $\underline{x}^{(k+1)}(s)$ and may be obtained by interpolation or extrapolation. Often, $u^{(k)}(s)$ is taken to be the value of $u^{(k)}$ at a gridpoint nearest to $\underline{x}^{(k+1)}(s)$. Many applications of integral methods have been made. NICHELL and CASWELL [1974] compute the flow of a viscous jet extruded from a tube. Most of the applications show, the integral method has proved itself of great value, and it seems clear that many more applications will be found.

5.3.3 MOVEMENT STRATEGY: GLOBAL APPROACH

In a global method of moving the free boundary $\partial R^{(k)}$, a set of perturbed boundaries,

$$\partial R^{(k+j)} = \partial R[a_1^{(k)}, a_2^{(k)}, \ldots, a_{j-1}^{(k)}, a_j^{(k)}, \delta a_j^{(k)}, a_{j+1}^{(k)}, \ldots, a_n^{(k)}] ,$$

$$\text{for } j=1,2,\ldots,n , \quad (5.21)$$

and corresponding solutions $\hat{u}^{(k,j)}$ are generated.

Thus, this information makes it possible to estimate the dependence of the error $Cu^{(k)}$ on the parameters $\underline{a}^{(k)}$.

The new position of the free boundary $\partial R^{(k+1)}$ is chosen to minimize the error $Cu^{(k+1)}$ in some sense.

In order to minimize the error $Cu^{(k+1)}$ we must have a measure for this error, $E(\underline{a})$, say. Some choices for E have been used:

(i) $$E_1(a^{(k)}) = [\sum_{j=1}^{n} (Cu^{(k)}(p_j^{(k)}))^2]^{\frac{1}{2}}, \qquad (5.22)$$

that is, $E_1(\underline{a}^{(k)})$ is the m-vector of the errors at n points $p_j^{(k)}$, where E_1 is the least square error.

Examples of the use of this method are:

(1) SANKAR [1967, p.153] and FOX and SANKAR [1973] solves the flow in an axially symmetric Riabouchinaky cavity.

(2) MCCORQUODALE and LI [1971] consider the problem of sluice gate flow.

5.4 NUMERICAL RESULTS

From a description of the general trial free boundary methods, we will now apply the global strategy to the free surface problem presented in Section (5.2). For the trial global strategy, the elliptic curve which is given in equation(5.12) was selected to describe the outflow free surface CD, which can be made to satisfy condition (5.10), by setting the constants a_1 and a_2, the streamline ABCD is assigned a stream function $\psi=Q_o$, then an approximate solution for the internal flow field can be found from solving the finite element equations (5.7)...(511).Conditions along the free surface $\partial R^{(k)}$, (k=0,1,...,m) can be checked. If an error is found between the given energy E, and the computed energy E_c along CD the assumed a_1 and a_2 should then be adjusted so as to minimize this error,

$$s = \sum_{i=1}^{m} (E_i - E_{ci})^2 \quad , \quad \text{where m is the number of mesh points along CD.} \qquad (5.23)$$

We computed s, the sum of squares of the error for the points along CD, for several values of a_1 and a_2. A search is then made for a_1 and a_2 which gives a minimum value of s.

The flow geometries, b/H=0.4 and 0.36 were investigated, and a typical surface profile is shown in Figure (5.5) with both downstream profiles, shown in Figure (5.6).

Quadratic triangular elements were used to model the problem and the geometry of the top layer of the free surface elements was allowed to change, i.e. the constants a_1 and a_2 were varied above and below the first guess, in general if the guess is close to the correct values, the

algorithm will converge to the correct solution with only a few iterations. Only a few of the many results have been plotted so as to avoid confusion.

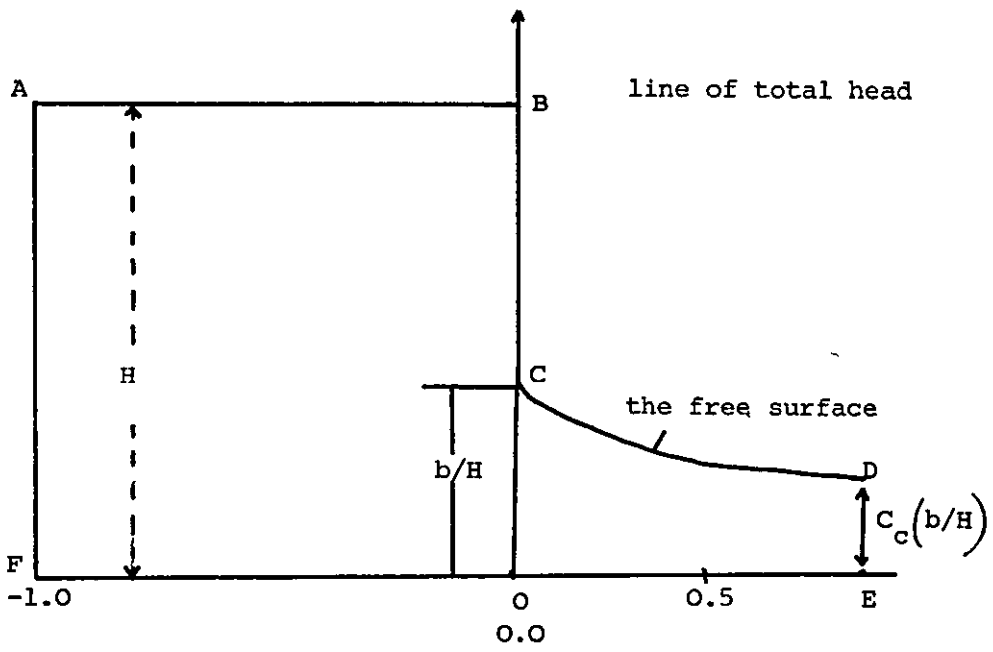


FIGURE 5.5: Typical flow profile

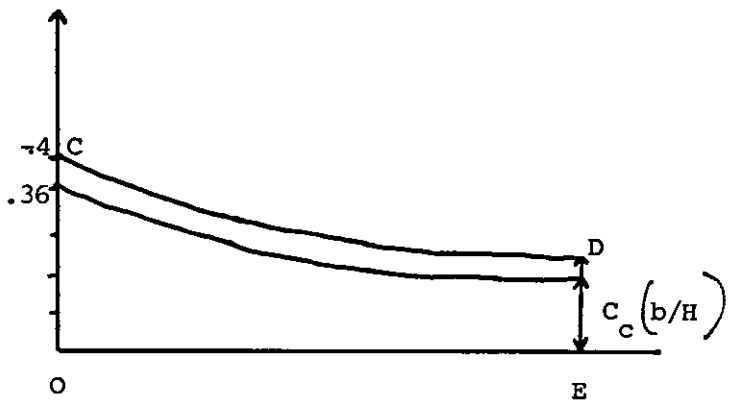


FIGURE 5.6: Downstream surface profile

The computed constants a_1, a_2 and the ratio $C_c b/H$ for the flow geometries b/H = .4 and .36 are given below.

b/H	a_1	a_2	C_c	$C_c b/H$
0.4	1.0	0.16016	0.5996	0.23984
0.36	1.0	0.14335	0.6079	0.21665

where, b is the gate opening,

H total head on the gate,

C_c contraction coefficient.

A plot of the potential flow solution in the given region with b/H=0.4 is shown in Figure (5.7).

Southwell and Vaisey determined the gate opening for a given discharge, but the region near the gate has a large curvature, making an accurate determination too difficult because there the contraction coefficient C_c, and the profile is too large.

Experimental C_c values reported by BENJAMIN [1956] are much larger than theoretical values, he explained that different contraction coefficients for the two gate openings indicate that variables other than the geometric ratio b/H have a significant influence on the flow. He clearly shows that apparently a major discrepancy which appears in

in the plot of C_c versus b/H is due to the presence of a boundary layer on the channel bottom in the real flow downstream from the gate, but when a proper allowance is made for the boundary layer, theory and experiment agree satisfactorily.

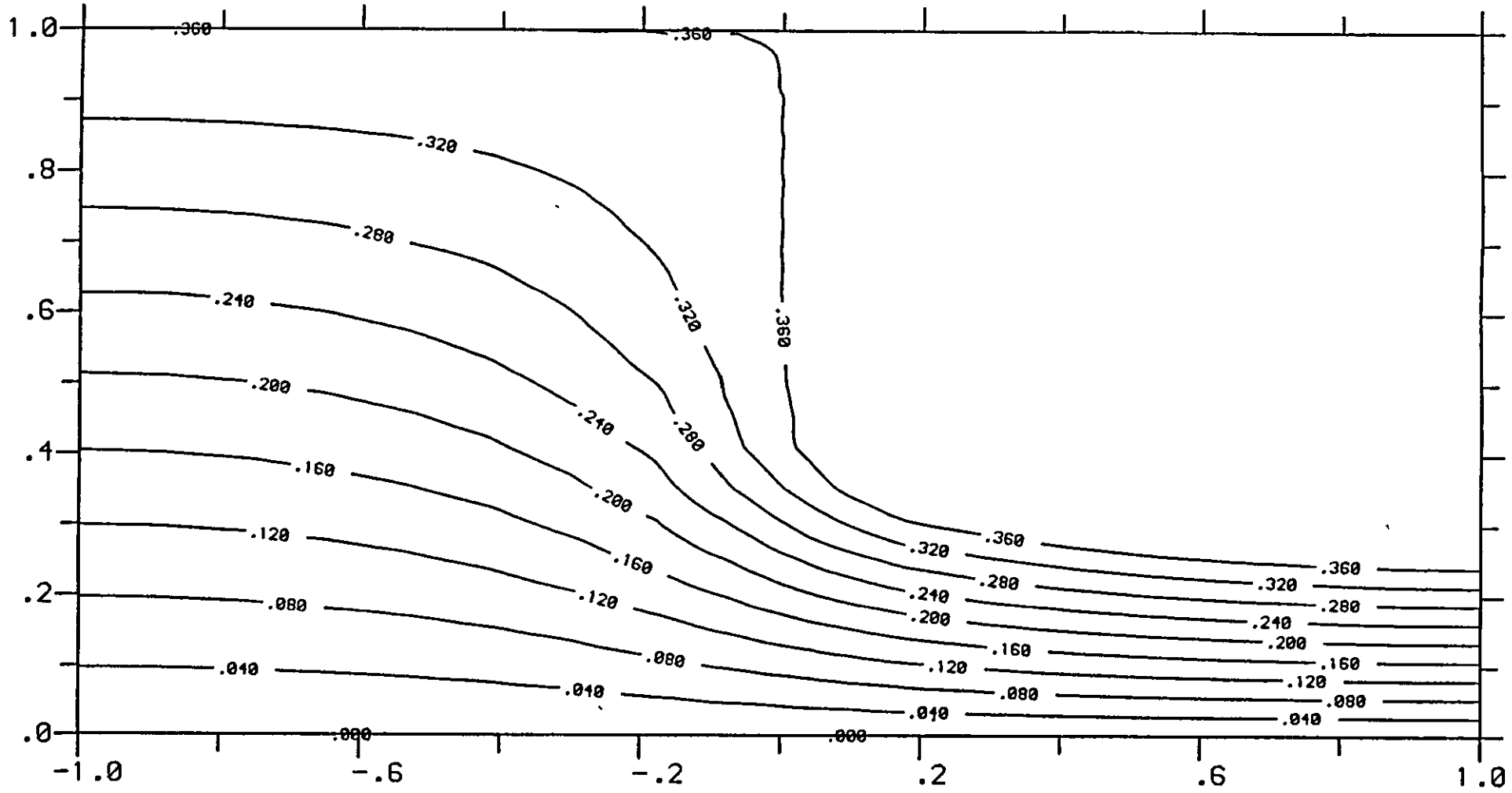


FIGURE 5.7: Potential flow solution with b/H = .4

5.5 CONVERGENCE AND ERROR ANALYSIS FOR THE FREE BOUNDARY PROBLEM

Some theoretical work has been done on the convergence of the trial free boundary problem. On a fixed region it is possible to analyse the errors even if the region has a curved boundary. This is done for the finite element method by, for example, STRANG and FIX [1974] and their analysis can be applied to the problems studied here for the fixed region.

It is much more difficult to analyse the errors in a free boundary problem. CRYER [1976] has given a proof for the following model problem in ideal flow. The model free boundary problem to be considered is as follows. Find u satisfying,

$$\nabla^2 u = u_{xx} + u_{yy} = 0, \text{ in } R, \tag{5.24}$$

where R is as shown in Figure (5.8).

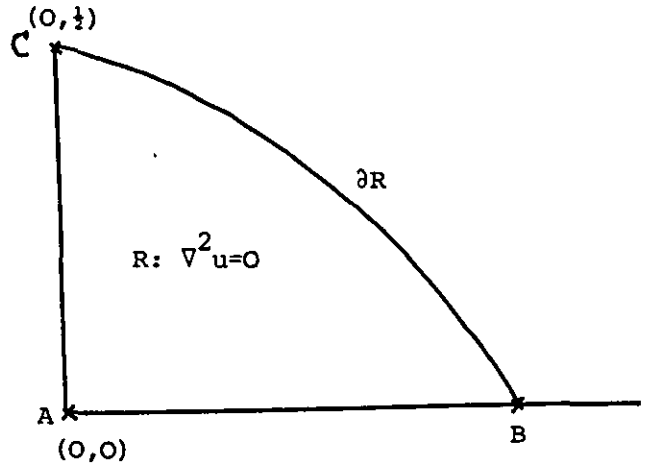


FIGURE 5.8: Cryer model free boundary problem

The boundary conditions are given by:

$$Lu = 0 = \begin{cases} u-(1-y), & \text{on AC,} \\ \dfrac{\partial u}{\partial n} + \sqrt{10}, & \text{on BC,} \\ \dfrac{\partial u}{\partial n} - 1, & \text{on AB.} \end{cases}$$

and the extra free boundary condition is,

$$Cu = 0 = u - \tfrac{1}{2}, \quad \text{on BC}, \tag{5.25}$$

The free boundary is the curve ∂R.

The auxiliary restraints are that ∂R should pass through the fixed point C on ∂R and should be a monotone decreasing function of x.

The problem is constructed so that if $\partial R^{(k)}$ is a straight line passing through C, with the condition $Cu^{(k)} = 0$ on $\partial R^{(k+1)}$ for moving the boundary, then $\partial R^{(k+1)}$ is also a straight line passing through C, that is, $\partial R^{(k)}$ is assumed to be of the form,

$$y = \tfrac{1}{2} + m^{(k)}x, \quad \text{on } \partial R^{(k)}, \tag{5.26}$$

and also: $$y = \tfrac{1}{2} + m^{(k+1)}x, \text{on } \partial R^{(k+1)},$$

where $m^{(k)}$ and $m^{(k+1)}$ are the gradients of the lines. The true solution of this problem is given by,

$$u: u = 1 - y - 3x, \tag{5.27}$$

$$\partial R: y = \tfrac{1}{2} - 3x. \tag{5.28}$$

Then, the problem,

$$u_{xx}^{(k)} + u_{yy}^{(k)} = 0, \text{ in } R^{(k)},$$
$$Lu^{(k)} = 0, \quad \text{on } \partial R^{(k)},$$

has the exact solution,

$$u^{(k)} = 1-y+x\{[10[m^{(k)}]^2 + 1)]^{\frac{1}{2}} - 1\}/m^{(k)} \quad . \tag{5.29}$$

The condition that $Cu^{(k)}=0$ on $\partial R^{(k+1)}$ is satisfied exactly if $m^{(k+1)}$ is defined by:

$$m^{(k+1)} = \{[10([m^{(k)}]^2 +1)]^{\frac{1}{2}}-1\}/m^{(k)} \quad . \tag{5.30}$$

Thus, in this simple problem the approximate solutions $u^{(k)}$ and the approximate free boundary solution $\partial R^{(k)}$ are known exactly. To analyse the behaviour of the gradient $m^{(k)}$ it is helpful to observe that if,

$$f(m) = [1+m^2]^{\frac{1}{2}} - 10^{\frac{1}{2}} \quad , \tag{5.31}$$

then,
$$m^{(k+1)} = m^{(k)} - f(m^{(k)})/f'(m^{(k)}) \quad , \tag{5.32}$$

so the sequence $m^{(k)}$ is identical with the sequence which would be obtained by starting with the initial guess $m^{(0)}$ and applying one of the iterative methods like Newton's method to the equation $f(m)=0$, noting that $f(m)$ is convex for $m\leq 0$ and that for $f(0)<0$, we have the following theorem from Henrici [1964, p.79].

THEOREM 1

For any initial guess $m^{(0)}<0$, the sequence of approximate free boundary's solution $\partial R^{(k)}$ converges quadratically to the free boundary ∂R (true solution).

Now, given an approximate free boundary $\partial R^{(k)}$ and an approximate solution $u^{(k)}$, to obtain error estimates, we must be able to estimate two quantities:

(i) The difference $u - u^{(k)}$, where $u^{(k)}$ is the approximate solution of the problem,

$$\begin{aligned} Du^{(k)} &= 0 \ , \text{ in } R^{(k)} \quad , \\ Cu^{(k)} &= 0 \ , \text{ in } \partial R^{(k)} \quad . \end{aligned} \tag{5.33}$$

(ii) The difference $u-u^{(k)}$, where $u^{(k)}$ satisfies (5.33) and u is the solution of the problem,

$$Du = 0 \text{ in } R, \tag{5.34}$$

$$Cu = 0 \text{ on } \partial R.$$

There are often several different approaches to estimate the given equations (5.33), (5.34), for further details, many references are given in CRYER [1976].

Much of the literature is based on the assumption that $\partial R^{(k)}$ is smooth and therefore not always applicable to free boundary problems which usually involves corners, but the case when $\partial R^{(k)}$ has corners has been considered. WIGLEY [1969] has derived asymptotic expansions for the solution of second order elliptic equations in the neighbourhood of corners. The elimination of the singularity by conformal mapping was applied by Mason and Farkas [1972] in conjunction with a trial free boundary. The question of domain variations arises in the theory of the finite elements, because in general the boundaries of the finite elements do not always coincide with the boundary of the domain of the problem being solved.

STRANG and BERGER [1974] give estimates for the difference $u-u^{(k)}$ and $\text{grad}(u-u^{(k)})$ for Poissons' equation in the plane. AITCHISON [1977] used complex variable analysis to obtain an expansion for the free boundary in the neighbourhood of the singularity.

CHAPTER SIX

FINITE ELEMENT FOR PROBLEMS INVOLVING SINGULARITIES

6.1 INTRODUCTION

The problem of boundary singularities in the numerical solution of elliptic and parabolic partial differential equations has received a great deal of attention. These singularities arise when sudden changes occur either in the direction of the boundary, as at a re-entrant corner, or they may be associated with mixed boundary conditions. Such singularities are found in a wide variety of physical problems, such as stress analysis in regions with cracks, discontinuities, point sources, etc. (see BERNAL and WHITEMAN [1970]), flow around an obstacle, seepage of a water through a dam (AITCHISON [1972]), heat flow, diffusion or potential problems in regions with re-entrant corners, electrodes heat sources or sinks (BELL and CRANK [1973]).

The approximate solutions of the boundary value problem of mathematical physics can usually be found by methods such as the finite difference, or finite element method, as long as the problems contain no *singularities* inside the integration domain or on its boundary, as is often the case with mixed boundary value problems when singularities occur with one or more coefficients of the partial differential equation becoming singular there. In such problems, the solution will ordinarily also become singular, and the method which we are using (the finite element or the finite difference, etc.) will produce inaccurate results in the neighbourhood of the boundary singularities. It is often possible to reduce the region affected by the singularity, by using analytical solutions based on separable-variable or integral transform techniques for infinite or semi-infinite regions with relatively simple governing equations (usually Laplace's), however, such solutions are, in general,

difficult to obtain for finite regions with more complicated equations and boundary conditions and so a numerical solution is considered.

Special numerical schemes have been devised to obtain accurate solutions. The most popular methods being:

1. By using modified approximations to the governing differential equation and its solution near the singularity.
2. Methods based on conformal transformations, modified integral equations, modified collocation, power series, dual series for the removal of the singularities.
3. Grid refinement in the neighbourhood of the singularity.

In the approach 1, the standard approximations near the singularity are replaced by modified approximations based on the local analytical form of the singularity, such as a form of an asymptotic expansion by separable-variable or complex variable techniques. WAIT ET AL [1971] used finite element method, with bilinear basis functions supplemented by singular functions to solve the elliptic boundary value problems with corner singularities.

Approach 2 proved to be accurate and efficient for the solution of elliptic problems in simply-connected polygonal regions with general mixed boundary conditions, but the method is limited to differential equations which remain invariant under conformal transformations, PAPAMICHAEL [1978] considered the use of a conformal transformation method for the solution of some class of the two dimensional linear elliptic boundary value problems in simply-connected domains. He shows that this type of transformation of the problem overcomes the difficulties

associated with the numerical solution of the problems involving curved boundaries and boundary singularities and produces solutions of good accuracy.

Although approach 3, is more computationally involved than the other methods, since the order of the matrix is increased it is a viable alternative in that no knowledge of the form of the singularity is required and any symmetry present is preserved. In addition, with the fast growth of high speed computers in recent years, it appears that this concept to use finite elements that allow an easy transition from a region where a finite element solution is required at a high degree of refinement to a region where the degree of refinement is sufficient is most promising has proved to be highly accurate for the solution of elliptic problems in simply-connected polygonal regions with general mixed boundary conditions, as can be seen from the results obtained from the next sections.

6.2 PROBLEM FORMULATIONS

The boundary value problems which are considered here fall into two classes; one consisting of problems from potential theory and the other consisting of problems from elastostatics. Both classes are discussed for two-space dimensions.

The potential problems in two-space dimensions have the general forms,

$$\begin{aligned} \nabla^2 u &= f_1(x,y) \quad , \quad (x,y) \in R, \\ u &= g_1(x,y) \quad , \quad (x,y) \in \partial R_1 \ , \\ \frac{\partial u}{\partial n} &= g_2(x,y) \quad , \quad (x,y) \in \partial R_2 \ , \end{aligned} \tag{6.1}$$

where $R \in \mathbb{R}^2$ in (6.1) is a simply connected open bounded polygonal domain, with the boundary ∂R, in (6.1) the polygonal boundary ∂R consists of disjoint parts ∂R_1 and ∂R_2 so that $\partial R \equiv \partial R_1 \cup \partial R_2$, and $\frac{\partial}{\partial n}$ is the derivative in the direction of the outward normal to the boundary. The homogeneous Dirichlet forms of (6.1) can be written as,

$$\nabla^2 u = f \ , \quad \text{in } R \ , \tag{6.2}$$

with $u=0$ on ∂R.

In the usual Sobolev space setting the weak solution $u \in H_0^1(R)$ of (6.2) satisfies the relation,

$$a(u,v) \equiv \int_R \nabla u . \nabla v \, dR = \int_{R_1} f v dR \equiv F(v) \ , \tag{6.3}$$
$$\forall \ v \in H_0^1(R) \ .$$

Many two dimensional problems of linear elasticity can be formulated in terms of the biharmonic operator so that,

$$\begin{aligned} \nabla^4 u &= f_2(x,y) \ , \quad (x,y) \in R \\ u &= g_3(x,y) \ , \quad (x,y) \in \partial R, \\ \frac{\partial u}{\partial n} &= g_4(x,y) \ , \quad (x,y) \in \partial R \ , \end{aligned} \tag{6.4}$$

where R, ∂R and $\frac{\partial}{\partial n}$ are as defined for (6.1).

The weak solution $u \in H_0^2(R)$ of (6.4) satisfies,

$$a(u,v) = \int_R fv \, dR \quad , \quad \forall\, v \in H_0^2(R) \ . \tag{6.5}$$

Examples of two dimensional linear elastic problems are those of the bending of a thin plate, for which u is the transverse deflection from the equilibrium position under the action of a load, and of plane strain in which u is the Airy stress function.

Typical two-dimensional regions which present singularities when the boundary of R contain a re-entrant corner is that of an L shaped region. A re-entrant corner is a point where the boundary changes direction through an angle exceeding π as shown in Figure (6.1) below.

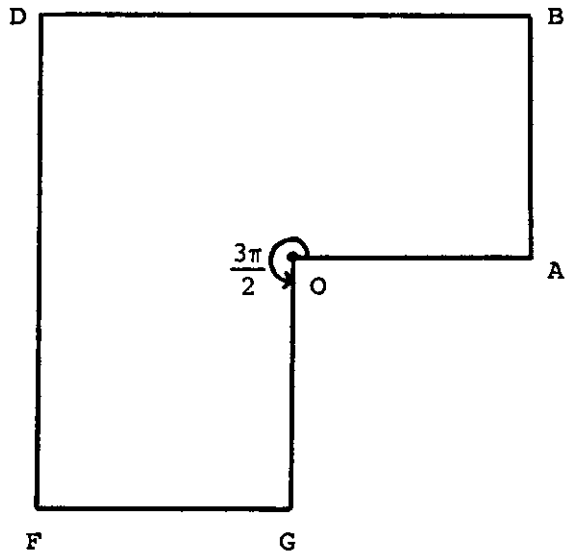


FIGURE 6.1: Re-entrant corner at O.

6.3 SINGULARITIES IN TWO-SPACE DIMENSIONS AND THE FINITE ELEMENT METHOD

We consider the two-dimensional problems of the type (6.2) and refer to Figure (6.2) below. Suppose that the boundary ∂R has vertices t_j, $j=1,2,\ldots,M$, with associated interior angles α_i, where

$$0 < \alpha_1 \leq \alpha_2 \ \ldots \leq \alpha_M \leq 2\pi,$$

and that at the jth vertex R_j denotes the intersection of R with a disc centred on t_j and containing no other corner. Let $R_0 \equiv R \setminus (\bigcup_{j=1}^{M} R_j)$.

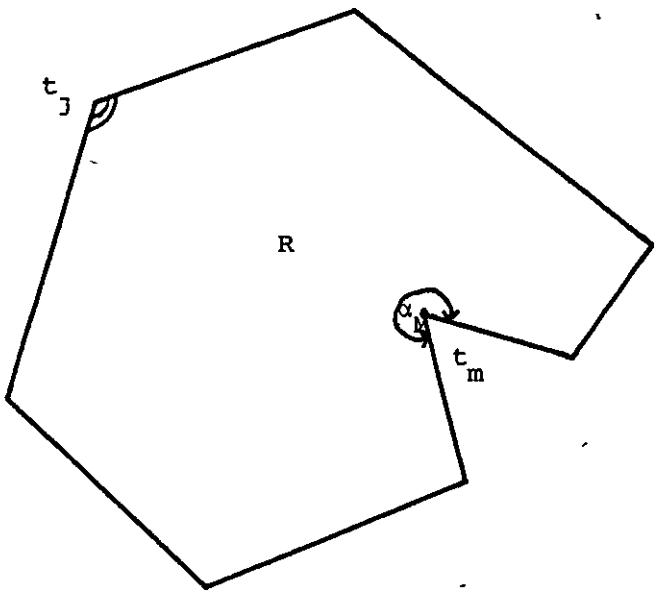


FIGURE 6.2

When the finite element method is applied to the two-dimensional form of (6.2) and (6.4) the solutions $u \in H_0^m(R)$, are approximated by $u_h \in S^h$, where $S^h \subset H_0^m(R)$ is a finite dimensional space and u_h satisfies

$$a(u_h, v_h) = F(v_h) \ , \quad \forall \ v_h \in S^h \ , \tag{6.6}$$

If $a(u,v)$ is continuous over $H_0^m(R)$ and H_0^m - elliptic, then it is well-known that,

$$||u-u_h||_{H_0^m(R)} \leq c||u-v_h||_{H_0^m(R)} \quad , \quad \forall\, v_h \in S^h \ , \qquad (6.7)$$

and further than, if S^h consists of piecewise polynomial conforming trial functions of degree p on a uniform triangular partition of R with mesh size h then the right-hand side of (6.7) can be bounded so that,

$$||u-u_h||_{H_0^m(R)} \leq kh^{\gamma}|u|_k \ , \qquad (6.8)$$

where γ depends on both k and p. The major determining factor for γ is the regularity of the solution u.

If we restrict ourselves further to two-dimensional second order problems of the form (6.2), for the bound (6.8) to be O(h), the solution u must be in $H^2(R)$. When re-entrant corners are present, which reduce the rate of convergence of $||u-u_h||_{L_\infty(R)}$. SCHATZ and WAHLBIN [1978] have for the two dimensional problems of the type (6.2) shown that, for a domain R with corners $\alpha_1,\alpha_2,\ldots,\alpha_M$ and the definitions given previously

$$||u-u_h||_{L_\infty(R)} \leq ch^{\min(\pi/\alpha_j,\, p+1,\, 2\pi/\alpha_M)-\varepsilon} \quad , \quad j=1,2,\ldots,M$$

$$||u-u_h||_{L_\infty(R_0)} \leq ch^{\min(p+1,\, 2\pi/\alpha_M)-\varepsilon} \ , \qquad (6.9)$$

The bound (6.9) indicates that the singularity causes a reduction in the rate of convergence both in the neighbourhood of the singularity and also away from it.

Taking the example of an L-shaped region with corners $\alpha_1=\alpha_2=\ldots=\alpha_5=\pi/2$, $\alpha_M\equiv\alpha_6=3\pi/2$, the respective rates of convergence, in the case of S^h consisting of piecewise linear functions, are $O(h^{2/3})$ and $O(h^{4/3})$.

These should be compared with the $O(h^2)$ which is expected when no singularities are present.

The analysis for the two-dimensional Poisson problem shows that some special adaptation of the finite element method is necessary in the neighbourhood of a singularity. A survey of different strategies is given by WHITEMAN and AITKIN [1979].

6.4 NUMERICAL RESULTS

TEST PROBLEM 1

The problem of Motz requires the solution of Laplace's equation in a rectangle with a slit, i.e. a re-entrant corner of internal angle 2π [see Figure (6.3)]. It has been treated by many authors to demonstrate the effectiveness of their singularity treatments.

WOODS [1953], and WAIT ET AL[1971] both gave an alternative formulation based on the fact that (u-500) is antisymmetric about the line EB containing the slit and by imposing the boundary condition u=500 on OE, only needed to consider the top half of the rectangle (see Figure (6.4)). It is in this same form that the problem is treated in the literature later on.

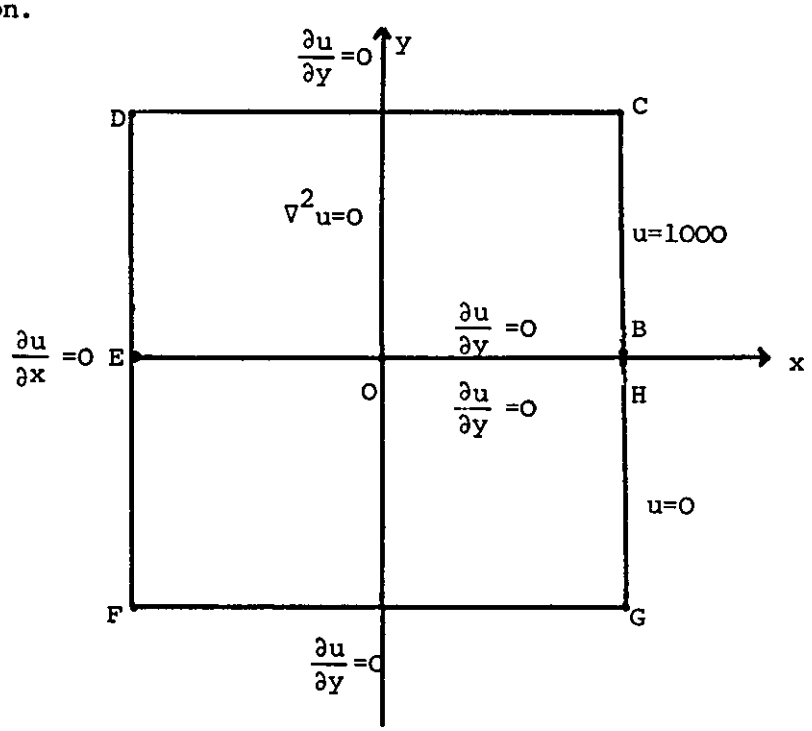


FIGURE 6.3

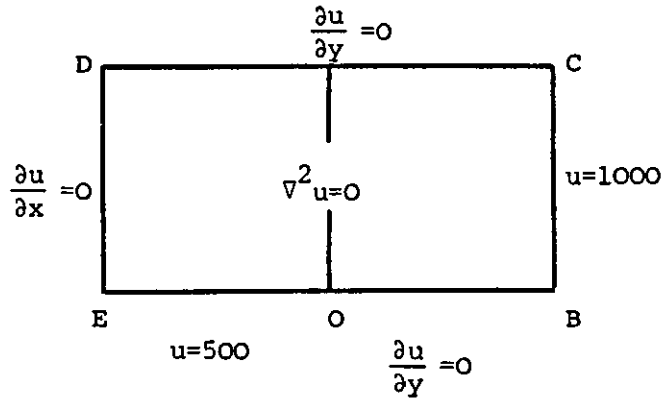


FIGURE 6.4: The modified source problem

Examples of such singularities occur in potential problems concerning transmission lines with microstrips, and in diffusion problems with narrow band type electrodes. The modified region and the boundary conditions are illustrated in Figure (6.4), where the problem is scaled by setting, $\nabla^2 u=0$,

in the region EDCB, $\{-1\leq x\leq 1\}\times\{0\leq y\leq 1\}$ containing the singular point O.

A printer plot of the geometry of the region with the initial triangulation generated by TWODEPEP is shown in Figure (6.5).

The results are quoted in Table (6.1) of the finite element solution of 300 elements and by using different basis functions (i.e. quadratic and the more accurate cubic basis functions), with the 2 different techniques i.e.,

i. equally distributed elements (both quadratic, and cubic) over the region as shown in Figure (6.6).

ii. with more refined elements in the neighbourhood of the singularity region as shown in Figure (6.7).

Comparing the results for both techniques, we note that the improvement in accuracy is better when we use (ii). The results of Table (6.2) are compared with the table of results given by J. CRANK and R.M. FURZELAND [1977].

The results in Table (6.2) show that a high degree of accuracy can be obtained, and agree very well over the solution region with the high accuracy of the results given by PAPAMICHAEL ET AL [1973]. The results of the highly accurate refined cubic elements is plotted in Figures (6.8a) and (6.8b) which shows the behaviour of the solution u in the given region.

FIGURE 6.5

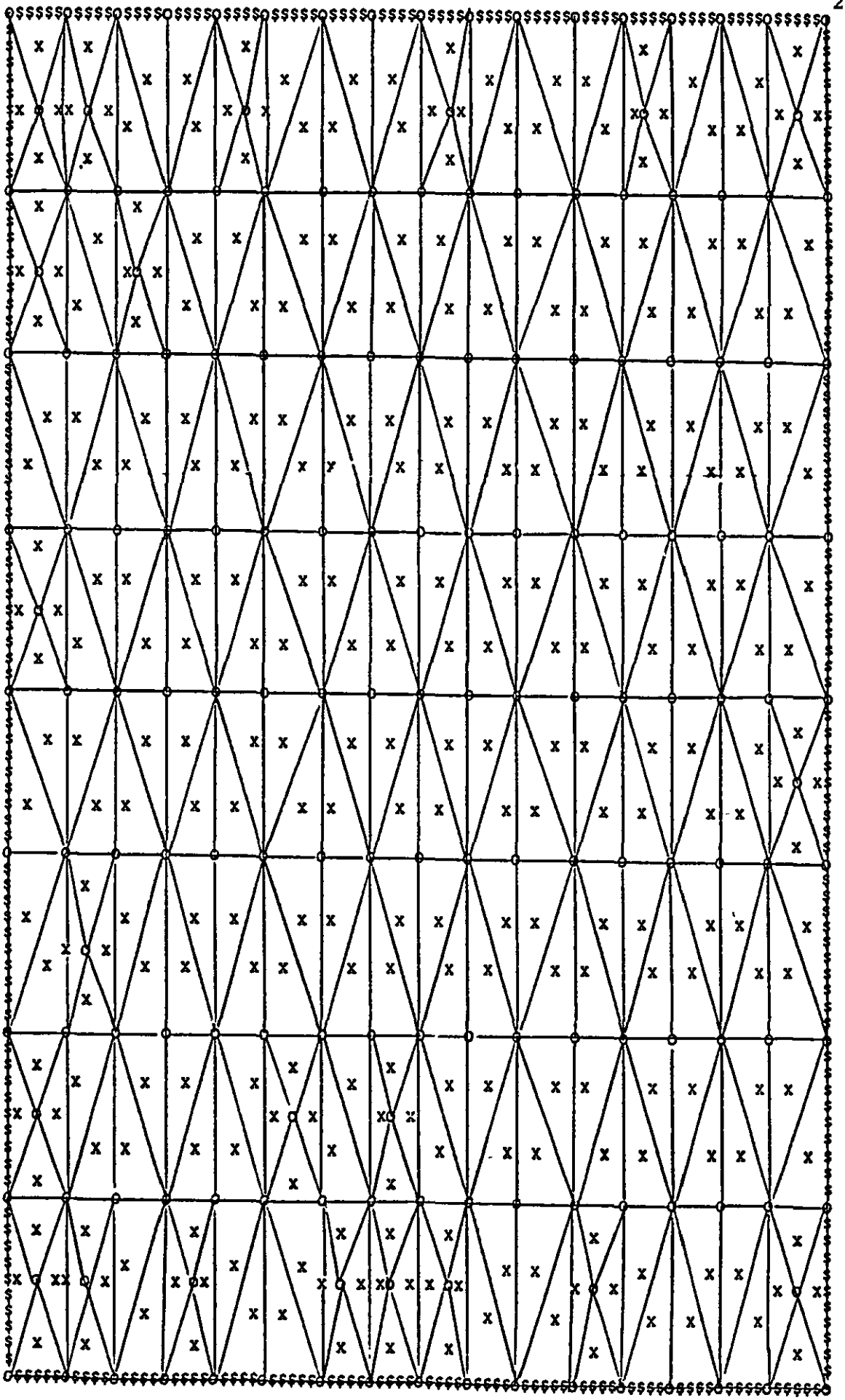


FIGURE 6.6

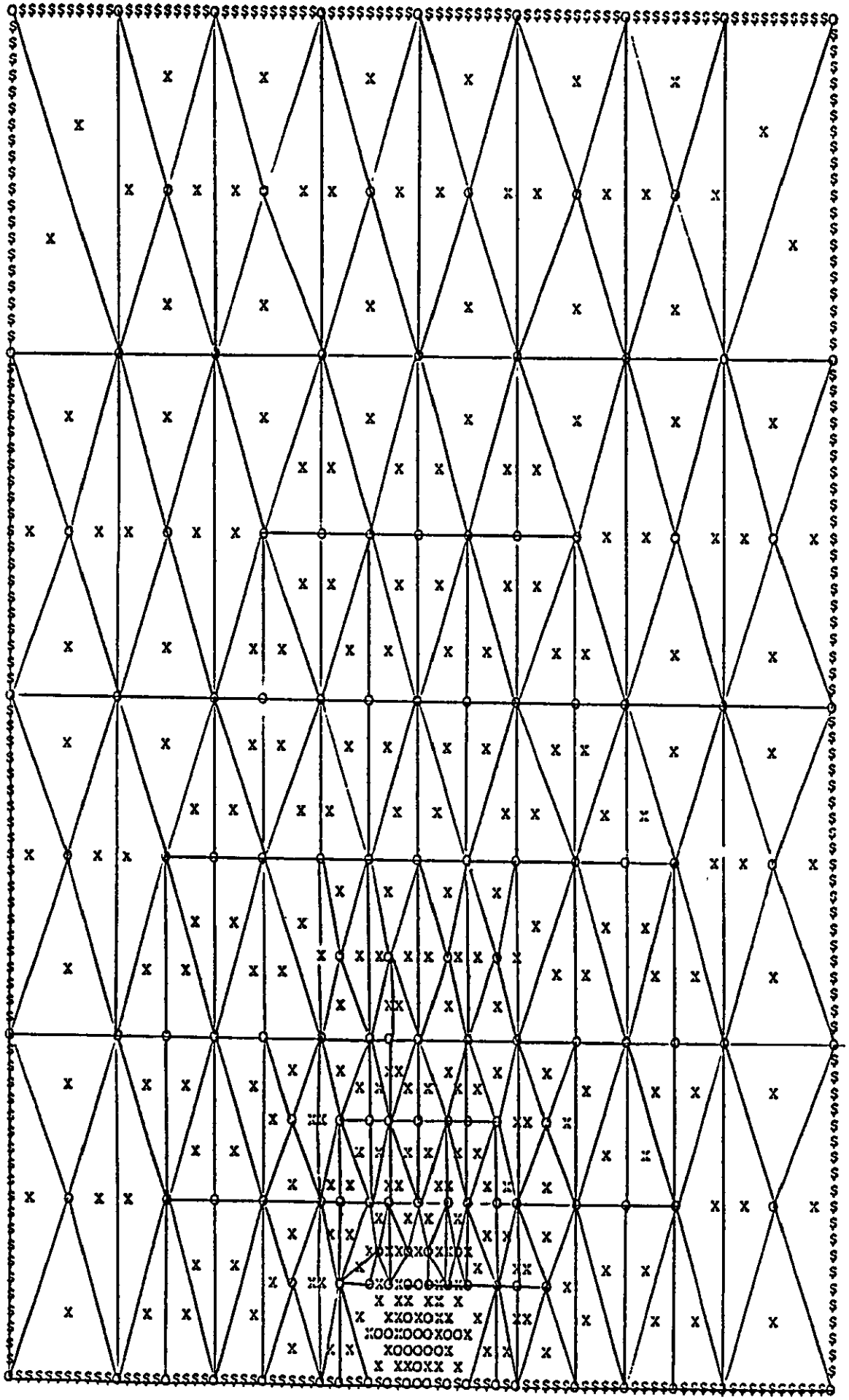


FIGURE 6.7

$\frac{\partial u}{\partial y} = 0$

D C

591.33	590.96	608.87	608.45	645.48	644.96	702.12	701.55	776.28	775.74	862.01	861.64	953.45	953.33
591.31	590.59	608.86	608.02	645.45	644.45	702.09	700.96	776.28	775.21	862.01	861.26	953.44	953.20
574.09	573.78	589.79	589.41	624.74	624.21	683.89	683.20	764.82	764.15	856.66	856.21	951.98	951.82
574.09	573.47	589.78	589.02	624.73	623.68	683.86	682.50	764.80	763.48	856.64	855.76	951.98	951.67
541.75	541.57	551.97	551.71	578.54	578.05	641.53	640.47	743.78	742.77	848.62	848.04	949.92	949.73
541.78	541.38	551.95	551.45	578.54	577.51	641.53	639.36	743.69	741.70	848.61	847.46	949.92	949.55
500		500		500		500		728.43	727.03	844.35	843.68	948.93	948.72
							0	728.34	725.61	844.32	842.99	949.91	948.51

