ANALYSIS OF SHELLS AND FOLDED PLATE STRUCTURES BY DIFFERENT DISPLACEMENT FUNCTIONS

ANALYSIS OF SHELLS AND FOLDED PLATE STRUCTURES BY DIFFERENT DISPLACEMENT FUNCTIONS

By

MAUNG MAUNG WIN B.Sc.(Eng)

A Thesis

Submitted to the Faculty of Graduate Studies

in Partial Fulfilment of the Requirements

for the Degree

Master of Engineering

McMaster University

November 1970

MASTER OF ENGINEERING (1970)
(Mechanical Design)

McMASTER UNIVERSITY
Hamilton, Ontario

TITLE: ANALYSIS OF SHELLS AND FOLDED PLATE STRUCTURES BY DIFFERENT DISPLACEMENT FUNCTIONS

AUTHOR: MAUNG MAUNG WIN, B.Sc. (ENG) (Rangoon University)

SUPERVISOR: Dr. M.A. DOKAINISH

NUMBER OF PAGES: x, 143

SCOPE AND CONTENT:

Three different displacement functions are used in the theoretical analysis of shells of general nature.

An oblique truncated pyramid like structure, made of aluminum plates is used for illustration and experimental verification.

Theoretical values of deflections obtained from three different displacement functions and stresses are compared with the experimental results.

ABSTRACT

The finite element displacement method with triangular plate elements, is used in the present research program to establish an analytical approach for the analysis of shells and folded plate structures. The Tocher, Rawtani and Cowper displacement functions are used in this analysis and the theoretical displections u, v and w are found to compare satisfactorily with the experimental results.

Owing to the limited storage capacity of the computer, the method of tridiagonalization is used, rather than the method of direct stiffness assembly.

Special considerations for stiffness assembly are necessary; (a) all the elements meeting at a node lie in the same plane and (b) the nodal points lie on the fixed boundary.

Experimental values obtained from experiments on an oblique, truncated, pyramid made of aluminum compared well with the theoretical results based on the same structure.

ACKNOWLEDGEMENTS

The author gratefully acknowledges the assistance and guidance of Professor Dr. M.A. Dokainish and for his encouragement at every phase of the project.

The author feels indebted to the Department of Mechanical Engineering and the Canadian Armament Research and Development Establishment for giving an opportunity to work on this project.

The study leave granted by Rangoon Institute of Technology, Rangoon, and the financial assistance supported by the Canadian International Development Agency are gratefully acknowledged.

Finally the author wishes to thank the technicians from the Applied Dynamic Laboratory and Mrs. S. McQueen for typing the manuscript.

NOMENCLATURE

SYMBOL	DEFINITION
x, y, z	Global or system coordinates
x', y', z'	Local or member coordinates
u, v, w	Displacements along global axes
u', v', w'	Displacements along local axes
$\theta_x, \theta_y, \theta_z$	Rotational displacements about global axes
$\theta_{x'}, \theta_{y'}, \theta_{z'}$	Rotational displacements about local axes
$\{d\}$	Displacement vector of any point within the element.
$\{\delta\}_p^e$	Nodal displacement vector due to in-plane forces
$\{\delta\}_b^e$	Nodal displacement vector due to bending.
$[N]$	Shape function of the element
$\{\varepsilon\}$	Strain vector
$\{\sigma\}$	Stress vector
$[D]$	Elasticity matrix
$[K_e']_p$	In-plane Stiffness matrix of element (6x6)
$[K_e']_b$	Bending Stiffness matrix of element (9x9)or(18x18)
$[K_e']$	Element Stiffness matrix with respect to local coordinates (18x18) or (27x27)
$[K_e]$	Element stiffness matrix with respect to global coordinates (18x18) or (27x27)
$[T]$	Transformation matrix (18x18) or (27x27)
$[R]$	Rotational matrix (9x9)

Symbol	Definition
l,m,n	Direction cosines of local axes
t	Thickness of plate element
μ	Poisson's ratio
E	Young's modulus of plate material
$\{F\}$	External force vector
a,b,c	Triangular element dimensions, Figure-22
α_i	Coefficients of polynomial expression
D	Flexural rigidity of isotropic plate $= \dfrac{Et^3}{12(1-\mu^2)}$
$F(m,n)$	Modified Euler's beta function
m_i, n_i	Exponents of x', y' in i^{th} term of polynomial expression for w.
θ	Angle between global x and local x' axes.
$[T_1]$	Function of coordinates of finite element (18x20)
$[T_2]$	Function of coordinates of finite element (20x20)
$[T_3]$	Matrix consists of the first 18 columns of $[T_2]^{-1}$
U_e	Strain energy of plate bending
w_{xx}, w_{xy}, w_{yy}	Second derivatives of w about global axes
$w_{x'x'}, w_{x'y'}, w_{y'y'}$	Second derivatives of w about local axes
$[K_i]$	Stiffness matrix of each partition on the diagonal of tridiagonalized matrix
$[C_i]$	Stiffness matrix of the coupling term on off diagonal of tridiagonalized matrix
$[\overline{K}_i]$	Modified stiffness matrix
$\{\overline{P}_i\}$	Modified load vector

Symbol	Definition
$\{F_{ip}\}$	Force vector at node i due to in-plane forces
$\{F_{ib}\}$	Force vector at node i due to bending forces

TABLE OF CONTENTS

	page
ABSTRACT	iii
ACKNOWLEDGEMENTS	iv
NOMENCLATURE	v
LIST OF ILLUSTRATIONS	x
1.0 INTRODUCTION	1
2.0 PHYSICAL MODEL	6
3.0 THEORETICAL ANALYSIS	7
3.1 Introduction	7
3.2 Displacement Functions	9
3.3 Triangular Plate Element Subjected to In-Plane Forces	12
3.4 Triangular Plate Element Subjected to Bending Forces	15
3.5 Grouping of In-Plane and Bending Stiffness Matrices	18
3.6 Transformation to Global Coordinate System	21
3.7 Assembly of Overall Stiffness Matrix	23
3.8 Solution of Equations	24
3.9 Formation of the Overall Stiffness Matrix in Tridiagonalized Partitions	25
4.0 EXPERIMENTAL ANALYSIS	29
4.1 Introduction	29
4.2 Loading Device	29
4.3 Displacement Transducer	29
4.4 Stress Calculation	31
5.0 RESULTS AND CONCLUSIONS	32
FIGURES	34b
TABLES	52a
BIBLIOGRAPHY	77a

		page
APPENDICES		80a
I	Stiffness Matrix of a Triangular Element in Plane Stress	81
II	Stiffness Matrix of a Triangular Element in Bending Using the Tocher Displacement Function	83
III	Derivation of Elastic Properties of Triangular Element in Bending Using the Rawtani Displacement Function.	85
IV	Derivation of Elastic Properties of Triangular Element in Bending Using the Cowper Displacement Function	90
V	Computer Program Documentation	97
VI	Computer Programs	105
VII	List of Equipment	142

LIST OF ILLUSTRATIONS

		page
1.	GRAPHS	35-40
2.	ORTHOGRAPHIC PROJECTIONS OF THE STRUCTURE	41
3.	ORIENTATION OF THE GLOBAL AXES	42
4.	DIVISION OF TRIANGULAR ELEMENTS	43,44
5.	IDEALIZATION OF THE SHELL	47
6.	PARTITIONING OF THE STRUCTURE	48
7.	OVERALL PICTURE OF SET UP	49-51

1.0 INTRODUCTION

The finite element method has proved to be an extremely powerful tool in the analysis of discrete and continuous structures consisting of one, two and three dimensional continua. The basic concept of the finite element method states that every structure may be considered to be an assemblage of individual structural components or elements. The structure must consist of a finite number of joints or nodal points. The finite element analysis may be classified into three phases, namely structural idealization, evaluation of element properties, and analysis of element assemblage. Careful judgement is necessary in making the structural idealization because the analysis is actually performed on this substitute structure and the results can be valid only when the behaviour of the substitute structure simulates the actual structure.

The finite element method provides a unified approach to the analysis of any type of structure, and any combination of one, two and three dimensional elements of different characteristics. Problems consisting of openings, anisotropy and variation of thickness are no longer of consequence, once general programs are written [1].* This method was developed originally by the aircraft industry [2] in response to a procedure which could provide a solution for extremely complex air frame configurations.

* Number in square brackets indicates references in Bibliography p. 77a.

It was observed that analysis of mechanical engineering structures has lagged to some extent, because they are much more difficult to categorise, and the unified approach to the analysis is not available. Only with the advent of the finite element method and high speed digital computers, has it become feasible to analyse such complex structures.

The present project has the general objective of offering a wide scope for the finite element method in its application to shells and folded plate structures. The solutions to these structures may be obtained by using different displacement functions such as those given by Tocher, Rawtani, Cowper etc.

Previous work in this area on spatial frames (one dimenstional beam like structure) was done by Tiwari [3] and Raghava [4]; an optimization study was done by Garunathan [5]; and the static analysis of folded plate structures was done by Bhat [6].

The present work is to study the deformation and the stress distribution of a folded plate structure by using different displacement functions of Tocher, Rawtani and Cowper, within the triangular element. Although many finite elements for plate bending have been developed in recent years, only rectangular elements have so far met the requirement with high accuracy and good convergence. But rectangular elements are, however, not suitable for a great variety of boundary conditions, and there is a need for a general triangular element having adequate accuracy and convergence properties [7].

If the force-displacement relations at the nodes of the finite elements can be found and expressed in a matrix form, the overall stiffness matrix can be obtained by superposition of the element stiffness matrices. If the assumed displacement function (s) ensures certain criteria; namely, continuity of deflection along the common edges between adjacent elements, continuity of transverse slopes and the size of subdivision of element decreases, then the deformational behaviour of the idealised structure converges to that of the continuous structure.

The idealization concept was first introduced by Turner et al [8] in 1956. This method is applicable to non-structural problems such as heat flow, fluid flow, and distribution of electric and magnetic potential. A book written by Zienkiewicz and Cheung [1] is comprehensive and very useful in these respects.

For any arbitrary boundary condition, triangular elements are the most suitable for structural idealization. This concept was originally suggested by Greene et al [9] in 1961, but the approach was not completely successful owing to the lack of good bending stiffness matrix for triangular element. A discussion of the present method using different displacement functions is given in section 3.0.

A shell structure may be considered to be made up of small flat elements interconnected at a finite number of nodal points. The elements in a shell will be subjected to both in-plane and bending forces. For a flat element these forces cause independent deformations, provided that these deformations are small. Hence, the stiffness matrix of the element can be evaluated separately for in-plane and bending forces, and combined together to establish the stiffness matrix for the general case.

A good convergence to the true solution depends on a reasonably refined mesh size. Accordingly, the number of equations to be solved and hence the stiffness matrix of the entire structure, becomes very large. This fact is especially true for shell problems when both in-plane and bending degrees of freedom are assigned to each node, but then the limited storage capacity of CDC 6400 computer does not permit the use of as fine a mesh as can be used for such problems. Hence special techniques using the sparse nature and symmetry of the overall stiffness matrix must be introduced. The computation is based on a partitioning technique and the solution of the equations is achieved by a method of recursion [1].

When all the elements joining at a particular node are in one plane, as often happens in a folded plate structure, a difficulty arises [1]. In this case, the equations corresponding to the particular node become linearly dependent, and the stiffness matrix becomes singular due to the omission of the rotation perpendicular to the plane. This difficulty is overcome by assembling part of the stiffness matrix corresponding to such nodes in local coordinates.

It is noticed that no contribution to the stiffness matrix is made for the nodal points on fixed boundaries by the

Tocher and Rawtani displacement functions; however, special care must be taken for the Cowper displacement function because normal curvatures will not be always zero at such nodes.

In order to verify the validity of the finite element method for shell problems, an oblique, four sided, truncated pyramid was made of aluminum plate. Deflections at certain nodal points were measured experimentally and these compared well with the theoretical values obtained by the three given displacement functions. Similarly the theoretical stresses at the centroid of any arbitrarily chosen element compared satisfactorily with the experimental stresses.

2.0 PHYSICAL MODEL

The results obtained theoretically need to be checked for validity. For this purpose, an oblique four faced truncated pyramid, made of aluminum was built and tested as mentioned earlier. Figures 14, 15 and 16 show the overall picture of the structure from two different angles and Figure 13 illustrates the details and dimensions in the orthographic projections. The obliquity of the structure ensures assymmetry of static and dynamic response. The faces are 1/8 inch thick and are welded together at the edges. A top plane 4 1/2" x 4 1/2" x 1/2" thick plate was welded at the top of the structure.

The base of the structure was assumed to be rigidly fixed to the foundation in the theoretical analysis. It is extremely important to the accuracy of the results that the deflections at the base are small compared to the relative displacements between nodal points or structure due to applied load. Considerable rigidity was achieved by bolting down the base of the structure to 1" thick steel plate, and four heavy cast iron blocks 6" x 6" in cross-section. 1" thick steel cover straps with lock washers were used to hold the base of the structure on the heavy cast iron blocks. Before conducting the experiment, the base rigidity was checked by a dial gauge and found to be satisfactory. The structure was loaded on the top plate in each of the three orthogonal x,y and z directions respectively.

3.0 THEORETICAL ANALYSIS

3.1 INTRODUCTION

The basic concept of the finite element method is that conventional engineering structures can be visualized as an assemblage of structural elements interconnected at a discrete number of nodal points. If the force-displacement relationships for each individual element are known, it is possible to analyse the behaviour of the assembled structure by available techniques in the structural analysis. It is important to obtain element stiffness matrix and it was discussed in detail in references [1] and [10].

The procedure for deriving the element stiffness matrix is given by the following steps.

(a) The continuum is separated by imaginary lines or surfaces into a number of 'finite elements'.

(b) the elements are assumed to be interconnected at a discrete number of nodal points situated on their boundaries.

(c) A function(s) is chosen to define uniquely the state of displacement within each finite element in terms of its nodal displacements.

$$\left.\begin{aligned} \{d\} &= [N_i N_j N_k --] \begin{Bmatrix} \delta_i \\ \delta_j \\ \delta_k \\ \vdots \end{Bmatrix} \\ \{d\} &= [N]\{\delta\}^e \end{aligned}\right\} \qquad (3.1.1)$$

(d) The displacement function defines the state of strain in terms of nodal displacements, which again define the state of stress within the element and on its boundaries.

When displacements within the element are known, it is easy to find strain at any point on the element by the relation,

$$\{\varepsilon\} = [B]\{\delta\}^e \quad (3.1.2)$$

(e) The stresses can be calculated accordingly, using the linear elastic relation between stresses and strains i.e.

$$\{\sigma\} = [D][B]\{\delta\}^e \quad (3.1.3)$$

(f) The stiffness matrix $[K_e]$ of the finite element is obtained by equating the internal work done and the external work done.

$$[K_e] = \int_v [B]^T [D][B] dv \quad (3.1.4)$$

= stiffness matrix of the element, taking integration over the whole volume of the element.

The characteristics of an individual element can be conveniently established in a coordinate system [1] which is different from the system in which external forces and displacements of the structure will be measured. Hence local or member coordinates will be used for every element, and transformation of the force and displacement components to global or common coordinates is necessary before an assembly of the stiffness matrix of the structure is made.

It can be shown that the element stiffness matrix in global coordinates is related to the element stiffness matrix in local coordinates as follows

$$[K_e] = [T]^T [K'_e] [T] \tag{3.1.5}$$

Once all the element stiffness matrices are derived in the global coordinates, the general procedure of the assembly of the overall stiffness matrix of the structure, and the solution of the equations will follow a standard structural routine.

3.2 DISPLACEMENT FUNCTIONS

The accuracy of the solution by the finite element method depends mainly on the physical approximation or idealization, and the type of displacement function used within the finite elements. It is rather difficult to assume the displacement function which should be able to represent the true displacement distribution as closely as possible. The result will tend to the correct one, provided the following conditions as given by Bazeley et al [11] are satisfied.

(a) The displacement function should be such that it does not permit straining of an element when the nodal displacements are caused by 'rigid' body displacements. Otherwise, constant strain conditions will not prevail as element gets smaller in size.

(b) The assumed displacement function should ensure continuity of deflection and transverse slope along common boundaries between adjacent elements.

At present, various conforming and non-conforming expressions for triangular elements have been in use. The non-conforming shape function ensures continuity of transverse deflection along common edges between adjacent elements, but not of transverse slope, whereas the conforming shape functions satisfy the continuity of transverse slope and deflection along common boundaries [12].

In general, the finer the mesh, the more realistic will be the results, but the deformations will not necessarily converge to the correct values even with an infinitesimal mesh size, unless the deformation patterns within the element are properly chosen [13].

The most commonly used cubic polynomial expression for transverse deflection in x and y is as follows

$$w(x,y) = \alpha_1 + \alpha_2 x + \alpha_3 y + \alpha_4 x^2 + \alpha_5 xy + \alpha_6 y^2 + \alpha_7 x^3 + \alpha_8 x^2 y + \alpha_9 xy^2 + \alpha_{10} y^3$$

The expression in x and y involves ten arbitrary coefficients and since only nine degrees of freedom are assigned to each element (three to each node), a certain assumption, regarding one of the coefficients must be established.

Adini [14] assumed the coefficient of the twisting term xy to be zero, hence the expression for transverse deflection becomes,

$$w'(x',y') = \alpha_1 + \alpha_2 x' + \alpha_3 y' + \alpha_4 x'^2 + \alpha_5 y'^2 + \alpha_6 x'^3 + \alpha_7 x'^2 y' + \alpha_8 x' y'^2 + \alpha_9 y'^3$$

Tocher [15] suggested that the twisting terms x^2y and xy^2 be combined and assigned the same coefficient, so that nine coefficients which correspond to nine degrees of freedom in the displacement function becomes,

$$w'(x',y') = \alpha_1 + \alpha_2 x' + \alpha_3 y' + \alpha_4 x'^2 + \alpha_5 x'y' + \alpha_6 y'^2 + \alpha_7 x'^3 + \alpha_8 (x'^2 y' + x'y'^2) + \alpha_9 y'^3$$

Rawtani [12], after carefully selecting the local coordinates of the finite element which satisfies transverse slope continuity along one edge of the element, suggested that the coefficient of the twisting terms xy^2 could be zero. Thus his expression for the displacement function is

$$w'(x'y') = \alpha_1 + \alpha_2 x' + \alpha_3 y' + \alpha_4 x^2 + \alpha_5 x'y' + \alpha_6 y'^2 + \alpha_7 x'^3 + \alpha_8 x'^2 y' + \alpha_9 y'^3$$

Full slope and deflection compatibility can be achieved by dividing the triangular element into three subtraingles and choosing the subelement local coordinates as suggested by Tocher and Clough [13]. Monotonic convergence to the true results is obtained by this procedure as the elements are refined, but for coarse subdivisions of structure the results obtained will be much inferior to that of non conforming shape functions mentioned earlier.

Cowper et al [16] recently introduced a conforming shape function based on a fifth degree polynomial expression for w' which gives the best convergence.

$$w'(x'y') = \alpha_1 + \alpha_2 x' + \alpha_3 y' + \alpha_4 x'^2 + \alpha_5 x'y' + \alpha_6 y'^2 + \alpha_7 x'^3$$
$$+ \alpha_8 x'^2 y' + \alpha_9 x'y'^2 + \alpha_{10} y'^3 + \alpha_{11} x'^4 + \alpha_{12} x'^3 y'$$
$$+ \alpha_{13} x'^2 y'^2 + \alpha_{14} x'y'^3 + \alpha_{15} y'^4 + \alpha_{16} x'^5 + \alpha_{17} x'^3 y'^2$$
$$+ \alpha_{18} x'^2 y'^3 + \alpha_{19} x'y'^4 + \alpha_{20} y'^5$$

This was developed for a right angled triangular element and it requires, six degrees of freedom (transverse deflection w', two first derivatives, and three second derivatives of w'), at each node.

