EXPLICIT STIFFNESS MATRICES

FOR TRIANGULAR, PLATE BENDING

FINITE ELEMENTS

A thesis submitted for examination for
the degree Ph.D. in Engineering by:

R.L. Harrison

Department of Civil Engineering,
University of Cape Town,
October, 1973.

LIST OF CONTENTS

List of Appendices

List of Figures

List of Tables

LIST OF SYMBOLS

To avoid the use of unusual characters, some symbols have different meanings in different sections of the thesis. Where this has occurred, a numeral at the beginning of the definition in the list below, indicates the section to which it applies. The sections are numbered as follows:-

(1) The assumed displacement method
(2) The assumed stress method
(3) The assumed stiffness method
(4) The appendices.

Special Meanings

{ }	a vector
[ ]	a matrix
(i × j)	the dimensions of a matrix
(3)	equation reference number
d	differentiation with respect to
∂	partial differentiation with respect to
− 1 superscript	the inverse of a matrix
t superscript	the transpose of a matrix
$\hat{A}$	the angle A
vol	the volume of material
i, j subscript	numeric subscripts
x subscript	orthogonal axis subscripts
u subscript	oblique axis subscripts

Lower Case Characters

a	(1) a factor defined where used
a ?	(2) an axis tangential to the edge of a finite element
b	(2) an axis normal to the edge of a finite element
c	(2) a factor defined where used
a, b, c	(3) the lengths of the three sides of a triangle or triangular finite element
a, b, c, d	(4) variables defined where used
a1	(4) a defined value in the matrix [A]

Lower Case Characters (Continued)

e	the length of a median of a triangle
$f(x, y)$	any general function in x and y
$fn(m)$	variation number m of function number n
f, g	factors defined where used
h	the material thickness of an element
k	the number of degrees of freedom considered at a general point
l_{ij}	the length of side ij of an element
m	(1) and (2) the number of nodal degrees of freedom of an element
m	(3) the cyclic or mirror variation of a function of type $fn(m)$
n	the number of unknown coefficients in a function
p	(1) and (2) the number of stresses and strains considered at a point
p	(3) the degree of function P
q	a uniform distributed load
r	the base length of a triangle
s	the offset of the apex of a triangle along its base from one end
t	the height of a triangle
u_i, v_i	oblique axes defined at point i
w_i	vertical translation of point i
$w_{ij}(a)$	(2) the displacement function for edge ij of an element
x, y, z	orthogonal coordinate axis names or the coordinates of a general point
x_i, y_i	orthogonal axes at point i or the coordinates of a particular point
x_{ri}, y_{ri}	the coordinates of point i relative to a defined point 0

Upper Case Characters

A(m) to M(m)	particular functions of type fn(m)
A(i,j)	element (i, j) of matrix $[A]^{-1}$
$\hat{A}$, $\hat{B}$, $\hat{C}$	angles of a triangle
A_i to M_i	coefficient i of functions A(m) to M(m)
C_i	any general coefficient number i (defined)
D	the rigidity modulus of a plate

Upper Case Characters (continued)

E	Young's modulus of elasticity
$E(i, j)$	element (i, j) of matrix $[E]^{-1}$
F_i	the applied force at node i
$G(i, j)$	element (i, j) of matrix $[G]$
$K(i, j)$	element (i, j) of matrix $[K]$
L	the length of the side of a square
M_i	the resultant applied couple at node i
M_{xi}	the applied couple in direction x at node i
M_x	the bending moment in the x direction
M_{xy}	the twisting moment about the y axis
$M_{aij}(a)$	the moment distribution function along side ij of an element
P	(3) a cyclic homogeneous function in a, b, c, α, β, γ and Δ
Q_x	the unit shear causing rotation about the x axis
U	the total internal energy of an elastic body
W	the total external work done on an elastic body
$Z(i, j)$	element (i, j) of the matrix $[Z]$

Greek Characters

α, β, γ	(1) the three angles of a triangle
α_{ij}	(2) the inclination of side ij to the x axis
α, β, γ, Δ	(3) parameters defined in terms of a, b and c
δ	(1) an operator indicating a defined series of differentiations of a matrix
δx, δy	differential lengths of an element of material
γ_{xy}	the shear deformation of plane xy in the direction y
ϵ_x	the direct strain in direction x
θ_{xi}	the rotation of point i about the x axis
$\theta_{aij}(a)$	a function representing the rotation of edge ij about the axis 'a'
ν	Poisson's ratio
τ_{xy}	the shear stress causing γ_{xy}
ϕ_u	the inclination of the u axis with respect to the x axis

Matrices and Vectors

[A]	(1) the parameters of a chosen function in terms of the nodal coordinates
[A]	(2) a component and factor of matrix [H]
[A]	(4) an augmented matrix
$[A_{ij}]$, $[B_{ij}]$, $\{C_i\}$, $\{D_i\}$, $\{E_i\}$	(4) submatrices and subvectors defined in an example
[B(x, y)], [B]	(1) parameters of a strain function derived from the general displacements
[B]	(2) a component of matrix [H]
{c}	a vector of unknown coefficients
[C]	(4) a number of unknown vectors of coefficients side by side
$[C_i]$	(3) three sets of coefficients of the explicit stiffness matrix to be multiplied by 1, ν and ν^2 respectively
{d}	nodal displacements
{d(x, y)}	displacements at a point x, y
$\{d_i\}$	a particular example of {d}
[D]	(1) and (2) a matrix of material properties relating stress to strain
[D]	(3) a set of vectors of type {d} side by side
[E]	a factor of matrix [H]
{f}	nodal loads
$\{f_i\}$	a particular example of {f}
[F]	a set of vectors of type {f} side by side
[G]	a defined factor of matrix [K]
[H]	a defined factor of matrix [K]
[I]	the unit or identity matrix
[K]	the stiffness matrix for a structure or structural element
$[K_i]$	a defined matrix containing parts of a stiffness matrix or a specific example of a stiffness matrix
[L(a)], [L]	relates the displacements of the edges of an element to the nodal displacements
[M]	relates the stresses at a general point to the nodal displacements
[P(x, y)], [P]	parameters of the assumed functions

Matrices and Vectors (continued)

$\{P_i\}$	the i th column of [P]
$\{P\}$	parameters of an explicit stiffness matrix
$[R(a)]$, $[R]$	relates the edge loading to the coefficients of the assumed stress functions
$\{S(a)\}$	the edge loading function for each edge
$[T]$	a transformation matrix (defined where subscripted)
$\{U\}$	the coordinates of a general point in terms of oblique coordinates u, v, z
$\{u(a)\}$	the assumed edge displacement functions for each edge of an element
$\{W_i\}$	the total internal energy for a number of virtual displacement vectors
$\{W_e\}$	the external work done to cause $\{W_i\}$
$\{X\}$	the coordinates of a general point in terms of orthogonal coordinates x, y, z
$[X]$	known parameters or values multiplying the unknowns of a set of equations
$[Y]$	a set of known constant vectors each representing a set of equations
$[Z]$	a defined factor of the stiffness matrix
$\{\epsilon(x, y)\}$	the strains at a general point x, y
$[\epsilon_v(x, y)]$	the strains at a general point x, y caused by a set of virtual nodal displacements
$\{\sigma(x, y)\}$	the stresses at a general point x, y

Acknowledgements.

I am extremely grateful to Mr. W.S. Doyle for the genuine interest that he showed in my work at all times. The many long hours of assistance during his private time are sincerely appreciated.

Without the backing of AE and CI this project would never have been commenced. To them I owe gratitude for three years of freedom from financial burden.

To Miss B. Spalding go my sincere thanks for, and congratulations on, a masterly typed presentation of a difficult script.

Thanks are also due to the staff of the Civil Engineering Department, the Computer Centre and the Library of the University of Cape Town, in particular to Terry Griffin and Colin Goldberg for assistance with specialised problems.

Finally, I would like to thank Charlie Basson for the care which he exercised in duplicating copies of the thesis.

Bob Harrison.

CHAPTER 1

1. INTRODUCTION

1.1 Statement of the Problem

Explicit stiffness matrices are available for rectangular plate bending elements, rectangular plane stress and plane strain elements and triangular plane stress and plane strain elements. Triangular plate bending elements can at present only be formed by using a numeric algorithm.

The explicit version of a stiffness matrix is not only far more simple to program in a computer routine but its execution (as will be shown) requires approximately one twelth of the time of a numeric version.

Rectangular plate bending elements do not have a compliant shape (see definition) so their use is limited to plates which in many cases can be solved by other methods.

Judging from the number of attempts to find a successful triangular plate bending element, the simplicity of a triangular shape appeals to most investigators.

In the present investigation only triangular elements with a node on each corner and three degrees of freedom at each node will be considered.

Some investigators have included extra nodes on the edges and/or at the centroid of the triangle. This is done in order to overcome difficulties experienced in choosing a suitable displacement function for a nine degree of freedom triangle. Except for comparison of results (table 4) such elements will not be considered.

An explicit stiffness matrix for a small deflection theory, elastic, isotropic, triangular plate bending element will be developed.

To limit the field of research, only elements derived from functions which may be described as polynomials will be considered.

1.2 Synopsis

Introduction: Three different methods were used to find explicit matrices for triangular plate bending elements. The assumed displacement method and the assumed stress method were considered first. Because of certain shortcomings in these matrices, a third explicit matrix was developed by assuming the form of the resulting matrix and finding the values of unknown coefficients involved. This third method, to be known as the "assumed stiffness method" forms the major investigation of this thesis.

The assumed displacement method: A simple nine degree of freedom finite element of this type (Przemieniecki[1]) was investigated. The explicit matrix found required 0,512 seconds - compared with 6,103 seconds of the fully numerical version - to evaluate a single finite element stiffness matrix. This showed the worth of the explicit matrix but for certain configurations of the nodes (Zienkiewicz[5]) this method gives a singular matrix. Examination of the explicit matrix enables one to determine conveniently which configurations these are. It is also shown from the explicit formula that if a particular configuration has a singular matrix, it will be possible in every case to choose a new set of local coordinates for the element which results in a non-singular matrix.

The assumed stress method: A matrix of this type (described in papers by Pian[2], Severn and Taylor[3] and Allwood and Cornes[4]) was then treated in a similar manner. This element is known to give more reliable results. The explicit matrix found in this case was impractically large.

Algebraic manipulations in both of these developments were done on a digital computer using programs written in FORTRAN (which essentially handles numerical operations).

It was apparent from the above investigations that algorithms designed for a numerical approach were not well suited to an explicit treatment. However, the use of the explicit matrix is justified by the saving in computer time.

<u>The assumed stiffness method</u>: Noting that some function must be assumed in every stiffness matrix development, the obvious way to obtain an explicit solution would be to choose functions representing each element of the matrix. The form of these functions could be determined by using the earlier developments as guide lines.

The assumed functions are formed from the dimensions of the finite element. They are homogeneous polynomials of a degree determined by the units of the matrix. By using this element in enough examples in which both the displacement and the loading are known, the unknown coefficients of the assumed functions could be found.

Very many coefficients are required to form a matrix of even the most simple functions. The solution of such large sets of simultaneous equations is a major undertaking, particularly where equations may be badly conditioned. Ways of reducing the number of independent unknowns to a minimum must be found.

Little more than half of the full number is required due to the symmetry of stiffness matrices. A further two thirds are eliminated by describing the dimensions and local coordinates in a systematic manner.

Large reductions could also be made by considering the static equilibrium of the finite element. To do this an energy conservation principle was applied. This required that no external work be done on an element if it moved in such a way that its shape was not deformed.

An explicit stiffness matrix with as few as twenty five unknown coefficients was obtained in this manner. The values of these coefficients can be found by doing a finite element analysis on a plate with a known solution and twenty five degrees of nodal freedom.

By varying the positions of the nodes and the shapes of the elements, many different solutions could be found for the coefficients. Three resulting explicit stiffness matrices were tested. All three were found to converge satisfactorily. The average time taken to generate the stiffness matrix for a single element was about 0,34 seconds. Results compared favourably with other finite element analyses. On a few of the examples tested, a greater accuracy is desirable.

There is a wide scope for improving the accuracy of the element without complicating the matrix or increasing the execution time. Suggestions are also made for the extension of the theory to more complex functions and element shapes.

General: A method of obtaining stresses from the nodal displacements by "best surface fitting" is suggested. Results found by this method are not reliable at this stage but other procedures for obtaining stresses can be used.

An "industrial package program" including a comprehensive "data debugging" routine was prepared to perform finite element analyses using the "assumed stiffness" element.

1.3 Original Contributions Included in the Thesis

Appart from the two main concepts in this thesis, namely: the development of an explicit stiffness matrix:

(a) by computerised algebraic substitution into a numerical algorithm and
(b) by the assumed stiffness method, the following are original:-

(i) two methods of representing and manipulating algebraic expressions numerically;
(ii) a method of inverting an algebraic matrix which can be resolved into the sum of a non-singular matrix which can easily be inverted and a component which is a very sparse matix;
(iii) the proof that the triangular stiffness matrix derived by the assumed displacement method has at least one non-singular orientation for any configuration of nodes;

(iv) the choice of orthogonal axis systems radiating from the centre of the inscribed circle which results in systematic submatrices of the stiffness matrix;

(v) the choice of oblique axes in the directions of the sides to simplify application of rigid body movements;

(vi) the choice of polynomials involving integer powers of surds as assumed functions;

(vii) the use of the non-similarity ratio for distinguishing between triangular shapes.

In addition the FORTRAN programs,

(i) using prime numbers algebra;

(ii) using powers algebra;

(iii) using an explicit assumed displacement method stiffness matrix;

(iv) for forming and solving a set of simultaneous identities;

(v) for reducing a random sparse set of equations to their simplest form;

(vi) for checking the validity of a set of data describing the topology of a triangular mesh;

(vii) for finding the values of stiffness matrix coefficients by application to a standard plate;

(viii) for performing a finite element structural analysis;

were all written as part of the investigation without reference to other computer programs or descriptions thereof.

1.4 Definitions

1. Explicit Matrix: Each element of an explicit matrix is an algebraic formula which can be evaluated by numerical substitution for the variables in it.

2. Algorithmic formation of an element: This is the alternative method of finding the numerical value of a matrix. All primary variables are first evaluated. The full matrix is then developed by following a sequence of numerical manipulations of the primary variables. If these

operations are followed using algebraic variables instead of their numerical values, the resulting matrix is an explicit matrix.

3. An element with a compliant shape: In the context of this thesis, this definition applies only to planar elements with rectilinear boundaries. If any shape of plate can be divided into a number of elements of a certain description, that type of element has a compliant shape, (e.g. a triangular plate can not be divided into rectangular elements; rectangular elements do not therefore have a compliant shape). Triangles and quadrilaterals have this property because even circular shapes can be closely approximated by an assembly of either.

4. The "Non-Similarity Ratio" for a triangle: This is a ratio to show whether two triangles are dissimilar. It is the ratio of the product of the lengths of the three sides of a triangle to the cube of the length of the longest side. If two triangles have different non-similarity ratios they are dissimilar. If two triangles have the same non-similarity ratio they are not necessarily similar.

5. The root length degree of a function: Certain surds are used as variables in chosen functions. The dimensions of these surds are the square root of length. A polynomial of degree k in these variables is said to have root length degree k.

6. A cyclic variation: If a, b and c are in cyclic order then the arrangements:

$$a\ b\ c, \quad b\ c\ a, \quad c\ a\ b$$

are cyclic variations of one another.

7. A minor variation: If a, b and c are in cyclic order then the arrangements:

$$a\ b\ c \quad \text{and} \quad a\ c\ b$$

are minor variations of one another.

8. Homogeneous: This usually means that all terms of an expression have the same degree. In this thesis the term is taken to mean that all terms have the same units (of length say).

1.5 General Remarks

It will be assumed that any reader of this thesis is familiar with the finite element method. It will also be assumed that the reader is conversant with the computer language FORTRAN V.

In order to maintain continuity of the main logic, certain self-contained sections are included as appendices. Often these appendices contain original and relevant descriptions, too long to be included in a concise main argument.

CHAPTER II

EXPLICIT MATRICES DERIVED ALGEBRAICALLY FROM PUBLISHED ALGORITHMS

2.1 Introduction

This chapter covers the minor research of this thesis which leads to the more important method described in subsequent chapters.

In published papers dealing with stiffness matrices for plate bending elements with a compliant shape (see definitions), a final resulting matrix is not published. The reason for this is that the labour involved in producing such a matrix, if done by hand, would be extensive. It is thus more convenient for authors to describe the final steps in general terms. terms. It is also fairly simple for a research worker to write a program which performs the final steps numerically.

For the user of such a method in practice, the unfinished matrix is uneconomical both from the programming point of view and because of the time wasted in the execution of a program.

It is far more simple to write a program to substitute values into a formula (even if this is very lengthy), than it is to write a program which performs operations such as matrix multiplications and numerical integrations in two dimensions. At the same time, such matrix manipulations often involve multiplications and additions of zero values which do not appear in a final formula. This is the reason why (as will be shown in this chapter) programs using an explicit stiffness matrix are not only more simple than a numerical program but require less computer time for their execution.

Two standard methods of developing a stiffness matrix for a plate bending finite element will be described in general terms. These are the assumed displacement method and the assumed stress method.

Explicit stiffness matrices for a triangular element are found using both methods. This element has a node on each corner with three degrees of freedom at each node. The derivations of the matrices are taken from published papers.[1, 2, 3, 4] The repetitive algebraic operations required in the final steps are performed on a digital computer using techniques described in appendix 4.

The explicit stiffness matrix developed by the assumed displacement method was found to require one twelfth of the time used by a numerical program on the computer. Unfortunately, the matrix itself has a few serious inherent problems as will be pointed out. The explicit stiffness matrix developed by the assumed stress method was found to be too extensive to be handled in a finite element program. No further deductions were made from the latter.

2.2 The Use of FORTRAN Programs for Simple Algebraic Operations

Supposing a stiffness matrix is required for a general triangular shape of element. The dimensions of this shape will be algebraic variables. Various matrices describing properties of the element will have to be developed as steps to the final matrix. The elements of these matrices are expressions in terms of the variable dimensions. By removing the lowest common denominator from the expressions, the matrix elements will be left as fairly long polynomials.

Supposing two such matrices are to be multiplied together. The matrices are often of the order of nine by nine. Hundreds of multiplications and additions may be required to find the expression representing a single element of the resulting matrix.

The probability of human error in a large number of simple operations such as these is great.

Two methods of performing these algebraic operations using FORTRAN programs (which are essentially numerical) were found. The methods of "algebra by prime numbers" and "algebra by powers" are described in Appendix 4.

The former method was developed and used in the assumed displacement method described in this chapter. When an attempt was made to use this in the assumed stress method, it was found to be unsatisfactory (see below). This is the reason why the "algebra by powers" was developed.

This was then used in the assumed stress method.

It was found in the prime numbers algebra that when six or seven variables were required in an expression, the size of the numbers representing the terms became impractically large, e.g. supposing a term has variables a, b, c, d, e, f, g, taking the lowest seven prime numbers respectively, the value representing

$$g^9 = 17^9 = 118\,588\,876\,497.$$

This value overflows the maximum size allowed on most computers for integers.

On the other hand, the algebra by powers can easily represent a number such as:

$$a^{99}\ b^{99}\ c^{99}\ d^9\ e^9\ f^9\ g^9.$$

The algebra by powers does not require exponentiation of input or factorisation of output and has all the advantages of the algebra by prime numbers technique.

2.3 The Assumed Displacement Method

2.3.1 Method in general terms

Supposing the nodal displacements of an element were known, but the internal displacements were not. An interpolation formula with as many unknown coefficients as the element had nodal degrees of freedom could be assumed. The values of the coefficients could be solved by substitution of the coordinates and displacements of the nodes.

Once the interpolation formula had been evaluated, the displacements at any point could be found by substitution of its coordinates. By appropriate differentiation of the formula, the strains could be found. If the material properties of the element were known the stresses could be determined from the strains.

Integration of a product of the stress and strain over the volume of the element gives the total internal strain energy.

A product of the nodal loads and the displacements caused by them gives the total external work done to cause this strain.

These two quantities should be equal. In fact, the nodal displacements are unknown. From the equation formed from the energy considerations, the nodal displacements can be solved as unknowns given the loading. Their values are dependent upon the form of the interpolation formula.

This is the principle used to develop the stiffness matrix by the assumed displacement method.

Supposing a general point (x,y) of the element has k degrees of freedom. Let the displacements of this point be represented by a column vector $\{d(x,y)\}$. If the element has m degrees of nodal freedom, the displacements of a general point may be represented in terms of functions involving its coordinates and m unknown coefficients contained in a vector $\{c\}$ as follows:-

$$\underset{(k \times 1)}{\{d(x,y)\}} = \underset{(k \times m)}{[P(x,y)]} . \underset{(m \times 1)}{\{c\}} \qquad (1)$$

Substituting the coordinates of each node of the element and a symbol representing the corresponding unknown nodal displacement, into equation (1), the vector of unknown nodal displacements $\{d\}$ can be represented as:

$$\{d\} = [A]\{c\} \qquad (2)$$

where, if n is the number of nodes on the element:

$$\{d\} = \underset{(m \times 1)}{\begin{bmatrix} \{d(x_1, y_1)\} \\ \{d(x_2, y_2)\} \\ \vdots \\ \{d(x_n, y_n)\} \end{bmatrix}} \quad \text{and } [A] = \underset{(m \times m)}{\begin{bmatrix} [f(x_1, y_1)] \\ [f(x_2, y_2)] \\ \vdots \\ [f(x_n, y_n)] \end{bmatrix}}$$

As the values of the unknown constants are in themselves of little value, it is useful to eliminate $\{c\}$ from the above equations. From equation (2):

$$\{c\} = [A]^{-1} \{d\} \tag{3}$$

Substitution of this into equation (1) gives the displacements at a general point in terms of the nodal displacements:

$$\{d(x,y)\} = [P(x,y)].[A]^{-1}.\{d\}.$$

An operator δ, of appropriate differentiation of these displacements with respect to the coordinates, can be devised to give a vector of strains $\{\varepsilon(x,y)\}$ at a general point:

$$\{\varepsilon(x,y)\} = \delta.\{d(x,y)\} = \delta.[P(x,y)].[A]^{-1}\{d\}$$

and by defining $[B(x,y)] = \delta.[P(x,y)]$:

$$\underset{(p \times 1)}{\{\varepsilon(x,y)\}} = \underset{(p \times m)}{[B(x,y)]} . \underset{(m \times m)}{[A]^{-1}} . \underset{(m \times 1)}{\{d\}} \tag{4}$$

where: p is the number of strains defined at any one point.

The matrix $[D]$ of material properties, relating the stresses $\{\sigma(x,y)\}$ which exist at a general point to the strains at that point, is defined by:

$$\underset{(p \times 1)}{\{\sigma(x,y)\}} = \underset{(p \times p)}{[D]} . \underset{(p \times 1)}{\{\varepsilon(x,y)\}} \tag{5}$$

The vector $\{\sigma(x,y)\}$ corresponds element by element to the strain vector $\{\varepsilon(x,y)\}$. As a result, the matrix $[D]$ is symmetric.

Substitution from equation (4) into equation (5) gives:

$$\{\sigma(x,y)\} = [D].[B(x,y)].[A]^{-1}\{d\}$$

and by defining $[M] = [D].[B(x,y)].[A]^{-1}$:

$$\{\sigma(x,y)\} = [M]\,\{d\} \tag{6}$$

The matrix $[M]$ is used to obtain the stresses (or moments) at a point once the nodal displacements $\{d\}$ are known.

Consider an element which has nodal deflections $\{d\}$ and internal stresses $\{\sigma(x,y)\}$ due to a vector of nodal loads $\{f\}$. Suppose that an infinitely small unit virtual displacement is applied separately, corresponding to each nodal degree of freedom of the element. The matrix representing all these displacement vectors (side by side) would be an identity matrix $[I]$. Call the matrix of strains caused by this matrix of displacements $[\epsilon_v(x,y)]$.

The increment in internal energy due to the stresses caused by the loads $\{f\}$, when the virtual strains take place, may be represented by:

$$\underset{(m\times 1)}{\{W_i\}} = \int_{VOL} \Big(\underset{(m\times m)}{[\epsilon_v(x,y)]^t} . \underset{(m\times 1)}{\{\sigma(x,y)\}} \Big)\, d\ vol \tag{7}$$

The increment in the work done by the loads $\{f\}$ when the virtual displacements take place may be represented by:

$$\underset{(m\times 1)}{\{W_e\}} = \underset{(m\times m)}{[I]} . \underset{(m\times 1)}{\{f\}} = \{f\} \tag{8}$$

The external work done must be equal to the internal increase in strain energy. Thus from equations (7), (8) and substitutions from (4) and (6):

$$\begin{aligned} \{f\} &= \int_{VOL} \Big([\epsilon_v(x,y)]^t . \{\sigma(x,y)\} \Big)\, d\ vol \\ &= \int_{VOL} \Big([B(x,y)] . [A]^{-1} . [I] \Big)^t . \Big([D] . [B(x,y)] . [A]^{-1} . \{d\} \Big)\, d\ vol \\ &= [A]^{-1t} . \int_{VOL} [B(x,y)]^t . [D] . [B(x,y)]\, d\ vol\ [A]^{-1} . \{d\} \end{aligned}$$

as $[A]^{-1}$ is independent of the variables x and y.

Comparing this to the basic equation of the stiffness method:

$$[K]\{d\} = \{f\} \tag{9}$$

where: [K] is the matrix representing the stiffness of any structure or structural element.

The matrix [K] for this particular finite element can be represented by:

$$[K] = [A]^{-1t} . \int [B(x,y)]^t . [D] . [B(x,y)] \, d \, vol \, [A]^{-1} \tag{10}$$

Numerous elements have been developed based on this method. Variations are possible in the shape of the element, the use of the element (e.g. for bending, membrane action, etc.) and the choice of the displacement function. A particular triangular plate bending element (Pryzemieniecki[1]) will be described in detail in subsequent subsections.

2.3.2 <u>The difficulty with nine degrees of freedom</u>

A plate bending element has three degrees of freedom (two rotations and a translation) at any point. A triangular element with a node at each corner has nine degrees of nodal freedom.

The reasoning behind the choice of a particular polynomial as a displacement function is:

A plate element is essentially two dimensional, (the thickness is assumed to remain unchanged during deformation). A polynomial in only the two coordinate variables is therefore chosen.

It is possible that the origin of axes may lie on the element and that this point may have non-zero deflection, slopes, curvatures, bending moments and shears associated with it. These properties are represented by various low order differentials of the assumed displacement function. Supposing this is:

$$w = f(x,y)$$

then it is essential that:

$$f(0, 0) \neq 0$$
$$(\partial f/\partial y)(0, 0) \neq 0$$
$$(\partial f/\partial x)(0, 0) \neq 0$$
$$(\partial^2 f/\partial x^2)(0, 0) \neq 0 \quad \text{etc.}$$

It is therefore necessary to choose all the lowest degree terms of a polynomial when choosing the assumed function.

A full third degree polynomial in two variables has ten coefficients:

$$f(x,y) = C_1 + C_2x + C_3y + C_4x^2 + C_5xy + C_6y^2 + C_7x^3 + C_8x^2y + C_9xy^2 + C_{10}y^3$$

As only nine degrees of freedom are to be considered, only nine coefficients can be used. It is possible to omit any of C_7 to C_{10} but in this case, in order to retain symmetry, the variables of C_8 and C_9 were linked to give a single coefficient.

$$f(x,y) = C_1 + C_2x + C_3y + C_4x^2 + C_5xy + C_6y^2 + C_7x^3 + C_8(x^2y + xy^2) + C_{10}y^3 \qquad (11)$$

The use of this function leads to problems (Zienkiewicz[5], p 185) in the inversion of the matrix [A] (equation (3)). The reason for this becomes plain if one considers a symmetrical element in the x,y plane which is not symmetrical with respect to the two axes. If a symmetrical nodal displacement vector exists for the element, it is possible for a non-symmetric internal displacement field to result from the above function. For example, consider figure 1.

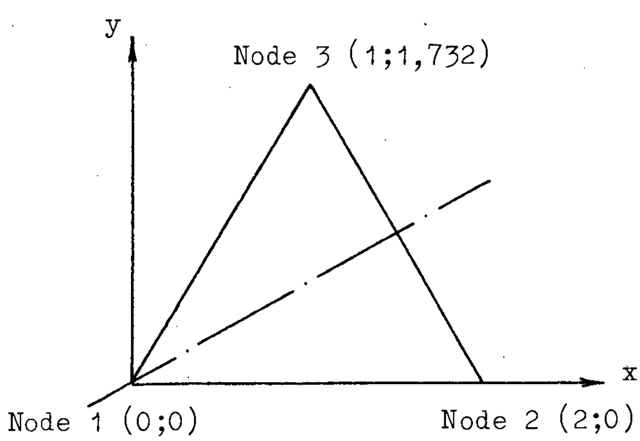


Figure 1: A symmetrical Element which does not lie Symmetrically with Respect to the Axes.

Prescribe the following symmetric nodal displacement field. (This represents an outward unit twist of nodes 2 and 3 about the line joining each to node 1).

$$\{d\} = \begin{bmatrix} w_1 \\ \theta_{x1} \\ \theta_{y1} \\ w_2 \\ \theta_{x2} \\ \theta_{y2} \\ w_3 \\ \theta_{x3} \\ \theta_{y3} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ -\frac{1}{2} \\ -.866 \end{bmatrix}$$

It is found, upon solution for the constants of equation (11) that the side 1, 2 remains at zero displacement for its entire length while the side 1, 3 arches up from the zero deflections of the nodes at its ends.

This sort of problem arises no matter which terms are omitted from the full polynomial.

Another inconsistency of a triangular element based on this function is that when it is rotated about an axis in its plane, without deforming in any way, there is an increase in its internal strain energy.

One may wonder at the value of pursuing this derivation any further. Some investigators, (Clough and Tocher[6] and Gellert and Gluck[7]) have had acceptable results using this stiffness matrix. The reason for continuing is that a comparison can be made between the execution times of an explicit formation and a numeric formation of a stiffness matrix.

It will also be possible to show, from the final form of the matrix, that at least one orientation, of the local coordinate axes for any shape of element, gives a non-singular stiffness matrix.

2.3.3. Derivation of a stiffness matrix for a triangular element

The following element stiffness matrix derivation was developed by Tocher and described by Przemieniecki.[1] Inversion of the explicit matrix [A], algebraic integration of the matrix [Z] and subsequent derivations were performed from first principles.

A triangular element is chosen with a local x-coordinate axis along one side and its middle surface in the positive quadrant of the x,y plane of a right hand axis system as shown in figure 2. The three nodes, numbered clockwise about the z axis, have the coordinates (0,0); (r,0) and (s,t) respectively. At each node, three degrees of freedom are prescribed:

(i) 'w' a translation perpendicular to the plane of the element in the positive z direction;
(ii) 'θ_x' a rotation about the x-axis in a clockwise direction;
(iii) 'θ_y' a rotation about the y-axis in a clockwise direction.

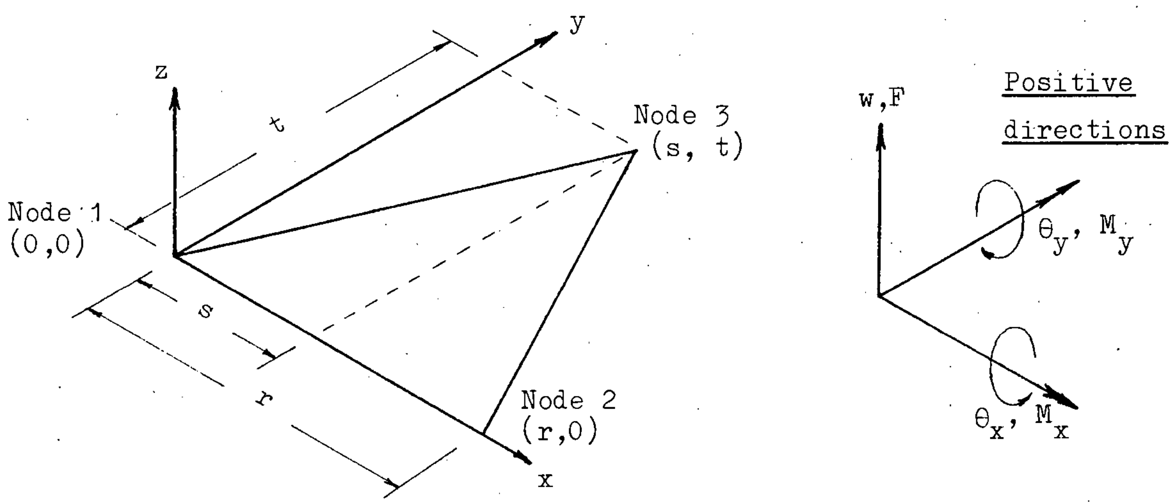


Figure 2: Dimensions and sign conventions for a general triangular finite element.

As the small deflection theory is to be used it will be noted that:

$$\left.\begin{aligned} \theta_x &= dw/dy \\ \theta_y &= -\,dw/dx \end{aligned}\right\} \qquad (12)$$

The appropriate strain vector for plate bending is found to be:

$$\{\epsilon(x,y)\} = \begin{bmatrix}\epsilon_x\\ \epsilon_y\\ \gamma_{xy}\end{bmatrix} = -z\begin{bmatrix}\partial^2 w/\partial x^2\\ \partial^2 w/\partial y^2\\ 2\partial^2 w/\partial x\partial y\end{bmatrix} \tag{13}$$

and the corresponding stress vector is:

$$\{\sigma(x,y)\} = \begin{bmatrix}\sigma_x\\ \sigma_y\\ \tau_{xy}\end{bmatrix} = [D]\,.\,\{\epsilon(x,y)\} \tag{14}$$

where:

$$[D] = \frac{E}{(1-\nu^2)}\begin{bmatrix}1 & \nu & 0\\ \nu & 1 & 0\\ 0 & 0 & (1-\nu)/2\end{bmatrix}$$

and: E is Young's modulus of elasticity

ν is Poisson's ratio for the material

From the assumed displacement function, (11) and the relationships of equations (12)

$$\begin{aligned}
w &= C_1 + C_2x + C_3y + C_4x^2 + C_5xy + C_6y^2 + C_7x^3 + C_8(x^2y + xy^2) + C_9y^3\\
\theta_x &= \partial w/\partial y = C_3 + C_5x + 2C_6y + C_8(x^2 + 2xy) + 3C_9y^2\\
\theta_y &= -\partial w/\partial x = -C_2 - 2C_4x - C_5y - 3C_7x^2 - C_8(2xy + y^2)
\end{aligned} \tag{15}$$

Substituting the nodal coordinates (as described in (2)):

$$\{d\} = \begin{bmatrix}w_1\\ \theta_{x1}\\ \theta_{y1}\\ w_2\\ \theta_{x2}\\ \theta_{y2}\\ w_3\\ \theta_{x3}\\ \theta_{y3}\end{bmatrix} = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
1 & r & 0 & r^2 & 0 & 0 & r^3 & 0 & 0\\
0 & 0 & 1 & 0 & r & 0 & 0 & r^2 & 0\\
0 & -1 & 0 & -2r & 0 & 0 & -3r & 0 & 0\\
1 & s & t & s^2 & st & t^2 & s^3 & s^2t+st^2 & t^3\\
0 & 0 & 1 & 0 & s & 2t & 0 & s^2+2st & 3t^2\\
0 & -1 & 0 & -2s & -t & 0 & -3s^2 & -2st-t^2 & 0
\end{bmatrix} \times \begin{bmatrix}C_1\\ C_2\\ C_3\\ C_4\\ C_5\\ C_6\\ C_7\\ C_8\\ C_9\end{bmatrix} \tag{16}$$

$$= [A]\,.\,\{c\}$$

From equations (4), (13) and (15):

$$\{\varepsilon(x,y)\} = -z \begin{bmatrix} 0 & 0 & 0 & 2 & 0 & 0 & 6x & 2y & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 2x & 6y \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 4(x+y) & 0 \end{bmatrix} \times \{c\}$$

$$= [B].\ [A]^{-1}.\ \{d\}$$

Hence from equation (10):

$$[K] = [A]^{-1t}.[Z].[A]^{-1} \qquad (17)$$

where:

$$[Z] = \int_{VOL} [B]^t,\ [D].\ [B]\ d\ vol$$

The multiplications required for the matrix [Z] are then performed. The limits of integration of the variable z are $-h/2$ and $+h/2$ where h is the element thickness. The matrix [Z] can then be represented as

$$[Z] = D \times \iint \left[\begin{array}{c|c|c|c|c|c|c|c|c} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 & 4\nu & 12x & 4(\nu x+y) & 12\nu y \\ 0 & 0 & 0 & 0 & 2(1-\nu) & 0 & 0 & 4(1-\nu)(x+y) & 0 \\ 0 & 0 & 0 & 4\nu & 0 & 4 & 12\nu x & 4(x+\nu y) & 12y \\ 0 & 0 & 0 & 12x & 0 & 12\nu x & 36x^2 & 12(\nu x+y)x & 36\nu xy \\ 0 & 0 & 0 & 4(\nu x+y) & 4(1-\nu)(x+y) & 4(x+\nu y) & 12(\nu x+y)x & -8(1-\nu)xy+(12-8\nu)(x+y)^2 & 12(x+\nu y)y \\ 0 & 0 & 0 & 12\nu y & 0 & 12y & 36\nu xy & 12(x+\nu y)y & 36y^2 \end{array}\right] dxdy \qquad (18)$$

where $D = Eh^3/12(1-\nu^2)$ the flexural rigidity.

This matrix is usually formed on a computer and integrated by a numerical method over the area of the element. The matrix [A] is also formed and inverted numerically for each particular element in turn. The stiffness matrix is then found by multiplication of these matrices according to equation (17).

To form the explicit stiffness matrix it is necessary to invert the matrix [A] (equation (16)) using a Gauss Jordon elimination as described in Appendix 1. The integration over the area of the triangle is performed by substitution from the standard integrals prepared in Appendix 2 into the matrix (18).

So that the algebra programs described in Appendix 4 can be used the elements of matrix $[A]^{-1}$ are expressed as multiples of their lowest common denominator which is removed as a scalor multiplier of the whole matrix. Each of the elements of the matrix $[A]^{-1}$ given below should be multiplied by a factor:

$$1/ar^3t^3$$

where $a = (-r + 2s + t)$

The elements are indexed A(row, column) and all elements omitted have zero value.

$$A(1,1) = ar^3t^3$$
$$A(2,3) = -ar^3t^3$$
$$A(3,2) = ar^3t^3$$
$$A(4,1) = -3\,art^3$$
$$A(4,3) = 2\,ar^2t^3$$
$$A(4,4) = 3\,art^3$$
$$A(4,6) = ar^2t^3$$
$$A(5,1) = -6\,r^2st^2 + 6\,rs^2t^2$$
$$A(5,2) = -ar^2t^3 - r^3t^3$$
$$A(5,3) = -r^4t^2 + 4\,r^3st^2 - 3\,r^2st^2$$
$$A(5,4) = 6\,r^2st^2 - 6\,rs^2t^2$$

$$A(5,5) = ar^2t^3 + r^3t^3$$
$$A(5,6) = 2\ r^3st^2 - 3\ r^2s^2t^2$$
$$A(5,9) = r^4t^2$$
$$A(6,1) = -\ 3\ ar^3t - 3\ r^2s^2t - 3\ rs^2t^2$$
$$A(6,2) = -\ 2\ ar^3t^2 + 2\ ar^2st^2 - r^2st^3 - 2\ r^2s^2t^2 + 2\ r^3st^2$$
$$A(6,3) = -\ r^4st + 2r^3s^2t + 2r^3st^2 - r^2s^3t - 2r^2s^2t^2 - 3rs^4t + 3rs^3t^2$$
$$A(6,4) = -\ 3\ r^2s^2t - 3\ rs^2t^2$$
$$A(6,5) = -\ 2\ r^2s^2t^2 - r^2st^3$$
$$A(6,6) = -\ r^3s^2t + r^2s^3t - r^2s^2t^2$$
$$A(6,7) = 3\ ar^3t$$
$$A(6,8) = -\ ar^3t^2$$
$$A(6,9) = r^3st^2 + 2\ r^3s^2t - 2\ r^4st$$
$$A(7,1) = 2\ at^3$$
$$A(7,3) = -\ art^3$$
$$A(7,4) = -\ 2\ at^3$$
$$A(7,6) = -\ art^3$$
$$A(8,1) = 6\ rst^2 - 6\ s^2t^2$$
$$A(8,2) = r^2t^3$$
$$A(8,3) = r^3t^2 - 4\ r^2st^2 + 3\ rs^2t^2$$
$$A(8,4) = -\ 6\ rst^2 + 6\ s^2t^2$$
$$A(8,5) = -\ r^2t^3$$
$$A(8,6) = -\ 2\ r^2st^2 + 3\ rs^2\ t^2$$
$$A(8,9) = -\ r^3t^2$$
$$A(9,1) = -\ 2r^4 + 2s^4 + 6r^3t - 6\ rs^2t$$
$$A(9,2) = ar^3t - ar^2st + r^2s^2t - r^3st$$
$$A(9,3) = r^4s - 2\ r^3st - 3\ r^3s^2 + 4\ r^2s^2t^2 + 3\ r^2s^3 - 2\ rs^3t - rs^4$$
$$A(9,4) = 4\ rs^3 + 6\ rs^2t - 2\ s^4 - 4\ s^3t$$
$$A(9,5) = r^2s^2t + r^2st^2$$
$$A(9,6) = r^2s^3 + 2\ r^2s^2t - 2\ rs^3t - rs^4$$
$$A(9,7) = -\ 2\ ar^3$$
$$A(9,8) = ar^3t$$
$$A(9,9) = -\ r^3s^2 + r^4s$$

Elements with zero value are also omitted from the following listing of the matrix [Z] which is indexed as above.

$$Z(4,4) = 2\ rt$$
$$Z(4,6) = Z(6,4) = 2\ \nu rt$$
$$Z(4,7) = Z(7,4) = 2\ r^2t + 2\ rst$$
$$Z(4,8) = Z(8,4) = \tfrac{2}{3}\ \nu r^2t + \tfrac{2}{3}\ \nu rst + \tfrac{2}{3}\ rt^2$$

$$Z(4,9) = Z(9,4) = 2\,\nu r t^2$$
$$Z(5,5) = rt - \nu rt$$
$$Z(5,8) = Z(8,5) = \tfrac{2}{3}\, r^2 t + \tfrac{2}{3}\, rst + \tfrac{2}{3}\, \nu r t^2 - \tfrac{2}{3}\, \nu r^2 t - \tfrac{2}{3}\, \nu rst - \tfrac{2}{3}\, \nu r t^2$$
$$Z(6,6) = 2\, rt$$
$$Z(6,7) = Z(7,6) = 2\, \nu r^2 t + 2\, \nu rst$$
$$Z(6,8) = Z(8,6) = \tfrac{2}{3}\, r^2 t + \tfrac{2}{3}\, rst + \tfrac{2}{3}\, \nu rt$$
$$Z(6,9) = Z(9,6) = 2\, r t^2$$
$$Z(7,7) = 3\, r^3 t + 3\, r^2 st + 3\, rs^2 t$$
$$Z(7,8) = Z(8,7) = \nu r^3 t + \nu r^2 st + \nu rs^2 t + \tfrac{1}{2}\, r^2 t^2 + rst^2$$
$$Z(7,9) = Z(9,7) = 1\tfrac{1}{2}\, \nu r^2 t^2 + 3\, \nu rst^2$$
$$Z(8,8) = \tfrac{2}{3}\, r^2 t^2 + 1\tfrac{1}{3}\, rst^2 - \tfrac{1}{3}\, \nu r^2 t^2 - \tfrac{2}{3}\, \nu rst^2 + r^3 t + r^2 st + rs^2 t - \tfrac{2}{3} \nu r^3 t - \tfrac{2}{3}\, \nu r^2 st - \tfrac{2}{3}\, \nu rs^2 t + rt^3 - \tfrac{2}{3}\, \nu rt^3$$
$$Z(8,9) = Z(9,8) = \tfrac{1}{2}\, r^2 t^2 + rst^2 + \nu rt^3$$
$$Z(9,9) = 3\, rt^3$$

The matrices [Z] and $[A]^{-1}$ are then multiplied according to equation (17) using a specially designed program based on the prime number algebra technique given in Appendix 4.

It is found that the rows and columns of the explicit stiffness matrix corresponding to nodal rotations contain a common factor 'r'. Consider, now, the fundamental matrix equation of the stiffness method (9). A factor can be divided into a row of the matrix [K] and the same row of the vector {f} without changing the equality. A factor can also be divided into a column of [K] and multiply the corresponding row of the vector {d} without altering the equality. If the factor 'r' is removed from the rows and columns of [K] indicated above and introduced to the corresponding rows of the vectors {d} and {f} the equation (9) still holds. The advantage of this operation is that each matrix of the resulting equation has all its elements in the same units. The two vectors become:

$$\{d\}^t = [w_1, r\theta_{x1}, r\theta_{y1}, w_2, r\theta_{x2}, r\theta_{y2}, w_3, r\theta_{x3}, r\theta_{y3}]$$

and

$$\{f\}^t = [F_1, M_{x1}/r, M_{y1}/r, F_2, M_{x2}/r, M_{y2}/r, F_3, M_{x3}/r, M_{y3}/r]$$

The homogeneous polynomials representing the elements of the explicit stiffness matrix are then all of the same degree in r, s and t.

It is then only necessary to store the list of coefficients for each polynomial representing an element in a particular order.

A program using this explicit stiffness matrix generates a vector of the terms of a homogeneous polynomial in the variables r, s, t and ν. Each element of the stiffness matrix is then formed by a simple multiplication of this vector by the appropriate set of coefficients.

A saving is made when it is realised that:

the power of ν never exceeds 1,
the power of r never exceeds 8,
the power of s never exceeds 8,
the power of t never exceeds 6.