E B

1000 1000 1000 1000

$\frac{\partial u}{\partial x} = 0$ u=1000

u = 500 $\frac{\partial u}{\partial y} = 0$

TABLE 6.1

At each point the numbers represent:

Finite element solution with cubic refined elements around O	Finite element solution with cubic equally distributed elements
Finite element solution with quadratic refined elements around O	Finite element solution with quadratic equally distributed elements

$\frac{\partial u}{\partial y} = 0$

D

591.33	591.3	608.87	608.9	645.48	645.5	702.12	702.1	776.28	776.3	862.01	862.0	953.46	953.5
592	591	609	608	645	644	702	701	777	776	863	863	954	954
574.09	574.1	589.79	589.8	624.74	624.8	683.89	683.9	764.82	764.8	856.66	856.7	951.98	952.0
575	574	591	590	625	624	683	683	766	765	857	858	952	953
541.75	541.8	551.97	552.0	578.54	578.6	641.53	641.6	743.78	743.8	848.62	848.6	949.92	949.4
543	542	553	553	579	579	640	642	746	744	849	849	950	950
500		500		500		500		728.43	728.5	844.35	844.4	948.93	948.9
								728	730	844	844	949	949

C

1000.00

1000.00

1000.00

1000.00

$\frac{\partial u}{\partial x} = 0$

$u = 1000$

E

B

$u = 500$

$\frac{\partial u}{\partial y} = 0$

TABLE 6.2

At each point the numbers represent:

our solution	Papamichael solution
Motz method	Woods method

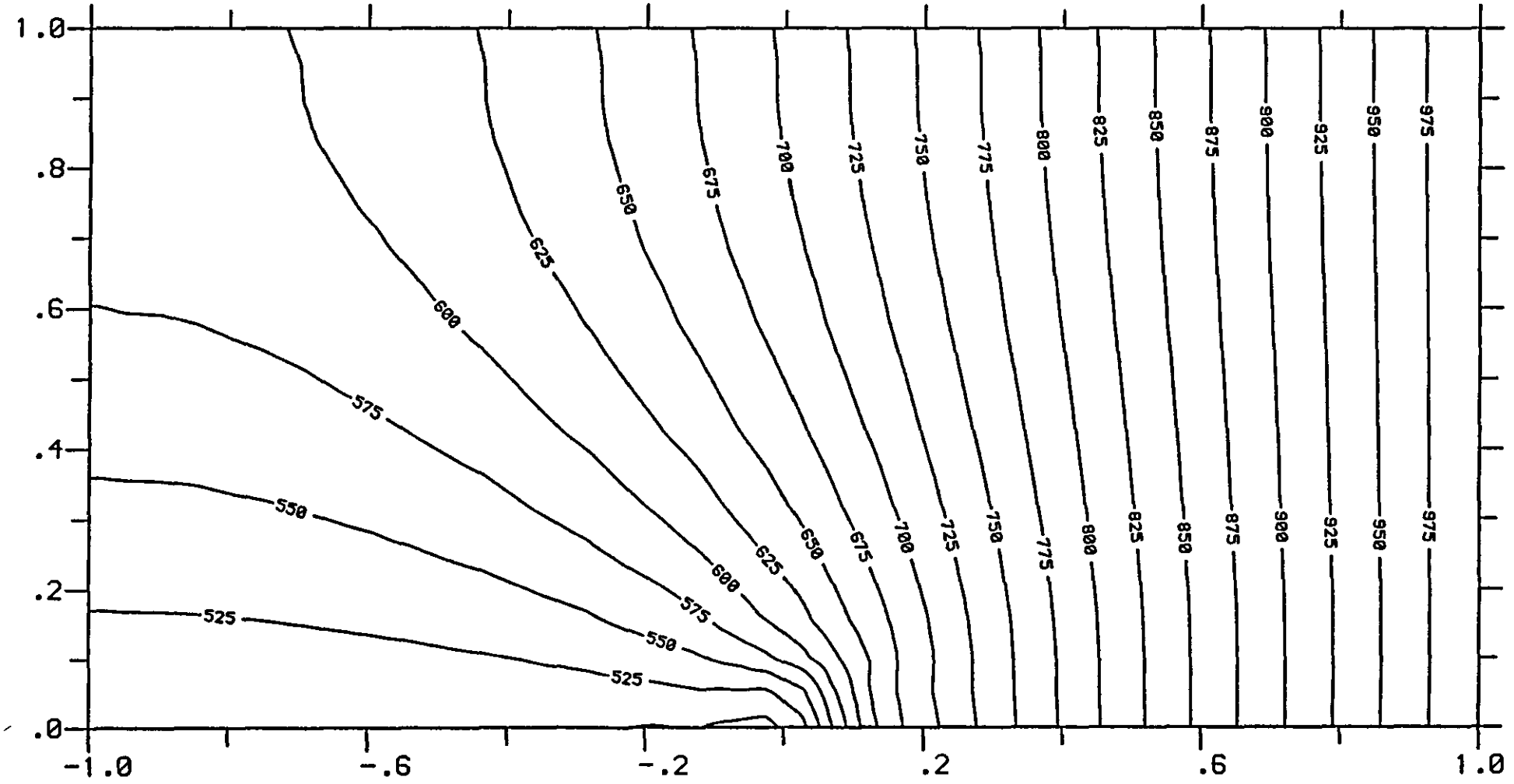

1.0
.8
.6
.4
.2
.0
-1.0
-.6
-.2
.2
.6
1.0
525
550
575
600
625
650
675
700
725
750
775
800
825
850
875
900
925
950
975

FIGURE 6.8a

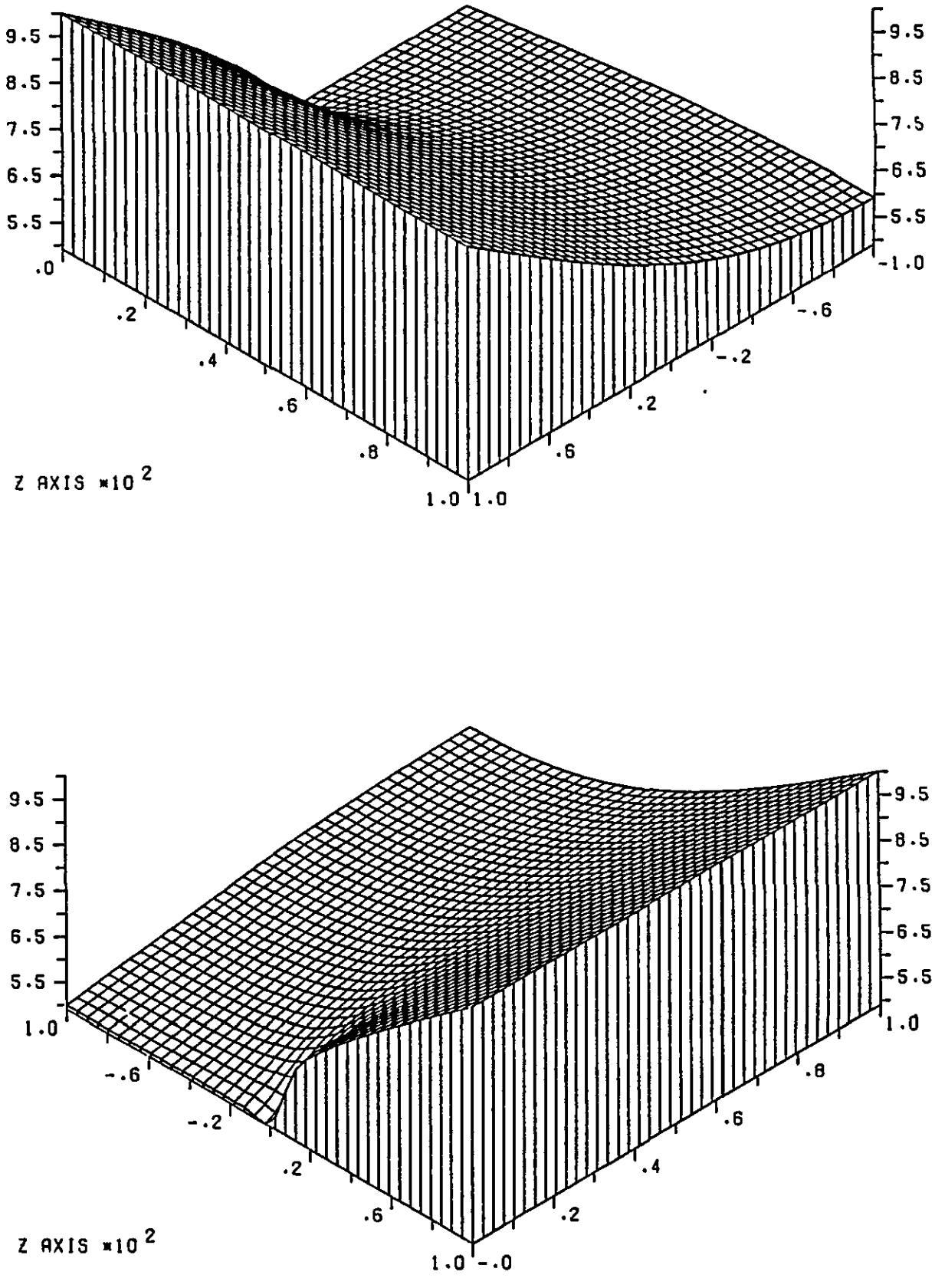

9.5
8.5
7.5
6.5
5.5
.0
.2
.4
.6
.8
1.0
1.0
.6
.2
-.2
-.6
-1.0
Z AXIS *10 2
9.5
8.5
7.5
6.5
5.5
1.0
-.6
-.2
.2
.6
1.0
-.0
.2
.4
.6
.8
1.0
Z AXIS *10 2

FIGURE 6.8b

TEST PROBLEM 2

The problem illustrated in Figure (6.9) has a boundary singularity at the origin. The given problem is,

$$\nabla^2 u(x,y) = 2u(x,y), \qquad (x,y) \in R,$$

$$u(x,y) = .2e^{x+y}, \qquad (x,y) \in \partial R,$$

where $R \equiv \{(x,y): x^2+y^2 \leq 1, \tan^{-1}(y/x) \leq \pi/4\}$.

This problem is chosen to illustrate the effectiveness of using the finite element p and h versions which were discussed in Chapter 3, and to demonstrate the effectiveness of the procedure for removing the singularity by mesh refining.

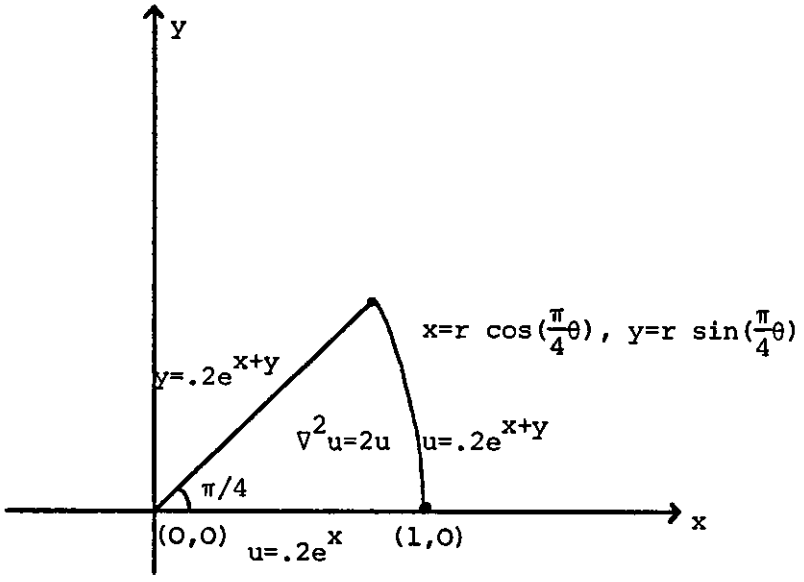


FIGURE 6.9

An estimate of the convergence of the numerical approximation to the exact solution can be obtained by computing $u-\hat{u}$, or $\frac{\partial u}{\partial n} - \frac{\partial \hat{u}}{\partial n}$ at a number of selected points.

Because of the high accuracy obtained and the reliability of the error estimates, then for comparison purposes we list the following results:

1. The error norm L_2 obtained by using the finite element p and h version directly to the given boundary value problem for the function u(x,y). The results are listed in Table (6.3).

2. The values obtained from the application of the 50,75 and 100 triangular finite elements and by applying quadratic, cubic and quartic basis functions for each case and also the values computed from the analytic solution

$$u = 0.2e^{x+y} ,$$

are listed in Table (6.4).

3. Values are obtained for the mesh points near the origin, for the both cases of equally distributed and more refined elements around the singularity. We note that the estimates computed at a set of test points in the region, suggest that an accuracy of five significant figures has been obtained for most cases. We list the results in Table (6.5).

4. Printer plots of the geometry of the region with the initial triangulation generated by TWODEPEP is shown in Figure (6.10), also Figures (6.11) and (6.12) show the discretised region of test problem 2 by using 300 triangular elements, with both equally distributed, and refined mesh procedures near the singularity respectively.

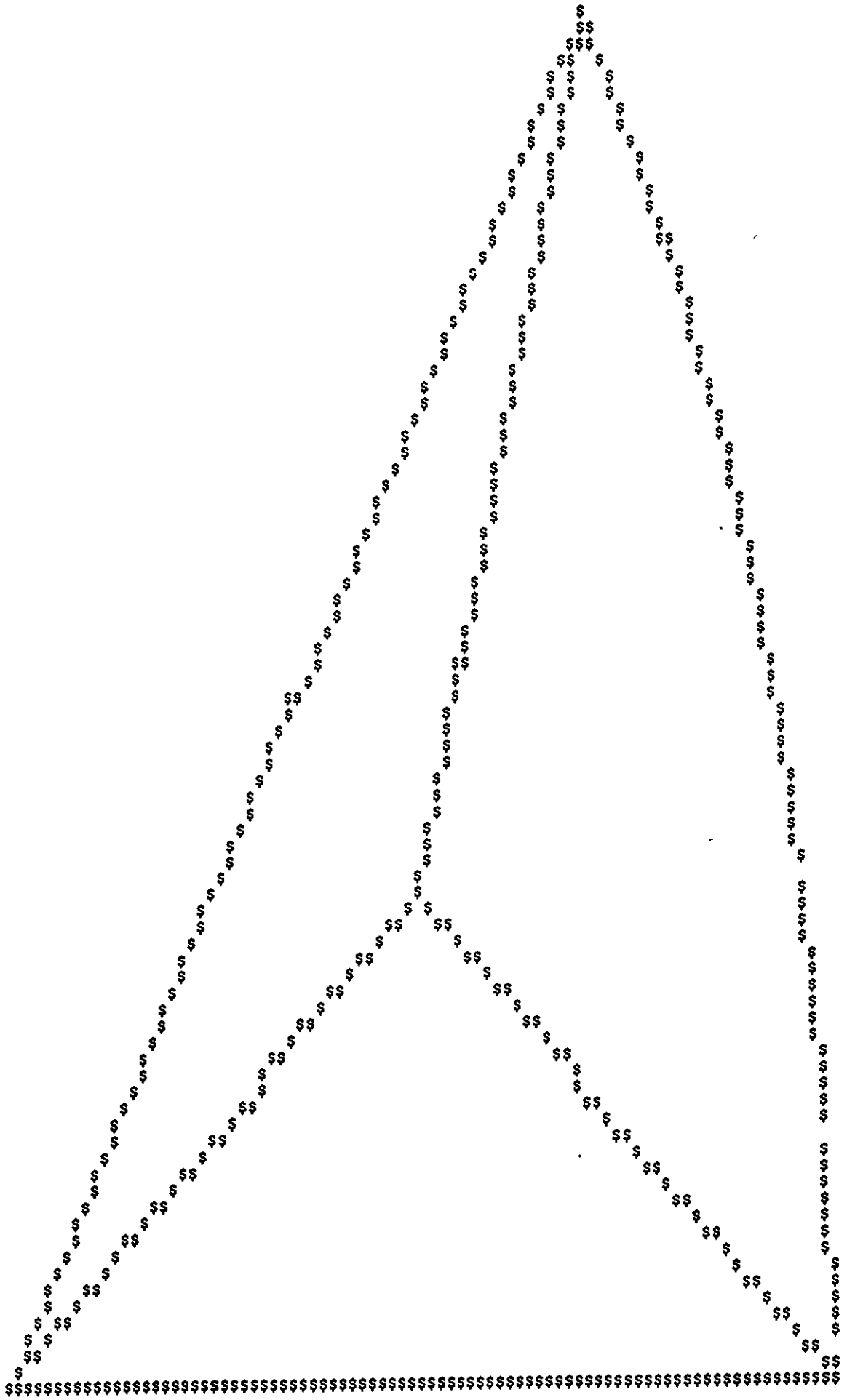


FIGURE 6.10

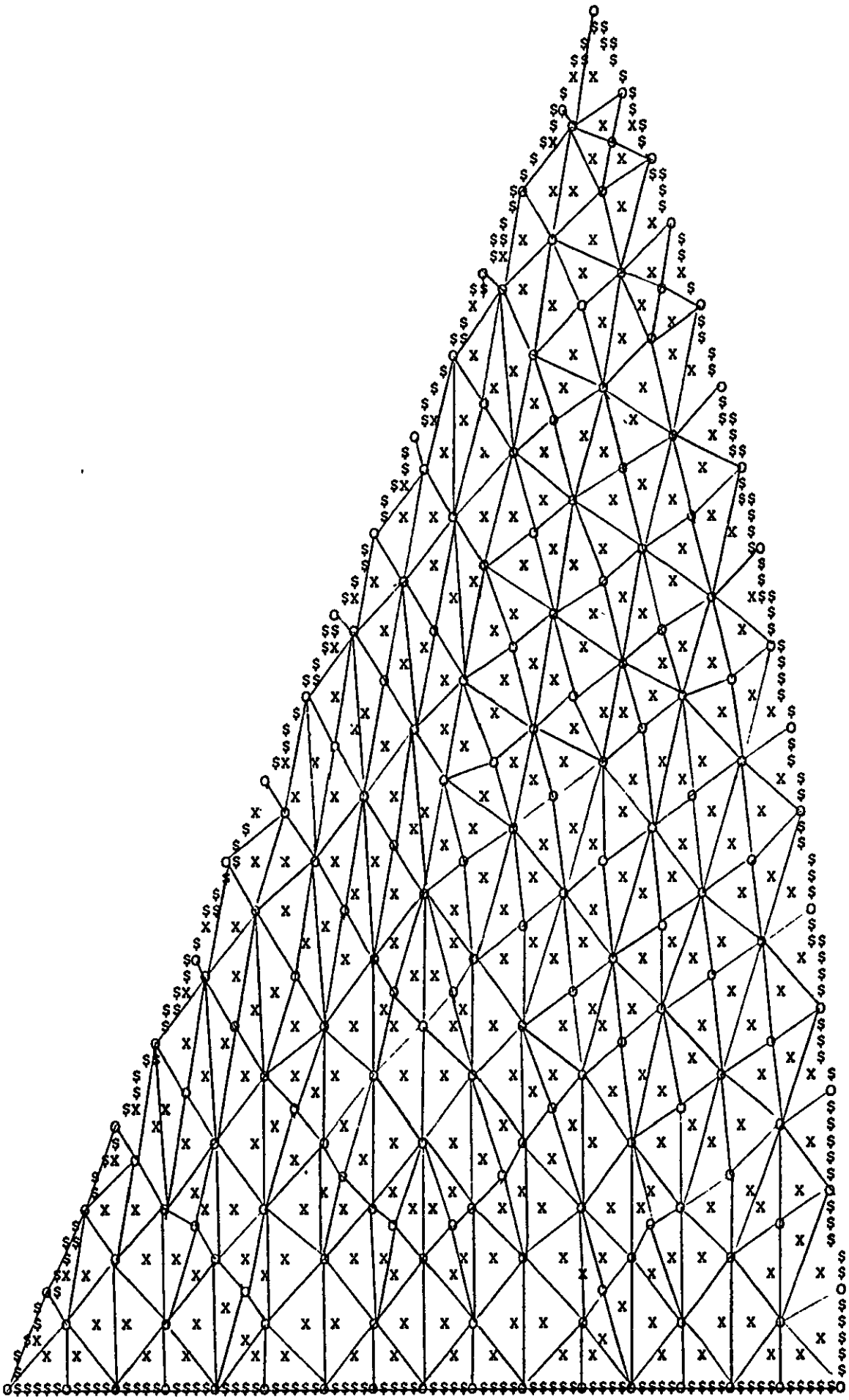


FIGURE 6.11

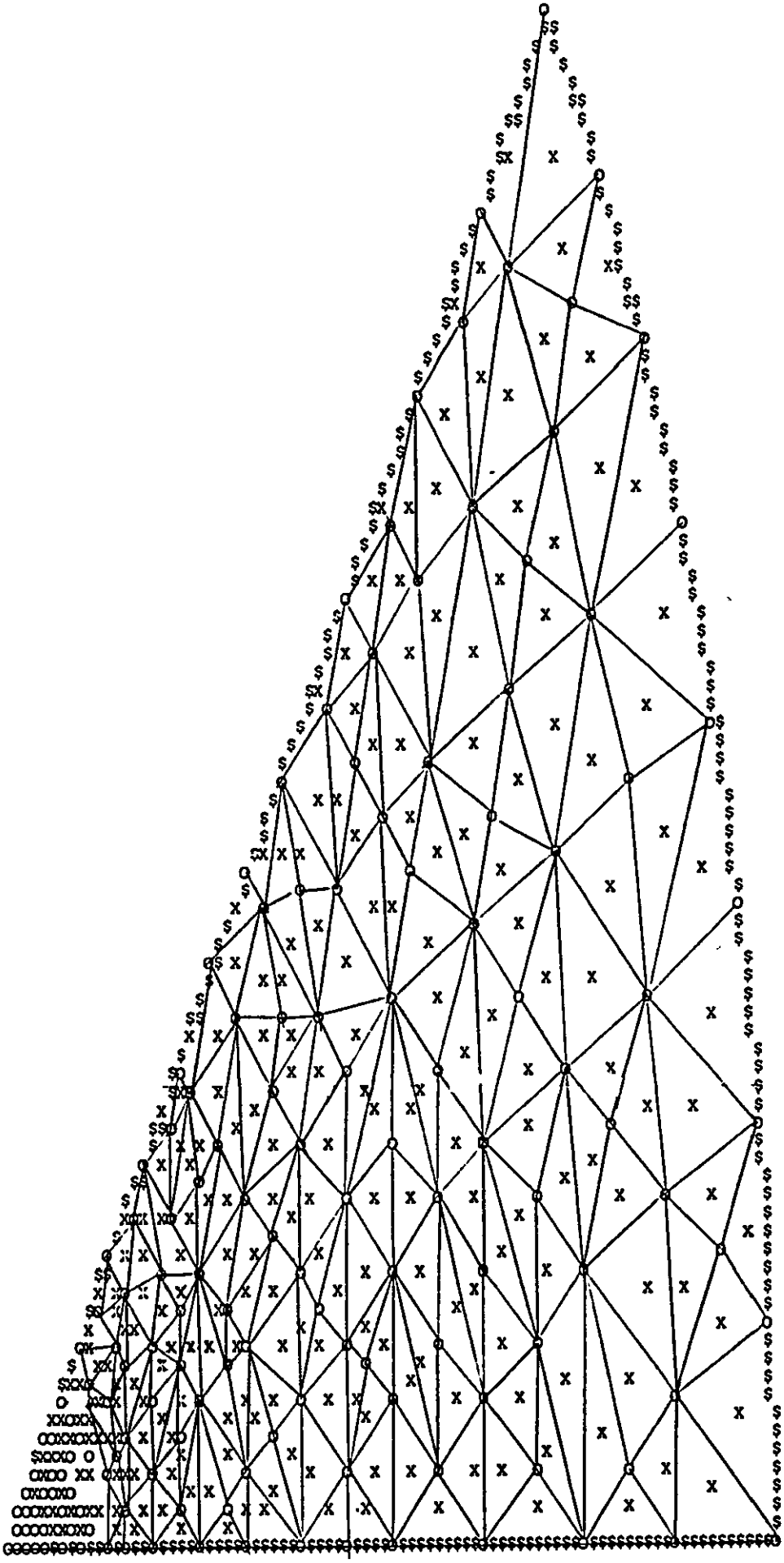


FIGURE 6.12

No.of elements \ Element order	Quadratics	Cubics	Quartics
50 Triangles	3.4×10^{-5}	3.55×10^{-6}	2.24×10^{-6}
75 Triangles	1.04×10^{-5}	2.00×10^{-6}	2.00×10^{-6}
100 Triangles	9.27×10^{-6}	2.00×10^{-6}	"

TABLE 6.3: Error L_2 norm

The error L_2, as the number of elements is subdivided, i.e. the h version, and also as the degree of the polynomial is increased, i.e. the p version of:

$$\nabla^2 u = 2u \in R$$

$$u = .2e^{x+y} \in \partial R$$

where,

$$R = \{(x,y): x^2+y^2 \leq 1, \quad \tan^{-1}(y/x) \leq \pi/4\} ;$$

with the exact solution is $0.2e^{x+y}$

x	y	50 ELEMENTS			75 ELEMENTS		100 ELEMENTS		Exact solution
		quad-ratics	cubics	quart-ics	quad-ratics	cubics	quad-ratics	cubics	
0.0	0.0	.20000	.20000	.20000	.20000	.20000	.20000	.20000	.20000
0.2	0.0	.24431	.24428	.24428	.24428	.24428	.24428	.24428	.24428
0.6	0.0	.36440	.36442	.36442	.36442	.36442	.36443	.36442	.36442
0.0	0.1	.22090	.22104	.22103	.22286	.22088	.22088	.22088	.22088
0.3	0.1	.29835	.29837	.29836	.29836	.29836	.29836	.29836	.29836
0.5	0.1	.36443	.36442	.36442	.36442	.36442	.36442	.36442	.36442
0.7	0.1	.44512	.44511	.44511	.44512	.44511	.44511	.44511	.44511
0.2	0.2	.29826	.29837	.29836	.29835	.29836	.29836	.29836	.29836
0.4	0.2	.36442	.36442	.36442	.36440	.36442	.36442	.36442	.36442
0.6	0.2	.44510	.44511	.44511	.44511	.44511	.44511	.44511	.44511
0.7	0.2	.49192	.49192	.49192	.49191	.49192	.49192	.49192	.49192
0.9	0.2	.60083	.60083	.60083	.60084	.60083	.60083	.60083	.60083
0.3	0.3	.36453	.36442	.36442	.36442	.36442	.36442	.36442	.36442
0.5	0.3	.44507	.44511	.44511	.44510	.44511	.44511	.44511	.44511
0.8	0.3	.60075	.60084	.60083	.60083	.60083	.60083	.60083	.60083
0.4	0.35	.42333	.42340	.42340	.42338	.42340	.42340	.42340	.42340
0.6	0.35	.51716	.51714	.51714	.51714	.51714	.51714	.51714	.51714
0.9	0.35	.69812	.69806	.69807	.69811	.69807	.69807	.69807	.69807
0.4	0.4	.44495	.44511	.44511	.44511	.44511	.44511	.44511	.44511
0.5	0.4	.49192	.49192	.49192	.49192	.49192	.49192	.49192	.49192
0.8	0.4	.66404	.66403	.66402	.66402	.66402	.66402	.66402	.66402
0.9	0.4	.73392	.73386	.73386	.73388	.73385	.73386	.73386	.73386
0.5	0.45	.51720	.51714	.51714	.51712	.51714	.51714	.51714	.51714
0.6	0.45	.57155	.57153	.57153	.57155	.57153	.57153	.57153	.57153
0.8	0.45	.69807	.69806	.69807	.69804	.69807	.69807	.69807	.69807
0.5	0.5	.54383	.54366	.54366	.54367	.54366	.54366	.54366	.54366
0.7	0.5	.66401	.66402	.66402	.66403	.66402	.66402	.66402	.66402
0.6	0.55	.63153	.63164	.63164	.63164	.63164	.63164	.63164	.63164
0.7	0.55	.69805	.69807	.69807	.69806	.69807	.69807	.69807	.69807
0.7	0.6	.73394	.73384	.73387	.73383	.73386	.73386	.73386	.73386
0.7	0.65	.77164	.77148	.77148	.77151	.77148	.77148	.77148	.77149
0.7	0.7	.81113	.81104	.81104	.81106	.81104	.81104	.81104	.81104

TABLE 6.4: Comparison of discretization errors, for test problem 2

At each point the numbers represent the values computed by using 300 elements with different basis functions as indicated below:

x	y	300 quadratics elements		300 cubics elements		Exact solution
		Equally distributed	Refined	Equally distributed	Refined	
0.0	0.0	.20000	.20000	.20000	.20000	.20000
0.1	0.0	.22103	.22103	.22103	.22103	.22103
0.2	0.0	.24428	.24428	.24428	.24428	.24428
0.3	0.0	.26997	.26997	.26997	.26997	.26997
0.4	0.0	.29836	.29836	.29836	.29836	.29836
0.5	0.0	.32974	.32974	.32974	.32974	.32974
0.0	0.05	.21044	.21025	.21025	.21025	.21025
0.1	0.05	.23237	.23237	.23237	.23237	.23237
0.2	0.05	.25680	.25680	.25681	.25681	.25681
0.3	0.05	.28381	.28281	.28381	.28381	.28381
0.4	0.05	.31366	.31366	.31366	.31366	.31366
0.5	0.05	.34665	.34665	.34665	.34665	.34665
0.0	0.1	.22103	.22104	.22103	.22103	.22103
0.1	0.1	.24428	.24428	.24428	.24428	.24428
0.2	0.1	.26997	.26997	.26997	.26997	.26997
0.3	0.1	.29836	.29836	.29836	.29836	.29836
0.4	0.1	.32975	.32974	.32974	.32974	.32974
0.5	0.1	.36442	.36442	.36442	.36442	.36442
0.0	0.15	.23237	.23237	.23237	.23237	.23237
0.1	0.15	.25676	.25681	.25680	.25681	.25681
0.2	0.15	.28381	.28381	.28381	.28381	.28381
0.3	0.15	.31366	.31366	.31366	.31366	.31366
0.4	0.15	.34665	.34665	.34665	.34665	.34665
0.5	0.15	.38311	.38311	.38311	.38311	.38311
0.0	0.2	.24429	.24429	.24429	.24428	.24428
0.1	0.2	.26982	.26694	.26997	.26997	.26997
0.2	0.2	.29837	.29837	.29836	.29836	.29836
0.3	0.2	.32974	.32974	.32974	.32974	.32974
0.4	0.2	.36442	.36442	.36442	.36442	.36442
0.5	0.2	.40275	.40275	.40275	.40275	.40275

TABLE 6.5

TEST PROBLEM 3

The problem illustrated in Figure (6.13) involves a re-entrant corner of internal angle $\frac{3\pi}{2}$, at which a boundary singularity occurs. Statement of the problem:

$$\nabla^2 u = -(16x^2+1)u + 4\cos(2x^2-y) \quad \in R$$

$$u = \sin(2x^2-y) \quad \in \partial R_1$$

$$\frac{\partial u}{\partial x} = 0 \quad \in \partial R_2$$

where R is the circular sector:

$$R = \{(x,y): \; x^2+y^2 \leq 1, \; \tan^{-1}(\frac{y}{x}) \leq \frac{3\pi}{2} \}$$

and $\partial R = \partial R_1 \cup \partial R_2$

$\partial R_1 = \{(x,y): -1<x<1, \; 0<y<1\}$,

$\partial R_2 = \{(0,y): -1<y<0\}$,

with the analytic solution,

$$u = \sin(2x^2-y) \; .$$

For comparison purposes we list the following results:

1. The error norm L_2 obtained by using the finite element p and h versions directly to the given boundary value problem for the function u(x,y); we list the result in Table (6.6).

2. Values obtained at the mesh points near the singularity by applying the procedure of mesh refining near the singularity; we list the results in Table (6.7).

3. Printer plots of the geometry of the region with the initial triangulation generated by TWODEPEP is shown in Figure (6.14), also Figures (6.15) and (6.16) shows the discretised region of this test problem by using 300 triangular elements with equally distributed and mesh refining procedure respectively.

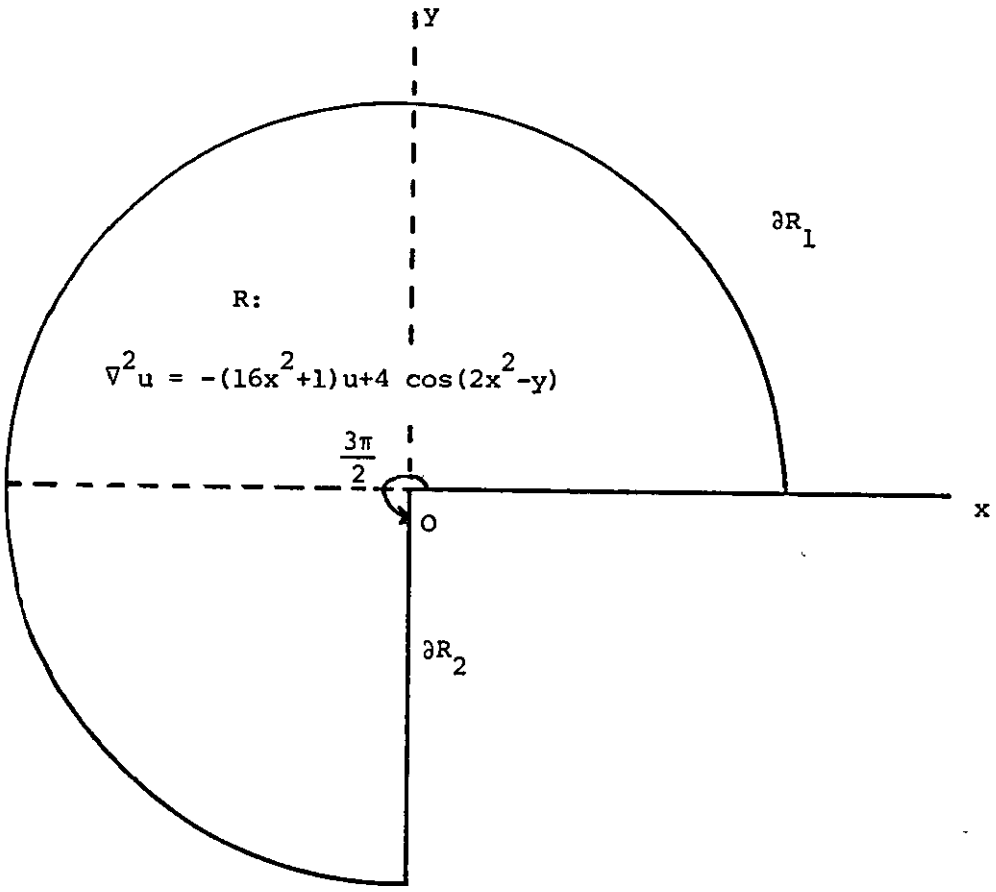

Y
∂R_1
R:
$\nabla^2 u = -(16x^2+1)u+4\cos(2x^2-y)$
$\frac{3\pi}{2}$
O
x
∂R_2

FIGURE 6.13

No. of elements \ Element order	Quadratics	Cubics	Quartics
50 Triangles	2.82×10^{-3}	4.33×10^{-4}	5.36×10^{-5}
75 Triangles	3.03×10^{-3}	1.65×10^{-4}	2.53×10^{-5}
100 Triangles	1.12×10^{-3}	1.11×10^{-4}	

TABLE 6.6: The error L_2 norm, as the number of elements is subdivided (the h version) and also as the order of the polynomial is increased (the p version) of test problem 3.