Cowper et al[7] again introduced the same fifth degree expression applicable to any general triangular element and it satisfies the conditions of the conforming shape function. Monotonic convergence to the true result is ensured and it gives best convergence to the solution. When a triangular element is considered as part of a shell, in-plane forces also come into play, bringing three more degrees of freedom at each node. Hence nine degrees of freedom are required for this displacement function at each node and obviously only coarse mesh can be used which results in poor representation of a shell.

3.3 TRIANGULAR PLATE ELEMENT SUBJECTED TO IN-PLANE FORCES

In the triangular element, the displacements due to in-plane forces at the nodal points i,j,k can be written in matrix notation as follows

$$\{\delta\}_p^e = \begin{Bmatrix} \delta_i \\ \delta_j \\ \delta_k \end{Bmatrix} \qquad (3.3.1)$$

The in-plane forces produce two linear displacement components at each node, one in the x-direction and the other in the y-direction. Hence the displacements at a particular node i are

$$\{\delta_i\}_p = \begin{Bmatrix} u_i \\ v_i \end{Bmatrix} \qquad (3.3.2)$$

Similarly the six components of element displacement at three nodes in matrix form are as follows

$$\{\delta\}_p^e = \begin{Bmatrix} u_i \\ v_i \\ u_j \\ v_j \\ u_k \\ v_k \end{Bmatrix} \qquad (3.3.3)$$

The corresponding in-plane force vector acting at the nodal points in the x and y direction in matrix form is as follows:

$$\{F\}_p^e = \begin{Bmatrix} P_i \\ P_j \\ P_k \end{Bmatrix} = \begin{Bmatrix} P_{xi} \\ P_{yi} \\ P_{xj} \\ P_{yj} \\ P_{xk} \\ P_{yk} \end{Bmatrix} \qquad (3.3.4)$$

The linear dispalcements within the triangular element can be expressed by two linear polynomials.

$$\begin{aligned} u &= \alpha_1 + \alpha_2 x + \alpha_3 y \\ v &= \alpha_4 + \alpha_5 x + \alpha_6 y \end{aligned} \qquad (3.3.5)$$

The strain vector at any point within the element which contributes to internal work can be written in matrix notation.

$$\{\varepsilon\} = \begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{xy} \end{Bmatrix} = \begin{Bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \end{Bmatrix} \tag{3.3.6}$$

As mentioned earlier in the general procedure, the strain matrix can be written as

$$\{\varepsilon\} = [B]\{\delta\}_p^e \tag{3.3.7}$$

where [B] is independent of coordinates of a point within the element, since the polynomials are of first degree. Hence strains are constant throughout the element which is the required criterion satisfied by the shape function [1].

Once the strains are known, the stresses can be found by the following relation

$$\{\sigma\} = [D]\{\varepsilon\} \tag{3.3.8}$$

$$\{\sigma\} = [D][B]\{\delta\}_p^e \tag{3.3.9}$$

From the principle of virtual work, the stiffness matrix due to in-plane forces can be worked out explicity in terms of element properties and the nodal coordinates as follows.

$$[K_e]_p = \int_v [B]^T[D][B]dv \tag{3.3.10}$$

The size of this stiffness matrix is 6x6, as two degrees of freedom are assigned at each node for the triangular element.

Since the displacement function is linear, the displacement variation along the boundaries will also be linear; hence, the displacement of any point along the boundary will therefore depend on the displacement variation of the two nodes at the end of each edge, thereby ensuring displacement compatibility along the common boundary of two adjacent triangular elements.

The two-dimensional elastic problems by the finite element method were the first successful examples and the results were found to be quite satisfactory even with coarse subdivisions of the structure [1].

3.4 TRIANGULAR PLATE ELEMENT SUBJECTED TO BENDING FORCES

According to Kirchhoff's hypothesis, the displacements u and v parallel to the x-y plane of the plate, for plate bending, are related to the normal displacement w by [17]

$$\left.\begin{aligned} u &= -z \frac{\partial w}{\partial x} \\ v &= -z \frac{\partial w}{\partial y} \end{aligned}\right\} \qquad (3.4.1)$$

It is assumed that the midplane of the plate is undeformed. Compatibility conditions between elements require both continuity of displacement w and continuity of transverse slope. The required continuity between elements can be obtained if second derivatives of w are considered as degrees of freedom [18].

Determination of the shape function for bending is much more complex. It is rather difficult to maintain both continuity of w and continuity of the transverse slope between elements, because computational difficulties often arise disproportionately fast [1]. Convergence to the results may still be found, if the shape function satisfies the constant strain criterion [1].

For bending, each nodal point is usually assigned three degrees of freedom; namely w, w_x, and w_y. When curvatures or second derivatives are considered as degrees of freedom, the number of degrees of freedom at each node becomes six; namely w, w_x, w_y, w_{xx}, w_{xy}, and w_{yy}.

Thus the deflections and generalized forces due to bending at a particular node i of a triangular element in matrix notation are respectively

$$\{\delta_i\}_b = \begin{Bmatrix} w_i \\ \theta_{xi} \\ \theta_{yi} \end{Bmatrix} = \begin{Bmatrix} w_i \\ w_{xi} \\ w_{yi} \end{Bmatrix} \tag{3.4.2}$$

or

$$\{\delta_i\}_b = \begin{Bmatrix} w_i \\ \theta_{xi} \\ \theta_{yi} \\ \theta_{xxi} \\ \theta_{xyi} \\ \theta_{yyi} \end{Bmatrix} = \begin{Bmatrix} w_i \\ w_{xi} \\ w_{yi} \\ w_{xxi} \\ w_{xyi} \\ w_{yyi} \end{Bmatrix} \tag{3.4.3}$$

$$\{F_i\}_b = \begin{Bmatrix} P_{zi} \\ M_{xi} \\ M_{yi} \end{Bmatrix} \tag{3.4.4}$$

or

$$\{F_i\}_b = \begin{Bmatrix} P_{zi} \\ M_{xi} \\ M_{yi} \\ M_{xxi} \\ M_{xyi} \\ M_{yyi} \end{Bmatrix} \tag{3.4.5}$$

if a right handed system of axes is assumed. Similarly element displacements at three nodal points in matrix notation are

$$\{\delta\}_b^e = \begin{Bmatrix} \delta_i \\ \delta_j \\ \delta_k \end{Bmatrix} = [A]\{\alpha\} \tag{3.4.6}$$

where [A] is function of coordinates of nodal points. The strain vector can be expressed as

$$\{\varepsilon\} = \begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{xy} \end{Bmatrix} = \begin{Bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \end{Bmatrix} = -Z \begin{Bmatrix} \frac{\partial^2 w}{\partial x^2} \\ \frac{\partial^2 w}{\partial y^2} \\ 2\frac{\partial^2 w}{\partial x \partial y} \end{Bmatrix} \tag{3.4.7}$$

$$\{\varepsilon\} = [H]\{\alpha\} \tag{3.4.8}$$

$$\{\varepsilon\} = [H][A]^{-1}\{\delta\}_b^e = [B]\{\delta\}_b^e \tag{3.4.9}$$

Similarly the stress vector becomes

$$\{\sigma\} = [D]\{\varepsilon\} = [D[[B]\{\delta\}_b^e \tag{3.4.10}$$

Using the principle of virtual work and equating the internal and external work done, the bending stiffness matrix of the triangular element can be worked out as follows

$$[K_e]_b = [A^{-1}]^T \int_V [H]^T [D][H] dv [A^{-1}] \tag{3.4.11}$$

The inverse of [A] must be obtained numerically for each element. The matrix product inside the volume integral can be explicitly carried out. It contains second degree terms in x and y and can be easily integrated over the area of the triangle. Appendices I to IV give the expressions necessary for the Tocher, Rawtani and Cowper displacement functions.

3.5 GROUPING OF IN-PLANE AND BENDING STIFFNESS MATRICES

An element in shell structure is subjected both to in-plane and bending forces. If the element is assumed flat, deformations caused by these forces are independent of one another. Hence the stiffness matrices of the element due to in-plane and bending can be evaluated separately, and the combined stiffness matrix of the element due to the combined in-plane and bending forces can be grouped when proper order is maintained.

The forces and deflections of an element in in-plane deformation are related in matrix notation as follows.

$$\{F\}_p^e = [K_e]_p\{\delta\}_p^e \tag{3.5.1}$$

At a particular node i

$$\{F_i\}_p = \begin{Bmatrix} P_{xi} \\ P_{yi} \end{Bmatrix} \tag{3.5.2}$$

$$\{\delta_i\}_p = \begin{Bmatrix} u_i \\ v_i \end{Bmatrix} \tag{3.5.3}$$

Similarly the forces and deflections of an element in bending are related in matrix notation as follows.

$$\{F\}_b^e = [K_e]_b\{\delta\}_b^e \tag{3.5.4}$$

At a particular node i

$$\{F_i\}_b = \begin{Bmatrix} P_{zi} \\ M_{xi} \\ M_{yi} \end{Bmatrix} \tag{3.5.5}$$

$$\{\delta_i\}_b = \begin{Bmatrix} w_i \\ \theta_{xi} \\ \theta_{yi} \end{Bmatrix} \tag{3.5.6}$$

When the curvatures are considered as degrees of freedom, the deflection and force vector are respectively,

$$\{\delta_i\}_b = \begin{Bmatrix} w_i \\ \theta_{xi} \\ \theta_{yi} \\ \theta_{xxi} \\ \theta_{xyi} \\ \theta_{yyi} \end{Bmatrix} \qquad (3.5.7)$$

$$\{F_i\}_b = \begin{Bmatrix} P_{zi} \\ M_{xi} \\ M_{yi} \\ M_{xxi} \\ M_{xyi} \\ M_{yyi} \end{Bmatrix} \qquad (3.5.8)$$

The rotation θ_z is not present in either mode. However it is necessary to consider θ_z and the fictitious couple M_z, before transformation of the stiffness matrix to the global system. Since they do not enter the minimization procedure, appropriate zeros are inserted into the rows and columns of the combined stiffness matrix of the element corresponding to θ_z.

Hence at a particular node i

$$\{\delta_i\} = \begin{Bmatrix} u_i \\ v_i \\ w_i \\ \theta_{xi} \\ \theta_{yi} \\ \theta_{zi} \end{Bmatrix} \qquad (3.5.9)$$

When the curvatures are considered as degrees of freedom, the deflection vector becomes

$$\{\delta_i\} = \begin{Bmatrix} u_i \\ v_i \\ w_i \\ \theta_{xi} \\ \theta_{yi} \\ \theta_{zi} \\ \theta_{xxi} \\ \theta_{xyi} \\ \theta_{yyi} \end{Bmatrix} \tag{3.5.10}$$

or

$$\{\delta_i\} = \begin{Bmatrix} \delta_{ip} \\ \hline \delta_{ib} \\ \hline \theta_{zi} \end{Bmatrix} \tag{3.5.11}$$

The corresponding generalized forces are

$$\{F_i\} = \begin{Bmatrix} P_{xi} \\ P_{yi} \\ P_{zi} \\ M_{xi} \\ M_{yi} \\ M_{zi} \\ M_{xxi} \\ M_{xyi} \\ M_{yyi} \end{Bmatrix} \tag{3.5.12}$$

or

$$\{F_i\} = \left\{\begin{array}{c} F_{ip} \\ \hline F_{ib} \\ \hline M_{zi} \end{array}\right\} \qquad (3.5.13)$$

Hence the combined stiffness matrix of a particular member m at a particular node n, will be made up of the following submatrices.

$$[K_{mn}] = \left[\begin{array}{cc|cccc|c} \multicolumn{2}{c|}{[K_{mn}]_p} & 0 & 0 & 0 & & 0 \\ & & 0 & 0 & 0 & & 0 \\ \hline 0 & 0 & & & & & 0 \\ 0 & 0 & & [K_{mn}]_b & & & 0 \\ 0 & 0 & & & & & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & & 0 \end{array}\right] \quad \text{or} \quad \left[\begin{array}{cc|cccccc|c} \multicolumn{2}{c|}{[K_{mn}]_p} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & & & & & & & 0 \\ 0 & 0 & & & & & & & 0 \\ 0 & 0 & & & [K_{mn}]_b & & & & 0 \\ 0 & 0 & & & & & & & 0 \\ 0 & 0 & & & & & & & 0 \\ 0 & 0 & & & & & & & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]$$

[6x6] [9x9] (3.5.14)

3.6 TRANSFORMATION TO GLOBAL COORDINATE SYSTEM

The in-plane and bending stiffness matrices are derived in local coordinates with the x-y plane coinciding with the plane of the element. This coordinate system, in general, is different from the global coordinate system of structure.

The forces and displacements in the two systems are related in matrix notation as follows.

$$\{F_i'\} = [T]\{F_i\} \qquad (3.6.1)$$

$$\{\delta_i'\} = [T]\{\delta_i\} \qquad (3.6.2)$$

Hence the combined stiffness matrix of the element in the global system is

$$[K_e] = [T]^T [K_e'] [T] \tag{3.6.3}$$

Similarly all the stiffness matrices of the individual elements must each be transformed to the global system. When all the elements joining at a particular node are in the same plane, transformation to global system will make six or nine equations singular because only five or eight equations are linearly independent since the rotations θ_z is omitted. Such nodes must be assigned five or eight degrees of freedom, depending on the displacement function used, and forces and displacements should be considered in local coordinates.

Consider a triangular element 1-2-3 in which 2 is such a node. θ_z is not considered and the submatrix at node 2 is [5x5] or [8x8], depending on the displacement function used.

$$[K_{22}'] = \underset{(5\text{x}5)}{\left[\begin{array}{cc|ccc} \multicolumn{2}{c|}{\multirow{2}{*}{$[K_{22}']_p$}} & 0 & 0 & 0 \\ & & 0 & 0 & 0 \\ \hline 0 & 0 & & & \\ 0 & 0 & & [K_{22}']_b & \\ 0 & 0 & & & \end{array}\right]} \quad \text{or} \quad \underset{(8\text{x}8)}{\left[\begin{array}{cc|cccccc} \multicolumn{2}{c|}{\multirow{2}{*}{$[K_{22}']_p$}} & 0 & 0 & 0 & 0 & 0 & 0 \\ & & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & & & & & & \\ 0 & 0 & & & & & & \\ 0 & 0 & & & [K_{22}']_b & & & \\ 0 & 0 & & & & & & \\ 0 & 0 & & & & & & \\ 0 & 0 & & & & & & \end{array}\right]} \tag{3.6.4}$$

Again

$$[K_{12}'] = [K_{21}']^T$$

$$(6\text{x}5) \text{ or } (9\text{x}8) = (5\text{x}6)^T \text{ or } (8\text{x}9)^T$$

$$[K'_{12}] = \left[\begin{array}{cc|ccc} \multicolumn{2}{c|}{[K'_{12}]_p} & 0 & 0 & 0 \\ & & 0 & 0 & 0 \\ \hline 0 & 0 & & & \\ 0 & 0 & \multicolumn{3}{c}{[K'_{12}]_b} \\ 0 & 0 & & & \\ \hline 0 & 0 & 0 & 0 & 0 \end{array}\right] \quad \text{or} \quad \left[\begin{array}{cc|cccccc} \multicolumn{2}{c|}{[K'_{12}]_p} & 0 & 0 & 0 & 0 & 0 & 0 \\ & & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & & & & & & \\ 0 & 0 & & & & & & \\ 0 & 0 & & & \multicolumn{2}{c}{[K'_{12}]_b} & & \\ 0 & 0 & & & & & & \\ 0 & 0 & & & & & & \\ 0 & 0 & & & & & & \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]$$

$$(6x5) \qquad\qquad (9x8)$$

(3.6.5)

$[K'_{22}]$ is retained in local coordinates while $[K'_{12}]$ and $[K'_{21}]$ are transformed as follows

$$[K_{12}] = [T]^T[K'_{12}]$$
$$= (6x6)^T(6x5) \text{ or } (9x9)^T(9x8)$$

$$[K_{21}] = [K_{12}]^T = [[T]^T[K_{12}]]^T = [K_{12}]^T[T]$$
$$= [K'_{21}][T] \qquad (3.6.7)$$
$$= (5x6)\ (6x6) \text{ or } (8x9)(9x9)$$

3.7 ASSEMBLY OF OVERALL STIFFNESS MATRIX

Once the transformation of the combined stiffness matrix of individual element to the global system has been made, the overall stiffness matrix of the entire assembly can be found using equilibrium conditions at the node of the structure. If $\{R_i\}$ are the external forces acting at node i and maintaining equilibrium conditions, then each component of $\{R_i\}$ is equal to the sum of the component forces $\{F_i\}$ contributed by the elements meeting at that node. Hence

$$\{R_i\} = \Sigma\{F_i\} \qquad (3.7.1)$$

= summation being taken over all the elements.

$$\{R_i\} = \sum_{m=1}^{n} \sum_{M=1}^{N} [K_{im}]^M \{\delta_M\} \qquad (3.7.2)$$

ignoring distributed loads and initial strains.

N = number of elements

n = number of nodes

If a particular element does not infact include the node in question, it will not contain submatrices with an i suffix and therefore, its contribution will simply be zero.

Hence the submatrices of an assembled stiffness matrices are

$$[K_{im}] = \sum_{m=1}^{n} \sum_{M=1}^{N} [K_{im}]^M \qquad (3.7.3)$$

3.8 <u>SOLUTION OF EQUATIONS</u>

If the overall stiffness matrix of the entire structure can be stored in the computer memory, nodal deflections can easily be evaluated by inverting the overall stiffness matrix and then multiplying by the load-vector. Since stresses are a function of nodal displacements, the stresses at any point within the element can be calculated accordingly. In analysing shell structure, convergence to the true results can be achieved only when the mesh size is refined, resulting in an increase in the number of nodal points. If six or nine degrees of freedom are assigned to each node, the total number of degrees of freedom will be six or nine times the total number of nodes. Obviously as the number of nodes increases, the size of the overall stiffness matrix increases rapidly. Hence a direct stiffness assembly and inversion of the stiffness matrix is limited by the available computer central memory. By this method, variation of results are quite high and obviously more refined mesh are required [6] for this particular problem. Refinement can not be accomplished with the present CDC 6400 computer central memory.

3.9 FORMATION OF THE OVERALL STIFFNESS MATRIX IN TRIDIAGONALIZED PARTITIONS

The method of tridagonalization which makes use of symmetry and sparseness of the stiffness matrix, is most applicable to solve the large number of equations that frequently occur in analyzing multistorey buildings [19]. The structure is divided into a number of partitions and by properly numbering the nodal points and elements, the stiffness matrix of these elements in that particular partition can be arranged in the form of a tridiagonal set of submatrices which can be evaluated and stored on magnetic tape. The solution is obtained by a method of recursion. A computer program to analyse in-plane problem by this method is given in reference [1].