The homogeneous polynomials have units of length to the power 8. The factor by which the whole matrix must be multiplied is

$$Eh^3/\left(72(1 - \nu^2)\, a^2 r^6 t^3\right) \qquad (19)$$

Appendix 5 contains a listing of both the user program and the coefficients of the explicit stiffness matrix.

2.3.4 <u>Examination of the explicit stiffness matrix</u>
Two properties are worth examining.

The first is to consider the circumstances under which the Tocher stiffness matrix is singular. From the common factor (19), the matrix will be singular if

$$a^2 r^6 t^3 = 0.$$

Figure 2 shows that a singular matrix is the correct result if either r or t have a zero value. The problem arises when:

$$a = -r + 2s + t = 0 \tag{20}$$

It will be shown that if this situation does arise, it is only necessary to rotate the local axes so that the local x-axis lies along a different side of the element. If the new values of r, s and t still give a singular matrix, a final orientation of the local axes will definitely give a non-singular stiffness matrix.

So that the proof can be done in non-dimensional terms, the parameters r, s and t have been divided by r to give 1, x_i and y_i respectively as shown in figure 3.

For these non-dimensional parameters the value of variable 'a' from equation (20) is:

orientation I : $a = -1 + 2x_3 + y_3$ (21a)

orientation II : $a = -1 + 2x_1 + y_1$ (21b)

orientation III: $a = -1 + 2x_2 + y_2$ (21c)

It is now required to show that these three equations cannot be simultaneously equal to zero. More specifically, since a general triangle was chosen and since each orientation is described in exactly the same way, it must be shown that if (21a) and (21b) are equal to zero then (21c) is definitely not equal to zero.

i.e. given $2x_3 + y_3 = 1$
and $2x_1 + y_1 = 1$ } (22)
prove $2x_2 + y_2 \neq 1$

Proof

From figure 3:

$$\left.\begin{aligned} y_3 - x_3 \tan\alpha &= 0 \\ y_3 + x_3 \tan\beta &= \tan\beta \\ y_1 - x_1 \tan\beta &= 0 \\ y_1 + x_1 \tan\gamma &= \tan\gamma \\ y_2 - x_2 \tan\gamma &= 0 \\ y_2 + x_2 \tan\alpha &= \tan\alpha \end{aligned}\right\} \tag{23}$$

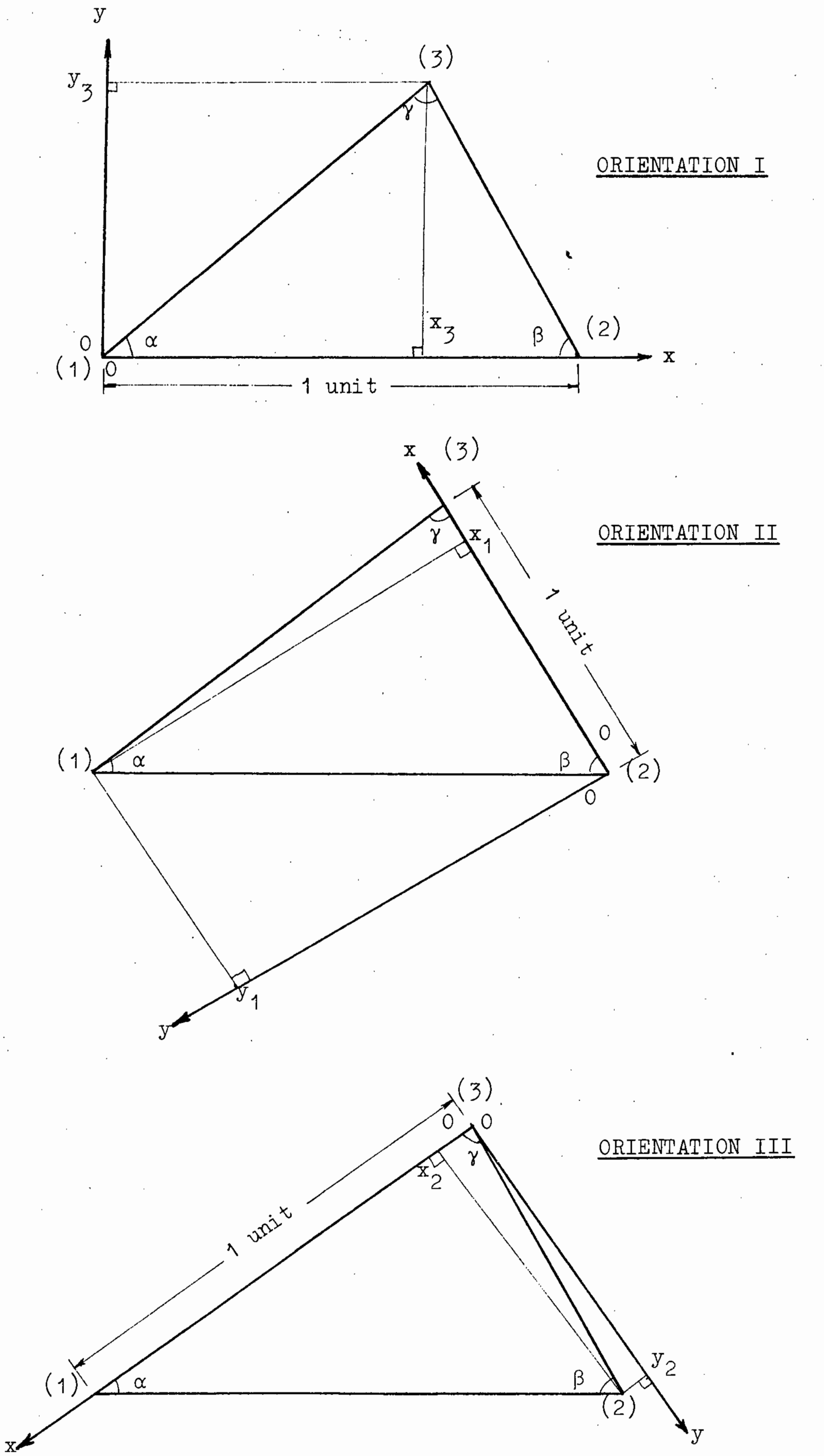


Figure 3: The Three Possible Positions of Local Axes on a Triangular Element of General Shape.

By elimination from (23):

$$x_3(\tan\alpha + \tan\beta) = \tan\beta$$
$$y_3(\tan\alpha + \tan\beta) = \tan\alpha \tan\beta$$

$$\therefore \quad 2x_3 + y_3 = \frac{2\tan\beta + \tan\alpha\tan\beta}{\tan\alpha + \tan\beta} = 1 \qquad (24)$$

from equation (22).

Similarly:

$$2x_1 + y_1 = \frac{2\tan\gamma + \tan\beta\tan\gamma}{\tan\beta + \tan\gamma} = 1 \qquad (25)$$

$$2x_2 + y_2 = \frac{2\tan\alpha + \tan\gamma\tan\alpha}{\tan\gamma + \tan\alpha} = \,? \qquad (26)$$

From (24):

$$\tan\beta = \frac{\tan\alpha}{1 + \tan\alpha} \qquad (27)$$

From (25):

$$\tan\beta = \frac{\tan\gamma}{1 - \tan\gamma} \qquad (28)$$

From (27) and (28):

$$\frac{\tan\alpha}{1 + \tan\alpha} = \frac{\tan\gamma}{1 - \tan\gamma}$$

$$\therefore \tan\alpha - \tan\gamma + \tan\alpha\tan\gamma = 3\tan\alpha\tan\gamma$$

$$\therefore \tan\alpha - \tan\gamma + \tan\alpha\tan\gamma = 0$$

if and only if α or γ = zero or π.

Thus for a triangle of finite non-zero area:

$$\tan\alpha - \tan\gamma + \tan\alpha\tan\gamma \neq 0$$

$$\therefore \frac{2\tan\alpha + \tan\alpha\tan\gamma}{\tan\alpha + \tan\gamma} \neq 1$$

Comparing this to equation (26):

$$2x_2 + y_2 \neq 1$$

Hence there will always be at least one orientation of any triangle, with finite, non-zero area, which will have a non-singular stiffness matrix.

The second important result of this investigation is the advantage in execution time of an explicit stiffness matrix formation over a numeric stiffness matrix formation.

A simple program using numerical matrices formed as described in the algorithm developed by Tocher was written. The results obtained by this method were identical to those obtained by a program using the explicit matrix. (Appendix 5). The comparison of the times of execution of the two matrices are summarised below.

TABLE 1: COMPARATIVE PROGRAM EXECUTION TIMES

Times are given in Secs and Millisecs	Numeric	Explicit
Compilation time	2,844	2,733
Time to form first stiffness matrix	8,947	8,661
Time to form two stiffness matrices	15,050	9,173
Additional time to form second matrix	6,103	0,512

The extra time taken by the explicit version in the formation of the first stiffness matrix is used to read in the coefficients. Once these are in core, they remain there until the stiffness matrices for all elements have been formed. Subsequent element stiffness matrices each take only 0,512 seconds. The numeric formation on the other hand, apart from the compilation, has to be executed in full for each stiffness matrix formation.

The saving in time for a large number of elements tends to:

$$100 \times (6,103 - 0,512)/6,103 = \underline{91,5\ \%}$$

This shows the value of an explicit version of a stiffness matrix. A more reliable element treated in the same way would be advantageous.

It must be observed that the explicit matrix requires more computer core space than an entirely numerical program. The space is taken up by the array of coefficients which requires less than 4K (3780 words) of core. This is negligible in a large computer (50 K or more) but makes the use of an explicit matrix impractical in a small computer (10 K or less).

2.4 The Assumed Stress Method

2.4.1 Method in general terms

This method is very similar to that of the previous section. Stress functions are assumed and the related strains are found from a knowledge of the material properties. Shape functions are chosen along the edges of the element to relate nodal displacements to the assumed stress functions.

It is inconvenient to use the principle of virtual work to find the internal energy caused by external work done. A minimisation of complementary energy is preferred.

The method (as developed by Pian[2]) is described in general terms below.

A series of functions, in terms of at least as many unknown constants as the element has degrees of nodal freedom, is chosen to describe the stress situation within the element. This is done arbitrarily except that stress equilibrium conditions must be satisfied for all values of x and y:

$$\underset{(p \times 1)}{\{\sigma(x,y)\}} = \underset{(p \times n)}{[P(x,y)]} \cdot \underset{(n \times 1)}{\{c\}} \qquad (29)$$

where: $\{\sigma(x,y)\}$ is a column vector of 'p' stresses at a point (x,y)

$[P(x,y)]$ is a matrix of terms of the chosen stress function,

$\{c\}$ is a column vector of 'n' arbitrary unknown constants, where 'n' is greater than or equal to 'm' the number of nodal degrees of freedom of the element.

A second set of functions is chosen to describe the displacements of an edge of the element. These functions are in terms of the displacements at the two nodes which terminate the edge and a distance 'a' from one of the nodes.

$$\{u(a)\} = [L(a)]\ \{d\} \qquad (30)$$

where: $\{u(a)\}$ is a vector of edge deflections. The number of edge deflections at the general point (a) corresponds to the number of degrees of freedom at a node.

$[L(a)]$ is a square matrix of terms of the chosen function.

$\{d\}$ is a vector of nodal displacements.

As the deflections of an edge depend only on the nodes which terminate it, the edges of adjacent elements conform perfectly under all circumstances.

By integrating appropriately the stress functions chosen in equation (29), the edge forces causing these stresses can be determined.

Let $\{S(a)\}$ be a vector of such edge forces corresponding term by term to the edge displacement vector $\{u(a)\}$ (equation (30)). $\{S(a)\}$ can be expressed in terms of the unknown constants $\{c\}$ if a matrix $[R(a)]$ is defined by:

$$\{S(a)\} = [R(a)]\ .\ \{c\}$$

From equation (29), by using the stress/strain relationships (14):

$$\begin{aligned}\{\epsilon(x,y)\} &= [D^{-1}]\ \{\sigma(x,y)\} \\ &= [D^{-1}].\ [P(x,y)]\ .\ \{c\} \qquad (31)\end{aligned}$$

where $\{\epsilon(x,y)\}$ is a column vector of strains corresponding to $\{\sigma(x,y)\}$.

A method of minimising the complementary energy with respect to the arbitrary constants is used to relate the external loading to the internal stress situation. The total internal energy of any elastic system with stiffness [K] and prescribed displacements {d} may be written as:

$$U = \frac{1}{2}\{d\}^t[K]\{d\} \tag{32}$$

In terms of stresses and strains, the total internal energy of the element is represented by:

$$U = \frac{1}{2}\int_{VOL}\Big(\{\epsilon(x,y)\}^t.\ \{\sigma(x,y)\}\Big)\, d\ vol$$

$$= \frac{1}{2}\int_{VOL}\Big(\{c\}^t.\ [P(x,y)]^t.\ [D^{-1}]\ .\ [P(x,y)]\ .\ \{c\}\Big)\, d\ vol \tag{33}$$

from equations (29) and (31). (Note that $[D^{-1}]$ is symmetrical).

The external work by the boundary loading {S(a)} moving through the boundary displacements {u(a)} is:

$$W = \int\Big(\{S(a)\}^t.\ \{u(a)\}\Big)\, da$$

$$= \int\Big(\{c\}^t.\ [R(a)]^t.[L(a)]\ \{d\}\Big)\, da \tag{34}$$

Defining:

$$[H] = \int_{VOL}\Big([P(x,y)]^t.\ [D^{-1}]\ .\ [P(x,y)]\Big)\, d\ vol$$

and: (35)

$$[G] = \int [R(a)]^t.[L(a)]\, da$$

The total complementary energy is expressed as the difference between W.D. and internal strain energy:

$$U - W = \frac{1}{2}\{c\}^t[H]\{c\} - \{c\}^t[G]\{d\}$$

Differentiating this scalor quantity with respect to each arbitrary constant in turn, equating them to zero and writing the resulting equations under each other:

$$[H]\{c\} = [G]\{d\}$$

Hence $\{c\} = [H^{-1}] \cdot [G] \{d\}$ (36)

Substituting into equation (33) from (35) and (36) and noting that [H] is symmetrical:

$$U = \tfrac{1}{2} \{d\}^t . [G]^t . [H^{-1}] . [G] . \{d\}$$

Comparing this with equation (32) gives:

$$[K] = [G]^t . [H^{-1}] . [G] \qquad (37)$$

The application of this method to polygonal plate bending elements was described by Severn and Taylor[3] and Allwood and Cornes[4]. The simplest element of this group, the triangle is described below.

A triangle of general shape is oriented in a local axis system in the same manner as was used in the assumed displacement method. (See figure 2). The following sign convention is used:

Shear stresses (τ) and strains (γ) are subscripted according to the plane in which they tend to cause rotation. The second subscript gives the axial direction of the stress or strain. If an acute angle forms at the origin of a rectangular element in the first quadrant of the plane, positive stresses and strains exist.

Direct stresses (σ) and strains (ϵ) are subscripted according to the direction in which they act and are positive in tension. A rectangular element, positively stressed and with positive strains, is shown in figure 4.

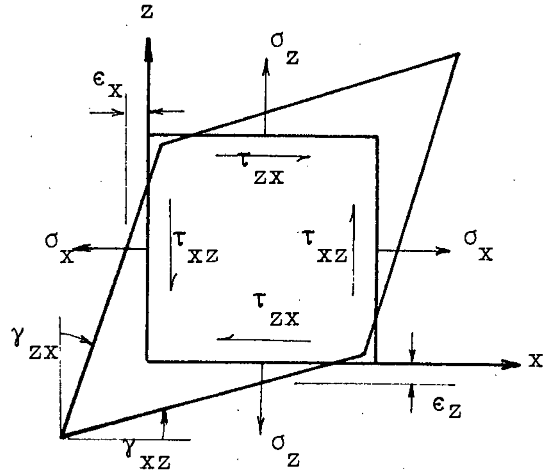


Figure 4: Positive Directions of Stresses and Strains on a Small Element

Unit moments (M) and unit shears (Q) are subscripted according to the axes about which they cause rotations. (The second subscript is the axis about which a twisting moment acts.) These unit moments are obtained by integrating the corresponding stresses between the limits $-h/2$ and $+h/2$, where h is the material thickness.

The positive directions of unit moments and shears are shown in figure 5.

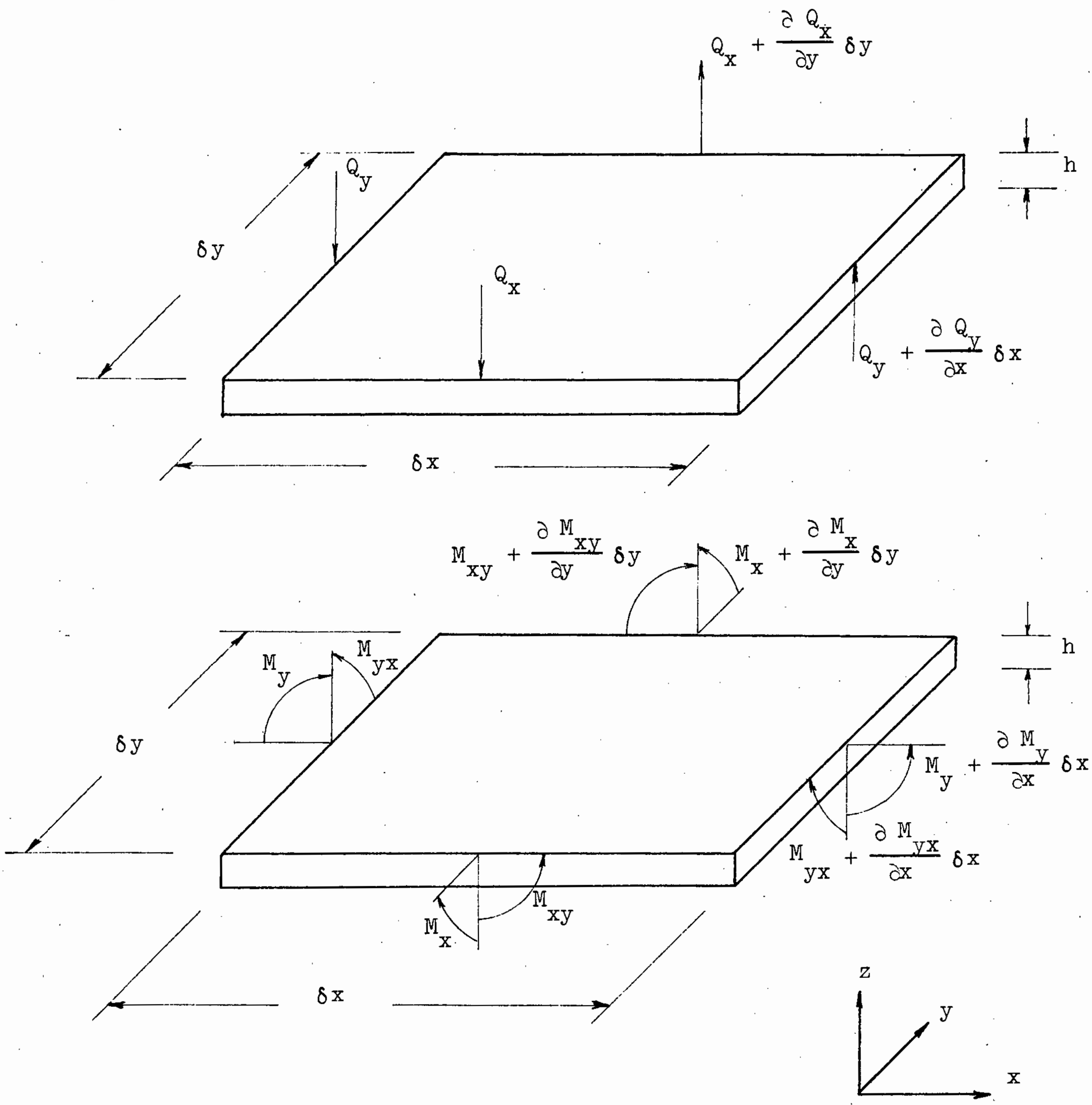


Figure 5: Positive Directions of Unit Shears and Moments on a Small Element

From the sign conventions: if $\sigma_y = 0$ when $z = 0$; σ_y is positive when z is positive and moments causing sagging are positive:

$$M_x = \int_{-h/2}^{h/2} \sigma_y \, z dz$$

Similarly:

$$M_y = \int_{-h/2}^{h/2} \sigma_x \, z dz$$

$$M_{xy} = \int_{-h/2}^{h/2} \tau_{yx} \, z dz \qquad (38)$$

$$Q_x = \int_{-h/2}^{h/2} \tau_{yz} \, dz$$

$$Q_y = \int_{-h/2}^{h/2} \tau_{xz} \, dz$$

Considering the equilibrium of an element it can be shown that:

$$\partial Q_x/\partial y + \partial Q_y/\partial x = 0 \qquad (39a)$$

$$\partial M_x/\partial y - \partial M_{xy}/\partial x + Q_x = 0 \qquad (39b)$$

$$- \partial M_y/\partial x + \partial M_{xy}/\partial y - Q_x = 0 \qquad (39c)$$

Also: $M_{xy} = M_{yx}$ (40)

The polynomials assumed to represent the stress distribution are as follows:

$$\sigma_x = (C_1 + C_2 x + C_3 y)\ 8z/h \qquad (41a)$$

$$\sigma_y = (C_4 + C_5 x + C_6 y)\ 8z/h \qquad (41b)$$

$$\tau_{xy} = (C_7 + C_8 x + C_9 y)\ 8z/h \qquad (41c)$$

$$\tau_{xy} = C_{10}\ (1 - 4z^2/h^2) \qquad (41d)$$

$$\tau_{yz} = C_{11}\ (1 - 4z^2/h^2) \qquad (41e)$$

From (38) and (41b):

$$M_x = \tfrac{2}{3} h^2(C_4 + C_5 x + C_6 y) \qquad (42a)$$

Similarly:

$$M_y = \tfrac{2}{3} h^2(C_1 + C_2 x + C_3 y) \qquad (42b)$$

$$M_{yx} = M_{xy} = \tfrac{2}{3} h^2(C_7 + C_8 x + C_9 y) \qquad (42c)$$

$$Q_x = \tfrac{2}{3} h\, C_{11} \qquad (42d)$$

$$Q_y = \tfrac{2}{3} h\, C_{10} \qquad (42e)$$

Hence from (39b) and (42):

$$C_{11} = h(C_8 - C_6)$$

Similarly:

$$C_{10} = h(C_9 - C_2)$$

Hence the matrix $[P(x,y)]$ defined in equation (29) is as follows:

$$[P(x,y)] = 1/h \begin{bmatrix} 8z & 8xy & 8yz & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 8z & 8yz & 8yz & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 8z & 8xz & 8yz \\ 0 & -(h^2-4z^2) & 0 & 0 & 0 & 0 & 0 & 0 & h^2-4z^2 \\ 0 & 0 & 0 & 0 & 0 & -(h^2-4z^2) & 0 & h^2-4z^2 & 0 \end{bmatrix} \begin{matrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \\ \tau_{xz} \\ \tau_{yz} \end{matrix}$$

The $[D^{-1}]$ matrix as defined in (31) is as follows:

$$[D^{-1}] = 1/E \begin{bmatrix} 1 & -\nu & 0 & 0 & 0 \\ -\nu & 1 & 0 & 0 & 0 \\ 0 & 0 & 2(1+\nu) & 0 & 0 \\ 0 & 0 & 0 & 2(1+\nu) & 0 \\ 0 & 0 & 0 & 0 & 2(1+\nu) \end{bmatrix} \begin{matrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \\ \tau_{xz} \\ \tau_{yz} \end{matrix}$$

Hense from equation (35) noting that:

$$\int_{-h/2}^{h/2} 2(1 + \nu)(h^2 - 8z^2)^2\, dz = \frac{16}{15}(1 + \nu)\, h^5$$

and

$$\int_{-h/2}^{h/2} z^2 \, dz \;=\; h^3/12$$

and defining $q \;=\; h^2/5(1 + \nu)$:

$$[H] = \frac{16h}{3E} \int_{AREA} \begin{bmatrix} 1 & x & y & -\nu & -\nu x & -\nu y & 0 & 0 & 0 \\ x & x^2+q & xy & -\nu x & -\nu x^2 & -\nu xy & 0 & 0 & -q \\ y & xy & y^2 & -\nu y & -\nu xy & -\nu y^2 & 0 & 0 & 0 \\ -\nu & -\nu x & -\nu y & 1 & x & y & 0 & 0 & 0 \\ -\nu x & -\nu x^2 & -\nu xy & x & x^2 & xy & 0 & 0 & 0 \\ -\nu y & -\nu xy & -\nu y^2 & y & xy & y^2+q & 0 & -q & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2(1+\nu) & 2(1+\nu)x & 2(1+\nu)y \\ 0 & 0 & 0 & 0 & 0 & -q & 2(1+\nu)x & 2(1+\nu)x^2+q & 2(1+\nu)xy \\ 0 & -q & 0 & 0 & 0 & 0 & 2(1+\nu)y & 2(1+\nu)xy & 2(1+\nu)y^2+q \end{bmatrix} dxdy \qquad (43)$$

The function relating the general displacements along an edge to the displacements of the nodes at its ends is derived from figure 6. A local 'a'-axis has its origin at the first node of each side and points along that side. A local 'b'-axis forms a right hand set with each 'a'-axis.

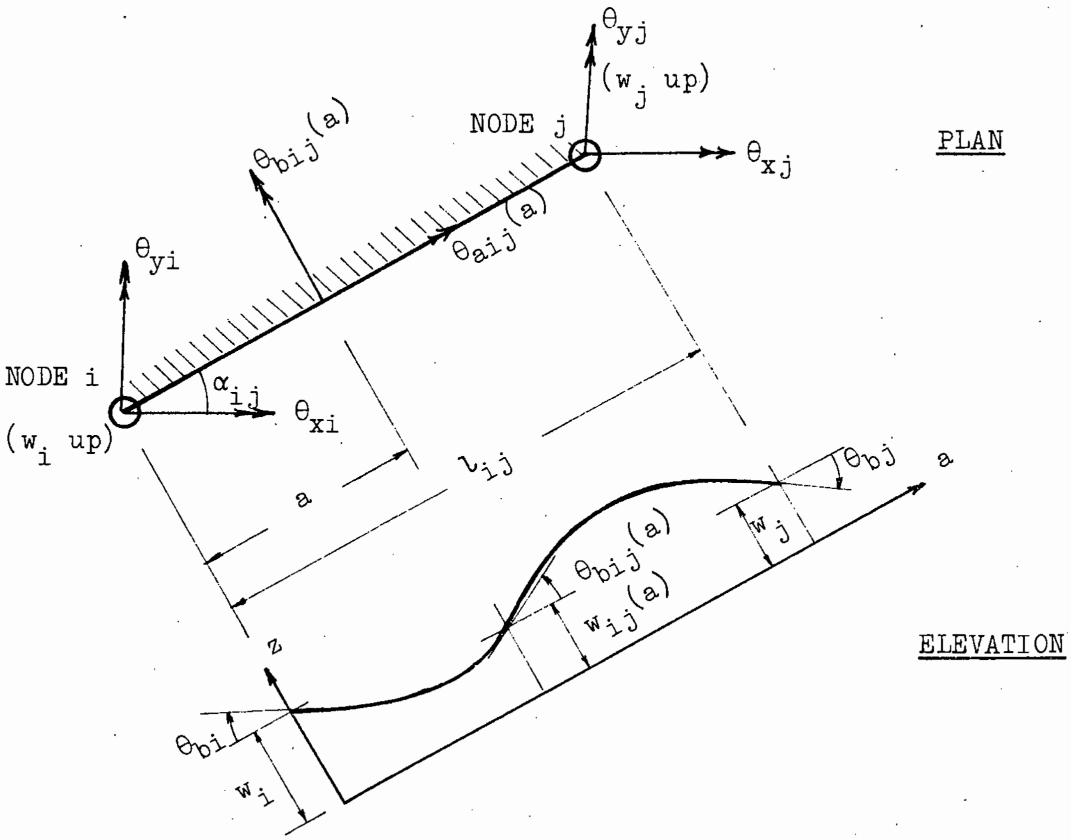


Figure 6: Displacements of the edge of an element.

The nodal displacements can be transformed to the (a,b) axis system as follows:

$$\theta_{ai} = \theta_{xi}\cos\alpha_{ij} + \theta_{yi}\sin\alpha_{ij}$$
$$\theta_{bi} = -\theta_{xi}\sin\alpha_{ij} + \theta_{yi}\cos\alpha_{ij}$$

The values of θ_{aj} and θ_{bj} are similarly related.

Assuming $\theta_{aij}(a)$ to be linear with respect to 'a':

$$\begin{aligned}\theta_{aij}(a) &= \theta_{ai}(l_{ij} - a)/l_{ij} + \theta_{aj}\, a/l_{ij} \\ &= (\theta_{xi}\cos\alpha_{ij} + \theta_{yi}\sin\alpha_{ij})(l_{ij} - a)/l_{ij} \\ &\quad + (\theta_{xj}\cos\alpha_{ij} + \theta_{yj}\sin\alpha_{ij})a/l_{ij}\end{aligned}$$

As there are four boundary values $(w_i,\ \theta_{bi},\ w_j$ and $\theta_{bj})$, $w_{ij}(a)$ can be chosen as a cubic polynomial in 'a':-

$$w_{ij}(a) = A + Ba/l_{ij} + Ca^2/l_{ij}^2 + Da^3/l_{ij}^3 \tag{44}$$

Solving these coefficients for $w_{ij}(a)$ and $dw_{ij}(a)/da$ at $a = 0$ and $a = l_{ij}$ and back substituting into (44) gives

$$\begin{aligned}w_{ij}(a) &= w_i(1 - 3a^2/l_{ij}^2 + 2a^3/l_{ij}^3) + w_j(3a^2/l_{ij}^2 - 2a^3/l_{ij}^3) \\ &\quad + l_{ij}(-\theta_{xi}\sin\alpha_{ij} + \theta_{yi}\cos\alpha_{ij})(-a/l_{ij} + 2a^2/l_{ij}^2 - a^3/l_{ij}^3) \\ &\quad + l_{ij}(-\theta_{xj}\sin\alpha_{ij} + \theta_{yj}\cos\alpha_{ij})(a^2/l_{ij}^2 - a^3/l_{ij}^3)\end{aligned}$$

Differentiation with respect to 'a' gives:

$$\begin{aligned}\theta_{bij}(a) &= 6(w_i - w_j)(a/l_{ij}^2 - a^2/l_{ij}^3) \\ &\quad +(-\theta_{xi}\sin\alpha_{ij} + \theta_{yi}\cos\alpha_{ij})(1 - 4a/l_{ij} + 3a^2/l_{ij}^2) \\ &\quad +(-\theta_{xj}\sin\alpha_{ij} + \theta_{yj}\cos\alpha_{ij})(-2a/l_{ij} + 3a^2/l_{ij}^2)\end{aligned}$$

From the dimensions chosen for the element (see figure 2):

$$\begin{aligned}
&l_{12} = r &&\sin\alpha_{12} = 0 &&\cos\alpha_{12} = 1 \\
&l_{23} = \sqrt{(r - s)^2 + t^2} &&\sin\alpha_{23} = t/l_{23} &&\cos\alpha_{23} = -(r - s)/l_{23} \\
&l_{31} = \sqrt{s^2 + t^2} &&\sin\alpha_{31} = -t/l_{31} &&\cos\alpha_{31} = -s/l_{31}
\end{aligned} \tag{45}$$

By substitution of these values into the assumed edge displacement functions above, the matrix [L] of equation (30) can be formed. If:

$$\{u(a)\}^t = \{w_{12}, \theta_{a12}, \theta_{b12}, w_{23}, \theta_{a23}, \theta_{b23}, w_{31}, \theta_{a31}, \theta_{b31}\}$$

and:

$$\{d\}^t = \{w_1, \theta_{x1}, \theta_{y1}, w_2, \theta_{x2}, \theta_{y2}, w_3, \theta_{x3}, \theta_{y3}\}$$

then a typical column of the matrix [L] is column two (given transposed here):

$$[0,\ 1 - a/r,\ 0,\ 0,\ 0,\ 0,\ t(a^2/l_{31}^2 - a^3/l_{31}^3),\ - as/l_{31}^2,\ (2a/l_{31}^2 - 3a^2/l_{31}^3)t] \quad (46)$$

(The full matrix is nine by nine)

The loading along the edges which causes the assumed stress situation (equations (41)) can be found by resolution (according to Mohr's circle).

The signs of the components of loading can be established from the sign conventions chosen (figure 5) and from figure 7.

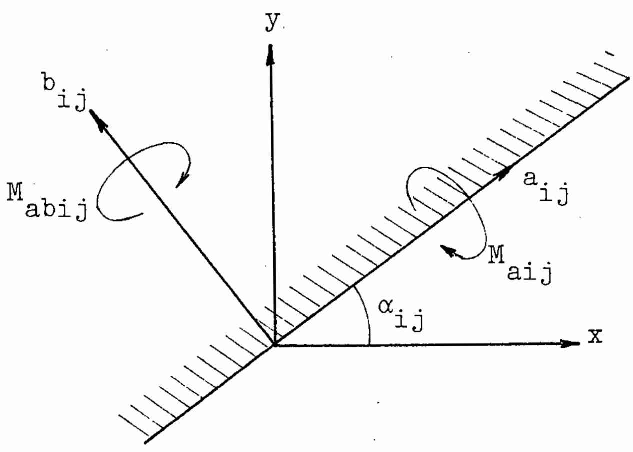


Figure 7: The positive directions of edge loading on an element

$$M_{aij} = - M_x \cos^2 \alpha_{ij} - M_y \sin^2 \alpha_{ij} - 2 M_{xy} \sin \alpha_{ij} \cos \alpha_{ij}$$
$$M_{abij} = - M_{xy}(\cos^2\alpha_{ij} - \sin^2\alpha_{ij}) + (M_x - M_y) \sin \alpha_{ij} \cos \alpha_{ij} \qquad (47)$$
$$Q_{aij} = Q_x \cos \alpha_{ij} - Q_y \sin \alpha_{ij}$$

As a one to one correspondence is required between the edge displacement vector $\{u(a)\}$ and the edge loading vector $\{S(a)\}$ (equation (34)), the vector

$$\{S(a)\}^t = \{- Q_{a12}, M_{a12}, M_{ab12}, - Q_{a23}, M_{a23}, M_{ab23}, - Q_{a31}, M_{a31}, M_{ab31}\}$$

Substituting into equations (47) for the values of $\sin \alpha_{ij}$ and $\cos \alpha_{ij}$ (from (45)) and the unit loads (from equations (42)), the values of the elements of $\{S(a)\}$ can be determined, e.g.

$$- Q_{a31} = \Big(s(C_8 - C_6) - t(C_9 - C_2)\Big) 2h^2/3l_{31}$$

The vector $\{S(a)\}$ may be represented as the product

$$\{S(a)\} = [R] \{c\}$$

where: $\{c\}^t = \{C_1, C_2, \ldots \ldots \ldots C_9\}$

and a typical column of the matrix $[R]$ is column five (given transposed here):-

$$[0, - a, 0, 0, -\Big(r - a(r - s)/l_{23}\Big) (r - s)^2/l_{23}^2,$$
$$-\Big(r - a(r - s)/l_{23}\Big) (r - s)t/l_{23}^2, 0, - (1 - a/l_{31})s^3/l_{31}^2,$$
$$(1 - a/l_{31})s^2t/l_{31}^2] \qquad (48)$$

(The matrix $[R]$ is a nine by nine matrix)

The matrix $[L]$ is premultiplied by the transpose of matrix $[R]$ (see equation (35)) and integrated round the boundaries with respect to 'a'.

The matrix [H] is integrated and inverted. The matrices $[H]^{-1}$ and [G] are then multiplied according to equation (37) to obtain the stiffness matrix.

These operations are usually performed numerically. A description of the algebraic manipulations required to obtain an explicit stiffness matrix is described in a subsequent subsection.

2.4.2. Problems and advantages

This element has definite advantages over the assumed displacement element, the stiffness matrix of which, was developed in the previous section.

No matter how this element is oriented within the local axis system, the stiffness matrix which results is exactly the same when transformed to a global axis system.

Provided that the element has finite, non-zero area, the stiffness matrix can never be singular.

When the element is rotated or translated in space without deforming, there is no increase in the internal energy. This is the correct behaviour which is violated by the assumed displacement triangle.

Elements with common sides deflect identically along that side. This compatability is also violated by the assumed displacement triangle.

It would be advantageous if a useful explicit stiffness matrix such as the one derived by the assumed displacement method could be produced. Unfortunately, the size of an explicit version of this matrix is so large that only a small portion of it can be fitted into the computer core space at a time. The time saved in computation would therefore be negated by the time taken to read the stiffness matrix piece by piece from disc storage for each element of a system.

Further development of the explicit version of this matrix is of academic interest only. Long lists of matrix elements have therefore been omitted. Certain problems were handled in a manner which may have a wider application. They are recounted in the following subsection.

2.4.3. Selected details of derivation

The integration and inversion of the [H] matrix:

The dimensions r, s and t of the triangle shown in figure 2 are substituted into the standard integration formulae given in Appendix 2. The integrals of all the functions of x and y which occur in the matrix [H] (equation (43) are:

$$\iint_{AREA} y^2 dxdy = 1/12\ rt^3$$

$$\iint_{AREA} x^2 dxdy = 1/12\ r^3t + 1/12\ r^2st + 1/12\ rs^2t$$

$$\iint_{AREA} xy\ dxdy = 1/24\ r^2t^2 + 1/12\ rst^2$$

$$\iint_{AREA} y\ dxdy = 1/6\ rt^2$$

$$\iint_{AREA} x\ dxdy = 1/6\ r^2t + 1/6\ rst$$

$$\iint_{AREA} 1\ dxdy = \tfrac{1}{2}\ rt$$

Substituting these functions into the matrix [H] (defined in equation (43)) the explicit version is as follows:-

$$[H] = [A] + [B]$$

where:

$$[B] = \frac{8hrt(1-\nu^2)}{3E} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & c/12 & 0 & 0 & 0 & 0 & 0 & 0 & -c/12 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & c/12 & 0 & -c/12 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -c/12 & 0 & c/12 & 0 \\ 0 & -c/12 & 0 & 0 & 0 & 0 & 0 & 0 & c/12 \end{bmatrix} \quad (49)$$

$$c = 12h^2/5(1-\nu)$$

$$[A] = \left[\begin{array}{c:c:c} [A_1] & -\nu\,[A_1] & [0\ ] \\ -\nu\,[A_1] & [A_1] & [0\ ] \\ [0\ ] & [0\ ] & 2(1+\nu)[A_1] \end{array}\right] \times \frac{8hrt}{3E}$$

$$[A_1] = \left[\begin{array}{c:c:c} 1 & \frac{1}{3}(r+s) & \frac{1}{3}t \\ \frac{1}{3}(r+s) & \frac{1}{6}(r^2+rs+s^2) & \frac{1}{12}(rt+2st) \\ \frac{1}{3}t & \frac{1}{12}(rt+2st) & \frac{1}{6}t^2 \end{array}\right]$$

(49) (contd)

It is now necessary to find the inverse of the $[H]$ matrix. If the component matrices are added together and a direct inversion by the Gauss Jordan methods of Appendix 1 attempted, the complexity of the final steps is so great that an error-free solution is difficult to obtain. A more suitable method of inversion (by parts) was then developed as follows:-

By definition: $[H] \times [H]^{-1} = [I]$ where $[I]$ is the identity matrix.

Substituting from equations (49):

$$[A + B] \times [A + B]^{-1} = [I]$$

Premultiplying both sides of this equation by $[A]^{-1}$ (a non-singular matrix):

$$[I + A^{-1} B] \times [A + B]^{-1} = [A]^{-1}$$

Premultiplying both sides by $[I + A^{-1} B]^{-1}$ gives:

$$[H]^{-1} = [A + B]^{-1} = [I + A^{-1} B]^{-1} \times [A]^{-1}$$

Defining: $[I + A^{-1} B] = [E]$ (50)

$$[H]^{-1} = [E]^{-1} . [A]^{-1} \qquad (51)$$

This may not appear to be an important result until one realises that the matrix [A] is easily inverted and that because of the nature of matrix [B], the matrix [E] is comparatively easily inverted. A computerised algebraic multiplication gives the inverse of [H] with little trouble.

The matrix [A] is inverted in two levels by partitioning and then using the Gauss Jordan method.

$$[A]^{-1} = 3E/8hrt(1 - \nu^2) \left[\begin{array}{c|c|c} [A_1]^{-1} & \nu\,[A_1]^{-1} & [0\,] \\ \nu\,[A_1]^{-1} & [A_1]^{-1} & [0\,] \\ [0\,] & [0\,] & (1 - \nu)/2[A_1]^{-1} \end{array}\right]$$

and:

$$[A_1]^{-1} = \left[\begin{array}{l|l|l} 9 & -\,12/r & -\,12(r - s)/rt \\ -\,12/r & 24/r^2 & 12(r - 2s)/r^2t \\ -\,12(r - s)rt & 12(r - 2s)/r^2t & 24(r^2 - rs + s^2)/r^2t^2 \end{array}\right]$$

The matrix [E] is then formed from the definition in equation (50):

$$[E] = \left[\begin{array}{ll|l|l|l|l|l|l|l}
1 & -\,c/r & 0 & 0 & 0 & -\,\nu c(r - s)/rt & 0 & \nu c(r - s)/rt & c/r \\
0 & 1 + 2c/r & 0 & 0 & 0 & \nu c(r - 2s)/r^2t & 0 & -\,\nu c(r - 2s)/r^2t & -\,2c/r^2 \\
0 & c(r - 2s)r^2t & 1 & 0 & 0 & 2\nu c(r^2 - rs + s^2)/r^2t^2 & 0 & -\,2\nu c(r^2 - rs + s^2)/r^2t^2 & -\,c(r - 2s)/r^2t \\
0 & -\,\nu c/r & 0 & 1 & 0 & -\,c(r - s)/rt & 0 & c(r - s)/rt & \nu c/r \\
0 & 2\nu c/r^2 & 0 & 0 & 1 & c(r - 2s)/r^2t & 0 & -\,c(r - 2s)/r^2t & -\,2\nu c/r^2 \\
0 & \nu c(r - 2s)/r^2t & 0 & 0 & 0 & 1 + 2c(r^2 - rs + s^2)/r^2t^2 & 0 & -\,2c(r^2 - rs + s^2)/r^2t^2 & -\,\nu c(r - 2s)/r^2t \\
0 & (1 - \nu)c(r - s)/2rt & 0 & 0 & 0 & (1 - \nu)c/2r & 1 & -\,(1 - \nu)c/2r & -\,(1 - \nu)c(r - s)/2rt \\
0 & -\,(1 - \nu)c(r - 2s)/2r^2t & 0 & 0 & 0 & -\,(1 - \nu)c/r^2 & 0 & 1 + (1 - \nu)c/r^2 & (1 - \nu)c(r - 2s)/2r^2t \\
0 & -\,(1 - \nu)c(r^2 - rs + s^2)/r^2t^2 & 0 & 0 & 0 & -\,(1 - \nu)c(r - 2s)/2r^2t & 0 & (1 - \nu)(r - 2s)/2r^2t & 1 + (1 - \nu)c(r^2 - rs + s^2)/r^2t^2
\end{array}\right]$$

As $([E]^{-1})^t = ([E]^t)^{-1}$ it is permissible and convenient to invert the matrix [E] by a column-wise Gauss Jordan elimination. The addition of column 9 to column 2 and column 8 to column 6, immediately eliminates all but two unit values in each of column 2 and column 6. Unit values are then used to eliminate all except a small 2 by 2 submatrix. This submatrix is reduced to an identity matrix by assigning new variables, f and g, to the two lengthy expressions which develop on the leading diagonal. This redefinition of variables simplifies the division. f and g are included in the expression in the stiffness matrix.

The matrix $[E]^{-1}$ is then represented as a nine by nine matrix of zeroes with unit values on the leading diagonal except for columns two, six, eight and nine, where a typical element is:

$$\begin{aligned} E(1,\ 9) = & - rt^2cg + 0{,}5\ r^3t^2c^2\nu gf - 1{,}5\ r^2st^2c^2\nu gf + rs^2t^2c^2\nu gf \\ & + 0{,}5\ r^3t^3c^2\nu^2 gf - 1{,}5\ r^2st^2c^2\nu^2 gf + rs^2t^2c^2\nu^2 gf \end{aligned}$$

where: if h is the material thickness

$$\begin{aligned} c &= 12h^2/\left(5(1-\nu)\right) \\ f &= 1/\left(r^2t^2 + (1-\nu)ct^2 + 2c(r^2 - rs + s^2)\right) \\ g &= 1/\Big((r^2 - rs + s^2)c(1-\nu) + r^2t^2 + 2ct^2 - \tfrac{1}{4}(r^2 - 4rs \\ &\qquad + 4s^2)(1+\nu)^2t^2c^2f\Big) \end{aligned}$$

A program using the computer algebra by powers (described in Appendix 4) is then used to multiply the matrices $[E]^{-1}$ and $[A]^{-1}$ according to equation (51).

This is a far more practical way of inverting the [H] matrix than a direct inversion by hand.

The formation of the matrix [G]

The operations for the formation of the matrix [G], defined in equation (35) are simple enough to be performed by hand. It will be noticed that the matrices [L] and [R] ((46) and (48)) can be partitioned in sets of three rows. Each submatrix includes only expressions referring to one particular edge of the element shown in figure 2. When the matrix [L] is premultiplied by the transpose of matrix [R], the terms of the result can still be singled out as referring to a particular side. The integration round the perimeter can be performed as the sum of separate linear integrations along each side.