At each point the numbers represent the values computed by using 300 elements with different basis functions as indicated below:

x	y	300 quadratics elements		300 cubics elements		Exact solution
		Refined around 0	Equally distributed	Refined around 0	Equally distributed	
0.0	-1.0	.84147	.84147	.84147	.84147	.84147
-.4	-.8	.90018	.90025	.90011	.90005	.90010
-.2	-.8	.77050	.77224	.77074	.77078	.77074
0.0	-.8	.71700	.71700	.71736	.71738	.71736
-.6	-.6	.96929	.96707	.96872	.96869	.96872
-.4	-.6	.79524	.79464	.79560	.79558	.79560
-.2	-.6	.62890	.62981	.6288	.62875	.62879
0.0	-.6	.56473	.56453	.56464	.56464	.56464
-.8	-.4	.99394	.99347	.99404	.99393	.99404
-.6	-.4	.90006	.89996	.90010	.90014	.90010
-.4	-.4	.65972	.65956	.65939	.65939	.65938
-.2	-.4	.46176	.46187	.46178	.46178	.46178
0.0	-.4	.38943	.38949	.38942	.38942	.38942
-.8	-.2	.99609	.99623	.99589	.99589	.99588
-.6	-.2	.79582	.79581	.79560	.79561	.79560
-.4	-.2	.49686	.49649	.49688	.49687	.49688
-.2	-.2	.27632	.27641	.27636	.27636	.27636
0.0	-.2	.19866	.91871	.19867	.19867	.19867
-1.0	0.0	.90930	.90930	.90930	.90930	.90930
-.8	0.0	.95901	.96341	.95803	.95807	.95802
-.6	0.0	.66156	.66156	.65951	.65951	.65934
-.4	0.0	.31450	.31481	.31457	.31457	.31457
-.2	0.0	.079881	.079956	.079914	.079916	.079915
0.0	0.0	0.0	0.0	0.0	0.0	0.0
-.8	.2	.88226	.88201	.88195	.88194	.88196
-.6	.2	.49679	.49696	.49688	.49686	.49688
-.4	.2	.11953	.11982	.11969	.11968	.11971
-.2	.2	-.11961	-.11961	-.11972	-.11971	-.11971
0.0	.2	-.19862	-.19859	-.19867	-.19867	-.19867
-.2	.4	-.31444	-.31436	-.31457	-.31457	-.31457
0.0	.4	-.39841	-.38937	.-38942	-.38942	-.38942

TABLE 6.7

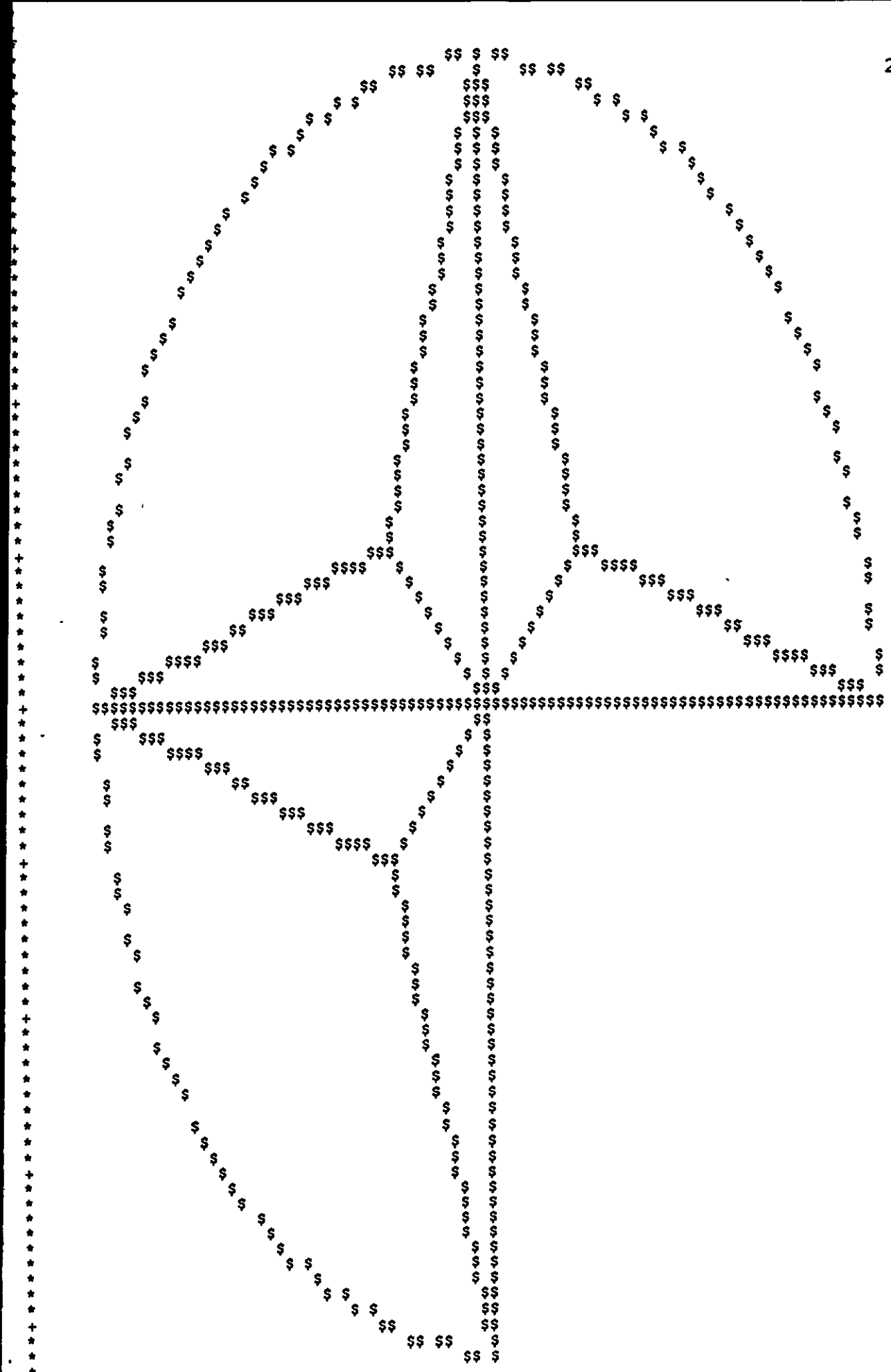


FIGURE 6.14

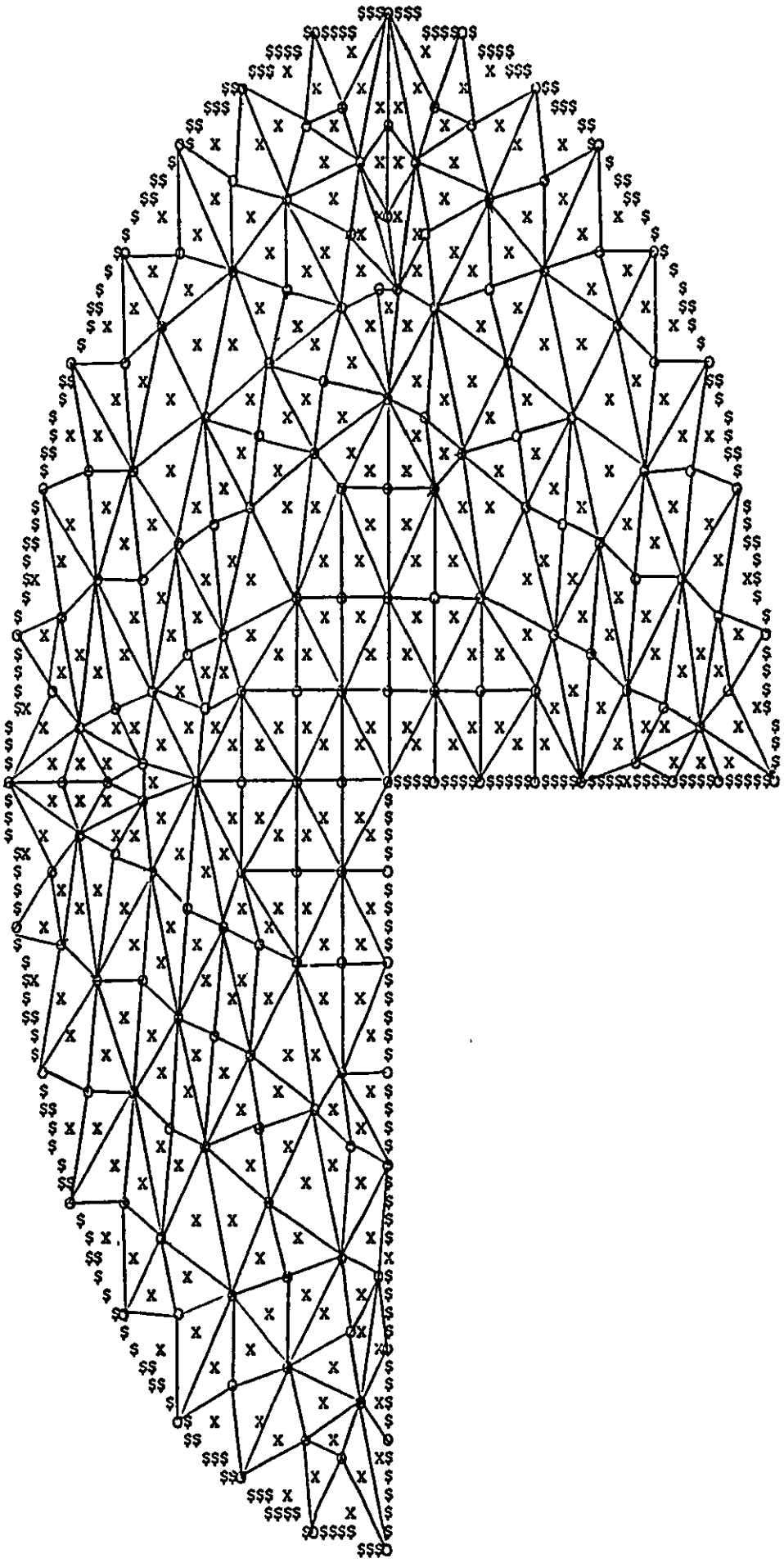


FIGURE 6.15

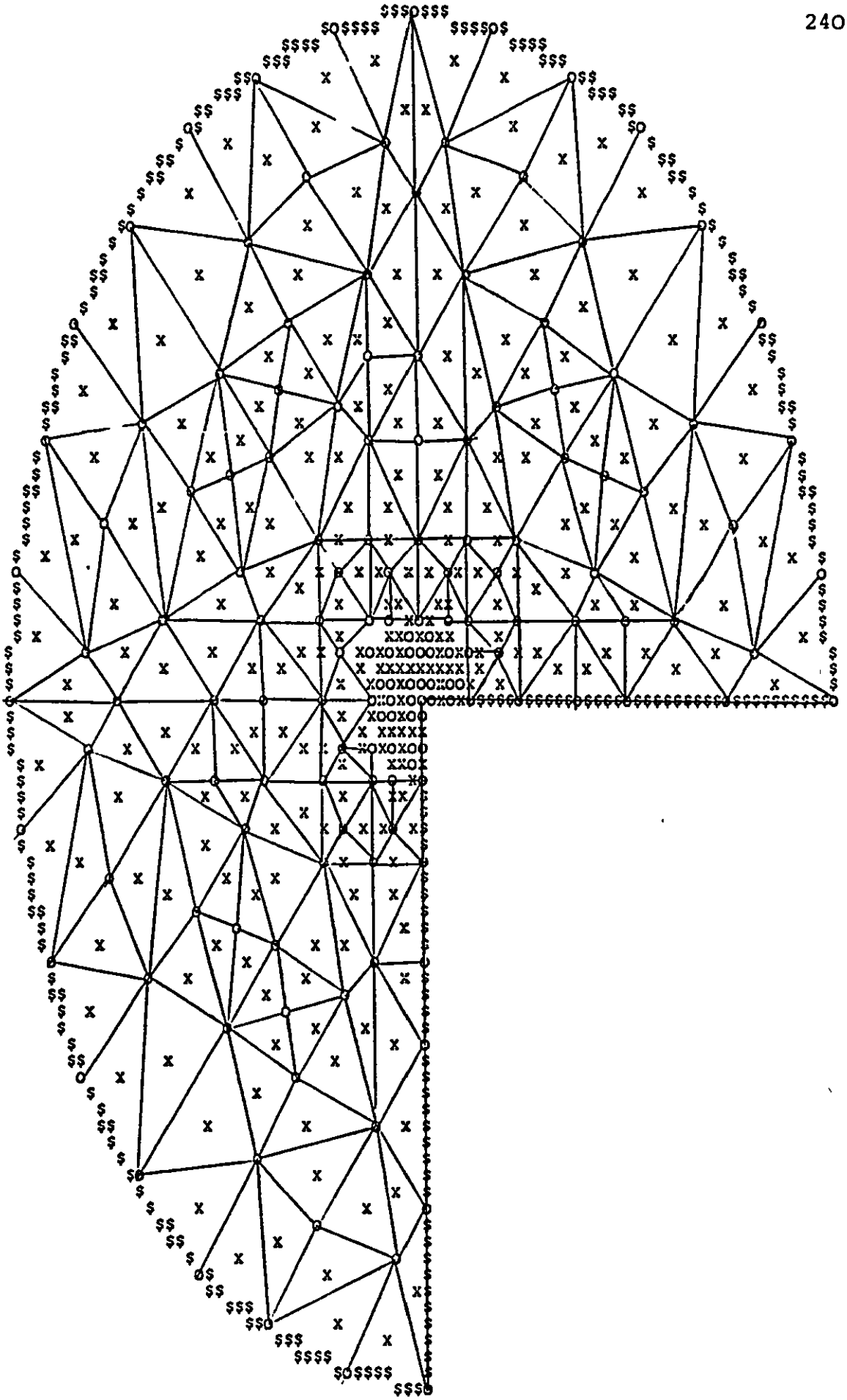


FIGURE 6.16

TEST PROBLEM 4

The harmonic problem illustrated in Figure (6.17) involves a re-entrant corner, of internal angle $3\pi/2$, at which a boundary singularity occurs. The problem arises in a study of diffusion in a continuum containing non-permeable rectangular prisms; [see BELL and CRANK (1974)].

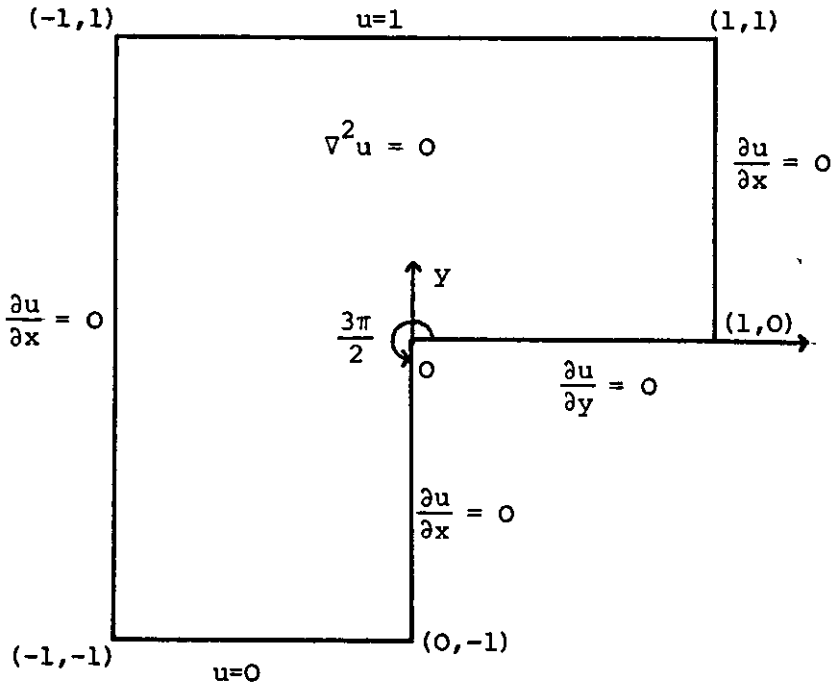


FIGURE 6.17

1. The results obtained are given in Table (6.8) and are compared with the conformal transformation method of PAPAMICHAEL (1978). These results are extremely accurate, and correct to the number of figures quoted.

2. The results in Table (6.9) are obtained by using quadratic triangular elements as basis functions, and are compared with the numerical solution given by SYMM (1973), who uses an integral equation approach modified to deal with the singularity at a re-entrant corner. The results obtained are also extremely accurate.

3. The results given in Table (6.10) compare two finite element solutions by using the piecewise polynomial functions of the same degree, but the first set is obtained with a mesh refinement around the singularity. The same number of elements was used in both procedures. The first set of results in Table (6.10) indicates that we can attain the accuracy required without refining the whole region.

4. Printer plots of the geometry of the region with the initial triangulation generated by TWODEPEP is shown in Figure (6.18). Also Figures (6.19) and (6.20) show the discretised region of test problem 4 with equally distributed triangles and mesh refining near the singularity respectively.

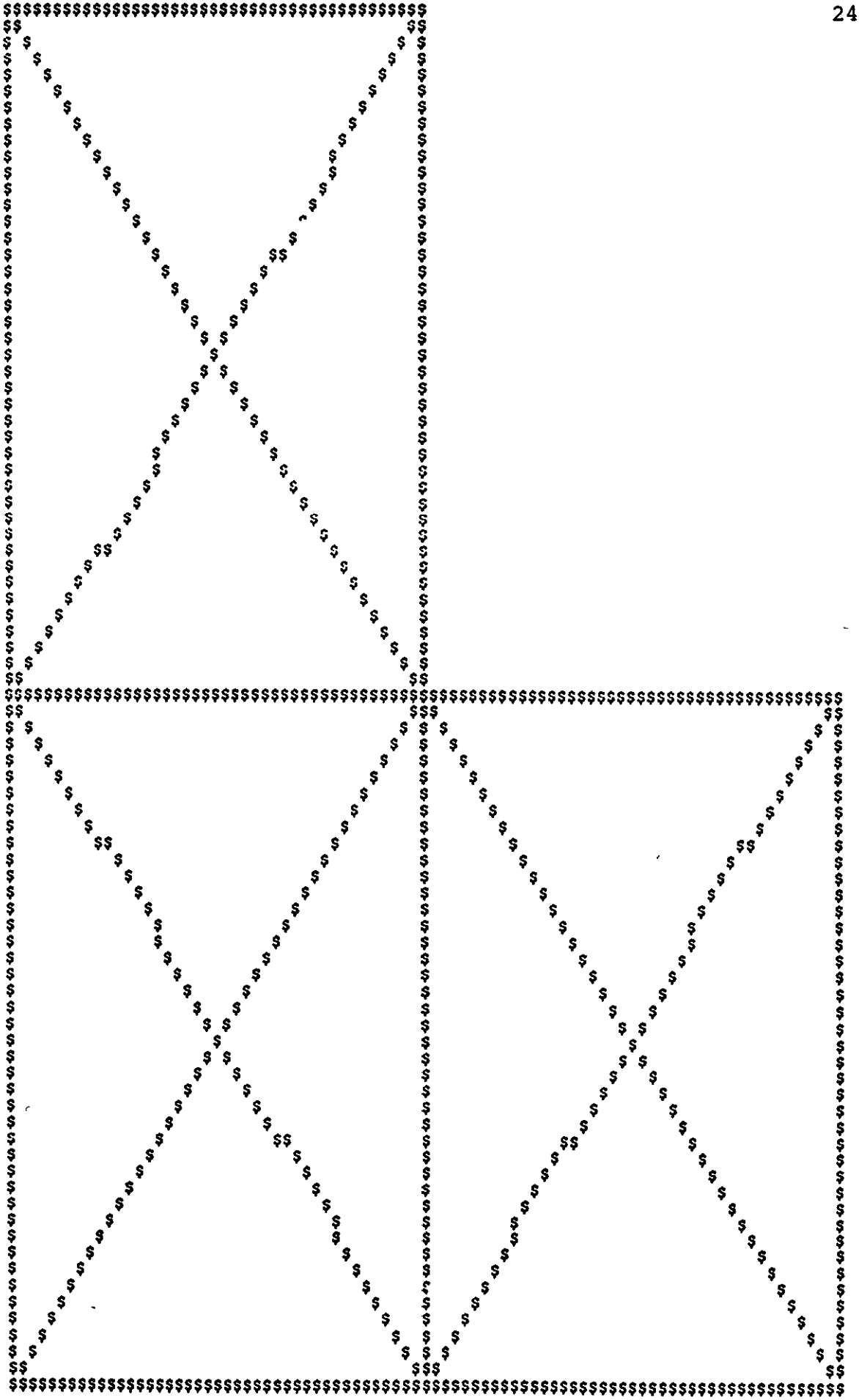


FIGURE 6.18

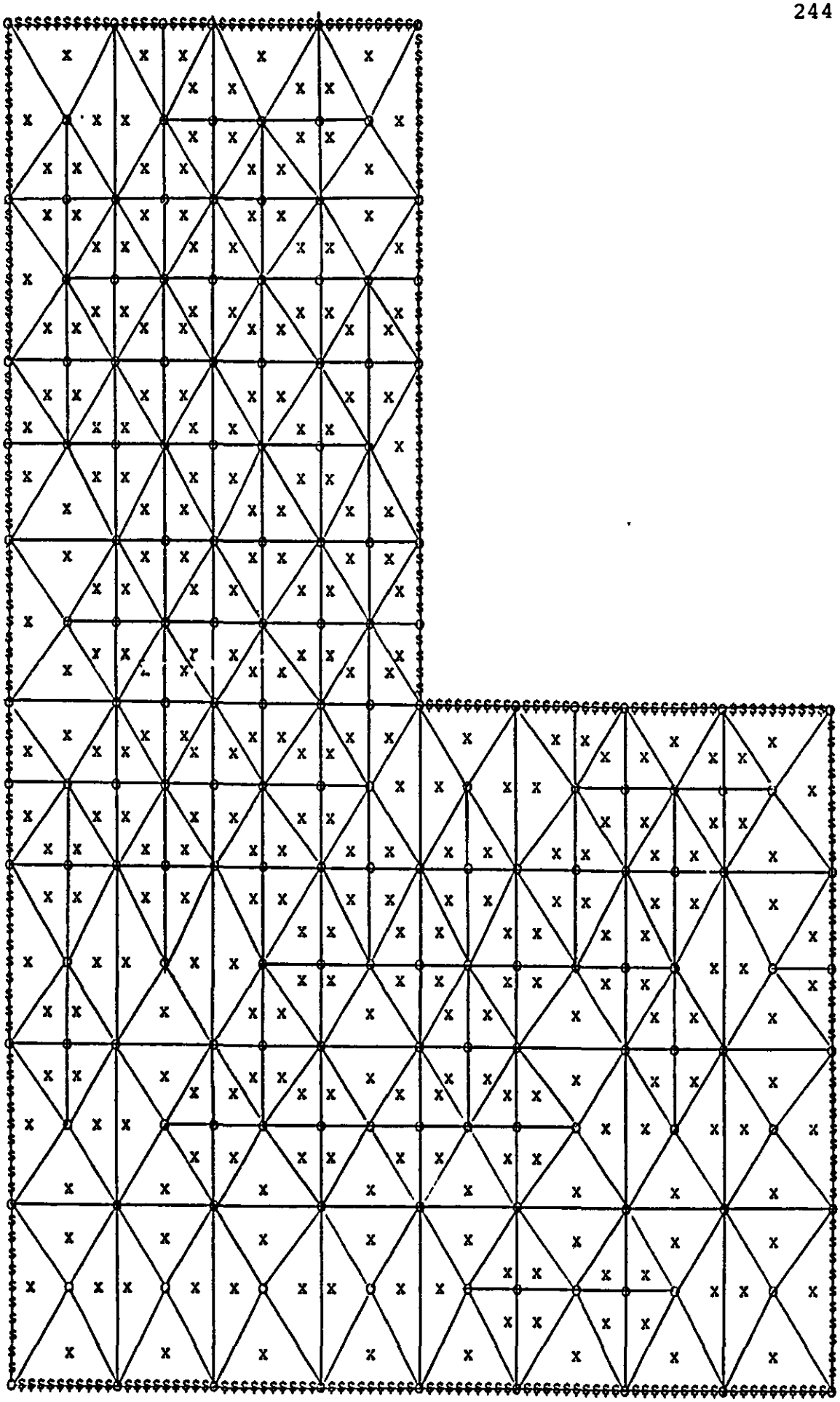


FIGURE 6.19

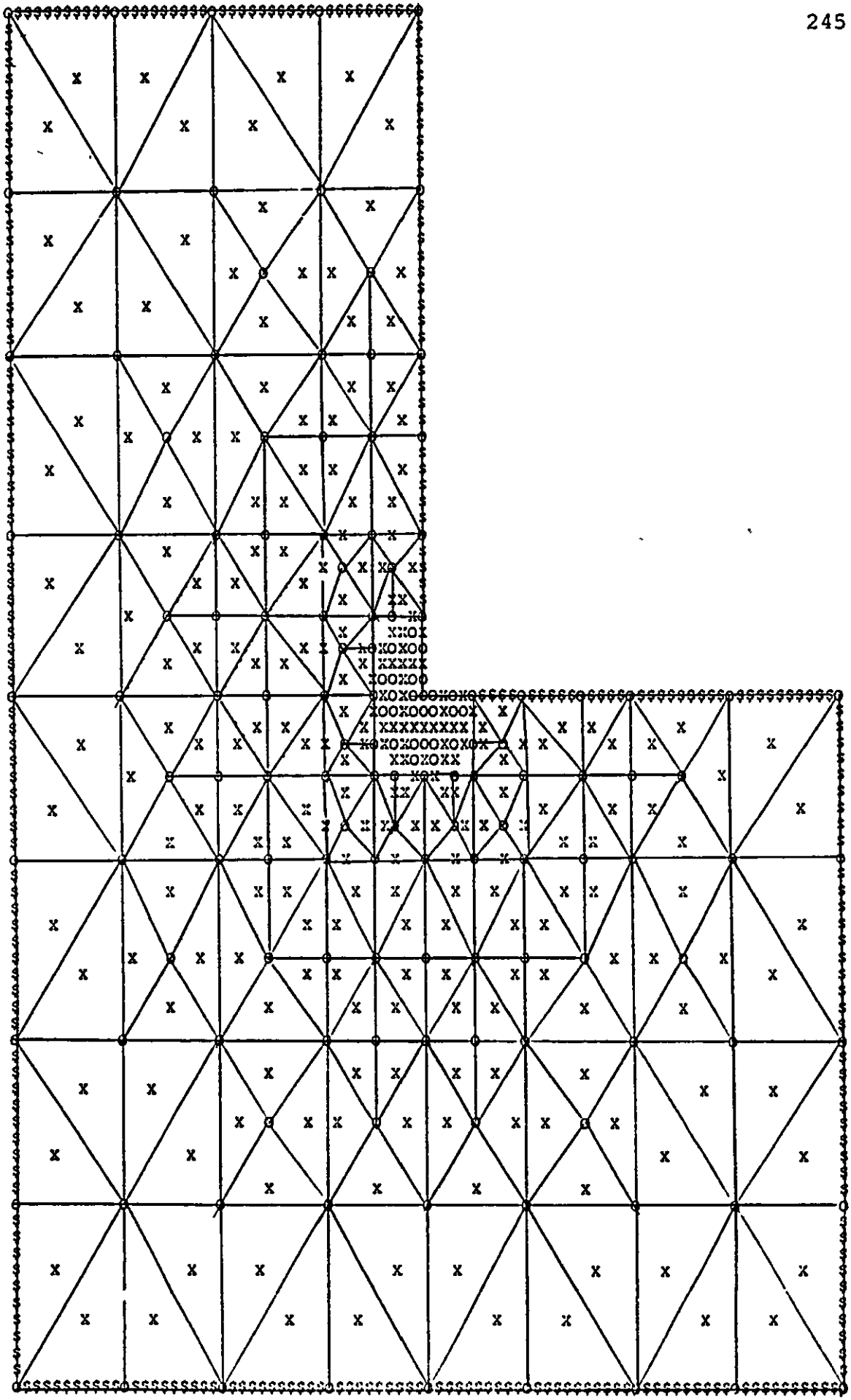


FIGURE 6.20

1.0000	.96425	.9425	.9206	.9056	.9007
1.000	.9700	.9426	.9205	.9059	.9009
	.9700	.9426	.9205	.9059	.9009
1.0000	.9687	.9399	.9165	.9009	.8955
1.0000	.9687	.9400	.9166	.9011	.8957
1.0000	.9687	.9400	.9166	.9012	.8957
1.0000	.9648	.9320	.9046	.8858	.8790
1.0000	.9648	.9321	.9048	.8860	.8792
1.0000	.9648	.9322	.9048	.8860	.8793
1.0000	.9585	.9189	.8841	.8583	.8484
1.0000	.9585	.9191	.8843	.8586	.8487
1.0000	.9585	.9191	.8843	.8586	.8487
1.0000	.9502	.9014	.8550	.8150	.7956
1.0000	.9503	.9015	.8552	.8154	.7961
1.0000	.9503	.9015	.8553	.8154	.7961

TABLE 6.8

At each point the numbers represent:

1	Equally distributed cubic triangular elements
2	Cubic triangular elements (dense around the singularities)
3	Conformal transformations method of Papamichael (1978).

1.0000	.9411	.8817	.8209	.7563	.6652	.4882	.3582	.2366	.1178	0.0000
1.0000	.9412	.8818	.8210	.7565	.6667	.4870	.3580	.2364	.1177	0.0000
1.0000	.9412	.8818	.8210	.7565	.6667	.4870	.3580	.2365	.1179	0.0000
1.0000	.9325	.8633	.7898	.7066	.6026	.4784	.3552	.2353	.1173	0.0000
1.0000	.9325	.8633	.7898	.7066	.6019	.4780	.3550	.2352	.1172	0.0000
1.0000	.9325	.8633	.7898	.7066	.6020	.4781	.3550	.2352	.1173	

continued.....

1.0000	.9254	.8487	.7672	.6673	.5758	.4644	.3488	.2324	.1161	0.0000
1.0000	.9254	.8487	.7671	.6771	.5756	.4642	.3489	.2322	.1161	0.0000
1.0000	.9254	.8487	.7671	.6772	.5757	.4642	.3486	.2323	.1161	0.0000
1.0000	.9204	.8388	.7528	.6604	.5605	.4538	.3426	.2292	.1148	0.0000
1.0000	.9204	.8387	.7528	.6604	.5604	.4536	.3425	.2291	.1148	0.0000
1.0000	.9204	.8387	.7528	.6604	.5604	.4537	.3425	.2291	.1148	0.0000
1.0000	.9175	.8331	.7450	.6516	.5523	.4475	.3386	.2270	.1138	0.0000
1.0000	.9175	.8331	.7449	.6515	.5522	.4474	.3385	.2269	.1138	0.0000
1.0000	.9175	.8331	.7449	.6515	.5522	.4474	.3385	.2269	.1138	0.0000
1.0000	.9166	.8313	.7425	.6488	.5497	.4455	.3372	.2262	.1136	0.0000
1.000	.9166	.8313	.7424	.6487	.5496	.4454	.3371	.2261	.1134	0.0000
1.0000	.9166	.8313	.7424	.6487	.5496	.4454	.3371	.2261	.1134	0.0000

TABLE 6.8: continued

1.0000	.9701	.9427	.9204	.9059	.9008					
1.0000	.9700	.9427	.9205	.9060	.9009					
1.0000	.9686	.9401	.9165	.9011	.8956					
1.0000	.9687	.9400	.9166	.9012	.8957					
1.0000	.9647	.9321	.9047	.8860	.8792					
1.0000	.9648	.9322	.9048	.8860	.8793					
1.0000	.9585	.9190	.8842	.8586	.8487					
1.0000	.9585	.9191	.8843	.8586	.8487					
1.0000	.9503	.9015	.8552	.8154	.7961					
1.0000	.9503	.9015	.8553	.8154	.7961					
1.0000	.9411	.8818	.8210	.7565	.6667	.4870	.3580	.2365	.1177	0.0000
1.0000	.9412	.8818	.8210	.7565	.6667	.4869	.3579	.2364	.1177	
1.0000	.9325	.8633	.7898	.7066	.6019	.4781	.3550	.2352	.1172	0.0000
1.0000	.9325	.8633	.7898	.7066	.6019	.4780	.3549	.2352	.1172	
1.0000	.9254	.8489	.7672	.6772	.5757	.4642	.3486	.2323	.1161	0.0000
1.0000	.9254	.8427	.7671	.6772	.5756	.4642	.3486	.2323	.1161	
1.0000	.9204	.8388	.7528	.6604	.5604	.4537	.3425	.2292	.1147	0.0000
1.0000	.9204	.8387	.7527	.6603	.5604	.4536	.3425	.2291	.1147	
1.0000	.9176	.8331	.7450	.6515	.5522	.4474	.3385	.2269	.1138	0.0000
1.0000	.9175	.8331	.7448	.6515	.5521	.4474	.3385	.2269	.1138	
1.0000	.9165	.8313	.7424	.6487	.5496	.4454	.3371	.2261	.1134	0.0000
1.0000	.9166	.8313	.7424	.6487	.5495	.4453	.3371	.2261	.1134	

TABLE 6.9

At each point the numbers represent:

1	Finite element method with (quadratic triangular elements)
2	Integral equation method of Symm (1973)

1.0000	.9701	.9427	.9204	.9059	.9008				
1.0000	.9699	.9429	.9201	.9055	.9004				
1.0000	.9686	.9401	.9165	.9011					
1.0000	.9686	.9398	.9162	.9007	.8952				
1.0000	.9647	.9321	.9047	.8860					
1.0000	.9647	.9319	.9043	.8855	.8787				
1.0000	.9585	.9190	.8842	.8586					
1.0000	.9584	.9188	.8838	.8579	.8481				
1.0000	.9503	.9015	.8552	.8154	.79489				
1.0000	.9502	.9012	.8547	.8145					
1.0000	.9503	.8818	.8210	.7565	.6667	4870	.3580	.2365	.1177
1.0000	.9501	.8816	.8207	.7560	.6654	4855	.3587	.2368	.1179
1.0000	.9325	.8633	.7898	.7066	.6019	.4781	.3550	.2352	.1172
1.0000	.9325	.8632	.7897	.7066	.6027	.4788	.3556	.2356	.1162
1.0000	.9254	.8487	.7672	.6772	.5757	.4642	.3486	.2323	.1161
1.0000	.9254	.8487	.7672	.6774	.5760	.4647	.3490	.2326	.1162
1.0000	.9204	.8388	.7528	.6604	.5604	.4537	.3425	.2292	.1147
1.0000	.9204	.8388	.7529	.6606	.5607	.4540	.3428	.2293	.1149
1.0000	.9176	.8331	.7450	.6515	.5522	.4474	.3385	.2269	.1138
1.0000	.9176	.8332	.7450	.6517	.5524	.4477	.3387	.2271	.1139
1.0000	.9165	.8313	.7424	.6487	.5496	.4454	.3371	.2261	.1134
1.0000	.9166	.8313	.7425	.6489	.5498	.4456	.3374	.2262	.1135

0.0000
0.0000
0.0000
0.0000
0.0000
0.0000

TABLE 6.10

At each point the numbers represent:

1	Finite element method with dense elements around singularity
2	Finite element method with equally distributed element

6.5 DISCUSSION

The importance of singularities in certain problems has over recent years caused a large number of special finite element adaptations to be proposed for their treatment. We have discussed here the mesh refinement approach which is one of the more successful treatments. The shortcoming of refining over the whole of the region R is that many mesh points remote from the singularity are introduced needlessly so that the resulting master matrix becomes unnecessarily large. Thus, in order to keep the total number of elements in the discretization to as small a number as possible for a given significant figure accuracy, we refine only in the neighbourhood of the singularity O.

The success of mesh refinement in improving the accuracy of the numerical solutions is evident.

Indeed with continued *mesh refinement* we reach the stage that the finite element solution is more accurate near the singularity than at nodes in the far region R remote from the singularity.

The effect of the singularity on the numerical solution has thus been neutralized by the mesh refinement and by using a space of piecewise polynomial function of higher degree.

For the problem of the type discussed, there seems to be little to choose between the finite element and the other specialised methods used to solve the problem, when comparing the accuracy obtained, but it is indeed the finite element method which is more general and the range of the problems to which it can be applied far wider.

CHAPTER 7

FINITE ELEMENT SOLUTION FOR NONLINEAR PARTIAL DIFFERENTIAL EQUATIONS

7.1 INTRODUCTION

In this chapter we look at the solution of two-dimensional nonlinear partial differential equations on general domains. A finite element solution for a given set of test problems will be obtained by TWODEPEP.

As basis functions we use a class of polynomials which are of:

i. degree two - with six node triangular elements

ii. degree three - with ten node triangular elements

iii. degree four - with fifteen node triangular elements

Newton's method which is described in detail in Chapter 2 is used to solve the resulting nonlinear system of finite element equations. The computational performance of the method is measured over a problem population of:

i. The minimal surface problem

ii. A set of nonlinear elliptic partial differential equations

iii. The highly nonlinear coupled elliptic semi-conductor problem.

We present here the solution of this set of problems using the different classes of polynomials as given above, and with a different number of elements for each problem.

7.2 THE NUMERICAL SOLUTION OF THE MINIMAL SURFACE EQUATION BY USING THE FINITE ELEMENT METHOD

In this section, the numerical solution of a second order, elliptic quasi-linear partial differential equation arising in two-dimensional magnetostatic field problems is considered (Plateau's Problem). The type of problems discussed are those arising, for example, in the design of particle accelerators where the desired magnetic field strengths are so large as to be principally in the domain of the nonlinear. The two-dimensional triangular element is used to solve the test problem.

The performance of the method is verified by numerically solving a sample problem and comparing the results according to the degree of the polynomials used and the number of triangular elements used in each type of polynomial. A graphical output of the solution is also presented.

7.2.1 FORMULATION OF THE PROBLEM

Consider a two-dimensional simply connected region R in the (x-y) plane with boundary ∂R.

Let $f(x,y)$ be a single-valued function defined on the boundary ∂R and γ represents the height of a given space curve above the point (x,y) on R. Let $u(x,y)$ represent the (single-valued) height, above the point (x,y) in R of the surface of minimal area through the given space curves then the problem in variational form, is that of finding a function $u(x,y)$ twice continuously differentiable in R satisfying,

$$u(x,y) = f(x,y), \quad \text{on} \quad \partial R , \tag{7.1}$$

and minimizing the surface area

$$A = \iint_R (1 + u_x^2 + u_y^2)^{\frac{1}{2}} \, dxdy \ . \tag{7.2}$$

The Euler-Lagrange equation corresponding to Equation (7.2)

$$r(1+q^2) - 2spq + t(1+p^2) = 0 \ , \tag{7.3}$$

where $p=\frac{\partial u}{\partial x}$, $q=\frac{\partial u}{\partial y}$, $r=\frac{\partial^2 u}{\partial x^2}$, $s=\frac{\partial^2 u}{\partial x \partial y}$ and $t=\frac{\partial^2 u}{\partial y^2}$, or in vector-operator notation the Euler Equation (7.2) takes the form,

$$\nabla.[\gamma(|\nabla u|^2)\nabla u] = 0 \ , \tag{7.4}$$

where,

$$\gamma[|\nabla u|^2] = (1+|\nabla u|^2)^{\frac{1}{2}} \ .$$

If the differentiations in Equation (7.4) are carried out, one obtains the more familiar form of the minimal surface equation as,

$$\left\{ \frac{(1+u_y^2)}{(1+u_x^2+u_y^2)^{3/2}} \right\} u_{xx} - \left\{ \frac{2u_x u_y}{(1+u_x^2+u_y^2)^{3/2}} \right\} u_{xy} + \left\{ \frac{(1+u_x^2)}{(1+u_x^2+u_y^2)^{3/2}} \right\} u_{yy} = 0. \tag{7.5}$$

In order to satisfy the requirement of the partial differential equation given by TWODEPEP which has the general form,

$$\frac{\partial}{\partial x}[\sigma_{xx}(x,y,u_x,u_y,u)] + \frac{\partial}{\partial y}[\sigma_{xy}(x,y,u_x,u_y,u)] = 0 \tag{7.6}$$

we can rewrite equation (7.5) in the form,

$$\frac{\partial}{\partial x}[u_x(1+u_x^2+u_y^2)^{-\frac{1}{2}}] + \frac{\partial}{\partial y}[u_y(1+u_x^2+u_y^2)^{-\frac{1}{2}}] = 0 \tag{7.7}$$

where

$$\frac{\partial}{\partial x}[u_x(1+u_x^2+u_y^2)^{-\frac{1}{2}}] = -\frac{1}{2} u_x(1+u_x^2+u_y^2)^{-3/2}(2u_x u_{xx}+2u_y u_{yx}) + u_{xx}(1+u_x^2+u_y^2)^{-\frac{1}{2}}$$

(7.8)

and $$\frac{\partial}{\partial y}[u_y(1+u_x^2+u_y^2)^{-\frac{1}{2}}] = -\frac{1}{2}u_y(1+u_x^2+u_y^2)^{-3/2}(2u_x u_{xy}+2u_y u_{yy})+u_{yy}$$
$$(1+u_x^2+u_y^2)^{-\frac{1}{2}} \qquad (7.9)$$

If we add (7.8) and (7.9) we get,

$$\left\{\frac{(1+u_y^2)}{(1+u_x^2+u_y^2)^{3/2}}\right\}u_{xx} - \left\{\frac{2u_x u_y}{(1+u_x^2+u_y^2)^{3/2}}\right\}u_{xy} + \left\{\frac{(1+u_x^2)}{(1+u_x^2+u_y^2)^{3/2}}\right\}u_{yy} = 0$$

which is similar to Equation (7.5).

7.2.2 TEST PROBLEM 1

Solve the nonlinear minimal surface problem (7.5) over the region $0<x<1$, $0<y<1$ with the boundary conditions,

$$u = [\cosh^2 y - x^2]^{\frac{1}{2}} \text{ on } \partial R. \qquad (7.10)$$

Figure (7.1) is shown below which illustrates Test Problem 1.

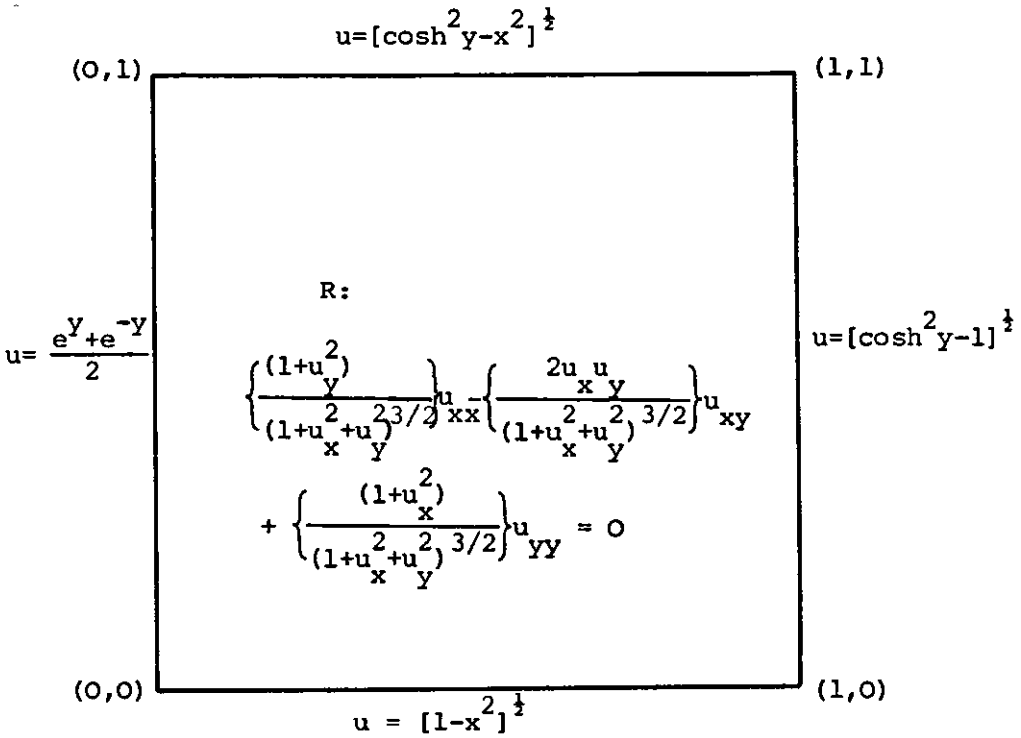


FIGURE 7.1: Test Problem 1

Results for Test Problem 1 are given in Table (7.1) which compare the discretization error obtained by the present Finite Element method with those of the Concus and Greenspan methods. The problem has the exact solution $u_E=(\cosh^2 y-x^2)^{\frac{1}{2}}$ which is used as an initial value to give a good estimate to solve the non-linear system generated by TWODEPEP.

x	y	F.E. Method Solution	Concus Solution	Greenspan Solution	Exact Solution
0.95	0.00	.32539	.31225	.31225	.31225
0.65	0.05	.76156	.76158	.76097	.76158
0.30	0.10	.95918	.95918	.95901	.95918
0.80	0.10	.60833	.60833	.60584	.60850
0.55	0.20	.84863	.84863	.84776	.84863
0.40	0.20	.93837	.93837	.93792	.93837
0.95	0.20	.37264	.37241	.36439	.37153
0.20	0.25	1.0118	1.0118	1.0118	1.0118
0.70	0.30	.77638	.77641	.77434	.77636
0.55	0.40	.93072	.93074	.92997	.93071
0.95	0.45	.56043	.56054	.55868	.56040
0.35	0.50	1.0719	1.0719	1.0720	1.0719
0.65	0.65	1.0310	1.0310	1.0311	1.0310
0.85	0.70	.92356	.92363	.92352	.92355
0.15	0.75	1.2860	1.2860	1.2863	1.2860
0.50	0.75	1.1942	1.1943	1.1948	1.1942
0.20	0.80	1.3224	1.3224	1.3228	1.3224
0.45	0.85	1.3083	1.3083	1.3088	1.3083
0.95	0.85	1.0058	1.0058	1.0060	1.0058
0.30	0.90	1.4013	1.4013	1.4017	1.4013
0.70	0.90	1.2505	1.2505	1.2509	1.2505
0.05	0.95	1.4854	1.4854	1.4855	1.4854
0.65	0.95	1.3366	1.3366	1.3368	1.3365

TABLE 7.1: Comparison of the discretization errors with those of the Concus and Greenspan methods

For 300 quadratic basis functions, the present method converges with an error $||u_N-u_E||_2=3.2453\times10^{-4}$, while with 300 cubic basis functions the error is $||u_N-u_E||_2=6.3584\times10^{-5}$. It is immediately apparent that the cubic basis function behaves better than the quadratic basis function for the same number of elements. Plots of the solution u showing the behaviour of the function over the given region are given in Figure (7.2) and (7.3). The promising results obtained for this minimal surface problem suggest the method discussed in this thesis is very useful for solving nonlinear partial differential equations.