After dividing the structure into a number of partitions, the nodal points as well as the elements are numbered in consecutive order. The partitioning technique is explained in detail in reference [1]. The assembled stiffness matrix of the structure is arranged in tridiagonalized form as follows.

$$\begin{bmatrix} K_1 & C_1 & - & - & - & - & - & - & - & 0 & 0 \\ C_1^T & K_{11} & C_{11} & - & - & - & - & - & - & 0 & 0 \\ - & C_{11}^T & K_{111} & C_{111} & - & - & - & - & - & - & - \\ - & - & C_{111}^T & K_{1V} & C_{1V} & - & - & - & - & - & - \\ - & - & - & C_{1V}^T & K_V & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - & K_{N-2} & C_{N-2} & \\ 0 & 0 & - & - & - & - & - & - & C_{N-2}^T & K_{N-1} & C_{N-1} \\ 0 & 0 & - & - & - & - & - & - & - & C_{N-1}^T & K_N \end{bmatrix} \begin{Bmatrix} \delta_1 \\ \delta_{11} \\ \delta_{111} \\ - \\ - \\ - \\ - \\ - \\ \delta_{N-2} \\ \delta_{N-1} \\ \delta_N \end{Bmatrix} = \begin{Bmatrix} P_1 \\ P_{11} \\ P_{111} \\ - \\ - \\ - \\ - \\ - \\ P_{N-2} \\ P_{N-1} \\ P_N \end{Bmatrix} \tag{3.9.1}$$

The first two matrix equations can be written as

$$[K_1]\{\delta_1\} + [C_1]\{\delta_{11}\} = \{P_1\} \tag{3.9.2}$$

$$[C_1]^T\{\delta_1\} + [K_{11}]\{\delta_{11}\} + [C_{11}]\{\delta_{111}\} = \{P_{11}\} \tag{3.9.3}$$

From equation (3.9.2)

$$\{\delta_1\} = [K_1]^{-1}\{P_1\} - [K_1]^{-1}[C_1]\{\delta_{11}\}$$

Substituting for $\{\delta_1\}$ in equation (3.9.3), yields

$$([K_{11}]-[C_1]^T[K_1]^{-1}[C_1])\{\delta_{11}\}+[C_{11}]\{\delta_{111}\}=\{P_{11}\}-[C_1]^T[K_1]^{-1}\{P_1\} \tag{3.9.4}$$

Finally defining new symbols

$$[\bar{K}_{11}] = ([K_{11}] - [C_1]^T[K_1]^{-1}[C_1]) \tag{3.9.5}$$

$$\{\bar{P}_{11}\} = \{P_{11}\} - [C_1]^T[K_1]^{-1}\{P_1\} \tag{3.9.6}$$

Hence equation (3.9.4) becomes

$$[\bar{K}_{11}]\{\delta_{11}\} + [C_{11}]\{\delta_{111}\} = \{\bar{P}_{11}\} \tag{3.9.7}$$

From which $\{\delta_{11}\}$ can be obtained and substituted into the next row of equations to give a modified $[\bar{K}_{111}]$ and $\{\bar{P}_{111}\}$. Such processes of substitution and elimination can go on until the second last row equation is reached.

$$[\bar{K}_{N-1}]\{\delta_{N-1}\} + [C_{N-1}]\{\delta_N\} = \{\bar{P}_{N-1}\} \tag{3.9.8}$$

Final equation is

$$[C_{N-1}]^T\{\delta_{N-1}\} + [K_N]\{\delta_N\} = \{P_N\} \tag{3.9.9}$$

From equation (3.9.8)

$$\{\delta_{N-1}\} = [\bar{K}_{N-1}]^{-1}\{\bar{P}_{N-1}\} - [\bar{K}_{N-1}]^{-1}[C_{N-1}]\{\delta_N\} \tag{3.9.10}$$

Substituting for $\{\delta_{N-1}\}$ in equation (3.9.9)

$$([K_N]-[C_{N-1}]^T[K_{N-1}]^{-1}[C_{N-1}])\{\delta_N\}=\{P_N\}-[C_{N-1}]^T[\bar{K}_{N-1}]^{-1}\{\bar{P}_{N-1}\}$$

or

$$[\bar{K}_N]\{\delta_N\} = \{\bar{P}_N\} \tag{3.9.11}$$

when a direct inversion will yield $\{\delta_N\}$, the other deflections can be obtained by back substitution in the equations concerned.

The significant points in the computer program are as follows.

(1) The zero elements outside the tridiagonal band are not stored.

(2) Due to symmetry, it is necessary to store only $[C_i]$ and $[K_i]$, but not $[C_i]^T$.

(3) Very little additional computer time is required for solution of more than one load conditions.

(4) The $[\bar{K}_i]^{-1}$ and $\{\bar{P}_i\}$ are stored on magnetic tapes as soon as they are generated in the forward elimination process, and they are used subsequently in the backward substitution.

(5) Only two submatrices $[K_i]$ and $[C_i]$ are required at a time in the memory of the computer.

A computer program, using the above method to analyse shells and folded plate structures was obtained [6] and modified and new subprograms were written wherever necessary for different displacement functions (Appendices V and VI).

The partitioning technique is shown in Figure 20 and computation are carried out from five to ten partitions. It is possible to divide the structure into more than ten partitions, but the height of each partition becomes decreased, making the triangular element long and narrow [1].

There is no limit to the number of partitions into which the structure can be divided as the matrices $[K_i]$ and $[C_i]$ are eliminated in blocks. A considerable storage is required for $[K_i]$ and $[C_i]$ for intermediate computations. Hence the tridiagonal band or the number of nodes in each partition is limited by the available storage capacity of the computer. It is also noticed that a matrix with a narrower band requires less solution time, hence the band width depends on the way the nodal points are numbered and is numerically equal to the product of the number of degrees of freedom per node and the maximum difference in numbering of adjoining nodes.

4.0 EXPERIMENTAL ANALYSIS

4.1 INTRODUCTION

The object of the experimental analysis was to confirm the validity of the theoretical results obtained. Some nodal points and triangular elements were chosen arbitrarily to verify deflections and stresses developed under a particular external load. The structure was loaded by means of a combination of load cell and turnbuckle connected by wire cord to one of the nodes at the top of the structure.

4.2 LOADING DEVICE

The load cell was calibrated against a Tinius Olsen Machine up to 1500 pounds. The load could be applied directly to the structure by turning the turnbuckle which was connected to the load cell axially through a wire cord and the amount of load could be read directly from the strain indicator unit.

4.3 DISPLACEMENT TRANSDUCER

A capacitance type proximity transducer coupled through an oscillator and reactance converter to a cathode ray oscilloscope was adopted as the linear measuring device. The transducer consisted of a fixed electrode. Any flat conducting surface parallel to the fixed electrode can act as the moving electrode. Normally the moving electrode is fastened to the component whose displacement is to be measured. In the present application the structure had both linear and angular displacements whereas the proximity transducer is designed to work when electrodes remain parallel while moving towards or away from each other.

To eliminate angular displacement, the moving electrode was mounted on the transducer itself. The spindle of the moving electrode was supported jointly by two 2 1/2" x 1/2" x 5/1000" thick stainless steel strips, parallel to each other, which forced the electrode surfaces to remain parallel during relative motion. This is illustrated in Figure 24.

The transducer system is based on frequency modulation of a carrier wave. The capacitance of the electrode is in parallel with another fixed capacitance. The combination forms a series resonant circuit with an inductance. The change in distance between the electrodes, due to the loading of the structure caused a change in reactance in the resonant circuit which is used to change the frequency of the signal delivered by the oscillator. The signal is amplified and detected to provide a proportional D.C. voltage which is metered on the oscilloscope. Unloading the structure restores initial gap between the electrodes. The transducer can then be calibrated by the integral micrometer producing a deflection on the oscilloscope of the same order as that obtained due to the load. The calibration enables the displacement to be evaluated. One subdivision on the micrometer thimble is 0.01 mm which could be further divided by the oscilloscope. An initial gap of 0.5 to 1.5 cm between the electrodes is used.

4.4 STRESS CALCULATION

The stresses obtained theoretically need to be compared with those from the experiment. Electrical resistance strain gauges were used to measure strains at a point on the surface of the shell. Delta rosette type strain gauges were fixed at arbitrary points on the outer surface of the structure and strains in the directions of gauges were measured to calculate five components of stresses (normal stress in the x-direction, normal stress in the y-direction, shear stress in the x-y plane, maximum and minimum principal stresses) at a point. Four sets of delta rosette strain gauges were fixed on four arbitrarily chosen elements near the base of the structure where maximum effect of external load occurred. These strain gauges were connected to the strain indicator through switch and balance unit, so that readings can be taken one at a time. A dummy strain gauge fixed on an unloaded plate, the material of which is the same as that of the structure, was used for temperature compensation. Having obtained strain gauge readings, stresses can be calculated. The techniques of using strain gauges and theory can be found in references [20] and [21] in detail.

5.0 RESULTS AND CONCLUSIONS

Tables 1a to 3d show the computed values of deflections u, v and w obtained from the three displacement functions used in the anlysis for a different number of partitions, ranging from five to ten. As the partitions increase in numbers, it can be assumed that each deflection in the x,y,z directions converges to a constant value for each direction. This assumption is valid since there is negligible change in values of deflections between partitions nine and ten. Thus ten partitions are assumed enough for accuracy and the value of deflections for ten partitions are accepted as true values for experimental verification.

The graphs, in Figures 1 to 8, are curves of deflections u, v and w obtained from different functions versus the number of partitions, illustrating the convergence of the results. It is also seen from Tables 4a to 4f that the computed results obtained by three displacement functions agree with one another within 5 percent. Much higher fineness is achieved and improvement in results is observed by the method of tridiagonalization. Although it is possible to divide the structure into more than ten partitions, the results may not be expected to be better, since, as the partition height decreases, the triangular elements become narrow and long.

The graphs in Figures 9 to 12 show the deflections v and w against the nodal numbers along the centre-line of the plates CC and DD respectively. The nature of deflections is again found to be consistent with three displacement functions. Tables 7a and 7c show the rotations θ_x, θ_y and θ_z at nodal points A, B, C and D by three different displacement functions. As these functions are non-conforming, partial conforming and full conforming, the above rotations are not consistent with one another.

In Tables 5a and 5b, a comparative study is presented of analytically calculated and experimentally measured deflections, for different directions of load system (Figures 21 to 23). It can be seen that the theoretical deflection u, v and w are in agreement within 20.0 to 26.8 percent of the measured deflections for a particular loading.

A comparison of theoretical and experimental values of stresses is given in Table 6. Although the available laboratory set up made it possible to load the structure to develop sufficiently high stresses, the measured stresses are found to differ from the the theoretical stresses.

It can be seen that the relatively higher calculated and experimental stress values are in fairly good agreement with each other, whereas the low stress values have very poor agreement. This maybe attributed to the following factors

(a) Measurements of very small stress, using strain gauges are not reliable.

(b) a slight alteration made at the base for rigidity after fixing the strain gauges to the structure

(c) displacement method gives good agreement in deflections but not in stress distribution

(d) elastic properties of the material used were not actually determined before and after the structure was built, but simply accepted at the value which was suggested by the manufacturer

It is interesting to note that the line and the central memory storage taken by the CDC 6400 computer for each of three different cases for the same idealization is very much different (Table 8). The solution to the Tocher displacement function takes 1.67 minutes whereas the solution to the Rawtani and Cowper displacement functions takes 2.962 and 14.8 minutes respectively for ten partitions of the structure for the same idealization.

It can be seen that the time required is minimum for the Tocher displacement function because of the simplicity of the computation involved. The difference in the deflections u, v and w obtained from the Tocher and Rawtani displacement functions are negligibly small (0.00679 to 0.8045%) while the deflections u, v and w obtained by the Cowper displacement function differ by 2.073 to 4.35% (Tables 4a to 4f).

It has been shown by Cowper et al [7] that their conforming shape function gives the best convergence to the results. It is important that the computer time, the central memory storage and the accuracy required should be considered in choosing the displacement function for the solution of a particular problem.

At present the finite element displacement method using general triangular flat elements for analysis of shells and folded plate structures is powerful and the most suitable for any arbitrary shape. Out of the three displacement functions, it is found that the Tocher displacement functions requires the least computer time and the least central memory for this problem which is very important for practical applications that require a large number of solutions.

FIGURES

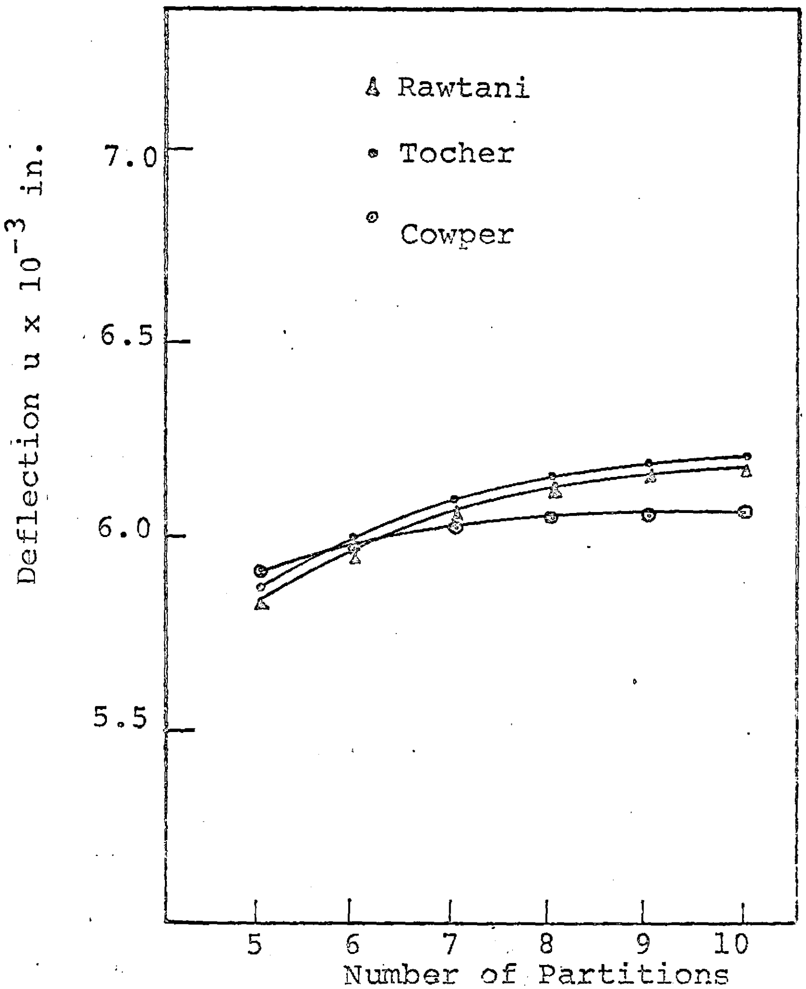


Fig. 1
Deflection u at Node A for Different Partitions under Load P=450 pounds at Node B (Fig.15) in X-Direction.

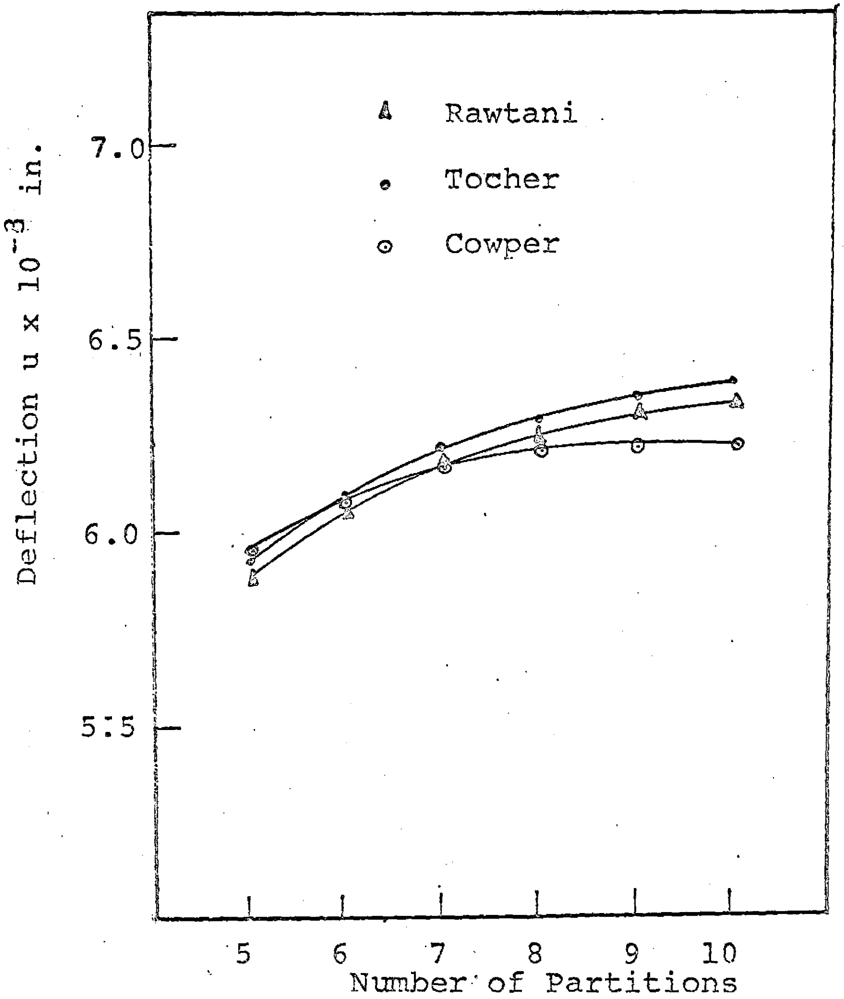


Fig. 2
Deflection u at Node B for Different Partitions under Load P=450 pounds at Node B (Fig.15) in X-Direction.

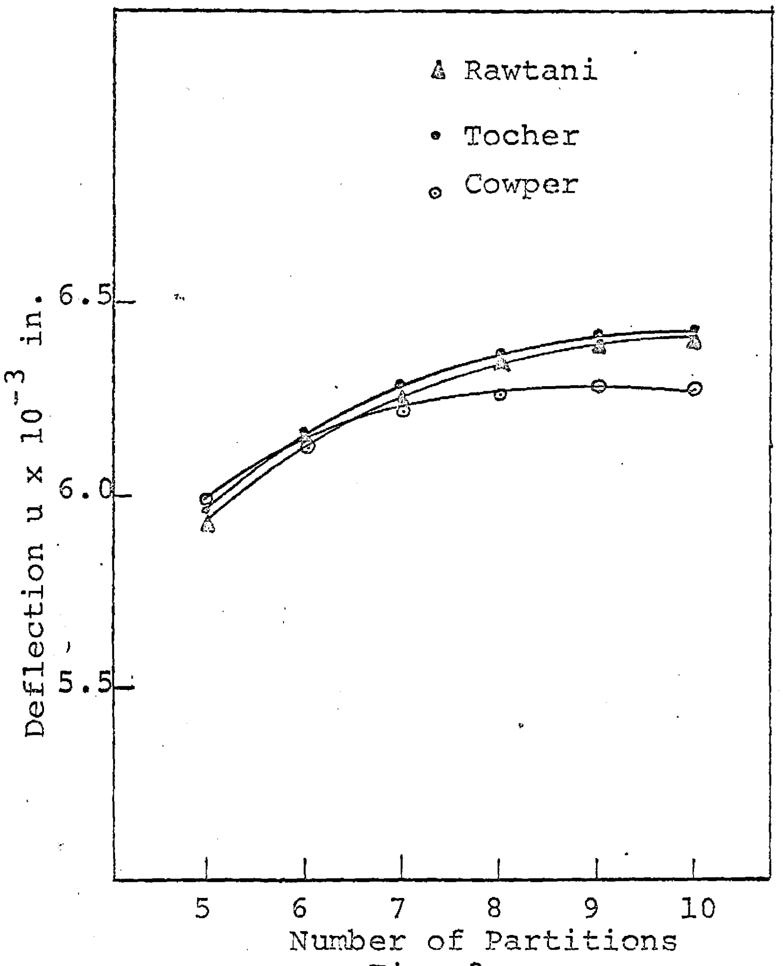


Fig. 3

Deflection u at Node C for Different Partitions under Load P=450 pounds at Node B (Fig. 15) in X-Direction.