If: $f(a) = \text{terms } (a)_{12} + \text{terms } (a)_{23} + \text{terms } (a)_{31}$

then

$$\int f(a)\, da = \int_0^{l_{12}} \text{terms } (a)_{12} da + \int_0^{l_{23}} \text{terms } (a)_{23} da + \int_0^{l_{31}} \text{terms } (a)_{31} da$$

where l_{ij} is the length of side ij.

A typical example of the calculation of an element of the matrix [G] would be:

$$G(5, 2) = 2h^2/3 \Big(\int_0^{l_{12}} (- a + a^2/r) da + \int_0^{l_{31}} \Big((a - a^2/l_{31}) s^4/l_{31}^4$$

$$+ (2a/l_{31} - 5a^2/l_{31}^2 + 3a^3/l_{31}^3) s^2t^2/l_{31}^3 \Big) da \Big)$$

$$= h^2(- l_{12}^2 + (2s^4 + s^2t^2)/18\ l_{31}^2)$$

So that the matrix [G] can be used in the computer algebra programs, the following factor is removed from the whole matrix.

$$h^2/\Big(18(r^2s^2 - 2rs^3 + s^4 + 2s^2t^2 + t^4 - 2rst^2 + r^2t^2)\Big) \qquad (52)$$

The values of the lengths of the sides of the finite element are also substituted (from equations (45)). The example from the matrix [G] becomes:

$$G(5, 2) = - 2r^2t^4 + 4r^3st^2 - 2r^4t^2 + s^2t^4.$$

<u>The formation of the matrix [K]</u>

The multiplication of the matrices $[H]^{-1}$ and [G] according to equation (37) to form the stiffness matrix [K] appears to be a simple matter.

In fact the size of these explicit matrices is a problem. It is impossible to fit the last two matrices to be multiplied together, into the computer core. The following steps are used to overcome this problem.

Two programs are written, one to perform a multiplication of two explicit matrices and the other to transpose an explicit matrix. The computer algebra by powers (Appendix 4) is used in these programs.

The matrix [K] can be formed (from equations(37) and (51)) as follows:-

1.	Postmultiply	$[E]^{-1}$ by	$[A]^{-1}$	to get $[H]^{-1}$
2.	Postmultiply	$[H]^{-1}$ by	$[G]$	to get $[H^{-1}G]$
3.	Transpose	$[H^{-1}G]$ to get $[H^{-1}G]^t$		$= [G^tH^{-1t}]$
		(as [H] is symmetric)		$= [G^tH^{-1}]$
4.	Postmultiply	$[G^tH^{-1}]$ by	$[G]$	to get $[K]$

It will be noted that each time the multiplication program is applied, the premultiplying matrix is the one with the most terms in its elements.

This fact is used to reduce the core capacity requirements. Only one row of the premultiplier is read into core at once. The whole of the post-multiplier is kept in core. Once a row of the resultant matrix is complete, the next row of the premultiplier is read from storage to overwrite the previous row. (It would be wasteful to read in only one column of the post-multiplier at a time as the same column would have to be continually recalled after each new row of the premultiplier was introduced.)

Even this space saver is not enough to fit the final multiplication into core. The final manipulation which eventually allows the matrix multiplication to run it course is as follows:-

An examination of the matrices [E], [A] and [G] shows that the variables ν, c, g and f only occur in the matrix [E]. A closer inspection of the matrix $[E]^{-1}$ reveals that these variables occur with r, s and t only in seventeen different combinations. The powers of these combinations given in the order c, ν, g, f are:

0000, 0100, 1001, 1101, 1201, 1010, 1110, 1210, 2011,
2111, 2211, 2311, 3012, 3112, 3212, 3312, 3412.

The matrix $[E]^{-1}$ can be expressed as the sum of seventeen component matrices in terms of r, s and t and each multiplied by a factor comprising the variables c, ν, g and f.

The whole formation of the matrix [K] can then be performed component by component and the final result formed as a sum of all these matrices.

2.4.4. Results of investigations

Part of the explicit matrix was printed out. This required more than fifty pages. To use such an explicit matrix in practice would require each component to be read into the computer in turn. The time taken to do this would greatly exceed the execution time of an algorithmic formation.

Further investigation into this explicit matrix was considered to be unjustified.

CHAPTER III

THE ASSUMED STIFFNESS MATRIX METHOD

3.1 Introduction

In the previous chapter, it was shown that if the whole explicit stiffness matrix was small enough to be fitted into core together with other matrices which are required contemporarily, there was a significant saving in the program execution time. It was also evident that it is not very practical to obtain an explicit matrix by following a numerical algorithm.

The developments of the previous chapter were useful to the present section in showing the nature of an explicit, plate bending, triangular element stiffness matrix. The characteristics to be noted are:

(i) The elements of a stiffness matrix usually have a common factor which can be removed as a scalor multiplier.

(ii) This factor is a simple function of Young's modulus (E), Poisson's ratio (ν), the thickness of the element (h) and has a denominator which is an expression in the plan dimensions of the triangle. (See factors (19), (49) and (52)).

(iii) The elements of the matrices are then homogeneous polynomials in the plan dimensions of the triangle. The degree of the polynomial is determined by the units of the factor in (ii) above and the units of the corresponding rows of the load vector {f} and displacement vector {d}.

Using this format it is possible to assume the layout of a stiffness matrix in terms of fairly arbitrarily chosen functions involving a number of unknown coefficients. The choice of these functions and the solution for the numerical values of the coefficients form the basis of this investigation.

3.2 Method in General Terms

The fundamental equation of the stiffness method is:

$$[K]\{d\} = \{f\} \tag{53}$$

where $[K]$ is a square matrix representing the stiffness of a structure or a discrete part of a structure;

$\{d\}$ is a column vector representing the displacements of chosen fixed nodes on that structure or structural element;

$\{f\}$ is a column vector representing the effective loads on the nodes.

In the usual case, the stiffness of the structure and the loading on it are known. The displacements are to be found by solution of equation (53).

In the assumed stiffness matrix method, the underlying principle is as follows:-

A certain discrete structural element, the stiffness of which is not known, is subjected to a number of different loadings. A number of displacement vectors $\{d_i\}$ corresponding to each load vector $\{f_i\}$ is found either by measurement or from an independent calculation. If the nodes of the element have a total of m degrees of freedom, a total of m linearly independent displacement vectors must be derived.

The equations (of the stiffness method (53)) representing all of the load cases written as a single equation are:

$$[K][D] = [F]$$

where

$$[D] = [\{d_1\}, \{d_2\} \ldots\ldots\ldots\ldots \{d_m\},]$$
$$[F] = [\{f_1\}, \{f_2\} \ldots\ldots\ldots\ldots \{f_m\},]$$

As the columns of matrix $[D]$ are linearly independent, the matrix is non-singular. The unknown stiffness matrix may then be solved uniquely as:

$$[K] = [F].[D]^{-1}$$

This stiffness matrix may then be used to predict the displacement vector for any load situation on the element.

Consider the characteristics of an explicit stiffness matrix enumerated in the previous section. From these, the common factor and the homogeneous polynomials representing the stiffness matrix of this type of element can be assumed. The formula for each element of the matrix contains a number of unknown coefficients. The dimensions of the structural element considered above are substituted into one of these formulae. The resulting expression is put equal to the corresponding numerical value in the stiffness matrix found for the element. An equation in the unknown coefficients of that assumed function has been set up. This can be repeated for each element of the stiffness matrix.

Supposing each polynomial of the assumed stiffness matrix involves n unknown coefficients. If n different examples of this type of element were treated as before, n equations for the coefficients in each polynomial could be set up. Provided that these equations were not linear combinations of each other, the values of the coefficients could be found. Two equations would only be linear combinations of one another if the structural elements considered were similar. It is therefore important to consider structural elements with a range of different proportions. (A trivial example of this derivation of a stiffness matrix is given as an illustration in the next subsection (3.2.1). A pin-ended bar is shown.)

If the degree of the denominator of the common factor of the stiffness matrix was chosen to be very low, the degree of the polynomials in the matrix elements will also be low. This means that only very few coefficients are involved. If the degree is too low, all the coefficients may take the value zero, which is clearly incorrect. If a different set of structural elements gives different values for the coefficients, then the chosen function is also incorrect. The degree of the functions must be increased until a reasonable result is obtained.

If the structural element is connected to its neighbouring elements through its nodes only, provided that the correct functions have been chosen, the result will be unique.

A finite element of a continuum is connected to neighbouring elements through continuous edges, although it is considered to be joined only at its nodes. No unique stiffness matrix exists for such an element as its stiffness depends on the surrounding contact.

To use the assumed stiffness method on a finite element of a continuum, therefore, it is meaningless to consider isolated pieces of material of the chosen shape. All the elements must be treated as part of the continuum.

To find the stiffness matrix for a particular type of element, a single standard structure built up entirely from elements of that type, must be considered. The displacements of all points on the structure must be known for some standard load case. The structure is arranged to have as many degrees of nodal freedom as the assumed stiffness matrix has unknown coefficients. All of the elements must have different proportions to one another. A system stiffness matrix containing the unknown coefficients is set up in the usual manner. This matrix is multiplied by the known displacement vector and set equal to the known load vector. As there is a row in the stiffness equations for each degree of freedom of the structure, there will be enough equations to solve for the unknown coefficients.

A problem which arises is that the number of coefficients necessary to form a stiffness matrix of even very simple functions, is very large. This means that the number of elements required on the standard structure is also large and a wide range of proportions of elements is difficult to obtain. Solutions of coefficients are therefore not very accurate.

Some of the coefficients can be expressed in terms of others by:

(a) application of the Betti-Maxwell reciprocal theorem,

(b) a systematic choice of local coordinate axes and element dimensions,

(c) ensuring thatthe requirements of rigid body movements are fulfilled.

This reduces the number of independent unknown values to a minimum.

This section of the thesis is an exploratory study of a method with a great potential for further development and improvement. A final answer is obtained which gives very encouraging results.

3.2.1. Illustrative example of the Assumed Stiffness Matrix Method

Supposing the explicit stiffness matrix for a pin-ended bar was required. A tensile testing machine and three steel bars with different dimensions were available.

Each bar in turn is placed in the machine which has two moving heads.

The first bar has a length (l) of 2 m, a cross-section (A) of 4 cm^2 and Young's modulus of elasticity (E) of 200 GN/m^2. The directions of the displacements (w) and forces (f) on the ends of the bar are given in figure 8.

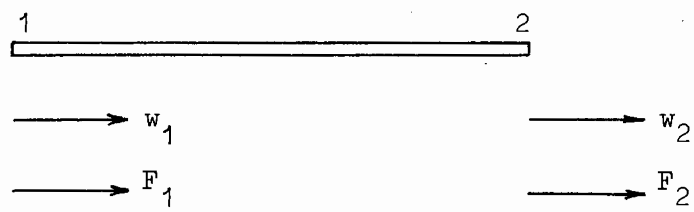


Figure 8: Positive directions of forces and displacements of a pin-ended bar.

Supposing the bar is loaded twice so that the displacements of the ends are respectively:

$$\{d_1\} = \begin{bmatrix} -0,001 \\ 0 \end{bmatrix} \text{m} \quad \text{and} \quad \{d_2\} = \begin{bmatrix} -0,001 \\ -0,0005 \end{bmatrix} \text{m}$$

The corresponding load vectors are found to be:

$$\{f_1\} = \begin{bmatrix} -40 \\ 40 \end{bmatrix} \text{kN}, \qquad \{f_2\} = \begin{bmatrix} -20 \\ 20 \end{bmatrix} \text{kN}$$

Using the method described in the previous section, the stiffness matrix for this bar is found to be:

$$[K_1] = \left[\begin{array}{c|c} -40 & -20 \\ 40 & 20 \end{array}\right] \times \left[\begin{array}{c|c} -0{,}001 & -0{,}001 \\ 0 & -0{,}0005 \end{array}\right]^{-1}$$

$$= 4 \times 10^4 \left[\begin{array}{c|c} 1 & -1 \\ -1 & 1 \end{array}\right]$$

The units of an element of this matrix $[K]$ are kN/m. Supposing the form of $[K]$ was assumed from the units to be:

$$[K] = \frac{AE}{l}\left[\begin{array}{c|c} C_1 & C_2 \\ C_3 & C_4 \end{array}\right] \text{kN/m}$$

Substitute the values of A, E and l and equate this to the numerical matrix, then:

$$C_1 = C_4 = -C_2 = -C_3 = 1$$

A different form of the stiffness matrix may have been assumed from the units of its elements. This could have been:

$$[K] = \frac{E}{Al}\left[\begin{array}{c|c} C_1A^2 + C_5l^4 & C_2A^2 + C_6l^4 \\ C_3A^2 + C_7l^4 & C_4A^2 + C_8l^4 \end{array}\right] \text{kN/m}$$

As eight coefficients must now be found, two bars with two degrees of freedom and two load vectors each must be used to set up the equations. Suppose the second bar tested in the machine had a length (l) of 4 m and all other properties the same as the first bar. For convenience the displacement vectors were arranged to be the same as before. The loads to cause these displacements were:

$$\{f_1\} = \begin{bmatrix} -20 \\ 20 \end{bmatrix} kN \quad ; \quad \{f_2\} = \begin{bmatrix} -10 \\ 10 \end{bmatrix} kN$$

The stiffness matrix would be found (in the manner described before) to be:

$$[K_2] = 2 \times 10^4 \left[\begin{array}{c|c} 1 & -1 \\ -1 & 1 \end{array}\right]$$

By substituting into element (1, 1) of the assumed matrix:

$$C_1 \times 4 \times 10^4 + C_5 \times 4 \times 10^{12} = 4 \times 10^4 \quad \text{from bar 1}$$
$$C_1 \times 2 \times 10^4 + C_5 \times 3,2 \times 10^{13} = 2 \times 10^4 \quad \text{from bar 2.}$$

By similar substitution into all of the elements and solution of the resulting equations:

$$C_1 = C_4 = -C_2 = -C_3 = 1$$
$$C_5 = C_6 = C_7 = C_8 = 0$$

If a test on the third bar gave linear combinations of the above equations then the correct explicit matrix has been found. If an inconsistent set of equations results, the wrong function has been assumed. If there is no better function then the best approximate stiffness matrix could be found by a least squares (see Appendix 3) solution of the set of equations (which has more equations than unknown values).

In the case of a finite element of a continuum, a function cannot be obtained to give exact results in every application. The best matrix can only be found by trial. A solution which gives satisfactory results is presented in this thesis.

3.3 Derivation Details

Certain principles are used in the detailed derivation of the explicit stiffness matrix by the assumed stiffness matrix method. As their full description is fairly lengthy and would break the line of argument if they were described where they are needed, they are described in detail first.

3.3.1. The requirements of Rigid Body Movements

If a structural element translates or rotates in space without undergoing any deformation, no net work is done on that element.

If the nodes of a finite element move in such a way that their positions relative to one another remain unchanged, a rigid body movement has taken place. On an element undergoing a rigid body movement, the net nodal loads corresponding to all non-zero nodal displacements must be zero if no external work is done.

Suppose, for example, a certain element of a structural system undergoes a rigid body movement for a certain loading. If a particular nodal displacement of that element is non-zero, then the load corresponding to that degree of freedom of the structure is carried only by the other elements of the system common to that node.

This principle must hold for any shape of element and for all possible rigid body movements. In the case of a small deflection theory, plate bending element, rotations about two non-parallel axes in its plane and a translation perpendicular to its plane describe all possible rigid body movements. In the case of a triangular element with sides of length a, b and c, the rigid body movement principle must hold identically for all values of a, b and c.

3.3.2. Trigonometric properties of a triangle

Consider the following triangle with dimensions a, b and c the lengths of the three sides:

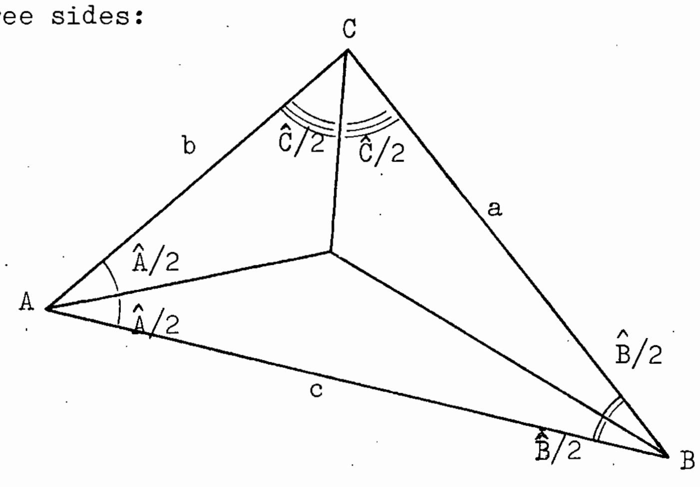


Figure 9: Dimensions of a general triangle

Define:

$$\alpha = \sqrt{-a+b+c}$$

$$\beta = \sqrt{a-b+c}$$

$$\gamma = \sqrt{a+b-c}$$

$$\Delta = \sqrt{a+b+c}$$

From the cosine formula for a triangle:

$$\cos \hat{A} = (b^2 + c^2 - a^2)/2bc \tag{54}$$

Also, for any angle $\hat{A}$:

$$\cos \hat{A} = \cos^2(\hat{A}/2) - \sin^2(\hat{A}/2)$$

$$= 2\cos^2(\hat{A}/2) - 1 \tag{55a}$$

$$= 1 - 2\sin^2(\hat{A}/2) \tag{55b}$$

Equating (54) and (55a):

$$2\cos^2(\hat{A}/2) - 1 = (b^2 + c^2 - a^2)/2bc$$

Hence

$$\cos(\hat{A}/2) = \tfrac{1}{2}\,\alpha\Delta/\sqrt{bc}$$

Similarly from (54) and (55b):

$$\sin(\hat{A}/2) = \tfrac{1}{2}\,\beta\gamma/\sqrt{bc}$$

Hence, applying these principles at each angle in turn:

$$\begin{aligned}
\cos(\hat{A}/2) &= \tfrac{1}{2}\,\alpha\Delta/\sqrt{bc} \\
\sin(\hat{A}/2) &= \tfrac{1}{2}\,\beta\gamma/\sqrt{bc} \\
\cos(\hat{B}/2) &= \tfrac{1}{2}\,\beta\Delta/\sqrt{ca} \\
\sin(\hat{B}/2) &= \tfrac{1}{2}\,\gamma\alpha/\sqrt{ca} \\
\cos(\hat{C}/2) &= \tfrac{1}{2}\,\gamma\Delta/\sqrt{ab} \\
\sin(\hat{C}/2) &= \tfrac{1}{2}\,\alpha\beta/\sqrt{ab}
\end{aligned} \qquad (56)$$

The area of the triangle may be found from:

$$\text{Area} = \tfrac{1}{2}\,bc\sin\hat{A} \qquad (57)$$

and for any angle of $\hat{A}$

$$\begin{aligned}
\sin\hat{A} &= 2\sin(\hat{A}/2)\cos(\hat{A}/2) \\
&= \alpha\beta\gamma\Delta/2bc
\end{aligned} \qquad (58)$$

from equations (56) above. Substitution into (57) from (58) gives:

$$\text{Area} = \alpha\beta\gamma\Delta/4 \qquad (59)$$

3.3.3. Oblique coordinates and Rigid Body Movements for a General triangle

For the purpose of applying rigid body movements to a triangular plate bending element with dimensions a, b and c the lengths of the three sides, it is convenient to define a set of oblique coordinates at each node as follows:-

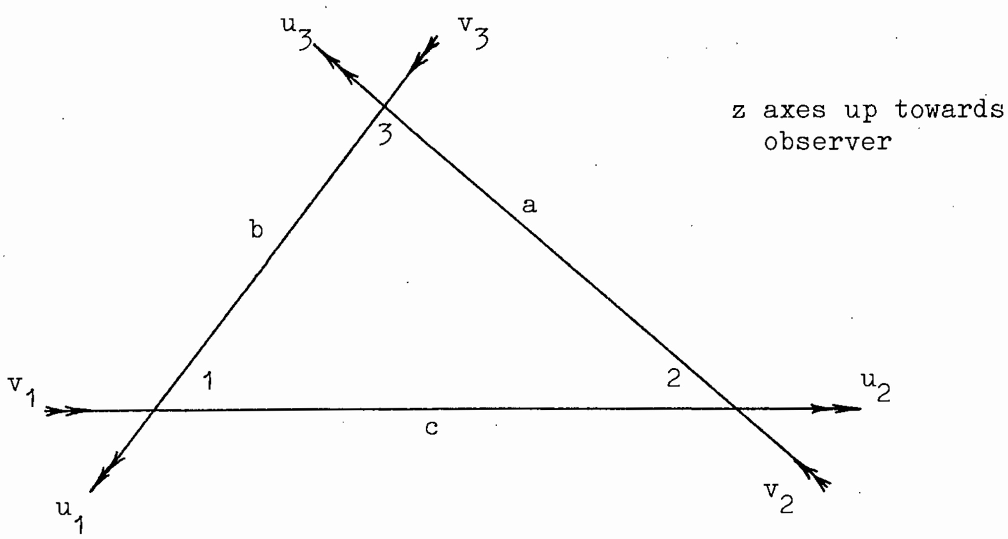


Figure 10: Oblique axes for a general triangle

Axes about which nodal rotations take place are in the directions of the sides of the elements if the nodes are taken in cyclic order, as shown in figure 10. Translations take place in a z direction. This is perpendicular to the plane of the element and forms a right hand set with the two oblique rotational axes taken in the order u, v.

Define a nodal displacement vector for this element as:

$$\{d\}^t = \{w_1, \theta_{u1}, \theta_{v1}, w_2, \theta_{u2}, \theta_{v2}, w_3, \theta_{u3}, \theta_{v3}\}$$

where w_i indicates a translation in the z direction at node i,

θ_{ui} indicates a right hand rotation about the ui axis.

By inspection, a rigid body parallel translation of 1 unit would take the form:

$$\{d\}^t = \{1, 0, 0, 1, 0, 0, 1, 0, 0, \} \tag{60a}$$

The most convenient rotations about non-parallel axes in the plane are rotations about the sides of the element proportional to twice the length of the side.

Consider a rotation of magnitude '2a' about the side of length 'a'.

Note that if 't' is the distance of node 1 from the axis of rotation then the area of the triangle given in equation (59) can be represented as:

$$\text{Area} = \alpha\beta\gamma\Delta/4 = ta/2$$

hence: $t = \alpha\beta\gamma\Delta/2a$

and the translation of node 1 when the rotation takes place is:

$$w_1 = 2at = \alpha\beta\gamma\Delta.$$

The nodal rotations of node 1 are found from the diagram of components which is similar to the shape of the element. (See figure 11).

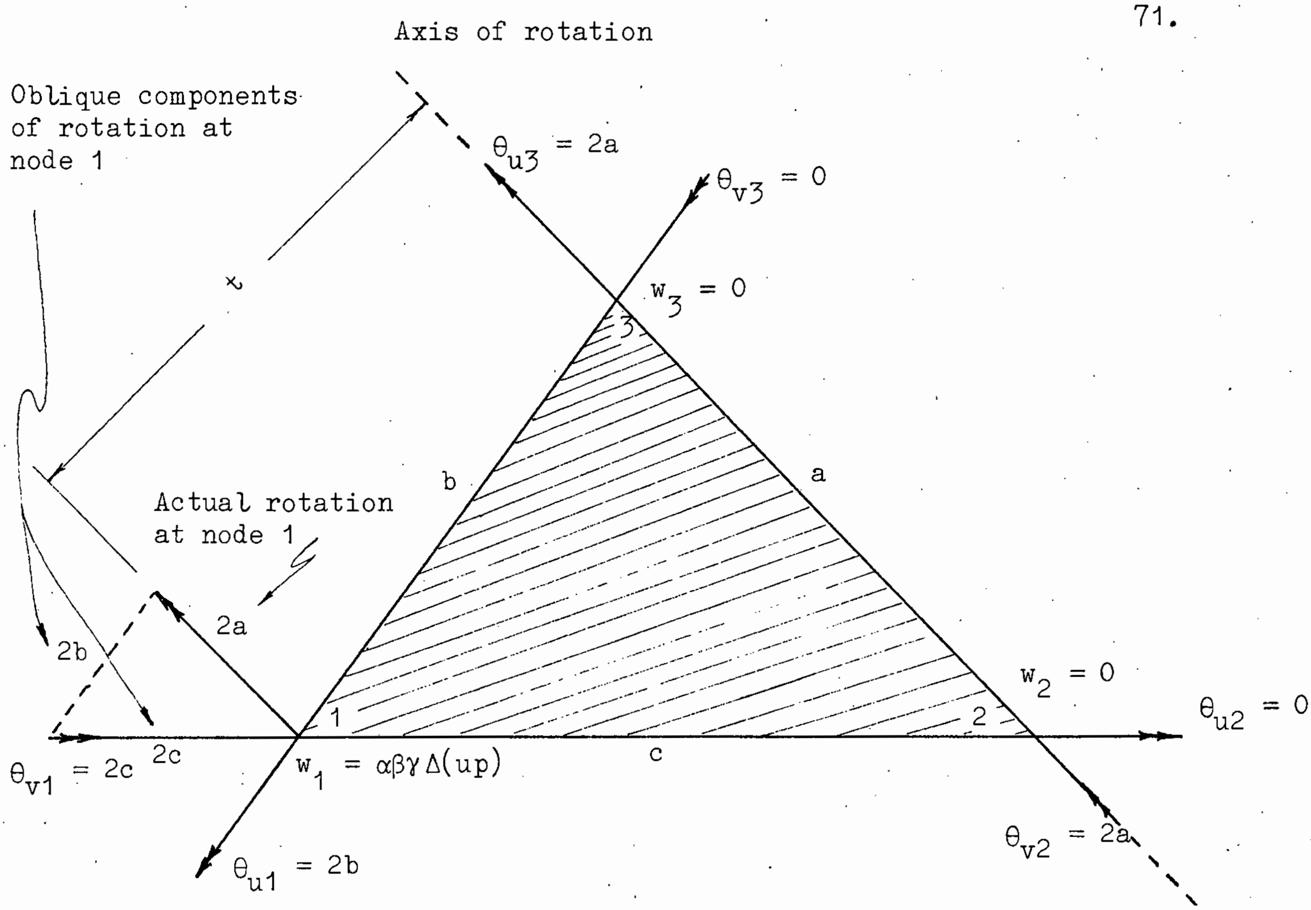


Figure 11: The nodal displacements in oblique coordinates for a triangle rigidly rotated an amount 2a.

Rotations about the other two sides produce similar results. The rotational rigid body movement vectors may be written as:

$$\{d\}^t = \{\alpha\beta\gamma\Delta, -2b, -2c, 0, 0, 2a, 0, 2a, 0\} \qquad (60b)$$

$$\{d\}^t = \{0, 2b, 0, \alpha\beta\gamma\Delta, -2c, -2a, 0, 0, 2b\} \qquad (60c)$$

$$\{d\}^t = \{0, 0, 2c, 0, 2c, 0, \alpha\beta\gamma\Delta, -2a, -2b\} \qquad (60d)$$

It can be seen that one of these vectors is redundant because if all three were applied, their sum is simply a multiple of the parallel translation vector (60a).

3.3.4. Transformation from Orthogonal to Oblique Coordinates

The matrix for a transformation from coordinate representation of the position of a point in terms of an oblique set of axes to representation in terms of an orthogonal set of axes can be derived from figure 12.

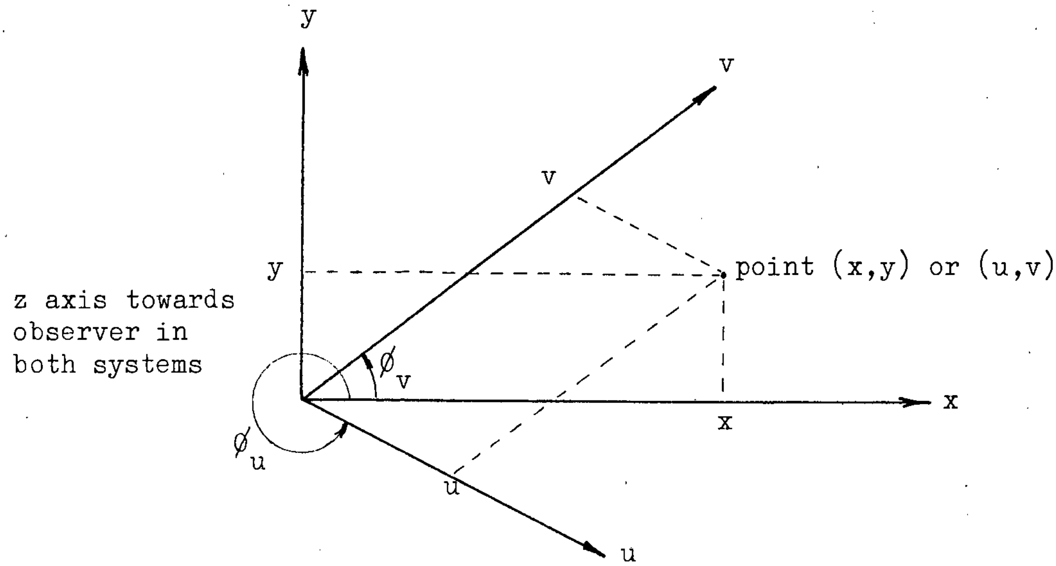


Figure 12: The coordinates of a general point in terms of one oblique and one orthogonal set of axes.

By inspection:

$$\begin{bmatrix} z \\ x \\ y \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi_u & \cos\phi_v \\ 0 & \sin\phi_u & \sin\phi_v \end{bmatrix} \times \begin{bmatrix} z \\ u \\ v \end{bmatrix} \tag{61}$$

or $\{X\} = [T]\{U\}$

The matrix $[T]$ is only an orthogonal matrix when:

$$\phi_u - \phi_v = \pm 3\pi/2 \quad \text{or} \pm \pi/2$$

In the general case:

$$[T]^{-1} \neq [T]^t$$

To find the matrix $[T]^{-1}$, divide the adjoint matrix of $[T]$ by its determinant:

$$\begin{bmatrix} z \\ u \\ v \end{bmatrix} = \frac{1}{\sin\phi_v \cos\phi_u - \sin\phi_u \cos\phi_v} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & \sin\phi_v & -\cos\phi_v \\ 0 & -\sin\phi_u & \cos\phi_u \end{bmatrix} \times \begin{bmatrix} z \\ x \\ y \end{bmatrix}$$

or $\{U\} = [T]^{-1} \{X\}$ (62)

If: $\phi_u = \hat{A}/2$; and $\phi_v = \pi - \hat{A}/2$; then from equations (56), (61) and (62):

$$[T_A] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{2}\,\alpha\Delta/\sqrt{bc} & -\frac{1}{2}\,\alpha\Delta/\sqrt{bc} \\ 0 & \frac{1}{2}\,\beta\gamma/\sqrt{bc} & \frac{1}{2}\,\beta\gamma/\sqrt{bc} \end{bmatrix} \tag{63}$$

and

$$[T_A]^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \sqrt{bc}/\alpha\Delta & \sqrt{bc}/\beta\gamma \\ 0 & -\sqrt{bc}/\alpha\Delta & \sqrt{bc}/\beta\gamma \end{bmatrix} \tag{64}$$

Similarly $[T_B]$, $[T_C]$, $[T_B]^{-1}$ and $[T_C]^{-1}$ can be determined.

3.3.5. A note on the Betti-Maxwell (Rayleigh) reciprocal theorem

The following is a statement of the theorem (Socolnikoff[8]):

If an elastic body is subjected to two systems of body and surface forces, then the work that would be done by the first system in acting through the displacements due to the second system of forces is equal to the work that would be done by the second system in acting through the displacements due to the first system of forces.

The load required to cause a unit displacement of its point of application on a structure or structural element in its own direction of action while all other displacements are prevented is the direct stiffness corresponding to the displaced nodal movement mode.

A load required to prevent the displacement of a node while another node is displaced a unit amount (in any one of its possible modes of movement) is an indirect stiffness relationship.

Consider a structure which is fixed so that it only has two degrees of nodal freedom. Loads are applied corresponding to these two degrees of freedom so that one load causes a unit displacement while the other prevents any movement at its point of application. The loads are then removed and a second pair of loads is applied causing the other point only to displace a unit amount. The only displacement of the first load case is a unit amount corresponding to the load which contained this displacement in the second load case. The same applies to the second load case with respect to the first.

Applying the reciprocal theorem it can be seen that the two loads which contained a degree of freedom in the two cases are equal to one another. The indirect stiffnesses describing the structural behaviour at each degree of freedom with respect to the other are therefore equal to one another.

If a matrix is written with direct stiffnesses on the leading diagonal and indirect stiffnesses in appropriate positions off the diagonal, the stiffness matrix will be symmetric.

This only applies if it is possible to displace a node in one mode at a time without causing any other displacements. If all the degrees of freedom at a single point are described in the directions of mutually perpendicular axes, this is possible. If the degrees of freedom at a point are described in the directions of oblique (non-orthogonal) axes, this is not possible and a corresponding stiffness matrix is not symmetrical.

The stiffness matrix in terms of one set of axes can be transformed to another as follows:

If the degrees of freedom of some structure are described in terms of one set of axes, let $\{d_1\}$, $\{f_1\}$ and $[K_1]$ be the displacement vector, the load vector and the stiffness matrix of the structure. Let $\{d_2\}$, $\{f_2\}$ and $[K_2]$ be the corresponding vectors and matrix of the same structure described in terms of a second set of axes.

Define the transformation matrix $[T]$ from:

$$\{d_2\} = [T].\{d_1\} \tag{65}$$

then:

$$\{f_2\} = [T] \cdot \{f_1\}$$

$$\{d_1\} = [T]^{-1} \cdot \{d_2\} \qquad (66)$$

$$\{f_1\} = [T]^{-1} \cdot \{f_2\} \qquad (67)$$

From the fundamental equation of the stiffness method (53):

$$[K_1]\ \{d_1\} = \{f_1\} \qquad (68)$$

$$[K_2]\ \{d_2\} = \{f_2\} \qquad (69)$$

Substituting from equations (66) and (67) into equation (68):

$$[K_1]\ [T]^{-1}\{d_2\} = [T]^{-1}\{f_2\}$$

Premultiplying this by $[T]$:

$$[T]\ [K_1]\ [T]^{-1}\{d_2\} = \{f_2\}$$

Comparing this to equation (69):

$$[K_2] = [T]\ [K_1]\ [T]^{-1} \qquad (70)$$

Now if both axis systems were orthogonal at each node, the matrix $[T]$ would represent an orthogonal transformation matrix and

$$[T]^{-1} = [T]^t$$

In this case, the operation of the matrix $[T]$ in equation (70) upon a symmetric matrix $[K_1]$ produces a symmetric matrix $[K_2]$ as expected.

If only the first axis system was orthogonal at each node and the second was oblique at each point, then the matrix $[T]$ is not orthogonal. The operation of the matrix $[T]$ in equation (70) upon a symmetric matrix $[K_1]$ then produces a non-symmetric matrix $[K_2]$ as described in this section.

3.3.6 A standard example in plate bending

A thin, elastic, rectangular plate subjected to a unit moment per unit length parallel to one side is shown in figure 13.

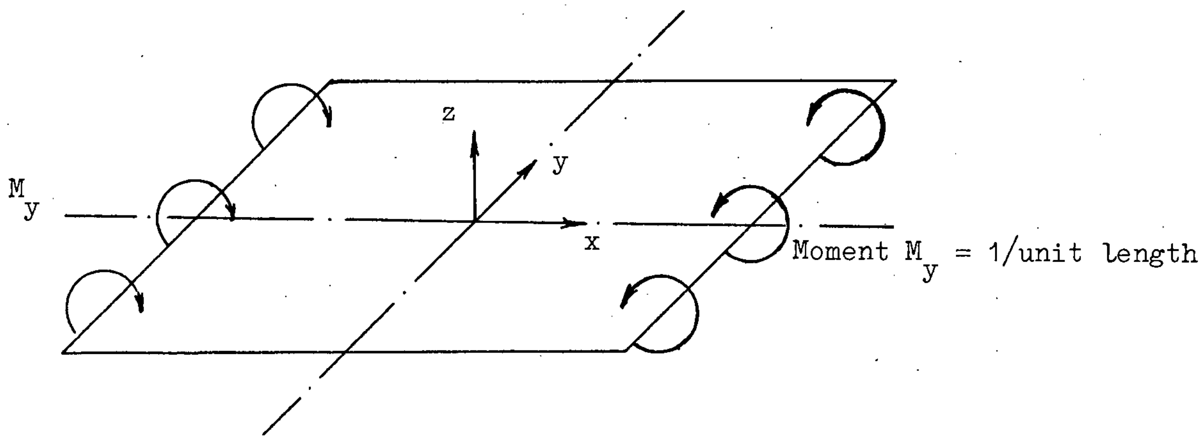


Figure 13: A rectangular plate subjected to a uniform moment along two opposite edges.

The deflection w of this plate in the z direction, relative to the deflection of the origin (which is chosen to be at the centre of the plate) is (Hearmon and Adams[9]):

$$w = \frac{-6 M_y}{Eh^3}(x^2 - \nu y^2) \tag{71}$$

where E is Young's modulus of elasticity,
h is the material thickness,
ν is Poisson's ratio for the material,
M_y is defined in figure 13.

The slopes of any general point x,y may be found by differentiation of w with respect to x and y (c.f. equations (12)).

As the figure is symmetrical about the x and y axes, the part of the structure in the positive quadrant may be used to represent the whole structure.

The deflections of all points on this structure are known. The plate can be treated as an assembly of triangular bending elements. It will thus be used as a standard structure for finding the numerical values of stiffness coefficients (as described in section 3.2).

Other structures (such as a square plate with point supports at three corners and a point load at the fourth) could be used. The advantages of the structure chosen are:

(i) In the rectangular plate the value of Poisson's ratio (ν) affects the interrelationship between displacements, not only the magnitudes (as in the case in the square plate mentioned above).

(ii) The rectangular plate can be chosen to have any number of degrees of nodal freedom by allowing a determined number to lie on axes of symmetry.

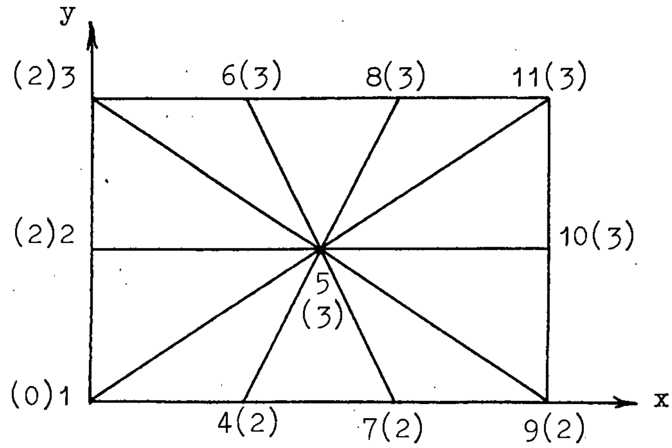
Suppose for example the plate was to be divided into elements so that it has twenty five degrees of nodal freedom. Only one quadrant of the structure will be considered. The node at the origin has no degrees of freedom. Nodes on the two axes (being lines of symmetry) only have two degrees of freedom. All other nodes on the structure have three degrees of freedom. The twenty five degrees of freedom could be as follows:-

(i)	one node at the origin	0	degrees	of	freedom
	one internal node	3	"	"	"
	eleven nodes on axes	22	"	"	"
	thirteen nodes total	25	"	"	"
(ii)	one node at the origin	0	"	"	"
	three internal nodes	9	"	"	"
	eight nodes on axes	16	"	"	"
	twelve nodes total	25	"	"	"
(iii)	one node at the origin	0	"	"	"
	five internal nodes	15	"	"	"
	five nodes on axes	10	"	"	"
	eleven nodes total	25	"	"	"

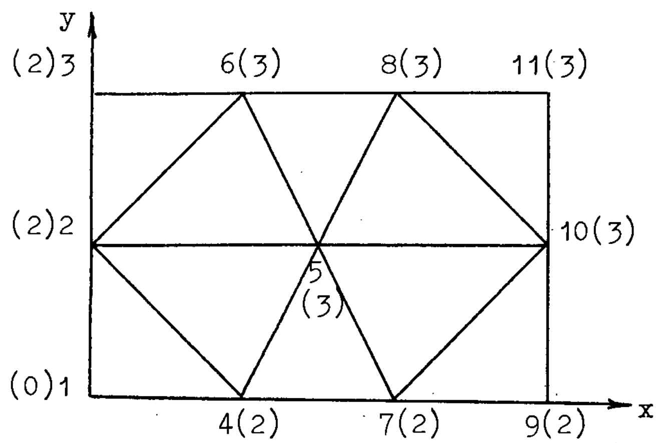
etc.

Drawing these node situations up into a reasonable configuration, one finds that the arrangement (iii) above is the most convenient.

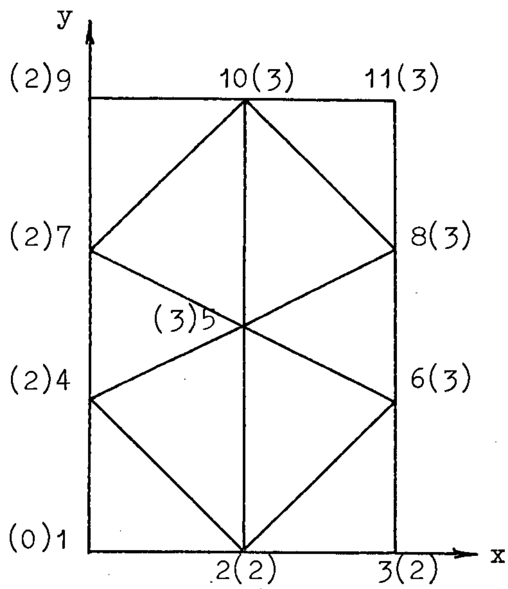
Some possible element configurations which may result from this arrangement are shown in figure 14.



Mark I Configuration



Mark II Configuration



Mark III Configuration

Degrees of freedom of each node are given in parentheses.

Figure 14: Some possible element configurations for a rectangular plate with twenty five degrees of nodal freedom.

As was mentioned in section 3.2, it is important that the shapes of elements should be as different as possible. It will be found that if two elements are similar, the resulting system of simultaneous equations is badly conditioned.

A ratio to be known as the "non-similarity ratio" was used to distinguish between elements with a triangular shape. Unlike the aspect ratio for rectangules if two triangles have the same non-similarity ratio they are not necessarily similar. If they have different non-similarity ratios, then they are definitely dissimilar. If the lengths of the sides of a triangle are a, b and c and a is the longest side, then the non-similarity ratio is defined as:

$$\text{N.S.R.} = b/a \times c/a$$

This is easily programmed as:

$$abc/(\text{longest side})^3.$$

It was found by trial that the greater the spread of non-similarity ratios in an element conformation, the more stable the solution became. The range of non-similarity ratios of triangles is:

$$0 < \text{N.S.R.} \leqslant 1.$$

3.3.7. Solution of identities and formation of functions

If an equation is to hold identically for all values of the variables occurring in it, then the coefficients of all like terms can be summed and equated to zero. The set of equations formed in this manner holds simultaneously and can be reduced to its simplest form or solved by elimination.

e.g., if:

$$C_1x + C_2y + C_2z + C_3x \equiv 0$$

for all values of x,y and z, then:

$$C_1 + C_3 = 0 \quad \text{and} \quad C_2 = 0.$$

There are not necessarily enough linearly independent equations to solve all the values of the coefficients uniquely. Some equations may also exist which are linearly dependent upon other equations. A direct elimination method when used results in as many coefficients as possible being expressed as linear combinations of the remaining ones. The Gauss Jordan method (given in Appendix 1) is one of the most convenient reduction techniques.

In this thesis the theory of identities is used to reduce the number of independent unknown coefficients representing the stiffness matrix of a triangular plate bending element, to a minimum.

As will be shown in a later section (3.4.3), elements of the stiffness matrix are chosen to be polynomials in the variables a, b and c, the lengths of the three sides of the triangle and α, β, γ and Δ which are defined (again) by equations (72). The terms of these polynomial expressions must all have units of length raised to the same power.

It will be noted that as:

$$\begin{aligned} \alpha^2 &= -a + b + c \\ \beta^2 &= a - b + c \\ \gamma^2 &= a + b - c \\ \Delta^2 &= a + b + c \end{aligned} \qquad (72)$$

(i) the powers of the variables α, β. γ and Δ must be twice the powers of the variables a, b and c in order to keep consistent units of length in a function involving all seven variables.

(ii) In order to identify like terms in a polynomial identity expression, powers of α, β, γ and Δ greater than unity must be substituted for from equation (72).

For example the full "homogeneous" polynomial identity of root length degree two (see definitions):

$$C_1\alpha^2 + C_2\alpha\beta + C_3\beta^2 + C_4\beta\gamma + C_5\gamma^2 + C_6\alpha\gamma + C_7\Delta^2 + C_8\alpha\Delta + C_9\beta\Delta + C_{10}\gamma\Delta + C_{11}a + C_{12}b + C_{13}c \equiv 0$$

must be expressed as:

$$C_2\alpha\beta + C_4\beta\gamma + C_6\alpha\gamma + C_8\alpha\Delta + C_9\beta\Delta + C_{10}\gamma\Delta + D_1a + D_2b + D_3c \equiv 0$$

where
$$D_1 = C_{11} - C_1 + C_3 + C_5 + C_7$$
$$D_2 = C_{12} + C_1 - C_3 + C_5 + C_7$$
$$D_3 = C_{13} + C_1 + C_3 - C_5 + C_7$$

before like terms can be collected. (The result of a collection before this is done is clearly different from the correct result.)