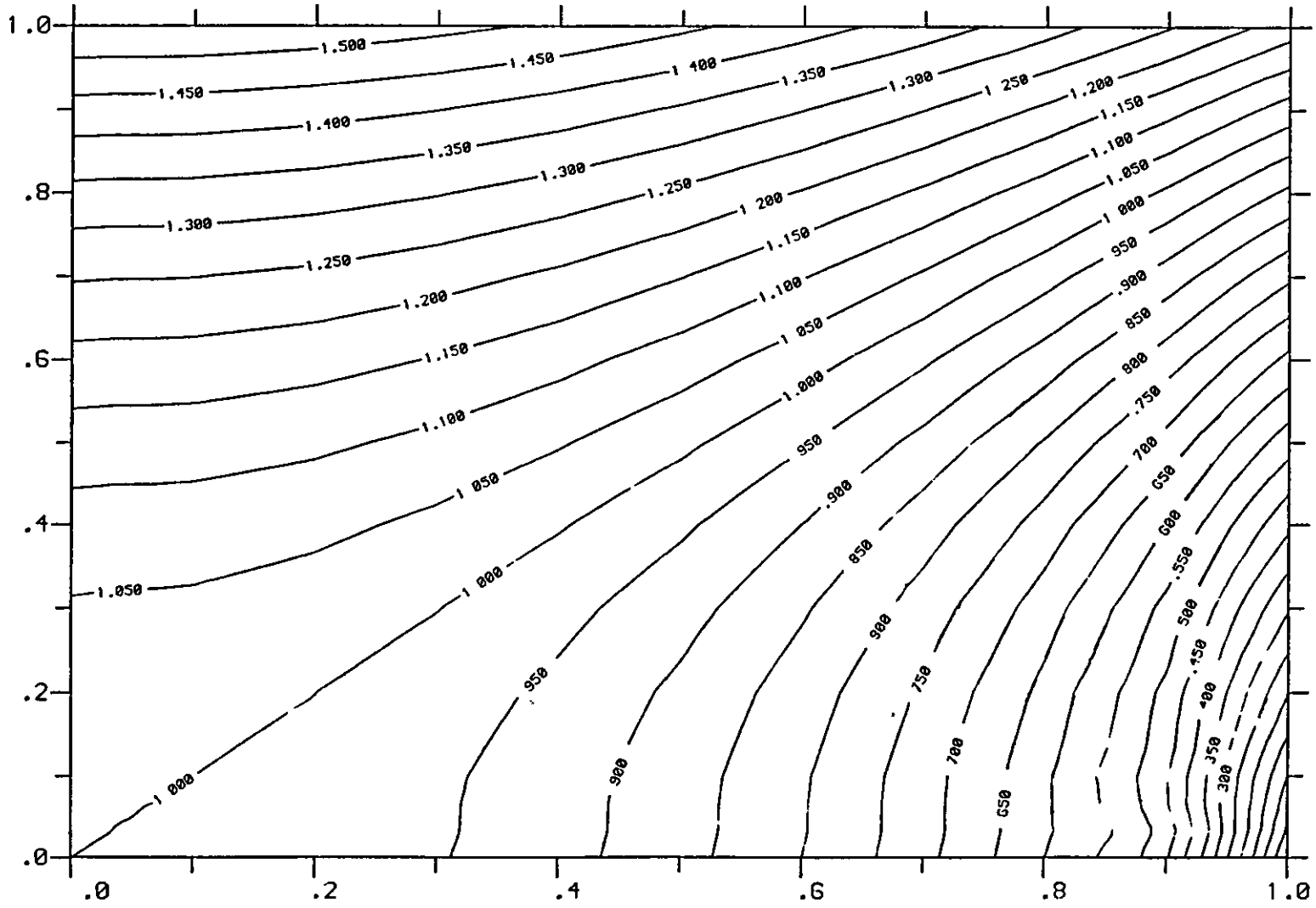

1.0
.8
.6
.4
.2
.0
.0
.2
.4
.6
.8
1.0
1.500
1.450
1.400
1.350
1.300
1.250
1.200
1.150
1.100
1.050
1.000
950
900
850
800
750
700
650
600
550
500
450
400
350
300

FIGURE 7.2

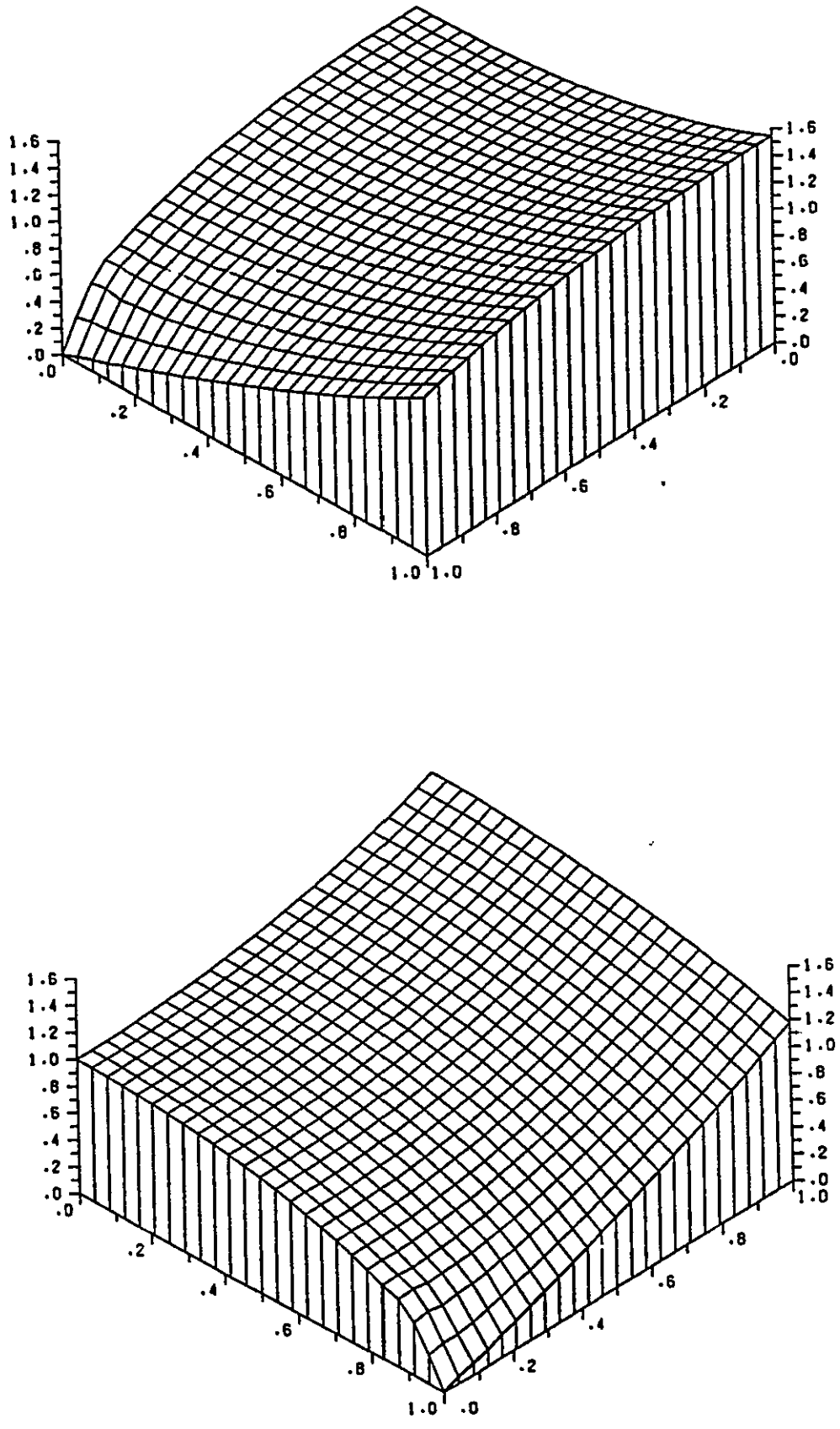


FIGURE 7.3: Isoparametric projection with different angles for the minimal surface problem

7.3 A POPULATION OF TWO DIMENSIONAL MILDLY NONLINEAR ELLIPTIC PARTIAL DIFFERENTIAL EQUATIONS

7.3.1 THE MODEL PROBLEM

We consider here the numerical approximation of two-dimensional mildly non-linear elliptic boundary value problems of the form,

$$Du \equiv f(u,u_x,u_y) \ , \quad (x,y) \in \partial R \ , \tag{7.11}$$

subject to the mixed type boundary conditions,

$$Lu = g(u) \ , \quad \text{on } \partial R \ . \tag{7.12}$$

Several authors G.F. ALIER [1971], E.N. HOUSTIS [1979], etc. have studied the solution of (7.11), (7.12) using finite-difference discretization. In this work we use the finite element method which is based on the class of piecewise polynomials approximation given in Section (7.1).

The procedure consists of the following components:

Elements:

A number of triangular elements are placed over the domain of the given problem.

Approximating Space:

A space of piecewise polynomials of second, third or fourth degree are used.

The resulting non-linear algebraic system is solved by Newton's method.

7.3.2 COMPUTATIONAL PERFORMANCE

We present the results of a population of second order mildly nonlinear equations which represents characteristics from both the

"real world" and "ideal" situations.

A summary of results for the 4 test problems, is presented as follows:

(a) Definition of the test problems

(b) Tables which give the solution u, the exact solution in the x,y dimensions and also the error norm 2 for a different polynomial order.

(c) Plots of the solution u showing the behaviour of the function u over the given region.

Mildly Nonlinear Elliptic Partial Differential Equations

Test Problem 1

Equation: $\nabla^2 u+(2-\sin y\cos x)u = U_x U_y$

Boundary condition: Dirichlet

Domain: Unit square

Exact solution: $u = \sin x\cos y$.

Comments: Non-constant coefficient, nonlinearities in the derivatives of the solution, nonhomogeneous boundary conditions.

Results: The tabulated results show the error, as the number of elements is subdivided, (the h version and also as the degree of the polynomial is increased, (the p version).

Order of the element \ Number of elements	50 Elements	70 Elements	100 Elements
Quadratic	8.0812×10^{-5}	3.4991×10^{-5}	2.0802×10^{-6}
Cubic	2.4988×10^{-6}	2.082×10^{-6}	2.056×10^{-6}
Quartic	2.0254×10^{-6}	2.0254×10^{-6}	-

TABLE 7.2

Figures (7.4) and (7.5) show the contour lines and the surface of the solution u for 100 cubic elements.

Test Problem 2

Equation: $\nabla^2 u - u(u_x+u_y)e^{-(x+y)}$,

Boundary condition: Dirichlet,

Domain: Unit square

Exact solution: e^{x+y}

Comments: Nonlinearities in the solution and the first derivatives of the solution, nonhomogeneous boundary conditions.

Results: The tables show the error as the number of elements is subdiviced (the h version) and also as the degree of the polynomial is increased (the p version).

Order of the element \ Number of elements	50 Elements	70 Elements	100 Elements
Quadratic	3.9549×10^{-4}	2.0377×10^{-4}	1.5582×10^{-4}
Cubic	2.4954×10^{-5}	2.2791×10^{-5}	2.2720×10^{-5}
Quartic	2.2507×10^{-5}	2.2507×10^{-5}	-

TABLE 7.3

Figures (7.6) and (7.7) show the contour lines and the surface of the solution u, for 70 quartic elements.

Test Problem 3

Equation: $\nabla^2 u = e^u + f(x,y)$

Boundary condition: Dirichlet, homogeneous,

Domain: Rectangle $(0,\frac{1}{2})\times(0,\frac{1}{4})$

Exact solution: $u = \sin 2\pi x \sin 4\pi y$.

Comments: Adapted from real world problem.

Results: The tables show the error, as the number of elements is subdivided, (the h version) and also as the degree of the polynomial is increased (the p version).

Order of the element \ Number of elements	50 Elements	70 Elements	100 Elements
Quadratic	3.2712×10^{-3}	2.1388×10^{-3}	7.3088×10^{-4}
Cubic	2.5027×10^{-4}	1.4850×10^{-4}	2.7599×10^{-5}
Quartic	1.5412×10^{-5}	7.6831×10^{-6}	-

TABLE 7.4

Figures (7.8) and (7.9) show the contour lines and the surface of the solution u, for 70 quartic elements.

Test Problem 4

Equation: $$\nabla^2 u - \frac{u}{u+10} = f(x,y) \ ,$$

Boundary condition: Dirichlet

Domain: Unit square

Exact solution: $\cos\beta y + \sin\beta(x-y)$

(a) $\beta=\pi$, (b) $\beta = 8$.

Comments: The value of f(x,y) is determined so that the given true solution is correct. Nonhomogeneous boundary conditions, oscillatory solution.

Results: The tables show the errors as the number of elements is subdivided (the h version) and also as the degree of the polynomial is increased (the p version) with $\beta=\pi$.

Order of the element \ Number of elements	23 Elements	46 Elements	69 Elements
Quadratic	1.269×10^{-2}	4.6156×10^{-3}	1.7081×10^{-3}
Cubic	7.9156×10^{-4}	2.5092×10^{-4}	5.842×10^{-5}
Quartic	6.1906×10^{-5}	2.1912×10^{-5}	-

TABLE 7.5

Figures (7.10) and (7.11) show the contour lines and the surface of the solution y for 69 cubic elements.

The object of the present set of test problems is to show how the finite element method when supplemented by adequate quadratic, cubic or the highly accurate quartic basis functions can produce highly accurate results and present no difficulty in dealing with mildly nonlinear elliptic partial differential equations.

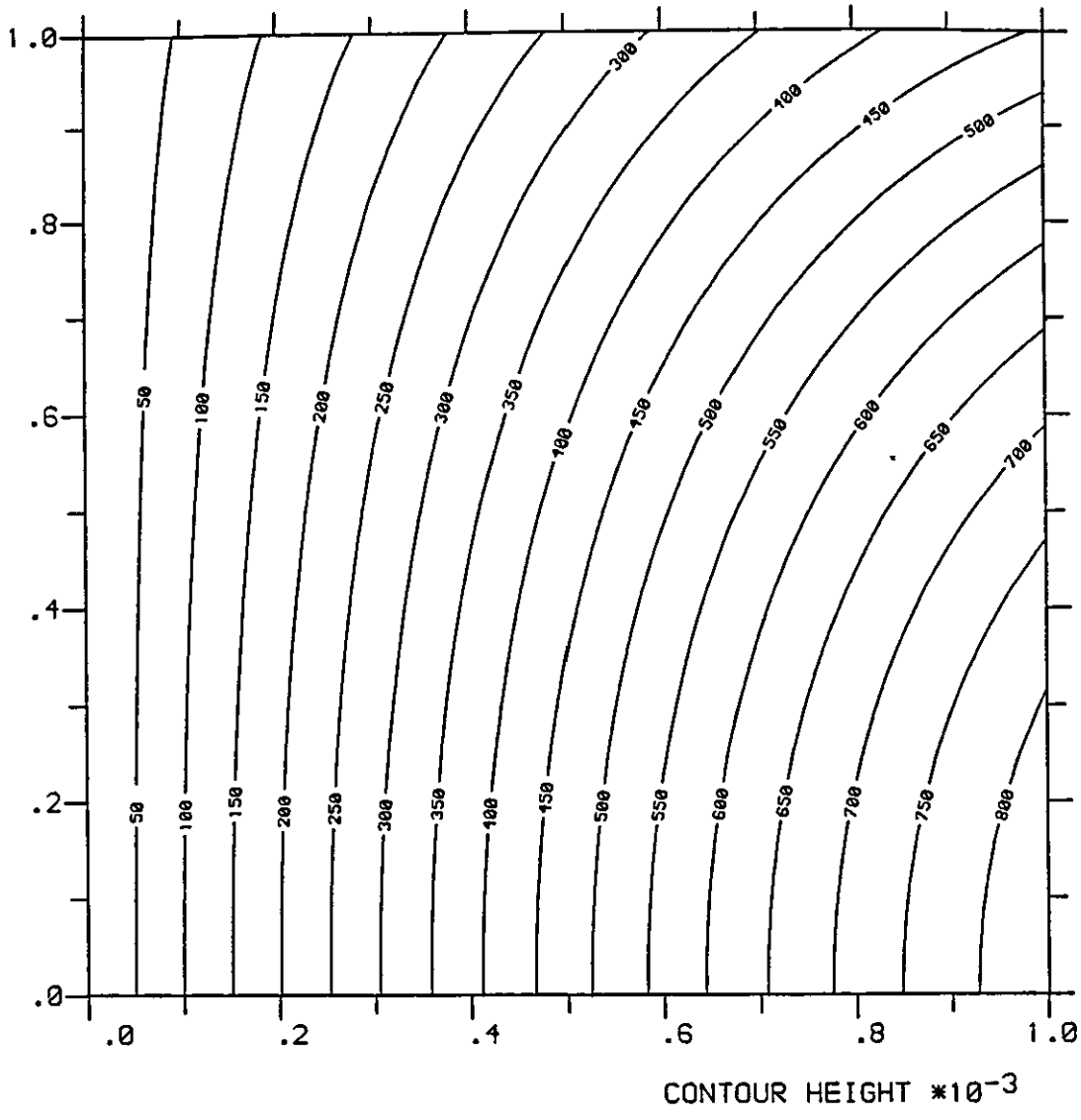

1.0
.8
.6
.4
.2
.0
.0
.2
.4
.6
.8
1.0
CONTOUR HEIGHT *10-3
50
100
150
200
250
300
350
100
150
500
550
600
650
700
750
800
300
400
450
500

FIGURE 7.4

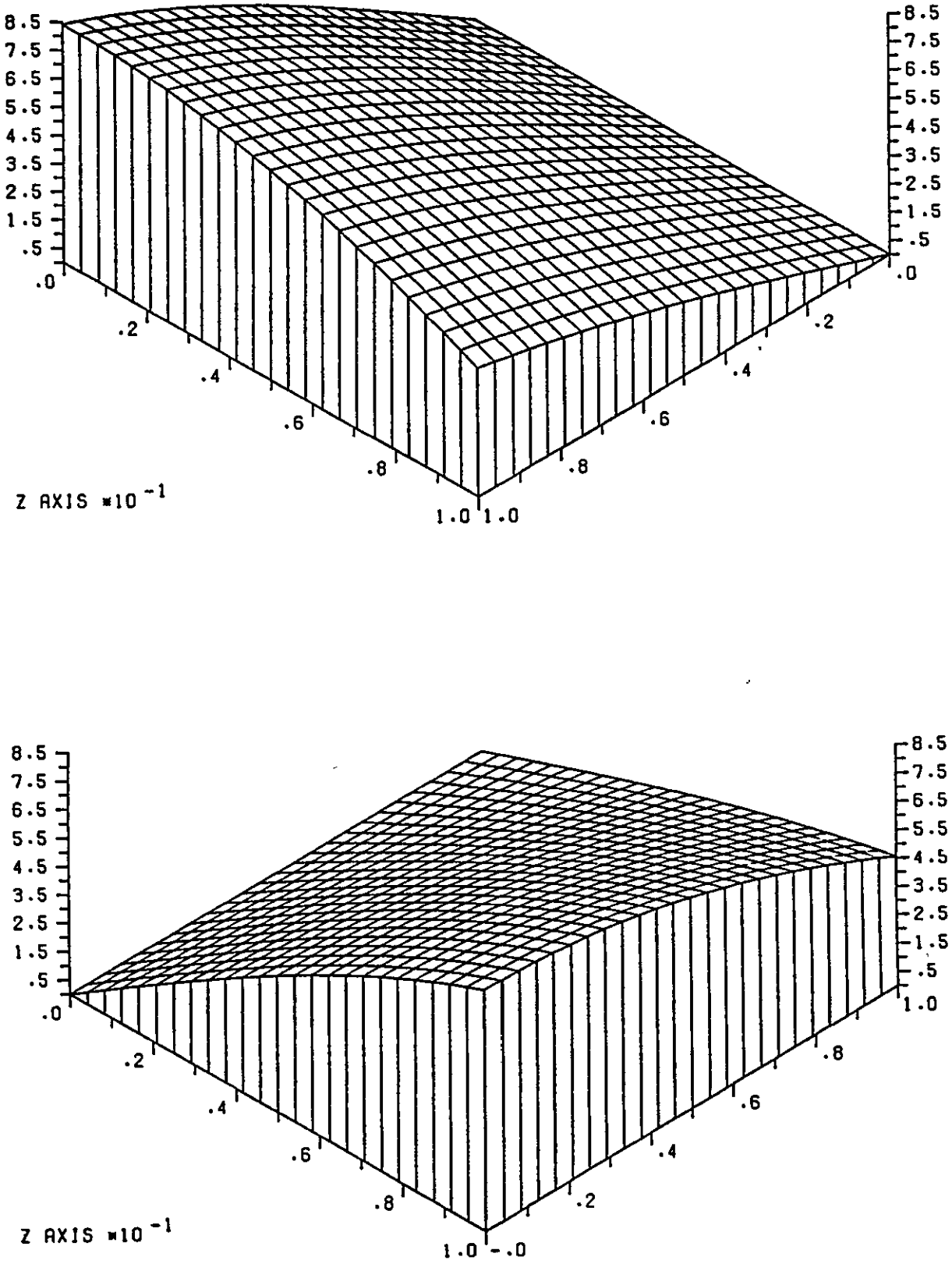


FIGURE 7.5: Isoparametric projection with different angle for Problem 1

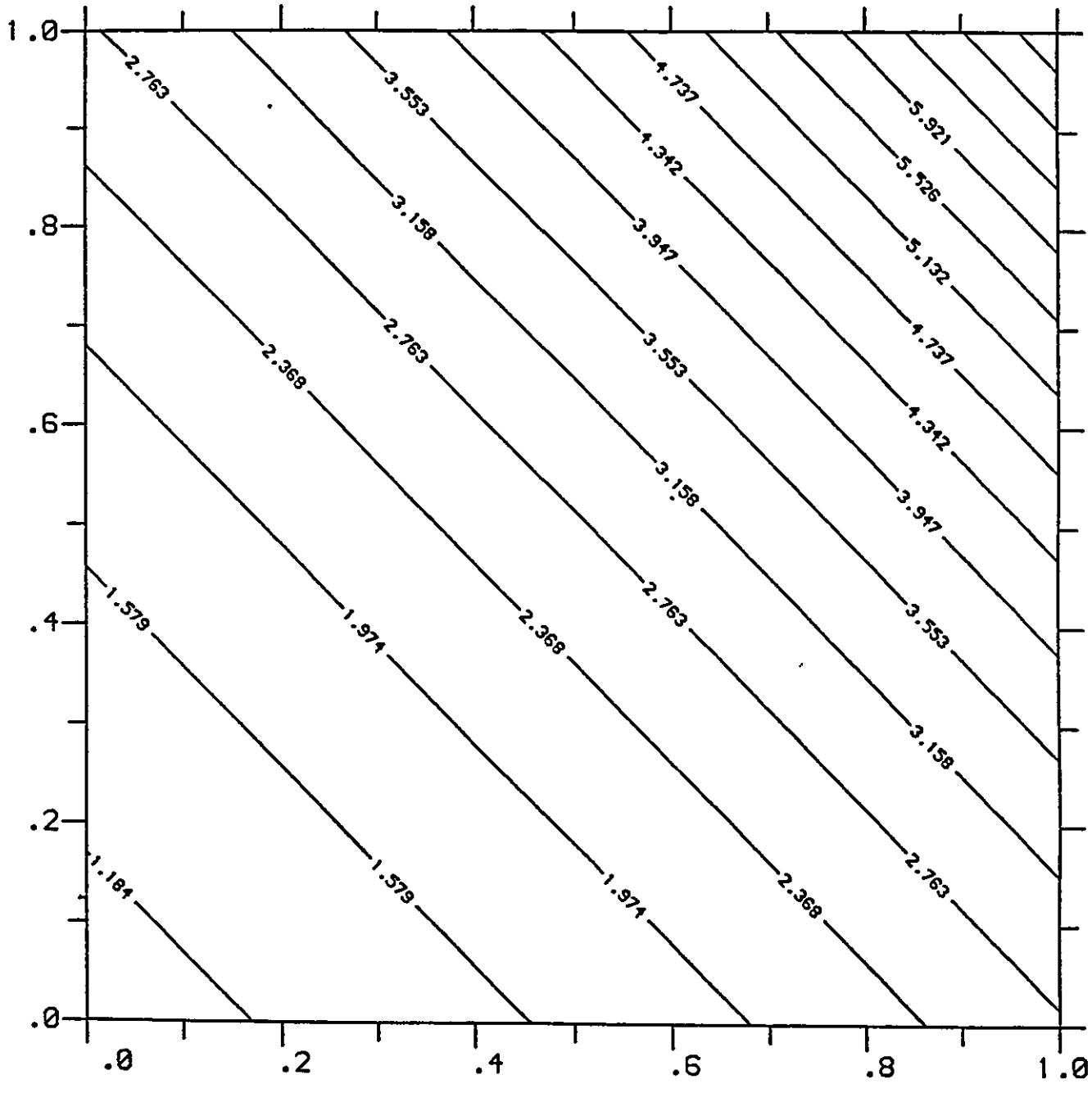

1.0
.8
.6
.4
.2
.0
.0
.2
.4
.6
.8
1.0
2.763
3.553
1.737
5.921
1.342
5.526
3.158
3.947
5.132
2.763
3.553
1.737
2.368
4.342
3.158
3.947
2.763
3.553
1.579
1.971
2.368
3.158
2.763
1.184
1.579
1.971
2.368
2.763

FIGURE 7.6

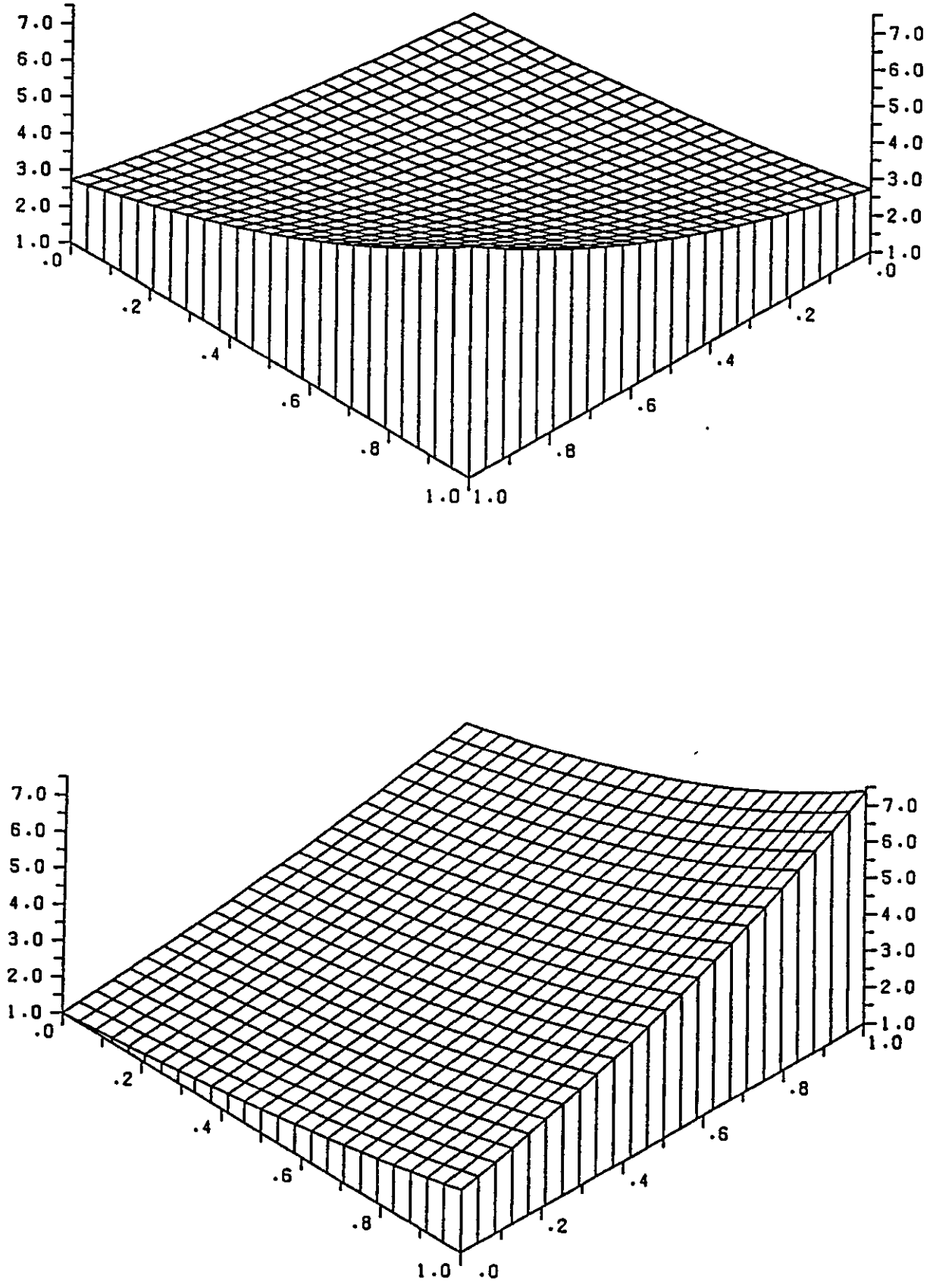


FIGURE 7.7: Isoparametric projection with different angles of Problem 2

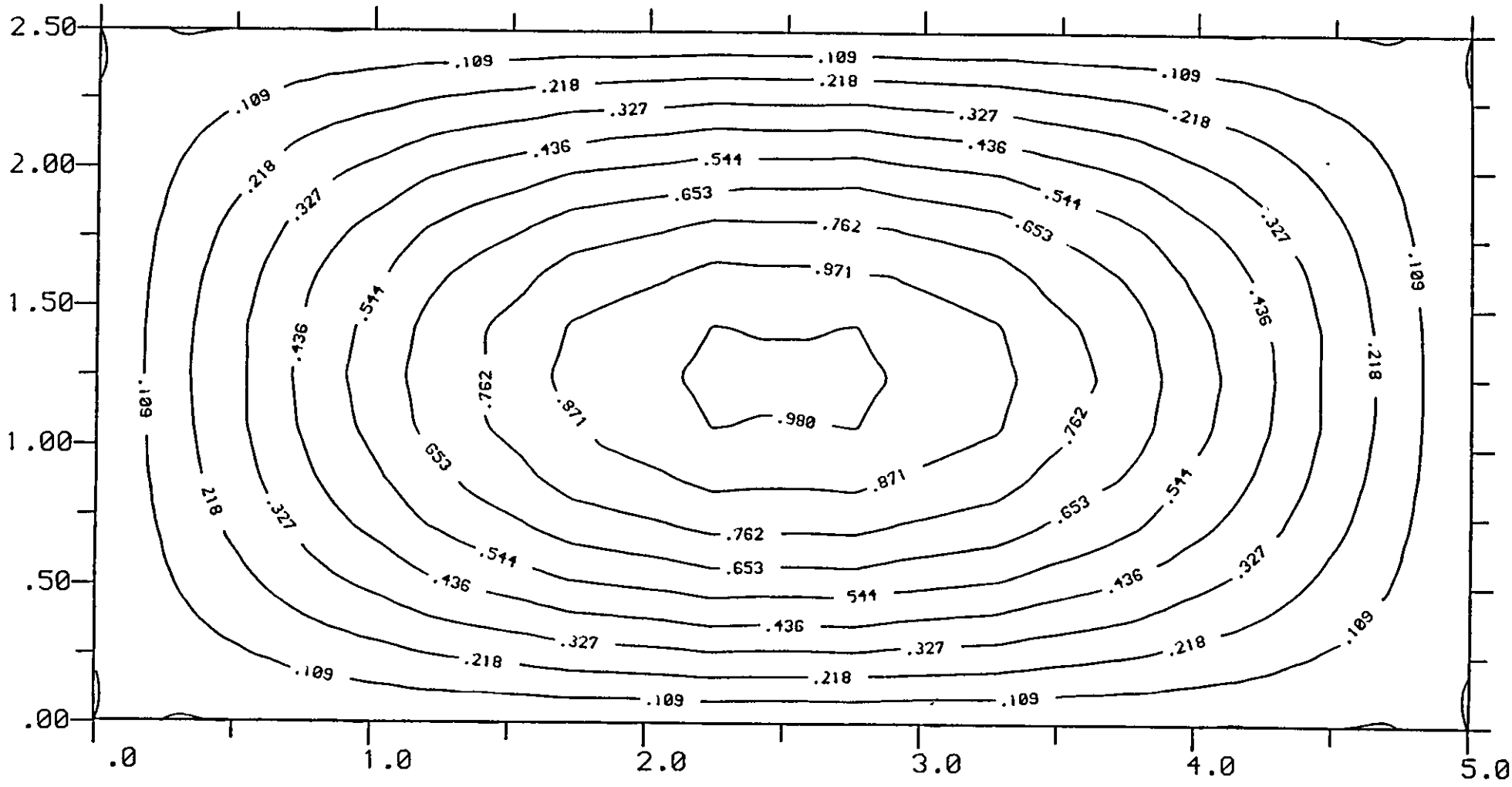

2.50
2.00
1.50
1.00
.50
.00
.0
1.0
2.0
3.0
4.0
5.0
.109
.218
.327
.436
.544
.653
.762
.871
.980
X AXIS *10^-1
Y AXIS *10^-1

FIGURE 7.8

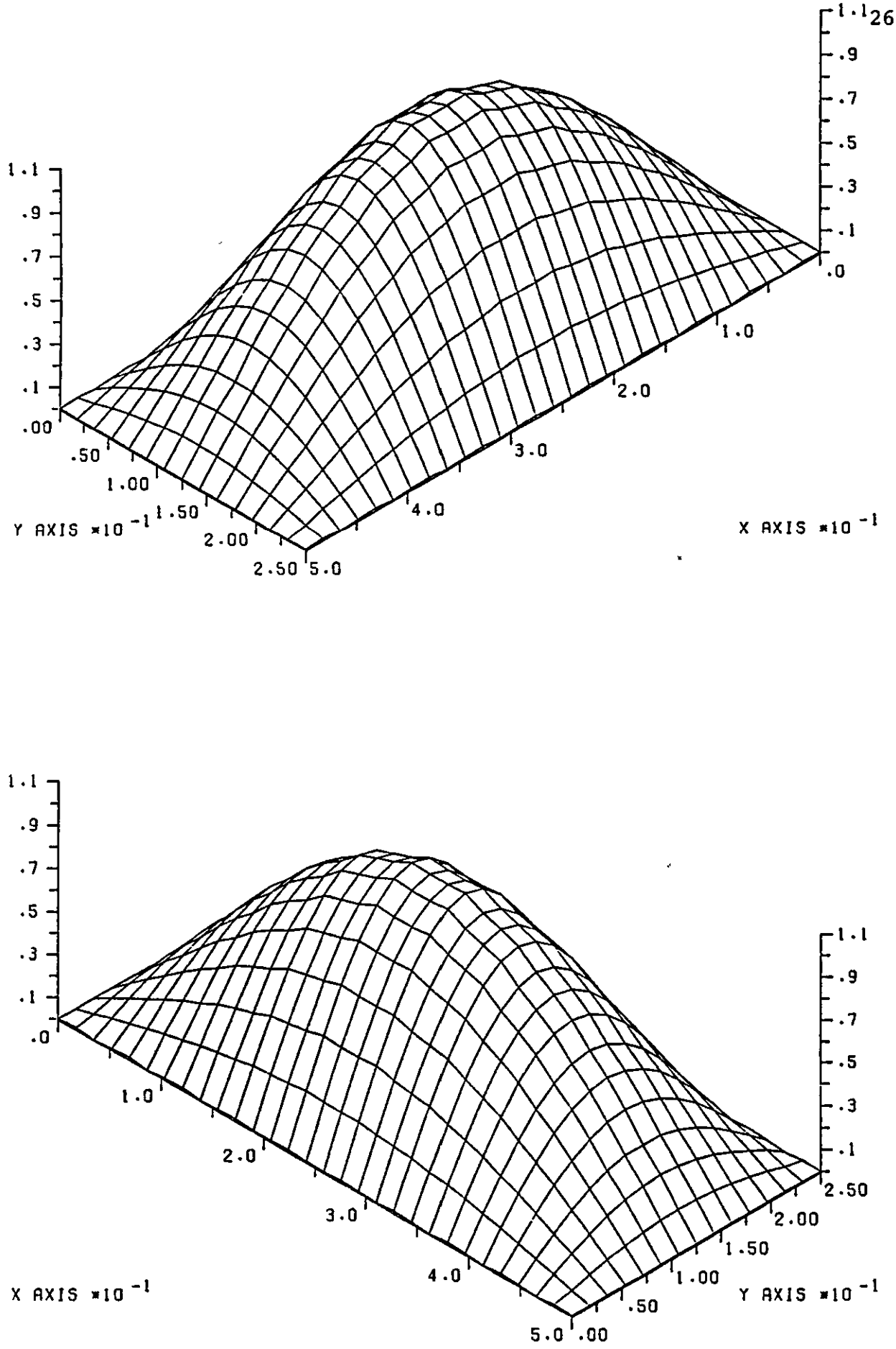


FIGURE 7.9: Isoparametric projection with different angles for Problem 3

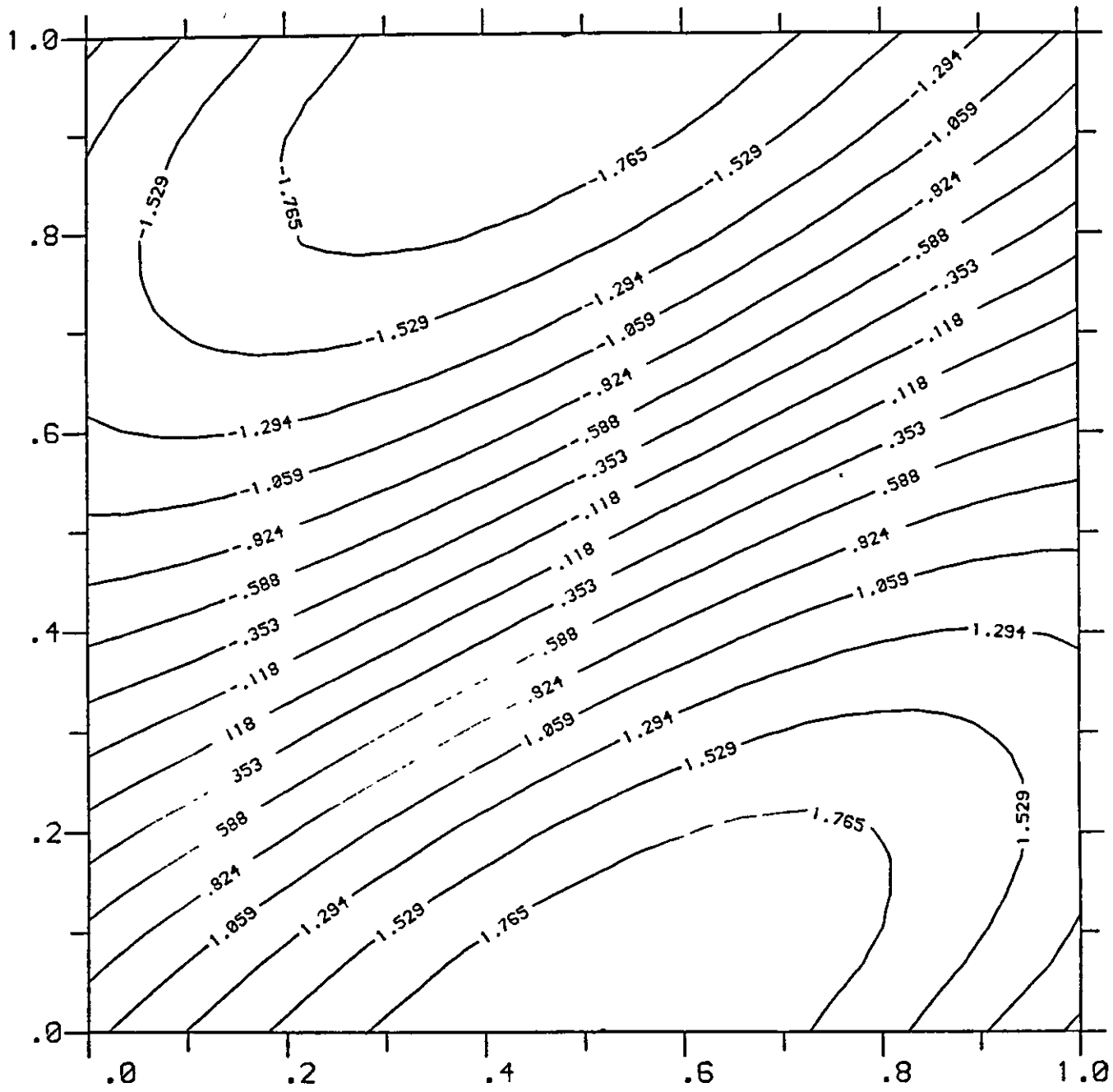

1.0
.8
.6
.4
.2
.0
.0
.2
.4
.6
.8
1.0
-1.529
-1.765
-1.765
-1.529
-1.529
-1.294
-1.059
-.824
-.588
-.353
-.118
.118
.353
.588
.824
1.059
1.294
1.529
1.765

FIGURE 7.10

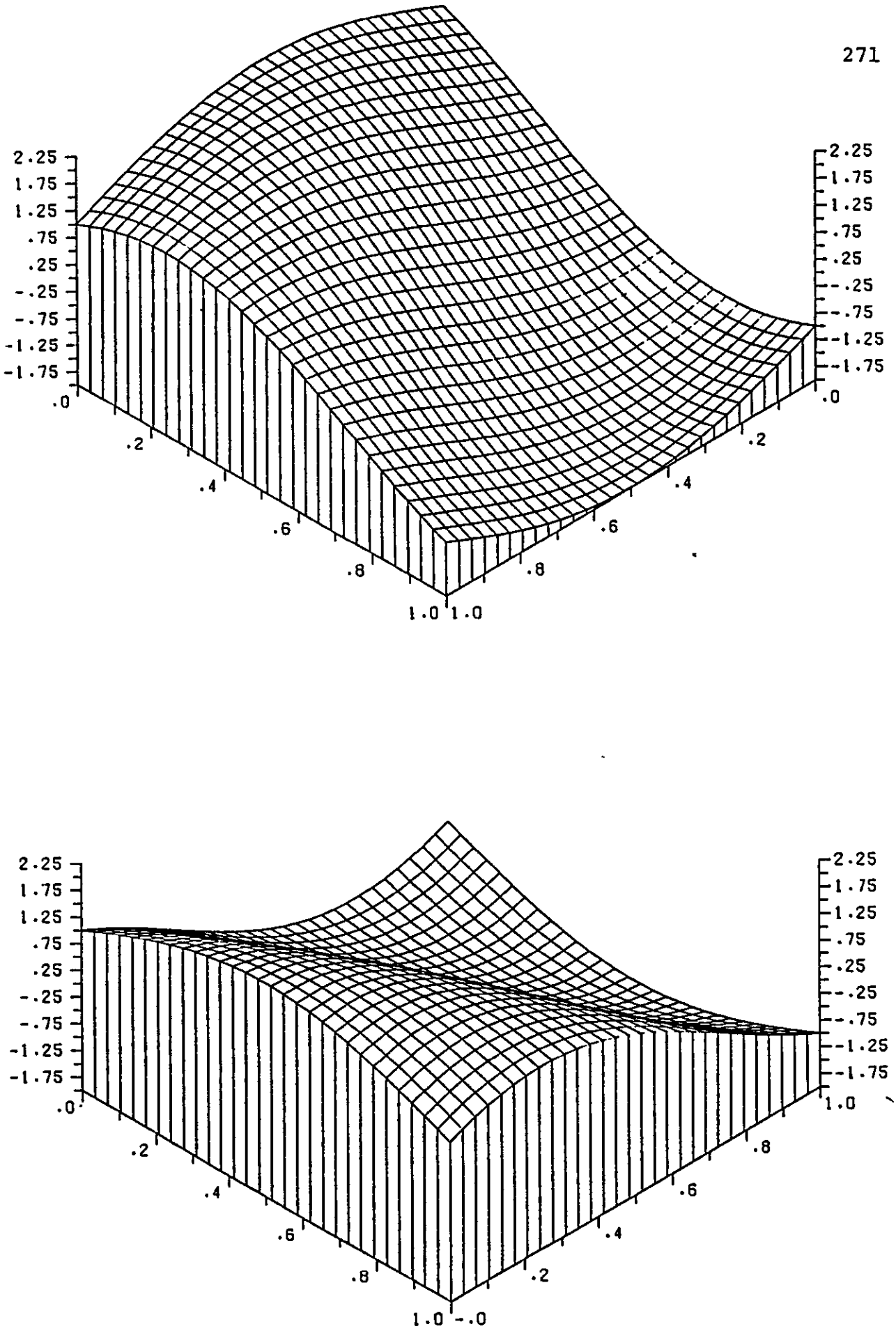


FIGURE 7.11: Isoparametric projection with different angles for Problem 4

7.4 A SEMI-CONDUCTOR PROBLEM

A highly non-linear coupled elliptic partial differential equation which is one of the most important problems for the scientific computing community is that which models the intrinsic behaviour of semi-conductor devices.