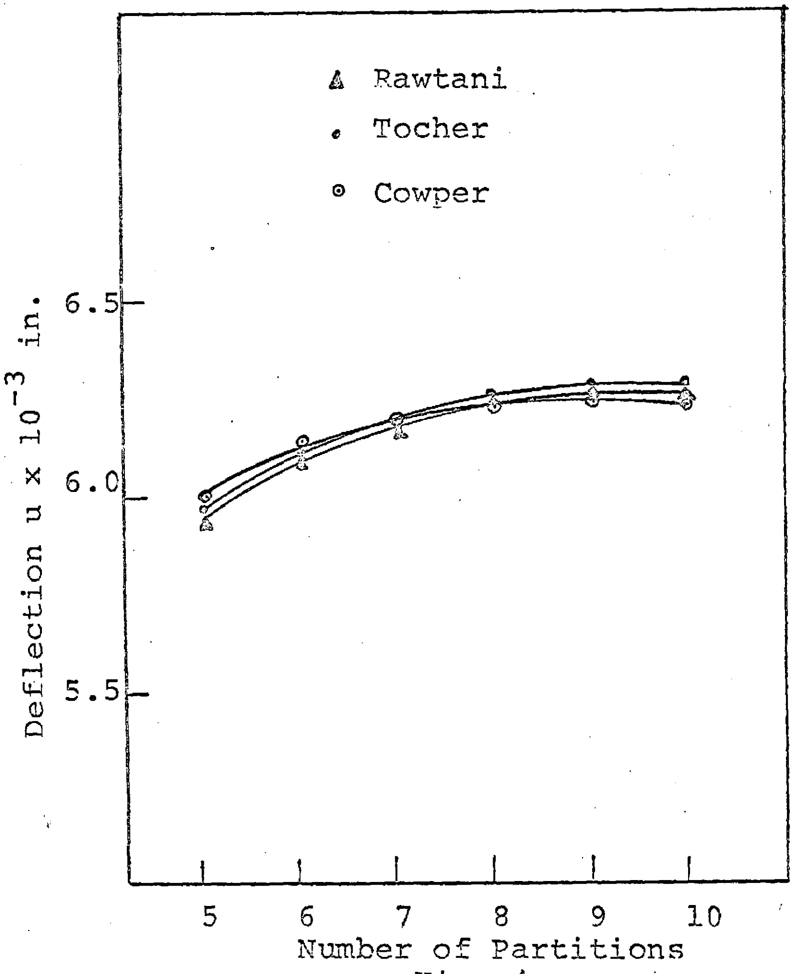


Fig. 4

Deflection u at Node D for Different Partitions under Load P=450 pounds at Node B (Fig. 15) in X-Direction.

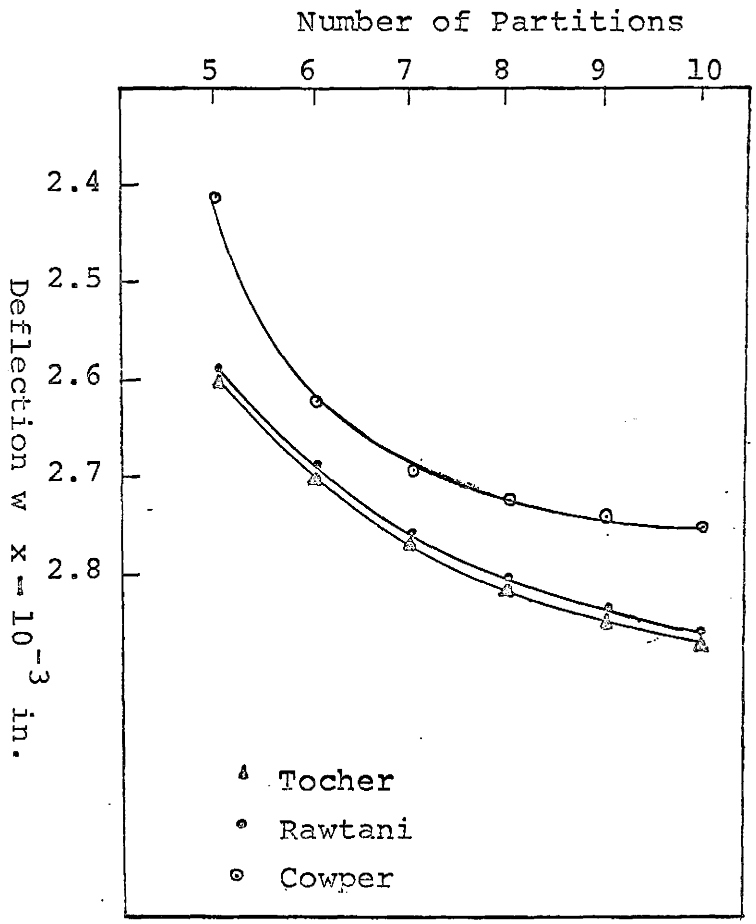


Fig. 5
Deflection w at Node C for Different Partitons under Load P=450 pounds at Node B (Fig. 15) in X-Direction.

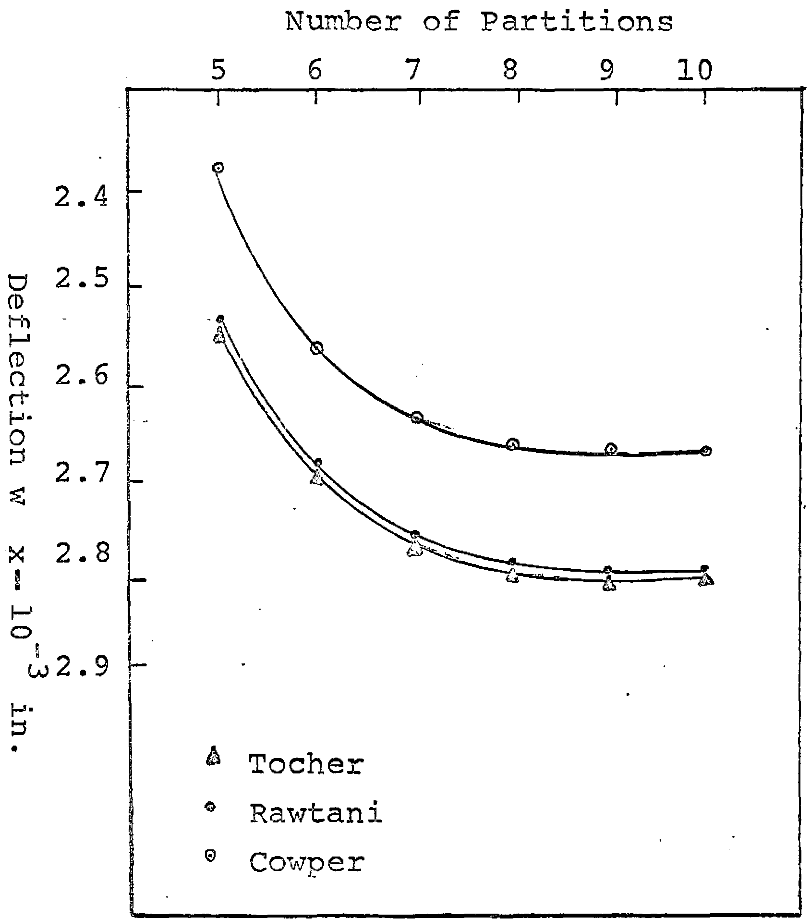


Fig. 6
Deflection w at Node B for Different Partitions under Load P=450 pounds at Node B (Fig. 15) in X-Direction.

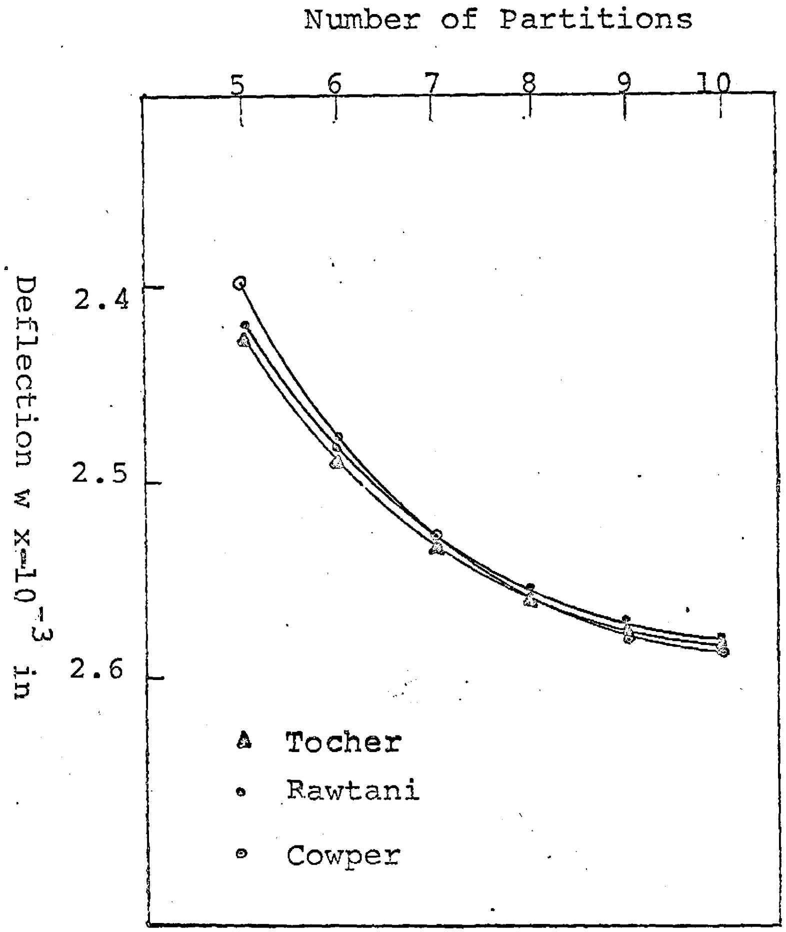


Fig. 7
Deflection w at Node D for Different Partitions under Load P=450 pounds at Node B (Fig. 15) in X-Direction

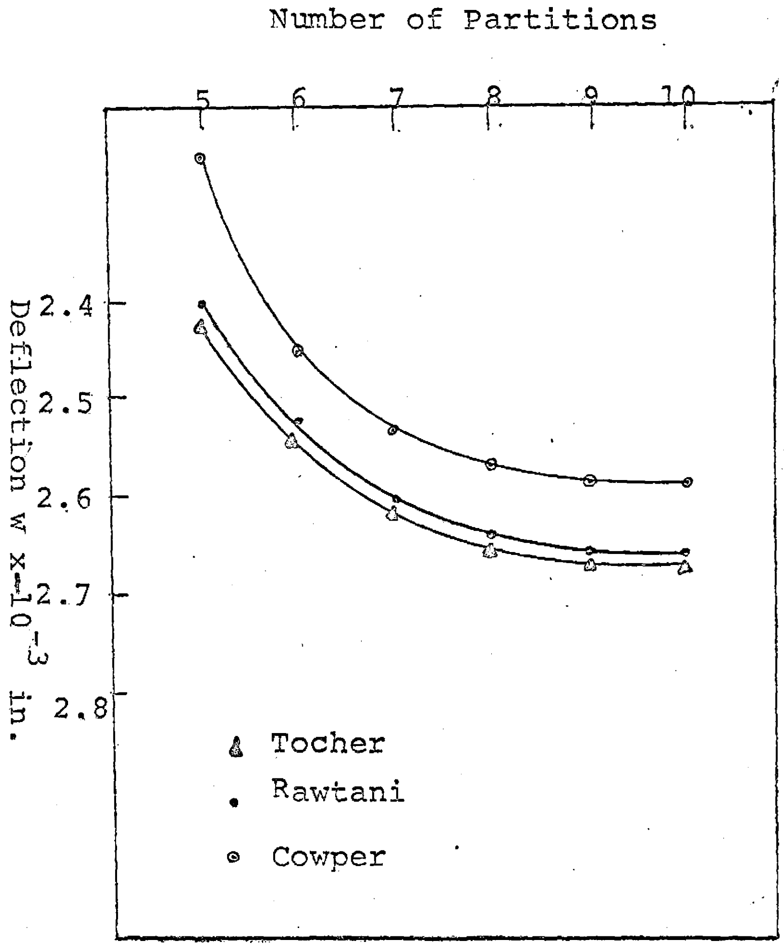


Fig. 8
Deflection w at Node A for Different Partitions under Load P=450 pounds at Node B (Fig. 15) in X-Direction.

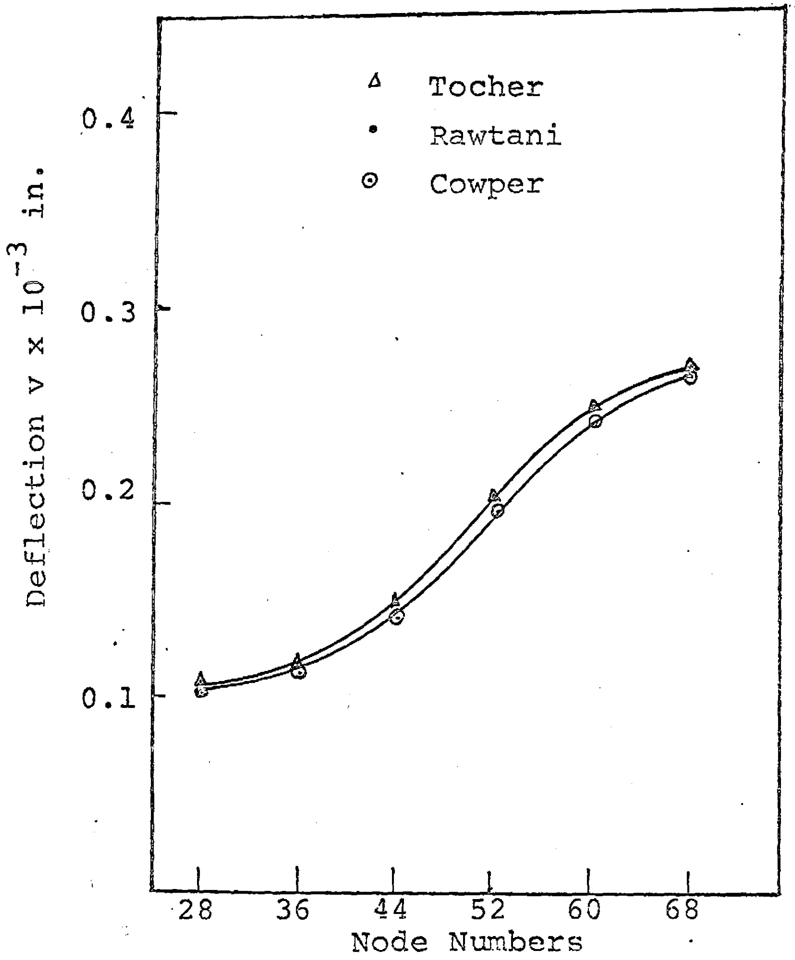


Fig. 9
Deflection v Along the Centre line of Plate CC under Load P=450 pounds at Node B (Fig. 15) in X-Direction for 10 Partitions

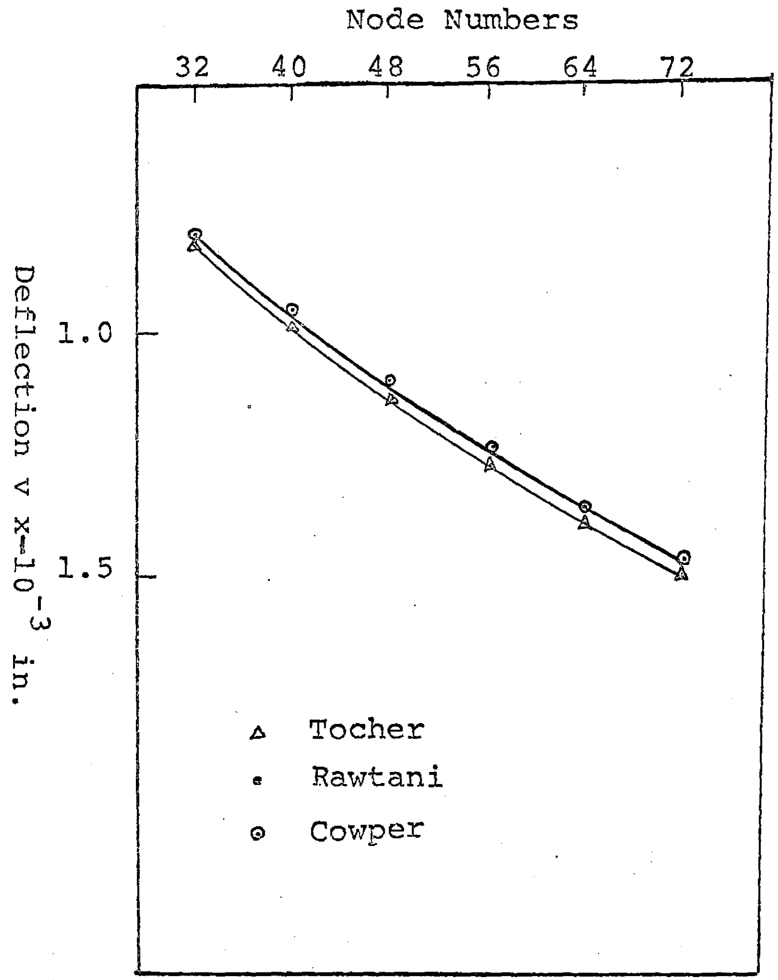


Fig. 10
Deflection v Along the Centre line of Plate DD under Load P=450 pounds at Node B (Fig. 15) in X-Direction for 10 Partitions

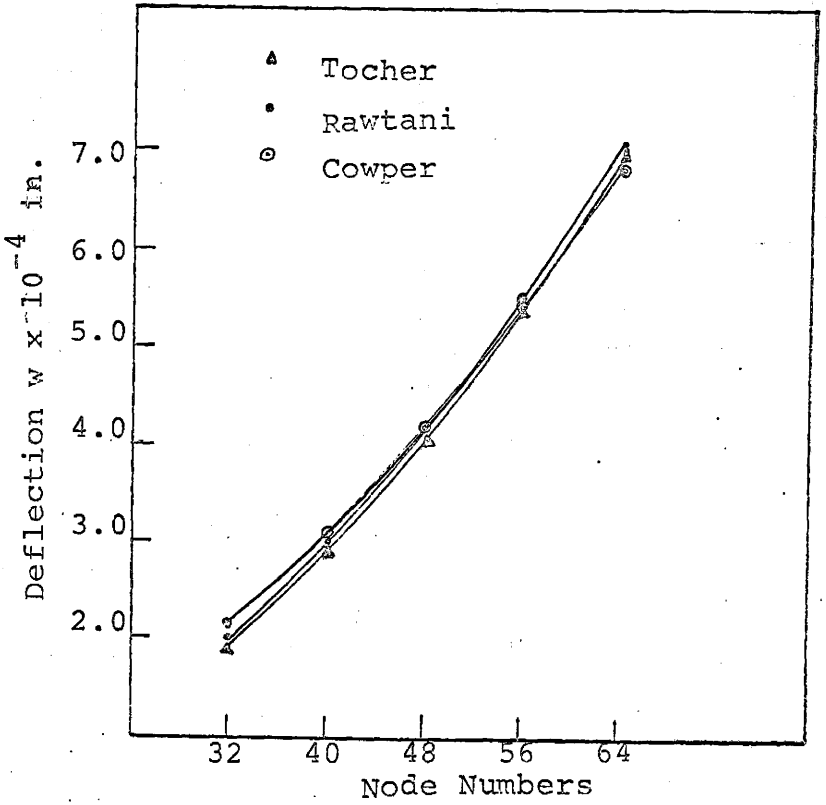


Fig. 11

Deflection w Along the Centre line of Plate DD under Load P=450 pounds at Node B (Fig. 15) in X-Direction for 10 Partitions.