When polynomials of any root length degree are generated, only first degree powers of α, β, γ and Δ need be included to form a full "homogeneous" function. For reasons which appear later it is necessary to be able to generate any cyclic or mirror variation (see definitions) of a function number n, i.e. given the variation number m one should be able to generate $f_n(m)$ where:

$$\begin{aligned} f_n(1) &= f_n(a, b, c) \\ f_n(2) &= f_n(a, c, b) \\ f_n(3) &= f_n(b, c, a) \\ f_n(4) &= f_n(b, a, c) \\ f_n(5) &= f_n(c, a, b) \\ f_n(6) &= f_n(c, b, a) \end{aligned} \qquad (73)$$

Full "homogeneous" polynomials of any root length degree and any given variation are generated automatically by the FORTRAN program described in Appendix 6.

Some examples of full homogeneous polynomials of root length degree one are:

$$\begin{aligned} f_3(1) &= C_1\,\alpha + C_2\,\beta + C_3\,\gamma + C_4\,\Delta \\ f_3(2) &= C_1\,\alpha + C_2\,\gamma + C_3\,\beta + C_4\,\Delta \\ f_3(3) &= C_1\,\beta + C_2\,\gamma + C_3\,\alpha + C_4\,\Delta. \end{aligned}$$

3.4 Deriving the Explicit Stiffness Matrix

3.4.1. The choice of axes and dimensions

In order to describe a finite element, three types of feature must be considered:

(i) The material properties of the element.

(ii) The geometric dimensions of the element giving its physical shape.

(iii) Some set of local coordinate axes linking the directions of the nodal movements to the physical shape of the element.

In elastic plate bending analysis, the parameters; E, the Young's modulus of elasticity; ν, the Poisson's ratio and h the thickness of the material are always used.

On the other hand, various possibilities exist for describing the shape of a triangular element and the orientation of local axes. Some of these are:

(i) A simple orthogonal axis system with the coordinates of each node given.

(ii) A single axis system with one node lying at the origin and one axis lying along a side of the element. The positions of nodes are then given:

   (a) as shown before in figure 2, i.e. the length of the base and the height and position of the apex;

   (b) as the length of the three sides.

(iii) The inclination of the three sides to a fixed single axis system and either:

   (a) the length of a side,

   (b) the radius of the inscribed or circumscribed circle,

   (c) the area of the element.

The actual method which has been chosen to describe the triangle is preferred because:

(a) The six parameters of description (i) (namely the two co-ordinates at each of three nodes), compared with the three of description (ii), mean that full homogeneous polynomials in all the parameters require many more unknown coefficients. The disadvantage of this will become obvious.

(b) In description (ii), each node has a different relationship to the axis system. This means that savings due to repetition are lost.

(c) Description (iii) has the disadvantage that the parameters are in different units. As angles are dimensionless parameters the assumed polynomials formed from them are not necessarily homogeneous and their degree cannot be predicted from the units of the stiffness matrix elements.

Based on these objections, the following two methods of description were used.

The lengths of the three sides (a, b and c) give the shape of the triangle in both cases. The two sets of axes are as follows:-

(i) A local x axis radiates out from the centre of the inscribed circle at each node. The z axis points up perpendicular to the plane of the plate. A local y axis forms an orthogonal right hand set with these axes at each node. (See figure 15.)

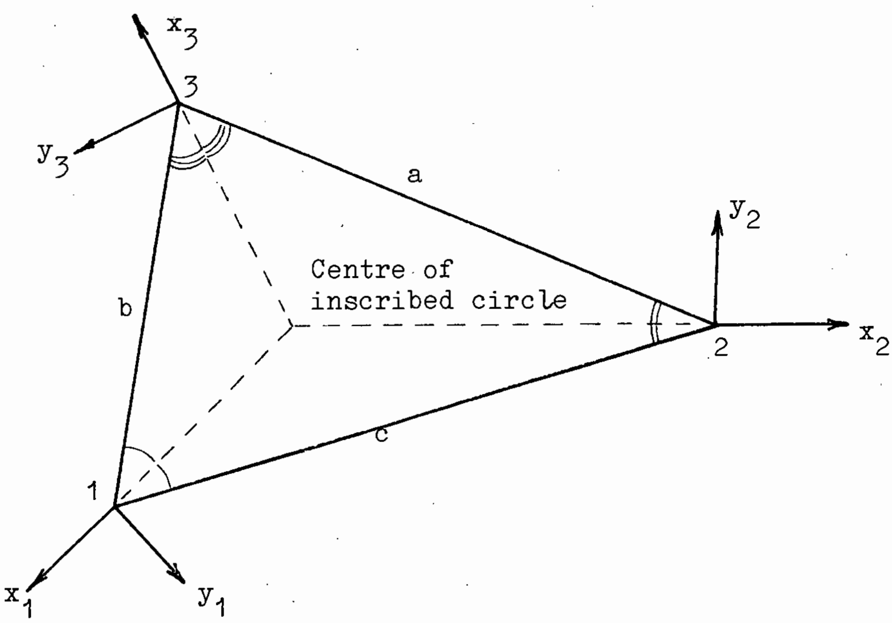


Figure 15: The chosen set of local orthogonal axes.

(ii) Axes are in the directions of the sides at each node as shown in figure 11.

The first method of description has the advantage that the resulting stiffness matrix is symmetrical according to the Betti-Maxwell reciprocal theorem. (See subsection 3.3.5.)

The second method of description has the advantage that rigid body movements of the triangle are represented by comparatively simple displacement vectors. (See subsection 3.3.3).

The reason for choosing the centre of the inscribed circle (and not, say, the centroid, circumscribed circle or previous node in order) for the point from which the x axis radiates in system (i) above can be seen from section 3.3.2: the matrices for transformation to other axis systems are comprised of relatively simple expressions.

3.4.2. <u>The stiffness matrix in general terms</u>

It would be convenient if every element of the stiffness matrix were in the same dimensions. All elements could then be represented by a homogeneous polynomial of the same root length degree. This can easily be arranged by multiplying various elements of the displacement vector and load vector by chosen values as follows:-

$$\{d_1\}^t = \{w_1, \sqrt{bc}\ \theta_{x1}, \sqrt{bc}\ \theta_{y1}, w_2, \sqrt{ca}\ \theta_{x2}, \sqrt{ca}\ \theta_{y2}, w_3, \sqrt{ab}\ \theta_{x3}, \sqrt{ab}\ \theta_{y3}\}$$

$$\{f_1\}^t = \{F_1, M_{x1}/\sqrt{bc}, M_{y1}/\sqrt{bc}, F_2, M_{x2}/\sqrt{ca}, M_{y2}/\sqrt{ca}, F_3, M_{x3}/\sqrt{ab}, M_{y3}/\sqrt{ab}\} \qquad (74)$$

where: w represents a translation in the z direction,

θ represents a rotation in a right hand sense about the axis indicated as a suffix,

F represents a force in the z direction,

M represents a couple acting in a right hand sense about the axis indicated as a suffix.

A systematic nodal description would still be maintained if the factors $\sqrt{bc}$, $\sqrt{ca}$ and $\sqrt{ab}$ were replaced by a, b and c, say. It will be found though (in a later section) that the factors chosen can be cancelled with other factors which develop in the stiffness matrix.

The units of vector $\{d\}$ are length and the units of vector $\{f\}$ are force. The units of Young's modulus, E, are force per length squared. The units of the thickness, h, are length. Supposing a homogeneous function, P, is cyclic in a, b and c and has units of length to a power, p, (e.g. $P = abc + a^3 + b^3 + c^3$ and $p = 3$).

If the stiffness matrix:

$$[K] = \frac{Eh^3}{P} [K_1]$$

Then from the units of the other vectors and the fundamental equation of the stiffness method (53), the units of elements of matrix $[K_1]$ are length to the power

$$p - 2$$

The homogeneous polynomials which form the elements of matrix $[K_1]$ are of root length degree:

$$2p - 4 \qquad (75)$$

Supposing the number n of a function f_n indicates one of twelve chosen homogeneous polynomials of the above degree as follows:

Function name:	A	B	C	D	E	F	G	H	J	K	L	M
n:	1	2	3	4	5	6	7	8	9	10	11	12

Then if the cyclic or mirror variation of a function is represented as $f_n(m)$ (according to equations (73)), the matrix $[K_1]$ shown in (76) is derived as described below.

$[K_1] =$

	1			2			3			
	1	2	3	4	5	6	7	8	9	
A(1)= A(2)	C(1)=-C(2)	D(1)= D(2)	B(1)= B(4)	E(1)	F(1)	B(2)= B(5)	-E(2)	F(2)	1	
C(1)=-C(2)	G(1)= G(2)	J(1)=-J(2)	-E(4)	K(1)= K(4)	M(1)	E(5)	K(2)= K(5)	-M(2)	2	1
D(1)= D(2)	J(1)=-J(2)	H(1)= H(2)	F(4)	-M(4)	L(1)= L(4)	F(5)	M(5)	L(2)= L(5)	3	
B(4)= B(1)	-E(4)	F(4)	A(3)= A(4)	C(3)=-C(4)	D(3)= D(4)	B(3)= B(6)	E(3)	F(3)	4	
E(1)	K(4)= K(1)	-M(4)	C(3)=-C(4)	G(3)= G(4)	J(3)=-J(4)	-E(6)	K(3)= K(6)	M(3)	5	2
F(1)	M(1)	L(4)= L(1)	D(3)= D(4)	J(3)=-J(4)	H(3)= H(4)	F(6)	-M(6)	L(3)= L(6)	6	
B(5)= B(2)	E(5)	F(5)	B(6)= B(3)	-E(6)	F(6)	A(5)= A(6)	C(5)=-C(6)	D(5)= D(6)	7	
-E(2)	K(5)= K(2)	M(5)	E(3)	K(6)= K(3)	-M(6)	C(5)=-C(6)	G(5)= G(6)	J(5)=-J(6)	8	3
F(2)	-M(2)	L(5)= L(2)	F(3)	M(3)	L(6)= L(3)	D(5)= D(6)	J(5)=-J(6)	H(5)= H(6)	9	

All identities hold for all values of a, b and c.

This matrix was derived in the following way.

Eighty one different full functions could have been chosen to represent this matrix in general terms. If the functions have root length degree three, then each function would contain sixteen unknown coefficients, e.g.

$$\begin{aligned} f_1(1) \;=\; & A_1 a\alpha + A_2 a\beta + A_3 a\gamma + A_4 a\Delta + A_5 b\alpha + A_6 b\beta + A_7 b\gamma + A_8 b\Delta \\ & + A_9 c\alpha + A_{10} c\beta + A_{11} c\gamma + A_{12} c\Delta + A_{13}\beta\gamma\Delta + A_{14}\alpha\gamma\Delta + A_{15}\alpha\beta\Delta \\ & + A_{16}\alpha\beta\gamma \end{aligned}$$

In this case the matrix would depend on $1296 (= 16 \times 81)$ different unknown coefficients.

Certain properties which the element has – cf. the Betti-Maxwell theorem (section 3.3.5) and the choice of axes and dimensions (3.3.6) – allow this number to be reduced considerably.

Because the axis system chosen is orthogonal at each point, the matrix is symmetrical. Each element (i, j) is equal to element (j, i) for all values of a, b and c, (i.e. for any shape of element).

Consider the cyclic nature of the dimensions and the similarity of the description of the local axis systems at each node in figure 15.

It can be seen that if a property of node one is described by function:

$$f_n(1) = f_n(a, b, c)$$

then the same property would be described by the cyclic variations (see definitions):

$$f_n(3) = f_n(b, c, a)$$

and

$$f_n(5) = f_n(c, a, b)$$

at nodes two and three respectively.

It can also be seen that if a property of node one with respect to node two is represented by a function:

$$f_n(1) = f_n(a, b, c)$$

then the same property of node two with respect to node three is represented by the cyclic variation:

$$f_n(3) = f_n(b, c, a)$$

etc. and the same property of node one with respect to node three is represented by the mirror variation:

$$f_n(2) = f_n(a, c, b)$$

etc. with a possible reversal of sign where the axis y is involved.

The application of these principles in all possible positions reduces the number of different functions to be chosen to twelve as shown in matrix (76). If polynomials of root length degree three are chosen, now, there will be 192(= 12 × 16) different unknown coefficients.

It is possible to use the upper half of submatrix (1, 1) of matrix (76) and the whole of submatrix (1, 2) in general terms to store a description of the whole matrix (See figure 16).

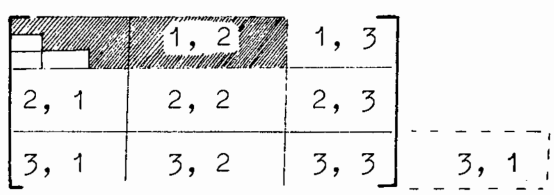


Figure 16: Essential Submatrices for Description of the whole Matrix.

The remainder of the matrix can be found from cyclic variations of the submatrices and from the Betti-Maxwell symmetry property. (Note that submatrix (3, 1) is the second cyclic variation of submatrix (1,2)).

3.4.3. The application of rigid body movements

The stiffness matrix developed in the previous section must now be transformed (according to equation (70)) to a representation in terms of oblique axes (figure 10) before the rigid body movement vectors can be applied. The matrix [T] of equation (70) is represented (from equation (63)) as:

$$[T] = \begin{bmatrix} [T_A] & [0] & [0] \\ [0] & [T_B] & [0] \\ [0] & [0] & [T_C] \end{bmatrix}$$

The matrix $[T]^{-1}$ of equation (70) is represented (from equation (64)) as:

$$[T]^{-1} = \begin{bmatrix} [T_A]^{-1} & [0] & [0] \\ [0] & [T_B]^{-1} & [0] \\ [0] & [0] & [T_C]^{-1} \end{bmatrix}$$

If the displacement and load vectors of the stiffness equation (53) are represented as:

$$\{d_2\}^t = \{w_1,\ \theta_{u1},\ \theta_{v1},\ w_2,\ \theta_{u2},\ \theta_{v2},\ w_3,\ \theta_{u3},\ \theta_{v3}\}$$

$$\{f_2\}^t = \{F_1,\ M_{u1}/bc,\ M_{v1}/bc,\ F_2,\ M_{u2}/ca,\ M_{v2}/ca,\ F_3,\ M_{u3}/ab,\ M_{v3}/ab\}$$

then matrix $[K_2]$ is represented in (77).

The matrix $[K_2]$ is defined by :

$$\frac{Eh^3}{P}\,[K_2]\,\{d_2\} = \{f_2\}$$

where P was defined before to be a cyclic factor in a, b and c.

It will be found that as each row of submatrices of matrix $[K_2]$ is a cyclic variation of the first row, and as the rigid body displacement vectors are cyclic variations of one another, no new equations are formed by applying rigid body displacement vectors to the second and third rows. They are thus not shown in the matrix (77).

SUBMATRIX (1, 1)

$$
[K_2] = \left[\begin{array}{c|c|c}
A(1) & C(1)\alpha\Delta + D(1)\beta\gamma & -C(1)\alpha\Delta + D(1)\beta\gamma \\
\hline
C(1)/\alpha\Delta + D(1)/\beta\gamma & \begin{array}{l} G(1) + H(1) \\ + J(1)(-\beta\gamma/\alpha\Delta + \alpha\Delta/\beta\gamma) \end{array} & \begin{array}{l} G(1) + H(1) \\ + J(1)(\beta\gamma/\alpha\Delta + \alpha\Delta/\beta\gamma) \end{array} \\
\hline
-C(1)/\alpha\Delta + D(1)/\beta\gamma & \begin{array}{l} -G(1) + H(1) \\ + J(1)(-\beta\gamma/\alpha\Delta + \alpha\Delta/\beta\gamma) \end{array} & \begin{array}{l} G(1) + H(1) \\ - J(1)(\beta\gamma/\alpha\Delta + \alpha\Delta/\beta\gamma) \end{array}
\end{array} \right.
$$

SUBMATRIX (1, 2)

$$
\begin{array}{c|c|c}
B(1) & E(1)\beta\Delta + F(1)\alpha\gamma & -E(1)\beta\Delta + F(1)\alpha\gamma \\
\hline
-E(4)/\alpha\Delta + F(4)/\beta\gamma & \begin{array}{l} K(1)\beta/\alpha + M(1)\gamma/\Delta \\ - M(4)\Delta/\gamma + L(1)\alpha/\beta \end{array} & \begin{array}{l} -K(1)\beta/\alpha + M(1)\gamma/\Delta \\ + M(4)\Delta/\gamma + L(1)\alpha/\beta \end{array} \\
\hline
E(4)/\alpha\Delta + F(4)/\beta\gamma & \begin{array}{l} -K(1)\beta/\alpha - M(1)\gamma/\Delta \\ - M(4)\Delta/\gamma + L(1)\alpha/\beta \end{array} & \begin{array}{l} K(1)\beta/\alpha - M(1)\gamma/\Delta \\ + M(4)\Delta/\gamma + L(1)\alpha/\beta \end{array}
\end{array}
$$

SUBMATRIX (1,3)

$$
\left. \begin{array}{c|c|c}
B(2) & -E(2)\gamma\Delta + F(2)\alpha\beta & E(2)\gamma\Delta + F(2)\alpha\beta \\
\hline
E(5)/\alpha\Delta + F(5)/\beta\gamma & \begin{array}{l} K(2)\gamma/\alpha - M(2)\beta/\Delta \\ + M(5)\Delta/\beta + L(2)\alpha/\gamma \end{array} & \begin{array}{l} -K(2)\gamma/\alpha - M(2)\beta/\Delta \\ - M(5)\Delta/\beta + L(2)\alpha/\gamma \end{array} \\
\hline
-E(5)/\alpha\Delta + F(5)/\beta\gamma & \begin{array}{l} -K(2)\gamma/\alpha + M(2)\beta/\Delta \\ + M(5)\Delta/\beta + L(2)\alpha/\gamma \end{array} & \begin{array}{l} K(2)\gamma/\alpha + M(2)\beta/\Delta \\ - M(5)\Delta/\beta + L(2)\alpha/\gamma \end{array}
\end{array} \right] \quad (77)
$$

It will be noticed that the factors $\sqrt{ab}$, $\sqrt{bc}$ and $\sqrt{ca}$ which appeared in the transformation matrix [T] now only occur in the load vector as ab, bc and ca. This saves including the parameters $\sqrt{ab}$, $\sqrt{bc}$ and $\sqrt{ca}$ in the assumed polynomials. (This was the reason for the choice of load and displacement vectors in (74)).

The rigid body movement displacement vectors can now be applied to matrix $[K_2]$.

The rigid body parallel translation vector (60a) is multiplied by the top rows of the submatrices (77) only. This is because the rotational displacements at node one are zero during this rigid body movement. The rotational loading need not therefore, necessarily, be zero to give zero external work. The resulting identity is given in (78a).

The rigid rotation about the side of length 'a' (60b) is multiplied by each of the three rows in turn. This is because a movement occurs corresponding to all three degrees of freedom at node one when this vector is applied. All the applied loads at this node must therefore be zero to do the required zero external work on the element. The resulting identities are (78b) to (78d).

The rotation about the side of length 'b' (60c) is applied to the second row of the submatrices only. This is because node one rotates corresponding to the rotational degree of freedom about axis u1. The resulting identity is given as (78e).

These identities are to hold simultaneously for any shape of element. This means that the set of equations formed from each identity holds simultaneously with the set of equations formed from the other identities. The identities, which hold for all values of a, b and c are multiplied by their lowest common denominators, and squared terms in α, β, γ and Δ are eliminated by substitution from (72).

$$A(1) + B(1) + B(2) \equiv 0 \qquad (78a)$$

$$A(1)\alpha\beta\gamma\Delta - 2C(1)b\alpha\Delta - 2D(1)b\beta\gamma + 2C(1)c\alpha\Delta - 2D(1)c\beta\gamma - 2E(1)a\beta\Delta$$
$$+ 2F(1)a\alpha\gamma - 2E(2)a\gamma\Delta + 2F(2)a\alpha\beta \equiv 0 \qquad (78b)$$

The identity (78c) adds no new equations to the set and is omitted.

$$C(1)(-a^2 + b^2 - 2bc + c^2)\alpha\Delta + D(1)(-a^2 + b^2 + 2bc + c^2)\beta\gamma + 2G(1)b\alpha\beta\gamma\Delta$$
$$- 2H(1)b\alpha\beta\gamma\Delta + 4J(1)(a^2b - b^3 + bc^2) - 2G(1)c\alpha\beta\gamma\Delta + 2K(1)(a^2 - ab + ac)\gamma\Delta$$
$$- 2M(1)(a^2 + ab - ac)\alpha\beta + 2M(4)(a^2 + ab + ac)\alpha\beta + 2L(1)(-a^2 + ab + ac)\gamma\Delta$$
$$- 2K(2)(a^2 + ab - ac)\beta\Delta + 2M(2)(a^2 - ab + ac)\alpha\gamma + 2M(5)(a^2 + ab + ac)\alpha\gamma$$
$$+ 2L(2)(-a^2 + ab + ac)\beta\Delta - 2H(1)c\alpha\beta\gamma\Delta \equiv 0 \qquad (78d)$$

$$2G(1)b\alpha\beta\gamma\Delta + 2H(1)b\alpha\beta\gamma\Delta + 8J(1)b^2c - E(4)(a^2 - b^2 + 2bc - c^2)\alpha\Delta$$
$$+ F(4)(-a^2 + b^2 + 2bc + c^2)\beta\gamma - 2K(1)(ac - bc + c^2)\gamma\Delta - 2M(1)(ac$$
$$+ bc - c^2)\alpha\beta + 2M(4)(ac + bc + c^2)\alpha\beta - 2L(1)(- ac + bc + c^2)\gamma\Delta$$
$$+ 2K(1)(a^2 - ab + ac)\gamma\Delta - 2M(1)(a^2 + ab - ac)\alpha\beta - 2M(4)(a^2 + ab$$
$$+ ac)\alpha\beta - 2L(1)(- a^2 + ab + ac)\gamma\Delta - 2K(2)(ab + b^2 - bc)\beta\Delta - 2M(2)(ab$$
$$- b^2 + bc)\alpha\gamma - 2M(5)(ab + b^2 + bc)\alpha\gamma + 2L(2)(- ab + b^2 + bc)\beta\Delta \equiv 0$$

(78e)

As these identities are only in terms of a, b, c, α, β, γ and Δ it seems reasonable that the functions of the matrix (76) will also contain only these variables. The degree of the functions is still not known. The data representing the identities (78) is punched for the FORTRAN identity solution routine described in Appendix 6. The value of the root length degree variable of the routine is increased in steps from 1. Certain of the unknown coefficients are then expressed in terms of the others. If the root length degree is too low, it will be found that all of the coefficients of certain elements of matrix $[K_1]$ (76) have been reduced to zero. This is clearly unacceptable and the root length degree must be increased.

Once a valid set of coefficients has been determined, the degree of the factor P can be determined from (75).

The minimum root length degree was found to be three. The degree of the factor P was found as:

$$2p - 4 = 3$$
$$p = 3{,}5$$

The cyclic factor:

$$P = abc\,\Delta \qquad (79)$$

was arbitrarily chosen.

When the root length degree of the polynomials of matrix $[K_1]$ is chosen to be three, it is found that all of the coefficients of the matrix can be expressed in terms of twenty five independent coefficients. The coefficients of the twelve functions of matrix $[K_1]$ are related to the twenty five independent coefficients in a matrix given in Appendix 7.

The form of a function is:

$$f_1(1) = A_1 a\alpha + A_2 a\beta + A_3 a\gamma + A_4 a\Delta + A_5 b\alpha + A_6 b\beta + A_7 b\gamma + A_8 b\Delta + A_9 c\alpha + A_{10} c\beta + A_{11} c\gamma + A_{12} c\Delta + A_{13}\beta\gamma\Delta + A_{14}\alpha\gamma\Delta + A_{15}\alpha\beta\Delta + A_{16}\alpha\beta\gamma.$$

3.4.4. Finding the values of coefficients

As was shown in the previous section, it is possible to express the whole explicit stiffness matrix for a triangular plate bending element in terms of twenty five unknown coefficients and the seven parameters a, b, c, α, β, γ and Δ. Once the values of the unknown coefficients have been found, it will be possible to use the explicit matrix to solve any plate bending problem.

The method used to find the values of these independent coefficients is to apply the stiffness matrix with its unknown coefficients to the standard plate described in section 3.3.6. The plate is arranged to have twenty five degrees of freedom. The coordinates of nodes are chosen so as to give the greatest variation (of the values of a, b, c, α, β, γ and Δ) between the elements of the plate.

As the loading, displacements and dimensions of the plate are known, each row of the fundamental stiffness relationship (53) is an equation in the unknown coefficients. As there are twenty five rows and twenty five unknown coefficients, provided that the rows of the matrix are not linear combinations of one another, a solution for the coefficients will be found. If the elements of the system are all dissimilar, the rows will not be linear combinations of one another.

A FORTRAN program given in Appendix 7 solves the values of the twenty five coefficients and substitutes them into the expressions representing <u>all</u> the coefficients of matrix $[K_1]$. The results of such a solution appear in a later section.

3.4.5. The effects of Poisson's ratio

The variable ν, Poisson's ratio, has not been included in the explicit stiffness matrix up to this stage. This variable should have been included as one of the parameters in the assumed functions of the matrix $[K_1]$ (76). This can be seen from the nature of the explicit stiffness matrices developed in Chapter II.

To have included the ratio, ν, as one of the variables would have caused difficulties. Because ν is a dimensionless ratio, there would be no limit to the power of ν allowable in an element function. If no power of ν above, say, the second was considered to be significant, then in the assumed functions, there would have to be three times as many terms, i.e. each of those considered with values of ν^o, ν^1 and ν^2. All identities would have had to hold for all values of ν as well as a, b and c. In this case, each resulting coefficient found in the previous sections of this Chapter represent a quadratic function in ν.

$$\text{e.g. } A_1 = (a_{11} + a_{12}\nu + a_{13}\nu^2)$$
$$A_2 = (a_{21} + a_{22}\nu + a_{23}\nu^2) \qquad \text{etc.}$$

One might expect that it would be necessary to use a standard structure (such as is described in subsection 3.3.6) with three times as many degrees of nodal freedom. This would in fact give answers to the problem. In this case a particular value of ν would have to be used. The constants a_{11}, a_{12}, etc. may, then, only have held for that value of ν and for values of ν very close to it.

The method actually used to introduce ν allows the function involving ν to be chosen as a penultimate step. This simplifies the choice of the function.

ν is included in the displacement vector for the standard solution (see equation (71)). If the coefficient solution program described in Appendix 7 were used for a series of values of ν, a number of sets of corresponding coefficients A_1, A_2, etc. would result. A polynomial curve in the variable ν could then be fitted to the values of each coefficient (A_i) to find the values of the coefficients a_{ij}, etc.

In this way, the useful range of ν could be used, (namely about 0,2 to 0,35, Popov[10]). The validity of assuming some limit to the power of ν which occurs in the functions could also be checked.

The values of the coefficients are solved by computer for each value of ν in the range and are written onto disc. A curve fitting program (given in Appendix 7) then reads these values for each coefficient in turn.

Using a regression analysis technique (Appendix 3), a parabolic curve is fitted. The constants of the fitted curve are written into a new file on disc.

A user program then only has to substitute the value of ν for a particular problem. A set of coefficients for the sixteen terms (independent of ν) of each of the twelve functions of matrix $[K_1]$ (76) results. The value of ν can then be neglected for the rest of the formation of the system stiffness matrix.

CHAPTER IV

SECONDARY CONSIDERATIONS

4.1 Introduction

This Chapter contains a description of the sections of a finite element analysis which must be considered but which do not form part of the main research. The organization of the data for a finite element program, layout of such a program and the solution of stress conditions from resultant displacements are of secondary interest in this thesis.

There are two ways of entering data into a finite element program. One method is to describe the problem through the position of the nodes and the way they are formed into elements. The other is to describe the shape of the structure and have an automatic mesh generator to determine the positions of nodes and the shapes of elements.

The latter method would appear to be the more satisfactory. There is less work for the user to do by hand and thus less chance of human error. The problem with automatic mesh generators is that it is always the special problems that require a finite element analysis and it is invariably these problems that cannot be handled by the available program. Although strides have been made in automatic mesh generation programs with the use of concepts such as the curvilinear element, no really satisfactory completely general approaches have been proposed. In this thesis, as this is not the main topic of research, data is prepared by hand. Because it is essential that data be perfectly correct, some check on data must be performed. One method of checking data is to plot it out as a mesh. Errors can then be found visually. In the program using the data in this thesis, it is essential that element nodes be numbered in anti-clockwise order. This could not be checked by a plotter program.

A comprehensive analytical program was written to accept data in a form which is convenient to the user. The program checks the validity of data and files it on disc in a suitable form for the finite element program. This "debugging" program is described below.

Structural analyses are usually performed to find stresses. Displacements are only a secondary consideration. In the stiffness method, the displacements are the primary result. The main objective of this thesis is to improve the method of obtaining displacements from the loading. The stress derivation has not been considered in detail.

Certain possibilities exist for the secondary step. The usual method is to use a stress matrix which is formed in the process of deriving a stiffness matrix. In the assumed stiffness matrix method, no such stress matrix is derived. There is no objection however to using the stress matrix from one of the other methods (e.g. matrix [M] of equation (6) in the assumed displacement procedure).

Another suggestion for finding stresses is to use a spline function interpolation of the displacements. The stresses could then be derived from the functions. A complete analysis using spline functions is suggested by Deak and Pian[11]. The problem is that this method has only been used for rectangular meshes.

A third original method of obtaining stresses is suggested in this Chapter. The method has not been developed at all and is included only in its experimental form.

An industrial package program is given in Appendix 7. This is the program used for finding stress and displacement results (Chapter V). The layout of this program is given in general terms in this Chapter. The details, where they are important, are given elsewhere. In general, package programs are uneconomical except on problems which fit their specifications very closely. The accompanying program is only included as a guide to writing specific programs using the assumed stiffness matrix. In its present form, it gives highly encouraging results although stress calculations are unreliable.

4.2 The Use of the Data "Debugging" Program

Consider a mesh of triangular elements such as the one in figure 17.

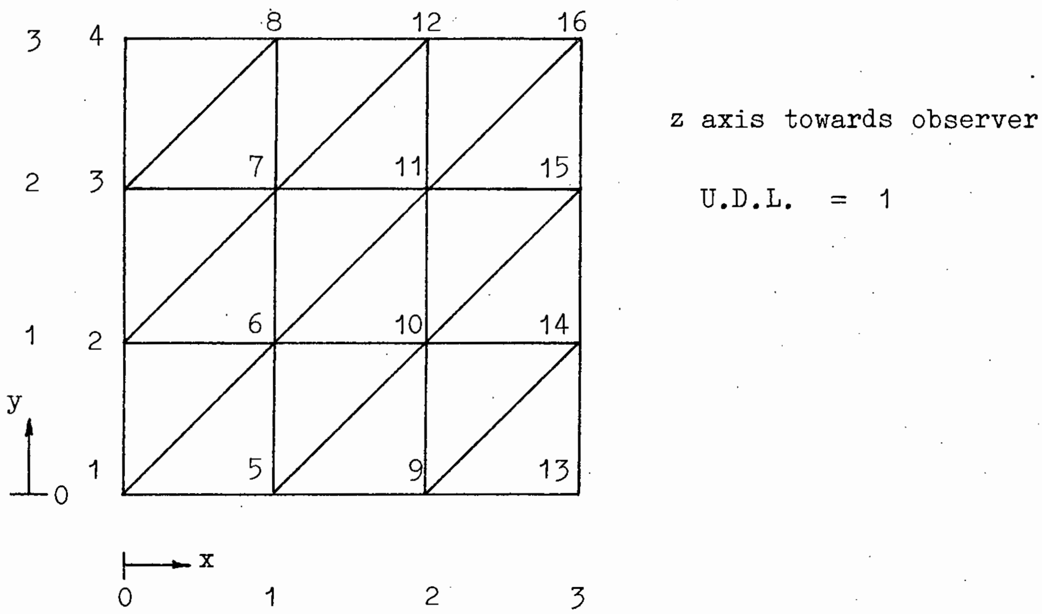


Figure 17: An Example of a Triangular Element Mesh

The data for this mesh is as follows:-

(i) Chains of integers in free format follow lines joining nodes in the figure. A zero in the chain breaks it. A comma is automatically placed at the end of a card read in free format and blanks are read as zeroes. These features can be used to break chains. The chains appear under a heading as follows:-

Card 1: TOPOLOGY LIST
Card 2: 1, 5, 9, 13,, 2, 6, 10, 14,, 3, 7, 11, 15,
Card 3: 4, 8, 12, 16,, 2, 1, 6, 5, 10, 9, 14, 13,
Card 4: 3, 2, 7,* 6, 11, 10, 15, 14,
Card 5: 4, 3, 8, 7, 12, 11, 16, 15,

(ii) The boundary of the problem is defined by a chain similar to those of the topology list under a heading as follows:

Card 6: BOUNDARY LIST
Card 7: 1, 5, 9, 13, 14, 15, 16,
Card 8: 16, 12, 8, 4, 3, 2, 1,

*Referred to later.

(iii) A card giving the highest number of a node appearing in the mesh followed by a list of node numbers and coordinates in free format and a card of zeroes appear under a heading as follows:-

```
Card 9:  NODAL POINT LIST
Card 10: 16,
Card 11: 1,,,
Card 12: 2,, 1.,
Card 13: 3,, 2.,
  :
  :
Card 26: 16, 3., 3.,
Card 27: ,,,
```

(iv) A uniformly distributed load over the whole structure or a blank card may follow.

Other load vectors and support boundary conditions cannot be checked and are thus read directly into the finite element program.

The form in which this data is required by the finite element program is as follows:-

(i) A list of coordinates given in the order of node numbers

```
e.g.     0.    0.
         0.    1.
         0.    2.
         :     :
         3.    3.
```

(ii) A list of the node numbers of each triangle given in clockwise order about the z axis (anti-clockwise in figure 17) followed by the uniformly distributed load on that triangle.

```
e.g.     1     6     2     1.0
         1     5     6     1.0
         2     7     3     1.0
         :     :     :     :
        11    15    16     1.0
```

(iii) A list of node connections for each node:

e.g.	1	2	
	1	5	
	1	6	
	2	1	
	2	3	
	2	6	
	2	7	etc.

The program given in Appendix 7 performs this transformation on the data. At the same time it checks the data for the following errors.

1. If a double comma has been omitted between two chains which end on the same node, e.g. if the comma at the end of Card 7 had been omitted, the following message would be given:

   NODE 16 IS JOINED TO ITSELF. INVALID.

2. If the same junction between nodes was repeated in the topology list, e.g. if the number 7 was left off Card 4, the following message would be given:

   JOIN 2 - 6 IS DUPLICATED IN TOPOLOGY LIST.

3. If a heading is spelt incorrectly or if for example a $ sign is punched instead of a comma, the following message is given:

   HEADING ERROR OR ALPHA CHARACTER IN BOUNDARY OR TOPOLOGY LIST.

4. If a boundary connection occurs in the boundary list, but not in the topology list, e.g. if the first '4' on Card 5 were omitted, the following message is given:

   BOUNDARY 3 - 4 IS NOT INCLUDED IN TOPOLOGY LIST.

5. If a boundary is discontinuous, for example if the boundary were not closed from node 2 onto node 1, the following two error messages are given:

   THERE IS A DISCONTINUITY IN THE BOUNDARY AT NODE 1.
   THERE IS A DISCONTINUITY IN THE BOUNDARY AT NODE 2.

6. If a connection in the interior bounds more or less than two triangles, or if a connection on the boundary bounds more or less than one triangle, e.g. if number 7 is omitted from Card 4, the following

five error messages are given:

```
JOIN  2 -  3  BOUNDS  0  TRIANGLES
JOIN  2 -  6  BOUNDS  1  TRIANGLES
JOIN  3 -  7  BOUNDS  1  TRIANGLES
JOIN  6 - 11  BOUNDS  1  TRIANGLES
JOIN  7 - 11  BOUNDS  1  TRIANGLES.
```

7. If a particular node is duplicated in, or omitted from, the nodal point list, one of the following messages is given. (The nodes do not have to be sequential but if a node is omitted, it cannot appear in the topology list.):

```
NODE 6 IS DUPLICATED IN NODAL POINT LISTS.
NODE 6 IS MISSING FROM NODAL POINT LIST.  THIS IS VALID.
```

8. If a node appears with a number higher than the size specification, for example if the first card in the nodal point list bears the number one (i.e. card 10 omitted), the error message given terminates further debugging:

```
INCORRECT SIZE SPECIFICATION.  FATAL ERROR.
```

9. If a triangle is long and narrow so that an apex lies closer to the base than one third of the base length, an error message is given. It is possible that all three nodes lie close to their opposite sides. (See figure 18.)

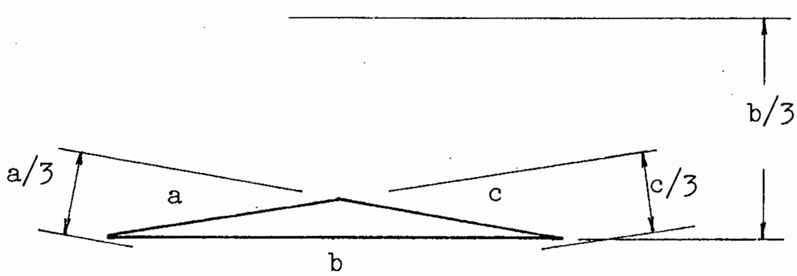


Figure 18: A badly conditioned triangle.

In this case the following message is given three times:

```
TRIANGLE  1-2-3  IS BADLY CONDITIONED BUT VALID.
```

10. If the nodes lie in a straight line, e.g. if the coordinates of node 7 are given as (2., 1.) instead of (1., 2) then the

following message is given:

TRIANGLE 2-6-7 IS INVALID. NODES FORM A STRAIGHT LINE.

11. If the nodes of a triangle coincide, e.g. if the coordinates of node 7 were given as (1., 1.) the message given is:

NODES OF TRIANGLE 2-6-7 COINCIDE.

12. If a node is used in the topology list but its coordinates are not given, e.g. if node 11 was listed as 1, 2., 2., the message is:

TRIANGLE 6-11-7 IS INVALID. NODE 11 IS NOT LISTED.

13. If a node has coordinates which cause the connections to it to overlap other triangles, e.g. if node 7 fell outside the hexagon formed by nodes 2, 6, 11, 12, 8, and 3 by having coordinates (2., 1.) the messages are:

```
NODE  7  FALLS INSIDE TRIANGLE  6-10-11
NODE  6  FALLS INSIDE TRIANGLE  2- 7- 3
NODE  6  FALLS INSIDE TRIANGLE  2- 6- 7
NODE 11  FALLS INSIDE TRIANGLE  7-12- 8
NODE 11  FALLS INSIDE TRIANGLE  7-11-12
```

If no errors are found, the data is filed. If errors are found, a message follows the error messages:

THE DATA ERROR LEVEL IS 13.

The number given is the total number of error messages.

If gross errors are made, vast quantities of error messages with little meaning are written.

The mechanics of the debugging system follow coordinate geometry principles. A description of these is outside the scope of this thesis. The program is given in Appendix 7, however.

4.3 Stress Solution by a Regression Method

One usually wants the values of bending moments and shears at the nodes chosen on a structure (as opposed to at the centroids of elements). If the stress matrices of the assumed displacement or stress method are used, different values are found at a particular node from each element adjoining it. The mean of these values is usually chosen to be closest to the correct value.

The objection to this method of stress derivation is that one is calculating values on the very edge of the domain of validity of each function. This is where errors are likely to be highest. It would be more accurate if a single value of stress could be calculated in the centre of a domain of validity for some function.

A displacement field bounded by all of the nodes adjoining the one at which stress conditions are required, could be used; e.g. in figure 17 consider node 6 to be at the centre of a domain with nodes 1, 5, 10, 8, 7 and 2 on the edges. The deflections at all these nodes are known.

Supposing a third order polynomial surface in two variables is assumed to hold over such a domain. By substitution of the coordinates and displacements of each node relative to the coordinates and displacements of the central node, a set of equations could be formed in terms of the unknown coefficients of the assumed polynomial. The values of these coefficients could be found using a regression technique (Appendix 3) to solve the rectangular matrix.

Once the displacement function is known, the bending moments and shears at the central origin follow very simply.

The next node can then be considered as the centre of a domain. This method has the added advantage that it is not necessary to first transform displacements from a global system to a local system of coordinates before solving for stresses.

In detail:

By appropriate differentiation of the assumed function, the displacements at any general point (x,y) are:

$$
\begin{aligned}
w &= C_1 + C_2x + C_3y + C_4x^2 + C_5xy + C_6y^2 + C_7x^3 + C_8x^2y + C_9xy^2 + C_{10}y^3 \\
\theta_x &= C_3 + C_5x + 2C_6y + C_8x^2 + 2C_9xy + 3C_{10}y^2 \\
\theta_y &= -C_2 - 2C_4x - C_5y - 3C_7x^2 - 2C_8xy - C_9y^2
\end{aligned}
\qquad (80)
$$

Supposing the coordinates and displacements of the central node are given the suffix 'o' while those of any other node in the domain are given the suffix 'i'.

Let the coordinates of the node i with respect to node o be:

$$x_{ri} = x_i - x_o$$
$$y_{ri} = y_i - y_o$$

$$\therefore \quad x_{ro} = 0 \quad \text{and} \quad y_{ro} = 0$$

Using these relative coordinates in the functions (80):

$$w_o = C_1, \qquad \theta_{xo} = C_3, \qquad \theta_{yo} = -C_2 \tag{81}$$

and

$$\begin{aligned} w_i &= C_1 + C_2 x_{ri} + C_3 y_{ri} + C_4 x_{ri}^2 + C_5 x_{ri} y_{ri} + C_6 y_{ri}^2 + C_7 x_{ri}^3 \\ &\quad + C_8 x_{ri}^2 y_{ri} + C_9 x_{ri} y_{ri}^2 + C_{10} y_{ri}^3 \\ \theta_{xi} &= C_3 + C_5 x_{ri} + 2C_6 y_{ri} + C_8 x_{ri}^2 + 2C_9 x_{ri} y_{ri} + 3C_{10} y_{ri}^2 \\ \theta_{yi} &= -C_2 - 2C_4 x_{ri} - C_5 y_{ri} - 3C_7 x_{ri}^2 - 2C_8 x_{ri} y_{ri} - C_9 y_{ri}^2 \end{aligned} \tag{82}$$

Substituting from (81) into (82) and putting

$$\{c\}^t = \{C_4, C_5, C_6, C_7, C_8, C_9, C_{10}\} :$$

$$\begin{bmatrix} x_{ri}^2 & x_{ri}\, y_{ri} & y_{ri}^2 & x_{ri}^3 & x_{ri}^2\, y_{ri} & x_{ri}\, y_{ri} & y_{ri} \\ 0 & x_{ri} & 2y_{ri} & 0 & x_{ri}^2 & 2x_{ri}\, y_{ri} & 3y_{ri}^2 \\ -2x_{ri} & -y_{ri} & 0 & -3x_{ri}^2 & -2x_{ri}\, y_{ri} & y_{ri}^2 & 0 \end{bmatrix} \times \{c\}$$

$$= \begin{bmatrix} w_i - w_o + x_{ri}\,\theta_{yo} - y_{ri}\,\theta_{xo} \\ \theta_{xi} - \theta_{xo} \\ \theta_{yi} - \theta_{yo} \end{bmatrix}$$

If the values of each boundary node are now substituted, provided there are at least three boundary nodes, the best values of $\{c\}$ can be found by regression analysis (Appendix 3).

The differential equations for finding the stress conditions at a point on a thin plate, applying the small deflection theory, are as follows:-

The moment in the x direction:

$$M_x = -D(\partial^2 w/\partial x^2 + \nu \partial^2 w/\partial y^2)$$

The moment in the y direction:

$$M_y = -D(\partial^2 w/\partial y^2 + \nu \partial^2 w/\partial x^2)$$

The twisting moment:

$$M_{xy} = D(1 - \nu)\ \partial^2 w/\partial x \partial y$$

The shear on an xz plane in the z direction:

$$Q_{xz} = -D(\partial^3 w/\partial x^3 + \partial^3 w/\partial x \partial y^2)$$

The shear on a yz plane in the z direction:

$$Q_{yz} = -D(\partial^3 w/\partial y^3 + \partial^3 w/\partial x^2 \partial y)$$

where $D = Eh^3/12(1 - \nu^2)$

and E is Young's modulus,

h is the material thickness,

ν is Poisson's ratio.

Applying these to the assumed functions and then substituting the values $x_{ro} = 0$ and $y_{ro} = 0$

$$M_x = -2D(C_4 + \nu C_6)$$

$$M_y = -2D(C_6 + \nu C_4)$$

$$M_{xy} = D(1 - \nu) \times C_5$$

$$Q_{xz} = -D(6C_7 + 2C_9)$$

$$Q_{yz} = -D(6C_{10} + 2C_8)$$

These values are then found by substitution from the vector $\{c\}$.

4.4 The Load Vector

The formation of the load vector corresponding to point loads and couples follows the usual finite element procedure. As no displacement function was assumed, no consistent load function can be used to decide how to apply loads which are distributed along a line or over an area. The following interpretation is used in this thesis.