These equations may be written in the form of a two-dimensional model,

$$-\nabla^2 u + n(x,y) - p(x,y) = k(x,y) \ , \tag{7.13}$$

$$\nabla J_n = 0 \ , \tag{7.14}$$

$$\nabla J_p = 0 \ , \tag{7.15}$$

where $u(x,y)$ is the electrostatic potential, $n(x,y)$, $p(x,y)$ are the electron and hole densities respectively, and $k(x,y)$ is the doping profile (impurity concentration), J_n and J_p are, respectively, the electrons and holes current densities.

They are further specified in the usual drift-diffusion equations by

$$J_n = -M_n(x,y)\ n(x,y)\nabla u + D_n(x,y)\nabla n \ , \tag{7.16}$$

$$J_p = -M_p(x,y)\ p(x,y)\nabla u - D_p(x,y)\nabla p \ . \tag{7.17}$$

The current densities J_n and J_p are composed of a drift component, $-M_n n\nabla u$ or $-M_p p\nabla u$, and a diffusion component, $D_n\nabla n$ or $D_p\nabla p$. Assuming the validity of the Einstein relation, M=D and no recombination occurs, Equations (7.16) and (7.17) can be rewritten as,

$$J_n = M_n[-n\nabla u+\nabla n] = -M_n e^{u-v}\nabla v \ , \tag{7.18}$$

$$J_p = M_p[-p\nabla u-\nabla p] = -M_p e^{w-u}\nabla w \ , \tag{7.19}$$

where $n(x,y) = e^{u-v}$, and $p = e^{w-u}$ define implicitly the quasi-Fermi potential levels v and w for electrons and holes, respectively. Using

this change of variables in Equations (7.13) to (7.15) leads to the equations,

$$-\nabla^2 u + e^{u-v} - e^{w-u} = k(x,y) \ , \tag{7.20}$$

$$-\nabla(M_n e^{u-v}\nabla v) = 0 \ , \tag{7.21}$$

$$-\nabla(M_p e^{w-u}\nabla w) = 0 \ . \tag{7.22}$$

If we compare the original equations with Equations (7.20) to (7.22), then it is clear that both the Einstein relation and the changing of the variable have significantly reduced the degree of difficulty of the original problem.

In Equation (7.21) if we know u then we have a self-adjoint elliptic PDE in v which perhaps can be easily solved. But in (7.14) if u is known we still have the ∇n term in addition to the $\nabla^2 n$ term. A similar problem occurs in Equation (7.15).

This formulation of the semi-conductor equations are highly non-linear and computational difficulties may be encountered when Equations (7.20) to (7.22) are solved numerically by TWODEPEP. The first of these difficulties is the very large dynamic range of the solutions.

Another difficulty is the fact that for very small devices the validity of the Einstein relation for high electric field strengths is questionable. Hence Equations (7.21) and (7.22) may not apply.

Problem Definition and Results

For given mobility coefficients, M_n, M_p and diffusion coefficients D_n, D_p Equations (7.20) to (7.22) are posed on the unions of rectangular regions as shown in Figure (7.12). Dirichlet boundary conditions are

imposed on the gate (G), source (S), drain (D) and substrate (B) by the applied bias voltage; Neumann boundary conditions are assumed at the unspecified edges.

No attempt is made to solve the Poisson equation for the potential distribution inside the gate, but rather an approximate boundary condition along the interface is made.

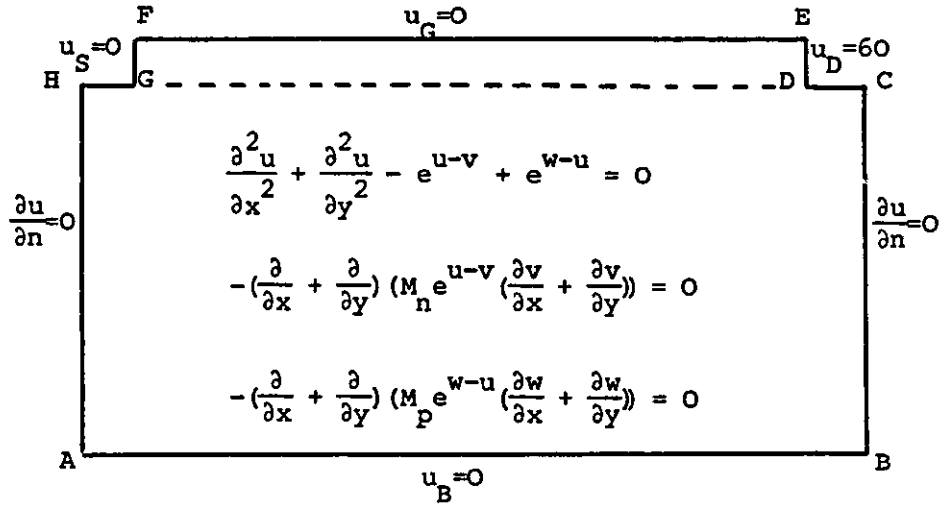


FIGURE 7.12

Figures (7.13) and (7.14) summarise the results of the finite element solution for the electrostatic potential u in the region,

$$0 \leq x \leq 2.8, \quad 0 \leq y \leq .92,$$

with the boundary condition given in Figure (7.12). 150 quadratic triangular elements were used to solve this highly non-linear problem.

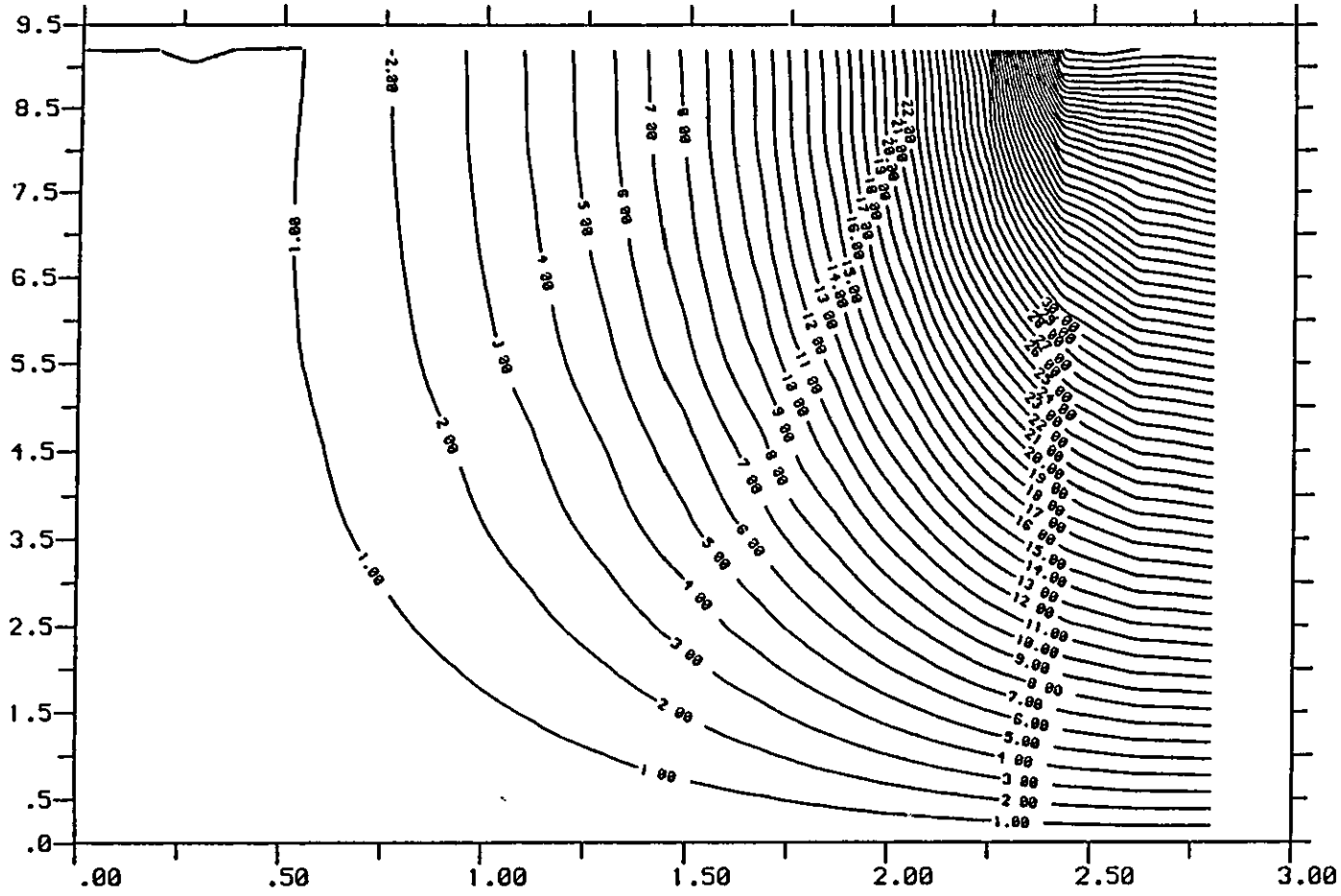


FIGURE 7.13

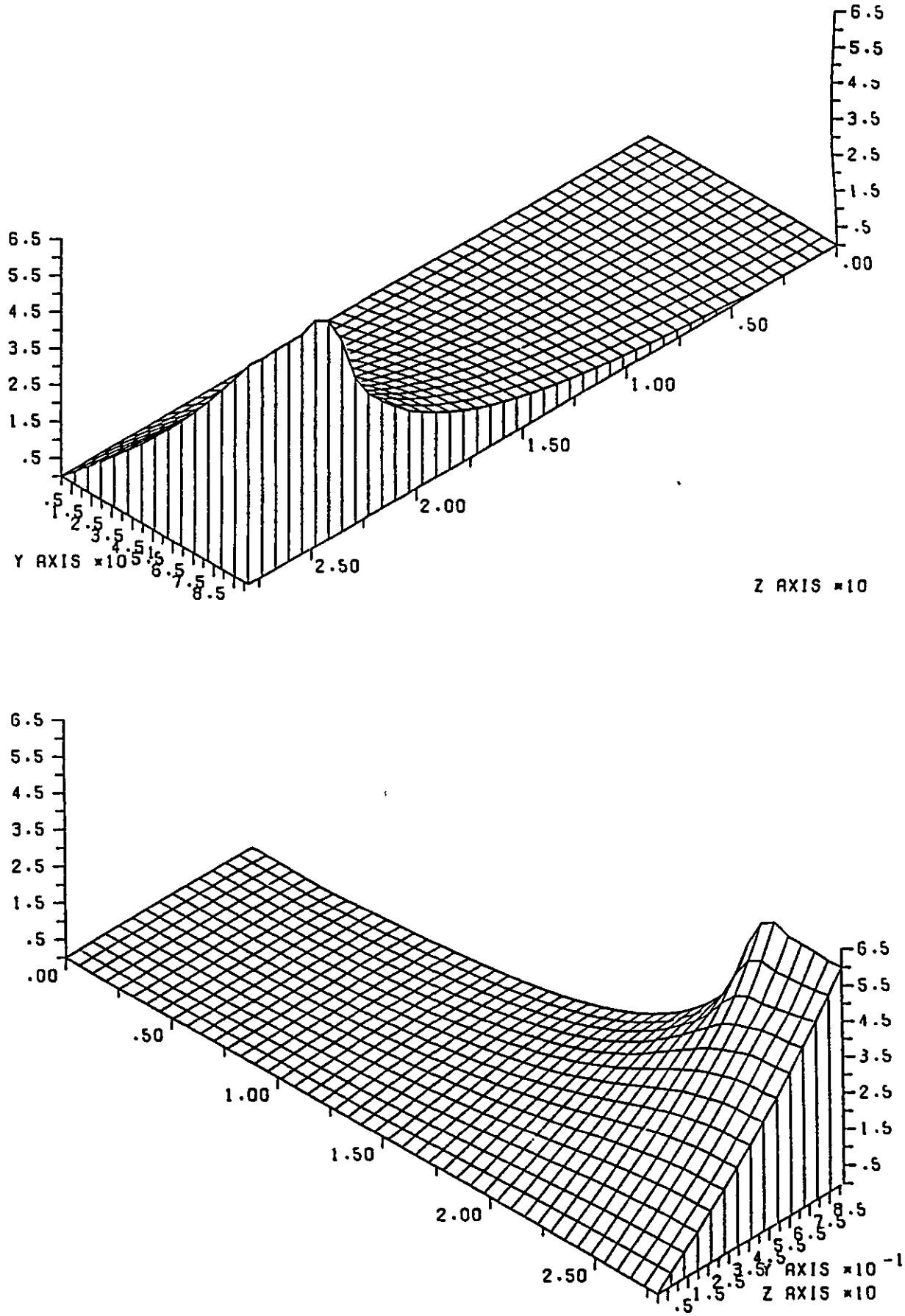


FIGURE 7.14: Isoparametric projection with different angles for the semi-conductor problem

CHAPTER 8

APPLICATION OF THE FINITE ELEMENT METHOD TO THE SOLUTION OF COMPLEX PROBLEMS

8.1 INTRODUCTION

In this chapter the finite element solution of a class of partial differential equations for the following complex problems will be presented. *Firstly*, from Elasticity we will consider the numerical solution of the Biharmonic problem of a simply supported rectangular plate in the two dimensional plane. *Secondly*, two problems of viscous flow in fixed regions, namely the potential flow around an elliptic obstacle in a channel, and that of inviscid laminar flow in a channel past a disc. *Thirdly*, we look at the solution of the two dimensional unsteady incompressible Navier Stokes equations in a rectangular region, and finally we will consider the solution of the eigenvalue problem for the Laplace Operator in an L-shaped region.

8.2 THE BIHARMONIC EQUATION

In this section we shall consider the numerical approximation of the 4th order linear partial differential equation

$$\frac{\partial^4 u}{\partial x^4} + 2\frac{\partial^4 u}{\partial x^2 \partial y^2} + \frac{\partial^4 u}{\partial y^4} = f(x,y) \ , \tag{8.1}$$

in which $f(x,y)$ is some specified function of x and y.

This equation is termed the Biharmonic equation, and is well known in many branches of Mathematical Physics - notably Hydrodynamics, where it governs the slow two-dimensional motion of a viscous fluid, u representing the stream-function (usually denoted by ψ), and f being zero when the body forces are conservative. Biharmonic equations also appear in the theories of extension and of flexure for flat elastic plates in Elasticity.

The basic laws of elasticity corresponding to the general conservation principles are the equations of equilibrium and compatibility. In the general application of these equations then to relate the stress and the strain in an elastic body, it is convenient to define a stress function ϕ according to

$$\phi_{xx} = \sigma_x \ ,$$

$$\phi_{yy} = \sigma_y \ ,$$

and

$$\phi_{xy} = \rho_{xy} \ ,$$

where σ_x and σ_y are the normal stresses in the x and y directions, respectively, and ρ_{xy} is the corresponding shear stress.

Under static conditions, equilibrium and compatibility then leads to the biharmonic equation which takes the form (8.1). Equation (8.1) is an elliptic equation analogous to Laplace's equation in other systems, if f=0.

In general, we can classify the Biharmonic equations into three classes of problems:

Static Beam: $$\frac{\partial^4\phi}{\partial x^4} + 2\frac{\partial^4\phi}{\partial x^2\partial y^2} + \frac{\partial^4\phi}{\partial y^4} = 0 ,$$

Beam Vibration: $$\frac{\partial^4\phi}{\partial x^4} + 2\frac{\partial^4\phi}{\partial x^2\partial y^2} + \frac{\partial^4\phi}{\partial y^4} = k\frac{\partial^2\phi}{\partial t^2} ,$$

and Loaded Beam: $$\frac{\partial^4\phi}{\partial x^4} + 2\frac{\partial^4\phi}{\partial x^2\partial y^2} + \frac{\partial^4\phi}{\partial y^4} = f(x,y) .$$

A RECTANGULAR PLATE PROBLEM

We consider a rectangular plate in the two dimensional plane bounded by the lines $0\leq x\leq a$, $0\leq y\leq b$.

A load $q=q(x,y)$ is assumed to be distributed over the surface of the plate.

Then, the differential equation for the deflection $u=u(x,y)$ is found to be

$$\frac{\partial^4 u}{\partial x^4} + 2\frac{\partial^4 u}{\partial x^2\partial y^2} + \frac{\partial^4 u}{\partial y^4} = \frac{q}{D} , \qquad (8.2)$$

where D is a physical quantity called the "flexural rigidity of the plate".

If the edges of the plate are simply supported, the boundary conditions are,

$$u = 0, \quad \frac{\partial^2 u}{\partial x^2} = 0, \quad \text{for } x=0 \text{ and } x=a \ ,$$
$$u = 0, \quad \frac{\partial^2 u}{\partial y^2} = 0, \quad \text{for } y=0 \text{ and } y=b \ , \tag{8.3}$$

We will consider the case in which $q=q_0 \sin \frac{\pi x}{a} \sin \frac{\pi y}{b}$, where q_0 denotes the intensity of the load at the centre of the plate.

It is clear that all the boundary conditions (8.3) are satisfied if we take for the deflection, the expression,

$$u = c \sin \frac{\pi x}{a} \sin \frac{\pi y}{b} \ , \tag{8.4}$$

in which c is a constant that must be chosen so that u will satisfy equation (8.2) with $q=q_0 \sin \frac{\pi x}{a} \sin \frac{\pi y}{b}$, if we substitute (8.4) into equation (8.2) we find that,

$$\pi^4 (\frac{1}{a^2} + \frac{1}{b^2})^2 c = \frac{q_0}{D} \ ,$$

solving for c we find that the solution of this special problem is given by,

$$u = q_0 (\pi^4 D)^{-1} (\frac{1}{a^2} + \frac{1}{b^2})^{-2} \sin(\frac{\pi x}{a}) \sin(\frac{\pi y}{b}) \ . \tag{8.5}$$

Now, the fourth order plate problem (8.2) can be solved by TWODEPEP by defining,

$$u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \ , \tag{8.6}$$

Then equation (8.2), becomes a system of two simultaneous second order equations,

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = v \ ,$$
$$\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} = \frac{q}{D} \ , \tag{8.7}$$

and the boundary conditions become

$$u = 0, \quad v = 0, \quad \text{on } x=0 \text{ and } x=a \ ,$$
$$u = 0, \quad v = 0 \ , \quad \text{on } y=0 \text{ and } y=b \ , \tag{8.8}$$

We solve the set of two simultaneous equations (8.7) and the given boundary conditions (8.8) with a=1, b=1. The input to the preprocessor should be manipulated into a symmetric form, in which greater efficiency and, often, greater stability is achieved. To illustrate the effectiveness and the accuracy of using the p and h versions, and for comparison purposes, we list the following results:

1. Results are given in Table (8.1) which compare the numerical solution with the given exact results. The very good agreement between the two sets of results displayed in Table (8.1) indicates that the "Numerical" finite element method solution of this type of problem is extremely accurate.
2. The error norms L_2 obtained by using the finite element p and h versions to the given problem are listed in Table (8.2).
3. The results of the highly accurate cubic elements are plotted in Figures (8.1) and (8.2) and shows the behaviour of the function u over the given region.

mesh lengths $\delta x=.1$, $\delta y=.2$

(0,1) 0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	(1,1) 0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	4.6591×10^{-4}	8.8671×10^{-4}	1.2203×10^{-3}	1.4346×10^{-3}	1.5085×10^{-3}	1.4346×10^{-3}	1.2203×10^{-3}	8.8669×10^{-4}	4.6590×10^{-4}	0.0000
0.0000	4.6613×10^{-4}	8.8663×10^{-4}	1.2203×10^{-3}	1.4346×10^{-3}	1.5684×10^{-3}	1.4345×10^{-3}	1.2203×10^{-3}	8.8651×10^{-4}	4.6599×10^{-4}	0.0000
0.0000	7.5424×10^{-4}	1.4347×10^{-3}	1.9745×10^{-3}	2.3214×10^{-3}	2.4406×10^{-3}	2.3213×10^{-3}	1.9745×10^{-4}	1.4346×10^{-3}	7.5422×10^{-4}	0.0000
0.0000	7.5428×10^{-4}	1.4347×10^{-3}	1.9747×10^{-3}	2.3214×10^{-3}	2.4408×10^{-3}	2.3213×10^{-3}	1.9746×10^{-4}	1.4345×10^{-3}	7.5406×10^{-4}	0.0000
0.0000	7.5428×10^{-4}	1.4347×10^{-3}	1.9745×10^{-3}	2.3214×10^{-3}	2.4407×10^{-3}	2.3214×10^{-3}	1.9745×10^{-3}	1.4346×10^{-3}	7.5423×10^{-4}	0.0000
0.0000	7.5431×10^{-4}	1.4348×10^{-3}	1.9748×10^{-3}	2.3215×10^{-3}	2.4409×10^{-3}	2.3214×10^{-3}	1.9746×10^{-3}	1.4346×10^{-3}	7.5408×10^{-4}	0.0000
0.0000	4.6591×10^{-4}	8.8675×10^{-4}	1.2203×10^{-3}	1.4347×10^{-3}	1.5085×10^{-3}	1.4347×10^{-3}	1.2203×10^{-3}	8.8670×10^{-4}	4.6591×10^{-4}	0.0000
0.0000	4.6617×10^{-4}	8.8670×10^{-4}	1.2204×10^{-3}	1.4347×10^{-3}	1.5085×10^{-3}	1.4347×10^{-3}	1.2204×10^{-3}	8.8670×10^{-4}	4.6617×10^{-4}	0.0000
(0,0) 0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	(1,0) 0.0000

TABLE 8.1: At each point the numbers represent:

Finite element solution with 100 cubic basis function

The exact solution

No. of elements \ Element order	Quadratic	Cubic
25	1.594×10^{-5}	9.06×10^{-7}
50	6.027×10^{-6}	3.258×10^{-7}
75	2.003×10^{-6}	1.202×10^{-7}

TABLE 8.2

The error norms L_2, as the number of elements is subdivided, i.e. the h version, and also as the degree of the polynomial is increased i.e. the p version of the rectangular plate problem.

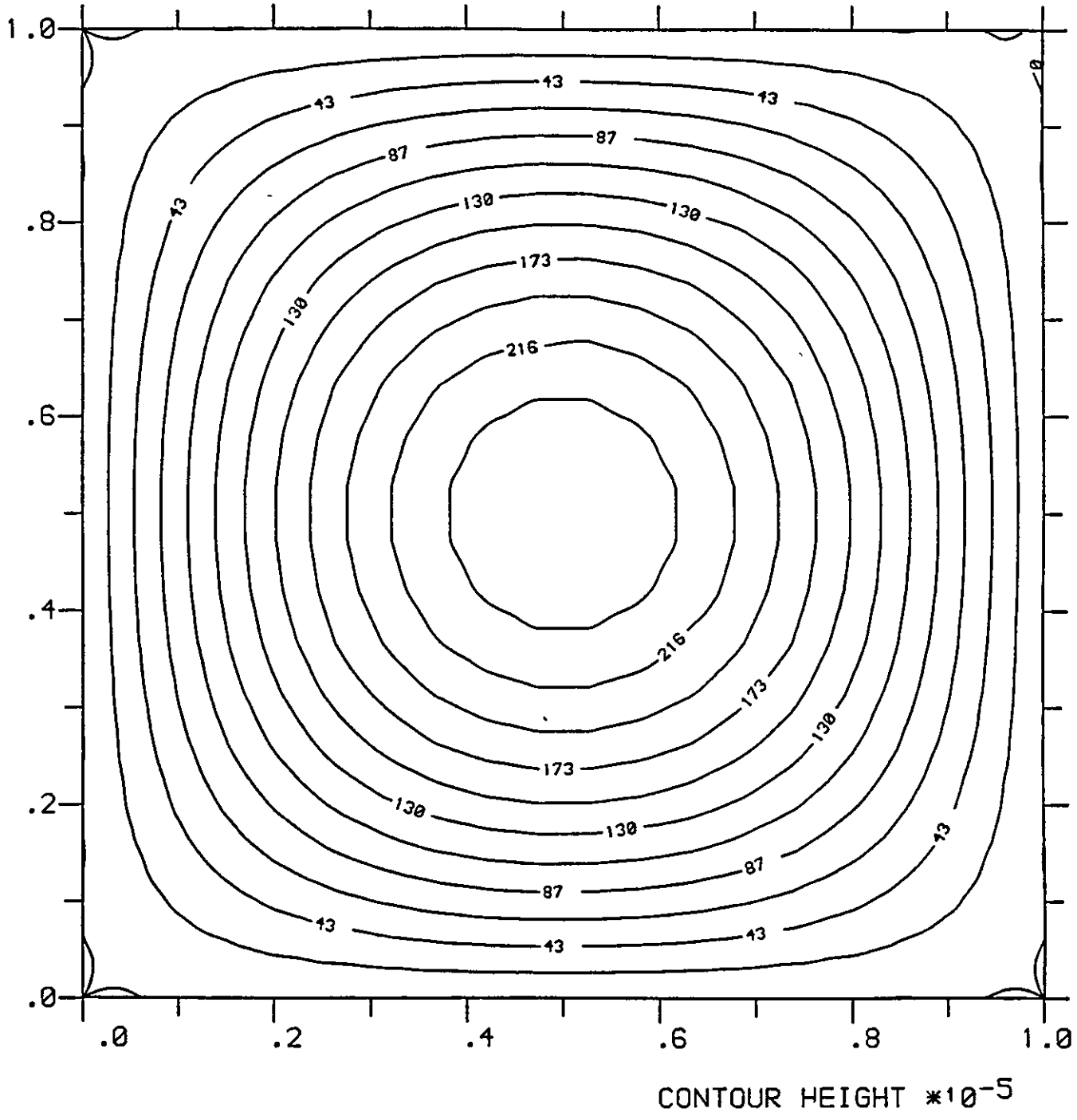


FIGURE 8.1

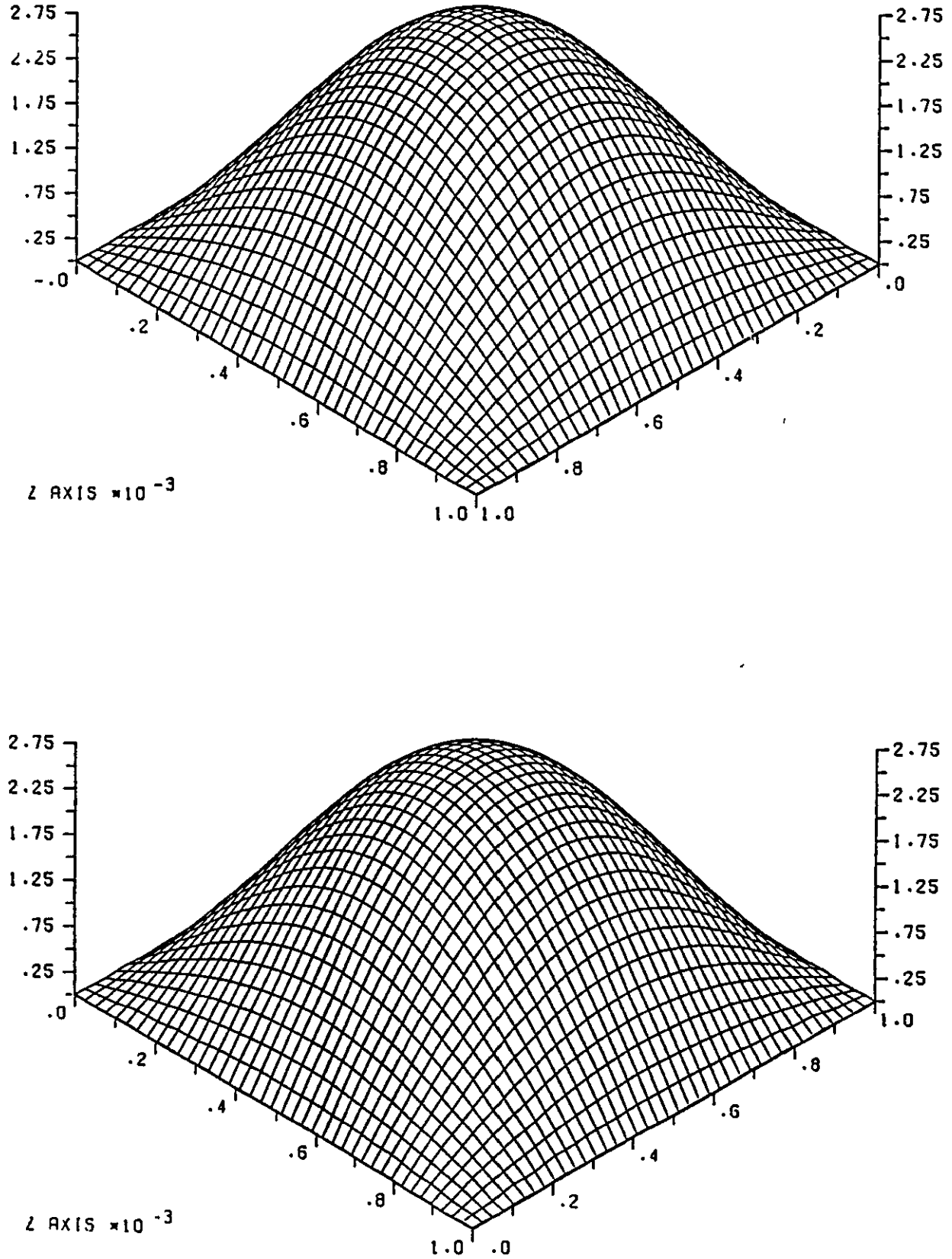


FIGURE 8.2: Isoparametric projection for different angles for the rectangular plate problem

8.3 POTENTIAL FLOW PROBLEM

We determine the potential flow past a right circular cylinder with the direction of flow perpendicular to the axis of the cylinder and the stream flow in a channel at a normal incidence to the disc. The fluid is assumed to be both inviscid and incompressible. Inviscid fluids experience no shearing stress, and when they come into contact with a solid boundary, they slip tangentially along it without resistance.

Dynamic aspects of fluid motion can be characterized by such concepts as laminar or irrotational and turbulent or rotational.

Inviscid irrotational flow is called potential flow because the velocity field in the flow can be derived from a potential function, traditionally denoted by the letter ϕ.

For a two-dimensional, incompressible, irrotational flow, the governing equation for the problem is,

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = 0 \ . \tag{8.9}$$

Two-dimensional flows can also be characterised by introducing the stream function ψ, which also satisfies Laplace's equation

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = 0 \tag{8.10}$$

The potential function ϕ and the stream function ψ are related to the x- and y- components of velocity, denoted by u and v respectively. Then,

$$u = -\frac{\partial \psi}{\partial y} \ , \quad v = \frac{\partial \psi}{\partial x} \ , \tag{8.11}$$

and

$$u = -\frac{\partial \phi}{\partial x} \ , \quad v = -\frac{\partial \phi}{\partial y} \ ,$$

whether we use the potential or stream function formulation mathematically the problem is the same as that of solving Laplace's equation, the difference arising only in the application of the boundary conditions.

APPLICATIONS

i. Inviscid laminar flow around an elliptic obstacle in a channel

The actual solution domain is infinite; for computational purposes it is necessary to construct a finite domain as shown in Figure (8.3).

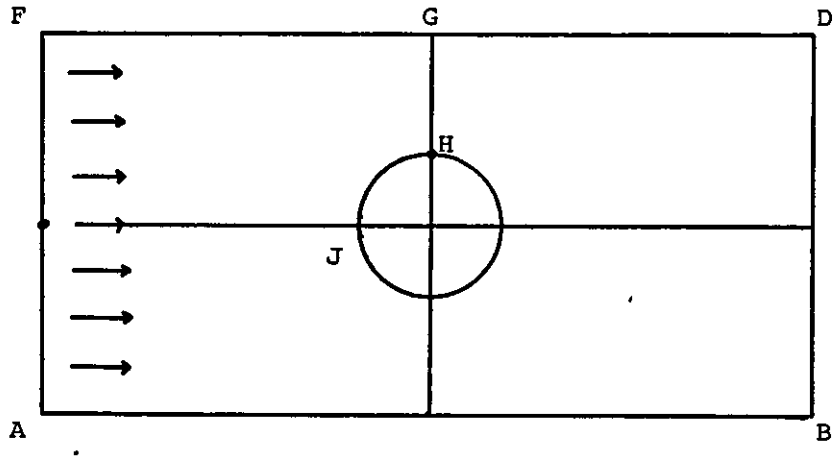


FIGURE 8.3: Flow past a circular obstacle in a channel

The nature of the boundary conditions for this rectangular domain are as follows,

on AF and BD $$\frac{\partial \phi}{\partial n} = \frac{\partial \phi}{\partial x} = - u_o ,$$

the velocity of the fluid is undisturbed by the solid body, because the

flow is *laminar*, there is no flow across the line AB and FD, that is,

$$\frac{\partial \phi}{\partial n} = \frac{\partial \phi}{\partial y} = 0 \text{ , along AB and FD.}$$

In addition, because there can be no flow through the cylinder wall

$$\frac{\partial \phi}{\partial n} = 0$$

along the circumference of the circle.

With the properly specified boundary conditions it is possible by taking advantage of symmetry to consider only a quarter of the domain (Figure (8.4)). The boundary conditions on FG, FK, KJ and JH are the same as those determined earlier, while the boundary condition on GH is $\phi=0$. Thus, we have Neumann boundary conditions everywhere except on the line GH, where the Dirichlet conditions apply.

We will study the following two cases:

(1) Solve Laplace's Equation (8.9) in the region,

$$-4 \leq x \leq 0, \quad 0 \leq y \leq 2 ,$$

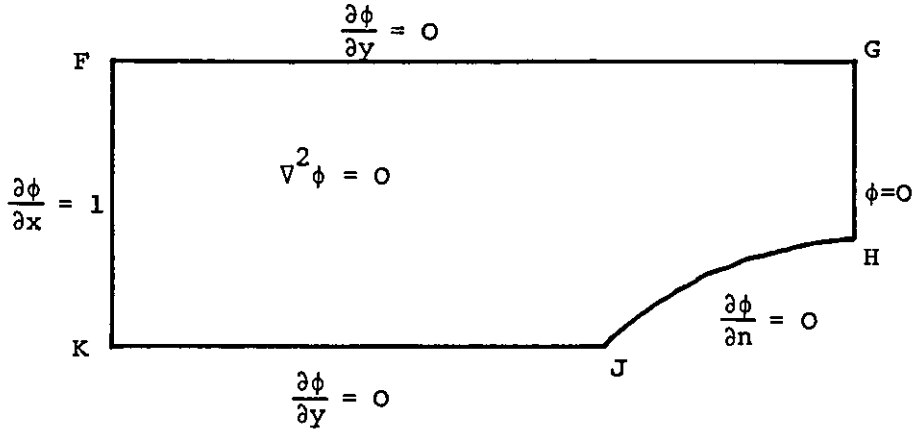
and on JH we define the ellipse,

$$\left(\frac{x-h}{2}\right)^2 + \left(\frac{y-k}{1}\right)^2 = 0 .$$

(2) Solve Laplace's Equation (8.9) in the region,

$$-4 \leq x \leq 0, \quad 0 \leq y \leq 2 ,$$

and on JH we define the ellipse $\left(\frac{x-h}{1}\right)^2+\left(\frac{y-k}{2}\right)^2=0$



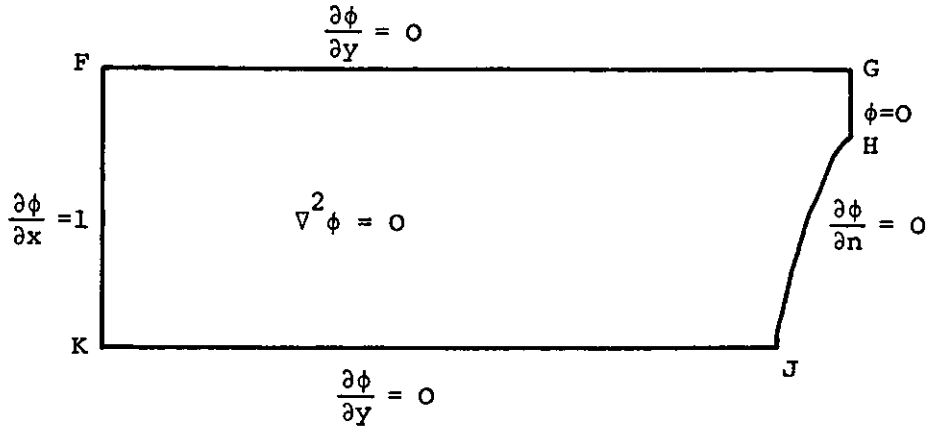


FIGURE 8.4: Boundary conditions for the quarter domain for both cases (1) and (2) respectively

The Numerical Results

The numerical values obtained with the finite element method for the regions given in Figure (8.4) are shown in Tables (8.3) and (8.4) respectively.

Solutions were calculated with 300 cubic elements, for the case (1), where JH has the form $(\frac{x-h}{2})^2+(\frac{y-k}{1})^2=0$, we note that ϕ has values: $0\leq\phi\leq5.2516$. While for the case (2), when JH has the form $(\frac{x-h}{1})^2+(\frac{y-k}{2})=0$, we note that ϕ has values: $0\leq\phi\leq6.1655$.

This slight difference in the values of ϕ are due to replacing the minor and major axes of the ellipse within the same region. This is shown more clearly in Figures (8.5) and (8.6) which shows the behaviour of both solutions in the given region.