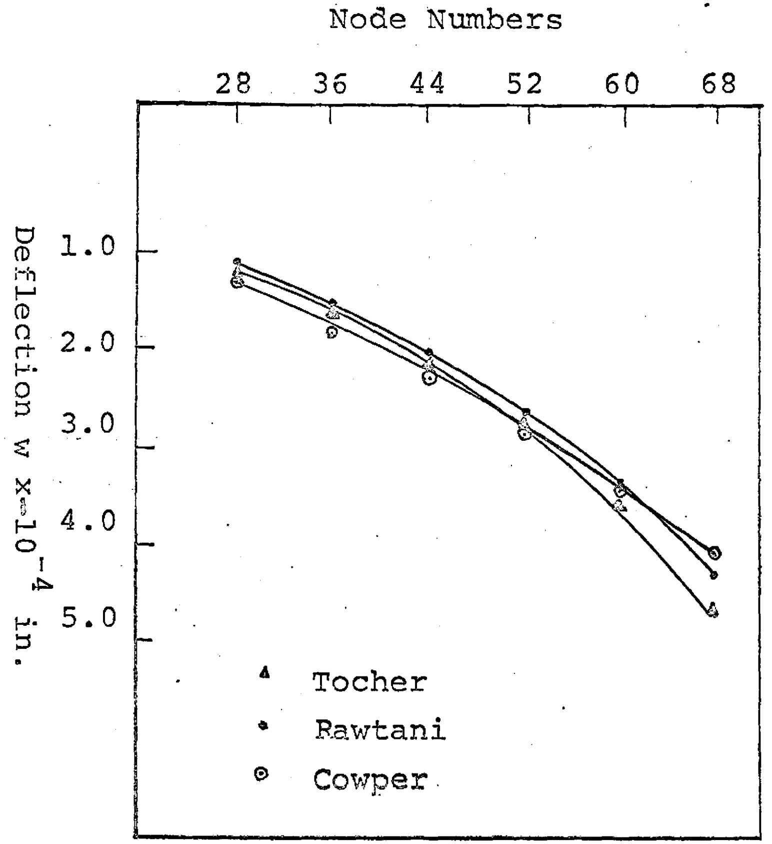
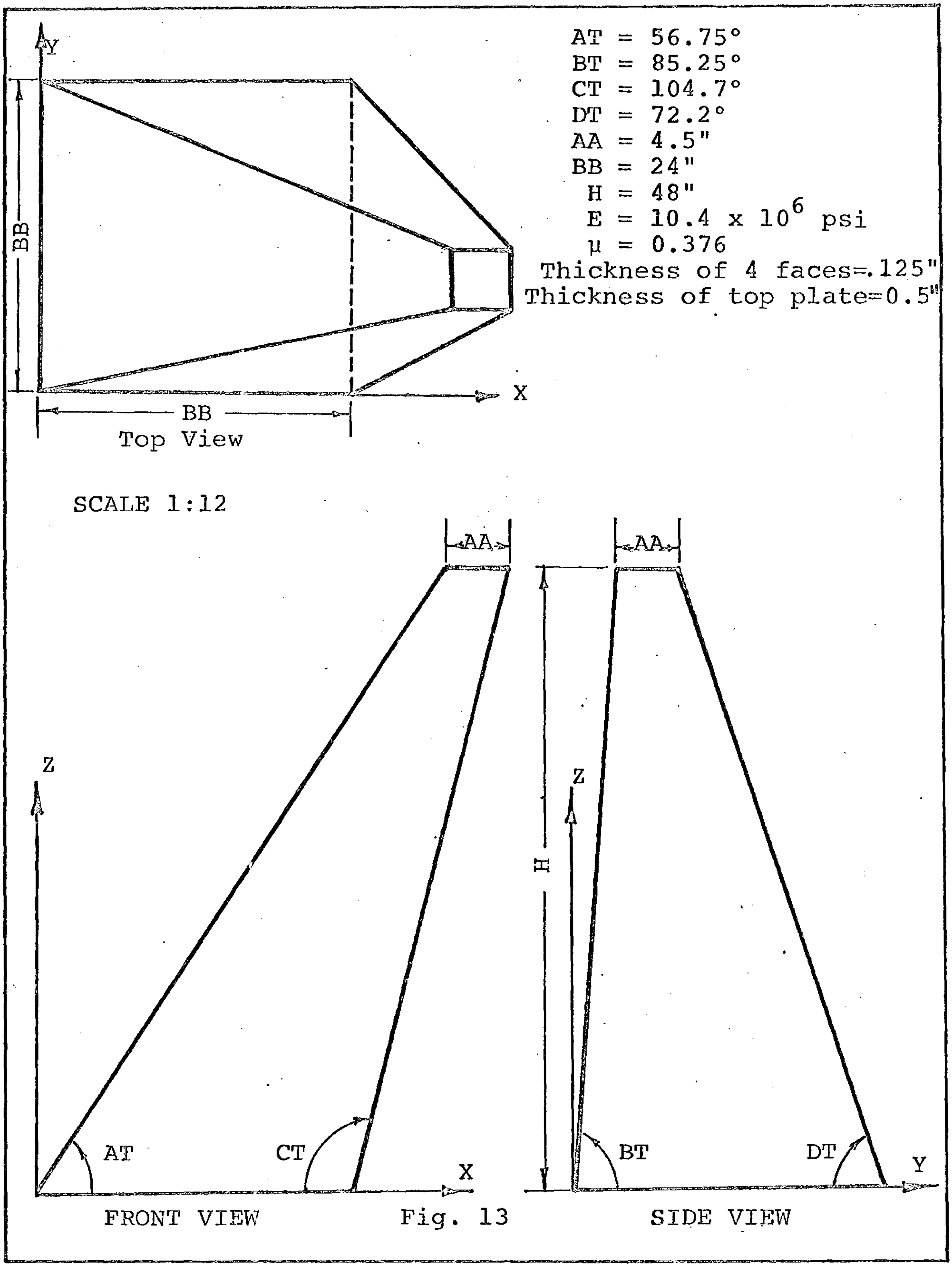


Fig. 12

Deflection w along the Centre line of Plate CC under Load P=450 pounds at Node B (Fig. 15) in X-Direction for 10 Partitions.



AT = 56.75°
BT = 85.25°
CT = 104.7°
DT = 72.2°
AA = 4.5"
BB = 24"
H = 48"
E = 10.4×10^6 psi
μ = 0.376
Thickness of 4 faces=.125"
Thickness of top plate=0.5"

Fig. 13

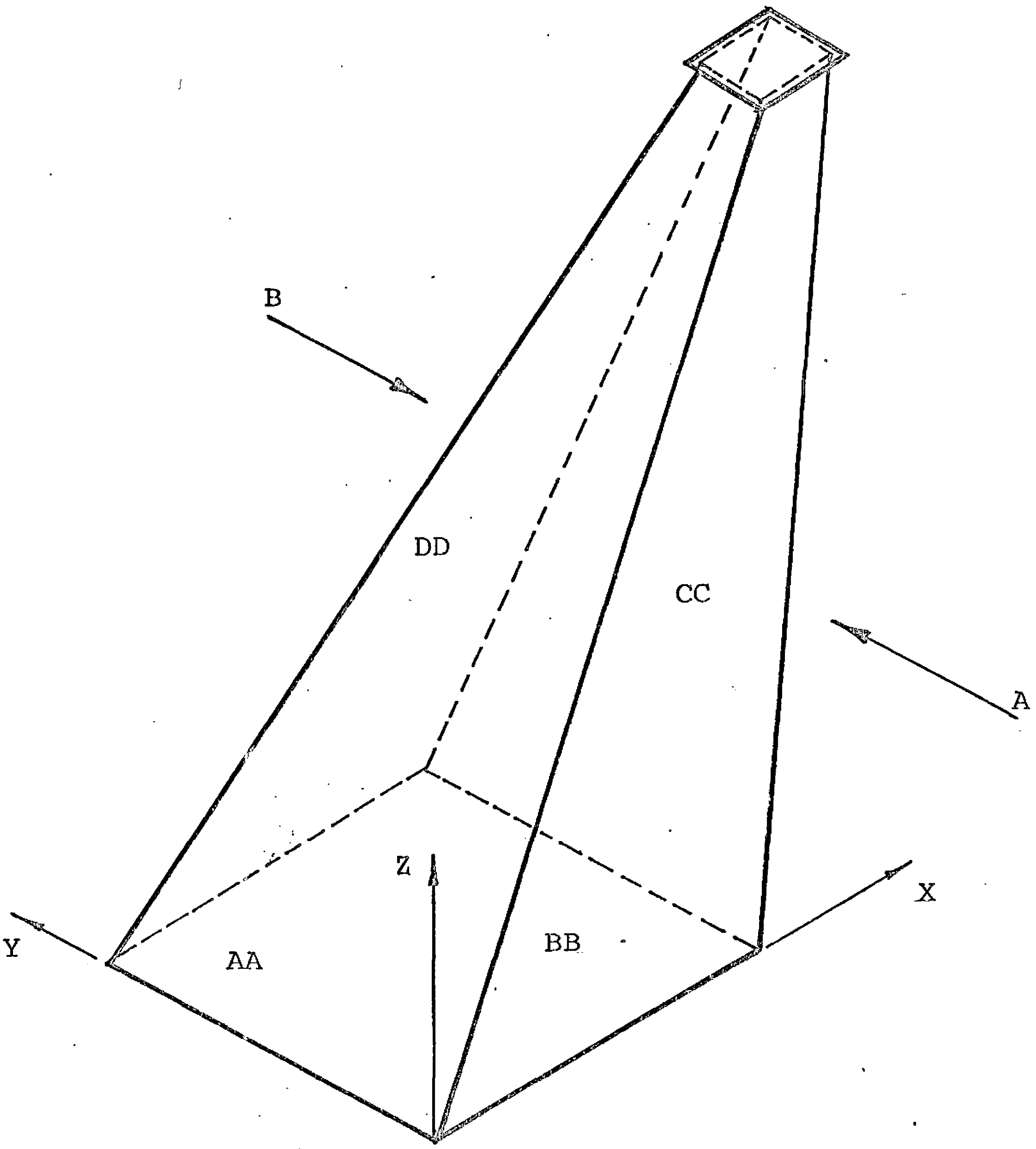


Fig. 14
(Not Drawn to Scale)

AA, BB, CC, DD - 4 Faces of the Structure

ISOMETRIC VIEW OF THE FOLDED PLATE STRUCTURE

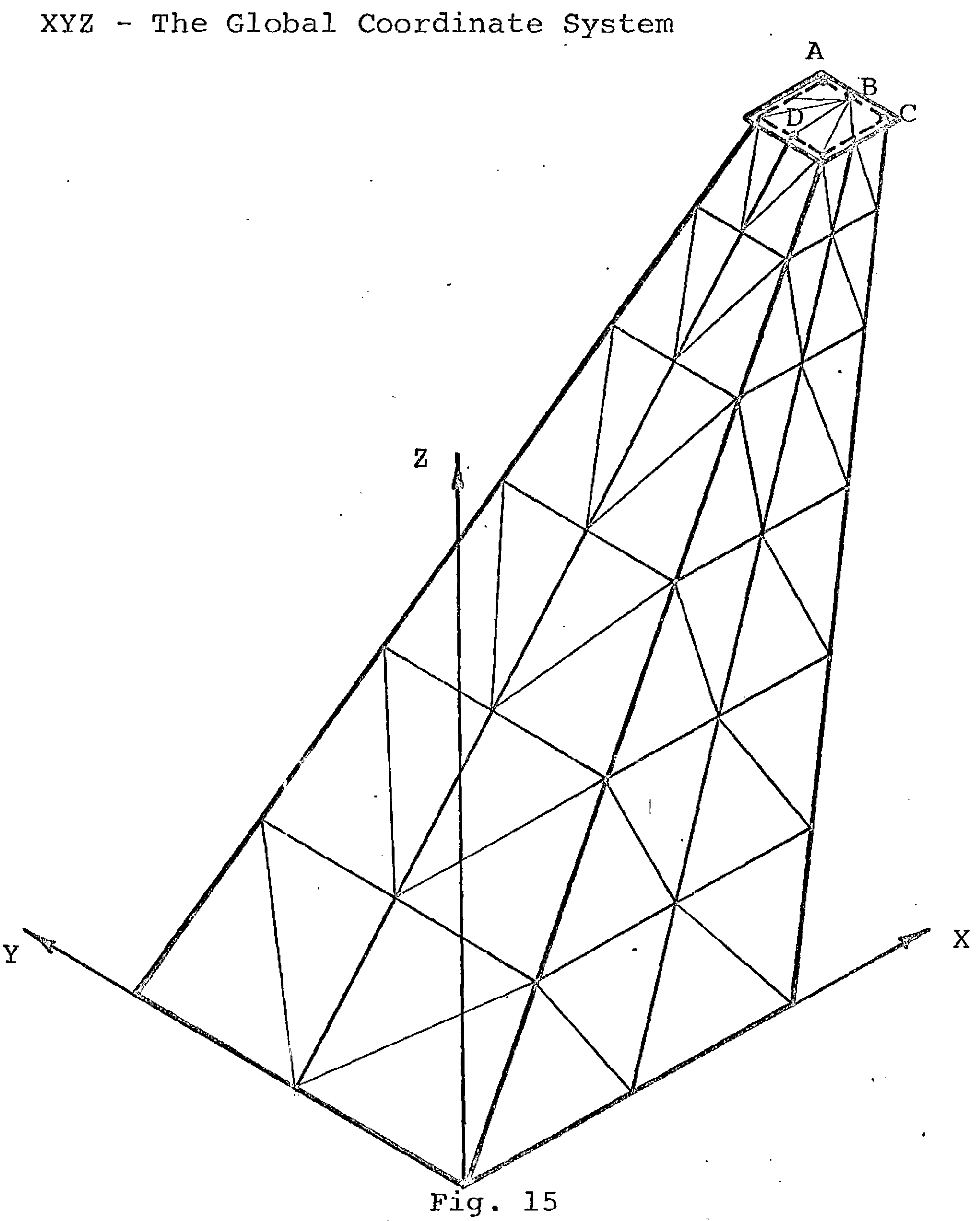


Fig. 15
(Not Drawn to Scale)
DIVISION OF TRIANGULAR ELEMENTS AS SEEN FROM THE DIRECTION A(Fig. 14)

XYZ - The Global Coordinate
System

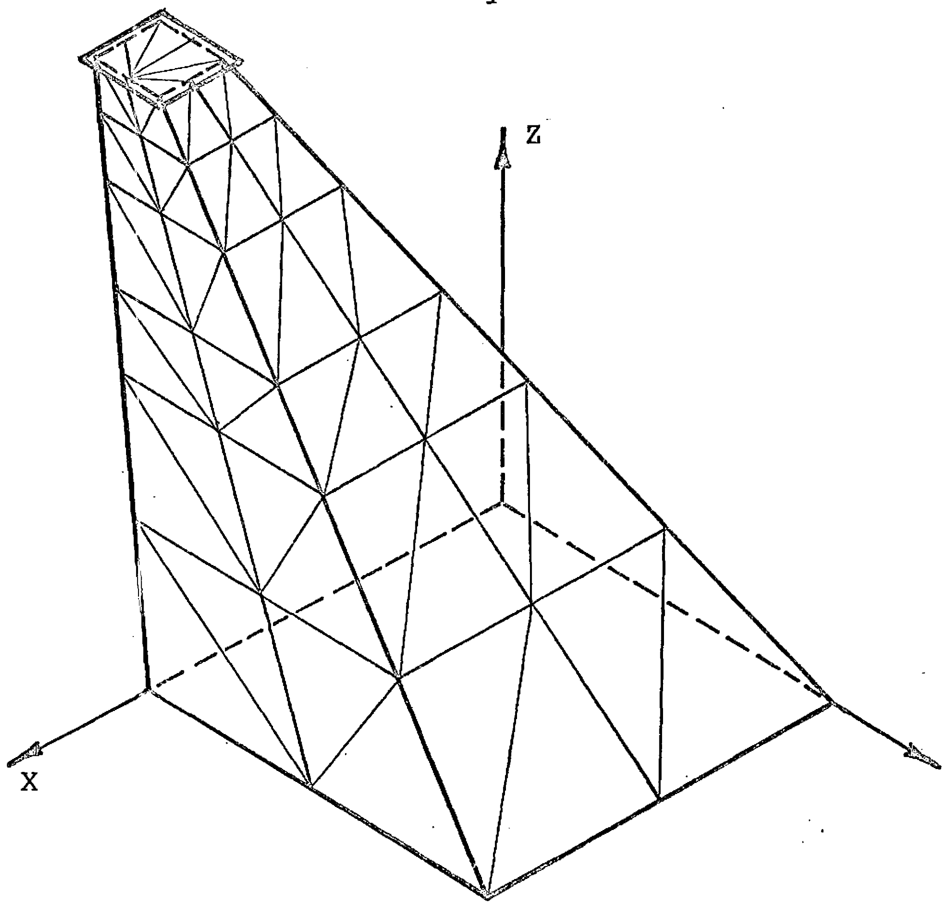


Fig. 16
(Not Drawn to Scale)

DIVISION OF TRIANGULAR ELEMENTS AS SEEN FROM
THE DIRECTION B (Fig. 14)