Loads (forces or couples) applied along a line joining two nodes have half their total value applied at each node as an equivalent point load. Forces also cause a fixed end couple along the direction of the line. Consider figure 19 for example:

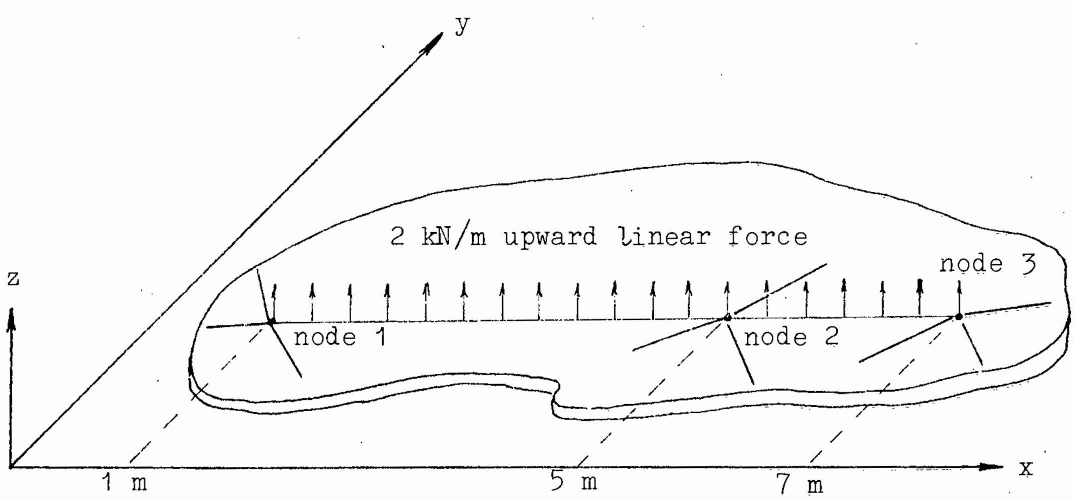


Figure 19: The Loading at Three Nodes on an undefined flat plate.

The load vector in this case would be calculated as:

$F_1 = 2 \times (5 - 1)/2 = 4$ kN

$M_{x1} = 0$

$M_{y1} = -2 \times (5 - 1)^2/12 = -8/3$ kNm

$F_2 = 4 + 2 \times (7 - 5)/2 = 6$ kN

$M_{x2} = 0$

$M_{y2} = 8/3 - 2 \times (7 - 5)^2/12 = 6$ kNm

$F_3 = 2$ kN

$M_{x3} = 0$

$M_{y3} = 2/3$ kNm

The loading on each node of a triangle with a uniformly distributed load q over its area is determined as follows.

The vertical load at each node is taken as one third of the total load on the triangle (i.e. $q/3$ × area of the triangle).

To calculate the applied moments at each node, it is assumed that as the median of a triangle bisects its area, no applied moment is caused about the median through that node. All applied moments are thus caused about an axis at right angles to the median through each node. (See figure 20).

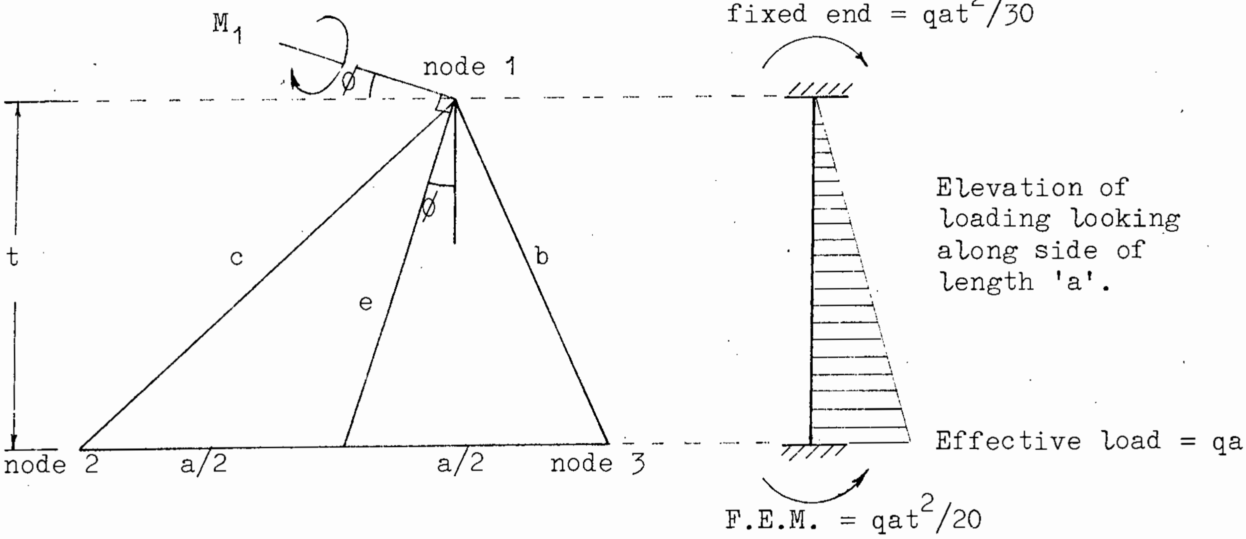


Figure 20: A Triangular Plate Loaded with a Uniformly Distributed Load q.

The length of the median e is found to be:

$$e = \tfrac{1}{2}\sqrt{2b^2 + 2c^2 - a^2} \qquad (83)$$

If t is the height of the triangle above the base of length 'a'; from figure 20:

$$M_1 \cos \phi = M_1 \, t/e$$
$$= qat^2/30$$

Hence $$M_1 = qate/30$$
$$= q \times \text{triangle area} \times \text{median length}/15$$
$$= q \, \alpha\beta\gamma\Delta \sqrt{2b^2 + 2c^2 - a^2}/120$$

by substitution from (59) and (83).

As the coordinates of the centroid of a triangle are simple to find, the loading in global coordinates is found by a transformation of the type shown in equation (65).

4.5 Application of Boundary Conditions

In the usual case, displacement boundary conditions have the prescribed value, zero. In this case, the row and column of the augmented stiffness matrix corresponding to the boundary displacement are deleted and replaced by zeroes. To prevent this from becoming a singular matrix, the leading diagonal element is replaced by a one. This results in a zero in the correct place in the solved displacement vector.

The above is the mechanical operation. What has in fact happened is that one of the displacements was not an unknown value. The known value was substituted into each of the equations and removed to the right hand side of the equation. This left more equations than unknowns. One of them must have been a linear combination of the others. As the stiffness matrix must still be symmetrical after removal of the redundant equation (Betti-Maxwell), the obvious equation to delete must be the one corresponding to the row removed from the matrix.

The mechanical procedure accounting for a non-zero boundary condition is as follows:-

1. The product of the value of the boundary condition and its corresponding column in the stiffness matrix is subtracted from each load vector.

2. The row and column corresponding to the boundary condition displacement are replaced by a row and column of zeroes respectively.

3. The leading diagonal (zero) element corresponding to the boundary condition is replaced by a one.

4. The row of the load vector corresponding to the boundary condition is replaced by the value of the boundary condition.

Step 3 prevents the matrix from being singular. Steps 3 and 4 give the value of the boundary condition in the correct position in the solved displacement vector.

4.6 Layout and Use of the Industrial Package Program

This program is designed so that an explicit stiffness matrix (with coefficients as accurate as the computer using it can attain) is first created on disc. A finite element program and a data debugging program are then compiled and finally the unnecessary elements of the explicit stiffness matrix creation file are deleted. It is then only necessary to use a minimum number of control cards and data cards for a particular problem.

The details of the program are given in Appendix 7. The sequence of a first run of this program is as follows:-

1. A disc file containing the data for the explicit stiffness matrix in terms of twenty five unknown coefficients is created.

2. The data "debugging" program is stored for use with the finite element program.

3. A coefficient solution program with its subroutines is compiled and stored.

4. A curve fitting routine for introducing the effect of Poisson's ratio to the coefficients is compiled and stored.

5. The data giving the nodal coordinates, element description, load vector and boundary conditions for the standard plate solution is calculated. The range and number of increments of Poisson's ratio for which solutions of the coefficients are required (see section 3.4.5) are given. An execution of the coefficient solution program 3. uses this data and the data in 1. to form a number of sets of coefficients through the range of Poisson's ratio. The coefficients are written into a new disc file.

6. An execution of the curve fitting program 4. uses the file created in 5. to fit the best parabolic curve in ν (Poisson's ratio) to each coefficient in turn. The coefficients of these curves are stored as three matrices (to be multiplied by 1, ν and ν^2 respectively - see matrices (84a), (84b) and (84c)) in a new disc file.

7. A finite element program which uses the coefficients in 6. above is formed by updating the coefficient solution program 3. This is done because the assumed functions, their parameters and the transformation matrices are the same in both programs.

CHAPTER V

RESULTS USING THE ASSUMED STIFFNESS METHOD

5.1 Introduction

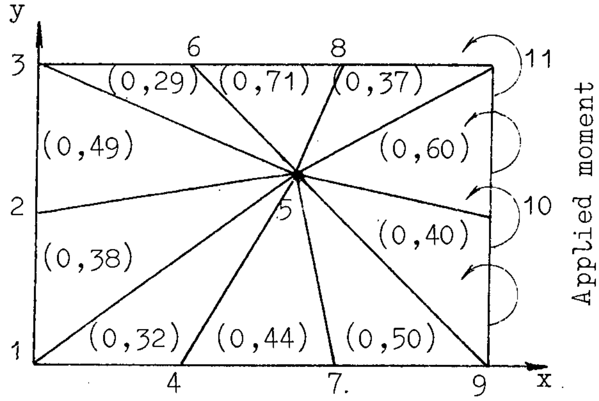
Any flat plate bending problem can be solved by using the industrial package program prepared. Data for simple shapes of plates was stored on disc. Boundary restrictions were introduced to simulate different support conditions. Solutions to many standard problems were obtained. A selection of these is presented here.

5.2 Three Different sets of Coefficients of the Explicit Stiffness Matrix

As was mentioned in the introduction to the method of assumed stiffness, it is possible to obtain different stiffness matrices for the same element. The stiffness varies according to the position of the element within a structure. In subsection 3.3.6 it was shown that a plate with a given number of degrees of nodal freedom could have various configurations of elements. Three possibilities were given in figure 14. The positions of the nodes can also be varied. If the values of the coefficients of the stiffness matrix are solved as described in section 3.4.4, there are as many solutions as there are possible element conformations of the plate.

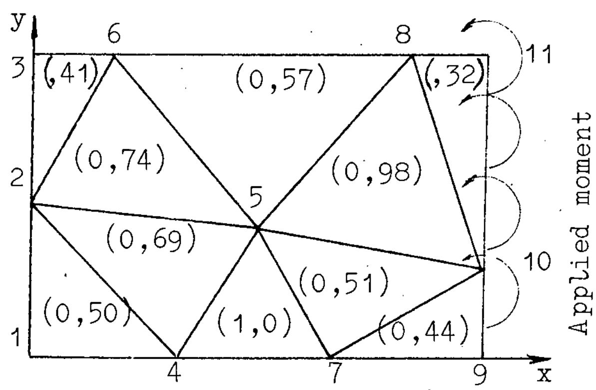
The matrix to be solved for the values of the coefficients is ill-conditioned if elements of the structure resemble one another closely. When two or more elements are similar, therefore, the values of coefficients are completely unreliable.

The coordinates of nodes must be carefully chosen so that each element has a different non-similarity ratio (see definition). It can be assumed that an optimum set of coordinates exists to give the best conditioning of the matrix. No attempt was made to optimise the coordinates of nodes. Each layout of elements given in figure 14 was considered in turn. Co-ordinates were varied until a wide range of non-similarity ratios was obtained for the elements. The final sets of coefficients for each of the three configurations are identified as: Mark I, II and III corresponding to figures 21(a), (b) and (c) respectively.



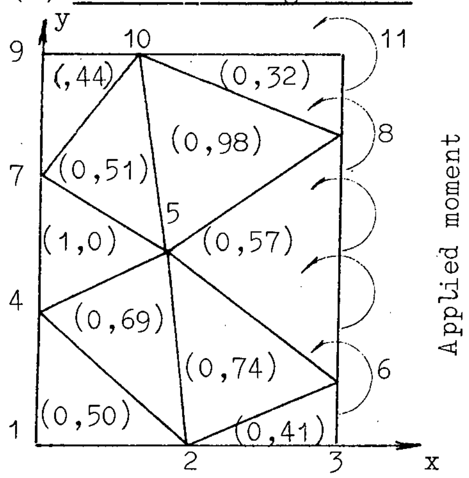
NODE	X	Y
1	0	0
2	0	1
3	0	2
4	1	0
5	1,75	1,25
6	1	2
7	1,75	0
8	2	2
9	3	0
10	3	1
11	3	2

(a) <u>Mark I Configuration</u>



1	0	0
2	0	1
3	0	2
4	1	0
5	1,5	0,866
6	0,5	2
7	2	0
8	2,5	2
9	3	0
10	3	0,6
11	3	2

(b) <u>Mark II Configuration</u>



1	0	0
2	1	0
3	2	0
4	0	1
5	0,866	1,5
6	2	0,5
7	2	0
8	2	2,5
9	0	3
10	0,6	3
11	2	3

(c) <u>Mark III Configuration</u>

The non-similarity ratio for each element is given in brackets.

<u>Figure 21</u>: <u>Element Topology used on the Standard Plate to find Mark I Mark II and Mark III Sets of Coordinates Respectively</u>.

It is found that the values of a coefficient vary almost parabolically with respect to Poisson's ratio only if all elements are dissimilar. The variation is random if similar elements exist. (See section 3.4.5.)

A comparison of the results obtained from each set of coefficients is given in section 5.6. The Mark III coefficients are given in matrices (84).

5.3 The Layout of a Set of Results

A full set of results is written for each load condition on a structure. Each set of results starts on a new page. The left hand column gives the numbers of the nodes in succession. The displacements and stresses (calculated as described in section 4.3) appear under appropriate headings in subsequent columns.

Results for standard problems are usually published as non-dimensional values and a general multiplier. An example of the results of a square plate with point supports on its corners and uniformly loaded is given in Table 2. In this case the multiplier is:

$$qL^4/D = 12(1 - \nu^2)\ qL^4/Eh^4$$

Results were obtained for:

Poisson's ratio	ν	=	0,3
Distributed load	q	=	1,0
Young's modulus	E	=	10,92
Length of a side	L	=	0,5

The results given must therefore be multiplied by 16 for comparison with published non-dimensional values. As the mesh is coarse, results are not good. (A comparison of these results is made in a later section.)

5.4 Convergence of Results

All examples tested showed a definite convergence to a correct solution. Table 3 gives an example of the behaviour of one set of results.

LOAD VECTOR 1

NODE NO.	VERTICAL DISPL. (LENGTH)	ROTATION ABOUT X (DIMENSIONLESS)	ROTATION ABOUT Y	MOMENT IN X DIRN.	MOMENT IN Y DIRN. (FORCE*LENGTH/LENGTH)	TWISTING MOMENT	SHEAR IN X DIRN. (FORCE/LENGTH)	SHEAR IN Y DIRN.
1	.0000000	.7052478-02	-.7052454-02	-.1519016+00	-.1519016+00	-.5798263-01	.2038796+01	.2038796+01
2	.1174021-02	.4351947-02	-.2855558-02	-.2202602-01	.2385972-01	-.2787419-01	.4088953+00	-.3415031-01
3	.1549653-02	-.1450610-03	-.1441512-02	-.2242343-02	.3416437-01	-.4829767-04	.1313326-01	.7354163-02
4	.1145110-02	-.4287348-02	-.2834884-02	-.2068451-01	.2219350-01	.2856675-01	.3178199+00	.1426777+00
5	.0000000	-.6935146-02	-.6935118-02	-.1489928+00	-.1489929+00	.5685680-01	.1997729+01	-.1997730+01
6	.1174019-02	.2855582-02	-.4351933-02	.2385961-01	-.2202604-01	-.2787419-01	-.3415098-01	.4088965+00
7	.1577457-02	.2725363-02	-.2725856-02	.1855957-01	.1855965-01	-.1733115-01	.5667147-01	.5667176-01
8	.1796014-02	-.1392574-04	-.1871450-02	.1515792-01	.2607966-01	-.1694773-02	.4264782-01	.1541609-01
9	.1574361-02	-.2754355-02	-.2754330-02	.1865549-01	.1865560-01	.1400636-01	.4298126-01	-.4298158-01
10	.1145107-02	-.2834917-02	-.4287324-02	.2219345-01	-.2068446-01	.2856659-01	-.1426764+00	-.3178186+00
11	.1549650-02	.1441534-02	.1450567-03	.3416422-01	-.2242420-02	-.4833910-04	.7353849-02	.1313534-01
12	.1796014-02	.1871463-02	.1392173-04	.2607956-01	.1515799-01	-.1694764-02	.1541611-01	.4264811-01
13	.1941141-02	-.4792713-08	.1395726-09	.2300811-01	.2300822-01	-.3114944-02	-.4548374-07	.1470261-06
14	.1796013-02	-.1871478-02	-.1392160-04	.2607955-01	.1515806-01	-.1694739-02	-.1541615-01	-.4264785-01
15	.1549646-02	-.1441571-02	-.1450566-03	.3416419-01	-.2242168-02	-.4830908-04	-.7353067-02	-.1313529-01
16	.1145110-02	.2834898-02	.4287333-02	.2219350-01	-.2068451-01	.2856674-01	.1426781+00	.3178191+00
17	.1574362-02	.2754344-02	.2754336-02	.1865551-01	.1865555-01	.1400642-01	-.4298135-01	.4298172-01
18	.1796014-02	.1391848-04	.1871450-02	.1515792-01	.2607965-01	-.1694732-02	-.4264788-01	-.1541592-01
19	.1577456-02	-.2725878-02	.2725852-02	.1855955-01	.1855966-01	-.1733109-01	-.5667161-01	-.5667177-01
20	.1174016-02	-.2855596-02	.4351923-02	.2385960-01	-.2202593-01	-.2787403-01	.3414953-01	-.4088943+00
21	.0000000	.6935139-02	.6935134-02	-.1489932+00	-.1489932+00	.5685694-01	-.1997734+01	.1997733+01
22	.1145110-02	.4287342-02	.2834891-02	-.2068457-01	.2219347-01	.2856684-01	-.3178209+00	-.1426785+00
23	.1549653-02	.1450574-03	.1441511-02	-.2242383-02	.3416435-01	-.4822864-04	-.1313352-01	-.7354392-02
24	.1174021-02	-.4351951-02	.2855551-02	-.2202602-01	.2385971-01	-.2787412-01	-.4088950+00	.3414993-01
25	.0000000	-.7052480-02	.7052439-02	-.1519014+00	-.1519014+00	-.5798250-01	-.2038792+01	-.2038794+01

TABLE 2. AN EXAMPLE OF THE FORMAT OF A SET OF RESULTS.

A square plate of side, L, with point supports at its four corners under a uniformly distributed load, q, is considered. Meshes of triangular elements similar to figure 17 are used in this problem. Results of systems of 4 squares by 4 squares, 8 by 8 and 16 by 16 are given. The results do not converge as rapidly as the square element solutions given by Zienkiewicz and Cheung[12] (because the problem is ideally suited to the square element.)

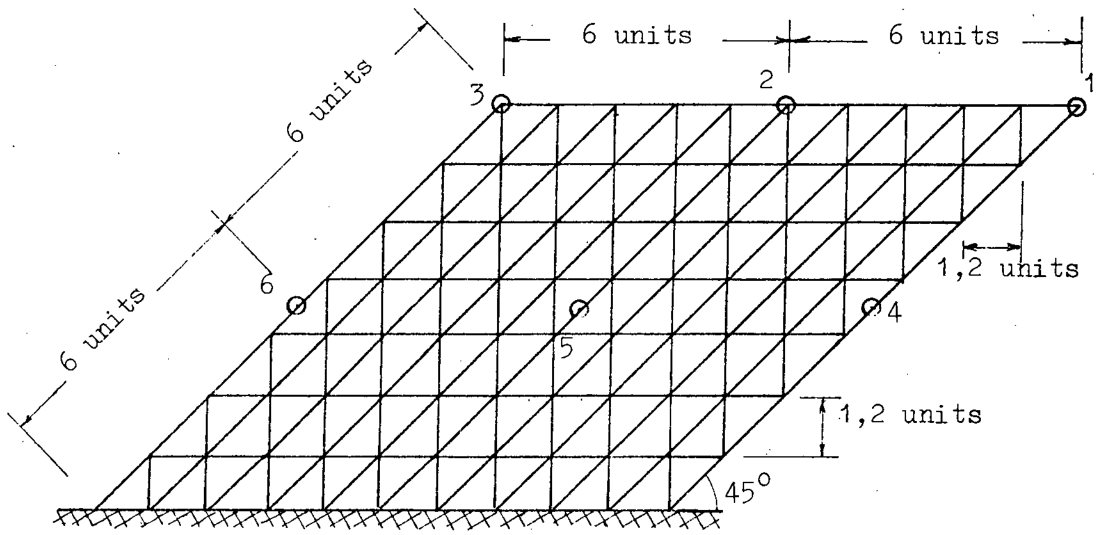
TABLE 3

A COMPARISON OF THE CONVERGENCE OF RESULTS OBTAINED BY THE ASSUMED STIFFNESS METHOD AND SQUARE ELEMENTS

Method of Solution	Deflection of Centre of Side	Deflection of Centre of Plate
TRIANGULAR FINITE ELEMENT		
4 × 4	0,0246	0,0319
8 × 8	0,0205	0,0278
16 × 16	0,0194	0,0268
SQUARE FINITE ELEMENT[12]		
2 × 2	0,0126	0,0176
4 × 4	0,0165	0,0232
6 × 6	0,0173	0,0244
Marcus[12]	0,0180	0,0281
Lee and Balleros[12]	0,0170	0,0265
Multiplier	$q L^4/D$	$q L^4/D$

5.5 A Comparison with other Numerical Methods

Gellert and Gluck[7] analysed the skew cantilever shown in figure 22 using eight different types of finite element, a finite difference approach and by experimental measurement. The same element mesh was used for all solutions. Although the assumed stiffness triangle developed here gives comparable results with only 25 nodes, the results given at the bottom of Table 4 are those for the 88 node configuration shown.



Uniformly distributed load	q	=	0,26066 force/length^2 units
Young's Modulus	E	=	$10,5 \times 10^6$ force/length^2 units
The plate thickness	h	=	0,125 units
Poisson's ratio	ν	=	0,3

Figure 22: Skew Cantilever analysed by various Numerical Methods

TABLE 4

COMPARISON OF DEFLECTIONS OF A SKEW CANTILEVERED PLATE OBTAINED BY VARIOUS ANALYSIS METHODS

Analysis Method No.	Deflection at Point					
	1	2	3	4	5	6
1	0,296	0,198	0,114	0,114	0,052	0,020
2	0,294	0,197	0,118	0,113	0,051	0,020
3	0,279	0,187	0,116	0,108	0,050	0,021
4	0,257	0,170	0,105	0,106	0,047	0,020
5	0,278	0,184	0,105	0,111	0,047	0,016
6*	0,421	0,296	0,199	0,171	0,082	0,044
7	0,281	0,188	0,111	0,111	0,049	0,018
8	0,273	0,184	0,110	0,106	0,048	0,019
9	0,286	0,191	0,112	0,116	0,052	0,020
10	0,297	0,204	0,121	0,129	0,056	0,022
11	0,282	0,187	0,108	0,114	0,052	0,019
Mean	0,2823	0,1890	0,1120	0,1128	0,0504	0,0195

Key to analysis methods (Gellert and Cluck[7]):

1. Rectangular element based on twelve-term polynomial.
2. Rectangular element due to Melosh.
3. Rectangular fully compatable element.
4. Triangular assumed displacement element omitting xy term
5. Triangular assumed displacement element combining x^2y and xy^2 terms.
6. Triangular assumed displacement element including a central nodal translation.
7. Triangular fully compatable element.
8. Triangular element due to Shie et al.
9. Variational finite difference method.
10. Experimental measurement.
11. Assumed stiffness method of this thesis.

---

* These results were excluded from the mean.

This shows that the assumed stiffness matrix method of analysis is certainly as reliable in this problem as any of the other methods. The coefficients derived from element configuration, Mark II (figure 21) were used.

5.6 The Behaviour of the Three Sets of Coefficients

A series of examples with published solutions[12] was analysed using each set of coefficients (described in section 5.2). A comparison is made between results obtained using the three sets of coefficients and these solutions.

TABLE 5

A COMPARISON ON THREE SETS OF COEFFICIENTS ON CERTAIN STRUCTURAL DEFLECTIONS FOR VARIOUS GRIDS

Deflection Identity No.	Published Solution	Grid Density	Coefficient Mark I	Coefficient Mark II	Coefficient Mark III
1	0,282	8 × 11	0,320	0,229	0,282
2	0,133	8 × 8	0,143	0,117	0,133
3	0,0272	4 × 4	0,0310	0,0200	0,0319
		8 × 8	0,0290	0,0210	0,0278
		16 × 16	0,0352	0,0157	0,0268
4	0,0175	4 × 4	0,0248	0,0128	0,0246
		8 × 8	0,0220	0,0101	0,0205
		16 × 16	0,0256	0,0114	0,0194

Key to deflection identification numbers:

1. The deflection of point 1. in figure 22.
2. The maximum deflection of a cantilevered square plate under a uniformly distributed load.
3. The deflection of the centre of a square plate with point supports.
4. The deflection of the midpoint of a side of the plate described in 3. above.

It was also found that the anticlastic curvature effect in the analysis of a cantilevered square plate was more apparent in the Mark III coefficient analysis than in the other two. Figure 23 shows the element conformation for the analysis of a square plate subjected to a uniformly distributed load. Table 6 gives a comparison of the deflections of the edge opposite the fixed edge obtained by using the three sets of coefficients.

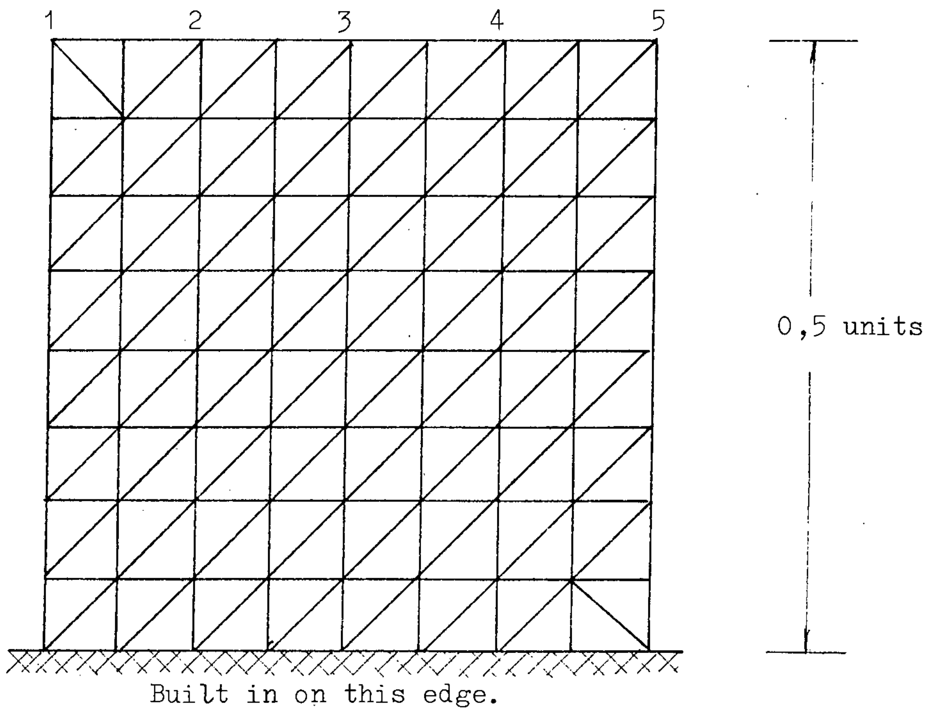


Figure 23: Element Topology for a Cantilevered Square Plate

TABLE 6

COMPARISON OF THE ANTICLASTIC CURVATURE EFFECTS OF THREE SETS OF COEFFICIENTS

Displacement Number (cf. Figure 23)	Coefficient Marks (cf. Figure 21)		
	I	II	III
1	0,00876	0,00720	0,00819
2	0,00883	0,00714	0,00827
3	0,00887	0,00728	0,00830
4	0,00883	0,00714	0,00827
5	0,00876	0,00719	0,00819

5.7 Application to a Range of Problems

Analyses were performed on square plates with different boundary conditions and types of loading. To illustrate the compliant shape of the element, a circular plate was also analysed. It was found that the Mark II coefficients gave better results than the other sets of coefficients when the plate had fully fixed (built-in) edges. In all other cases the Mark III results were better. In Table 7 the results of some analyses are given. As each plate was symmetrical about two orthogonal lines, only one quarter was considered. In all the results, 81 nodes were used. Only the maximum deflection on each plate is given.

TABLE 7

PERCENTAGE ACCURACY OF THE MAXIMUM DEFLECTION OF VARIOUS PLATES USING AN 81 NODE MESH

Plate Description	Coefficient Mark	Result	Published Value	% Error
1	III	0,0268	0,0265	1,1
2	III	0,133	0,134	0,8
3	III	0,00420	0,00406	3,1
3	II	0,00398	0,00406	2,0
4	III	0,00127	0,0116	9,5
4	II	0,00112	0,0116	3,5
5	III	0,00142	0,00126	12,7
5	II	0,00124	0,00126	1,6
6	III	0,00663	0,0056	18,0
6	II	0,00527	0,0056	5,3
7	III	0,0176	0,0156	12,8
7	II	0,0139	0,0156	10,9
8	III	0,0234	0,0199	17,6
8	II	0,0165	0,0199	17,1

Plate description key:

1. Square plate with point supports at its corners and under a uniformly distributed load.[12]

2. Square cantilever under a uniformly distributed load.[12]
3. Square plate simply supported all round under a uniformly distributed load.[5]
4. Square plate simply supported all round with a central point load.[5]
5. Square plate clamped all round under a uniformly distributed load.[5]
6. Square plate clamped all round with a central point load.[5]
7. Circular plate with clamped edges under a uniformly distributed load.[13]
8. Circular plate with clamped edges and a central point load.[13]

5.8 Stress Results

As the shear and bending moment differential equations are of a lower order with respect to the loading, the accuracies of stresses should be better than the deflection accuracy.

The stress matrix from another finite element method, a finite difference operator or a spline interpolation could be used to find stresses. If this were done nothing new would be added to this thesis. The undeveloped method described in section 4.3 was therefore used experimentally.

The results at nodes on the edge of a problem are not good and corners are completely unreliable unless fictitious nodes (e.g. with mirror image deflections for a clamped edge) are introduced outside the boundaries of the plate.

In the simply supported square plate with a uniformly distributed load (Table 6, No. 3) the errors in the bending moments were:

$< 2\%$ for an internal node
$< 4\%$ for an edge node
9% for a corner node.

It can always be arranged that an edge node appears to be an internal node, in which case stress accuracies are within 2% compared with 3,1% deflection accuracy on the same problem. This compares favourably with results obtained by any of the other methods of stress solution.

5.9 Execution Times

An example in which 128 elements were used took 48 seconds for a 'complete solution'. This is 0,375 seconds per element. A 'complete solution' involves:

1. The calculation of the dimensions of the elements from the nodal coordinates.
2. The formation of the stiffness matrix for each element in local coordinates.
3. The compilation and application of transformation matrices to create the system stiffness matrix.
4. The formation of the load vector.
5. The solution of the set of equations formed.
6. The determination of the streses from the displacements calculated.

In the half band width reduction program (Appendices 1 and 7) the solution time varies linearly with the number of nodes in the length of a plate and as the square of the number across the width. (Nodes must be numbered across the width.) If the number of elements in each direction is doubled, the inversion time is eight times as long. A small problem therefore takes a shorter average time per element.

A 32 element problem takes 11 seconds, an average of 0,34 seconds per element for a 'complete solution'.

The time taken just to form the element stiffness matrix for the explicit assumed displacement method matrix was 0,512 seconds. The time taken for an arithmetic formation of the same matrix was 6,103 seconds.

5.10 The Explicit Assumed Stiffness Matrix

The package program described, creates the coefficient matrices and stores them on disc. The accuracies of these coefficients depend only on the capabilities of computer forming them. To show the form of the explicit matrix, the Mark III coefficients have been rounded to two decimal places. These rounded coefficients will not necessarily give reliable results.

The stiffness matrix, load and displacement vectors for the element given in figure 15 are as follows:-

$$[K] = \begin{bmatrix} K(1,1) & K(1,2) & K(1,3) & K(1,4) & K(1,5) & K(1,6) & K(1,7) & K(1,8) & K(1,9) \\ & K(2,2) & K(2,3) & K(2,4) & K(2,5) & K(2,6) & K(2,7) & K(2,8) & K(2,9) \\ & & K(3,3) & K(3,4) & K(3,5) & K(3,6) & K(3,7) & K(3,8) & K(3,9) \\ & & & K(4,4) & K(4,5) & K(4,6) & K(4,7) & K(4,8) & K(4,9) \\ & & & & K(5,5) & K(5,6) & K(5,7) & K(5,8) & K(5,9) \\ \text{SYMMETRICAL} & & & & & K(6,6) & K(6,7) & K(6,8) & K(6,9) \\ & & & & & & K(7,7) & K(7,8) & K(7,9) \\ & & & & & & & K(8,8) & K(8,9) \\ & & & & & & & & K(9,9) \end{bmatrix}$$

$$\{d\}^t = \{w_1, \sqrt{bc}\ \theta_{x1}, \sqrt{bc}\ \theta_{y1}, w_2, \sqrt{ca}\ \theta_{x2}, \sqrt{ca}\ \theta_{y2}, w_3, \sqrt{ab}\ \theta_{x3}, \sqrt{ab}\ \theta_{y3}\}$$

$$\{f\}^t = \{F_1, M_{x1}/\sqrt{bc}, M_{y1}/\sqrt{bc}, F_2, M_{x2}/\sqrt{ca}, M_{y2}/\sqrt{ca}, F_3, M_{x3}/\sqrt{ab}, M_{y3}/\sqrt{ab}\}$$

The values of the matrix [K] can be found by substitution from matrix $[K_3]$ which is given by:

$$[K_3] = \left(Eh^3/6(1-\nu)\ abc\Delta\right)\left([C_1] + \nu[C_2] + \nu^2[C_3]\right)[P]$$

where:

$$[P] = \begin{bmatrix} a\alpha & b\beta & c\gamma \\ a\beta & b\gamma & c\alpha \\ a\gamma & b\alpha & c\beta \\ a\Delta & b\Delta & c\Delta \\ b\alpha & c\beta & a\gamma \\ b\beta & c\gamma & a\alpha \\ b\gamma & c\alpha & a\beta \\ b\Delta & c\Delta & a\Delta \\ c\alpha & a\beta & b\gamma \\ c\beta & a\gamma & b\alpha \\ c\gamma & a\alpha & b\beta \\ c\Delta & a\Delta & b\Delta \\ \beta\gamma\Delta & \alpha\gamma\Delta & \alpha\beta\Delta \\ \alpha\gamma\Delta & \alpha\beta\Delta & \beta\gamma\Delta \\ \alpha\beta\Delta & \beta\gamma\Delta & \alpha\gamma\Delta \\ \alpha\beta\gamma & \alpha\beta\gamma & \alpha\beta\gamma \end{bmatrix} ; \quad [K_3] = \begin{bmatrix} K(1,1) & K(4,4) & K(7,7) \\ K(1,2) & K(4,5) & K(7,8) \\ K(1,3) & K(4,6) & K(7,9) \\ K(1,4) & K(4,7) & K(1,7) \\ K(1,5) & K(4,8) & K(2,7) \\ K(1,6) & K(4,9) & K(3,7) \\ K(2,2) & K(5,5) & K(8,8) \\ K(2,3) & K(5,6) & K(8,9) \\ K(2,4) & K(5,7) & K(1,8) \\ K(2,5) & K(5,8) & K(2,8) \\ K(2,6) & K(5,9) & K(3,8) \\ K(3,3) & K(6,6) & K(9,9) \\ K(3,4) & K(6,7) & K(1,9) \\ K(3,5) & K(6,8) & K(2,9) \\ K(3,6) & K(6,9) & K(3,9) \end{bmatrix}$$

$[C_1] =$ (84a)

41.04	18.21	18.21	-17.03	27.25	35.09	25.08	-27.69	27.25	25.08	35.09	-27.69	-22.90	-30.13	-30.13	19.68
.00	1.22	-1.22	.00	-5.71	2.32	-7.32	7.59	5.71	7.32	-2.32	-7.59	.00	4.91	-4.91	.00
-12.35	-1.97	-1.97	-.44	-13.84	-10.15	-10.15	13.62	-13.84	-10.15	-10.15	13.62	4.13	10.22	10.22	-3.86
-20.52	-10.19	-8.02	8.52	-10.19	-20.52	-8.02	8.52	-17.06	-17.06	-14.57	19.17	11.45	11.45	18.67	-9.84
-.43	-2.86	-4.47	3.80	-.26	-1.27	-1.49	.99	2.84	2.86	-2.75	-3.80	3.67	1.92	-1.25	-1.26
5.67	6.18	4.48	-6.35	1.35	6.18	.62	.22	5.67	7.67	4.48	-7.27	-4.44	-2.06	-5.79	1.93
.57	-.08	-.08	-1.61	.00	-1.25	1.25	.00	.00	1.25	-1.25	.00	1.87	.00	.00	-.79
.00	-.98	.98	.00	1.93	-2.46	2.46	-2.06	-1.93	-2.46	2.46	2.06	.00	-2.41	2.41	.00
1.27	.26	1.49	-.99	2.86	.43	4.47	-3.80	-2.86	-2.84	2.75	3.80	-1.92	-3.67	1.25	1.26
.30	.33	1.00	-1.38	.33	.30	1.00	-1.38	-.33	-.33	-1.06	1.38	.34	.34	-.11	-.18
.28	-.16	-1.07	-.04	-.28	-.37	-.32	1.25	.28	1.56	-1.07	-1.25	1.04	.30	-.37	-.45
3.49	-.90	-.90	4.14	6.81	2.70	2.70	-6.92	6.81	2.70	2.70	-6.92	.00	-2.62	-2.62	.00
6.18	1.35	.62	.22	6.18	5.67	4.48	-6.35	7.67	5.67	4.48	-7.27	-2.06	-4.44	-5.79	1.93
.37	.28	.32	-1.25	.16	-.28	1.07	.04	-1.56	-.28	1.07	1.25	-.30	-1.04	.37	.45
-2.59	-.93	-.67	1.13	-.93	-2.59	-.67	1.13	-2.59	-2.59	-.67	2.38	.60	.68	2.76	-.97

$[C_2] =$ (84b)

205.24	-99.94	-99.94	115.84	-181.45	-173.52	-142.58	183.54	-181.45	-142.58	-173.52	183.54	93.40	143.88	143.88	-55.95
.00	-7.33	7.33	.00	16.25	-16.09	31.56	-28.44	-16.25	-31.56	16.09	28.44	.00	-21.07	21.07	.00
80.81	.94	.94	27.57	91.77	50.87	50.87	-90.72	91.77	50.87	50.87	-90.72	-11.73	-55.20	-55.20	18.26
102.62	69.40	30.53	-57.92	69.40	102.62	30.53	-57.92	112.05	112.05	70.90	-125.62	-46.70	-46.70	-97.18	27.97
4.75	8.12	16.13	-14.22	2.72	3.60	10.06	-3.21	-15.43	-8.12	20.84	14.22	-17.19	-11.45	3.88	7.29
-31.40	-40.41	-19.47	42.00	.27	-40.41	-1.21	-13.79	-31.40	-51.36	-19.47	48.73	20.01	5.86	35.18	-9.13
-2.48	.65	.65	4.45	.00	3.87	-3.87	.00	.00	-3.87	3.87	.00	-5.09	.00	.00	2.26
.00	.47	-.47	.00	-9.13	10.54	-10.54	5.86	9.13	10.54	-10.54	-5.86	.00	11.91	-11.91	.00
-3.60	-2.72	-10.06	3.21	-8.12	-4.75	-16.13	14.22	8.12	15.43	-20.84	-14.22	11.45	17.19	-3.88	-7.29
-2.84	-2.55	-3.73	5.31	-2.55	-2.84	-3.73	5.31	2.55	2.55	2.13	-5.31	-.19	-.19	-.38	3.07
-1.75	1.41	5.34	1.71	1.75	2.21	-.13	-4.64	-1.75	-6.92	5.34	4.64	-7.87	-1.10	1.93	3.28
-14.32	10.56	10.56	-35.47	-45.36	-15.69	-15.69	45.88	-45.36	-15.69	-15.69	45.88	.00	14.90	14.90	.00
-40.41	.27	-1.21	-13.79	-40.41	-31.40	-19.47	42.00	-51.36	-31.40	-19.47	48.73	5.86	20.01	35.18	-9.13
-2.21	-1.75	.13	4.64	-1.41	1.75	-5.34	-1.71	6.92	1.75	-5.34	-4.64	1.10	7.87	-1.93	-3.28
16.50	.98	1.00	-3.15	.98	16.50	1.00	-3.15	16.50	16.50	1.00	-12.42	-2.82	-2.82	-20.34	5.36

$[C_3] =$ (84c)

331.02	173.54	173.54	-234.67	375.32	274.25	266.55	-377.83	375.32	266.55	274.25	-377.83	-110.04	-221.08	-221.08	11.48
.00	14.92	-14.92	.00	-2.11	35.32	-39.17	26.68	2.11	39.17	-35.32	-26.68	.00	26.30	-26.30	.00
-165.87	12.28	12.28	-85.36	-188.92	-84.24	-84.24	187.66	-188.92	-84.24	-84.24	187.66	3.62	97.96	97.96	-28.34
-165.51	-141.15	-32.39	117.34	-141.15	-165.51	-32.39	117.34	-234.17	-234.17	-108.75	260.50	55.02	55.02	166.06	-5.74
-13.92	-1.06	-14.90	13.34	-8.34	-2.57	-23.26	5.44	24.26	1.06	-49.23	-13.34	26.59	24.27	.28	-15.85
56.44	82.93	27.79	-86.42	-11.73	82.93	-.54	42.68	56.44	105.98	27.79	-101.24	-30.11	-1.81	-67.85	14.17
-4.05	-4.71	-4.71	8.70	.00	-.96	.96	.00	.00	.96	-.96	.00	-.83	.00	.00	.75
.00	6.14	-6.14	.00	14.17	-13.15	13.15	-1.81	-14.17	-13.15	13.15	1.81	.00	-18.62	18.62	.00
2.57	8.34	23.26	-5.44	1.06	13.92	14.30	-13.34	-1.06	-24.26	49.23	13.34	-24.27	-26.59	-.28	15.85
4.77	8.22	4.59	-6.41	8.22	4.77	4.59	-6.41	-8.22	-8.22	2.78	6.41	-2.91	-2.91	5.36	-8.30
2.77	-5.63	-6.77	-4.13	-2.77	-4.59	4.75	3.83	2.77	9.58	-6.77	-3.83	16.40	-.30	-3.31	-6.88
15.50	-31.25	-31.25	90.42	93.83	30.36	30.36	-94.46	93.83	30.36	30.36	-94.46	.00	-28.97	-28.97	.00
82.93	-11.73	-.54	42.68	82.93	56.44	27.79	-86.42	105.98	56.44	27.79	-101.24	-1.81	-30.11	-67.85	14.17
4.59	2.77	-4.75	-3.83	5.63	-2.77	6.77	4.13	-9.58	-2.77	6.77	3.83	.30	-16.40	3.31	6.88
-34.22	4.94	2.40	1.01	4.94	-34.22	2.40	1.01	-34.22	-34.22	2.40	21.34	4.32	4.32	46.22	-11.22

$[C_1]$, $[C_2]$ and $[C_3]$ are given in matrices (84a), (b) and (c) respectively and:

E is Young's Modulus

ν is Poisson's Ratio

h is the material thickness

$$\alpha = \sqrt{-a + b + c}$$

$$\beta = \sqrt{a - b + c}$$

$$\gamma = \sqrt{a + b - c}$$

$$\Delta = \sqrt{a + b + c}.$$

CHAPTER VI

SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

6.1 Summary

An explicit stiffness matrix for an assumed displacement method trianglular plate bending element was developed. This was done by performing all the steps algebraically using a specially designed computer technique.

To form a stiffness matrix explicitly requires one twelfth of the time of a fully numerical computation.

The derivation of the stiffness matrix for this element is unsatisfactory. The explicit version allows one to examine the reasons for this. The assumed displacement function is the cause of the irregular behaviour of the element.

The derivation of a stiffness matrix using the assumed stress method was then considered. Although an explicit matrix was found, it was considered to be too large for practical use.

As neither of these essentially numerical methods yielded a satisfactory explicit matrix, an original method was developed from first principles. This, the assumed stiffness method, gave the most satisfactory explicit matrix. Using this matrix, a problem can be solved in one twentieth of the time required by a numerical stiffness matrix formation.

Three different sets of coefficients for the explicit matrix were derived. One of the sets (Mark III) gave more reliable results in most cases. The other two sets of coefficients consistently gave answers which were on opposite sides of the 'exact' solution. Results obtained using all these sets of coefficients improved with the number of elements considered.

The anticlastic curvature effect was more noticeable in results found using two of the sets of coefficients (Mark I and Mark III) while the other set of coefficients was found to give better results in clamped edge problems.

A method of obtaining stress conditions at nodes from the relative displacements of surrounding nodes was suggested. Acceptable results were obtained using this method but as the technique has not been fully expanded, they are not given here.

6.2 Conclusion

Solutions can be obtained far more rapidly using an explicit stiffness matrix. Such a matrix is therefore highly desirable. The assumed displacement and the assumed stress method do not give suitable explicit stiffness matrices.

By assuming the final form and desired properties of the matrix and solving the values of coefficients from a standard problem, an explicit stiffness matrix can be obtained more easily. This matrix represents the stiffness of an element which is (because of the applied rigid body movements) always in static equilibrium. Because of the regular choice of dimensions and local coordinate axes for the element, its orientation does not change its properties. The results obtained converge towards the correct solution. The element has potential for improvement of performance without making the matrix or its formation more involved. More complex assumed functions could also be introduced and the method could be used on elements with a different shape. Such changes will be discussed in more detail under 'Recommendations'.

The results obtained from the various analyses bring out the following points:-

1. Solutions of certain problems using this element are very satisfactory (Table 4).
2. A large number of elements is required to get acceptable results for other problems.
3. One set of coefficients (Mark III) gives good results on

problems involving pure bending. The other two sets of coefficients give results which lie on opposite sides of the correct results (Table 5).