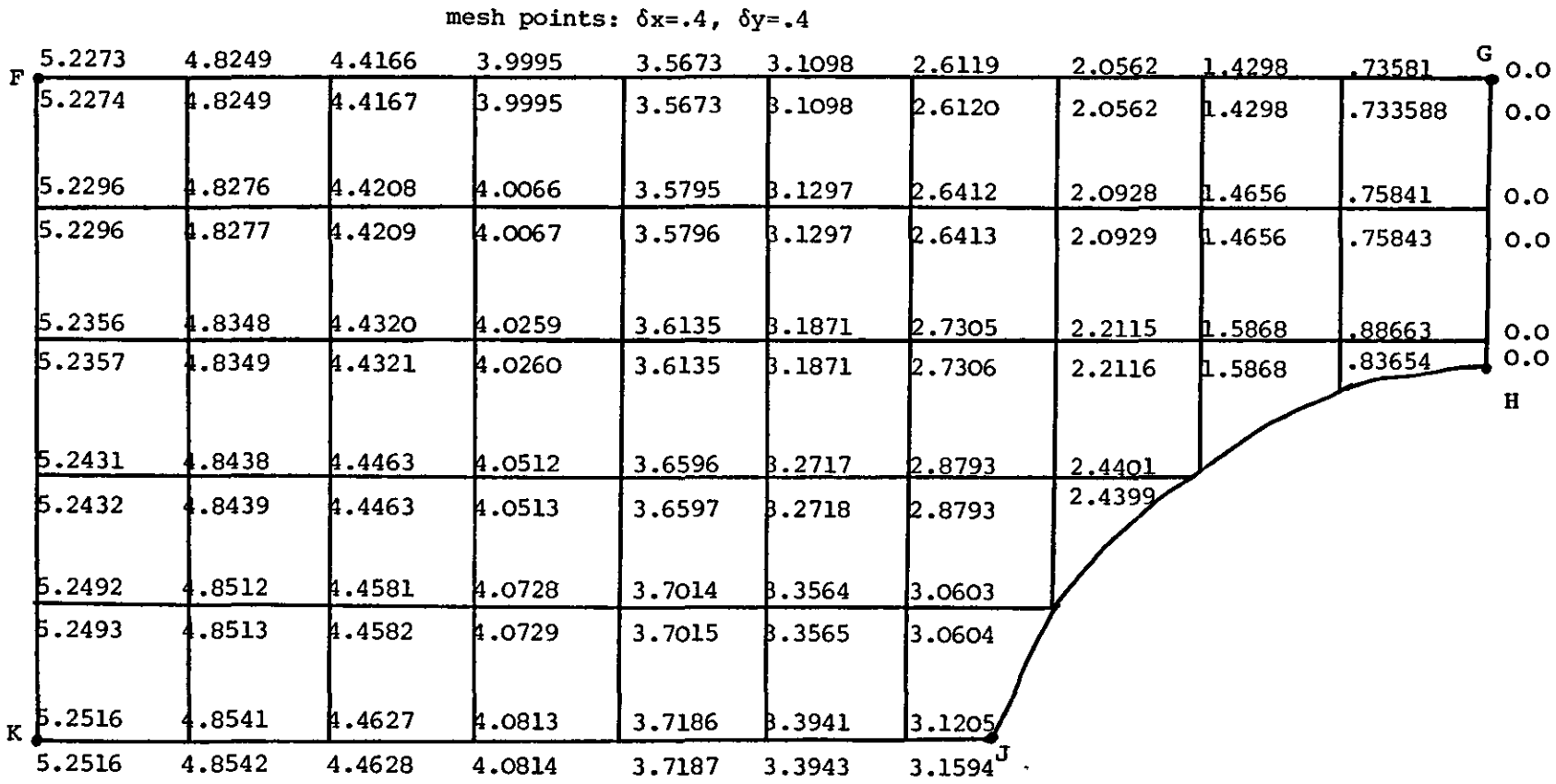


TABLE 8.3: At each point the numbers represent:

Finite element solution with quadratic 300 elements

Finite element solution with cubic 300 elements

mesh point $\delta x = \delta y = .4$

F 6.1434	5.7411	5.3334	4.9172	4.4860	4.0272	3.5179	2.9180	2.1632	1.1844	G 0.0
6.143	5.7414	5.3337	4.9175	4.4862	4.0275	3.5182	2.9182	2.1637	1.1848	0.0
6.1455	5.7436	5.3374	4.9243	4.4989	4.0510	3.5613	2.9932	2.2806	1.3091	0.0
6.1457	5.7439	5.3377	4.9246	4.492	4.0513	3.5613	2.9938	2.2815	1.3099	0.0
6.1510	5.7503	5.3479	4.9428	4.5330	4.1141	3.6775	3.2048	2.6496		H
6.1512	5.7505	5.3482	4.9431	4.5332	4.1143	3.6778	3.2052	2.6505		
6.1578	5.7585	5.3609	4.9659	4.5754	4.1933	3.8261	3.4869			
6.1581	5.7588	5.3611	4.9661	4.5757	4.1936	3.8266	3.4868			
6.1634	5.7652	5.3714	4.9846	4.6101	4.2584	3.9496	3.7223			
6.1636	5.7654	5.3717	4.9848	4.6103	4.2586	3.9498	3.7223			
6.1635	5.7677	5.3754	4.9917	4.6233	4.2834	3.9971	3.8118			
K 6.1657	5.7680	5.3757	4.9920	4.6236	4.2836	3.9972	3.8124	J		

TABLE 8.4: At each point the numbers represent:

Finite element with quadratic 300 elements

Finite element with cubic 300 elements

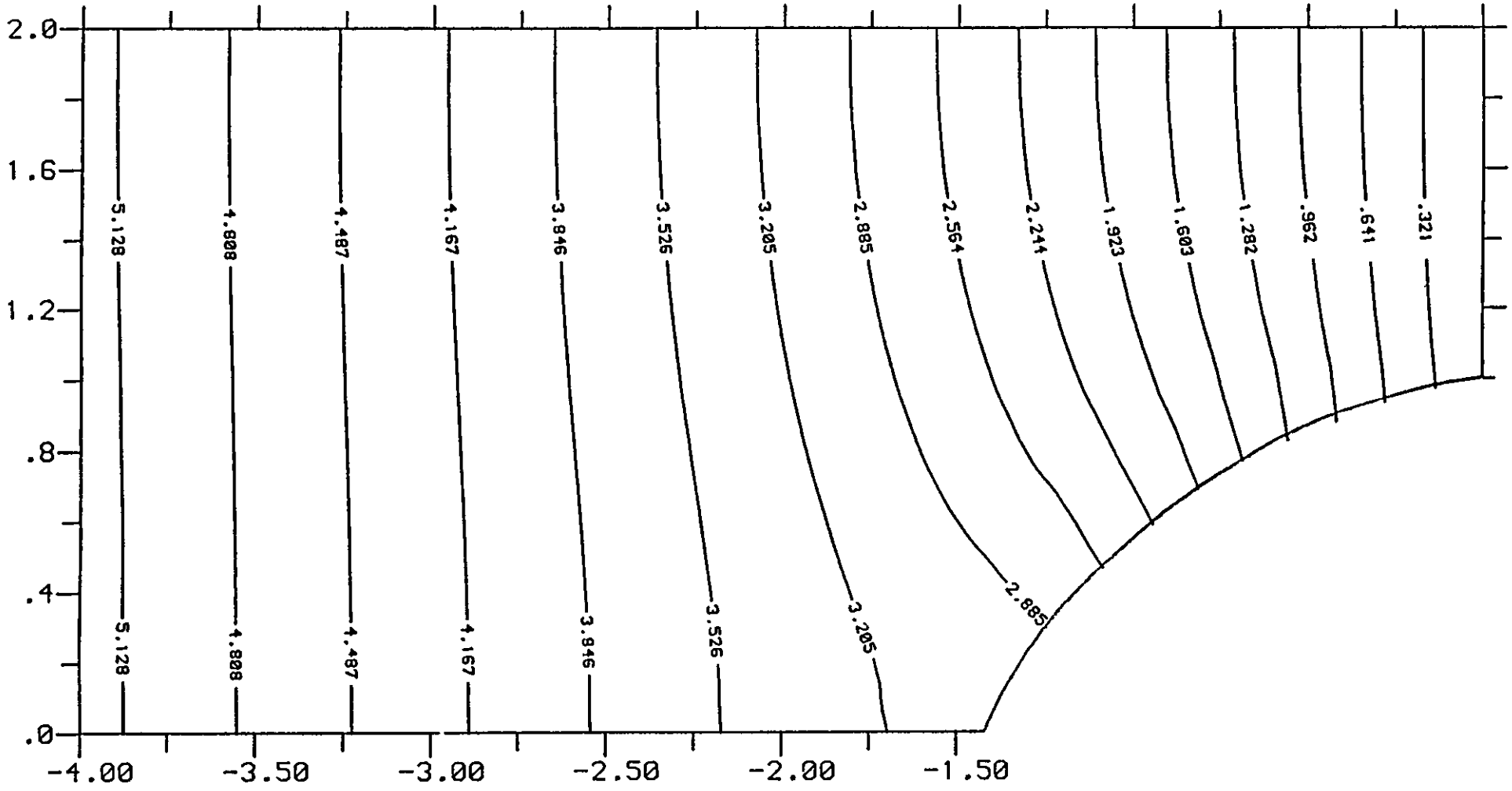

2.0
1.6
1.2
.8
.4
.0
-4.00
-3.50
-3.00
-2.50
-2.00
-1.50
5.128
4.808
4.487
4.167
3.846
3.525
3.205
2.885
2.564
2.241
1.923
1.603
1.282
.962
.641
.321

FIGURE 8.5

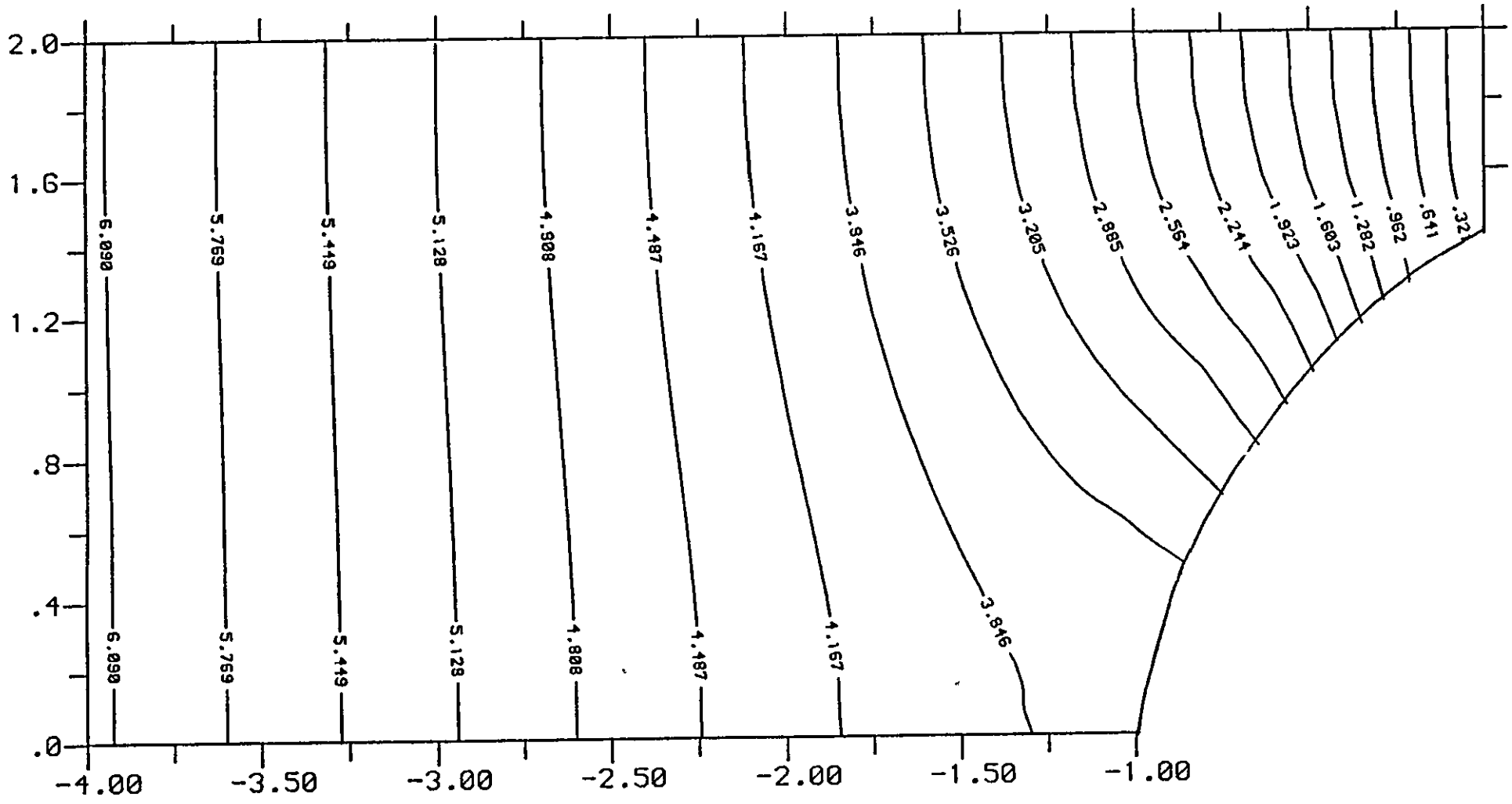

2.0
1.6
1.2
.8
.4
.0
-4.00
-3.50
-3.00
-2.50
-2.00
-1.50
-1.00
6.090
5.769
5.449
5.128
4.808
4.487
4.167
3.846
3.526
3.205
2.885
2.564
2.244
1.923
1.603
1.282
.962
.641
.321

FIGURE 8.6

ii. Inviscid laminar flow in a channel past a disc

The stream function ψ represents the flow in a channel at a normal incidence to the disc, as shown in Figure (8.7) which because of the symmetry only a quarter of the domain needs to be considered.

The prescribed boundary conditions for the problem are as follows:

on DC $\psi = 2$,

on AD $\psi = y$,

on AB $\psi = 0$,

on CO $\frac{\partial \psi}{\partial n} = 0$,

and on OB $\psi = 0$.

An infinite speed will be acquired by the stream at point O, the edge of the plane, giving rise to a singularity in the solution.

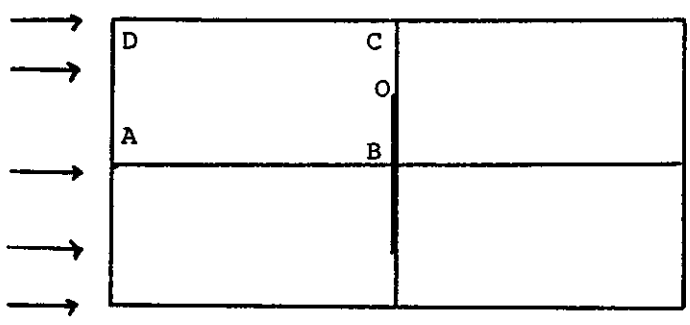


FIGURE 8.7a: Flow past a disc in a channel

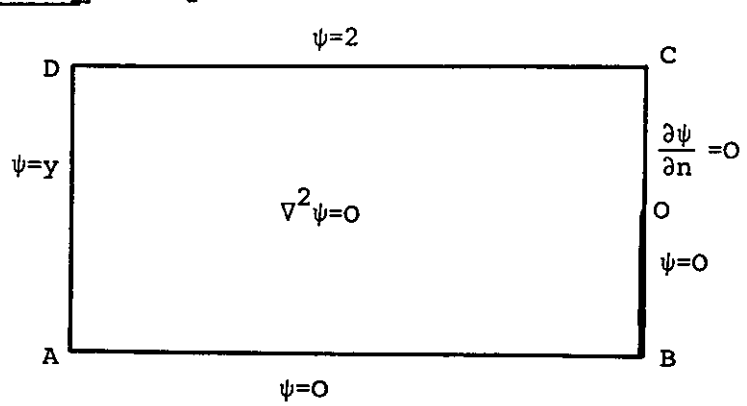


FIGURE 8.7b: Boundary conditions for a quarter domain

The Numerical Results

Table (8.5) summarizes the results of the finite element solution of the stream function ψ in the region,

$$-2 \leq x \leq 0 , \quad 0 \leq y \leq 2 ,$$

and the boundary conditions given in Figure (8.7b). Both 300 quadratic, and cubic triangular elements were used to solve this problem. For the Q.B.F. (Quadratic Basis Function) and the C.B.F. (Cubic Basis Function), the small differences in the results given in Table (8.5) around the boundary line OC which are higher than anywhere else in the given domain are due to the infinite speed at the point O, the edge of the plate, i.e. the sudden change of the boundary condition at O from ψ=0 to $\frac{\partial \psi}{\partial n} = 0$ gives rise to a singularity in the solution which we tried to minimize by using the same procedure as that applied in Chapter 6. Graphs of the solution ψ showing the behaviour of the solution over the given region are presented also in Figures (8.8a) and (8.8b).

2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.000	2.0000	2.0000	2.0000
2.0000 1.6000	2.0000 1.5901	2.0000 1.5792	2.0000 1.5667	2.0000 1.5516	2.0000 1.5329	2.0000 1.5102	2.0000 1.4839	2.0000 1.4564	2.0000 1.4338	2.0000 1.4247
1.6000 1.2000	1.5901 1.1836	1.5793 1.1653	1.5667 1.440	1.5515 1.1173	1.5329 1.0829	1.5102 1.0379	1.4839 .97841	1.4565 .90134	1.4340 .80927	1.4250 .75083
1.2000 .8000	1.1835 .78310	1.654 .76414	1.440 .74180	1.1174 .71286	1.0830 .67467	1.0380 .62197	.97868 .54715	.90161 .43715	.80957 .26599	.75127 0.00000
.80000 .40000	.78302 .38935	.76427 .37737	.74176 .36302	.71302 .34474	.67470 .31991	.62201 .28616	.54734 .23955	.43738 .17650	.26612 .094887	0.00000 0.00000
.40000 0.00000	.38929 0.00000	.37742 0.00000	.36307 0.00000	.34461 0.00000	.31991 0.00000	.28610 0.00000	.23957 0.00000	.17645 0.00000	.094947 0.00000	0.00000 0.00000
0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

D C O A B

TABLE 8.5: At each point the numbers represent:

Finite element solution with quadratic elements

Finite element solution with cubic elements

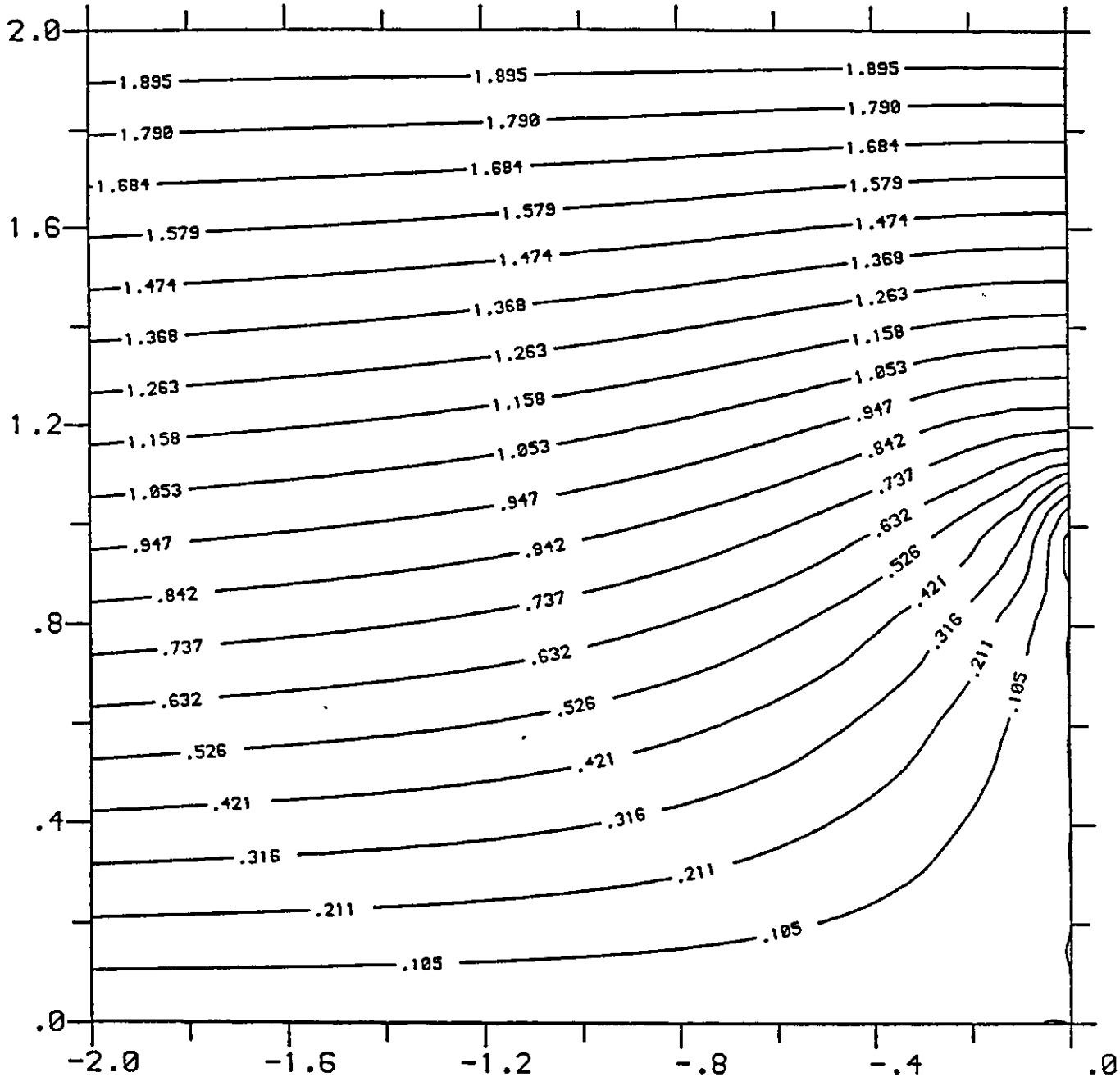

2.0
1.6
1.2
.8
.4
.0
-2.0
-1.6
-1.2
-.8
-.4
.0
1.895
1.790
1.684
1.579
1.474
1.368
1.263
1.158
1.053
.947
.842
.737
.632
.526
.421
.316
.211
.105

FIGURE 8.8a

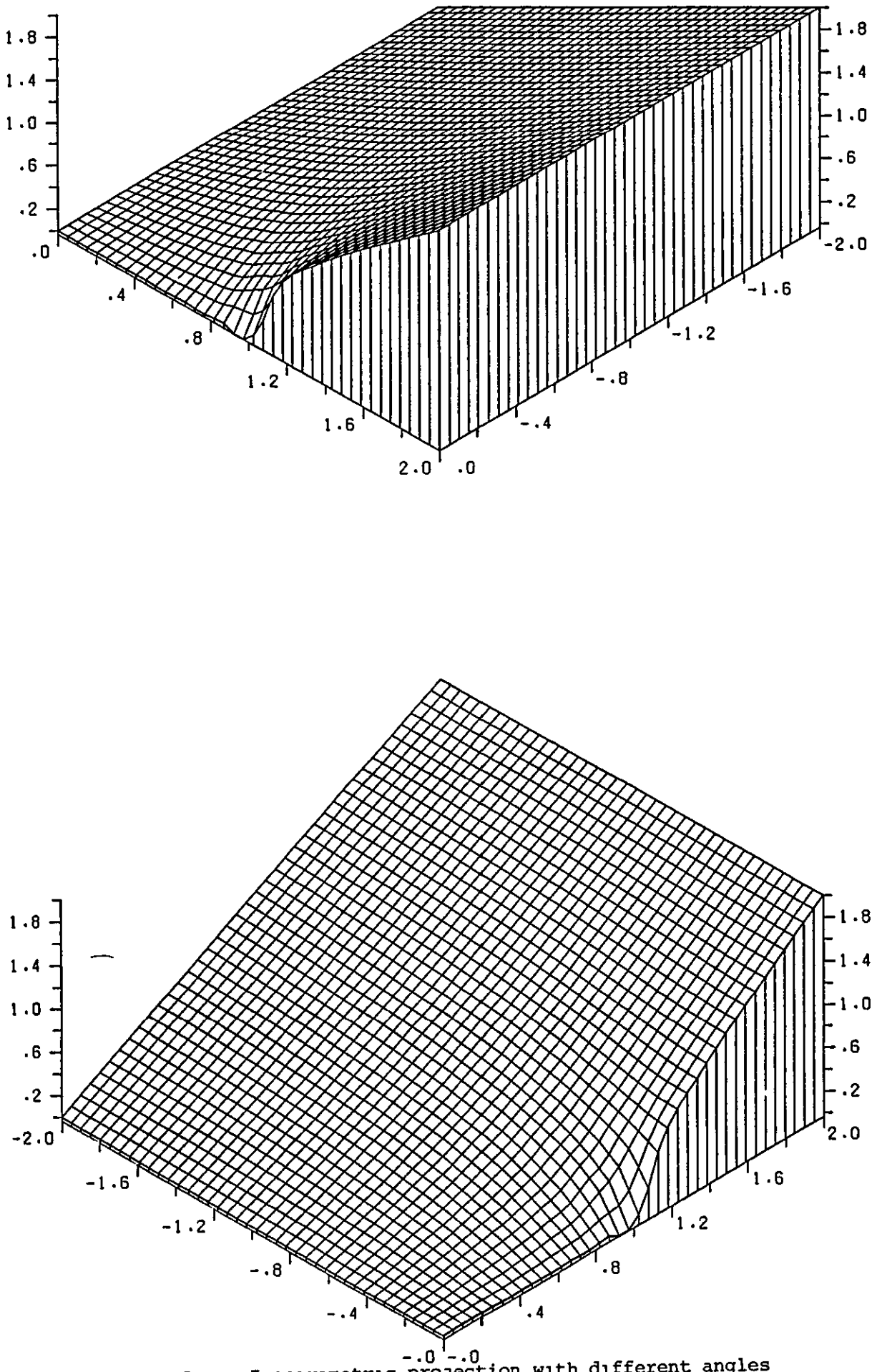


FIGURE 8.8b: Isoparametric projection with different angles

8.4 THE EIGENVALUE PROBLEM

Let R be a bounded two-dimensional domain with boundary ∂R, λ an eigenvalue of Laplace operator ∇^2 over the region R, for which there exists a non-zero function u defined on R, such that,

$$\nabla^2 u + \lambda u = 0 , \qquad (x,y) \in R,$$

$$\text{with} \qquad u = g_1, \qquad (x,y) \in \partial R_1 , \qquad (8.12)$$

$$\text{and} \qquad \frac{\partial u}{\partial n} = g_2, \qquad (x,y) \in \partial R_2 .$$

The eigenfunctions may be normalized so that,

$$\int_R u^{(k)} u^{(\ell)} dxdy = \left\{ \begin{array}{ll} 1, & k=\ell \\ 0, & k\neq\ell \end{array} \right\} \qquad (8.13)$$

We are interested in computing an accurate approximation to the smallest eigenvalue and its corresponding eigenfunction. Furthermore, we want to estimate the accuracy of our approximation by comparing the finite element solution of different approximations. We also examine the possibilities of accelerating the convergence, as the size of the elements gets smaller, by various adaptations of the procedure of *grid refinement* as used in Chapter 6.

Firstly, we consider the L-shaped membrane eigenvalue problem, in which we determine the smallest eigenvalue and eigenfunction of equation (8.12), in the region of Figure (8.9) below, with u=0 on the boundary.

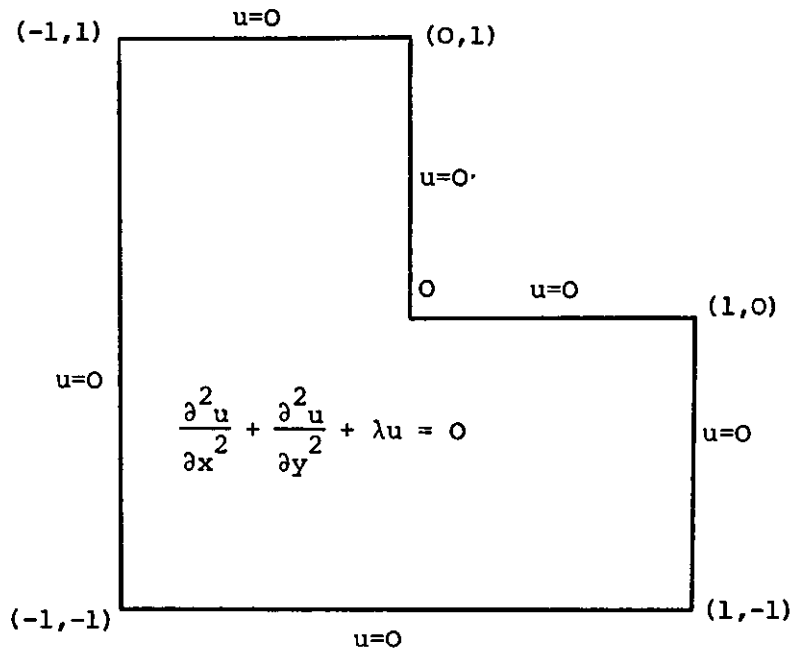


FIGURE 8.9: L-Shaped region for the eigenvalue problem

Since the problem illustrated in Figure (8.9) involves a re-entrant corner of internal angle $\frac{3\pi}{2}$ at which a boundary singularity occurs, therefore, we obviously need the mesh refinement procedure which is discussed in Chapter Six in order to produce a useful answer, so we will consider the procedure of refining the elements in the neighbourhood of the singular point 0, and examine the possibilities of accelerating the convergence as the number of elements and the degree of the basis function is increased. When the finite element method is applied to this problem it gives rise to an approximating matrix eigenvalue problem which is solved by the inverse power method.

An estimate of the smallest eigenvalue λ_n is given in Table (8.6), where the problem is solved without the adoption of a grid refinement procedure, the results obtained reveal that the convergence to the correct value λ_n for the three cases (quadratic, cubic and quartic elements) are slow indeed as the number of elements increases.

No. of elements with n=50	Value of λ_n for the *quadratic* case	Value of λ_n for the *cubic* case	Value of λ_n for the *quartic* case
n	9.74199	9.67078	9.65595
2n	9.70149	9.66194	9.65119
3n	9.69670	9.66191	-

TABLE 8.6

We note that the value of λ_n=9.6397 which is correct to five significant figures was obtained by REID and WALSH [1965]. Other estimates of the smallest eigenvalue λ_n are given in Table (8.7), where this time the problem was solved with the adoption of a *grid refinement procedure* around the singular point O. This time the values of λ_n appear to be converging quite rapidly to the accurate value. Note that we obtained the best estimated value of λ_n which is equal to 9.63990 by solving the problem with 300 cubic elements dense around the singular point O.

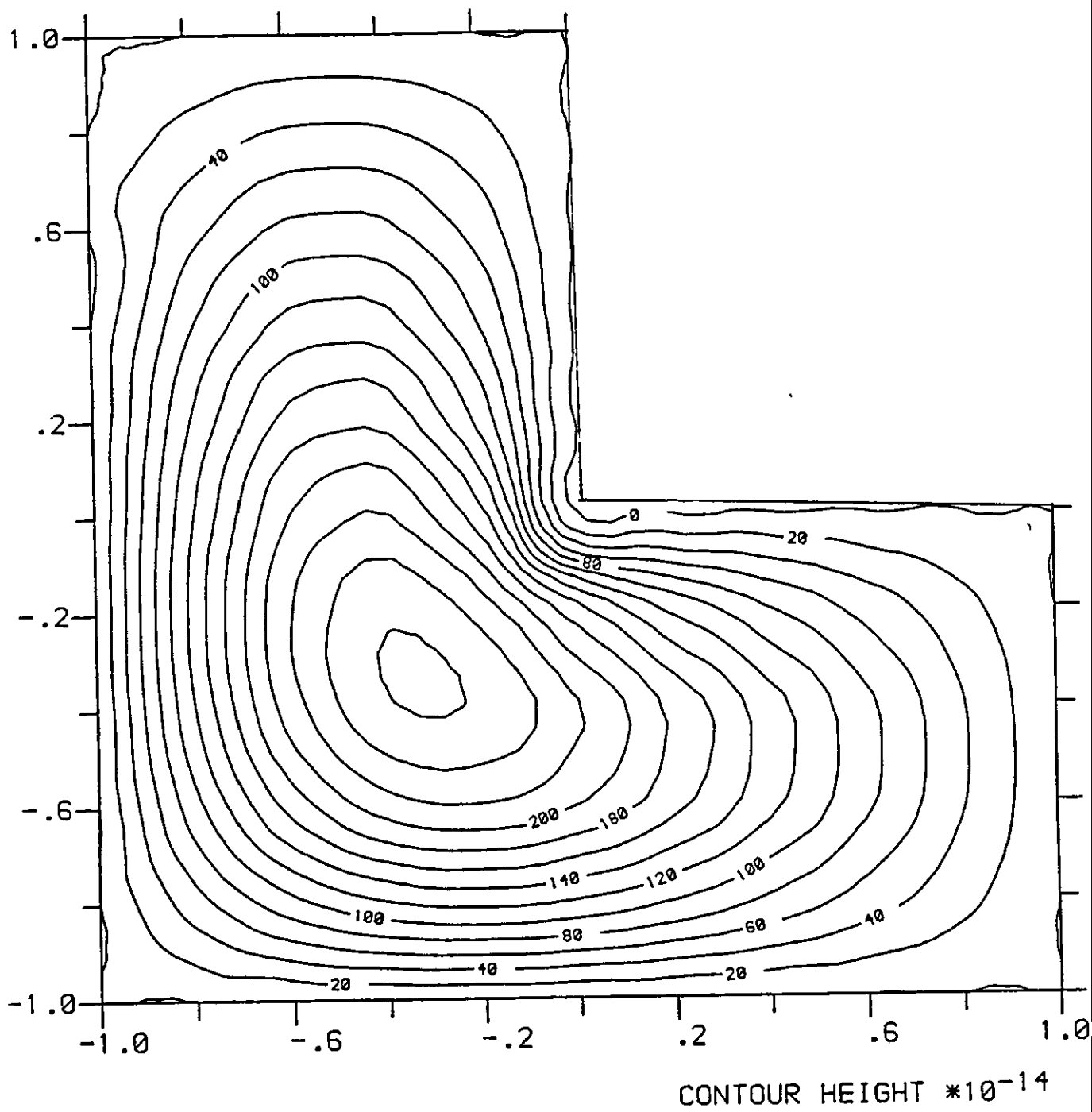


FIGURE 8.10: The Eigenfunction u corresponding to λ_n=9.6399

No. of elements (with n=50)	Value of λ_n for the *quadratic* case	Value of λ_n for the *cubic* case	Value of λ_n for the *quartic* case
n	9.83687	9.64859	9.64294
2n	9.69799	9.64207	9.64049
3n	9.65760	9.64047	-

TABLE 8.7: The dependence of λ_n upon the number of elements and the order of the elements

The results of the eigenfunction u corresponding to the smaller eigenvalue λ_n=9.63990 is plotted in Figure (8.10) and shows the behaviour of the eigenfunction u in the given region.

Secondly we consider the two-dimensional problem given by Equation (8.12a) defined on the following three regions.

The *first* region is given in Figure (8.11) which is reduced by symmetry to only one quarter i.e. Figure (8.12). The boundary conditions thus are such that the function u(x,y) vanishes on the boundary and has zero normal derivative on the lines of symmetry.

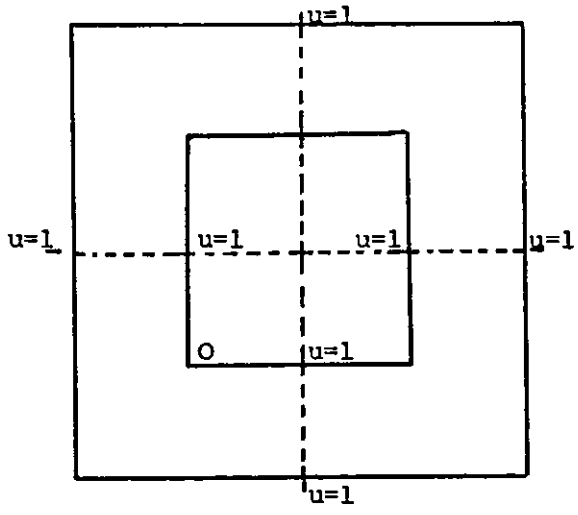


FIGURE 8.11

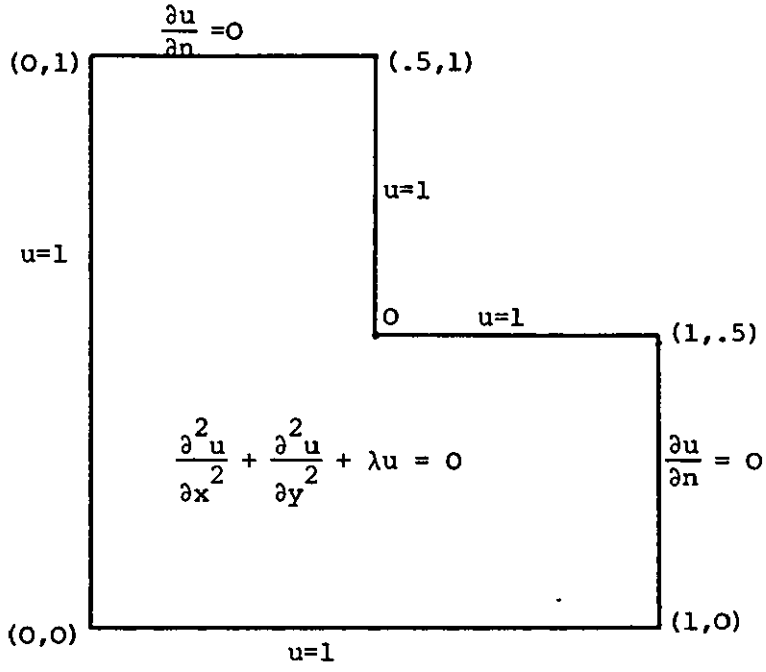


FIGURE 8.12

Now Table (8.8) lists the value of the smallest eigenvalue λ_n obtained with 300 quadratic and cubic elements. The results of the eigenfunction u corresponding to the smallest eigenvalue λ_n=.963360 is plotted in Figure (8.13) which shows the behaviour of the eigenfunction u in the given region.

Order of the element	Equally Distributed elements	Dense elements around the singular point 0
Quadratic	.963449	.963471
Cubic	.963360	.963360

TABLE 8.8: Values of λ_n with 300 elements

The *second* region is given in Figure (8.14) which again is reduced to only one half by symmetry in Figure (8.15). The function u(x,y) vanishes on the boundary and has zero normal derivative on the lines of symmetry as shown in Figure (8.15).

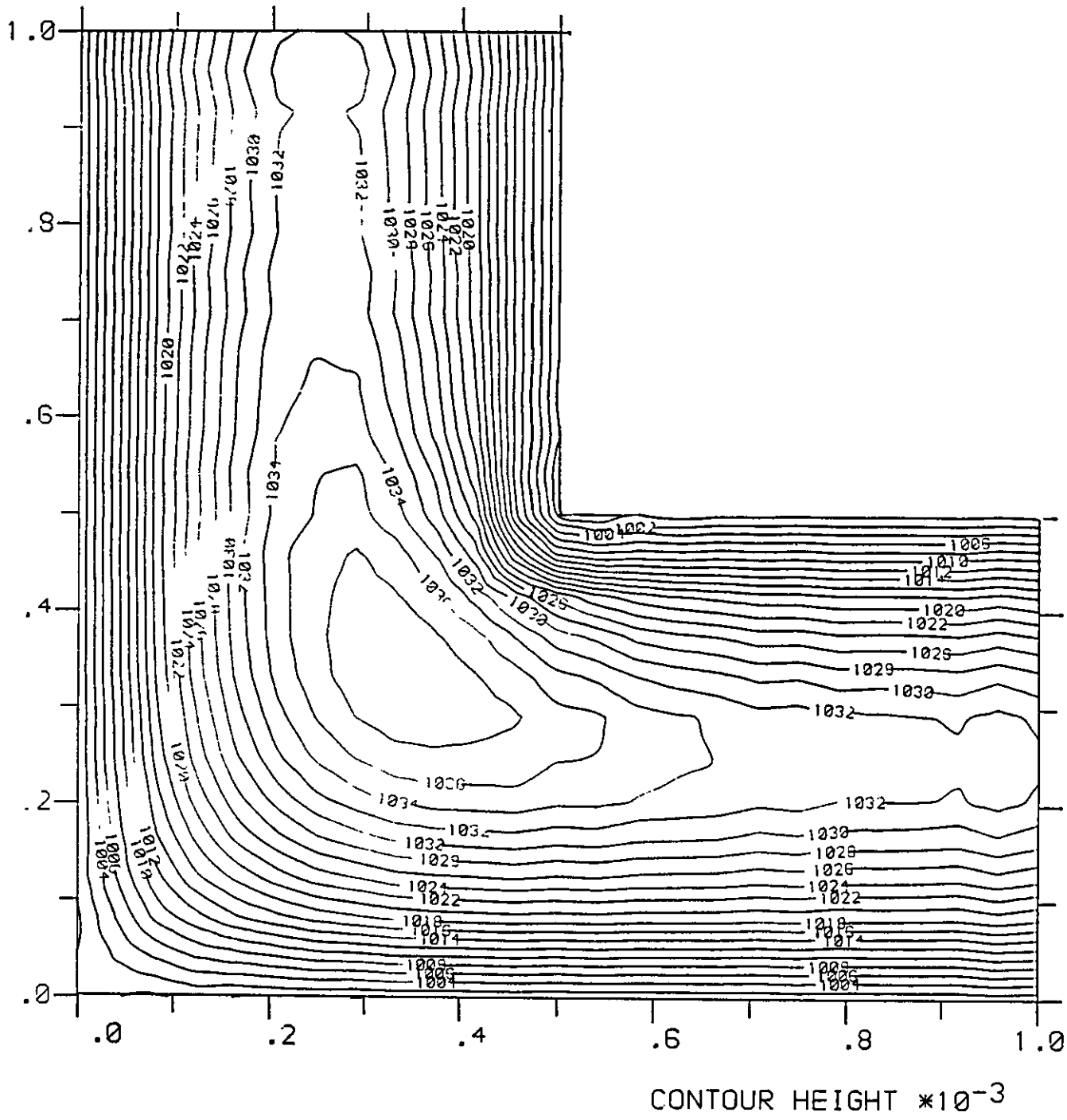


FIGURE 8.13: The Eigenfunction u corresponding to λ_n=.96330

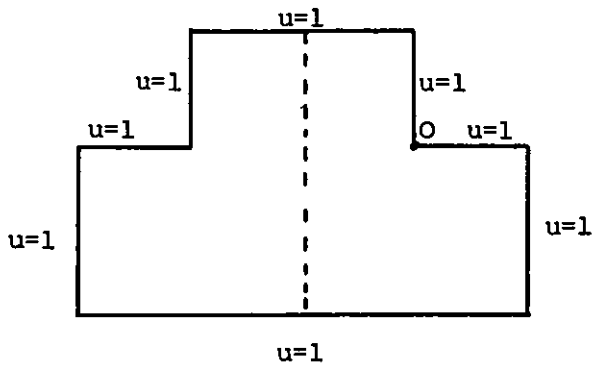


FIGURE 8.14

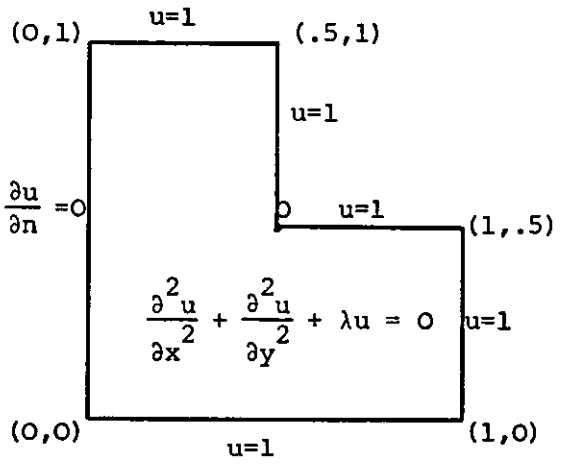


FIGURE 8.15

Table (8.9) below lists the value of the smallest eigenvalue for the region of Figure (8.15) and is obtained with 300 quadratic and cubic elements.