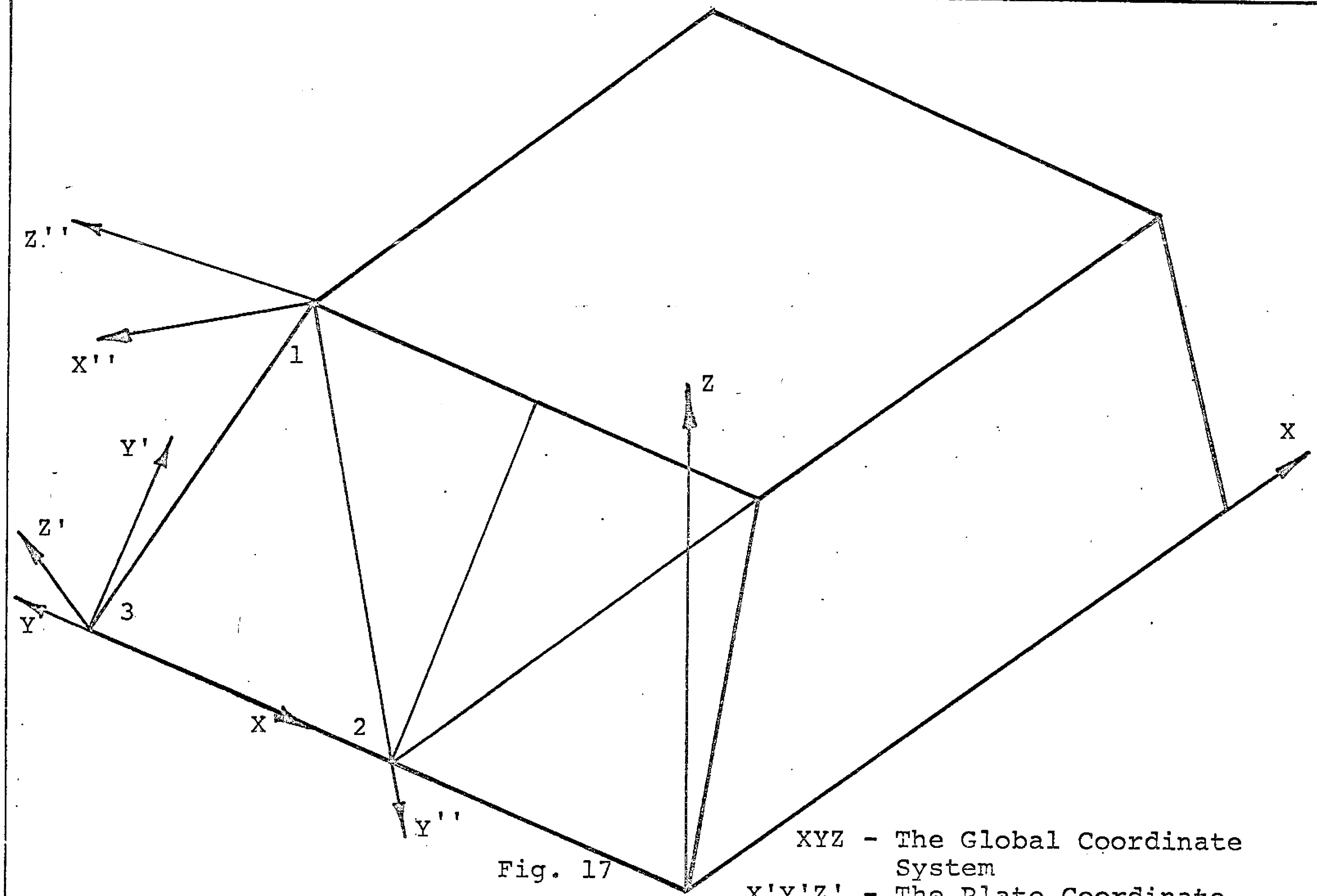


Fig. 17

THE COORDINATE SYSTEMS USED FOR THE RAWTANI DISPLACEMENT FUNCTION

XYZ - The Global Coordinate System

X'Y'Z' - The Plate Coordinate System

X''Y''Z'' - The Local Coordinate System

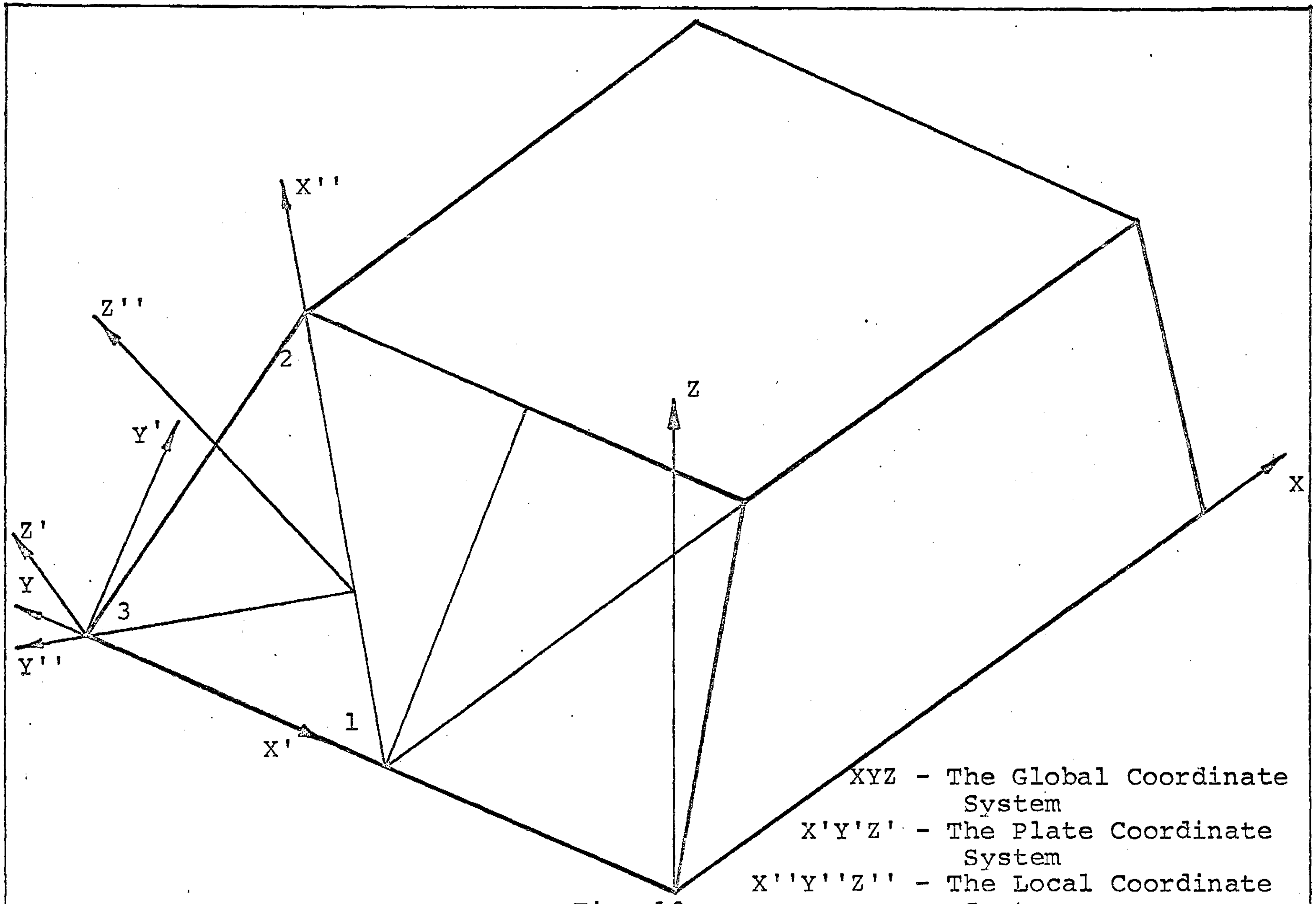


Fig. 18

THE COORDINATE SYSTEMS USED FOR THE COWPER DISPLACEMENT FUNCTION

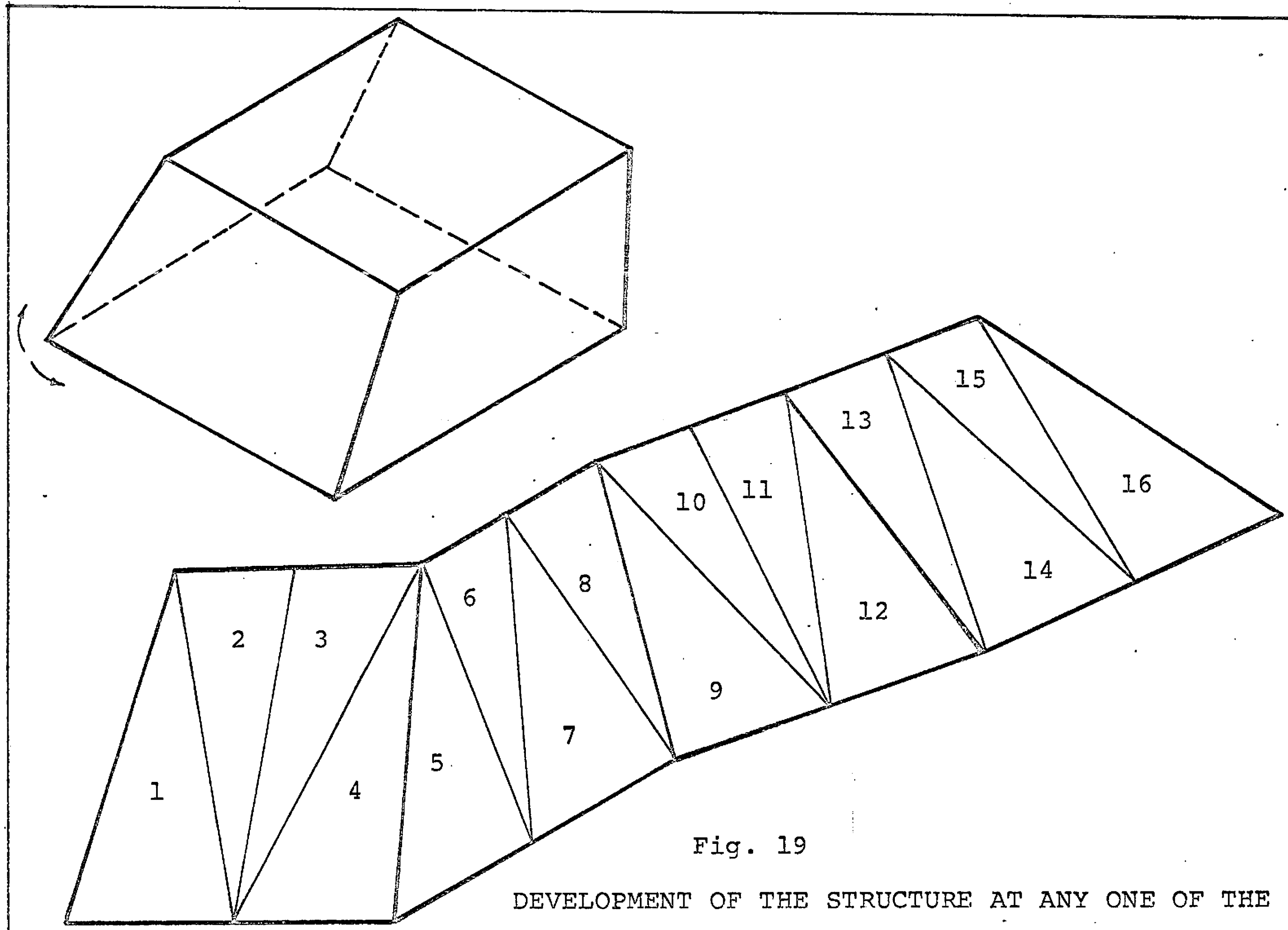


Fig. 19

DEVELOPMENT OF THE STRUCTURE AT ANY ONE OF THE PARTITIONS, SHOWING THE ARRANGEMENT OF THE ELEMENTS IN EACH PARTITION.