4. Better results are obtained for problems where edges are clamped and bending does not follow a simple form by using the Mark II coefficients (Table 7).

5. The Mark III coefficients show the anticlastic curvature of a cantilevered plate. This appears to be because the short dimension of the standard plate lies in the principal bending direction.

6. The higher the ratio of the number of degrees of nodal freedom to the number of degrees of nodal fixity on a structure, the more accurate the results seem to be (Table 7).

7. The accuracy of results is not greatly affected by the shape of the plate as shown by the deflection of square and circular clamped plates in Table 7.

8. Results for plates subjected to point loads are less accurate than those for uniformly loaded plates (Table 7). An applied moment on the edge of a rectangular plate is effectively a back substitution of coefficients into the equations from which they were found. The results are therefore exact. (This analysis was used as a check on the development of the matrix.)

As a general rule of the finite element method no one type of element can be described as the "best" element in all applications. (The "best" element, here means one which gives a required accuracy with the smallest number of elements.) If one type of element gives a best result in a particular application, it is because that element resembles the actual displacement situation closely. In a completely different structure that element may not resemble the actual situation at all and a different element might be better.

This applies to the elements developed in this thesis. The advantage of these elements is that a determinate structure, closely resembling the type of structure to be analysed, could be used as the standard structure for finding coefficients. In this case, the finite element developed would be the best for this application.

This principle was used when it was found that the anticlastic curvature effect was not very noticeable in the analysis of a cantilevered plate using Mark II coefficients. By choosing more nodes across the principal bending direction (see figure 21c) the anticlastic curvature effect on the standard structure is emphasised. This is the reason for the Mark III coefficients giving better results in simple bending problems.

It must be emphasised that although an element may not be the best in a particular application, results from it can be obtained to any desired accuracy by increasing the number of elements. There may be a particular set of coefficients which gives best results in most problems. Some methods for obtaining such a set are suggested in the 'Recommendations'.

It may also be found that two different sets of coefficients can be used to give an upper and a lower bound on results. The Mark I and Mark II coefficients formed such a pair in all the problems tested. Two sets of this type would be more useful than a single more accurate set of coefficients.

A direct comparison between the accuracies of rectangular and triangular elements is not justified. This is because only rectangular problems are considered and these are obviously more suited to rectangular elements. A triangular element across the corner of a clamped square plate has all its nodes fully fixed. Its removal from the problem would have no effect. The large perturbations which are caused in the corner of a square plate are therefore not transmitted into the problem. This is not the case with a rectangular element.

6.3 Recommendations

Further work which will be done, based on the success of the assumed stiffness matrix method developed in this thesis, will be along the following lines:-

1. Coefficient improvement

Four possible ways of determining coefficients which will give better results in more cases are as follows:-

(a) A mean can be taken of a number of sets of coefficients determined in the way described in this thesis.

(b) Consider a plate, with many more degrees of freedom than there are unknown independent coefficients, as a standard structure. There will be more equations than unknowns. These may be solved by a regression technique such as is described in Appendix 3.

(c) Let the coordinates of nodes on the standard plate (figure 21) be variables. Set up the equations for the solution of the independent coefficients in terms of the variable coordinates. Set up inequalities giving the limits of variation of the coordinates. Optimise the coordinates so that the set of equations is the most stable possible, i.e. so that the determinant of the set of equations is as close to unity as possible. Once the coordinate values have been optimised, the coefficients can be solved as before.

(d) Different sets of coefficients could be used in different areas of a problem. The coefficients could be derived using standard plates closely resembling the bending in each area of a problem.

(Limited investigations have already been made into the methods (a) and (b) above, with little success as yet.)

2. <u>Different assumed functions</u>:

More complex assumed polynomials can be used by choosing a higher root length degree (than three, which was used here) in the identity formation and solution program. This will increase the size and formation time of the explicit stiffness matrix for the same shape of element. Added accuracy may justify this.

Different forms of the cyclic common denominator P (equation (79)) could also be considered, e.g.

$$P = \alpha^2 \beta^2 \gamma^2 \Delta$$

3. <u>More complex shapes</u>:

The assumed stiffness method can be extended to any polygonal element shape. Difficulty may be experienced in finding rigid body rotation vectors and in choosing unbiased element dimensions. An attempt to develop such an element emphasises the appealing simplicity of the triangular shape over all other shapes of element.

4. <u>Other types of element</u>:

Elements for handling different types of stress situations in any number of dimensions (e.g. two dimensional plane strain elements) can be derived using the assumed stiffness method. Rigid body movement vectors will possibly be described in terms of different local coordinate axes.

Orthotropic elements would require a greater number of description parameters. One parameter would define the direction of the major stiffness with respect to the axes. Others would describe the various elastic properties. The final matrix would be far more complex than the isotropic element stiffness matrix. More elements would be required in order to determine the values of the coefficients.

Other problems which arise out of this thesis and which would bear further investigation are:

(1) The method of determining stresses mentioned in section 4.3 could be improved for nodes along the edges of a problem.

(2) A Gauss solution using the half band width of a symmetric banded matrix and dividing a problem into sub-matrices would allow large systems of elements to be solved on small computers. This is because finite element system stiffness matrices form tri-diagonal matrices which can be divided into diagonal sub-matrices. For example the element conformation shown in figure 22 gives a matrix of the form:

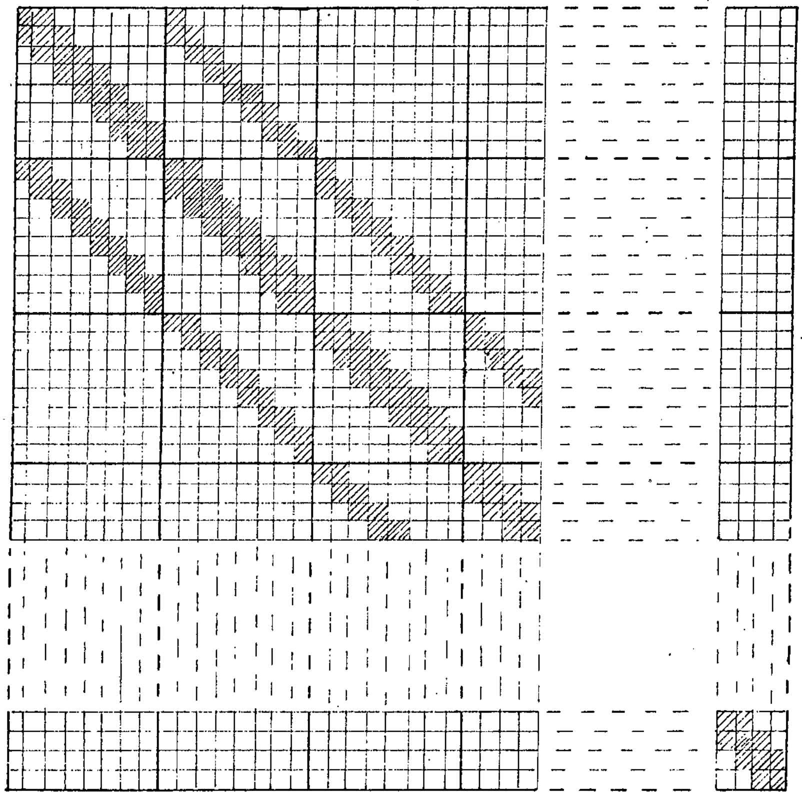


Figure 24: A Tri-Diagonal Matrix

The leading diagonal submatrices being symmetrical can be stored as an oblique half band width - see Appendix 1. The full band of the off-diagonal matrix must be stored. An inversion of the matrix (described in Appendix 1) requires only two submatrix bands and part of the load vector in core simultaneously. The rest of the matrix can be retained in a disc file.

(3) The convergence criteria for the elements developed should be investigated. A comparison between the accuracy of various solutions and some ratio representing the degree of fixity of the system of elements would be informative. Such a ratio might be the number of fixed nodal movements (due to boundary conditions) divided by the total number of nodal degrees of freedom for the system.

REFERENCES

1. PRZEMIENIECKI, J.S. Theory of matrix structural analyses. New York, McGraw-Hill, 1968

2. PIAN, T.H.H. Derivation of element stiffness matrices by assumed stress distributions. J. Am. Inst. Aero. Astro., 1964 vol. 2 (JULY), pp. 1333-1336.

3. SEVERN, R.T. and TAYLOR, P.R. The finite element method for flexure of slabs when stress distributions are assumed. Proc. Instn. Civ. Engrs., vol. 34, pp. 153-170 (1966).

4. ALLWOOD, R.J. and CORNES, C.M.M. A polygonal finite element for plate bending problems using the assumed stress approach. Internat. J. for Numerical Methods in Engineering. vol. 1, No. 2, 1969 (APR-JUNE).

5. ZIENKIEWICZ, O.C. The finite element method in engineering science. London, McGraw-Hill, 1971.

6. CLOUGH, R.W. and TOCHER, J.L. Finite element stiffness matrices for analyses of plate bending. Matrix Methods in Structural Mechanics Proceedings of the Conference held at Wright-Patterson Air Force Base, Ohio, Oct. 1965.

7. GELLERT, M. and GLUCK, J. Elastic plate analysis by Euler's variational method. Build. Sci. vol.7, pp. 145-150. Pergamon Press 1972.

8. SOCOLNIKOFF, I.S. Mathematical theory of elasticity. New York, McGraw-Hill, 1956.

9. HEARMON, R.F.S. and ADAMS, E.H. The bending and twisting of anisotropic plates. Brit. J. Appl. Phys., vol. 3, May 1952, pp. 140-156.

10. POPOV, E.P. Mechanics of materials. Englewood Cliffs, Prentice-Hall, 1952.

11. DEAK, A.L. and PIAN, T.H.H. Application of the smooth surface interpolation to the finite element analysis. J. Am. Inst. Aero. Astro., 1967, vol. 5, No. 1 (JANUARY), pp. 187-189.

12. ZIENKIEWICZ, O.C. and CHEUNG, Y.K. The finite element method for analysis of elastic isotropic and orthotropic slabs. Proc. Instn. Civ. Engrs., vol. 28, pp 471-488 (1964).

13. TIMOSHENKO, S. and WOINOWSKY-KRIEGER, S. Theory of plates and shells. New York, McGraw-Hill, 1959.

14. ROGERS, A. Matrix methods in urban and regional analysis. San Francisco, Holden-Day, 1971.

APPENDIX I

The Gauss Jordan Reduction Technique

Supposing an augmented matrix [A] is to be reduced using this method.

The first column of the matrix [A] is scanned for the element with the greatest absolute magnitude. (This is part of a process called pivoting which reduces the rounding error in computer routines. Any non-zero element of the column would suffice theoretically). If this value is zero, the whole column consists of zero elements and the next column is considered.

If the maximum absolute value of the element in the first column is non-zero, $|a1|$ say, then the whole row of the matrix [A] which contains this maximum, is divided by a1. This produces a unit value in the first column of the row. This row will now be referred to as the reducing row of column one.

Multiples of this row reduce the remaining values in the first column to zero by subtraction.

The same process is repeated for each column. A <u>different</u> reducing row must be chosen each time. The process is terminated when either all the rows of matrix [A] have been used as reducing rows or all the columns have been reduced.

There are three main applications of the Gauss Jordan reduction technique:

(i) To solve equations of the following form (e.g. to find the coefficients in an equation given the coordinates of a number of points satisfying it).

$$[X].[C] = [Y]$$

where [X] is a square non-singular set of parameters multiplying the unknown variables represented in columns of the matrix [C]

[Y] is a number of column vectors of constant values, each column representing an independent set of equations.

[C] is a set of solution vectors corresponding to the columns of [Y].

(ii) To find the inverse of a square non-singular matrix [X].

(iii) Where fewer independent equations exist than are necessary for a full unique solution, to express as many variables as there are independent equations, as linear combinations of the remaining variables and constants.

In the first application, the augmented matrix is formed by writing the matrix [X] and the constant vectors [Y], side by side:

$$[A] = \Big[[X]\ [Y]\Big]$$

At some stage of the routine, the reducing rows must be rearranged so that a unit matrix is formed in the position of [X].

If at any stage a full column of zeroes is formed then the matrix is singular. A program must give some warning of this.

The solutions [C] of the equations remain in the position occupied by [Y] in the augmented matrix. Answers are in the same order (top to bottom) as the unknowns were (left to right) in the original equations.

In the inversion process (ii) above, the augmented matrix is formed as before with an identity matrix [I] replacing the matrix [Y]. The ordering process and singularity check are still necessary. The inverted matrix $[X]^{-1}$ results in the position which was originally occupied by the matrix [Y].

In the third application of a reduction, all values of equations are moved to the left hand side of the equality leaving zeroes on the right hand side. The left hand side then forms the augmented matrix. No singularity check or ordering is necessary. At the end of the process, equations are still equal to zero. All but the first non-zero term in each row is removed to the right hand side of the equation, changing the sign in the usual manner. As the first term is a unit value, this expresses the corresponding variables as a linear combination of the other variables. Rows of zeroes are ignored.

<u>Example</u>: Reduce the following equations to their simplest form.

$$\begin{aligned} x_1 + x_2 + x_3 &= 3 \\ x_1 + x_2 + 2x_3 &= 4 \\ -2x_1 - 2x_2 - 2x_3 &= -6 \\ 3x_3 &= 3 \end{aligned}$$

Augmented matrix:

$$\left[\begin{array}{ccc|c} 1 & 1 & 1 & -3 \\ 1 & 1 & 2 & -4 \\ -2 & -2 & -2 & 6 \\ 0 & 0 & 3 & -3 \end{array}\right]$$

Steps of the reduction:

$$\left[\begin{array}{ccc|c} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 1 & 1 & 1 & -3 \\ 0 & 0 & 3 & -3 \end{array}\right] \quad \text{Reducing row.}$$

$$\left[\begin{array}{ccc|c} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & -2 \\ 0 & 0 & 1 & -1 \end{array}\right] \quad \text{Reducing row.}$$

Interpretation:

$$\begin{aligned} x_1 &= -x_2 + 2 \\ x_3 &= 1 \end{aligned}$$

If at the conclusion of any of the above reductions there is a statement that zero is equal to some definitely non-zero quantity, then a solution of an inconsistent set of equations has been attempted.

Solution of Banded Symmetrical Matrices

In structural analysis, large symmetrical banded matrices are common. It is possible to reduce a banded symmetric matrix operating only within half of the band width. It is only necessary to fit half of the band width plus the leading diagonal terms together with the constant matrix [Y] into a computer core in this case.

The process is as follows:-

No alteration in the order of rows is allowed as this disturbs the symmetry of the matrix. As the matrix is banded, it is unlikely that zero values occur on the leading diagonal. (In structural analysis this should never happen.)

The first row is the reducing row for column one and the second row for column two, etc. The elements of the reducing row are a record of the elements of the column to be reduced. This must not be upset by dividing the row through by its first term.

The ratio between the 'n'th element and the first element of the reducing row of column 'm' is the number by which the row must be multiplied in order to eliminate the 'n'th element of column 'm'.

Only the upper half of the band is stored. The whole of the lower (invisible) half of the band is first eliminated before the upper half is reduced. This is so that symmetry of remaining invisible terms is maintained. The process is best illustrated by example.

Example: ← half band width

$$\begin{bmatrix} 2 & 4 & 6 & 0 & 0 \\ 4 & 10 & 16 & 6 & 0 \\ 6 & 16 & 28 & 16 & 6 \\ 0 & 6 & 16 & 28 & 16 \\ 0 & 0 & 6 & 16 & 28 \end{bmatrix} \times \begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \end{bmatrix} = \begin{bmatrix} 12 \\ 36 \\ 72 \\ 66 \\ 50 \end{bmatrix}$$

This matrix has been designed so that arithmetic calculations are repeated with each row. This simplifies the example but the principle applies to any banded symmetrical matrix. Only half the band width and the constant vector need be stored. This is done in a two dimensional array. In the above example the upper half band width would be formed in the array as follows:

$$\left[\begin{array}{ccc|c} 2 & 4 & 6 & 12 \\ 10 & 16 & 6 & 36 \\ 28 & 16 & 6 & 72 \\ 28 & 16 & 0 & 66 \\ 28 & 0 & 0 & 50 \end{array}\right]$$

The rows of the array are the same as the rows of the original matrices. The columns of the coefficient matrix run obliquely in the array and are operated upon correspondingly. The constant vector appears conventionally in the array.

As will be seen in the solution to the present example, the top half band width represents the mirror image of the remaining part of the lower half band width after each column reduction. After the whole of the "invisible" lower half band width has been eliminated, the top "visible" half band width is reduced from the bottom row upwards. (It is in fact "wasted operation" to actually eliminate the upper half band width. A back substitution altering only the values in the constant vector is less time consuming.)

The elimination proceeds as follows:-

ACTUAL ELIMINATION

$$\left[\begin{array}{ccccc|c} 2 & 4 & 6 & 0 & 0 & 12 \\ 4 & 10 & 16 & 6 & 0 & 36 \\ 6 & 16 & 28 & 16 & 6 & 72 \\ 0 & 6 & 16 & 28 & 16 & 66 \\ 0 & 0 & 6 & 16 & 28 & 50 \end{array}\right]$$

COMPUTER IMAGE

$$\left[\begin{array}{ccc|c} 2 & 4 & 6 & 12 \\ 10 & 16 & 6 & 36 \\ 28 & 16 & 6 & 72 \\ 28 & 16 & 0 & 66 \\ 28 & 0 & 0 & 50 \end{array}\right]$$

First column elimination:

ACTUAL ELIMINATION | COMPUTER IMAGE

$$\left[\begin{array}{ccccc|c} 2 & 4 & 6 & 0 & 0 & 12 \\ 0 & 2 & 4 & 6 & 0 & 12 \\ 0 & 4 & 10 & 16 & 6 & 36 \\ 0 & 6 & 16 & 28 & 16 & 66 \\ 0 & 0 & 6 & 16 & 28 & 50 \end{array}\right] \qquad \left[\begin{array}{ccc|c} 2 & 4 & 6 & 12 \\ 2 & 4 & 6 & 12 \\ 10 & 16 & 6 & 36 \\ 28 & 16 & 0 & 66 \\ 28 & 0 & 0 & 50 \end{array}\right]$$

As was arranged, the remainder of the eliminations of the lower triagle follow the same operations so that at the end:

$$\left[\begin{array}{ccccc|c} 2 & 4 & 6 & 0 & 0 & 12 \\ 0 & 2 & 4 & 6 & 0 & 12 \\ 0 & 0 & 2 & 4 & 6 & 12 \\ 0 & 0 & 0 & 2 & 4 & 6 \\ 0 & 0 & 0 & 0 & 2 & 2 \end{array}\right] \qquad \left[\begin{array}{ccc|c} 2 & 4 & 6 & 12 \\ 2 & 4 & 6 & 12 \\ 2 & 4 & 6 & 12 \\ 2 & 4 & 0 & 6 \\ 2 & 0 & 0 & 2 \end{array}\right]$$

The back substitution is a simple process. The first step is as follows:-

$$\left[\begin{array}{ccccc|c} 2 & 4 & 6 & 0 & 0 & 12 \\ 0 & 2 & 4 & 6 & 0 & 12 \\ 0 & 0 & 2 & 4 & 0 & 6 \\ 0 & 0 & 0 & 2 & 0 & 2 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{array}\right] \qquad \left[\begin{array}{ccc|c} 2 & 4 & 6 & 12 \\ 2 & 4 & 6 & 12 \\ 2 & 4 & 6 & 6 \\ 2 & 4 & 0 & 2 \\ 2 & 0 & 0 & 1 \end{array}\right]$$

Note that no attempt was made to alter the values in the coefficient array as this wastes time in the computer. The next substitutions follow the same procedure and the final situation is:

$$\left[\begin{array}{ccccc|c} 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{array}\right] \qquad \left[\begin{array}{ccc|c} 2 & 4 & 6 & 1 \\ 2 & 4 & 6 & 1 \\ 2 & 4 & 6 & 1 \\ 2 & 4 & 0 & 1 \\ 2 & 0 & 0 & 1 \end{array}\right]$$

The constant vector column now contains the solutions of the variables in the order given in the original equations.

Simplification of a Randomly Sparse Matrix

The computer application of the reduction (iii) given above is also a special case. When equations are simply to be reduced without a full solution being expected the equations often have the following characteristics:-

(a) The coefficient matrix is sparse but randomly so, (i.e. not banded).

(b) There are more equations than unknowns, but many of them are linear combinations of others, leaving fewer linearly independent equations than unknowns. The set of equations is said to be under-determinate.

(c) The full matrices are very large.

The special technique for fitting the augmented matrix into core is as follows:-

Three one dimensional arrays store the non-zero elements of the whole augmented matrix as follows:

One array, IA(I), stores the column number of each non-zero element in succession, taken row by row. A second array, A(I), which corresponds exactly to the array IA(I), stores the values of each non-zero element in succession row by row. A third, "book keeping", array, JA(J), stores the position, I, in the first two arrays where the first non-zero element of the Jth row can be found.

In this way, reading the values of A(I) and IA(I) for values of I going from JA(J) to (JA(J + 1) - 1), the Jth equation can be recognised.

Routines are then written to:

(a) Organise the arrays IA(I) and correspondingly A(I) so that column numbers of one row run in ascending order.

(b) Shunt the values in all three matrices so that parts of equations can be erased by overwriting or so that spaces may be made in equations for adding extra terms.

(c) Remove positions of IA(I) and A(I) where the value of A(I) has become zero during manipulation. This is a simple application of (b) above.

(d) To sum terms of the array A(I) and delete one position of A(I) and IA(I) when column numbers in IA(I) are duplicated in the space of one row.

The procedure for reduction follows the mathematical algorithm described in general terms at the beginning of this Appendix. The pivoting process scan for the largest coefficient examines only the first value of IA(I) in each row. If this value is less than the number of the column being scanned then the row has already been used as a reducing row. If this value is greater than the number of the column being scanned then the row has a zero coefficient in the column being scanned.

The reduction process is performed by making a space (as described in (b)) large enough to fit the reducing row into the row being reduced. The reducing row is then multiplied by the correct factor and written into the space. Duplication of row numbers is removed by addition (as described in (d)). Zero values in A(I) are removed (as described in (c)) and the column numbers of the row just reduced are ordered (as described in (a)).

<u>Example</u>:

The following augmented matrix is to be reduced to its simplest form:

$$\begin{bmatrix} 0 & 0 & 1 & 2 \\ 1 & 0 & 2 & 0 \\ 1 & 3 & 0 & 0 \\ 1 & 0 & 3 & 2 \\ 2 & 6 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -2 & -4 \end{bmatrix}$$

This is represented as:

I	1	2	3	4	5	6	7	8	9	10	11	12	13
IA(I)	3	4	1	3	1	2	1	3	4	1	2	3	4
A(I)	1.	2.	1.	2.	1.	3.	1.	3.	2.	2.	6.	-2.	-4.
J	1	2	3	4	5	6	7	(8)					
JA(J)	1	3	5	7	10	12	12	(14)					

Note that if two successive values of JA are the same then the first represents a row of zeroes and is ignored. The values of IA(1), IA(3), IA(5) ...... IA(12) are checked for the value one. The largest corresponding value of A(I) is A(10). Row five is thus used as the reducing equation of column one. The only changes to the arrays are:

A(10) becomes 1.
A(11) becomes 3.

A space for this row multiplied by minus one is then opened up between I = 4 and I = 5. The new situation is:

I	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
IA(I)	3	4	1	3	1	2	1	2	1	3	4	1	2	3	4
A(I)	1.	2.	1.	2.	-1.	-3.	1.	3.	1.	3.	2.	1.	3.	-2.	-4.
J	1	2	3	4	5	6	7	(8)							
JA(J)	1	3	7	9	12	14	14	(16)							

The values of coefficients in like columns in row two are summed and zero values removed. Row two is ordered leaving:

I	1	2	3	4	5	6	7	8	9	10	11	12	13
IA(I)	3	4	2	3	1	2	1	3	4	1	2	3	4
A(I)	1.	2.	-3.	2.	1.	3.	1.	3.	2.	1.	3.	-2.	-4.
J	1	2	3	4	5	6	7	(8)					
JA(J)	1	3	5	7	10	12	12	(14)					

The process continues along the same lines. The final situation is:

I	1	2	3	4	5	6		
IA(I)	2	4	1	4	3	4		
A(I)	1.	1.33	1.	-4.	1.	2.		
J	1	2	3	4	5	6	7	(8)
JA(J)	1	1	3	3	3	5	5	(7)

The interpretation of this is:

$$x_2 + 1{,}33\ x_4 = 0 \qquad\qquad x_1 = 4{,}0\ x_4$$

$$x_1 - 4{,}0\ x_4 = 0 \quad \text{or} \quad x_2 = -\ 1{,}33\ x_4$$

$$x_3 + 2{,}0\ x_4 = 0 \qquad\qquad x_3 = 2{,}0\ x_4$$

The following subroutine uses this computer core space-saving technique. The identity solution program (Appendix 6) calls this subroutine:

(See following pages)

S U B R O U T I N E   F O R   D O I N G   A   G A U S S   R E D U C T I O N
O N   A   R A N D O M L Y   S P A R S E   M A T R I X .

```
      SUBROUTINE HEQSN
      COMMON A(15000),IA(15000),JA(1000),N,IST2,M,KB
O R D E R   E Q U A T I O N S  ,  R E M O V E   Z E R O S   &   D U P L I C A -
T I O N
      MIN=1
      DO 12 LB=1,N
      MAX=JA(LB)
      IF(MAX-MIN)12
      MAZ=MIN
      CALL HDUBL(MIN,MAX,MAZ)
      CALL HZERO(MIN,MAX,MAZ)
      CALL HORDER(MIN,MAX,MAZ)
12    MIN=MAX+1
C Y C L E   F I N D I N G   R E D U C T I O N   E Q U A T I O N   O F
C O L U M N   'LA'
      DO 10 LA=1,M
      NA=1
      C=0
      NB=0
S C A N   E Q U A T I O N S   F O R   M A X I M U M   P I V O T
      DO 4 LB=1,N
      IF(JA(LB)-NA)4
      IF(IA(NA)-LA),13
4     NA=JA(LB)+1
      GO TO 10
13    NB=LB
      C=A(NA)
      NC=1
      IF(NB.GT.1)NC=JA(NB-1)+1
      ND=JA(NB)
P R O D U C E   U N I T   1 S T   C O E F F I C I E N T   I N   R E D U C I N G
E Q U A T I O N   'NB'
      DO 5 LB=NC,ND
5     A(LB)=A(LB)/C
      NA=1
R E D U C T I O N   C Y C L E   O F   C O L U M N   'LA'
      DO 9 LB=1,N
      NE=JA(LB)
      IF(LB-NB),9
      IF(NE-NA)9
C H E C K   E X I S T E N C E   O F   V A R I A B L E   'LA'   I N
E Q U A T I O N   'LB'
      DO 6 LC=NA,NE
      IF(IA(LC)-LA)6,,6
      C=A(LC)
      GO TO 7
```

(continued)

```
6       CONTINUE
        GO TO 9
7       LR=ND-NC+1
        LEAD=NE+1
        MAZ=LEAD
I N S E R T   R E D U C I N G   E Q U A T I O N   I N   E Q U A T I O N   'LB'
        CALL HSHIFT(LEAD,LR,MAZ)
        DO 8 LC=NC,ND
        IA(NE+1+LC-NC)=IA(LC)
8       A(NE+1+LC-NC)=-A(LC)*C
        MIN=NA
        MAX=NE+ND-NC+1
        MAZ=NE+1
R E M O V E   D U P L I C A T I O N ,   Z E R O S   &   R E O R D E R
        CALL HDOUBL(MIN,MAX,MAZ)
        CALL HZERO(MIN,MAX,MAZ)
        CALL HORDER(MIN,MAX,MAZ)
9       NA=JA(LB)+1
10      CONTINUE
D E L E T I O N   O F   R E C O R D   O F   E L I M I N A T E D   E Q U A T N S
        I=1
15      IF(N-I)20
        J=I+1
16      IF(N-J)19
        IF(JA(I)-JA(J))18
        NF=N-1
        IF(NF-J)18
        DO 17 K=J,NF
17      JA(K)=JA(K+1)
        N=NF
        GO TO 16
18      J=J+1
        GO TO 16
19      I=I+1
        GO TO 15
20      CONTINUE
O U T P U T   O N L Y   T H E   L A S T   S O L U T I O N
        IF(KB.LE.12)GO TO 21
        CALL HWRITE
21      RETURN
```

S U B R O U T I N E   F O R   O R D E R I N G   V A R I A B L E S   I N
E A C H   E Q U A T I O N

```
      SUBROUTINE HORDER(MIN,MAX,MAZ)
      COMMON A(15000),IA(15000),JA(1000),N,IST2,M
      IF(MAX-MAZ)2
      MAY=MAX-1
      IF(MAY-MIN)2
      DO 1 I=MIN,MAY
      J=I+1
      IF(MAZ.GT.J)J=MAZ
      DO 1 K=J,MAX
      IF(IA(I)-IA(K))1,1
      KEEP=IA(I)
      IA(I)=IA(K)
      IA(K)=KEEP
      STORE=A(I)
      A(I)=A(K)
      A(K)=STORE
1     CONTINUE
2     RETURN
```

S U B R O U T I N E   R E M O V E S   D U P L I C A T I O N   I N   E A C H
E Q U A T I O N   B Y   A D D I T I O N

```
      SUBROUTINE HDOUBL(MIN,MAX,MAZ)
      COMMON A(15000),IA(15000),JA(1000),N,IST2,M
      I=MIN
1     J=I+1
      IF(J.LT.MAZ)J=MAZ
2     IF(MAX-J)4
      IF(IA(I)-IA(J))3,,3
      A(I)=A(I)+A(J)
      LEAD=J+1
      LR=-1
      CALL HSHIFT(LEAD,LR,MAZ)
      GO TO 2
3     J=J+1
      GO TO 2
4     I=I+1
      IF(I-MAX)1
      RETURN
```

SUBROUTINE REMOVES VARIABLES WITH ZERO COEFFICIENTS

```
      SUBROUTINE HZERO(MIN,MAX,MAZ)
      COMMON A(15000),IA(15000),JA(1000),N,IST2,M
      ACC=0.001
      I=MIN
1     IF(MAX-I)3
      ABB=ABS(A(I))
      IF(ACC-ABB)2
      LEAD=I+1
      LR=-1
      CALL HSHIFT(LEAD,LR,MAZ)
      GO TO 1
2     I=I+1
      GO TO 1
3     RETURN
```

SUBROUTINE SHIFTS EQUATIONS IN THE ARRAY

```
      SUBROUTINE HSHIFT(LEAD,LR,MAZ)
      COMMON A(15000),IA(15000),JA(1000),N,IST2,M
      IF(JA(N)-LEAD)5
      I1=LEAD
      I2=JA(N)-LEAD+1
      I4=1
      IF(LR)1,4
      I1=JA(N)
      I4=-1
1     DO 2 I5=1,I2
      I3=(I5-1)*I4+I1
      A(I3+LR)=A(I3)
2     IA(I3+LR)=IA(I3)
5     DO 3 I3=1,N
      IF(JA(I3)-LEAD+1)3
      JA(I3)=JA(I3)+LR
3     CONTINUE
      IF(JA(N).GT.IST2)IST2=JA(N)
4     MAX=MAX+LR
      IF(MAZ.GE.LEAD)MAZ=MAZ+LR
      IF(NC.GE.LEAD)NC=NC+LR
      IF(ND.GE.LEAD)ND=ND+LR
      RETURN
```

O U T P U T   S U B R O U T I N E

```
      SUBROUTINE HWRITE
      MA=1
      DO 14 I=1,N
      MB=JA(I)
      IF(MB-MA)14
      WRITE(5,102)
      WRITE(5,103)(A(J),IA(J),J=MA,MB)
102   FORMAT(1H ,)
103   FORMAT(1H ,10(F5.1,1X,'X(',I3,')'))
14    MA=MB+1
      RETURN
      END
```

Solution of a Set of Submatrix Equations

A Gauss Jordan elimination is possible if the elements of the augmented matrix are themselves submatrices. The division by a submatrix is represented by a premultiplication by the inverse of the submatrix. The check that the dividing value is non-zero has a matrix equivalent in checking that the premultiplying matrix is non-singular.

Again an example is used by way of description.

Example:

Solve the following augmented matrix of submatrices:
(This type of matrix often occurs in structural analysis, particularly finite element analysis. The submatrices are often diagonal matrices themselves which can be operated upon in a similar way to the symmetrical banded matrix above - see Recommendations).

$$\left[\begin{array}{ccc|c} [B_{11}] & [A_{12}] & [\ 0\ ] & \{D_1\} \\ [A_{21}] & [A_{22}] & [A_{23}] & \{C_2\} \\ [\ 0\ ] & [A_{32}] & [A_{33}] & \{C_3\} \end{array}\right]$$

Only the submatrices on the leading diagonal are of necessity square and non-singular. A premultiplication of the first row by $[B_{11}]^{-1}$ is the first step. The resulting row is then premultiplied by $[A_{21}]$ and subtracted from the second row. The result follows.

$$\left[\begin{array}{c|c|c|c} [I] & [B_{11}]^{-1}\,[A_{12}] & [\ 0\ ] & [B_{11}]^{-1}\,\{D_1\} \\ [0] & [B_{22}] & [A_{23}] & \{D_2\} \\ [0] & [A_{32}] & [A_{33}] & \{C_3\} \end{array}\right]$$

where:

$$[B_{22}] = [A_{22}] - [A_{21}]\,[B_{11}]^{-1}\,[A_{12}]$$

$$\{D_2\} = \{C_2\} - [A_{21}]\,[B_{11}]^{-1}\,\{D_1\}$$

The next step is similar (and if the matrix had any number of similar rows and columns, the process would continue identically and could be programmed very neatly):

$$\left[\begin{array}{c:c:c:c} [I] & [B_{11}]^{-1}\ [A_{12}] & [0] & [B_{11}]^{-1}\ \{D_1\} \\ [0] & [I] & [B_{22}]^{-1}\ [A_{23}] & [B_{22}]^{-1}\ \{D_2\} \\ [0] & [0] & [B_{33}] & \{D_3\} \end{array}\right]$$

where

$$[B_{33}] = [A_{33}] - [A_{32}]\ [B_{22}]^{-1}\ [A_{23}]$$

$$\{D_3\} = \{C_3\} - [A_{32}]\ [B_{22}]^{-1}\ \{D_2\}$$

Back substitution of the matrices gives:

$$\left[\begin{array}{c:c:c:c} [I] & [0] & [0] & \{E_1\} \\ [0] & [I] & [0] & \{E_2\} \\ [0] & [0] & [I] & \{E_3\} \end{array}\right]$$

where

$$\{E_3\} = [B_{33}]^{-1}\ \{D_3\}$$

$$\{E_2\} = [B_{22}]^{-1}\ (\{D_2\} - [A_{23}]\ \{E_3\})$$

$$\{E_1\} = [B_{11}]^{-1}\ (\{D_1\} - [A_{12}]\ \{E_2\})$$

Often solution and inversions of matrices are done in two levels. The matrix is first inverted as a matrix of submatrices. The submatrices requiring inversion are then inverted by a further Gauss Jordan reduction.

The use of submatrix inversion allows parts of matrices to be left in disc storage because only two submatrices are required in core simultaneously.

APPENDIX II

FORMULAE FOR INTEGRATION OVER AN AREA

Usually the integration of functions over the area of a triangle is done numerically. A Simpson's rule integration in two dimensions or a Gauss integration method may be used. The latter involves sampling at certain specified positions according to a table calculated from Gauss theory for smooth curves. A greater accuracy of result is obtained for the same number of sampling positions as the Simpson method.

To find algebraic integrals of various functions over the area of any rectilinear figure, the following procedure is adopted.

Integration is performed over the areas between one of the coordinate axes and each side of the polygon. If the lower limit of the integration is the lower value with respect to the coordinate axis in question, the resulting integrals will be positive. If the sides of the figure are considered sequentially in a clockwise order, the algebraic sum of the integrals will automatically result in an integration over the area of the figure. (See figure 25).

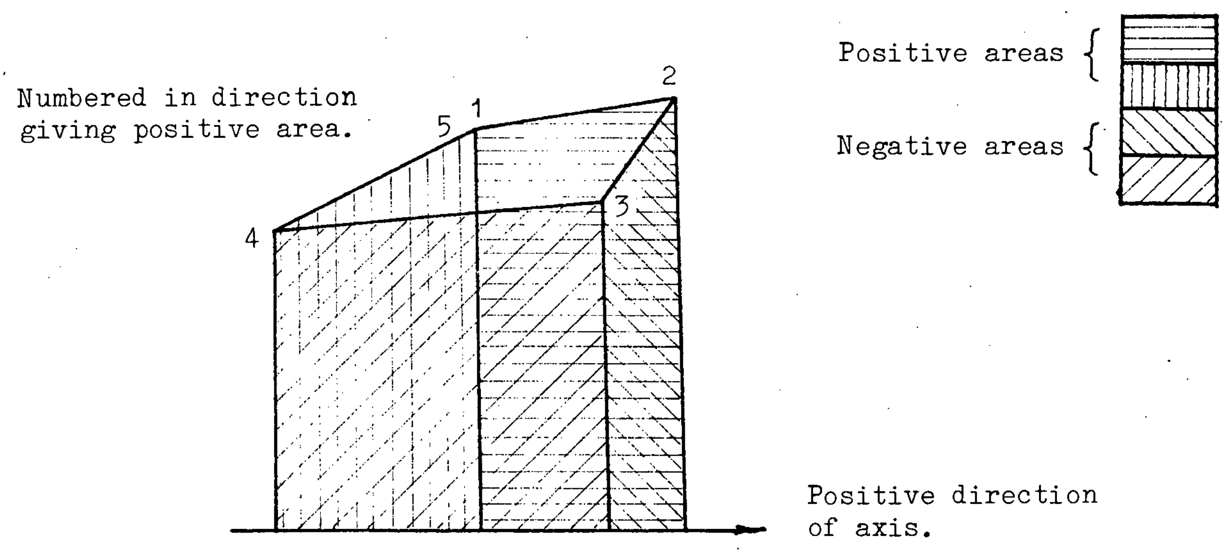


Figure 25: Area enclosed by a rectilinear figure

The integration between one side and an axis is performed as follows:-

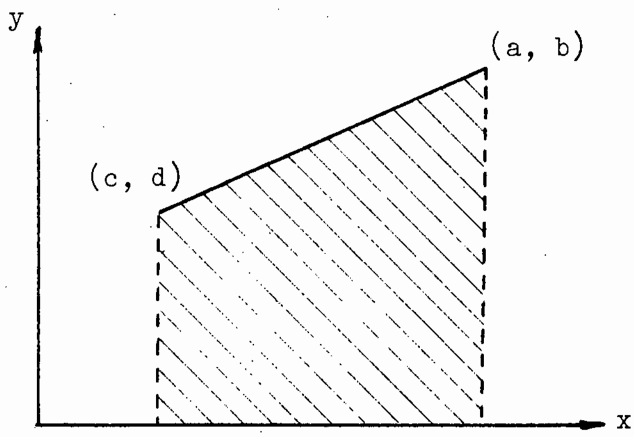


Figure 26: Area between a general line and the x-axis

In figure 26 the equation of the line is:

$$y = \frac{bx - dx + ad - bc}{a - c}$$

$$\therefore \iint\limits_{AREA} f(x,y)\ d\ AREA = \int_{c}^{a} \int_{0}^{\frac{bx - dx + ad - bc}{a - c}} f(x,y)\ dydx$$

For example:

$$\iint\limits_{AREA} y^2\ d\ AREA = \int_{c}^{a} \int_{0}^{\frac{bx - dx + ad - bc}{a - c}} y^2 dydx$$

$$= \frac{1}{3} \int_{c}^{a} \left(\frac{bx - dx + ad - bc}{a - c}\right)^3 dx$$

$$= \frac{(ba - da + ad - bc)^4 - (bc - dc + ad - bc)^4}{12(a - c)^3\ (b - d)}$$

$$= (b^3a + b^2da + bd^2a + d^3a - b^3c - b^2dc - bd^2c - d^3c)/12$$

Similarly:

$$\iint_{AREA} x^2 dA = -(a^3b + a^2cb + ac^2b + c^3b - a^3d - a^2cd - ac^2d - c^3d)/12$$

$$\iint_{AREA} xy dA = (3a^2b^2 + 2a^2bd + a^2d^2 - 2ab^2c + 2acd^2 - c^2b^2 - 2bc^2d - 3c^2d^2)/24$$

$$\iint_{AREA} y\, dA = (ab^2 + abd + ad^2 - cb^2 - cbd - cd^2)/6$$

$$\iint_{AREA} x\, dA = -(ba^2 + bac + bc^2 - da^2 - dac - dc^2)/6$$

$$\iint_{AREA} 1\, dA = (ab + ad - cb - cd)/2$$

APPENDIX III

AN APPLICATION OF REGRESSION ANALYSIS

A regression is the mathematical curve which fits a set of statistical data best. The criteria of "fit" may vary. The most common type of regression "fit" is that known as least squares. In this, the regression minimises the sum of the squares of the ordinate differences between actual sample values and corresponding values calculated from the regression. References to the least squares technique of curve fitting are numerous (e.g. Rogers[14]).

Supposing a parabolic curve is to be fitted to four points (x_1, y_1), (x_2, y_2), (x_3, y_3) and (x_4, y_4). A set of four equations in the three unknown constants of the parabola could be set up:

$$C_1 + C_2x_1 + C_3x_1^2 = y_1$$
$$C_1 + C_2x_2 + C_3x_2^2 = y_2$$
$$C_1 + C_2x_3 + C_3x_3^2 = y_3$$
$$C_1 + C_2x_4 + C_3x_4^2 = y_4$$

or $$[X]\,\{C\} = \{Y\} \qquad (85)$$

The least squares fit to this equation would be

$$\{C\} = \left[[X]^t . [X]\right]^{-1} . [X]\,\{Y\}$$

This is derived by premultiplying both sides of equation (85) by the transpose of the matrix $[X]$. The coefficient matrix of the vector $\{C\}$ is therefore square (and symmetric). If a minimum of three of the points (x_1, y_1) to (x_4, y_4) are different from one another so that there are at least as many linearly independent rows of matrix $[X]$ as there are columns, the product:

$$[X]^t\,[X]$$

is non-singular and can be inverted.

In general, the proposed expression which is to fit a set of values is chosen to have a number of arbitrary coefficients. The dependent variables form the constant vector $\{Y\}$. Substitution of the independent variables into the proposed function forms the matrix $[X]$. An augmented matrix is formed as:

$$\left[[X], \ \{Y\} \right]$$

This is premultiplied by the transpose of the coefficient matrix (i.e. $[X]^t$). The solution $\{C\}$ can then be found by the Gauss Jordan method (Appendix 1).

Surfaces in any number of variables can be fitted in this manner. Equivalent operations allow one to determine the standard deviations, goodness of fit and other statistical information. These will not be described.

The above method is used in this thesis to solve a set of relationships, where there are more equations than unknown values.

APPENDIX IV

COMPUTER ALGEBRA

Two techniques for doing simple algebraic operations using FORTRAN (which essentially handles numerical operations) were developed. One method was used with each of the explicit stiffness matrices of Chapter II. The second method, algebra by powers, was found to be more successful.

Both methods handle additions, subtractions and multiplications very simply. Both could be used for division provided the devisor was a factor of the dividend. The handling of remainders is difficult.

In the present application, only long polynomial expressions in a number of variables are to be manipulated. Division is done by hand calculation, by separate calculation of numerator and denominator or by redefinition of quotients as a new independent variable. The computer routines are designed to do only addition and multiplication, simple operations which could be done by hand but are programmed because of the quantity and the probability of human error.

Algebra by Prime Numbers

A polynomial term such as a^2bc^3 is to be expressed as a number which after manipulation can still be recognised as its original variables. One way of doing this is to give each variable a prime number value other than one;

e.g. put $a = 2$; $b = 3$; $c = 5$.

The value of a^2bc^3 is then 1500. This can be factorised by a routine which will be described later and identified in terms of a, b and c.

A whole polynomial expression in the above variables could then be recorded in two one-dimensional arrays, corresponding one to one. One array would be of real variables say COEF(I) and the other of integer variables say NTERM(I). The array COEF(I) would contain the coefficients (with algebraic sign) of terms and the array NTERM(I) would contain a corresponding integer number (such as described above) for the variables of each term;

e.g. put $a = 2$; $b = 3$; $c = 5$.

Then the expression $3a^2 + 4abc - 2bc^2$ would be represented by the arrays as:

COEF(1)	=	3,0;	COEF(2)	=	4,0;	COEF(3)	=	- 2,0
NTERM(1)	=	4 ;	NTERM(2)	=	30;	NTERM(3)	=	75

<u>Addition</u>: Two terms can be added by writing them side by side in an array. If their values in the array NTERM are the same, their coefficients are added and the result overwrites one coefficient. The other term is deleted. The addition of two expressions is illustrated in the following example.

e.g. Add the following two expressions:

$3a^2 - 2ab$

and $4ab + b^2 - ca$

Represented as arrays:

I	=	1	2	3
COEF1(I)	=	3,0	- 2,0	
NTERM1(I)	=	4	6	
COEF2(I)	=	4,0	1,0	- 1,0
NTERM2(I)	=	6	9	10

The answer array may first be written as:

I	=	1	2	3	4	5
ANS(I)	=	3,0	- 2,0	4,0	1,0	- 1,0
NANS(I)	=	4	6	6	9	10

Summing the coefficients of like terms into one position and deleting the other by overwriting:

I	=	1	2	3	4
ANS(I)	=	3,0	2,0	1,0	- 1,0
NANS(I)	=	4	6	9	10

Which is recognised by factorisation to be:

$$3a^2 + 2ab + b^2 - ac$$

Subtraction: This is similar to addition except that signs of coefficients of one expression are changed in the array ANS.