Also, the results of the eigenfunction u corresponding to the smallest eigenvalue λ_n=.925756, is plotted in Figure (8.16) which shows the behaviour of the eigenfunction u in the given region.

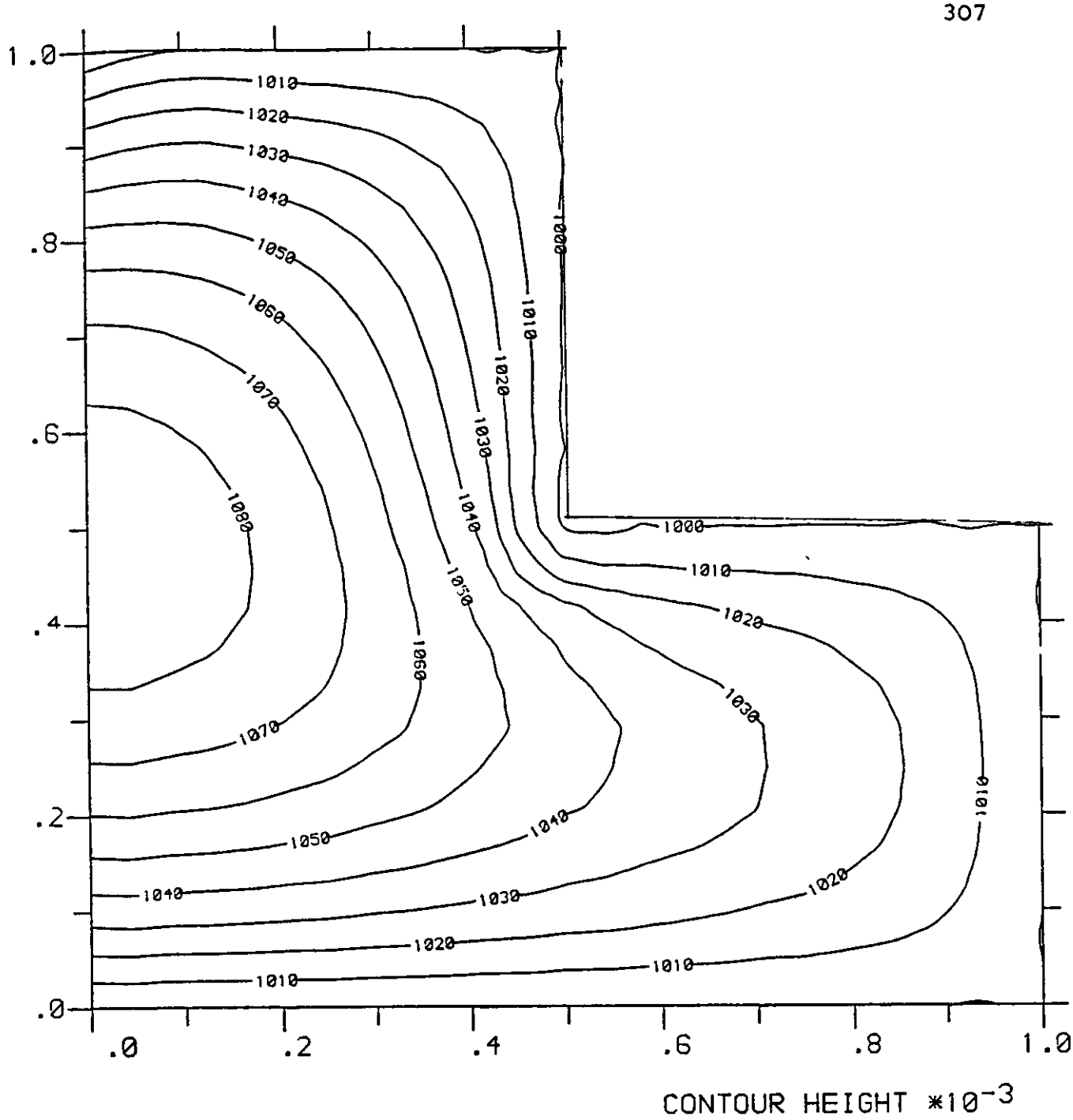


FIGURE 8.16: The Eigenfunction u corresponding to λ_n=.925756

Order of the elements	Equally distributed elements	Dense elements around the singular point O
Quadratic	.925815	.92514
Cubic	.925714	.925756

TABLE 8.9: Value of λ_n with 300 elements

The *third* region is given as in Figure (8.17) which is again reduced to only one quarter by symmetry in Figure (8.18).

The function u(x,y) vanishes on the boundary and has zero normal derivative on the line of symmetry, as given in Figure (8.18).

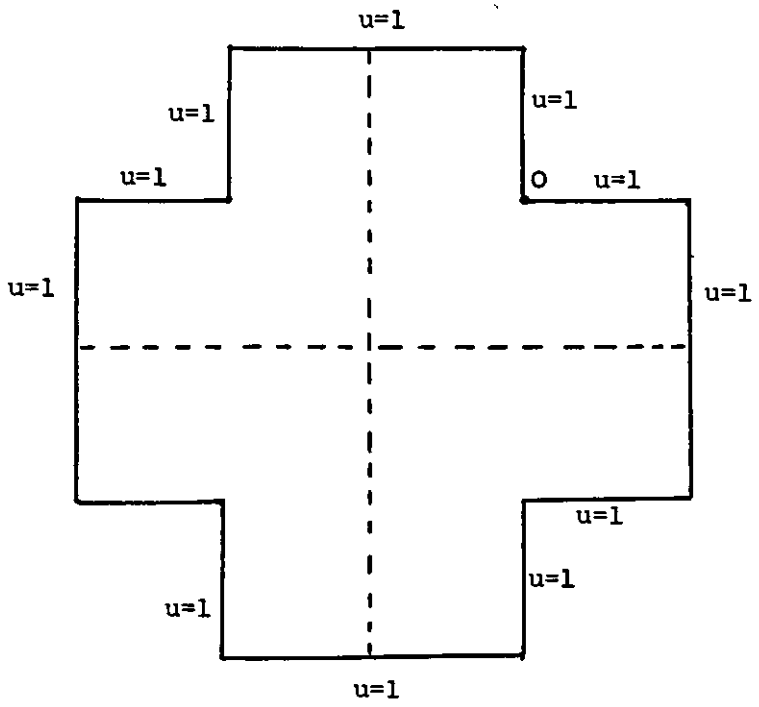


FIGURE 8.17

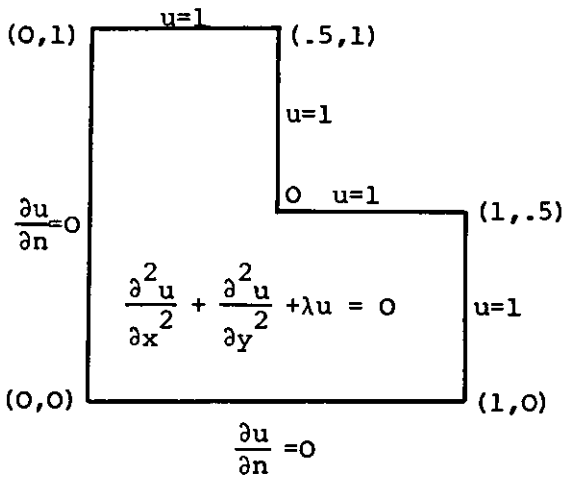


FIGURE 8.18

Table (8.10) below lists the value of the smallest eigenvalue λ_n for the L-shaped region of Figure (8.18) which is obtained with 300 quadratic and cubic elements.

Also the numerical results of the eigenfunction u corresponding to the smallest eigenvalue λ_n=.841250 is plotted in Figure (8.19) which shows the behaviour of the eigenfunction u in the given region.

Order of the elements	Equally distributed element	Dense element around the singular point 0
Quartic	.841995	.841511
Cubic	.841232	.841250

TABLE 8.10: Value of λ_n with 300 elements

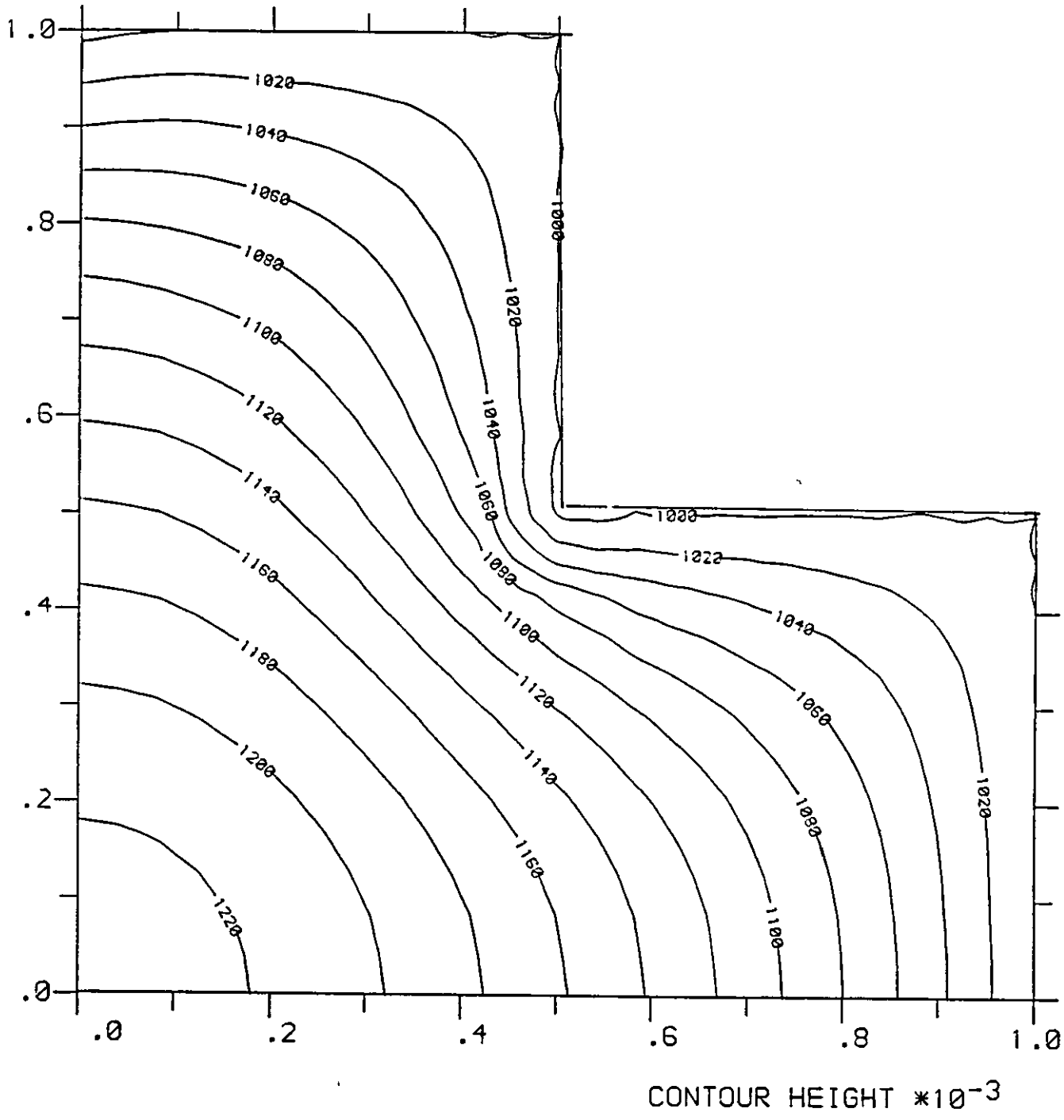


FIGURE 8.19: The Eigenfunction u corresponding to λ_n=.841250

8.5 FINITE ELEMENT SOLUTIONS OF THE NAVIER-STOKES EQUATIONS

The class of problems considered in this section consists of those which are governed by the two-dimensional Navier-Stokes equations. The fluid motion considered is assumed to be laminar, steady and isothermal and the fluid assumed to be incompressible.

With these assumptions the mathematical description of the fluid motion consists of the equations of motion,

$$\rho\, u \frac{\partial u}{\partial x} + \rho\, v \frac{\partial u}{\partial y} = - \frac{\partial P}{\partial x} + \nu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) \tag{8.14}$$

$$\rho\, u \frac{\partial v}{\partial x} + \rho\, v \frac{\partial v}{\partial y} = - \frac{\partial P}{\partial y} + \nu \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) \tag{8.15}$$

and the continuity equation,

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{8.16}$$

where u and v denote the velocity components, P the pressure, ρ the density and ν the kinematic viscosity.

Now if ν=1, then the analytic solution is given by,

$$\left.\begin{aligned} u(x,y) &= - \cos x \sin y \\ v(x,y) &= \sin x \cos y \end{aligned}\right\} \tag{8.17}$$

The substitution of (8.17) in (8.14) and (8.15), with ν=1 gives,

$$\left.\begin{aligned} \frac{1}{\rho}\frac{\partial P}{\partial x} &= (2 \sin y + \sin x) \cos x \\ \frac{1}{\rho}\frac{\partial P}{\partial y} &= (\sin y - 2 \sin x) \cos y \end{aligned}\right\} \tag{8.18}$$

It is clearly difficult to find a function p(x,y) satisfying both equations (8.18) exactly.

However, the case of the Navier-Stokes equations is of some interest and we can construct a problem based on the exact expressions (8.17), to illustrate the numerical solution of the pair of simultaneous equations (8.14) and (8.15) by the finite element method. This can be done by substituting the expressions (8.18) for the pressure terms in (8.14) and (8.15), then the resulting equations were solved by TWODEPEP within the square region: $0 \leq x \leq \pi, 0 \leq y \leq \pi$, using the boundary velocities given by (8.17).

Numerical solutions were obtained using 100 cubic elements and are compared with the numerical solutions given by DENNIS and HUDSON [1979] who used finite-difference approximations to solve the problem.

The results obtained for the velocity u(x,y) are accurate and are given in Table (8.11) below.

y/π	Finite element Solution	DENNIS solution (First aporox.)	DENNIS solution (Second approx.)	Exact Solution
0	0	0	0	0
0.1	0.18160	0.1828	0.18095	0.18164
0.2	0.34556	0.3475	0.34429	0.34549
0.3	0.47549	0.4781	0.47383	0.47553
0.4	0.55901	0.5620	0.55689	0.55902
0.5	0.58779	0.5909	0.58549	0.58797

TABLE 8.11: A comparison between the finite element solution of u(x,y) with those of DENNIS and HUDSON and the exact solution of the Navier-Stokes equation for $\nu=1$, $x=.7\pi$, and values of y/π in the range 0.0-0.5

Similar accuracy was obtained for v(x,y).

We note that also the solution given by ROSCOE [1975, p.300, Table 2] are not in agreement with (8.17), however our results are accurate, and with 100 cubic elements, the present method converges with an L_2 norm = 0.36403×10^{-4} which is very promising. A plot of the functions u(x,y) values showing the behaviour of the solution over the region are presented in Figures (8.20) and (8.21).

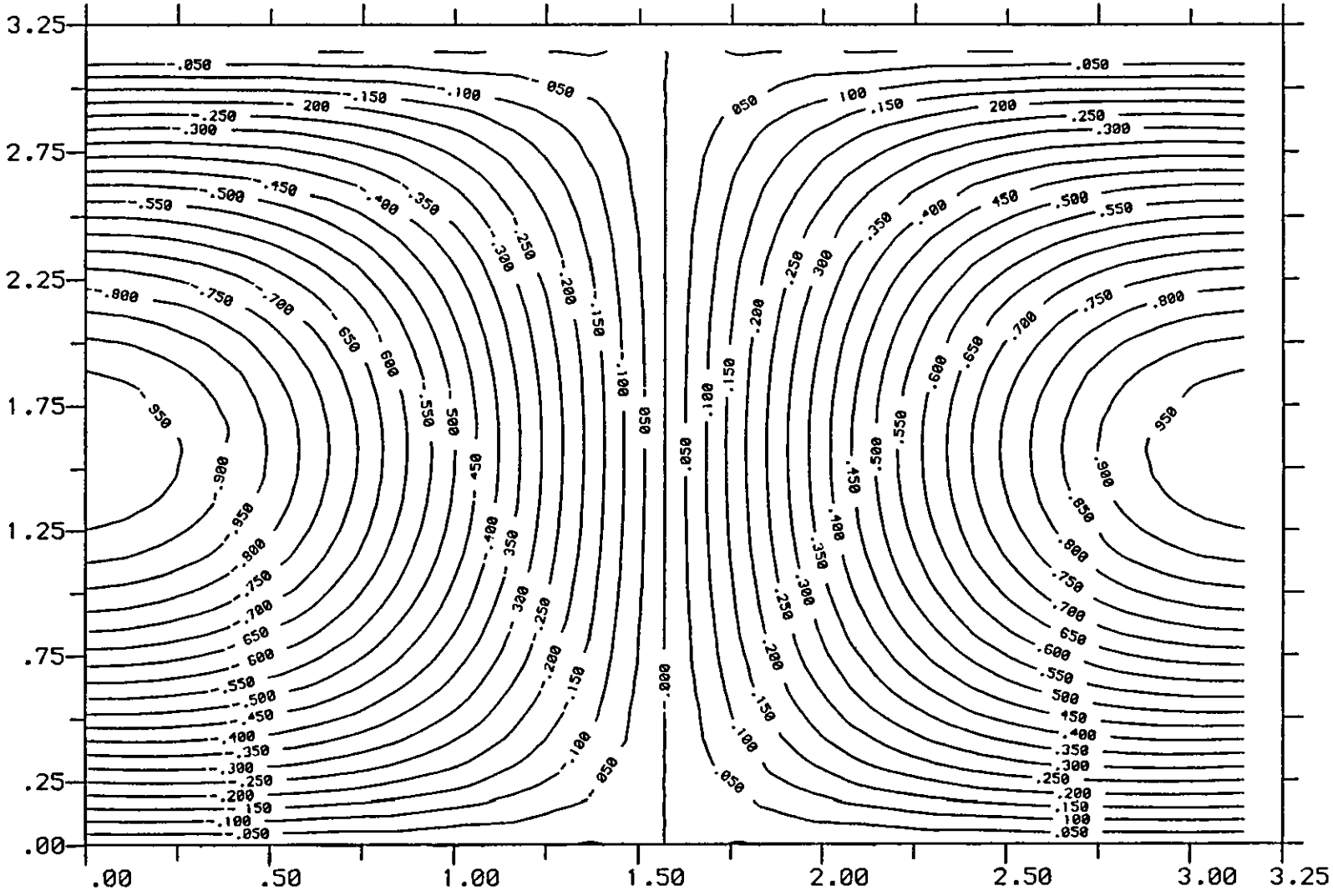


FIGURE 8.20: Contour lines for the velocity u of Navier-Stokes problem

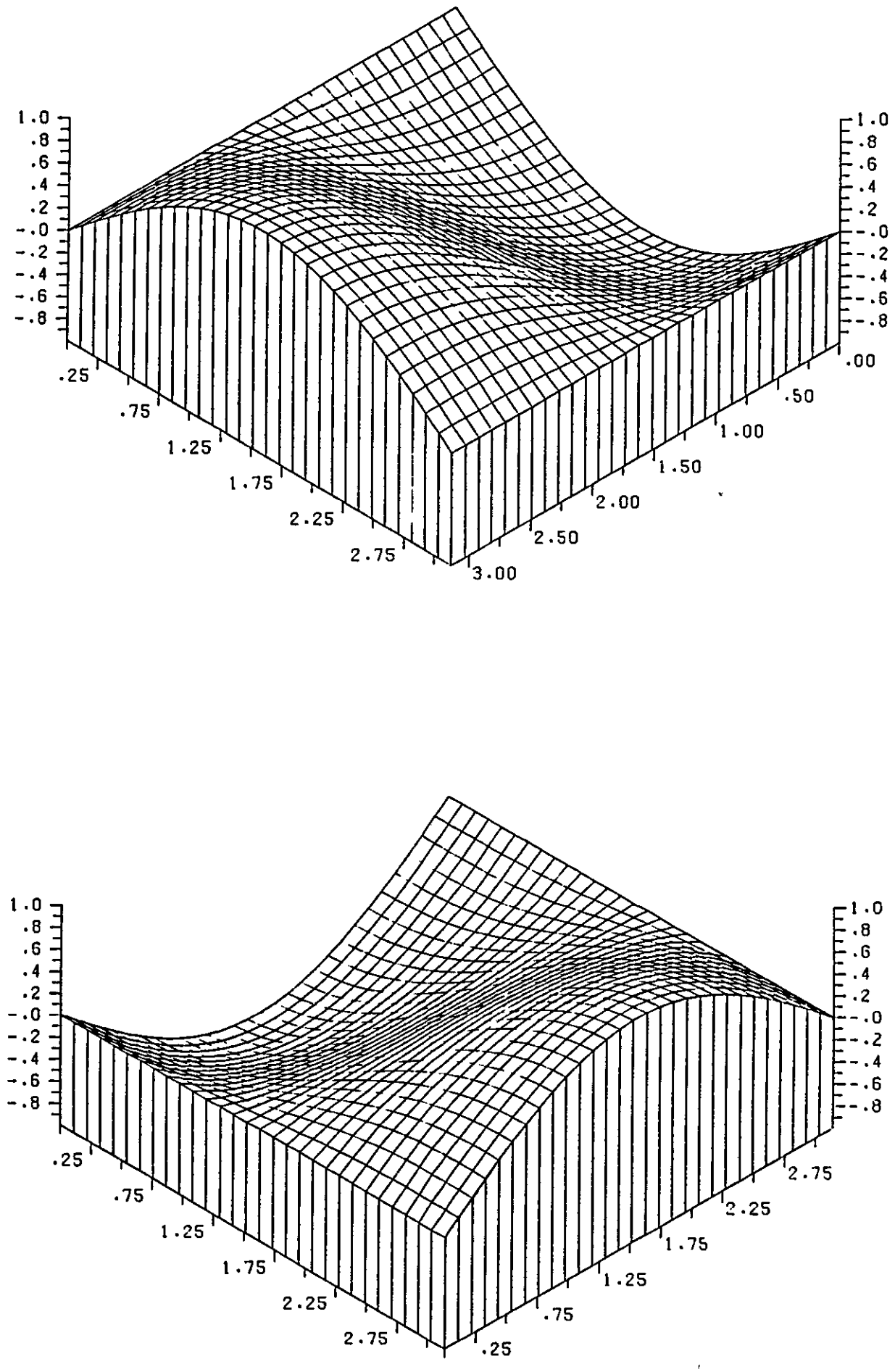


FIGURE 8.21: Isoparametric projection of the velocity u with different angles

CHAPTER NINE

CONCLUSIONS

In the foregoing chapters, the implementation of the finite element method has been studied on many different types of problems.

As a result of the research described in this thesis several general conclusions can be drawn in regard to the overall scope and use of the finite element procedure when applied to the problems discussed in the previous chapters. These conclusions are:-

- The accuracy of the finite element method will obviously depend upon how well the trial functions can approximate the true solution of a problem, the results confirm that if the approximating space is admissable and if the true solution u is smooth enough then an increase in the p version gives an equivalent decrease in the error bound.
- At various points we have presented the agreement of our finite element solutions with other numerical solutions and with the exact solution if available. We have also explored different sets of boundary conditions, the results obtained were extremely accurate.
- The ability of using finite elements of varying order (quadratic, cubic and quartic), i.e. the p version and that of increasing the numbers of elements, i.e. the h version has been investigated, from a practical point of view and the p version is found to be a better approximation for all the test problems.
- One of the difficulties associated with solving the free boundary problem is the almost total lack of any analytical results on convergence and error bounds. More is needed to be done in this field.

- The success of the mesh refinement technique for improving the accuracy of the numerical solution of a singularity region is evident, while the short comings of refining over the whole region is that many elements remote from the singularity are needlessly introduced so that the resulting master matrix becomes unnecessarily large. Thus, mesh generation and refinement for a fully automated finite element method to be used on a computer is very important so good mesh generation and refinement routines should be developed. Packages are now being produced which should improve this part of the algorithm; TWODEPEP has a good mesh generation and refinement strategy.
- Finally, it is likely that finite element programming systems will become more wider and economically written and hence easier to run, due to the development of new techniques in programming and the widespread introduction of software packages.

There remains a good deal of scope for work on the finite element method and especially so on free surface problems.

REFERENCES

AITCHISON, J.M., [1972]

Numerical treatment of a singularity in a free boundary problem,

Proc.Roy.Soc., London, A330: 573-580.

AITCHISON, J.M. [1977]

The numerical solution of minimization problem associated with a free surface flow,

J.Inst.Mach.Applics, 20: 33-44.

AITCHISON, J.M. [1979]

A variable finite element method for the calculation of flow over a weir,

RL-79-069, Rutherford Laboratory.

AITCHISON, J.M. [1980]

A finite element solution for critical flow over a weir,

Proc. 3rd International Conference of finite element in flow problems, Banff, Alberta, Canada.

ALLER, G.F. [1971]

Application of the method of integral relations to the solution of the quasilinear Goursat problem,

Azenbaidzan, Gos.Univ.Ulen.Zap.Ser.Fiz.Mat.Nauk., 2: 36-51.

AMES, W.F. [1977]

Numerical methods for partial differential equations,

2nd Ed., Nelson.

BABUSKA, I., DOOR, M.R. [1981]

Error estimates for combined h and p versions of finite element method,

Numer.Math. 37: 257-277.

BARNHILL, R.E. AND WHITEMAN, J.R. [1975]

Error analysis of Galerkin methods for Dirichlet problems containing boundary singularities,

J.Inst.Maths.Applics., 15: 121-125.

BARNHILL, R.E., BROWN, J.H., GREGORY, J.A. and MITCHELL, A.R. [1981]

Exact and approximate boundary data interpolation in the finite element method,

TR/01/81, Department of Mathematics, Brunel University.

BELL, G.E. and CRANK, J. [1973]

A method of treating boundary singularities in time dependent problems,

J.Inst.Maths.Applics. 12: 37-48.

BENJAMIN, T.B. [1956]

On the flow in channels when rigid obstacles are placed in the stream,

J. of Fluid Mechanics, 1: 227-248.

BERNAL, M.J.M. and WHITEMAN, J.R. [1970]

Numerical treatment of Biharmonic boundary value problems with re-entrant boundaries,

Computer J. 13: 87-91.

BIAOCCHI, C. [1972]

Annali.Mat.Pura.Appli. 92 pp.107-127.

CHRISTIANSEN, H.N. [1978]

The emerging role of color graphics,

Proc. of Second World Congress on Finite Element Methods,

Bournemouth, England.

CHUNG, T.J. [1978]

Finite element analysis in fluid dynamics,

McGraw-Hill, New York.

CONCUS, P. [1967]

Numerical solution of minimal surface equation,

Math.Comp. 21: 340-350.

CRANK, J., and OZIS, T. [1979]

Numerical solution of a free boundary problem by interchanging dependent and independent variables,

TR/90, Department of Mathematics, Brunel University.

CRANK, J. and FURZELAND, R.M. [1977]

The numerical solution of elliptic and parabolic partial differential equations with boundary singularities,

TR68, Department of Mathematics, Brunel University.

CRYER, C.W. [1976]

A survey of trial free boundary methods for the numerical solution of free boundary problems,

University of Wisconsin, Mathematics Research Centre, Tech.Summary Rep. No. 1693.

CRYER, C.W. [1970]

On the approximate solution of free boundary problems using finite differences,

J.Assoc.Comp.Mach. 17: 397-411.

CUTHILL, E. AND MCKEE, J. [1969]

Reducing the bandwidth of sparse symmetric matrices,

Proc. of the ACM National Conference: 157-172.

DAVIS, A.J. [1980]

The finite element method, a first approach,

Clarendon Press, Oxford.

DENIS, S.C.R. [1979]

Accurate representations of partial differential equations by finite difference schemes,

J.Inst.Mats.Applics. 23: 43-51.

ELSGOLTS, L. [1970]

Differential equations and the calculus of variations,

MIR Publishers, Moscow.

ERGATATOUDIS, I., IRONS, B.M., and ZIENKIEWICZ, O.C. [1968]

Curved isoparametric quadrilateral elements in finite element analysis,

Int.J. Solids Struct. 4:31-42.

EVANS, D.J. [1973a]

The analysis and application of sparse matrix algorithms in the finite element method,

The Maths. of Finite Elements and Applics., (Whiteman, J.R., ed.): 427-447, Academic Press, London.

FANGMEIER, D.D. and STRLKOFF, T.S. [1968]

Solution for gravity flow under a sluice gate,

Journal of the Engineering Mechanics Division, Proceeding of the American Society of Civil Engineers.

FICHTNER, W. and DONALD, J.R. [1981]

On the numerical solution of non-linear elliptic P.D.E.s arising from semi-conductor device modelling,

Elliptic Problem Solvers, Academic Press.

FINNEMORE, E.J. and PERRY, B. [1968]

Seepage through an earth dam computed by the relaxation technique,

Water Resources Res. 4: 1059-1067.

FLECHER, C.A.J. [1978]

The Galerkin method: an introduction,

Num.Sim. of Fluid Motion, pp.113-170.

FLECHER, R. [1980]

Practical methods of optimization, Vol. 1, unconstrained optimization,

John Wiley and Sons, Chichester.

FOX, L. [1971]

Some experiments with singularities in linear elliptic partial differential equations,

Proc.Roy.Soc., London A 323: 179-190.

FOX, L. and SANKAR, R. [1973]

The regular-falsi method for free boundary problems,

J.Inst.Math.Appl. 12: 49-54.

FORSYTHE, G. and WASOW, W.R. [1960]

Finite difference methods for partial differential equations,

Wiley, New York.

FURZELAND, R.M. [1977]

A survey of the formulation and solution of free and moving boundary (Stefan) problems,

TR/76, Department of Mathematics, Brunel University.

GLADWELL, I. and WAIT, R. (ed.) [1979]

A survey of numerical methods for partial differential equations,

Clarendon Press, Oxford.

GALLAGHER, R.H., NORRIE, D.H., ODEN, J.T. and ZIENKIEWICZ, O.C. [1982]

Finite elements in fluids - Vol. 4,

John Wiley and Sons.

GERALD, C.F. [1978]

Applied numerical analysis,

2nd Ed., Addison Wesley.

GOURLAY, A.R. and WATSON, G.A. [1973]

Computational methods for matrix eigenproblems

John Wiley and Sons.

HANGENEDER, E.W., and TAVOLATO, P. [1981]

A preprocessor for the finite element program SAP IV,

International Journal for Numerical Methods in Engineering, 17: 1779-1789.

HENRIC, P. [1964]

Elements of numerical analysis

New York, Wiley.

HINTON, E. and OWEN, D.R.J. [1979]

An introduction to finite element computations,

Pineridge Press Limited.

HOOD, P. [1976]

Frontal solution program for unsymmetric matrices,

International Journal for Numerical Methods in Engineering, 10:379-399.

HOUSTIS, E.N., MITCHELL, W.F. and PAPATHEODOROU, T.S., [1979]

A C^1-Collocation method for mildly non-linear elliptic equations on general 2-D domains,

Advances in Computer Methods for P.D.E.'s III: 18-27.

HUEBNER, K.H. [1975]

The finite element method for engineers,

John Wiley and Sons, New York.

ISAACSON, E. and KELLER, H.B. [1966]

Analysis of numerical methods,

John Wiley and Sons.

IRON, B.M. [1970]

Frontal solution program for finite element analysis,

International Journal for Numerical Methods in Engineering, Vol. 2, pp.5-32.

JAIN, M.K. [1978]

Numerical solution of differential equations,

Wiley Eastern Limited, New Delhi.

JENNINGS, A. [1977]

Matrix computation for engineers and scientists,

John Wiley.

LAROCK, B.E. and ASCE, A.M. [1969]

Gravity affected flow from planner sluice gates,

Journal of the Hydraulics Division Proceeding of the American Society of Civil Engineers.

LEVIN, D. and SIDERIDIS, A. [1977]

A collocation technique for certain singular harmonic mixed boundary value problems,

TR/73 Department of Mathematics, Brunel University.

MANSON, J.C. and FARKAS, I. [1972]

Continuous methods for free boundary problems,

Information Processing 71, pp.1305-1310, Amsterdam, North-Holland Publishing Co.

MARKLAND, E. [1965]

Calculation of the flow at a free overfall by relaxation method,

Proc.Inst. Civil Engineers, 31, Paper No. 686, pp.71-78.

MELOSH, R.J. [1963]

Basis for derivation of matrices for direct stiffness method,

J.A.I.A.A., 1, 1631-7.

MCCORQUODALE, J.A. and LI, C.Y. [1971]

Finite element analysis of sluice gate flow,

Transactions of the Engineering Institute of Canada, 14 No. C-2.

MITCHELL, A.R. [1969]

Computational methods in partial differential equations,

John Wiley and Sons.

MITCHELL, A.R., PHILIPS, G. and WACHSPRESS E. [1971]

Forbidden shapes in the finite element method,

J.Inst.Math.Applics. 8: 260-269.

MITCHELL, A.R. [1972]

Variational principles and the finite element method,

J.Inst.Math.Appl., 9: 378-389.

MITCHELL, A.R. [1973]

An introduction to the mathematics of the finite element method,

in J.R. Whiteman, ed. The Mathematics of Finite Elements and Applications, New York, Academic Press.

MITCHELL, A.R. and WAIT, R. [1977]

The finite element method in partial differential equations,

John Wiley and Sons.

MITCHELL, A.R. and GRIFFITH, D.F. [1980]

The finite difference method in partial differential equations,

John Wiley and Sons.

MOLER, C.B. [1965]

Finite difference methods for the eigenvalues of Laplace's operator,

Technical Report CS 22, Computer Science Department, Stanford Univ.

NICHELL, R.E., TANNER, R.I., CASWELL, B. [1974]

The solution of viscous incompressible jet and free-surface flow using finite element methods,

J.Fluid Mech. 65: 189-206.

NOOR, A.K. [1981]

Survey of computer programs for solution of non-linear structure and solid mechanics problems,

Computer and Structure 13: 425-465.

O'CARROLL, M.J. [1978]

Variational methods for free surface of cavitation, jets open channel flows, separation and wakes,

Chapter 16, Finite Elements in Fluids, John Wiley and Sons.

OCKENDON, H. AND TAYLER, A.B. [1979]

Inviscid Fluid Flows,

Mathematical Institute, Oxford University.

ODEN, J.T. and REDDY, J.N. [1976]

An introduction to the mathematical theory of finite elements,

John Wiley and Sons, New York.

ORTEGA, J.M. and RHEINBOLDT, W.C.]1970]

Iterative solution of nonlinear equations in several variables,

Academic Press, New York.

PAJER, G., [1937]

Über den "stromungsvorgang an einer unterstromten scharfkantigen planschutze",

Zeitschrift für Angewandte Mathematik und Mechanik, Vol. 17: 259-269.

PAPAMICHAEL, N. and WHITMAN, J.R. [1973]

A numerical conformal transformation method for harmonic mixed boundary value problems in polygonal domains,

Z. Angew.Math.Phys. (ZAMP) 24: 304-316.

PAPAMICHAEL, N. and SIDERIDIS [1978]

The use of conformal transformations for the numerical solution of elliptic boundary value problems with boundary singularities,

TR 74, Department of Mathematics, Brunel University.

PAPAMICHAEL, N. and WARBY, M.K. [1983]

Pole type singularities and the numerical conformal mapping of doubly-connected domains,

TR/02/83, Department of Mathematics, Brunel University.

PATTERSON, C. and SHEIKH, M.A. [1982]

A regular boundary element method for fluid flow,

International Journal for Numerical Methods in Fluids, 2: 239-251.

PERRY, B. [1957]

Methods for calculating the effect of gravity on two-dimensional free surface flows,

Dissertation presented to Stanford University, at Stanford Calif., in partial fulfilment of the requirement for the degree of Doctor of Philosophy.

PINDER, G.F. and GRAY, W.G. [1977]

Finite element simulation in surface and subsurface hydrology,

Academic Press, New York.

PHILLIPS, T.N. [1982]

Numerical solution of elliptic partial differential equations,

Oxford University, Ph.D. Thesis.

REID, J.K. and WALSH, J.E. [1965]

An elliptic eigenvalue problem for a re-entrant region,

SIAM.J.Appl.Math. 13: 837-850.

RAO, S.S. [1982]

The finite element in engineering,

Pergamon Press, Oxford.

SANKAR, R. [1967]

Numerical solution of differential equations,

D.Phil. Thesis, University of Oxford.

SCHITZ, A. and WAHLBIN, L. [1978]

Maximum norm estimates in the finite element method on plane polygonal domains, Part I,

Math.Comp. 32: 73-109.

SCHITZ, A. and WAHLBIN, L. [1979]

Maximum norm estimates in the finite element method on plane polygonal domains, Part II,

Math.Comp. 32: 465-462.

SIMMONS, G.F. [1963]

Introduction to Topology and Modern Analysis,

McGraw-Hill Book Company, New York.

ZLAMET, S. [1981]

Numerical simulation of semiconductor devices,

Report No. UIUCDCS-R-81-1072, Department of Computer Science,

University of Illinois at Urbana-Champaign, Urbana, Illinois 61801.

SMITH, G.D. [1978]

Numerical solution of partial differential methods,

Clarendon Press, Oxford.

SOUTHWELL, R.V. and VAISEY, G. [1946]

Relaxation methods applied to engineering problems XII. Fluid motions characterised by free streamlines,

Phil.Trans.Roy.Soc., A 240: 117-161

STRANG, G. and FIX, G. [1973]

An analysis of the finite element method

Prentice Hall, Englewood Cliffs.

STRANG, G. and BERGER, A.E. [1974]

The change in the solution due to change in domain,

Proceeding Amer.Math.Soc. Symposium 23.

STEWART, G.W. [1973]

Introduction to matrix computation,

Academic Press.

SYMM, G.T. [1973]

Treatment of singularities in the solution of Laplace's equation by an integral equation method,

Report NAC 31, National Physics Laboratory.

TONG, P. and ROSSETTOS, J.N. [1977]

Finite element method basic technique and implementation,

The MIT Press, Cambridge.

TORO, E.F. [1982]

Finite element computation of free surface problems,

Ph.D. Thesis, Teeside Polytechnic.

VAROGLU, E. and FINN, W.D.L. [1978]

Variable domain finite element analysis of free surface gravity flow

Computers and Fluids, 6: 103-114.

VARGA, R.S. [1962]

Matrix iterative analysis,

Prentice Hall, Englewood Cliffs, New Jersey.

VIGLEY, N.M. [1969]

On a method to subtract off a singularity at a corner for the Dirichlet or Neumann problem,

Mat.Comp. 23: 395-401.

VEMURI, V. and KARPLUS, W.J. [1981]

Digital computer treatment of partial differential equations,

Prentice Hall.

VICHNEVESTSKY, R. [1982]

Computer methods for partial differential equations,

Vol. 1, Prentice Hall.

WAIT, R. and MITCHELL, A.R. [1971]

Corner singularities in elliptic problems by finite element method,

J. of Computational Physics, 8: 45-52.

WHITEMAN, J.R. [1968]

Treatment of singularities in a harmonic mixed boundary value problem by dual series methods,

Quart.Journ.Mech. and Applied Math. Vol. XXI: 41-50.

WHITEMAN, J.R. and AKIN, J.E. [1978]

Finite element singularities and fracture,

35-41 of J.R. Whiteman (ed.), The Mathematics of Finite Elements and Applications III, Academic Press, London.

WHITEMAN, J.R. [1981]

Finite elements for singularities in two and three dimensions,

The Mathematics of Finite Elements and Application IV, Academic Press, London.

WILLIAMS, J.M. [1974]

An integral equation method for the computation of progressive gravity waves of finite height,

HRS, Report INT 136.

WOODS, L.C. [1953]

The relaxation treatment of singular points in Poissons equations,

Q.J.Mech.Appl.Math. 6: 163-189.