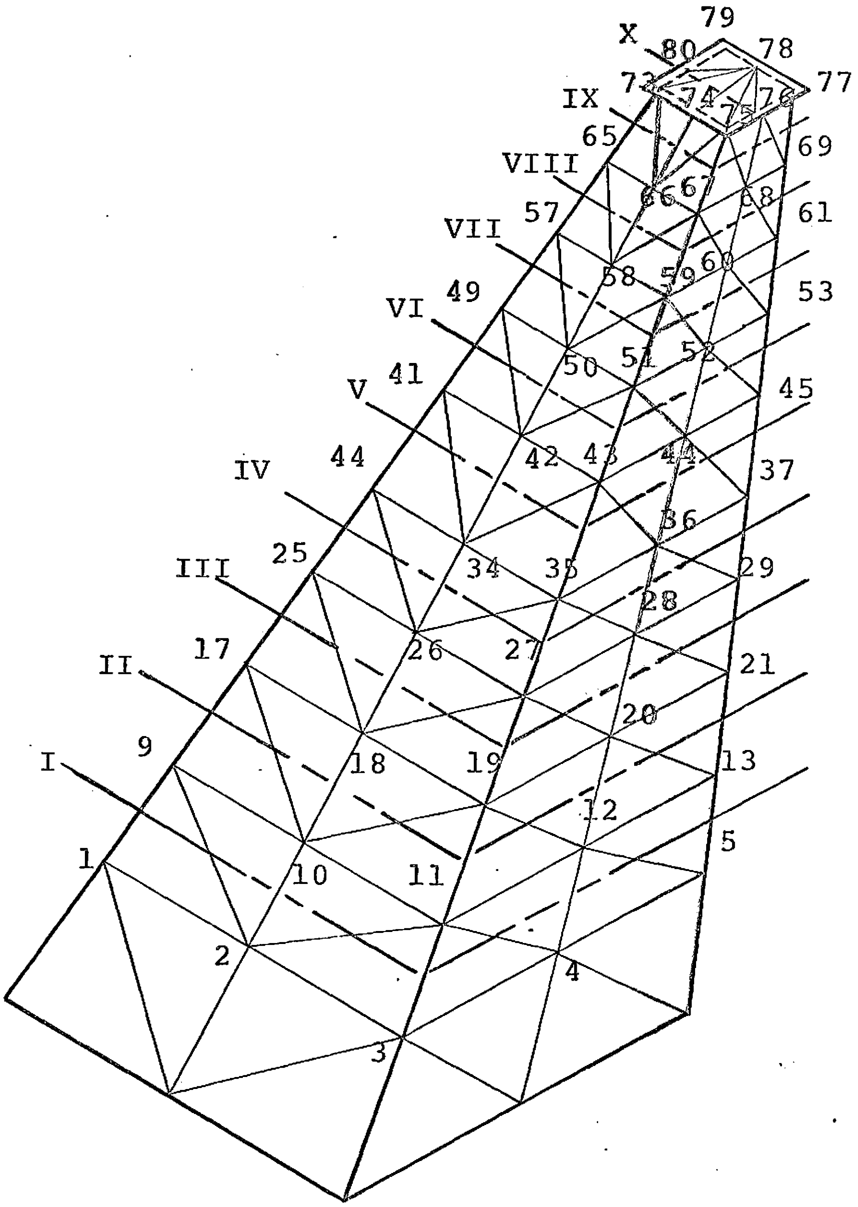


Fig. 20
(Not Drawn to Scale)
Partition Lines

PARTITIONING OF THE FOLDED PLATE STRUCTURE

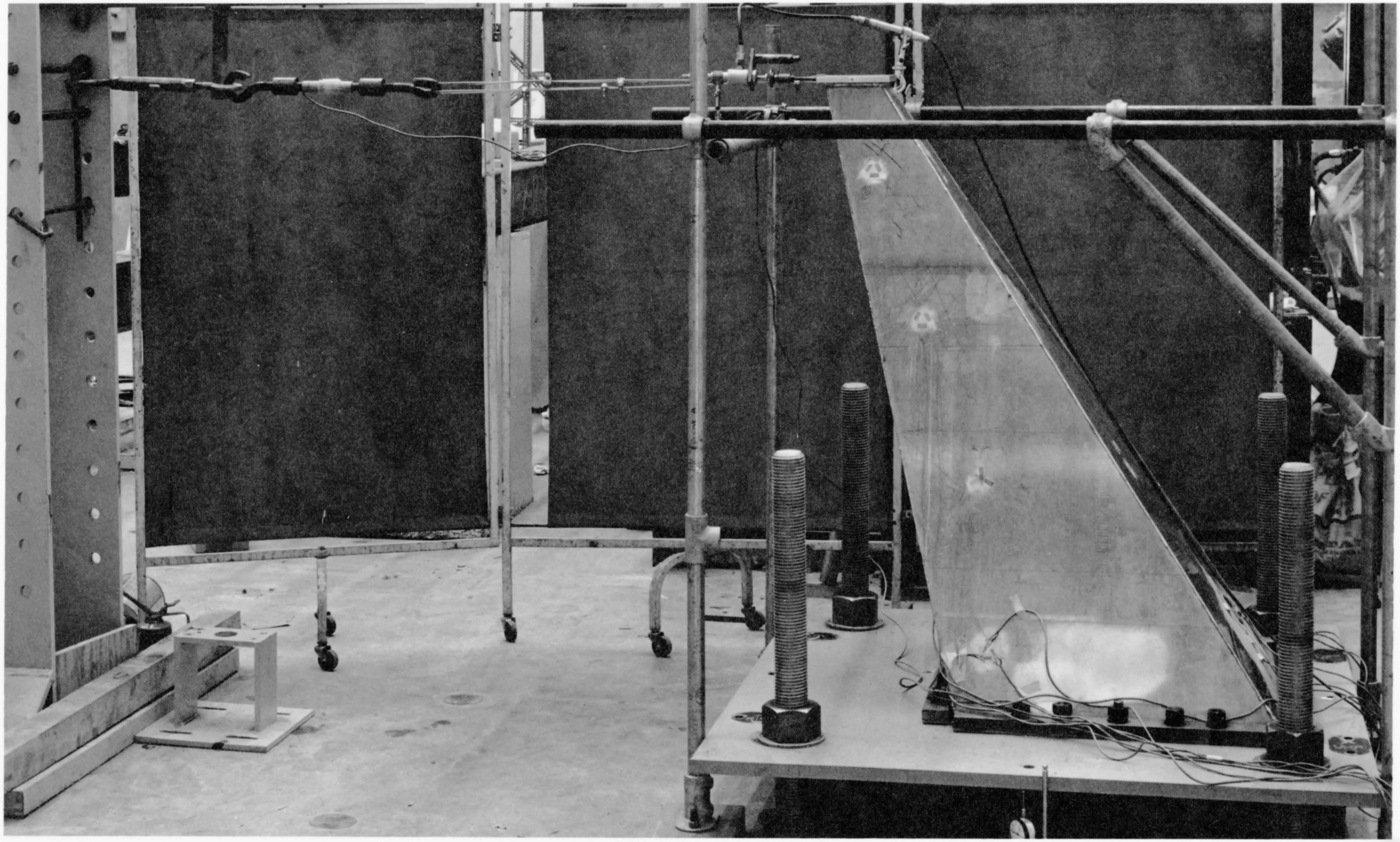


Fig. 21
LOADING THE STRUCTURE IN THE GLOBAL X-DIRECTION AT NODE B

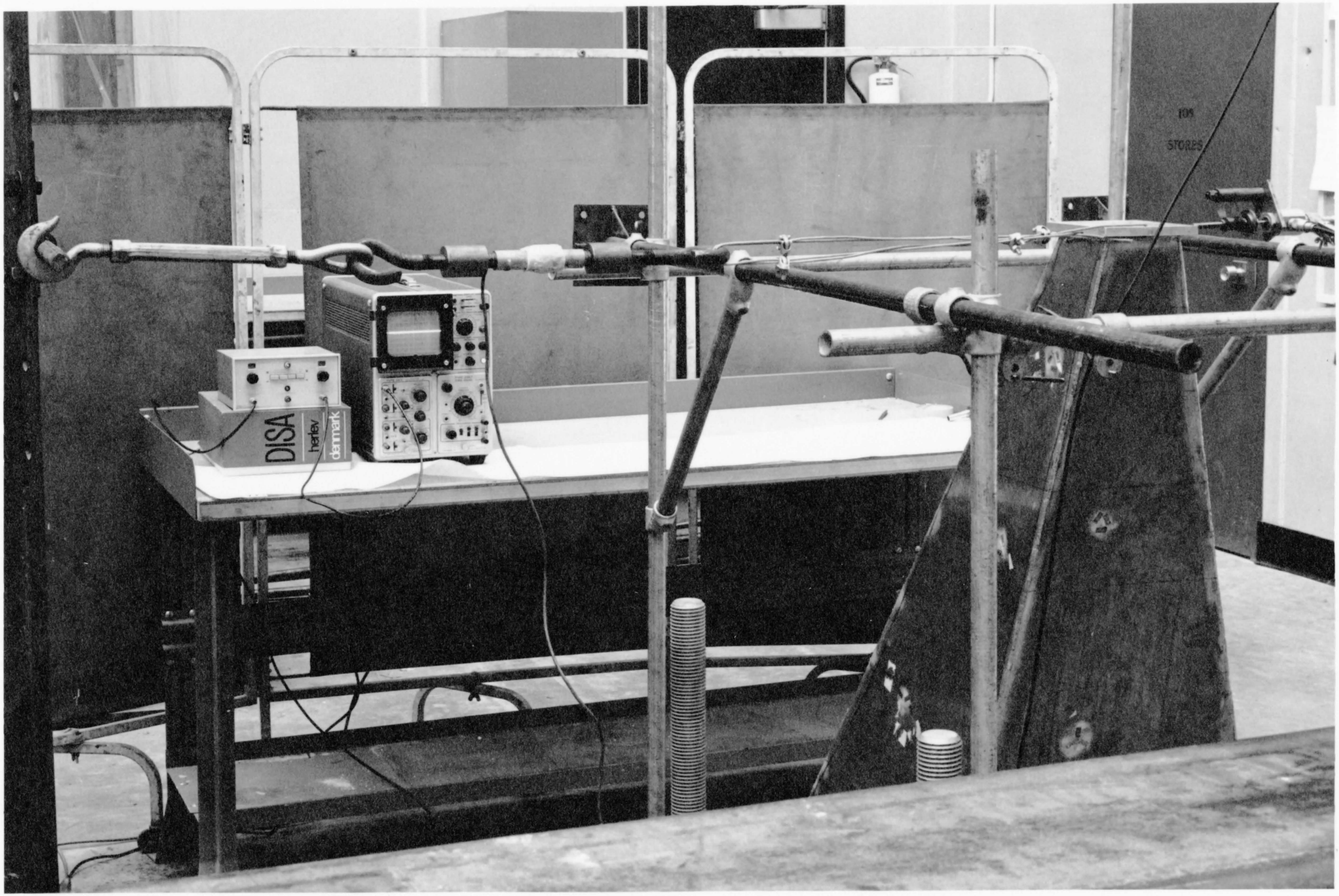


Fig. 22
LOADING THE STRUCTURE IN THE GLOBAL Y-DIRECTION AT NODE B

Fig. 23
LOADING THE STRUCTURE IN THE GLOBAL Z-DIRECTION AT NODE B

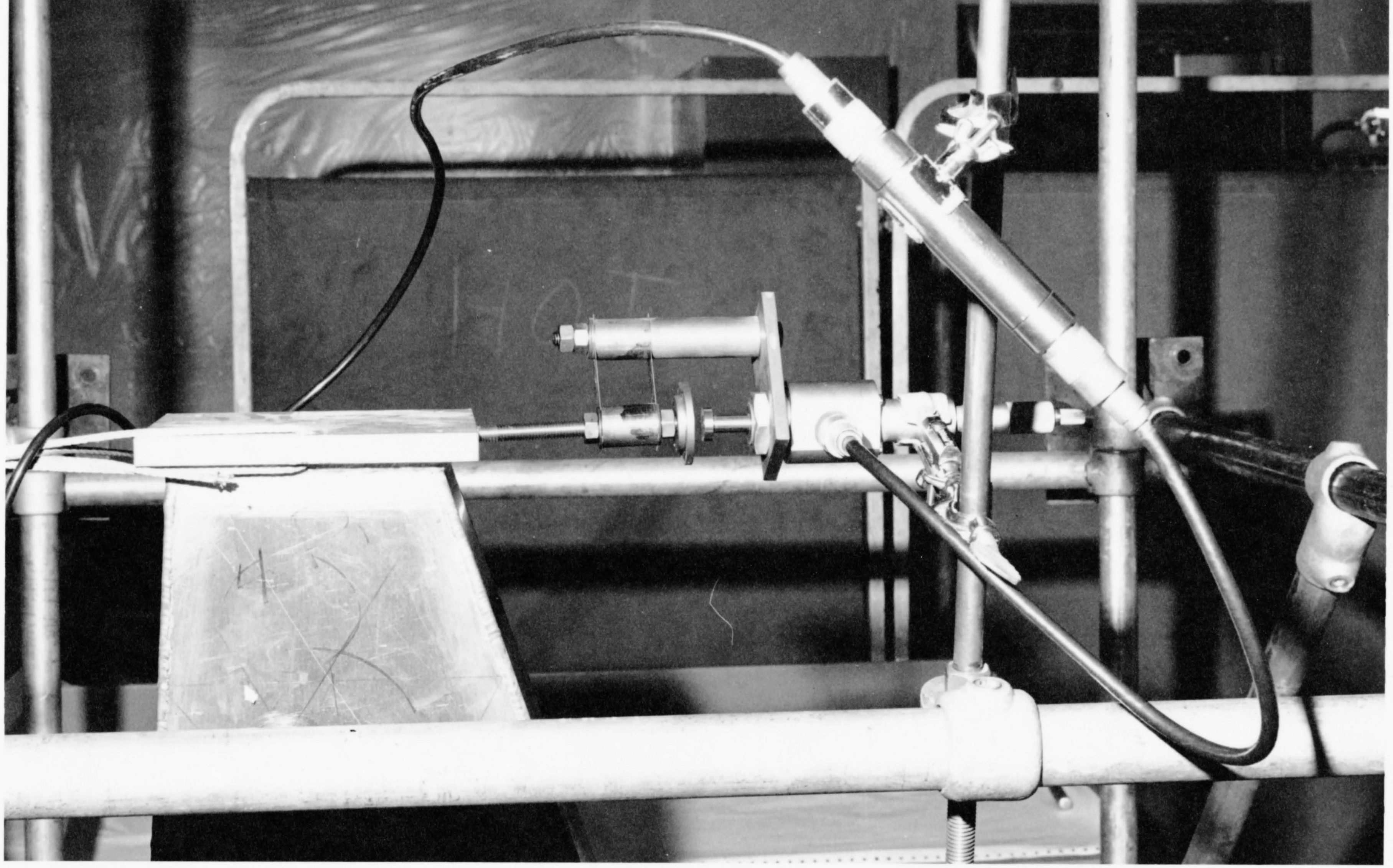


Fig. 24
INSTALLATION OF DEVICE TO ELIMINATE ROTATION ERROR IN LINEAR DISPLACEMENT MEASUREMENT

TABLES

TABLE 1a

NUMBER OF PARTITIONS	DISPLACEMENT FUNCTION		
	TOCHER x 10^{-3} IN.	RAWTANI x 10^{-3} IN.	COWPER x 10^{-3} IN.
5	5.74571	5.74554	5.41974
6	5.92644	5.92607	5.81869
7	6.07618	6.07583	5.96346
8	6.11979	6.11942	6.00359
9	6.16224	6.16179	6.03970
10	6.18404	6.18350	6.05586

DEFLECTION u AT NODAL POINT A FOR DIFFERENT PARTITIONS UNDER LOAD P = 450 POUNDS AT NODE B IN X-DIRECTION

TABLE 1b

NUMBER OF PARTITIONS	DISPLACEMENT FUNCTION		
	TOCHER x 10^{-3} IN.	RAWTANI x 10^{-3} IN.	COWPER x 10^{-3} IN.
5	5.89198	5.89165	5.55223
6	6.07963	6.07929	5.96193
7	6.22992	6.22959	6.10637
8	6.27479	6.27446	6.14820
9	6.31953	6.31915	6.1875
10	6.34341	6.34298	6.20729

DEFLECTION u AT NODAL POINT B FOR DIFFERENT

PARTITIONS UNDER LOAD P = 450 POUNDS AT NODE B IN X-DIRECTION

TABLE 1c

NUMBER OF PARTITIONS	DISPLACEMENT FUNCTION		
	TOCHER x 10^{-3} IN.	RAWTANI x 10^{-3} IN.	COWPER x 10^{-3} IN.
5	5.84844	5.84782	5.50665
6	6.02496	6.03741	5.91337
7	6.19015	6.18938	6.05876
8	6.23502	6.23420	6.10094
9	6.27943	6.27850	6.14067
10	6.30241	6.30135	6.16069

DEFLECTION u AT NODAL POINT C FOR DIFFERENT

PARTITIONS UNDER LOAD P = 450 POUNDS AT NODE B IN X-DIRECTION

TABLE 1d

NUMBER OF PARTITIONS	DISPLACEMENT FUNCTION		
	TOCHER x 10^{-3} IN.	RAWTANI x 10^{-3} IN.	COWPER x 10^{-3} IN.
5	5.80340	5.80306	5.49637
6	5.98983	5.98952	5.87603
7	6.13996	6.13965	6.02024
8	6.18496	6.18465	6.06238
9	6.23011	6.22974	6.10234
10	6.25475	6.25431	6.12292

DEFLECTION u AT NODAL POINT D FOR DIFFERENT PARTITIONS

UNDER LOAD P = 450 POUNDS AT NODE B IN X-DIRECTIONS.

TABLE 2a

NUMBER OF PARTITIONS	DISPLACEMENT FUNCTION		
	TOCHER x 10^{-3} IN.	RAWTANI x 10^{-3} IN.	COWPER x 10^{-3} IN.
5	6.00298	5.74554	6.04780
6	6.20924	5.92607	6.21040
7	6.50159	6.07583	6.40282
8	6.62038	6.11942	6.48533
9	6.71247	6.16179	6.53519
10	6.73920	6.18350	6.55384

DEFLECTION v AT NODAL POINT A FOR DIFFERENT
PARTITIONS UNDER LOAD P = 450 POUNDS AT NODE B IN Y-DIRECTION.

TABLE 2b

NUMBER OF PARTITIONS	DISPLACEMENT FUNCTION		
	TOCHER x 10^{-3} IN.	RAWTANI x 10^{-3} IN.	COWPER x 10^{-3} IN.
5	6.02917	5.89165	6.08143
6	6.23676	6.07929	6.24788
7	6.53764	6.22959	6.43974
8	6.65782	6.27446	6.52282
9	6.75296	6.31915	6.57350
10	6.77916	6.34298	6.56637

DEFLECTION v AT NODAL POINT B FOR DIFFERENT

PARTITIONS UNDER LOAD P = 450 POUNDS AT NODE B IN Y-DIRECTION

TABLE 2c

NUMBER OF PARTITIONS	DISPLACEMENT FUNCTION		
	TOCHER x 10^{-3} IN.	RAWTANI x 10^{-3} IN.	COWPER x 10^{-3} IN.
5	6.01911	5.84782	6.06288
6	6.21985	6.03741	6.22434
7	6.51872	6.18938	6.41724
8	6.64926	6.23420	6.49940
9	6.73256	6.27850	6.54863
10	6.75260	6.30135	6.56637

DEFLECTION v AT NODAL POINT C FOR DIFFERENT

PARTITIONS UNDER LOAD P = 450 POUNDS AT NODE B IN Y-DIRECTION

TABLE 2d

NUMBER OF PARTITIONS	DISPLACEMENT FUNCTION		
	TOCHER $\times 10^{-3}$ IN.	RAWTANI $\times 10^{-3}$ IN.	COWPER $\times 10^{-3}$ IN.
5	5.04234	5.80306	5.09061
6	5.20582	5.98952	5.21156
7	5.46921	6.13965	5.39197
8	5.59522	6.18465	5.46681
9	5.66721	6.22974	5.50598
10	5.67335	6.25431	5.51721

DEFLECTION v AT NODAL POINT D FOR DIFFERENT

PARTITIONS UNDER LOAD P = 450 POUNDS AT NODE B IN Y-DIRECTION

TABLE 3a

NUMBER OF PARTITIONS	DISPLACEMENT FUNCTION		
	TOCHER x 10^{-3} IN.	RAWTANI x 10^{-3} IN.	COWPER x 10^{-3} IN.
5	-2.62369	-2.62367	-2.45768
6	-2.73686	-2.73678	-2.61391
7	-2.81224	-2.81219	-2.73643
8	-2.83594	-2.83586	-2.75810
9	-2.86044	-2.86030	-2.77899
10	-2.87217	-2.87193	-2.78779

DEFLECTION w AT NODE A FOR DIFFERENT PARTITIONS

UNDER LOAD P = 450 POUNDS AT NODE B IN X-DIRECTION

TABLE 3b

NUMBER OF PARTITIONS	DISPLACEMENT FUNCTION		
	TOCHER x 10^{-3} IN.	RAWTANI x 10^{-3} IN.	COWPER x 10^{-3} IN.
5	-2.66484	-2.66470	-2.47168
6	-2.77632	-2.77629	-2.66806
7	-2.85243	-2.85239	-2.74130
8	-2.87601	-2.87600	-2.76169
9	-2.90058	-2.90061	-2.78107
10	-2.91285	-2.91290	-2.78959

DEFLECTION w AT NODE B FOR DIFFERENT PARTITIONS

UNDER LOAD P = 450 POUNDS AT NODE B IN X-DIRECTION

TABLE 3c

NUMBER OF PARTITIONS	DISPLACEMENT FUNCTION		
	TOCHER x 10^{-3} IN.	RAWTANI x 10^{-3} IN.	COWPER x 10^{-3} IN.
5	-2.69571	-2.69431	-2.51334
6	-2.81629	-2.81413	-2.72641
7	-2.89281	-2.89079	-2.79910
8	-2.91735	-2.91518	-2.82164
9	-2.94335	-2.94092	-2.84435
10	-2.95588	-2.95324	-2.85497

DEFLECTION w AT NODE C FOR DIFFERENT PARTITIONS

UNDER LOAD P = 450 POUNDS AT NODE B IN X-DIRECTION

TABLE 3d

NUMBER OF PARTITIONS	DISPLACEMENT FUNCTION		
	TOCHER x 10^{-3} IN.	RAWTANI x 10^{-3} IN.	COWPER x 10^{-3} IN.
5	-1.72717	-1.72695	-1.70237
6	-1.78673	-1.78678	-1.78942
7	-1.83863	-1.83868	-1.83847
8	-1.85510	-1.85515	-1.85589
9	-1.87247	-1.87244	-1.87446
10	-1.88534	-1.88519	-1.88814

DEFLECTION w AT NODE D FOR DIFFERENT PARTITIONS

UNDER LOAD P = 450 POUNDS AT NODE B IN X-DIRECTION

TABLE 4a

DISPLACEMENT FUNCTION	DEFLECTION u AT NODAL POINT A $\times 10^{-3}$ IN.	PERCENTAGE DEVIATION FROM TOCHER	DEFLECTION u AT NODAL POINT B $\times 10^{-3}$ IN.	PERCENTAGE DEVIATION FROM TOCHER
TOCHER	6.18404	-	6.34341	-
RAWTANI	6.1835	0.008732	6.34298	0.00679
COWPER	6.05586	2.073	6.20729	2.1460

Percentage Deviation of Deflection for 10 Partitions at Different Nodal Points Under Load P = 450 pounds in X-Direction from the Tocher Displacement Function

TABLE 4b

DISPLACEMENT FUNCTION	DEFLECTION u AT NODAL POINT C x 10^{-3} IN.	PERCENTAGE DEVIATION FROM TOCHER	DEFLECTION u AT NODAL POINT D	PERCENTAGE DEVIATION FROM TOCHER
TOCHER	6.30241	-	6.25475	-
RAWTANI	6.30135	0.0168	6.25431	0.00703
COWPER	6.16069	2.249	6.12293	2.108

Percentage Deviation of Deflection for 10 Partitions at Different Nodal Points under Load P = 450 pounds in X-Direction from the Tocher Displacement Function

TABLE 4c

DISPLACEMENT FUNCTION	DEFLECTION v AT NODAL POINT A $\times 10^{-3}$ IN.	PERCENTAGE DEVIATION FROM TOCHER	DEFLECTION v AT NODAL POINT B $\times 10^{-3}$ IN.	PERCENTAGE DEVIATION FROM TOCHER
TOCHER	6.73920	-	6.77916	-
RAWTANI	6.73854	0.8045	6.77846	0.0103
COWPER	6.55381	2.751	6.59278	2.749

Percentage Deviation of Deflection for 10 Partitions at Different Nodal Points under Load P = 450 pounds in Y-Direction from the Tocher Displacement Function

TABLE 4d

DISPLACEMENT FUNCTION	DEFLECTION v AT NODAL POINT C $\times 10^{-3}$ IN.	PERCENTAGE DEVIATION FROM TOCHER	DEFLECTION v AT NODAL POINT D $\times 10^{-3}$ IN.	PERCENTAGE DEVIATION FROM TOCHER
TOCHER	6.75260	-	5.67335	-
RAWTANI	6.75192	0.098	5.67265	0.0123
COWPER	6.56634	2.758	5.51718	2.750

Percentage Deviation of Deflection for 10 Partitions at Different Nodal Points under Load P = 450 pounds in Y-Direction from the Tocher Displacement Function

TABLE 4e

DISPLACEMENT FUNCTION	DEFLECTION w AT NODAL POINT A $\times 10^{-3}$ IN.	PERCENTAGE DEVIATION FROM TOCHER	DEFLECTION w AT NODAL POINT B $\times 10^{-3}$ IN.	PERCENTAGE DEVIATION FROM TOCHER
TOCHER	1.97708	-	2.30833	-
RAWTANI	1.96885	0.416	2.32398	0.678
COWPER	1.89116	4.35	2.21765	3.93

Percentage Deviation of Deflection for 10 Partitions at Different Nodal Points under Load P = 450 pounds in Z-Direction from the Tocher Displacement Function

TABLE 4f

DISPLACEMENT FUNCTION	DEFLECTION w AT NODAL POINT C $\times 10^{-3}$ IN.	PERCENTAGE DEVIATION FROM TOCHER	DEFLECTION w AT NODAL POINT D $\times 10^{-3}$ IN.	PERCENTAGE DEVIATION FROM TOCHER
TOCHER	2.13902	-	1.27292	-
RAWTANI	2.13772	0.061	1.27072	0.251
COWPER	2.03268	4.97	1.25935	1.144

Percentage Deviation of Deflection for 10 Partitions at Different Nodal Points under Load P = 450 pounds in Z-Direction from the Tocher Displacement Function

TABLE 5a

DEFLECTIONS AT NODAL POINT A	THEORETICAL VALUES BY TOCHER $\times 10^{-3}$ IN.	EXPERIMENTAL VALUES $\times 10^{-3}$ IN.	PERCENTAGE DEVIATION FROM THEORETICAL VALUES
u_x	6.18404	7.62000	23.30
v_y	6.73920	8.3000	23.20
w_z	1.97708	2.4550	24.20

Comparison of Theoretical and Experimental Deflection at Nodal Point A for Different Load Applications (P_x, P_y, P_z = 450 pounds) at Nodal Point B.

TABLE 5b

DEFLECTIONS AT NODAL POINT C	THEORETICAL VALUES BY TOCHER $\times 10^{-3}$ IN.	EXPERIMENTAL VALUES $\times 10^{-3}$ IN.	PERCENTAGE DEVIATION FROM THEORETICAL VALUES
u_x	6.30241	7.87000	25.00
v_y	6.75260	8.26000	22.40
w_z	2.13902	2.71000	26.80

Comparison of Theoretical and Experimental Deflection at Nodal Point C for Different Loal Application (P_x, P_y, P_z = 450 pounds) at Nodal Point B.