Multiplication: This is performed on two terms by multiplying their coefficient arrays together and their term arrays together. Two expressions are multiplied together by multiplying each term of one by each term of the other. Coefficients of like terms are then added. (See addition).

e.g. Multiply the two expressions of the previous example together.

I	=	1	2	3	4	5	6
ANS(I)	=	12,0	3,0	- 3,0	- 8,0	- 2,0	2,0
NANS(I)	=	24	36	40	36	54	60

Collecting and summing like terms:

I	=	1	2	3	4	5
ANS(I)	=	12,0	- 5,0	- 3,0	- 2,0	2,0
NANS(I)	=	24	36	40	54	60

Which is recognised by factorisation to be:

$$12a^3b - 5a^2b^2 - 3a^3c - 2ab^3 + 2a^2bc$$

Factorisation: To facilitate output of results after manipulation, the terms of the array NTERM(I) must be factorised. Use is made of the truncating property of division on the computer in integer mode.

When two variables in integer mode are divided, the result is the integer part of the quotient.

e.g. If NTERM = 11 and I = 3, then J = NTERM/I = 3.

i.e. the remaining 2/3 is truncated rather than rounded.

In order to find whether a particular prime number I is a factor of a value NTERM, the quotient J multiplied by the divisor I is compared with the divided NTERM.

e.g. In the above example: NTERM - I * J = 11 - 3 * 3 $\neq$ 0

If they are equal, I is a factor of NTERM. This factor can then be removed by division and the result tested again to see whether the prime number is still a factor. If not, the next prime number is considered.

Note 1: The prime number "one" must not be used as it is always a factor and will cause "looping".

Note 2: If all the integers from two upwards are tested sequentially as factors, none of the factors removed will be non-prime.

A FORTRAN routine for this process would be as follows for variables a, b and c in the above example.

```
          NWRT  =  5
          DO 3  I  =  2, 5
          NA(I)  =  0
    1     NDIV  =  NTERM/I
          IF(NTERM - NDIV*I)3,2,3
    2     NA(I)  =  NA(I) + 1
          NTERM  =  NDIV
          GO  TO  1
    3     CONTINUE
          WRITE (NWRT, 100) COEF, NA(2), NA(3), NA(5)
  100     FORMAT(1Hb/, E10,4, 'A TO POWER', I2,',B TO POWER',
     1           I2,',C TO POWER', I2)
```

Input and Output: The format of the input expressions is as follows:-

One term of the expression is represented by a coefficient (with its sign) followed by powers of variables (including the power zero) in a fixed position and order.

A series of terms forming an expression is given one after another on cards until a zero coefficient terminates the expression. A new expression then starts on the next new card.

e.g. $- 3,5\ a^2bc^3 + 7,2\ ac^2$

would be represented on cards in the format:

FORMAT ((F 5.0, I3, 2X))

as:

- 3 . 5 ƀ 2 1 3 ƀ ƀ 7 . 2 ƀ ƀ 1 0 2 ƀ ƀ ƀ ƀ ƀ . . . . . .

The output from a program would have the format used in the routine above.

e.g. Output of the expression used in the above example would be:

- .3500 + 01 A TO POWER 2, B TO POWER 1, C TO POWER 3

.7200 + 01 A TO POWER 1, B TO POWER 0, C TO POWER 2

Algebra by Powers: In this method, a term is represented by the powers of variables in fixed decimal positions of the integers NTERM(I).

In a^2bc^3, if the powers of a and c are not expected to exceed 9 and the powers of b are not expected to exceed 99, then one decimal place is required for the powers of a and c and two for b. The term could then be represented by the number:

2013

where the first digit represents the power of a,
the next two digits represent the power of b,
and the last digit represents the power of c.

Coefficients are stored as in the previous section. Expressions are represented in the same manner as before except that the values in the array NTERM(I) are determined as above.

Addition: This is identical to the previous section in all respects.

Multiplication: This is performed by multiplying coefficients and adding terms of the NTERM(I) array. Otherwide this operation follows the same steps as in the previous section.

No factorisation is necessary. Powers of the variables are easily recognisable at any stage.

Input and Output: Input cards are identical to those required for prime numbers algebra. In this case, powers are read in together as a single integer rather than separately as in the previous section. Answers are printed out by simply writing the coefficient and corresponding powers term under appropriate headings.

e.g.	COEFFICIENT	POWERS OF:
		A B C
	- 6.0	30001

represents: $- 6\ a^3 c$

APPENDIX V

A USER PROGRAM FOR THE EXPLICIT ASSUMED DISPLACEMENT STIFFNESS MATRIX

In the accompanying program, the dimensions and properties of a triangle are read into an array 'B' as follows:-

B(1)	=	r	the base length of the triangle,
B(2)	=	s	the offset of the apex along the base of the triangle, (See figure 2),
B(3)	=	t	the height of a triangle,
B(4)	=	ν	Poisson's ratio for the material,
B(5)	=	h	the material thickness,
B(6)	=	E	Young's modulus of the material.

The values of the coefficients of the explicit stiffness matrix are read from cards into array 'A'. The coefficients are printed below in a matrix form.

A vector of parameters {P} is generated in array 'C' using the values in array 'B'. The vector {P} is given after the stiffness matrix coefficients in this appendix.

The upper triangular part of the stiffness matrix is found by multiplying array 'A' by array 'C'. The full matrix is formed in array STIF using the symmetry property.

The stiffness matrix can then be used in a main program. It must be noted that this matrix is unidimensional. In order to obtain correct units, rows and columns 2, 3, 5, 6, 8 and 9 must all be multiplied by r, the base length of the triangle.

The program and coefficients follow:

```
PROGRAM FOR FORMING TOCHER STIFFNESS
MATRIX FROM PUNCHED COEFFICIENTS
-----------------------------------------------------------------
      DIMENSION A(45,84),B(6),C(84),STIF(9,9)
ARRAY 'A' CONTAINS EXPLICIT COEFFICIENTS
ARRAY 'B' CONTAINS DIMENSIONS & PROPERTIES
      DO 1 I=1,45
1     READ(8,100)(A(I,J),J=1,84)
      READ(8,101)N
      DO 5 IA=1,N
      READ(8,102)(B(I),I=1,6)
      WRITE(5,104)(B(I),I=1,6)
104   FORMAT(1X/1X,'R=',F10.4,',S=',F10.4,',T=',F10.4,',U=',F10.4,',H=',
     1F10.4,',E=',F20.4/1X,119('-'))
PARAMETERS OF A HOMOGENEOUS POLYNOMIAL
ARE FORMED IN ARRAY 'C'
      BA=B(1)/B(2)
      I=0
      DO 2 IB=0,1
      FACTOR=B(6)*(B(5)**3.)/(72.*(1.-B(4)*B(4))*((-B(1)+2.*B(2)+B(3))**
     12.)*(B(1)**5.)*(B(3)**4.))
      IF(IB.EQ.1)FACTOR=FACTOR*B(4)
      DO 2 IC=0,6
      FACTOR=FACTOR*B(3)
      ID=8-IC
      GACTOR=(B(2)**ID)/BA
      DO 2 IE=0,ID
      GACTOR=GACTOR*BA
      I=I+1
2     C(I)=FACTOR*GACTOR
ARRAYS 'A' & 'C' ARE MULTIPLIED TO FORM THE
STIFFNESS MATRIX IN ARRAY 'STIF'
      I=0
      DO 4 JA=1,9
      DO 4 JB=JA,9
      M=0
      JC=(JA-1)-((JA-1)/3)*3
      IF(JC.EQ.0)M=M+1
      JC=(JB-1)-((JB-1)/3)*3
      IF(JC.EQ.0)M=M+1
      I=I+1
      STIF(JA,JB)=0.
      DO 3 J=1,84
3     STIF(JA,JB)=STIF(JA,JB)+A(I,J)*C(J)*(B(1)**M)
      M=0
4     STIF(JB,JA)=STIF(JA,JB)
      DO 5 I=1,9
5     WRITE(5,103)(STIF(I,J),J=1,9)
THE NUMERICAL VALUES CAN NOW BE USED IN
A FINITE ELEMENT PROGRAM
```

(continued)

```
100   FORMAT(20F4.0)
101   FORMAT(I4)
102   FORMAT(6F10.0)
103   FORMAT(1H ,9F13.4)
      END
```

A R R A Y   O F   C O E F F I C I E N T S

(EACH PARTITION CONTAINS A SINGLE ROW OF THE ARRAY 'A')

K(1,1) =	72	-288	432	-288	108	-144	216	-144	36	144	-504	648	-360	144	-216	216	-72	216	-648	756
	-432	144	-72	36	288	-720	576	-144	0	0	216	-432	360	-144	36	144	-216	216	-72	72
	-72	36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0																
K(1,2) = K(2,1) =	0	0	0	0	0	0	0	0	0	0	12	-48	96	-144	156	-96	24	0	12	-48
	102	-150	126	-42	0	24	-54	42	-30	18	0	24	-36	12	0	0	12	-12	6	0
	12	-6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	-12	30	-24	6	0	0	-18	30	-12	0	0	-6
	6	0	0	0																
K(1,3) = K(3,1) =	-36	156	-264	204	-36	-60	48	-12	0	-72	276	-408	270	-48	-24	0	6	-108	360	-486
	336	-120	24	-6	-144	408	-408	168	-24	0	-108	252	-240	102	-18	-72	132	-126	42	-36
	48	-24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	12	-18	-6	18	-6	0	0	18	-12	-18	12	0	0	6	0	-6	0	0	0
	0	0	0	0																
K(1,4) = K(4,1) =	-72	288	-432	288	-108	72	-36	0	0	-144	504	-648	360	-144	108	-36	0	-216	648	-756
	432	-144	36	0	-288	720	-576	144	0	0	-216	432	-360	144	-36	-144	216	-216	72	-72
	72	-36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0																
K(1,5) = K(5,1) =	0	0	0	0	0	0	0	0	0	0	-12	24	-36	48	-24	0	0	0	-12	12
	-30	36	0	-6	0	-24	42	-24	0	6	0	-24	36	-12	0	0	-12	12	-6	0
	-12	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	-12	6	0	0	0	0	-18	6	0	0	0	-6
	0	0	0	0																
K(1,6) = K(6,1) =	-36	132	-180	120	-60	36	-12	0	0	-72	228	-264	150	-72	42	-12	0	-108	288	-306
	168	-54	12	0	-144	312	-216	48	0	0	-108	180	-132	54	-12	-72	84	-78	24	-36
	24	-12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	12	-30	36	-24	6	0	0	18	-42	36	-12	0	0	6	-12	6	0	0	0
	0	0	0	0																
K(1,7) = K(7,1) =	0	0	0	0	0	72	-180	144	-36	0	0	0	0	0	108	-180	72	0	0	0
	0	0	36	-36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0																
K(1,8) = K(8,1) =	0	0	0	0	0	0	0	0	0	0	0	24	-60	24	48	-48	12	0	0	36
	-72	6	54	-24	0	0	12	-18	-6	12	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	24	-36	24	-6	0	0	36	-36	12	0	0	12
	-6	0	0	0																
K(1,9) = K(9,1) =	0	0	12	-36	60	-84	72	-24	0	0	0	24	-60	84	-90	48	-6	0	0	36
	-72	66	-36	6	0	0	48	-72	24	0	0	0	12	-12	-6	0	0	-12	6	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	-24	48	-30	6	0	0	0	-36	54	-18	0	0	0	-12	12	0	0	0	0
	0	0	0	0																

x {P}

(continued)

K(2,2) =	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	18
	-72	108	-72	18	0	0	16	-58	68	-26	0	0	16	-24	14	0	0	8	-4	0
	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-8	4	0	0	0	-8
	2	0	0	-2																
K(2,3) = K(3,2) =	0	0	0	0	0	0	0	0	0	0	-6	18	-12	-12	18	-6	0	0	-6	18
	-19	13	-11	5	0	-12	29	-21	7	-3	0	-12	22	-12	2	0	-6	8	-3	0
	-6	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	14	-35	28	-7	0	0	17	-33	14	0	0	5
	-7	0	0	0																
K(2,4) = K(4,2) =	0	0	0	0	0	0	0	0	0	0	-12	48	-96	96	-36	0	0	0	-12	48
	-102	102	-30	0	0	-24	54	-42	18	0	0	-24	36	-12	0	0	-12	12	-6	0
	-12	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	12	-30	24	-6	0	0	18	-30	12	0	0	6
	-6	0	0	0																
K(2,5) = K(5,2) =	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-18
	36	-18	0	0	0	0	-16	24	2	-5	0	0	-16	16	-1	0	0	-8	4	0
	0	-4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	-8	2	0	0	8
	-4	0	0	2																
K(2,6) = K(6,2) =	0	0	0	0	0	0	0	0	0	0	-6	30	-54	42	-12	0	0	0	-6	30
	-53	39	-10	0	0	-12	25	-17	6	0	0	-12	14	-4	0	0	-6	4	-1	0
	-6	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	-2	9	-10	3	0	0	1	5	-4	0	0	1
	1	0	0	0																
K(2,7) = K(7,2) =	0	0	0	0	0	0	0	0	0	0	0	0	0	48	-120	96	-24	0	0	0
	0	48	-96	42	0	0	0	0	12	-18	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0																
K(2,8) = K(8,2) =	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	-12	30	-24	6	0	0	0	-14	26	-11	0	0	0	-4	5	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	-2	0	0	0
	2	0	0	0																
K(2,9) = K(9,2) =	0	0	0	0	0	0	0	0	0	0	0	0	18	-54	54	-18	0	0	0	0
	18	-46	33	-5	0	0	0	8	-13	3	0	0	0	4	-2	0	0	0	-2	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	-4	6	-2	0	0	0	-2	2	0	0	0
	0	0	0	0																
K(3,3) =	18	-84	162	-168	102	-36	6	0	0	36	-150	254	-220	96	-14	-2	0	54	-198	304
	-252	118	-30	4	72	-228	272	-152	40	-4	54	-144	160	-78	14	36	-78	80	-26	18
	-30	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	-20	54	-50	18	-2	0	0	-26	46	-22	2	0	0	-8	8	0	0	0	0
	0	0	0	0																
K(3,4) = K(4,3) =	36	-156	264	-204	60	0	0	0	0	72	-276	408	-270	72	-6	0	0	108	-360	486
	-336	126	-24	0	144	-408	408	-168	24	0	108	-252	240	-102	18	72	-132	126	-42	36
	-48	24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	-12	18	-18	6	0	0	0	-18	12	-6	0	0	0	-6	0	0	0	0	0
	0	0	0	0																

$\times \{P\}$

(continued)

K(3,5) = K(5,3) =	0	0	0	0	0	0	0	0	0	0	6	-6	-6	6	0	0	0	0	6	0
	-7	0	1	0	0	12	-23	14	-1	-2	0	12	-22	12	-2	0	6	-8	3	0
	6	-5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	-2	-3	2	0	0	0	1	-7	2	0	0	1
	-2	0	0	0																
K(3,6) = K(6,3) =	18	-72	108	-72	18	0	0	0	0	36	-126	166	-100	26	-2	0	0	54	-162	200
	-130	46	-8	0	72	-180	160	-60	8	0	54	-108	86	-32	5	36	-54	40	-11	18
	-18	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	-4	8	-5	1	0	0	0	-10	15	-9	2	0	0	-4	4	-2	0	0	0
	0	0	0	0																
K(3,7) = K(7,3) =	0	0	0	0	-24	60	-48	12	0	0	0	0	0	-24	30	0	-6	0	0	0
	0	-6	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	24	-24	6	0	0	0	0	24	-12	0	0	0	0	6	0	0	0
	0	0	0	0																
K(3,8) = K(8,3) =	0	0	0	0	0	0	0	0	0	0	0	-12	42	-54	30	-6	0	0	0	-18
	50	-43	10	1	0	0	-6	13	-6	-1	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	-12	14	-6	1	0	0	-18	16	-4	0	0	-6
	3	0	0	0																
K(3,9) = K(9,3) =	0	0	-6	12	0	-12	6	0	0	0	0	-12	26	-20	10	-4	0	0	0	-18
	40	-38	20	-4	0	0	-24	44	-24	4	0	0	-6	8	-1	0	0	6	-5	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	12	-20	13	-7	2	0	0	18	-25	13	-4	0	0	6	-6	2	0	0	0
	0	0	0	0																
K(4,4) =	72	-288	432	-288	108	0	0	0	0	144	-504	648	-360	144	0	0	0	216	-648	756
	-432	144	0	0	288	-720	576	-144	0	0	216	-432	360	-144	36	144	-216	216	-72	72
	-72	36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0																
K(4,5) = K(5,4) =	0	0	0	0	0	0	0	0	0	0	12	-24	36	0	0	0	0	0	12	-12
	30	12	0	0	0	24	-42	24	12	0	0	24	-36	12	0	0	12	-12	6	0
	12	-6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	12	-6	0	0	0	0	18	-6	0	0	0	6
	0	0	0	0																
K(4,6) = K(6,4) =	36	-132	180	-120	36	0	0	0	0	72	-228	264	-150	48	0	0	0	108	-288	306
	-168	48	0	0	144	-312	216	-48	0	0	108	-180	132	-54	12	72	-84	78	-24	36
	-24	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	-12	30	-12	0	0	0	0	-18	42	-12	0	0	0	-6	12	0	0	0	0
	0	0	0	0																
K(4,7) = K(7,4) =	0	0	0	0	0	-72	36	0	0	0	0	0	0	0	-108	36	0	0	0	0
	0	0	-36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0																
K(4,8) = K(8,4) =	0	0	0	0	0	0	0	0	0	0	0	-24	60	-24	0	0	0	0	0	-36
	72	-6	-6	0	0	0	-12	18	6	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	-24	36	-24	6	0	0	-36	36	-12	0	0	-12
	6	0	0	0																

$\times \{P\}$

(continued)

K(4,9) = K(9,4) =	0	0	-12	36	-60	36	0	0	0	0	0	-24	60	-84	42	0	0	0	0	-36
	72	-66	24	0	0	0	-48	72	-24	0	0	0	-12	12	6	0	0	12	-6	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	24	-48	30	-6	0	0	0	36	-54	18	0	0	0	12	-12	0	0	0	0
	0	0	0	0																
K(5,5) =	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	18
	0	0	0	0	0	0	16	10	0	0	0	0	16	-8	6	0	0	8	-4	0
	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-8	12	-4	0	0	-8
	6	0	0	-2																
K(5,6) = K(6,5) =	0	0	0	0	0	0	0	0	0	0	6	-18	12	0	0	0	0	0	6	-12
	7	4	0	0	0	12	-19	6	4	0	0	12	-14	4	0	0	6	-4	1	0
	6	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	14	-7	0	0	0	0	17	-1	-2	0	0	5
	2	0	0	0																
K(5,7) = K(7,5) =	0	0	0	0	0	0	0	0	0	0	0	0	0	-48	24	0	0	0	0	0
	0	-48	0	6	0	0	0	0	-12	-6	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0																
K(5,8) = K(8,5) =	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	12	-6	0	0	0	0	0	14	-2	-1	0	0	0	4	1	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-4	2	0	0	0
	-2	0	0	0																
K(5,9) = K(9,5) =	0	0	0	0	0	0	0	0	0	0	0	0	-18	18	0	0	0	0	0	0
	-18	8	5	0	0	0	0	-8	3	2	0	0	0	-4	2	0	0	0	2	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	4	-2	0	0	0	0	2	0	0	0	0
	0	0	0	0																
K(6,6) =	18	-60	78	-48	12	0	0	0	0	36	-102	110	-60	16	0	0	0	54	-126	124
	-64	16	0	0	72	-132	80	-16	0	0	54	-72	52	-26	6	36	-30	32	-12	18
	-6	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	-20	26	-8	0	0	0	0	-26	32	-8	0	0	0	-8	8	0	0	0	0
	0	0	0	0																
K(6,7) = K(7,6) =	0	0	0	0	24	-36	12	0	0	0	0	0	0	24	-42	12	0	0	0	0
	0	6	-12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	-24	24	-6	0	0	0	0	-24	12	0	0	0	0	-6	0	0	0
	0	0	0	0																
K(6,8) = K(8,6) =	0	0	0	0	0	0	0	0	0	0	0	-12	18	-6	0	0	0	0	0	-18
	22	-1	-2	0	0	0	-6	5	2	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	-12	22	-14	3	0	0	-18	20	-6	0	0	-6
	3	0	0	0																
K(6,9) = K(9,6) =	0	0	-6	24	-30	12	0	0	0	0	0	-12	34	-36	14	0	0	0	0	-18
	32	-26	8	0	0	0	-24	28	-8	0	0	0	-6	4	1	0	0	6	-1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	12	-28	25	-7	0	0	0	18	-29	17	-2	0	0	6	-6	2	0	0	0
	0	0	0	0																

x {P}

(continued)

	0	0	0	0	0	0	144	-144	36	0	0	0	0	0	0	144	-72	0	0	0
	0	0	0	36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
K(7,7) =	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0																
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-48	48	-12	0	0	0
	0	0	-48	24	0	0	0	0	0	-12	0	0	0	0	0	0	0	0	0	0
K(7,8) = K(8,7)=	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0																
	0	0	0	0	0	48	-72	24	0	0	0	0	0	0	48	-48	6	0	0	0
	0	0	12	-6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
K(7,9) = K(9,7)=	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0																
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	24	-24	6	0	0	0	0	24	-12	0	0	0	0	6	0	0	0	0	0
K(8,8) =	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0																
	0	0	0	0	0	0	0	0	0	0	0	0	0	-12	18	-6	0	0	0	0
	0	-10	10	-1	0	0	0	0	-2	1	0	0	0	0	0	0	0	0	0	0
K(8,9) = K(9,8)=	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	-4	2	0	0	0	0	-2	0	0	0
	0	0	0	0																
	0	0	0	0	18	-36	18	0	0	0	0	0	0	20	-30	10	0	0	0	0
	0	10	-10	4	0	0	0	0	8	-4	0	0	0	0	6	0	0	0	0	0
K(9,9) =	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	-8	8	-2	0	0	0	0	-12	6	0	0	0	0	-4	0	0	0
	0	0	0	0																

$\times \{P\}$

where: $\{P\}^t = \dfrac{E\,h^3}{72(1-\nu^2)(-r+2s+t)^2\, r^5 t^3}\Big[s^8,\ rs^7,\ r^2s^6,\ r^3s^5,\ r^4s^4,\ r^5s^3,\ r^6s^2,\ r^7s,\ r^8,$

$s^7t,\ rs^6t,\ r^2s^5t,\ r^3s^4t,\ r^4s^3t,\ r^5s^2t,\ r^6st,\ r^7t,\ s^6t^2,\ rs^5t^2,\ r^2s^4t^2,\ r^3s^3t^2,$

$r^4s^2t^2,\ r^5st^2,\ r^6t^2,\ s^5t^3,\ rs^4t^3,\ r^2s^3t^3,\ r^3s^2t^3,\ r^4st^3,\ r^5t^3,\ s^4t^4,\ rs^3t^4,\ r^2s^2t^4,$

$r^3st^4,\ r^4t^4,\ s^3t^5,\ rs^2t^5,\ r^2st^5,\ r^3t^5,\ s^2t^6,\ rst^6,\ r^2t^6,\ \nu s^8,\ \nu rs^7,\ \ldots\ldots$

$\ldots\ldots\ldots\ldots\ \nu r^2t^6\Big]$

APPENDIX VI

THE COMPUTER ROUTINE FOR GENERATING AND SOLVING POLYNOMIAL IDENTITIES

The program at the end of this appendix generates functions and solves identities such as the following (see also (78)):

$$1{,}0A(1)\alpha\beta\gamma\Delta + 2{,}0C(2)b\alpha\Delta - 8{,}0M(5)a^2 \equiv 0 \qquad (86)$$

Each term of this identity consists of:

(i) a coefficient (represented by the variable KA in the program),

(ii) the name of one of the twelve possible functions of matrix (76). (A variable LA represents this numerically),

(iii) the cyclic or mirror variation number (MA) corresponding to the numbering in (73),

(iv) a factor by which the whole function must be multiplied. (The powers of a, b, c, α, β, γ and Δ are given in successive decimal places of a variable NA as described in the algebra by powers of Appendix 4).

In the order KA, LA, MA, NA the above identity can be represented numerically as:

1, 1, 1, 0001111,
2, 3, 2, 0101001,
- 8,12, 5, 2000000,

where the values for LA are:

A	B	C	D	E	F	G	H	J	K	L	M
1	2	3	4	5	6	7	8	9	10	11	12

Steps in the accompanying program are explained as follows:

1. Preparation

The identifiers NPRM and NPRN are used to position the powers of a, b, c, α, β, γ, and Δ in the integer array NT.

The root length degree of functions is read into variable NL. The number of terms ML in a function of root length degree NL is calculated. (This depends upon whether NL is odd or even).

From the value of ML it can be seen that the total number of coefficients in all of the functions is:

$$12 * ML$$

Also the coefficients of function LA number from:

$$(LA - 1) * ML + 1$$

to:

$$LA * ML$$

2. Equation Formation Cycle

There are thirteen identities (see (78) and the small identities in (76)) which are used to reduce the number of unknown coefficients in the stiffness matrix. In each identity, the like terms are added and equated to zero forming a subset of simultaneous equations. The full set is very large. Many of the equations are redundant.

At the end of each equation formation cycle a Gauss Jordan reduction is performed on the equations. The reduction is described in Appendix 1 and mentioned in 7. below.

3. Input

The number of terms in any identity is read into NOPS. The numerical description of each term is read and operated upon in turn.

4. Polynomial Generation

The coefficients corresponding to polynomial number LA are identified.

The cyclic or mirror variation of the function is determined from the variable MA.

The polynomial is then formed, term by term, in a nest of "do-loops". Each term is multiplied by the factor NA in the manner described as algebra by powers in Appendix 4. The resulting parameters are stored in an array NT. The corresponding coefficient number and multiplier, KA, are stored in a separable manner in an array NC.

5. <u>Substitution for Greek Letters raised to a Power</u>

The values of the array NT are scanned for powers of the greek letters. If these are found to be two or greater they are substituted for, according to equations (72). This involves lengthening the array NT. The array NC must be adjusted accordingly.

6. <u>Collection of Coefficients of Like Terms</u>

A recorder is set to the value of the first member of the array NT. The whole array is then scanned for this value. Where it occurs, the coefficient number from array NC is written into another array, IA(LC). The multiplier of the coefficient from NC (corresponding to KA) is written into A(LC). When each term is collected, the value of NT is set to a negative quantity. This is so that terms are only considered once.

The first value of LC in equation number N is recorded in element JA(N) of an array.

The recorder is then set to the next non-negative value of the array NT. The next equation is formed in the same way.

7. <u>Solution of Equations</u>

The matrix of equations formed is very sparse and contains more equations than unknowns but fewer linearly independent equations than unknowns. The equations can only be reduced to their simplest form using the randomly sparse matrix reduction routine described in Appendix 1. No full solution is possible.

8. Output

The reduced set of equations is written out as a series of linear expressions which equal zero. One of the variables in each equation can be expressed as a linear combination of the others.

The matrix of inter-relationships given as the input to the coefficient evaluation program (Appendix 7) was determined in this way. The numbering of coefficients is changed before the results are used in the subsequent program. Only the format of the input matrix is shown (Appendix 7), to avoid confusion.

Example of an Identity Solution:

Supposing that the identity (86) (given at the start of this appendix) is to be solved for functions with root length degree 1:

$$\text{i.e.}\quad f_n(1) = C_{4n-3}\alpha + C_{4n-2}\beta + C_{4n-1}\gamma + C_{4n}\Delta$$

$$\begin{aligned}\text{hence: } A(1) &= C_1\alpha + C_2\beta + C_3\gamma + C_4\Delta \\ C(2) &= C_9\alpha + C_{10}\gamma + C_{11}\beta + C_{12}\Delta \\ M(5) &= C_{45}\gamma + C_{46}\alpha + C_{47}\beta + C_{48}\Delta\end{aligned}$$

The whole identity would then be expressed as:

$$\begin{aligned}&C_1\alpha^2\beta\gamma\Delta + C_2\alpha\beta^2\gamma\Delta + C_3\alpha\beta\gamma^2\Delta + C_4\alpha\beta\gamma\Delta^2 \\ &+ 2C_9b\alpha^2\Delta + 2C_{10}b\alpha\gamma\Delta + 2C_{11}b\alpha\beta\Delta + 2C_{12}b\alpha\Delta^2 \\ &- 8C_{45}a^2\gamma - 8C_{46}a^2\alpha - 8C_{47}a^2\beta - 8C_{48}a^2\Delta \equiv 0\end{aligned}$$

Substituting for squared greek letters from (72):

$$\begin{aligned}&- C_1a\beta\gamma\Delta + C_1b\beta\gamma\Delta + C_1c\beta\gamma\Delta + C_2a\alpha\gamma\Delta - C_2b\alpha\gamma\Delta + C_2c\alpha\gamma\Delta + C_3a\alpha\beta\Delta \\ &+ C_3b\alpha\beta\Delta - C_3c\alpha\beta\Delta + C_4a\alpha\beta\gamma + C_4b\alpha\beta\gamma + C_4c\alpha\beta\gamma - 2C_9ab\Delta + 2C_9b^2\Delta \\ &+ 2C_9bc\Delta + 2C_{10}b\alpha\gamma\Delta + 2C_{11}b\alpha\beta\Delta + 2C_{12}ab\alpha + 2C_{12}b^2\alpha + 2C_{12}bc\Delta \\ &- 8C_{45}a^2\gamma - 8C_{46}a^2\alpha - 8C_{47}a^2\beta - 8C_{48}a^2\Delta \equiv 0\end{aligned}$$

Two typical terms of this identity would be $2C_{10}b\alpha\gamma\Delta$ and $-8C_{48}a^2\Delta$. These would be represented in the arrays as:

$$NT(16) = 0101011$$
$$NC(16) = 10000 + 2 \times 1000 + 10 = 12010$$
$$NT(24) = 2000001$$
$$NC(24) = 10000 - 8 \times 1000 + 48 = 2048$$

As the coefficients never become less than -8, by adding 10000, NC is always positive. The multiplication by 1000 separates the two components of NC so that they can be recognised later.

Like values of NT cause corresponding members of NC to be collected to form an equation with the coefficients C as unknowns. If the values of NT are then made negative, they will not be considered twice.

The FORTRAN listing of the polynomial generation and identity solution program follows below. The equation reduction subroutine HEQSN which is called in this program is listed in Appendix 1.

PROGRAM FOR FORMING IDENTITIES AND REDUCING THE RESULTS TO THEIR SIMPLEST FORM

```
      DIMENSION NPRM(3),NT(10000),NC(10000),NPRN(3)
      COMMON A(15000),IA(15000),JA(1000),N,IST2,M,KB
P R E P A R A T I O N
      NPRM(1)=1000
      NPRM(2)=100
      NPRM(3)=10
      NPRN(1)=1000000
      NPRN(2)=100000
      NPRN(3)=10000
      LC=0
      N=0
      READ(8,100)NL
100   FORMAT(I4)
      WRITE(5,101)NL
101   FORMAT(' THE FUNCTION POLYNOMIAL HAS DEGREE',I4)
      N1=0
      NM=NL/2
      IF(NL-NM*2)1,1
      N1=1
1     ML=N1*(4*NM+3)+1
      DO 10 NN=1,NM
10    ML=ML+8*NN
      WRITE(5,102)ML
102   FORMAT(' THERE ARE ',I4,'TERMS IN EACH FUNCTION')
      N=12*ML
E Q U A T I O N   F O R M A T I O N   C Y C L E
      DO 8 KB=1,13
      READ(8,100)NOPS
      IE=0
      DO 11 I=1,NOPS
I N P U T   O F   I D E N T I T I E S
      READ(8,104)KA,LA,MA,NA
104   FORMAT(I2,I2,I1,I7)
      LB=(LA-1)*ML
D E T E R M I N I N G   T H E   C Y C L I C   V A R I A T I O N
      IB=(MA+1)/2
      ID=4-(MA-((MA-1)/3)*3)
      IC=6-IB-ID
      IG=0
P O L Y N O M I A L   G E N E R A T I O N
      DO 11 N2=N1,4,2
      N3=(NL-N2)/2
      DO 11 N4=0,N3
      N5=N3-N4
      DO 11 N6=0,N5
      N7=N5-N6
```

(continued)

```
      DO 11 N8=0,1
      DO 11 N9=0,1
      DO 11 N10=0,1
      DO 11 N11=0,1
      IF(N8+N9+N10+N11-N2)11,,11
      IG=IG+1
      IE=IE+1
      NT(IE)=N11+N10*NPRM(ID)+N9*NPRM(IC)+N8*NPRM(IB)+N7*NPRN(ID)+N6*NPR
     1N(IC)+N4*NPRN(IB)+NA
      NC(IE)=LB+IG+KA*1000+10000
11    CONTINUE
```

T E S T I N G   F O R   P O W E R S   O F   G R E E K   L E T T E R S

```
      DO 5 J=1,4
      JB=J-((J-1)/3)*3
      JC=JB+1
      JC=JC-((JC-1)/3)*3
      JD=JC+1
      JD=JD-((JD-1)/3)*3
      K=0
3     IF(IE-K)5,5
      K=K+1
4     NK=NT(K)/(10**(J-1))
      NJ=NK/10
      NJ=NK-10*NJ
      IF(NJ-2)3
```

S U B S T I T U T I N G   F O R   P O W E R E D   G R E E K   L E T T E R S

```
      IE=IE+2
      NT(IE)=NT(K)-2*(10**(J-1))+1000*(10**JC)
      NT(IE-1)=NT(K)-2*(10**(J-1))+1000*(10**JD)
      NT(K)=NT(K)-2*(10**(J-1))+1000*(10**JB)
      NC(IE)=NC(K)
      NC(IE-1)=NC(IE)
      IF(J.LE.1)GO TO 4
      NC(IE-1)=NC(K)+(10-NC(K)/1000)*2000
      GO TO 4
5     CONTINUE
```

C O L L E C T I O N   O F   C O E F F I C I E N T S   O F   L I K E   T E R M S

```
      DO 7 I=1,IE
      IF(NT(I))7
      N=N+1
      DO 6 J=I,IE
      IF(NT(I)-NT(J))6,,6
      LC=LC+1
      IA(LC)=NC(J)-(NC(J)/1000)*1000
      A(LC)=(NC(J)/1000)-10
      JA(N)=LC
      IF(J-I)6,6
      NT(J)=-NT(J)
6     CONTINUE
7     CONTINUE
```

(continued)

```
      WRITE(5,105)KB
105   FORMAT(' OPERATION NO',I4)
R E D U C T I O N   O F   E Q U A T I O N S   T O   S I M P L E S T   F O R M
( T H I S   C O N T A I N S   O U T P U T   F A C I L I T Y )
      CALL HEQSN
      LC=JA(N)
      WRITE(5,106)IE,IST2,N
106   FORMAT(1H ,'MAXIMUM TERMS AFTER OPERATION',I4/1X,
     1'MAXIMUM COEFFITIENTS IN EQUATION MATRIX',I5/1X,
     2'MAXIMUM NUMBER OF EQUATIONS',I4)
8     CONTINUE
9     STOP
```

APPENDIX VII

THE INDUSTRIAL PACKAGE PROGRAM

Function

The first function of this program is to determine and store the values of all the coefficients required for an explicit stiffness matrix generation. These constants could have been determined once and punched onto cards (in the form of the matrices (84)). If this were done, the accuracy to which coefficients could be punched would have to be limited. When coefficients are generated in the user computer, their accuracy matches the capabilities of the computer.

The second function of this program is to check the data describing an element conformation. As this program is a digression in this thesis, apart from details of its use given in section 4.2, only an edited listing will be included here.

The main function of the program is to generate and solve the stiffness matrix of a flat plate in bending. The plate may be of any shape but bending must conform to the requirements of the small deflection theory. The program uses the coefficients and data generated by the first two components of the package.

Components

The package is comprised of seven file elements - four main programs, two subroutines and one data file. The main programs are:

1. Coefficient evaluation program.
2. Curve fitting program.
3. Data checking program.
4. Updated version of 1. which performs a finite element structural analysis.

The subroutines are:

5. Reduction of a banded symmetric matrix.
6. Reduction of a full non-symmetric matrix.

The data file contains a matrix relating the coefficients of all the elements of submatrices (1,1) and (1,2) of the $[K_1]$ matrix (see figure 16) to twenty five different unknown coefficients. This matrix is compiled from the results of the program described in Appendix 6. The matrix is edited and listed at the end of this appendix.

All the elements of the two submatrices (18 elements giving a total of 288 coefficients) are represented in the matrix. In order to evaluate the full stiffness matrix it is necessary to generate only the cyclic variations (not the mirror variations) of a full homogeneous polynomial of root length degree three. The three variations are:

$$\{P_1\} = \begin{bmatrix} a\alpha \\ a\beta \\ a\gamma \\ a\Delta \\ b\alpha \\ b\beta \\ b\gamma \\ b\Delta \\ c\alpha \\ c\beta \\ c\gamma \\ c\Delta \\ \beta\gamma\Delta \\ \alpha\gamma\Delta \\ \alpha\beta\Delta \\ \alpha\beta\gamma \end{bmatrix} ; \quad \{P_2\} = \begin{bmatrix} b\beta \\ b\gamma \\ b\alpha \\ b\Delta \\ c\beta \\ c\gamma \\ c\alpha \\ c\Delta \\ a\beta \\ a\gamma \\ a\alpha \\ a\Delta \\ \alpha\gamma\Delta \\ \alpha\beta\Delta \\ \beta\gamma\Delta \\ \alpha\beta\gamma \end{bmatrix} ; \quad \{P_3\} = \begin{bmatrix} c\gamma \\ c\alpha \\ c\beta \\ c\Delta \\ a\gamma \\ a\alpha \\ a\beta \\ a\Delta \\ b\gamma \\ b\alpha \\ b\beta \\ b\Delta \\ \alpha\beta\Delta \\ \beta\gamma\Delta \\ \alpha\gamma\Delta \\ \alpha\beta\gamma \end{bmatrix} \qquad (87)$$

Variation 1: (a,b,c) Variation 2: (b,c,a) Variation 3: (c,a,b)

The data is read into a four dimensional array:

IH(I, J, K, L) dimensioned IH(3, 6, 16, 25)

where: L corresponds to the number of the unknown independent coefficient.

I corresponds to the row number of the element of the matrix $[K_1]$ (76)

J corresponds to the column number of the element of matrix $[K_1]$

K is the number of the coefficient of the function representing element (I,J).

Coefficient Evaluation Program

The steps of this program are as follows:-

1. Preparation

A function which gives the jth digit of a cyclic series: 1, 2, 3, 1, ... which starts with the digit i is first prepared. The elements of the transformation matrices which are zero in all cases (see equation (61)) are set. The data from the matrix described above is read into an array IH(I, J, K, L).

2. Input of Standard Plate Data

The number of nodes and elements on the standard plate as well as the range and number of increments of Poisson's ratio are read. The coordinates of the nodes are read. The node numbers of each element in turn named in clockwise order about the z axis are read into an array NODE. The load vector is read into an array F and the boundary conditions into an array NBC.

3. The Poisson's Ratio Cycle

For each increment in the value of Poisson's ratio, a whole matrix (array A) of linear equations in the unknown independent coefficients is set up and solved. The array is first set to zero in each case. The value of Poisson's ratio is calculated. The displacement vector (array D) for that value of Poisson's ratio is compiled (see equation (71)) for an applied unit edge moment.

4. The Element Cycle

For each element in turn, the lengths of the sides a, b and c (E(1), E(2) and E(3) respectively) and the parameters α, β. γ and Δ (G(1), G(2), G(3) and G(4) respectively) are calculated from the nodal coordinates. The coordinates of the centre of the inscribed circle are calculated.

From the global coordinates of the inscribed circle and the three nodes in turn, transformation matrices are formed. (The stiffness matrix was initially in terms of the local axis system shown in figure 15). As these are orthogonal transformations, the inverse transformation matrices (being the transpose) need not be formed.

5. Formation of System Stiffness Matrix

Each element of the element stiffness matrix consists of a number of the unknown coefficients. Each of these is multiplied by a factor composed of terms of one of the three cyclic variations of the assumed polynomial (vectors (87)). This can be seen from the edited data file containing the array IH(I, J, K, L) at the end of this appendix. Each element of the stiffness matrix must also be multiplied by an appropriate factor (see equation (79) and vectors (74)) in order to give it correct dimensions.

The element stiffness matrix must then be multiplied by the transformation matrices and the appropriate nodal displacements. The resulting value is added into the row of array A corresponding to the nodal displacement. Each column of row A corresponds to an unknown coefficient. The factor multiplying this coefficient is added to that column.

In the program the row and column numbers are determined in a number of nested cycles. The cyclic variation of the assumed function required is also generated. In the cycles:

<u>I</u> represents the number of the submatrix row of the element stiffness matrix (76). It also indicates which transformation submatrix must be transposed and applied as a premultiplier.

<u>J</u> represents the number of the submatrix column of the element stiffness matrix. It also indicates which transformation submatrix must be applied to the stiffness submatrix as a post-multiplier.

<u>K</u> and <u>L</u> are the row and column numbers respectively of elements of the submatrix (I, J).

This is done by shifting lower rows up in the matrix. Columns of the system stiffness matrix corresponding to boundary conditions do not have to be deleted as they are multiplied by displacements in the array D which have the value zero.

7. Solution and Back Substitution

The matrix A is solved by calling a subroutine which uses the Gauss Jordan elimination procedure of Appendix 1. The values of the unknown coefficients appear in order in the position of the load vector in array A.

These values are then substituted into the relationship array IH to find all 288 coefficients of the two submatrices (1,1) and (1,2).

8. Output

The values of the coefficients are written into the data file for use in the curve fitting program which introduces Poisson's ratio. The whole Poisson's ratio cycle is then repeated for a new increment within the range chosen.

Curve Fitting Program

This is a simple least squares curve fitting program using the theory given in Appendix 3.

All the coefficient values calculated in the previous program are read into an array B(I, J, K, NU). The variable NU corresponds to the cycle number of Poisson's ratio. For each value of B(I, J, K) a set of parabolic equations in the value U (Poisson's ratio) are set up in array A. Each row corresponds to a different value of U. The first column contains the value 1, the second U, the third U^2 and the fourth the corresponding value of the coefficient. NI increments in U give a matrix of dimension (NI by 4). This can be solved by regression techniques (Appendix 3).

The resulting three coefficients of 1, U and U^2 overwrite the first three NU values of B(I, J, K). When all of the coefficients have been considered, the resulting three matrices are written onto disc file (see matrices 84)).

Finite Element User Program

The preparation, calculation of dimensions of the finite element and other function parameters, the formation of transformation matrices and the generation of polynomial functions in this program are the same as in the coefficient solution program. For this reason, the finite element program is compiled as an updated version of the coefficient evaluation program. The changes made by the updating deck are as follows:-

1. Input

The device numbers of files containing element data and explicit stiffness matrix coefficients, the number of nodes and elements on the plate and the values of Poisson's ratio, Young's modulus and the thickness of the plate are on the first data card.

The values of the coefficients of the explicit stiffness matrix are read from disc as three matrices. These are multiplied by 1, U and U^2 respectively and summed to form a single coefficient matrix Q(I, J, K).

The nodal coordinates, element numbering, and any uniformly distributed loading is read either from disc file or cards depending upon the device numbers read. This disc file was created by the data debugging routine.

The number of load vectors, any point loading and the boundary condition restraints are read essentially from data cards.

2. System Stiffness Matrix

The element stiffness matrix is first formed in local coordinates as the array F by a number of program loops involving the cyclic variation function generator.

This is transformed to global coordinates and added into the appropriate rows and columns of the system stiffness matrix in array A, in a separate nest of program loops. The array A contains only the leading diagonal of the stiffness matrix and the upper half band width formed obliquely as shown in Appendix 1.

3. The Load Vector and Boundary Conditions

The coordinates of the centroid of each element are calculated. The load vector caused by the distributed loads is then formed as described in section 4.4. This is added to the appropriate vector of point loads. Any number of load vectors may be treated simultaneously. (See Appendix 1).

The boundary conditions are then applied as described in section 4.5. Allowance must be made for the oblique formation of the banded stiffness matrix.

4. Solution

A subroutine which solves a symmetric banded matrix (using the theory of Appendix 1) is then called. The results must be multiplied by a factor containing Young's modulus before the correct displacements can be printed.

5. Stresses

The stresses are solved according to the suggested theory of section 4.3. The node connections are first read from the disc file created by the data debugging program.

Three equations in seven unknowns are created for each nodal connection. If there are less than three connections to a node, an extra equation is created artificially; linking two unknowns together. If three or more nodal connections exist, there are more than the requisite seven equations to solve the seven unknowns. The set of equations is solved using the regression technique of Appendix 3 and the matrix elimination subroutine of the previous program.

The results of this solution are substituted into appropriate equations for the stresses. A vector of stresses corresponding to each load vector of the problem is formed in array C.

6. Output

A warning that stresses are unreliable at the edges of a problem is written above every set of results. In addition, if less than three connections to a node exist, then another warning against confidence in the accuracy of stresses is given.

The results of each load vector are written under appropriate headings as shown in the example. (See section 5.3).

A full listing of the package program follows below.