YANG, W.H. [1967]

On an integral equation solution for a plate with internal support,

Quar.J.Mech. and App.Maths. 21: 510-515.

YOUNG, D.M. [1971]

Iterative solution of large systems,

Academic Press, New York.

ZIENKIEWICZ, O.C. [1971]

The finite element method in engineering sciences,

McGraw-Hill, London.

ZIENKIEWICZ, O.C. [1977]

The finite element method,

Third Edition, McGraw-Hill, London.

APPENDIX

```
C **************
C       A SAMPLE OF OUR PROGRAM FOR OUR THREE DIMENSIONAL
C       SURFACE PLOTS, AND THE ISOPARAMETRIC PLOTS.
C  *************
C  ******
C     LIBRARY 'GINOGRAF'
C     LIBRARY 'GINOSURF'
C     LIBRARY 'GINO'
C     LIBRARY 'VAPPLB'
C     LIBRARY 'LUSUBV'
C ******
C
$INSERT SYSCOM A$KEYS
C  ******
C  ******
      DIMENSION X(900),Y(900),U(900),DX(900),DY(900)
     5,VX(900),VY(900),VV(900),AZ(50,50),W(5000),ERQ(900)
C ******
      CALL CLOS$A(1)
      CALL OPEN$A(1,'NAVSTOE2.OUTPUT2',16,1)
C ******
      WRITE(1,100)
100   FORMAT(2X,'1=T4010  2=VDU  3=TREND   4=SIGMA  5=PLOTEE
     *6=SE281'//)
C ******
       WRITE(1,199)
199    FORMAT(2X,'PLEASE SUPPLY DEVICE TEEMINAL'/)
C ******
      READ(1,*) IDIV
      IF(IDIV.EQ.1) GO TO 11
      IF(IDIV.EQ.2) GO TO 22
      IF(IDIV.EQ.3) GO TO 33
      IF(IDIV.EQ.4) GO TO 44
      IF(IDIV.EQ.5) GO TO 55
      IF(IDIV.EQ.6) GO TO 66
 11   CALL T4010
      GO TO 300
 22   CALL VDU
      GO TO 300
 33   CALL TREND
      GO TO 300
 44   CALL S5660
      GO TO 300
 55   CALL C1051N
C *****
C *****
C ******
      CALL ERRMAX(200)
      GO TO 300
66    CALL SE281
C
C ****
300   DO 400 I=1,1000
400   READ(5,*,END=500)X(I),Y(I),U(I),DX(I),DY(I),VX(I),VY(I),VV(I)
      CALL CHAMOD
      CALL PICCLE
```

```
500   NPOINT=I-1
      XMIN=X(1)
      YMIN=Y(1)
      NP=NPOINT
      WRITE(1,600) NP
 600  FORMAT(2X,I5)
C *****
C
C
C     COMPAIRING THE NUMERICAL SOL. WITH THE
C        THEORATIC SOL. AND FINDING THE
C          AVERIGE ERROR NORM.
C ******
C ******
C
      WRITE(1,30)
 30   FORMAT(/7X,'NUM AND THR SOL AND ERR',/)
      WRITE(1,40)
 40   FORMAT(5X,'X',10X,'Y',18X,'NUM.SOL',11X,'THE.SOL',11X,'ERROR',2X
     3,/4X,5('-'),5X,5('-'),9X,10('-'),9X,10('-'),7X,10('-'),2X)
C ******
C   THEORETICL SOL.
      ER1=0.0
      ER2=0.0
      THER1=0.0
      DO 50 J=1,NPOINT
      X1=X(J)
      Y1=Y(J)
      U1=U(J)
      THER1=(1./389.636364)*SIN(3.14169*X1)*SIN(3.14169*Y1)
      ER1=SQRT(ABS(U1-THER1)**2)
      ER2=ER2+ER1
      WRITE(1,70) X1,Y1,U1,THER1,ER1
 50   CONTINUE
      ER3=ER2/NPOINT
C *****
      WRITE(1,60)ER3
 60   FORMAT(//10X,'AVERAGE ERROR=',E13.5)
 70   FORMAT(2X,F8.4,1X,F8.4,2X,3E13.5,2X)
      DO 700 J=2,NP
      IF(XMIN.GT.X(J)) XMIN=X(J)
      IF(YMIN.GT.Y(J)) YMIN=Y(J)
 700   CONTINUE
      WRITE(1,800)XMIN,YMIN
 800  FORMAT(2X,2F9.5)
      DO 900 J=2,NP
      XMAX=X(1)
      YMAX=Y(1)
      IF(XMAX.LT.X(J)) XMAX=X(J)
      IF(YMAX.LT.Y(J)) YMAX=Y(J)
 900    CONTINUE
      WRITE(1,1000)XMAX,YMAX
 1000  FORMAT(2X,2F9.5)
C ******
      CALL WINDOW(3)
      CALL LEVELS(-1.0,1.000)
      CALL LABCON(0,1,3,0)
C  ******
```

```
      CALL RANGRD(NP,X,Y,U,30,XMIN,XMAX,30,YMIN,YMAX,AZ,4900,W)
      CALL DRACON(30,XMIN,XMAX,30,YMIN,YMAX,AZ,20,1,4900,W)
C  ******
      CALL CHAMOD
      READ(1,*)
      CALL PICCLE
      CALL RANGRD(NP,X,Y,U,30,XMIN,XMAX,30,YMIN,YMAX,AZ,4900,W)
C  ******
      CALL ISOPRJ(30,XMIN,XMAX,30,YMIN,YMAX,AZ,1,4900,W)
C  ******
      CALL CHAMOD
      READ(1,*)
      CALL PICCLE
      CALL RANGRD(NP,X,Y,U,30,XMIN,XMAX,30,YMIN,YMAX,AZ,4900,W)
      CALL ISOPRJ(30,XMIN,XMAX,30,YMIN,YMAX,AZ,2,4900,W)
      CALL CHAMOD
      CALL PICCLE
      CALL CHAMOD
      READ(1,*)
      CALL DEVEND
      CALL CLOS$A(1)
      CALL EXIT
      END
```

```
****  THE FOLLOWING PAGES DESCRIBES THE USER INPUT DATA SET,
****    WHICH COMPLETELY SPECIFIES THE TEST PROBLEMS
****     SOLVED PY TWODEPEP.
****
****        CHAPTER 3
****        SECTION(3-7)
****
****    THE FIRST LINE CONTAINS 3 INTEGERS-NEQ,NTF,NDIM IN FREE
****    FORMAT,WHERE
****    NT=NUMBER OF TRIANGLES IN THE INITIAL TRIANGULATIONS
****    NTF=NUMBER OF TRIANGLES DESIRED IN FINAL TRIANGULATION
****    NDIM= RESERVED FOR JACOBIAN IF NDIM=1 IN-CORE STORAGE
****     ONLY USED, AND IF NDIM=2 OUT-OF CORE STORAGE USED
****
 1    75   1
****
****   THE P.D.E
****
OXX          UX
OXX/UX       1.0
OXY          UY
OXY/UY       1.0
****
****
****  THE SOLUTION WILL BE OUTPUT AT THE POINTS OF THE
****  GRID,
****        X=XA +I*HX    I=0,...,NX
****        Y=YA +J*HY    J=0,...,NY
****
XA           0.0
HX           0.1
NX           10
****
YA          -1.0
HY           0.2
NY           10
****
****    PRINTER PLOT OF THE INITIAL TRIANGULATION WILL BE PLOTED
PLOT         1
****
****    THE PROBLEM IS SYMMETRIC
SYMMETRY     1
****
****
****    USING CUBIC ISOPARAMETRIC TRIANGULAR ELEMENTS
****
CUBICS       1
****
****    THE BOUNDEARY CONDITONS
ARC=-1001
FB1          0.0
ARC=-1002
FB1          0.207879576*DCOS(1.5707963*Y)
ARC=-1003
FB1          0.0
ARC=-1004
FB1          DCOS(1.5707963*Y)
****
```

```
****    INITIAL TRIANGULATION ARRAYS
****    THE COORDINATES OF THE VERTICES OF THE
****    TRIANGULATION IN THE FORM
****     VX(1),VY(1),...,VX(NV),VX(NV)
VXY       0.,-1., 1.,-1., 1.,1., 0.,1., 0.5,0.
****
****   LIST THE NUMBERS OF THE VERTICES OF EACH TRANGLE IN
****    IA(1),IB(1), IC(1),...,IA(NT),IB(NT),IC(NT)
****    THIS ORDER DEFINES THE INITIAL TRIANGLE NUMBERS.
IABC       1,2,5, 2,3,5, 3,4,5, 4,1,5
****
****  AN IDENTIFYING INTEGER OF THE BOUNDARY ARC CUT OFF BY
****  THE BASE,AB,OF TRIANGLE K. I(K)=0 IF NONE.
I          -1001,-1002,-1003,-1004
END.
****
```

```
****
****    CHAPTER 5.
****    SECTION(5-4)
****    TEST PROBLEM 1
****
****    FREE SURFACE PROBLEM FOR THE SLUICE GATE PROBLEM
****    WITH FLOW GEOMETRY B/H=0.4
****
****
1    300   1
****
OXX          UX
OXX/UX       1.0
OXY          UY
OXY/UY       1.0
****
XA          -1.0
HX           0.2
NX           10
YA          0.0
HY          0.1
NY          10
****
****  THE FREE SURFACE BOUNDARY
ARC=-1001
X           1.0-DCOS(1.57079*S)
Y           0.4-0.16016*DSIN(1.57079*S)
FB1         0.35996
ARC=1002
GB1         0.0
ARC=-1003
FB1         0.0
ARC=1004
GB1         0.0
ARC=-1005
FB1         0.35996
ARC=-1006
FB1         0.35996
****
****
SYMMETRY       1
****
PLOT           1
****
VXY       0.0,0.4, 1.0,0.23984, 1.0,0.0, 0.0,0.0,
VXY       -1.0,0.0, -1.0,0.4, -1.0,1.0, 0.0,1.0,
VXY       -0.5,0.70, -0.5,0.2, 0.0,0.1
****
IABC      2,1,11, 3,2,11, 4,3,11, 4,11,10, 11,1,10, 5,4,10,
IABC      6,5,10, 1,6,10, 6,1,9, 1,8,9, 8,7,9, 7,6,9
****
I         -1001, 1002, -1003, 0, 0,
I         -1003, 1004, 0, 0, -1006,
I         -1005, 1004
END.
```

```
****
****    CHAPTER 5.
****    SECTION(5-4)
****    TEST PROBLEM 2
****
****    THE FREE SURFACE PROBLEM OF THE SLUICE GATE PROBLEM
****    WITH FLOW GEOMETRY B/H=0.36
****
1    300   1
****
OXX          UX
OXX/UX       1.0
OXY          UY
OXY/UY       1.0
****
XA          -1.0
HX           0.1
NX           20
YA           0.0
HY           0.05
NY           20
****
****
ARC=-1001
X            1.0-DCOS(1.57079*S)
Y            0.36-0.143354769*DSIN(1.57079*S)
FB1          0.31995
ARC=1002
GB1          0.0
ARC=-1003
FB1          0.0
ARC=1004
GB1          0.0
ARC=-1005
FB1          0.31995
ARC=-1006
FB1          0.31995
****
****
SYMMETRY       1
****
PLOT           1
****
VXY      0.0,0.36, 1.0,0.21664523, 1.0,0.0, 0.0,0.0,
VXY      -1.0,0.0, -1.0,0.4, -1.0,1.0, 0.0,1.0,
VXY      -0.5,0.70, -0.5,0.2, 0.0,0.1
****
IABC     2,1,11, 3,2,11, 4,3,11, 4,11,10, 11,1,10, 5,4,10,
IABC     6,5,10, 1,6,10, 6,1,9, 1,8,9, 8,7,9, 7,6,9
****
I        -1001, 1002, -1003, 0, 0,
I        -1003, 1004, 0, 0, -1006,
I        -1005, 1004
END.
```

```
****
****      CHAPTER 6
****
****      SECTION(6-4)
****      TEST PROBLEM 1.
****      MOTZ= PROBLEM
****      CORNER SINGULARITIES IN ELLIPTIC PROBLEMS.
****
1    300    1
****
OXX        UX
OXY        UY
****
XA          -1.0
HX         0.28571428
NX          7
YA          0.0
HY         0.314159265
NY          10
****
****      MORE FINER ELEMENTS AROUND THE SINGULAR POINT O.
****
D3EST           1.0/(X**2+Y**2)
****
****
ARC=-1002
FB1          1000.0
****
ARC=-1005
FB1           500.0
****
****
CUBICS         1
****
SYMMETRY       1
****
****
VXY          0.,0., 1.,0., 1.,1., 0.,1.,
VXY         -1.,1., -1.,0., .5,.5, -.5,.5
****
IABC          1,2,7, 2,3,7, 3,4,7, 4,1,7,
IABC          1,4,8, 4,5,8, 5,6,8, 6,1,8
****
I             1001, -1002, 1003, 0,
I             0, 1003, 1004, -1005
END.
```

```
****      CHAPTER 6.
****      SECTION(6-4)
****      TEST PROBLEM 2.
****
****
****      NUMERICAL SOLUTION OF ELLIPTIC BOUNDARY
****      VALUE PROBLEM WITH BOUNDARY SINGULARITY
****
****
****
1   300    2
****
OXX         UX
OXY         UY
****
F1          -2.0*U
F1/U        -2.0
****
XA          0.0
HX          0.1
NX          10
****
YA          0.0
HY          0.05
NY          20
****
****
ARC=-1
X           DCOS(.785398*S)
Y           DSIN(.785398*S)
FB1         0.2*(DEXP(X+Y))
****
ARC=-1001
FB1          0.2*(DEXP(X+Y))
****
****
D3EST        1.0/(X**2+Y**2)
****
SYMMETRY     1
****
CUBICS       1
****
PLOT         1
****
VXY         0.0,0.0, 1.0,0.0, 0.7071069,0.7071069, 0.5,0.25
****
IABC        1,2,4, 2,3,4, 3,1,4
****
I            -1001,-1, -1001
END.
```

```
****
****      CHAPTER 6.
****      SECTION(6-4)
****
****      TEST PROBLEM 3
****
****      NUMERICAL SOLUTION OF ELLIPTIC BOUNDARY
****      VALUE PROBLEM WITH BOUNDARY SINGULARITY
****
1   300   1
****
OXX             UX
OXY             UY
****
F1            (16.*X*X+1.)*U-4.*DCOS(2.*X*X-Y)
F1/U          (16.*X*X+1.)
****
XA           -1.0
HX            0.2
NX            10
****
YA            -1.0
NY             10
HY             0.2
****
ARC=-1001
FB1           DSIN(2.0*X**2-Y)
****
ARC=-1
X             DCOS(4.71239*S)
Y             DSIN(4.71239*S)
FB1           DSIN(2.0*X**2-Y)
****
ARC=1002
GB1             0.0
****
SYMMETRY         1
****
CUBICS           1
****
PLOT             1
****
****
VXY           0.0,0.0, 1.0,0.0, 0.0,1.0, -1.0,0.0,
VXY           0.0,-1.0, 0.25,0.25, -0.25,0.25, -0.25,-0.25
****
IABC          1,2,6, 2,3,6, 3,1,6,
IABC          1,3,7, 3,4,7, 4,1,7,
IABC          1,4,8, 4,5,8, 5,1,8
****
I              -1001, -1, 0,
I                0, -1, 0
I               0, -1, 1002
END.
```

```
****
****    CHAPTER 6
****    SECION(6-4)
****    TEST PROBLEM 4.
****     NUMERICAL SOLUTION OF ELLIPTIC BOUNDARY
  ****  VALUE PROBLEM WITH BOUNDARY SINGULARITY.
****
1   300   1
****
OXX             UX
OXY             UY
****
XA              0.0
HX              0.1
NX              10
YA              0.0
HY              0.1
NY              10
****
****
ARC=1001
GB1             0.0
ARC=-1002
FB1             0.0
ARC=1003
GB1             0.0
ARC=1004
GB1             0.0
ARC=1005
GB1             0.0
ARC=-1006
FB1             1.0
****
SYMMETRY         1
****
D3EST            1./((X-.5)**2+(Y-.5)**2)
****
****
PLOT             1
****
VXY             0.0,0.0,0.5,0.0,1.0,0.0,1.0,0.5,0.5,0.5,0.5,1.0,
VXY             0.0,1.0,0.0,0.5,0.25,0.25,0.75,0.25,0.25,0.75
****
IABC             1,2,9, 2,5,9, 5,8,9, 8,1,9,
IABC             2,3,10, 3,4,10, 4,5,10, 5,2,10,
IABC             8,5,11, 5,6,11, 6,7,11, 7,8,11
****
I                  1001, 0, 0, -1006,
I                  1001, -1002, 1003, 0,
I                  0, 1004, 1005, -1006
END.
```

```
****
****       CHAPTER 7.
****       SECTION(7-1)
THE MINIMAL SURFACE PROBLEM
 1   300   2
****
OXX              UX/SQ(UX,UY)
OXX/UX           (1.+UY**2)/SQ(UX,UY)**3
OXY              UY/SQ(UX,UY)
OXY/UX           -UX*UY/SQ(UX,UY)**3
OXY/UY           (1.+UX**2)/SQ(UX,UY)**3
****
TF                4
****
XA               0.0
HX               0.1
NX               10
YA               0.0
HY               0.1
NY               10
****
ARC=-1001
FB1              (1.0-X**2)**(1./2.)
ARC=-1002
FB1               FFU(X,Y)
ARC=-1003
FB1              (2.381097845-X**2)**(1./2.)
ARC=-1004
FB1              (DEXP(Y)+DEXP(-Y))/2.0
****
SYMMETRY          1
****
CUBICS            1
****
ALPHA             2
****
VXY          0.0,0.0, 1.0,0.0, 1.0,1.0, 0.0,1.0, 0.5,0.5
****
IABC          1,2,5, 2,3,5, 3,4,5, 4,1,5
****
I             -1001, -1002,-1003,-1004
****
ADD.
      DOUBLE PRECISION FUNCTION SQ(UX,UY)
      IMPLICIT DOUBLE  PRECISION (A-H,O-Z)
      SQ=DSQRT(1.+(UX**2+UY**2))
      RETURN
      END
      DOUBLE PRECISION  FUNCTION FFU(X,Y)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      IF(X.EQ.1.0.AND.Y.EQ.0.0) FFU=0.0
      IF(X.EQ.1.0.AND.Y.GT.0.0) FFU=DSQRT(0.25*(DEXP(Y)+DEXP(-Y)
     4)**2-X**2)
      RETURN
      END
END.
```

```
****
****     CHAPTER 7.
****     SECTION(7-3)
****     TEST PROBLEM 1
****
****       COMMENTS:-
****    NON CONSTANT COEFFICIENT NON-LINEARITIES IN FIRST
****                 DERIVATIVES
****
1    300    1
****
OXX          UX
OXX/UX       1.0
OXY          UY
OXY/UY       1.0
****
F1          (2.-DSIN(Y)*DCOS(X))*U-(UX*UY)
F1/U        (2.-DSIN(Y)*DCOS(X))
F1/UX       -UY
F1/UY       -UX
****
TF           8.
****
NOUT         4
****
CUBICS       1
****
XA           0.0
HX           0.1
NX           10
YA           0.0
HY           0.1
NY           10
****
ALPHA        2
****
U0           DSIN(X)*DCOS(Y)
****
****
ARC=-1001
FB1          DSIN(X)
ARC=-1002
FB1          DSIN(X)*DCOS(Y)
ARC=-1003
FB1          DSIN(X)*DCOS(Y)
ARC=-1004
FB1          0.0
****
SYMMETRY      1
****
****
VXY         0.0,0.0, 1.0,0.0, 1.0,1.0, 0.0,1.0,0.5,0.5
****
IABC           1,2,5, 2,3,5, 3,4,5, 4,1,5
****
I               -1001,-1002,-1003,-1004
****
END.
```

```
****
****      CHAPTER 7.
****      SECTION(7-3)
****      TEST PROBLEM 2
****
****
****        COMMENTS:-
****      NON-LINEARITIES IN SOLUTION AND FIRST DERIVATIVES
****        OF SOLUTION NON HOMOGENEOUS BOUNDARY CONDITIONS.
****
1     50     1
****
OXX             UX
OXX/UX          1.0
OXY             UY
OXY/UY          1.0
****
F1             -U*(UX+UY)*(DEXP(-(X+Y)))
F1/U           -(UX+UY)*(DEXP(-(X+Y)))
F1/UX          -U*(DEXP(-(X+Y)))
F1/UY          -U*(DEXP(-(X+Y)))
****
TF              8.
****
NOUT            8
****
CUBICS          1
****
XA              0.0
HX              0.1
NX              10
YA              0.0
HY              0.1
NY              10
****
ARC=-1001
FB1               DEXP(X)
ARC=-1002
FB1               DEXP(Y+1)
ARC=-1003
FB1               DEXP(X+1)
ARC=-1004
FB1               DEXP(Y)
****
SYMMETRY        1
****
ALPHA           2
****
VXY            0.0,0.0, 1.0,0.0, 1.0,1.0, 0.0,1.0,0.5,0.5
****
IABC              1,2,5, 2,3,5, 3,4,5, 4,1,5
****
I                  -1001,-1002,-1003,-1004
END.
```

```
****
****      CHAPTER 7
****      SECTION(7-3)
****      TEST PROBLEM 3
****
****        COMMENTS:-
****    NON CONSTANT COEFFICIENT NON-LINEARITIES IN FIRST
****                    DERIVATIVES
1     50     1
****
OXX             UX
OXX/UX          1.0
OXY             UY
OXY/UY          1.0
****
F1          197.392088*DSIN(6.283185*X)*DSIN(12.56637*Y)+FUN(X,Y,U)
F1/U        -DEXP(U)
****
TF              4.
****
NOUT            4
****
****
XA              0.0
HX              0.05
NX              10
YA              0.0
HY              0.025
NY              10
****
ARC=-1001
FB1             0.0
ARC=-1002
FB1             0.0
ARC=-1003
FB1             0.0
ARC=-1004
FB1             0.0
****
SYMMETRY        1
****
U0              DSIN(6.283185*X)*DSIN(12.5663706*Y)
****
QUARTICS          1
****
VXY         0.0,0.0, 0.5,0.0, 0.5,0.25, 0.0,0.25, 0.25,0.125
****
IABC              1,2,5, 2,3,5, 3,4,5, 4,1,5
****
I                  -1001,-1002,-1003,-1004
****
ADD.
      DOUBLE PRECISION FUNCTION FUN(X,Y,U)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      FUN=DEXP(DSIN(6.2831853*X)*DSIN(12.566371*Y))-DEXP(U)
      RETURN
      END
****
END.
```

```
****
****      CHAPTER 7
****      SECTION(7-3)
****      TEST PROBLEM 4
****      COMMENTS:-
****      THE VALUE OF F(X,Y) IS DETERMINED SO THAT THE TRUE
****      SOLUTION IS CORRECT.NONHOMOGENEOUS BOUNDARY CONDITIONS
****      OSCILLATORY SOLUTION.
****
****
1    100   1
****
OXX          UX
OXX/UX      1.0
OXY         UY
OXY/UY      1.0
F1         -U/(U+10)+FF(X,Y)
F1/U        U/(U+10.)**2-1./(U+10.)
****
TF            3.
****
XA          0.0
HX          0.2
NX            5
YA          0.0
HY          0.2
NY           5
****
PLOT         1
****
****
ARC=-1001
FB1         1.+DSIN(3.14159*X)
ARC=-1002
FB1         DCOS(3.14159*Y)+DSIN(3.14159*(1-Y))
ARC=-1003
FB1        -1.+DSIN(3.14159*(X-Y))
ARC=-1004
FB1         DCOS(3.14159*(Y))-DSIN(3.14159*Y)
****
SYMMETRY      1
****
CUBICS        1
****
VXY         0.0,0.0, 1.0,0.0, 1.0,1.0, 0.0,1.0,0.5,0.5
****
IABC           1,2,5, 2,3,5, 3,4,5, 4,1,5
****
I               -1001,-1002,-1003,-1004
****
ADD.
      DOUBLE PRECISION FUNCTION FF(X,Y)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      FF=9.8696*(2.*DSIN(3.14159*(X-Y))+DCOS(3.14159*Y))
     *+(DCOS(3.14159*Y)+DSIN(3.14159*(X-Y)))/(DCOS(3.141
     *59*Y)+DSIN(3.14159*(X-Y))+10.)
      RETURN
      END
END.
```

```
****
****       CHAPTER 7
****       SECTION(7-4)
****
****
****     SEMICONDUCTOR PROBLEM
****
****
3     100    2
****
****         SEMICONDUCTOR PROBLEM
****
****         POISSON EQUATION
****
O1X              U1X
O1Y              U1Y
F1               B(T)*(-DEXP(U1-U2)+DEXP(U3-U1))
F1/U1            B(T)*(-DEXP(U1-U2)-DEXP(U3-U1))
F1/U2            B(T)*(DEXP(U1-U2))
F1/U3            B(T)*(-DEXP(U3-U1))
****
****           CONTINUITY (DIFFUSION) EQUATIONS
****
O2X              (DEXP((U1-U2)*B(T)))*U2X*0.588
O2X/U1           B(T)*(DEXP((U1-U2)*B(T)))*U2X*0.588
O2X/U2          -B(T)*(DEXP((U1-U2)*B(T)))*U2X*0.588
O2Y              (DEXP((U1-U2)*B(T)))*U2Y*0.588
O2Y/U1           B(T)*(DEXP((U1-U2)*B(T)))*U2Y*0.588
O2Y/U2          -B(T)*(DEXP((U1-U2)*B(T)))*U2Y*0.588
O3X              (DEXP((U3-U1)*B(T)))*U3X*0.0588
O3X/U1          -B(T)*(DEXP((U3-U1)*B(T)))*U3X*0.0588
O3X/U3           B(T)*(DEXP((U3-U1)*B(T)))*U3X*0.0588
O3Y              (DEXP((U3-U1)*B(T)))*U3Y*0.0588
O3Y/U1          -B(T)*(DEXP((U3-U1)*B(T)))*U3Y*0.0588
O3Y/U3           B(T)*(DEXP((U3-U1)*B(T)))*U3Y*0.0588
****
****           BOUNDARY CONDITIONS
ARC=-1001
FB1          0.0
FB2          0.0
FB3          0.0
****
ARC=1002
GB1          0.0
GB2          0.0
GB3          0.0
****
ARC=-1003
FB1          60
FB2          0.0
FB3          0.0
****
ARC=1004
GB1          0.0
GB2          0.0
GB3          0.0
****
```

```
ARC=-1005
FB1          0.0
FB2          0.0
FB3          0.0
****
ARC=1006
GB1          0.0
GB2          0.0
GB3          0.0
****
****
PLOT          1
****
TF           15
****
NOUT          5
****
NX           15
NY           8
****
U0           1.0
****
ALPHA        2
****
VXY         0.0, 0.0, 0.4,0.0, 2.40,0.0,
VXY         2.8,0.0, 2.8,0.46, 2.40,0.46,
VXY         0.40,0.46, 0.0,0.46,0.20,0.23,
VXY         1.40,0.23, 2.60,0.23
****
IABC        1,2,9, 2,7,9, 7,8,9, 8,1,9,
IABC        2,3,10, 3,6,10, 6,7,10, 7,2,10,
IABC        3,4,11, 4,5,11, 5,6,11, 6,3,11,
****
I           -1001, 0, -1005, 1006,
I           -1001, 0, 1004, 0,
I           -1001, 1002, -1003, 0
****
ADD.
        DOUBLE PRECISION FUNCTION B(T)
C       FUNCTION B VARIES FROM 0 TO 1 TO GRADUALLY INCREASE
C       THE DIFFICUILTY OF THE PROBLEM
        DOUBLE PRECISION T
        B=0.05*(T-1)
        B=DMIN1(B**2,1.0)
        RETURN
        END
END.
```

```
****
****      CHAPTER 8
****      SECTION(8-3)
****      TEST PROBLEM 1
****      CASE 1
****
****     THE FIRST LINE CONTAINS 3 INTEGERS-NEQ,NTF,NDIM IN FREE
****     FORMAT,WHERE
****     NT=NUMBER OF TRIANGLES IN THE INITIAL TRIANGULATIONS
****     NTF=NUMBER OF TRIANGLES DESIRED IN FINAL TRIANGULATION
****     NDIM= RESERVED FOR JACOBIAN IF NDIM=1 IN-CORE STORAGE
****      ONLY USED, AND IF NDIM=2 OUT-OF CORE STORAGE USED
****
1     300   1
****
****    THE P.D.E
****
OXX           UX
OXX/UX        1.0
OXY           UY
OXY/UY        1.0
****
****
****   THE SOLUTION WILL BE OUTPUT AT THE POINTS OF THE
****   GRID,
****        X=XA +I*HX    I=0,...,NX
****        Y=YA +J*HY    J=0,...,NY
****
XA            0.0
HX            0.1
NX            10
****
YA            -1.0
HY            0.2
NY            10
****
****     PRINTER PLOT OF THE INITIAL TRIANGULATION WILL BE PLOTED
PLOT          1
****
****     THE PROBLEM IS SYMMETRIC
SYMMETRY      1
****
****
****     USING CUBIC ISOPARAMETRIC TRIANGULAR ELEMENTS
****
CUBICS        1
****
****     THE BOUNDEARY CONDITONS
ARC=1001
Y              DSIN(1.570796*S)
X             -DSQRT(2.0)*DCOS(1.570796*S)
GB1            0.0
ARC=-1002
FB1            0.0
ARC=1003
GB1            0.0
ARC=1004
GB1            1.0
ARC=1005
GB1            0.0
****
```

```
****    INITIAL TRIANGULATION ARRAYS
****   THE COORDINATES OF THE VERTICES OF THE
****   TRIANGULATION IN THE FORM
****     VX(1),VY(1),...,VX(NV),VX(NV)
VXY          -1.414213356,0., -.99999969,.707107,
VXY           0.,1., 0.,2., -2.,2., -4.,2., -4.,0.,
VXY           -2.,0., -1.5,1.25, -2.,1., -3.,1.
****
****   LIST THE NUMBERS OF THE VERTICES OF EACH TRANGLE IN
****    IA(1),IB(1), IC(1),...,IA(NT),IB(NT),IC(NT)
****    THIS ORDER DEFINES THE INITIAL TRIANGLE NUMBERS.
IABC         1,2,9, 2,3,9, 3,4,9, 4,5,9, 5,10,9, 5,11,10,
IABC         5,6,11, 6,7,11, 7,8,11, 11,8,10, 10,8,9,
IABC         8,1,9
****
****  AN IDENTIFYING INTEGER OF THE BOUNDARY ARC CUT OFF BY
****  THE BASE,AB,OF TRIANGLE K. I(K)=0 IF NONE.
I            1001,1001, -1002, 1003, 0, 0,
I            1003, 1004, 1005, 0, 0, 1005
****
END.
****
```

```
****      CHAPTER 8.
****      SECTION(8-3)
****      TEST PROBLEM 1
****      CASE 2
****      POTENTIAL FLOW PROBLEM
****       PROGRAM NAME ILFX3SI.INPUT
****
1     300   1
****
OXX          UX
OXX/UX       1.0
OXY          UY
OXY/UY       1.0
****
****
XA          -4.0
HX           0.2
NX           20
****
YA           0.0
HY           0.2
NY           10
****
SYMMETRY      1
****
****
ARC=1001
X           -DCOS(1.570796*S)
Y            DSQRT(2.0)*DSIN(1.570796*S)
GB1          0.0
ARC=-1002
FB1          0.0
ARC=1003
GB1          0.0
ARC=1004
GB1          1.0
ARC=1005
GB1          0.0
****
VXY         -1.0,0, -0.707107,0.99999969, 0.,1.414213562,
VXY          0.,2., -2.,2., -4.,2., -4.,0., -2.,0.,
VXY         -1.5,1.55, -2.,1., -3.,1.
****
IABC         1,2,9, 2,3,9, 3,4,9, 4,5,9, 5,10,9, 5,11,10,
IABC         5,6,11, 6,7,11, 7,8,11, 11,8,10, 10,8,9,
IABC          8,1,9
****
I             1001,1001, -1002, 1003, 0, 0,
I             1003, 1004, 1005, 0, 0, 1005
END.
```

```
****
****     CHAPTER 8.
****    TEST PROBLEM 3.
****
****    INVISCID LAMINAR FLOW IN A CHANNEL
****          PAST A DISC .
****
1     300        1
****
OXX          UX
OXY          UY
****
NX            20
****
NY            10
****
SYMMETRY       1
****
PLOT           1
****
****
D3EST        1.0/(X**2+(Y-1.)**2)
****
****
ARC=-1001
FB1            0.0
ARC=-1002
FB1            0.0
ARC=1003
GB1            0.0
ARC=-1004
FB1            2.0
ARC=-1005
FB1            Y
****
CUBICS         1
****
VXY          0.,0., 0.,1., 0.,2., -1.,2., -2.,2.,
VXY          -2.,1., -2.,0., -1.,0., -1.,1.
****
IABC         1,2,9, 2,3,9, 3,4,9, 4,5,9,
IABC         5,6,9, 6,7,9, 7,8,9, 8,1,9
****
I            -1002, 1003, -1004, -1004,
I            -1005, -1005, -1001, -1001
END.
```

```
****
****      CHAPTER 8.
****      SECTION(8-2)
****
****      THE BIHARMONIC PROBLEM
****      A RECTANGULAR PLATE PROBLEM.
****
****
2    74     2
****
OXX          VX
OXX/VX       1.
OXY          VY
OXY/VY        1.
OYX          UX
OYX/UX        1.
OYY          UY
OYY/UY        1.
F1           -SIN(3.1416*X)*SIN(3.1416*Y)
F2           -V
F1/V         -1.
****
****
ARC=-1001
FB1          0.0
FB2          0.0
ARC=-1002
FB1          0.0
FB2          0.0
ARC=-1003
FB1          0.0
FB2          0.0
ARC=-1004
FB1          0.0
FB2          0.0
****
****
XA           0.0
HX           0.1
NX           20.0
YA           0.0
HY           0.1
NY           10.0
****
SYMMETRY      1
****
CUBICS        1
****
****
VXY          0.0,0.0, 1.0,0.0, 1.0,1.0, 0.0,1.0, 0.5,0.5
****
IABC            1,2,5, 2,3,5, 3,4,5, 4,1,5
****
I               -1001, -1002, -1003, -1004
END.
```

```
****
****
****        CHAPTER 8.
****       SECTION(8-4)
****       FIGURE(8-9)
****       TEST PROBLEM 1.
****
****
****      THE EIGEN-VALUE PROBLEM
****
1     300      1
****
OXX            UX
OXY            UY
****
F1             USET(1)
****
****
U0             1
****
****
****
XA             0.0
HX             0.5
NX             2
****
YA             0.0
HY             0.5
NY              2
****
TF             15.
****
NORMAL         1
****
ARC=-1001
FB1            0.0
ARC=-1002
FB1            0.0
ARC=-1003
FB1            0.0
ARC=-1004
FB1            0.0
ARC=-1005
FB1            0.0
ARC=-1006
FB1            0.0
ARC=-1006
FB1            0.0
ARC=-1007
FB1            0.0
ARC=-1008
FB1            0.0
****
SYMMETRY       1
****
CUBICS         1
****
D3EST          1./(X**2+Y**2)
****
```

```
NUPDT           0
****
****
VXY         0.0,0.0, 0.0,1.0, -1.0,1.0, -1.0,0.0,
VXY        -1.0,-1.0, 0.0,-1.0, 1.0,-1.0, 1.0,0.0,
VXY        -0.5,0.5, -0.5,-0.5, 0.5,-0.5
****
IABC       1,2,9, 2,3,9, 3,4,9, 4,1,9,
IABC       1,4,10, 4,5,10, 5,6,10, 6,1,10,
IABC       1,6,11, 6,7,11, 7,8,11, 8,1,11
****
I           -1001, -1002, -1003, 0,
I            0, -1004, -1005, 0,
I            0, -1006, -1007, -1008
END.
```

```
****
****    CHAPTER 8.
****    SECTION(8-4)
****    FIGURE(8-12)
****     TEST PROBLEM 1
****
****    THE EIGEN-VALUE PROBLEM
****
1     300     1
****
OXX           UX
OXY           UY
****
F1           USET(1)
****
U0           1
****
D3EST        1./(X**2+Y**2)
****
NORMAL
****
NX           12
NY           12
****
TF           9.
****
NOUT         3
****
ARC= -1001
FB1           1.0
ARC= -1002
FB1           1.0
ARC=-1004
FB1           1.0
ARC=-1005
FB1          1.0
****
SYMMETRY      1
****
NUPDT         0
****
VXY           0.0,0.0,0.5,0.0,1.0,0.0,1.0,0.5,0.5,0.5,0.5,1.0,
VXY           0.0,1.0,0.0,0.5,0.25,0.25,0.75,0.25,0.25,0.75
****
IABC           1,2,9, 2,5,9, 5,8,9, 8,1,9,
IABC           2,3,10, 3,4,10, 4,5,10, 5,2,10,
IABC           8,5,11, 5,6,11, 6,7,11, 7,8,11
****
I             -1001, 0, 0, -1002,
I             -1001, 1003, -1004, 0,
I             0, -1005, 1006, -1002
END.
```

```
****
****
****     CHAPTER 8.
****    SECTION(8-4)
****    TEST PROBLEM 2
****    FIGURE(8-15)
****
****    THE EIGEN-VALUE PROBLEM
****
1      300     1
OXX            UX
OXY            UY
****
F1            USET(1)
****
XA             0.0
HX             0.5
NX             2
YA             0.0
HY             0.5
NY             2
****
****
D3EST          1./(X**2+Y**2)
****
TF             12.
****
NOUT           3
****
ARC=-1001
FB1            1.0
ARC=-1003
FB1            1.0
ARC=-1004
FB1            1.0
ARC=-1005
FB1            1.0
ARC=-1006
FB1            1.0
****
SYMMETRY       1
****
NUPDT          0
****
U0             1
****
NORMAL
****
VXY            0.0,0.0,0.5,0.0,1.0,0.0,1.0,0.5,0.5,0.5,0.5,1.0,
VXY            0.0,1.0,0.0,0.5,0.25,0.25,0.75,0.25,0.25,0.75
****
IABC           1,2,9, 2,5,9, 5,8,9, 8,1,9,
IABC           2,3,10, 3,4,10, 4,5,10, 5,2,10,
IABC           8,5,11, 5,6,11, 6,7,11, 7,8,11
****
I              -1001, 0, 0, 1002,
I              -1001, -1003, -1004, 0,
I              0, -1005, -1006, 1002
END.
```

```
****
****    CHAPTER 8.
****    SECTION(8-4)
****    TEST PROBLEM 3
****    FIGURE(8-18)
****
****
****        THE EIGEN-VALUE PROBLEM
****
1      300     1
****
OXX             UX
OXY             UY
****
F1             USET(1)
****
U0              1
****
D3EST           1./(X**2+Y**2)
****
NORMAL          1
****
XA              0.0
HX              0.1
NX              10
YA              0.0
HY              0.1
NY              10
****
TF              9.
****
NOUT            3
****
ARC= -1003
FB1             1.0
ARC= -1004
FB1             1.0
ARC= -1005
FB1             1.0
ARC= -1006
FB1             1.0
****
SYMMETRY        1
****
NUPDT           0
****
VXY             0.0,0.0,0.5,0.0,1.0,0.0,1.0,0.5,0.5,0.5,0.5,1.0,
VXY             0.0,1.0,0.0,0.5,0.25,0.25,0.75,0.25,0.25,0.75
****
IABC             1,2,9, 2,5,9, 5,8,9, 8,1,9,
IABC             2,3,10, 3,4,10, 4,5,10, 5,2,10,
IABC             8,5,11, 5,6,11, 6,7,11, 7,8,11
****
I               1004, 0, 0, 1002,
I               1004, -1003, -1004, 0,
I               0, -1005, -1006, 1002
END.
```

```
****     CHAPTER 8.
****    SECTION(8-5)
****
****   FLUID MECHANICS PROBLEM
****
****     NAVIER-STOKES PROBLEM
****
2      150       2
OXX           -1.D5*REDINT*(UX+VY)+UX
OXY           UY
OYX           VX
OYY           -1.D5*REDINT*(UX+VY)+VY
F1            PARA(T)*(-U*UX-V*UY)-(2.*DSIN(Y)+DSIN(X))*DCOS(X)
F2            PARA(T)*(-U*VX-V*VY)-(DSIN(Y)-2.*DSIN(X))*DCOS(Y)
****
TF            10.
****
NOUT          5
****
ALPHA         2
****
U0            -DCOS(X)*DSIN(Y)
****
V0             DSIN(X)*DCOS(Y)
****
XA            0.0
HX            2.199114858
NX            2
****
YA            0.0
NY            10
HY            0.1
****
****
ARC=-1001
FB1              -DCOS(X)*DSIN(Y)
FB2               DSIN(X)*DCOS(Y)
ARC=-1002
FB1              -DCOS(X)*DSIN(Y)
FB2               DSIN(X)*DCOS(Y)
ARC=-1003
FB1              -DCOS(X)*DSIN(Y)
FB2               DSIN(X)*DCOS(Y)
ARC=-1004
FB1              -DCOS(X)*DSIN(Y)
FB2               DSIN(X)*DCOS(Y)
****
****
VXY           0.0,0.0, 3.14159,0., 3.14159,3.14159,
VXY           0.0,3.14159, 1.570796,1.570796
IABC          1,2,5, 2,3,5, 3,4,5, 4,1,5
****
I              -1001, -1002, -1003, -1004
****
```

ADD.

```
      DOUBLE PRECISION FUNCTION PARA(T)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARA=DMIN1(1.0,(T-1.0)/3.0)
      RETURN
      END
```

END.