MATRIX OF RELATIONSHIPS OF COEFICIENTS TO THE INDEPENDENT UNKNOWNS

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	
A(1)	A_1=	-8	-8	8													32	-16			32						× aα
	A_2=	-4	4	12				-8				8			-8			8			8	-8				8	× aβ
	A_3=	-4	4	12				-8				8			-8			8			8	-8				8	× aγ
	A_4=						-16				16			-16											16		× aΔ
	A_5=	-8																			16	-16					× bα
	A_6=	-4	-4	4				8							-16		16	-8			16					8	× bβ
	A_7=	4	4	-4				-8				8			-8		-16	-8			8	8				8	× bγ
	A_8=										-8			-8											16		× bΔ
	A_9=	-8																			16	-16					× cα
	A_{10}=	4	4	-4				-8				8			-8		-16	-8			8	9				8	× cβ
	A_{11}=	-4	-4	4				8							-16		16	-8			16					8	× cγ
	A_{12}=										-8			-8											16		× cΔ
	A_{13}=					16							-8			24											× βγΔ
	A_{14}=								16	-8			8			16				-32				16			× αγΔ
	A_{15}=								16	-8			8			16				-32				16			× αβΔ
	A_{16}=				16														64				48				× αβγ
C(1)																											
			4					-4				4					-4										
			-4					4				-4					4										
					-8														-24				-16				
		2	2	-2				-4				4					-8					4					
		2	2	-2				-4							4		-8				-4						
						-3										-8											
					8														24				16				
		-2	-2	2				4							-4		8				4						
		-2	-2	2				4				-4					8					-4					
						8										8											
									-4				-2			-2				4				-4			
									4				2			2				-4				4			
D(1)														-8											8		
									-4				2			2				4				4			
									-4				2			2				4				4			
																	8					3					
											-4			-4											8		
									4	-4			2			6				-12				4			
									4	-4			2			6				-12				4			
		-4																			8	-9					
											-4			-4											8		
									4	-4			2			6				-12				4			
									4	-4			2			6				-12				4			
		-4																			8	-3					

← INDEP. COEF. NOS.

A(1): $\Sigma = K(1,1)$

$x\{P_1\} = K(1,1)$

$x\{P_2\} = K(4,4)$

$x\{P_3\} = K(7,7)$

C(1):

$x\{P_1\} = K(1,2) = K(2,1)$

$x\{P_2\} = K(4,5) = K(5,4)$,

$x\{P_3\} = K(7,8) = K(8,7)$

D(1):

$x\{P_1\} = K(1,3) = K(3,1)$

$x\{P_2\} = K(4,6) = K(6,4)$

$x\{P_3\} = K(7,9) = K(9,7)$

(continued)

← INDEP. COEF. NOS.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
																		8				8			
														-4			-4			4					4
														-4			-4			4					4
												-4			4										
C(1)																									
		4					-4				4					-4									
		-4					4				-4					4									
				-8														-24				-16			
	2	2	-2				-4				4					-8					4				
	2	2	-2				-4							4		-8				-4					
					-8										-8										
				8														24				16			
	-2	-2	2				4							-4		8				4					
	-2	-2	2				4				-4					8					-4				
					8										8										
								-4				-2			-2				4				-4		
								4				2			2				-4				4		
G(1)		-2	2														-4								
	1	-1	1				-2				-1			-1		-4				-1	1				
	1	-1	1				-2				-1			-1		-4				-1	1				
						-4				-2			-2												
	1	1	-1				-2				1			1		-4				-1	1				
	-1	-1	1				2				-1			-1		4				1	-1				
	-1	-1	1				2				-1			-1		4				1	-1				
	1	1	-1				-2				1			1		-4				-1	1				
					-4							-2			-2										
				-4														-8				-4			
J(1)																									
								-2				1			1				2				2		
								2				-1			-1				-2				-2		
												2			-2										
								2				1			1				-2				2		
								-2				-1			-1				2				-2		
																		-4				-4			
												-2			2										
								2				1			1				-2				2		
								-2				-1			-1				2				-2		
																		4				4			
										-1			1							-1	-1				

C(1):

$x\{P_1\} = K(2,1) = K(1,2)$

$x\{P_2\} = K(5,4) = K(4,5)$

$x\{P_3\} = K(8,7) = K(7,8)$

G(1):

$x\{P_1\} = K(2,2)$

$x\{P_2\} = K(5,5)$

$x\{P_3\} = K(8,8)$

J(1):

$x\{P_1\} = K(2,3) = K(3,2)$

$x\{P_2\} = K(5,6) = K(6,5)$

$x\{P_3\} = K(8,9) = K(9,8)$

(continued)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	
											1			-1						1	1					INDEP. COEF. NOS.
													-8											8		
								-4				2			2				4				4			
								-4				2			2				4				4			
																8					8					$x\{P_1\} = K(3,1) = K(1,3)$
										-4			-4											8		
								4	-4			2			6				-12				4			
								4	-4			2			6				-12				4			
D(1)	-4																			8	-8					$x\{P_2\} = K(6,4) = K(4,6)$
										-4			-4											8		
								4	-4			2			6				-12				4			
								4	-4			2			6				-12				4			
	-4																			8	-8					
																		8				8				$x\{P_3\} = K(9,7) = K(7,9)$
														-4			-4			4					4	
														-4			-4			4					4	
												-4			4											
								-2				1			1				2				2			
								2				-1			-1				-2				-2			
																										$x\{P_1\} = K(3,2) = K(2,3)$
												2			-2											
								2				1			1				-2				2			
								-2				-1			-1				2				-2			
J(1)																		-4				-4				$x\{P_2\} = K(6,5) = K(5,6)$
												-2			2											
								2				1			1				-2				2			
								-2				-1			-1				2				-2			
																		4				4				$x\{P_3\} = K(9,8) = K(8,9)$
											-1			1						-1	-1					
											1			-1						1	1					
	-2																									
											-1			-1			-2			-1	1				-2	
											-1			-1			-2			-1	1				-2	
										-2			-2											-4		$x\{P_1\} = K(3,3)$
	-2																			4	-4					
											-1			-1			-2			1	-1				2	
											-1			-1			-2			1	-1				2	
H(1)										-2			-2											4		$x\{P_2\} = K(6,6)$
	-2																			4	-4					
											-1			-1			-2			1	-1				2	
											-1			-1			-2			1	-1				2	
										-2			-2											4		
																										$x\{P_3\} = K(9,9)$
									-2						2				-4							
									-2						2				-4							

(continued)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	← INDEP. COEF. NOS.
	4	4	-4													-16	8			-16						
	8		-8													-8	-8			-8	16					
	-4	-4	-4				8				-8			8		8					-8				-8	
						8				-8			8											-8		$x\{P_1\} = K(1,4) = K(4,1)$
	8		-8													-8	-8			-8	16					
	4	4	-4													-16	8			-16						
	-4	-4	-4				8				-8			8		8					-8				-2	
B(1)						8				-8			8											-8		$x\{P_2\} = K(4,7) = K(7,4)$
			8													8	8			-8						
			8													8	8			-8						
							-8							16											-8	
						-8				16														-8		
					-8							4			-12											$x\{P_3\} = K(7,1) = K(1,7)$
					-8							4			-12											
					8			-16	8			-12			-4				32				-16			
				-8														-32				-24				
				-4														-4								
							4				-4															
		4														-4										
					4				-4																	$x\{P_1\} = K(2,4) = K(4,2)$
				4														12				8				
	-2	-2	-2				4				-4					4					-4					
	-2	-2	2													8				4						
- E(4)					4										4											
				-4														-12				-8				$x\{P_2\} = K(5,7) = K(7,5)$
							-4							4												
			-4													-4										
					-4										-4											
						4				-4																$x\{P_3\} = K(8,1) = K(1,8)$
					-4							2			2				-4				4			
					-4			-4																		
			-4														-4									
													4											-4		
																			-4				-4			
								4				-2			-2											
																-4					-4					$x\{P_1\} = K(3,4) = K(4,3)$
													4											-4		
								-8	4			-4							12				-4			
								4				2			-6											
F(4)	4															-4				-4	8					$x\{P_2\} = K(6,7) = K(7,6)$
										4														-4		
								-8	4			-4							12				-4			
								4				2			-6											
																4				-4						
																		-4				-4				$x\{P_3\} = K(9,1) = K(1,9)$
														4											-4	
																	4			-4						
												2			-2											
	2	2	2				-4				4					-4					4					
				-4														-12				-8				

(continued)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
	2	2	-2													-8				-4					
					-4										-4										
							-4				4														
				4														4							
		-4														4									
E(1)					-4				4																
							4							-4											
				4														12				8			
			4													4									
					4										4										
					4							-2			-2				4				-4		
						-4				4															
					4			4																	
			4														4								
		-2																							
			-2																						
				-2																					
					-2							1			1								2		
			-2																						
		-2																							
				-2																					
K(1)					-2							1			1								2		
			2																						
			2																						
				-2														-4				-4			
					2							-1			-1								-2		
					-2																				
					-2																				
						-2																			
							-2																		
												2													
								-2																	
													2												
														2											
									2																
								2																	
										2															
-M(4)											2														
															2										
								2																	
										2															
														-2											
																	2								
																2									
																		2							
																			2						
								-8	4			-4							12				-4		
													4											-4	
								4				2			-6										
	4															-4				-4	9				

INDEP. COEF. NOS.

E(1):
x{P2} = K(4,8) = K(8,4)
x{P1} = K(1,5) = K(5,1)
x{P3} = K(7,2) = K(2,7)

K(1):
x{P1} = K(2,5) = K(5,2)
x{P2} = K(5,8) = K(8,5)
x{P3} = K(8,2) = K(2,8)

-M(4):
x{P1} = K(3,5) = K(5,3)
x{P2} = K(6,8) = K(8,6)
x{P3} = K(9,2) = K(2,9)

x{P1} = K(1,6) = K(6,1)
(continued)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
																			-4				-4		
													4											-4	
								4				-2			-2										
																-4					-4				
F(1)								-8	4			-4							12				-4		
										4														-4	
								4				2			-6										
																4				-4					
														4											-4
																		-4				-4			
																	4			-4					
												2			-2										
								-2																	
									-2																
										-2															
											-2														
								2																	
												-2													
													-2												
M(1)														-2											
								-2																	
															-2										
										-2															
														2											
																-2									
																	-2								
																		-2							
																			-2						
																				-2					
																					-2				
																						-2			
																							-2		
																					-2				
																				-2					
																						-2			
L(1)																							-2		
																				-2					
																				-2					
																						-2			
								-4	2			-2							4				-2		
								2				1			-1										
								2				1			-1										
																								-2	
																									-2

INDEP. COEF. NOS.

F(1):
$x\{P_2\} = K(4,9) = K(9,4)$
$x\{P_3\} = K(7,3) = K(3,7)$

M(1):
$x\{P_1\} = K(2,6) = K(6,2)$
$x\{P_2\} = K(5,9) = K(9,5)$
$x\{P_3\} = K(8,3) = K(3,8)$

L(1):
$x\{P_1\} = K(3,6) = K(6,3)$
$x\{P_2\} = K(6,9) = K(9,6)$
$x\{P_3\} = K(9,3) = K(3,9)$

D A T A   D E B U G G I N G   P R O G R A M

```
      DIMENSION LIST(500),JOIN(500,3), XA(3),YA(3),G(3),C(3),D(3),F(3),
     1LINK(200,3),MTRI(200,3),NODE(100),X(100),Y(100),B(3)
C     INPUT OF TOPOLOGY & BOUNDARY LISTS
C     ---------------------------------
      IRROR=0
      READ(8,100)LNAME
100   FORMAT(A6)
      IF(LNAME-'TOPOLO')1,2,1
1     WRITE(5,124)
      GO TO 56
2     DO 70 I=1,500
70    LIST(I)=-1
      READ(8,121,ERR=3)(LIS (I),I=1,500)
3     DO 71 J=500,1
      I=J
      IF(LIST(I))71,72,72
71    CONTINUE
72    REAU(0,100)LNAME
      IF(LNAME-'BOUNDA'),4,
      IF(LNAME-'NODAL  ')1,13,1
4     L=I-1
      WRITE(5,122)
      WRITE(5,125)(LIST(J),J=1,I)
C
C     ORDERING OF JOINS
C     -----------------
      DO 7 I=1,L
      IF(LIST(I)-LIST(I+1))5,,6
      WRITE(5,126)LIST(I)
126   FORMAT(2H ,'NODE',I6,'IS JOINED TO ITSELF. INVALID')
      IRROR=IRROR+1
5     JOIN(I,1)=LIST(I)
      JOIN(I,2)=LIST(I+1)
      GO TO 7
6     JOIN(I,1)=LIST(I+1)
      JOIN(I,2)=LIST(I)
7     JOIN(I,3)=0
      I=0
64    I=I+1
60    IF(JOIN(I,1)),,63
61    L=L-1
      IF(L-I)78
      DO 62 IA=I,L
      JOIN(IA,1)=JOIN(IA+1,1)
62    JOIN(IA,2)=JOIN(IA+1,2)
      GO TO 60
63    IF(JOIN(I,2))61,61
```

(continued)

```
      IF(L-I),,64
78    LA=L-1
      DO 12 I=1,LA
      M=I+1
      DO 12 K=M,L
65    IF(K-L),,12
      IF(JOIN(I,1)-JOIN(K,1))12,8,9
8     IF(JOIN(I,2)-JOIN(K,2))12,10,9
9     KEEP=JOIN(I,1)
      JOIN(I,1)=JOIN(K,1)
      JOIN(K,1)=KEEP
      KEEP=JOIN(I,2)
      JOIN(I,2)=JOIN(K,2)
      JOIN(K,2)=KEEP
      GO TO 12
10    L=L-1
      LA=L-1
      DO 11 IA=K,L
      JOIN(IA,1)=JOIN(IA+1,1)
11    JOIN(IA,2)=JOIN(IA+1,2)
      WRITE(5,101)JOIN(I,1),JOIN(I,2)
      GO TO 65
12    CONTINUE
      GO TO 2
101   FORMAT(1H ,'JOIN ',I4,'-',I4,' IS DUPLICATED IN TOPOLOGY LIST')
121   FORMAT( )
122   FORMAT(1H ,'TOPOLOGY LIST',/1X,13('-'))
123   FORMAT(1H ,'BOUNDARY CONNECTION LIST',/1X,24('-'))
124   FORMAT(1H ,'HEADING ERROROR ALPHA CHARACTER IN BOUNDARY OR TOPOLOG
     1Y LIST')
125   FORMAT(1H ,25I4)
C
C     BOUNDARY CONNECTION LIST
C     ------------------------
13    LA=I-1
      LB=0
      WRITE(5,123)
      WRITE(5,125)(LIST(J),J=1,I)
      DO 19 I=1,LA
      IF(LIST(I))19,19
      IF(LIST(I+1))19,19
      LB=LB+1
      IF(LIST(I)-LIST(I+1))14,15,15
14    NA=1
      NB=2
      GO TO 16
15    NA=2
      NB=1
16    LINK(LB,NA)=LIST(I)
      LINK(LB,NB)=LIST(I+1)
```

(continued)

```
      LINK(LB,3)=0
      DO 17 K=1,L
      IF(IABS(LIST(I)-JOIN(K,NA))+IABS(LIST(I+1)-JOIN(K,NB)))17,18,17
17    CONTINUE
      WRITE(5,102)LIST(I),LIST(I+1)
      IRROR=IRROR+1
      GO TO 19
18    JOIN(K,3)=JOIN(K,3)+1
19    CONTINUE
102   FORMAT(1H ,'BOUNDARY ',I4,'-',I4,' IS NOT INCLUDED IN TOPOLOGY LIS
     1T.')
C
C     TRIANGLE FORMULATION
C     --------------------
      NTRI=0
      DO 29 I=1,L
20    IF(JOIN(I,3)-2)21,29,28
21    KA=I+1
      DO 27 K=KA,L
      IF(JOIN(K,3)-2)22,27,27
22    IF(JOIN(K,1)-JOIN(I,1))28,23,28
23    MA=K+1
      DO 25 M=MA,L
      IF(IABS(JOIN(M,1)-JOIN(I,2))+IABS(JOIN(M,2)-JOIN(K,2)))24,26,24
24    IF(JOIN(M,1)-JOIN(I,2))25,25,27
25    CONTINUE
26    NTRI=NTRI+1
      JOIN(I,3)=JOIN(I,3)+1
      JOIN(K,3)=JOIN(K,3)+1
      JOIN(M,3)=JOIN(M,3)+1
      MTRI(NTRI,1)=JOIN(I,1)
      MTRI(NTRI,2)=JOIN(I,2)
      MTRI(NTRI,3)=JOIN(K,2)
      GO TO 20
27    CONTINUE
28    WRITE(5,103)(JOIN(I,J),J=1,3)
      IRROR=IRROR+1
29    CONTINUE
      DO 77 I=1,LB
      IF(LINK(I,3)),,77
      LINK(I,3)=1
      DO 76 NA=1,2
      NFOLO=LINK(I,NA)
73    DO 75 J=1,LB
      IF(I-J),75
      IF(LINK(J,1)-NFOLO)74,,74
      IF(LINK(J,3)),,76
      LINK(J,3)=1
      NFOLO=LINK(J,2)
      GO TO 73
```

(continued)

```
74    IF(LINK(J,2)-NFOLO)75,,75
      IF(LINK(J,3)),,76
      LINK(J,3)=1
      NFOLO=LINK(J,1)
      GO TO 73
75    CONTINUE
      WRITE(5,114)NFOLO
      IRROR=IRROR+1
76    CONTINUE
77    CONTINUE
114   FORMAT(1H ,'THERE IS A DISCONTINUITY IN THE BOUNDARY AT NODE ',I4)
103   FORMAT(1H ,'JOIN ',I4,'-',I4,' BOUNDS ',I2,' TRIANGLES.')
104   FORMAT(1H ,3I4)
C
C     NODAL POINT LIST
C     ----------------
      WRITE(5,115)
115   FORMAT(1H ,'NODAL POINT LIST'/1X,16('-'))
      READ(8,121)NSIZE
      DO 30 I=1,NSIZE
      NODE(I)=NSIZE+1
      X(I)=-1000
30    Y(I)=-1000
31    READ(8,121)I,XX,YY
      IF(I)33,33
      IF(I-NSIZE),,35
      IF(NODE(I)-NSIZE)32,32
      NODE(I)=I
      X(I)=XX
      Y(I)=YY
      GO TO 31
35    WRITE(5,127)
      GO TO 56
32    WRITE(5,106)I
      IRROR=IRROR+1
      GO TO 31
33    DO 34 I=1,NSIZE
      IF(NODE(I)-NSIZE)34,34
      WRITE(5,107)I
34    WRITE(5,116)I,X(I),Y(I)
116   FORMAT(1X,'NODE',I4,' IS AT X=',F10.4,' Y=',F10.4)
105   FORMAT(I4,2F10.0)
106   FORMAT(1H ,'NODE ',I4,' IS DUPLICATED IN NODAL POINT LIST.')
107   FORMAT(1H ,'NODE ',I4,' IS MISSING FROM NODAL POINT LIST. THIS IS
     1VALID.')
127   FORMAT(1H ,'INCORRECT SIZE SPECIFICATION.  FATAL ERROR')
C
C     TRIANGLE ARRANGEMENT AND COORDINATE VALIDITY CHECK
C     --------------------------------------------------
      ACC=3.
```

(continued)

```
      DO 54 I=1,NTRI
      DO 41 J=1,3
      N=MTRI(I,J)
      IF(X(N)+900.)53,53,40
40    XA(J)=X(N)
41    YA(J)=Y(N)
      DO 43 NA=1,3
      NB=1+NA-(NA/3)*3
      NC=2+NA-((NA+1)/3)*3
      G(NA)=XA(NA)-XA(NB)
      B(NA)=YA(NB)-YA(NA)
      IF(ABS(G(NA))+ABS(B(NA)))68,68
      C(NA)=YA(NA)*XA(NB)-XA(NA)*YA(NB)
      D(NA)=G(NA)*YA(NC)+B(NA)*XA(NC)+C(NA)
      F(NA)=SQRT(G(NA)*G(NA)+B(NA)*B(NA))
      IF(ABS(D(NA)/F(NA))-F(NA)/ACC)42,42,43
42    WRITE(5,108)(MTRI(I,K),K=1,3)
43    CONTINUE
      NA=1
      NC=3
      IF(G(NA))48,47,48
47    IF(B(NA)*(X(NA)-X(NC)))51,51,67
48    IF(D(NA))67,52,51
51    KEEP=MTRI(I,2)
      MTRI(I,2)=MTRI(I,3)
      MTRI(I,3)=KEEP
67    DO 46 J=1,NSIZE
      IF(X(J)+900.)46
      DO 44 K=1,3
      IF(J-MTRI(I,K))44,46,44
44    CONTINUE
      DO 45 NA=1,3
      E=G(NA)*Y(J)+B(NA)*X(J)+C(NA)
      IF(E*D(NA))46,45,45
45    CONTINUE
      WRITE(5,109)J,(MTRI(I,K),K=1,3)
      IRROR=IRROR+1
46    CONTINUE
      GO TO 54
52    WRITE(5,110)(MTRI(I,K),K=1,3)
      IRROR=IRROR+1
      GO TO 54
68    WRITE(5,113)(MTRI(I,J),J=1,3)
      IRROR=IRROR+1
      GO TO 54
53    WRITE(5,111)(MTRI(I,K),K=1,3),MTRI(I,J)
      IRROR=IRROR+1
54    CONTINUE
      IF(IRROR)57,57,55
55    WRITE(5,112)IRROR
```

(continued)

```
56    CALL EXIT
57    CONTINUE
108   FORMAT(1H ,'TRIANGLE ',3I5,' IS BADLY CONDITIONED BUT VALID.')
109   FORMAT(1H ,'NODE ',I4,' LIES INSIDE TRIANGLE ',3I5)
110   FORMAT(1H ,'TRIANGLE ',3I5,' IS INVALID.  NODES FORM A STRAIGHT LI
     1NE.')
111   FORMAT(1H ,'TRIANGLE ',3I5,' IS INVALID.  NODE ',I4,' IS NOT LISTE
     1D.')
112   FORMAT(1H ,'THE DATA ERROR LEVEL IS ',I5)
113   FORMAT(1H ,'NODES OF TRIANGLE',3I5,'COINCIDE')
D A T A   W R I T T E N   O N T O   F I L E   F O R   U S E   I N   T H E
M A I N   P R O G R A M
      WRITE(20)(X(I),Y(I),I=1,NSIZE)
      READ(8,121)UDL
      DO 66 I=1,NTRI
66    WRITE(20)(MTRI(I,J),J=1,3),UDL
      I=1
80    K=JOIN(I,1)
81    IF(K.NE.JOIN(I,1))GO TO 82
      WRITE(20)JOIN(I,1),JOIN(I,2)
      I=I+1
      GO TO 81
82    J=1
83    IF(K.NE.JOIN(J,2))GO TO 84
      WRITE(20)JOIN(J,2),JOIN(J,1)
84    J=J+1
      IF(J.LT.I)GO TO 83
      IF(I.LE.L)GO TO 80
      IF(K.EQ.JOIN(I-1,2))GO TO 85
      K=JOIN(I-1,2)
      GO TO 82
85    JOIN(I,1)=0
      JOIN(I,2)=0
      WRITE(20)JOIN(I,1),JOIN(I,2)
      STOP
      END
```

C O E F F I C I E N T   S O L U T I O N   P R O G R A M

```
      DIMENSION X(30),Y(30),IH(3,6,16,25),D(90),T(3,3,3),F(90),NBC(20)
     1,TS(3),TC(3),XA(3),YA(3),TA(3),E(3),G(4),NODE(50,3),Q(16),A(50,50)
P R E P A R A T I O N
      ICYC(I,J)=I+J-1-((I+J-2)/3)*3
      DO 80 I=1,3
      DO 80 J=2,3
      T(I,1,J)=0.
80    T(I,J,1)=0.
      DO 1 J=1,6
      DO 1 I=1,3
      DO 1 K=1,16
1     READ(8,101)(IH(I,J,K,L),L=1,25)
100   FORMAT()
101   FORMAT(25I3)
I N P U T   O F   S T A N D A R D   P L A T E   D A T A
      READ(8,100)NN,NE,US,UE,NU
      UI=(UE-US)/(NU-1)
      DO 2 I=1,NN
2     READ(8,100)X(I),Y(I)
      DO 3 NA=1,NE
3     READ(8,100)(NODE(NA,I),I=1,3)
      NM=3*NN
      READ(8,100)(F(I),I=1,NM)
      READ(8,100)MBC
      READ(8,100)(NBC(I),I=1,MBC)
P O I S S O N'S   R A T I O   C Y C L E
      DO 14 NB=1,NU
      DO 4 I=1,NM
      DO 4 J=1,25
4     A(I,J)=0.
      U=US+UI*(NB-1)
      DO 5 I=1,NN
      D(3*I-2)=X(I)*X(I)-U*Y(I)*Y(I)
      D(3*I-1)=-2.*U*Y(I)
5     D(3*I)=-2.*X(I)
T H E   E L E M E N T   C Y C L E
      DO 10 NA=1,NE
C A L C U L A T I O N   O F   D I M E N S I O N S
      DO 6 IA=1,3
      IB=ICYC(IA,2)
      IC=ICYC(IA,3)
      N1=NODE(NA,IA)
      N2=NODE(NA,IB)
      N3=NODE(NA,IC)
      TS(IA)=Y(N3)-Y(N2)
      TC(IA)=X(N3)-X(N2)
      XA(IA)=X(N1)
```

(continued)

```
      YA(IA)=Y(N1)
      TA(IA)=Y(N3)*X(N2)-Y(N2)*X(N3)
6     E(IA)=SQRT(TS(IA)*TS(IA)+TC(IA)*TC(IA))
      G(4)=E(1)+E(2)+E(3)
C A L C U L A T I O N   O F   I N S C R I B E D   C I R C L E   C E N T R E
      DEN=TC(1)/E(1)*TS(2)/E(2)+TC(2)/E(2)*TS(3)/E(3)+TC(3)/E(3)*TS(1)/E
     1(1)-TC(1)/E(1)*TS(3)/E(3)-TC(2)/E(2)*TS(1)/E(1)-TC(3)/E(3)*TS(2)/E
     2(2)
      XC=-(TA(1)/E(1)*TC(2)/E(2)+TA(2)/E(2)*TC(3)/E(3)+TA(3)/E(3)*TC(1)/
     1E(1)-(TA(1)/E(1)*TC(3)/E(3)+TA(2)/E(2)*TC(1)/E(1)+TA(3)/E(3)*TC(2)
     1/E(2)))/DEN
      YC=-(TA(1)/E(1)*TS(2)/E(2)+TA(2)/E(2)*TS(3)/E(3)+TA(3)/E(3)*TS(1)/
     1E(1)-(TA(1)/E(1)*TS(3)/E(3)+TA(2)/E(2)*TS(1)/E(1)+TA(3)/E(3)*TS(2)
     2/E(2)))/DEN
C A L C U L A T I O N   O F   P A R A M E T E R S
      DO 7 IA=1,3
      G(IA)=SQRT(G(4)-2.*E(IA))
F O R M A T I O N   O F   T R A N S F O R M A T I O N   M A T R I C E S
      TC(IA)=XA(IA)-XC
      TS(IA)=YA(IA)-YC
      TA(IA)=SQRT(TC(IA)*TC(IA)+TS(IA)*TS(IA))
      T(IA,2,2)=TC(IA)/TA(IA)
      T(IA,2,3)=TS(IA)/TA(IA)
      T(IA,3,2)=-TS(IA)/TA(IA)
      T(IA,3,3)=TC(IA)/TA(IA)
7     T(IA,1,1)=1.
      G(4)=SQRT(G(4))
F O R M A T I O N   O F   S O L U T I O N   E Q U A T I O N S
      DO 10 I=1,3
      DO 10 J=1,3
      DO 10 K=1,3
      NROW=(NODE(NA,I)-1)*3+K
      DO 10 L=1,3
      NCOL=(NODE(NA,J)-1)*3+L
      DO 10 M=1,3
      DO 10 N=1,3
T H E   T R I A N G L E   I S   T R A N S F O R M E D , M U L T I P L I E D
B Y   I T S   N O D A L   D I S P L A C E M E N T S   A N D   A   F A C T O R
T O   C O R R E C T   I T S   D I M E N S I O N S
      FACTOR=T(I,M,K)*T(J,N,L)*D(NCOL)/(E(1)*E(2)*E(3)*G(4))
      IF(M.GE.2)FACTOR=FACTOR*SQRT(E(1)*E(2)*E(3)/E(I))*2.
      IF(N.GE.2)FACTOR=FACTOR*SQRT(E(1)*E(2)*E(3)/E(J))*2.
      IF(ABS(FACTOR)-.00005)10
T H E   A P R O P R I A T E   V A R I A T I O N   O F   T H E   H O M O -
G E N E O U S   F U N C T I O N   I S   F O R M E D
      IJ=ICYC(I,3)
      I1=N
      J1=M+3
      IF(J-ICYC(I,3)),8,
      IJ=I
```

(continued)

```
      I1=M
      J1=N
      IF(I-J),8,
      J1=N+3
8     IC=0
      DO 9 M1=1,4
      M2=ICYC(M1,IJ)
      DO 9 N1=1,4
      IC=IC+1
      N2=ICYC(N1,IJ)
      IF(N1.GE.4)N2=4
      CON=E(M2)*G(N2)
      IF(M1.GE.4)CON=G(1)*G(2)*G(3)*G(4)/G(N2)
      DO 9 M3=1,25
A MATRIX OF COEFFICIENTS FOR THE 25 UN-
KNOWNS IS FORMED
9     A(NROW,M3)=A(NROW,M3)+FACTOR*CON*IH(I1,J1,IC,M3)
10    CONTINUE
      DO 11 I=1,NM
THE LOAD VECTOR IS INTRODUCED AND EQUAT-
IONS CORRESPONDING TO THE BOUNDARY CON-
DITIONS ARE OVERWRITTEN
11    A(I,26)=F(I)
      N=1
      DO 15 I=1,NM
      DO 12 J=1,26
12    A(I-N+1,J)=A(I,J)
      IF(N.GT.MBC)GO TO 15
      IF(I.EQ.NBC(N))N=N+1
15    CONTINUE
THE EQUATIONS ARE SOLVED AND THE SOLU-
TIONS ARE BACK-SUBSTITUTED TO FIND THE
VALUES OF ALL THE COEFFICIENTS
      CALL HSOLV(25,26,A,50,50)
      DO 14 I=1,3
      DO 14 J=1,6
      WRITE(5,100)I,J
      DO 13 K=1,16
      Q(K)=0.
      DO 13 L=1,25
13    Q(K)=Q(K)+IH(I,J,K,L)*A(L,26)
THE VALUES ARE FILED
      WRITE(20)(Q(K),K=1,16)
14    WRITE(5,102)(Q(K),K=1,16)
102   FORMAT(1H ,16E10.4)
THIS WHOLE OPERATION IS REPEATED FOR
VARIOUS VALUES OF POISSON'S RATIO
      END
```

A SIMPLE GAUSS JORDAN SOLUTION ROUTINE
FOR A FULL GENERAL MATRIX

```
      SUBROUTINE HSOLV(NROW,NCOL,A,IR,IC)
      DIMENSION A(IR,IC)
      DO 9 I=1,NROW
      K=I
      DO 3 L=I,NROW
      IF(ABS(A(K,I))-ABS(A(L,I)))3,3
      K=L
3     CONTINUE
      IF(A(K,I))4,,4
      WRITE(5,101)
101   FORMAT(1H ,'SINGULAR MATRIX')
      GO TO 16
4     DO 5 J=1,NCOL
      G=A(I,J)
      A(I,J)=A(K,J)
5     A(K,J)=G
      DIAG=A(I,I)
      DO 6 J=I,NCOL
6     A(I,J)=A(I,J)/DIAG
      DO 9 K=1,NROW
      IF(K-I)7,9,7
7     HEAD=A(K,I)
      DO 8 J=I,NCOL
8     A(K,J)=A(K,J)-A(I,J)*HEAD
9     CONTINUE
16    RETURN
      END
```

A GAUSS ROUTINE OPERATING ONLY ON HALF OF THE BAND WIDTH OF A SYMMETRICAL BANDED MATRIX

```
      SUBROUTINE HTGBND(IR,IC,NR,NC,NB,A)
      DIMENSION A(IR,IC)
      NA=NB+1
      N=NR-1
      DO 3 I=1,N
      IF(ABS(A(I,1))-1.0E-04)7,,
      DO 3 L=2,NB
      J=I+L-1
      IF(J.GT.NR)GO TO 3
      IF(ABS(A(I,L))-1.0E-04)3,,
      DO 1 K=NA,NC
1     A(J,K)=A(J,K)-A(I,K)*A(I,L)/A(I,1)
      ND=NB-L+1
      DO 2 K=1,ND
2     A(J,K)=A(J,K)-A(I,K+L-1)*A(I,L)/A(I,1)
3     CONTINUE
      DO 6 IA=1,NR
      I=NR-IA+1
      IF(ABS(A(I,1))-1.0E-04)7,,
      DO 4 K=NA,NC
4     A(I,K)=A(I,K)/A(I,1)
      DO 5 L=2,NB
      J=I-L+1
      IF(J.LE.0)GO TO 6
      DO 5 K=NA,NC
5     A(J,K)=A(J,K)-A(I,K)*A(J,L)
6     CONTINUE
      RETURN
7     WRITE(5,101)
101   FORMAT(1H ,'SINGULAR MATRIX')
      STOP
      END
```

A PROGRAM TO FIT THE CURVE WHICH INTRODUCES THE EFFECTS OF POISSON'S RATIO TO THE COEFFICIENTS

```
      DIMENSION A(10,4),B(3,6,16,10),C(50,50)
100   FORMAT()
      READ(8,100)US,UE,NI
      UI=(UE-US)/(NI-1)
      DO 1 NU=1,NI
      DO 1 I=1,3
      DO 1 J=1,6
1     READ(20)(B(I,J,K,NU),K=1,16)
      DO 4 I=1,3
      DO 4 J=1,6
      DO 4 K=1,16
      DO 2 NU=1,10
      U=US+UI*(NU-1)
      A(NU,1)=1.
      A(NU,2)=U
      A(NU,3)=U*U
2     A(NU,4)=B(I,J,K,NU)
REGRESSION TECHNIQUE IS APPLIED
      DO 3 IA=1,3
      DO 3 IB=1,4
      C(IA,IB)=0.
      DO 3 IC=1,10
3     C(IA,IB)=C(IA,IB)+A(IC,IA)*A(IC,IB)
      CALL HSOLV(3,4,C,50,50)
      DO 4 IA=1,3
4     B(I,J,K,IA)=C(IA,4)
      DO 5 NU=1,3
      DO 5 I=1,3
      DO 5 J=1,6
      WRITE(21)(B(I,J,K,NU),K=1,16)
5     WRITE(5,101)(B(I,J,K,NU),K=1,16)
101   FORMAT(1H ,10E12.4)
      END
```

A DECK TO UP DATE THE COEFFICIENT SOLUTION PROGRAM TO A FINITE ELEMENT USER PROGRAM

```
-1,2
(PREPARATION)
      DIMENSION T(3,3,3),TA(3),TC(3),TS(3),XA(3),YA(3),E(3),G(4)
     1,Q(3,6,16),F(9,11),B(30,11),C(100,5,4),X(100),Y(100),
     2NBC(100),BC(100),A(300,39),NODE(300,3),UDL(300,3)
-3,33
THE DESCRIPTION OF THE PROBLEM AND LOAD-
ING IS READ FROM DISC FILE OR CARDS.
THE COEFFICIENTS OF THE EXPLICIT MATRIX
ARE READ FROM DISC FILE
      READ(8,100)NREAD,MREAD,NN,NE,U,EY,H
      IF(U.LE.0.05)U=0.05
      IF(U.GE.0.4.OR.U.LE.0.05)WRITE(5,101)
101   FORMAT(1H ,'UNSTABLE VALUE OF POISSONS RATIO')
      DO 81 I=1,3
      DO 81 J=1,6
      DO 81 K=1,16
81    Q(I,J,K)=0.
      DO 1 NU=0,2
      DO 1 I=1,3
      DO 1 J=1,6
      READ(MREAD)(X(K),K=1,16)
      DO 1 K=1,16
1     Q(I,J,K)=Q(I,J,K)+X(K)*(U**NU)
      IF(NREAD.EQ.8)GO TO 83
      READ(NREAD)(X(I),Y(I),I=1,NN)
      GO TO 82
100   FORMAT()
83    READ(8,100)(X(I),Y(I),I=1,NN)
82    READ(8,100)NLOAD,MLOAD
      NBW=0
      DO 3 NA=1,NE
      DO 88 I=1,NLOAD
88    UDL(NA,I)=0.
      IF(NREAD.NE.8)GO TO 86
      READ(8,100)(NODE(NA,I),I=1,3),(UDL(NA,I),I=1,MLOAD)
      GO TO 87
86    READ(NREAD)(NODE(NA,I),I=1,3),(UDL(NA,I),I=1,MLOAD)
87    DO 3 I=1,3
      J=ICYC(I,2)
      IF(IABS(NODE(NA,I)-NODE(NA,J))-NBW)3,3
      NBW=IABS(NODE(NA,I)-NODE(NA,J))
3     CONTINUE
      NBW=3*NBW+3
      MM=NBW+NLOAD
```

(continued)

NOTE: -8,33 means: "In the coefficient evaluation program, replace cards no. 8 to no. 33 (inclusive) with the following cards:"

```
      NM=3*NN
      DO 4 I=1,NM
      DO 4 J=1,MM
4     A(I,J)=0.
      DO 5 J=1,NLOAD
5     READ(8,100)(A(I,NBW+J),I=1,NM)
      READ(8,100)MBC
      READ(8,100)(NBC(I),BC(I),I=1,MBC)
S T A R T   O F   E L E M E N T   B Y   E L E M E N T   C Y C L E
      DO 11 NA=1,NE
( T H E   D I M E N S I O N S , P A R A M E T E R S   &   T R A N S F O R -
M A T I O N   M A T R I C E S   A R E   F O R M E D   H E R E )
-67,75
P R O G R A M   L O O P S   F O R M   T H E   E L E M E N T   S T I F F N E S S
M A T R I X   I N   L O C A L   C O O R D I N A T E S   I N   A R R A Y   'F'
      DO 9 I=1,3
      DO 9 J=1,3
      DO 9 M=1,3
      MROW=(I-1)*3+M
      DO 9 N=1,3
      MCOL=(J-1)*3+N
      F(MROW,MCOL)=0.
      FACTOR=1./(E(1)*E(2)*E(3)*G(4))
-78,78
( T H E   F U N C T I O N   G E N E R A T O R   C O M E S   H E R E )
-97,119
9     F(MROW,MCOL)=F(MROW,MCOL)+FACTOR*CON*Q(I1,J1,IC)
H A L F   T H E   B A N D   W I D T H   O F   T H E   S T I F F N E S S
M A T R I X   I S   F O R M E D   I N   A R R A Y   'A'   A F T E R   T R A N S -
F O R M A T I O N   O F   A R R A Y   'F'
      DO 85 I=1,3
      DO 85 J=1,3
      DO 85 K=1,3
      NROW=(NODE(NA,I)-1)*3+K
      DO 85 L=1,3
      NCOL=(NODE(NA,J)-1)*3+L-NROW+1
      IF(NCOL.LE.0)GO TO 85
      DO 10 M=1,3
      MROW=(I-1)*3+M
      DO 10 N=1,3
      MCOL=(J-1)*3+N
10    A(NROW,NCOL)=A(NROW,NCOL)+T(I,M,K)*T(J,N,L)*F(MROW,MCOL)
85    CONTINUE
T H E   L O A D   V E C T O R   F O R   A   U N I F O R M   L O A D   I S
C A L C U L A T E D
      XCG=(XA(1)+XA(2)+XA(3))/3.
      YCG=(YA(1)+YA(2)+YA(3))/3.
      DO 11 J=1,NLOAD
      IF(UDL(NA,J)),11,
      DO 84 I=1,3
```

(continued)

```
      N=(NODE(NA,I)-1)*3
      D=UDL(NA,J)*G(1)*G(2)*G(3)*G(4)/12.
      A(N+1,NBW+J)=A(N+1,NBW+J)+0
      D=D*SQRT((2.*(E(1)*E(1)+E(2)*E(2)+E(3)*E(3))-3.*E(I)*E(I))/((XA(I)
     1-XCG)**2.+(YA(I)-YCG)**2.))
      A(N+2,NBW+J)=A(N+2,NBW+J)+D*(YA(I)-YCG)/10.
84    A(N+3,NBW+J)=A(N+3,NBW+J)-D*(XA(I)-XCG)/10.
11    CONTINUE
BOUNDARY CONDITION RESTRICTIONS ARE
APPLIED
      DO 14 N=1,MBC
      M=NBC(N)
      MBW=NBW-1
      DO 18 I=1,MBW
      DO 17 J=1,NLOAD
      IF(M-I)13,13
      A(M-I,NBW+J)=A(M-I,NBW+J)-BC(N)*A(M-I,I+1)
13    IF(NM-M-I)17
      A(M+I,NBW+J)=A(M+I,NBW+J)-BC(N)*A(M,I+1)
17    CONTINUE
      A(M-I,I+1)=0.
18    A(M,I+1)=0.
      A(M,1)=1.
      DO 14 J=1,NLOAD
14    A(M,NBW+J)=BC(N)
THE BANDED SYMMETRIC MATRIX IS SOLVED
      CALL HTGBND(300,34,NM,MM,NBW,A)
      WRITE(5,102)
102   FORMAT(1H ,'STRESS RESULTS ARE UNRELIABLE AT NODES WHICH LIE ON TH
     1E EDGE OF A PROBLEM')
THE RELATION BETWEEN EACH NODE AND ITS
SURROUNDING NODES IS READ FROM DISC FILE
      READ(NREAD)JC,JB
      NML=NLOAD+7
20    I=3
      JA=JC
21    XC=X(JB)-X(JC)
      YC=Y(JB)-Y(JC)
      XY=SQRT(XC*XC+YC*YC)
SUBSTITUTION INTO THE DISPLACEMENT
POLYNOMIAL TO FORM STRESS COEFF. MATRIX
      B(I-2,1)=XC*XC/XY
      B(I-2,2)=XC*YC/XY
      B(I-2,3)=YC*YC/XY
      B(I-2,4)=XC*XC*XC/XY
      B(I-2,5)=XC*XC*YC/XY
      B(I-2,6)=XC*YC*YC/XY
      B(I-2,7)=YC*YC*YC/XY
      B(I-1,1)=0.
      B(I-1,2)=XC
```

(continued)

```
      B(I-1,3)=2.*YC
      B(I-1,4)=0.
      B(I-1,5)=XC*XC
      B(I-1,6)=2.*XC*YC
      B(I-1,7)=3.*YC*YC
      B(I,1)=-2.*XC
      B(I,2)=-YC
      B(I,3)=0.
      B(I,4)=-3.*XC*XC
      B(I,5)=-2.*XC*YC
      B(I,6)=-YC*YC
      B(I,7)=0.
      DO 22 J=1,NLOAD
      B(I-2,7+J)=(A(3*JB-2,NBW+J)-A(3*JA-2,NBW+J)+XC*A(3*JA,NBW+J)-YC*A(
     13*JA-1,NBW+J))/XY
      B(I-1,7+J)=A(3*JB-1,NBW+J)-A(3*JA-1,NBW+J)
22    B(I,7+J)=A(3*JB,NBW+J)-A(3*JA,NBW+J)
23    READ(NREAD)JC,JB
      IF(JC.EQ.JA)GO TO 28
      IF(I.GE.7)GO TO 25
      DO 24 J=1,NML
24    B(7,J)=0.
      B(7,5)=1.
      B(7,6)=-1.
      WRITE(5,103)JA
103   FORMAT(1H ,'UNRELIABLE STRESS RESULTS AT NODE',I4)
```

R E G R E S S I O N   A N A L Y S I S   S O L V E S   R E S U L T I N G   R E C T A N G U L A R   M A T R I X

```
25    DO 26 J=1,7
      DO 26 K=1,NML
      F(J,K)=0.
      DO 26 L=1,I
26    F(J,K)=F(J,K)+B(L,J)*B(L,K)
      CALL HSOLV(7,NML,F,9,11)
      DO 27 J=1,NLOAD
```

S T R E S S E S   A R E   C A L C U L A T E D   F R O M   T H E   S O L U T I O N

```
      C(JA,1,J)=-(F(1,7+J)+U*F(3,7+J))/(1.-U*U)
      C(JA,2,J)=-(F(3,7+J)+U*F(1,7+J))/(1.-U*U)
      C(JA,3,J)=F(2,7+J)/(2.-2.*U)
      C(JA,4,J)=-(3.*F(4,7+J)+F(6,7+J))/(1.-U*U)
27    C(JA,5,J)=-(3.*F(7,7+J)+F(5,7+J))/(1.-U*U)
      IF(JC)29,29,20
28    I=I+3
      IF(I.LE.30)GO TO 21
      WRITE(5,104)JA
104   FORMAT(1H ,'THERE ARE MORE THAN TEN CONNECTIONS TO NODE',I4)
      GO TO 23
29    DO 31 K=1,NLOAD
```

R E S U L T S   A R E   W R I T T E N   U N D E R   H E A D I N G S

```
      WRITE(5,105)K
```

(continued)

```
105   FORMAT(1H1,53X,'LOAD VECTOR ',I1/54X,13('-')/1X,120('-')/3X,'NODE'
     1,5X,'VERTICAL',6X,'ROTATION',6X,'ROTATION',7X,'MOMENT',8X,'MOMENT'
     2,7X,'TWISTING',8X,'SHEAR',9X,'SHEAR'/4X,'NO.',6X,'DISPL.',7X,'ABOU
     3T X',7X,'ABOUT Y',6X,'IN X DIRN.',4X,'IN Y DIRN.',6X,'MOMENT',6X,'
     4IN X DIRN.',4X,'IN Y DIRN.'/12X,'(LENGTH)',9X,'(DIMENSIONLESS)',17
     5X,'(FORCE*LENGTH/LENGTH)',18X,'(FORCE/LENGTH)'/1X,120('-'))
      DO 31 I=1,NN
      IA=3*I-2
      IB=3*I
      DO 30 J=IA,IB
30    A(J,K)=A(J,N3W+K)*6./(EY*(H**3.))
31    WRITE(5,106)I,(A(J,K),J=IA,IB),(C(I,J,K),J=1,5)
106   FORMAT(1X,I4,3X,8(2X,E12.7))
```