TABLE 6

STRAIN GAGE LOCATION ON ELEMENT NUMBER	THEORETICAL OR EXPERIMENTAL STRESSES	σ_{xx}	σ_{yy}	σ_{xy}	σ_{MAX}	σ_{MIN}
3	THEORETICAL	5.5	243.13	12.899	243.83	4.8
	EXPERIMENTAL	-20.9	220.9	17.45	222.18	-22.18
6	THEORETICAL	39.77	85.01	42.46	103.187	-14.197
	EXPERIMENTAL	87.92	123.19	65.46	173.34	37.76
10	THEORETICAL	-12.26	-243.35	23.99	-9.8	-245.8
	EXPERIMENTAL	-22.997	-254.78	34.91	-17.85	-259.9
13	THEORETICAL	-45.80	-246.17	9.32	-45.36	-246.6
	EXPERIMENTAL	-85.01	-226.09	17.45	-82.89	-228.23

Stresses (IN lbs/IN^2) due to P_x = 450 lb.

TABLE 7a

NODE NUMBERS	DISPLACEMENT FUNCTIONS		
	TOCHER	RAWTANI	COWPER
A	5.18892×10^{-6}	-3.10531×10^{-6}	-3.22563×10^{-5}
B	4.20190×10^{-5}	1.17961×10^{-5}	4.93179×10^{-4}
C	-6.40242×10^{-8}	1.02841×10^{-5}	-2.75839×10^{-5}
D	2.51083×10^{-6}	-2.95351×10^{-5}	-2.31822×10^{-4}

ROTATION θ_x AT NODAL POINTS A, B, C AND D FOR PARTITION 10 UNDER LOAD P = 450 POUNDS AT NODE B IN X-DIRECTION.

TABLE 7b

NODE NUMBER	DISPLACEMENT FUNCTIONS		
	TOCHER	RAWTANI	COWPER
A	2.48585×10^{-4}	2.47152×10^{-4}	1.60388×10^{-4}
B	2.64275×10^{-4}	2.39451×10^{-4}	-8.84948×10^{-6}
C	2.96880×10^{-4}	2.78753×10^{-4}	-1.51542×10^{-4}
D	2.30831×10^{-4}	2.13060×10^{-4}	-9.80626×10^{-6}

ROTATION θ_y AT NODAL POINTS A, B, C AND D FOR PARTITION 10 UNDER LOAD P = 450 POUNDS AT NODE B IN X-DIRECTION

TABLE 7c

NODE NUMBERS	DISPLACEMENT FUNCTIONS		
	TOCHER	RAWTANI	COWPER
A	7.65590×10^{-5}	8.62546×10^{-5}	6.796622×10^{-5}
B	1.01319×10^{-5}	2.25062×10^{-5}	1.57661×10^{-5}
C	7.47683×10^{-6}	-4.41706×10^{-5}	9.02398×10^{-5}
D	1.83419×10^{-5}	1.58114×10^{-5}	7.58379×10^{-5}

ROTATION θ_z AT NODAL POINTS A,B, C AND D FOR PARTITION 10 UNDER LOAD P = 450 POUNDS AT NODE B IN X-DIRECTION.

TABLE 8

DISPLACEMENT FUNCTIONS	TIME SECONDS	CENTRAL MEMORY
TOCHER	100.2	66300
RAWTANI	177.7	71600
COWPER	888.2	113700

TIME AND CENTRAL MEMORY TAKEN BY CDC 6400 COMPUTER FOR 10 PARTITIONS FOR DIFFERENT DISPLACEMENT FUNCTIONS.

BIBLIOGRAPHY

BIBLIOGRAPHY

1. Zienkiewicz, O.C, 'The Finite Element Method in Structural and Continuum Mechanics', McGraw Hill Publishing Co., 1967.

2. Claugh, R.W., 'Stress Analysis', Chapter 7, Edited by Zienkiewicz and Hollister, J. Wiley and Sons, 1965.

3. Tiwari, S.K., M. Eng. Thesis Submitted to McMaster University, Hamilton, Ontario, Canada, 1968.

4. Raghava, R.S., M. Eng. Thesis submitted to McMaster University, Hamilton, Ontario, Canada, 1968.

5. Gurunathan, V., M. Eng. Thesis submitted to McMaster University, Hamilton, Ontario, Canada, 1968.

6. Bhat, M.S., M. Eng. Thesis submitted to McMaster University, Hamilton, Ontario, Canada, 1969.

7. Cowper, G.R., Kosko, E., Linberg, G.H., Olson, M.D., 'A High Precision Triangular Plate Bending Element,' Aeronautical Report LR-514, National Research Council of Canada, Dec. 1968.

8. Turner, M.J., Clough, R.W., Martin, H.C., Topp, L.J., 'Stiffness and Deflection Analysis of Complex Structure', Journal Aero. Sc., 23 pp. 805-823.

9. Greene, B.E., Strome, D.R., Wiekel, R.C., 'Application of the Stiffness Method in the Analysis of Shell Structures', Proc. Aviation Conf., ASME, Los Angeles, March 1961.

10. Przemieniecki, J.S., 'Theory of Matrix Structural Analysis', McGraw Hill 1968.

11. Bazeley, G.R., Cheung, Y.K., Irons, B.M., Zienkiewicz, O.C., 'Triangular Plate Elements in Bending, Conforming and Non-Conforming Solutions', Proc. Conf. Matrix Methods in Structural Mechanics, Air Force Inst, of Technology, Wright-Patterson AFB, Ohio, AFFDL-TR-66-80, 1966.

12. Rawtani, S., Dokainish, M.A., 'Bending of Pretwisted Cantilever Plates', C.A.S.I. Journal, Sept. 1969.

13. Clough, R.W., Tocher, J.L., 'Finite Element Stiffness Matrices for Analysis of Plate Bending', AFFDL-TR-66-80, Conf. Matrix Methods of Structural Analysis, Oct. 1966.

14. Adini, A., Clough, R.W., 'Analysis of Plate Bending by Finite Element Method', Ph.D. dissertation, Dept. of Civil Eng., Univ. of California, 1961.

15. Tocher, J.L., 'Analysis of Plate Bending Using Triangular Elements', Ph.D. Dissertation, Dept. of Civil Eng., Univ. of California, 1962.

16. Cowper, G.R., Kosko, E., Lindberg, G.H., Olson, M.D., 'Formulation of a New Triangular Plate Bending Element,' Canadian Aeronautics and Space Inst. Trans., Vol. 1, No. 2, Sept. 1968.

17. Besseling, J.F., 'Numerical Methods in Stress Analysis', Technische Hogeschool, Delft. 1967.

18. Visser, W., 'The Finite Element Method in Deformation and Heat Conduction Problems', Doctoral Thesis, T.H. Delft. 1968.

19. Clough, R.W., Wilson, E.L., King, I,P., 'Large Capacity Multistory Frame Analysis Programs', Proc. Amer. Soc.

Civ. Eng., 89, ST4, 1963.

20. Perry, C.C., Lisser, H.R., 'The Strain Gauge Primer', McGraw-Hill Publishing Co. Ltd. 1962.

21. Murray, W.M., Stein, P.K., 'Lectures and Laboratory Exercies, Strain Gauge Techniques', Presented at the Dept. of Eng., Univ. of Calif., Los Angeles.

22. Abramonitz, M., Stegun, I.A., ed. Handbook of Mathematical Functions U.S. National Bureau of Standards, Applied Mathematics Series, No. 55, Washington, Govt. Print. Off. 1964, p. 258.

APPENDIX I

STIFFNESS MATRIX OF A TRIANGULAR ELEMENT IN PLANE STRESS

Let a triangle be defined in the x-y plane by three points (x_1,y_1), (x_2,y_2) and (x_3,y_3)

Let Δ = area of the triangle

$$x_{ij} = x_i - x_j$$

and

$$y_{ij} = y_i - y_i, \quad (i,j = 1,2 \text{ or } 3)$$

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

The in-plane stiffness matrix is made up of two parts which can be written as follows.

$$[K_e]_p = [K]_{np} + [K]_{sp}$$

where

$[K]_{np}$ = stiffness due to normal stress

$[K]_{sp}$ = stiffness due to shear stress

$$[K]_{np} = C_1 \begin{bmatrix} y_{32}^2 & & \text{SYMMETRIC} & & & \\ -\mu y_{32}x_{32} & x_{32}^2 & & & & \\ -y_{32}y_{31} & \mu x_{32}y_{31} & y_{31}^2 & & & \\ \mu y_{32}x_{31} & -x_{32}x_{31} & -\mu y_{31}x_{31} & x_{31}^2 & & \\ y_{32}y_{21} & -\mu x_{32}y_{21} & -y_{31}y_{21} & \mu x_{31}y_{21} & y_{21}^2 & \\ -\mu y_{32}x_{21} & x_{32}x_{21} & \mu y_{31}x_{21} & -x_{31}x_{21} & -\mu y_{21}x_{21} & x_{21}^2 \end{bmatrix}$$

where

$$C_1 = \frac{Et}{4\Delta(1-\mu^2)}$$

$$[K]_{sp} = C_2 \begin{bmatrix} x_{32}^2 & & \text{SYMMETRIC} & & & \\ -y_{32}x_{32} & y_{32}^2 & & & & \\ -x_{32}x_{31} & y_{32}x_{31} & x_{31}^2 & & & \\ x_{32}y_{31} & -y_{32}y_{31} & -x_{31}y_{31} & y_{31}^2 & & \\ x_{32}x_{21} & -y_{32}x_{21} & -x_{31}x_{21} & y_{31}x_{21} & x_{21}^2 & \\ -x_{32}y_{21} & y_{32}y_{21} & x_{31}y_{21} & -y_{31}y_{21} & -x_{21}y_{21} & y_{21}^2 \end{bmatrix}$$

where

$$C_2 = \frac{Et}{8\Delta(1+\mu)}$$

APPENDIX II

STIFFNESS MATRIX OF A TRIANGULAR PLATE ELEMENT IN BENDING

$$[K_e]_b = [A^{-1}]^T \int_v [H]^T [D][H] dv [A^{-1}]$$

$$[A] = \begin{bmatrix} 1 & x_1 & y_1 & x_1^2 & x_1y_1 & y_1^2 & x_1^3 & x_1y_1^2+y_1x_1^2 & y_1^3 \\ 0 & 0 & 1 & 0 & x_1 & 2y_1 & 0 & 2x_1y_1+x_1^2 & 3y_1^2 \\ 0 & -1 & 0 & -2x_1 & -y_1 & 0 & -3x_1^2 & -(y_1^2+2x_1y_1) & 0 \\ 1 & x_2 & y_2 & x_2^2 & x_2y_2 & y_2^2 & x_2^3 & x_2y_2^2+y_2x_2^2 & y_2^3 \\ 0 & 0 & 1 & 0 & x_2 & 2y_2 & 0 & 2x_2y_2+x_2^2 & 3y_2^2 \\ 0 & -1 & 0 & -2x_2 & -y_2 & 0 & -3x_2^2 & -(y_2^2+2x_2y_2) & 0 \\ 1 & x_3 & y_3 & x_3^2 & x_3y_3 & y_3^2 & x_3^3 & x_3y_3^2+y_3x_3^2 & y_3^3 \\ 0 & 0 & 1 & 0 & x_3 & 2y_3 & 0 & 2x_3y_3+x_3^2 & 3y_3^2 \\ 0 & -1 & 0 & -2x_3 & -y_3 & 0 & -3x_3^2 & -(y_3^2+2x_3y_3) & 0 \end{bmatrix}$$

The inverse of [A] must be obtained numerically for each triangular element. The matrix product inside the integral can be multiplied and written as

$$[H]^T [D][H] dv =$$

$$z\iint \begin{bmatrix} 0 & & & & & \text{SYMMETRIX} & & & & \\ 0 & 0 & & & & & & & & \\ 0 & 0 & 0 & & & & & & & \\ 0 & 0 & 0 & 4 & & & & & & \\ 0 & 0 & 0 & 0 & 2(1-\mu) & & & & & \\ 0 & 0 & 0 & 4\mu & 0 & 4 & & & & \\ 0 & 0 & 0 & 12x & 0 & 12\mu x & 36x^2 & & & \\ 0 & 0 & 0 & 4(\mu x+y) & 4(1-\mu)(x+y) & 4(x+\mu y) & 12x(\mu x+y) & \begin{matrix}(12-8\mu)(x+y)^2\\ -(1-\mu)8xy\end{matrix} & \\ 0 & 0 & 0 & 12\mu y & 0 & 12y & 36\mu xy & 12(x+\mu y)y & 36y^2 \end{bmatrix} dxdy$$

where $$z = \frac{Et^3}{12(1-\mu^2)}$$

SOME INTEGRATION FORMULAE FOR A TRIANGLE

Let a triangle be defined by three points (x_1,y_1), (x_2,y_2), and (x_3,y_2). Let Δ be the area and $\bar{x},\bar{y}$ be the coordinates of the centroid of the triangle. Let $\bar{x}_1 = x_1-\bar{x}$; $\bar{y}_1 = y_1-\bar{y}$; $\bar{x}_2 = x_2-\bar{x}$; $\bar{y}_2 = y_2-\bar{y}$; $\bar{x}_3=x_3-\bar{x}$; $\bar{y}_3 = y_3-\bar{y}$.

Then the integrals

$$\iint x\,dxdy = \Delta\bar{x}$$

$$\iint y\,dxdy = \Delta\bar{y}$$

$$\iint x^2 dxdy = \frac{\Delta}{12}(\bar{x}_1^2 + \bar{x}_2^2 + \bar{x}_3^2) + \Delta\bar{x}^2$$

$$\iint y^2 dxdy = \frac{\Delta}{12}(\bar{y}_1^2 + \bar{y}_2^2 + \bar{y}_3^2) + \Delta\bar{y}^2$$

$$\iint xy\,dxdy = \frac{\Delta}{12}(\bar{x}_1\bar{y}_1 + \bar{x}_2\bar{y}_2 + \bar{x}_3\bar{y}_3) + \Delta\overline{xy}$$

APPENDIX III - DERIVATION OF ELASTIC PROPERTIES OF TRIANGULAR ELEMENT IN BENDING USING THE RAWTANI DISPLACEMENT FUNCTION.

DERIVATION OF ELASTIC PROPERTIES OF TRIANGULAR ELEMENT IN BENDING USING THE RAWTANI DISPLACEMENT FUNCTION.

The method of approach, using the above displacement function is given in reference [12] in detail.

The idealized structural element is the triangular element, using the rectangular coordinate system, is shown in Figure 17.

The most commonly used cubic polynomial expresssion for transverse deflection in x' and y' is

$$w'(x',y') = \alpha_1 + \alpha_2 x' + \alpha_3 y' + \alpha_4 x'^2 + \alpha_5 x'y' + \alpha_6 y'^2 + \alpha_7 x'^3 + \alpha_8 x'^2 y' + \alpha_9 x'y'^2 + \alpha_{10} y'^3$$

which involves ten arbitrary coefficients and since only nine degrees of freedom are assigned to triangular element in bending, a certain assumption must be made, regarding one of the coefficients. In this case, the extra coefficient is chosen in such a way that transverse slope continuity exists along one of the side of the element, making the displacement function partially conforming [12]. This is achieved by selecting the local coordinates for the element in such a way that the equation to the line along which transverse slope continuity is to be satisfied becomes $x'=0$ and making the coefficient of twisting term $x'y'^2$ to be zero. Thus the displacement function within the triangular element becomes

$$w'(x',y') = \alpha_1 + \alpha_2 x' + \alpha_3 y' + \alpha_4 x'^2 + \alpha_5 x'y' + \alpha_6 y'^2 + \alpha_7 x'^3 + \alpha_8 x'^2 y' + \alpha_9 y'^3.$$

Once the displacement function is selected, the derivation of the element bending stiffness matrix follows the standard procedure. The bending stiffness matrix for the triangular element is as follows.

$$[K_e']_b = \frac{Et^3}{12(1-\mu^2)} [A^{-1}]^T [B] [A^{-1}]$$

where

$$[A] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & y_J' & 0 & 0 & y_J'^2 & 0 & 0 & y_J'^3 \\ 0 & 0 & 1 & 0 & 0 & 2y_J' & 0 & 0 & 3y_J'^2 \\ 0 & -1 & 0 & 0 & -y_J' & 0 & 0 & 0 & 0 \\ 1 & x_K' & y_K' & x_K'^2 & x_K'y_K' & y_K'^2 & x_K'^3 & x_K'^2y_K' & y_K'^3 \\ 0 & 0 & 1 & 0 & x_K' & 2y_K' & 0 & x_K'^2 & 3y_K'^2 \\ 0 & -1 & 0 & -2x_K' & -y_K' & 0 & -3x_K'^2 & -2x_K'y_K' & 0 \end{bmatrix}$$

$$[B] = \begin{bmatrix} 0 & & & & \text{SYMMETRIC} & & & & & \\ 0 & 0 & & & & & & & & \\ 0 & 0 & 0 & & & & & & & \\ 0 & 0 & 0 & 4C_{11} & & & & & & \\ 0 & 0 & 0 & 0 & 2(1-\mu)C_{11} & & & & & \\ 0 & 0 & 0 & 4\mu C_{11} & 0 & 4C_{11} & & & & \\ 0 & 0 & 0 & 12C_{21} & 0 & 12\mu C_{21} & 36C_{31} & & & \\ 0 & 0 & 0 & 4C_{12} & 4(1-\mu)C_{21} & 4\mu C_{12} & 12C_{32} & 4C_{13}+8(1-\mu)C_{31} & & \\ 0 & 0 & 0 & 12\mu C_{12} & 0 & 12C_{12} & 36\mu C_{22} & 12\mu C_{13} & 36C_{13} & \end{bmatrix}$$

and

$$C_{11} = \frac{1}{2} x'_K y'_J$$

$$C_{21} = \frac{1}{6} x'^2_K y'_J$$

$$C_{31} = \frac{1}{12} x'^3_K y'_J$$

$$C_{12} = \frac{1}{6} x'_K y'_J (y'_J + y'_K)$$

$$C_{13} = \frac{1}{12} x'_K y'_J (y'^2_J + y'_J y'_K + y'^2_K)$$

$$C_{22} = \frac{1}{24} x'^2_K y'_J (y'_J + 2y'_K)$$

TRANSFORMATION MATRIX

Referring to Figure 21, the coordinates of vertices i, j, k are (0,0,0), $(0, y'_J, 0)$ and $(x'_K, y'_K, 0)$. It can be shown that

$$y'_J = \sqrt{a_1}$$

$$y'_K = (a_3 + a_1 - a_2)/2\sqrt{a_1}$$

$$x'_K = \sqrt{a_3 - y'^2_K}$$

Where a_1, a_2, a_3 are the squares of the lengths of the sides ij, jk, and ki respectively.

If (l_x, m_x, n_x), (l_y, m_y, n_y) and (l_z, m_z, n_z) denote the direction cosines of the x',y',z' axes respectively with the global axes, the equations for calculating them for the particular system of local coordinate axes chosen are:-

$$A = y_{ji}z_{ki} - y_{ki}z_{ji}$$

$$B = x_{ki}z_{ji} - x_{ji}z_{ki}$$

$$C = x_{ji}y_{ki} - x_{ki}y_{ji}$$

$$F = \sqrt{x_{ji}^2 + y_{ji}^2 + z_{ji}^2}$$

$$G = - \sqrt{A^2 + B^2 + C^2}$$

$$l_z = A/G \qquad m_z = B/G \qquad n_z = C/G$$

$$l_y = x_{ji}/F \qquad m_y = y_{ji}/F \qquad n_y = z_{ji}/F$$

$$l_x = m_y n_z - m_z n_y$$

$$m_x = l_z n_y - n_z l_y$$

$$n_x = l_y m_z - m_y l_z$$

where $y_{ji} = y_j - y_i$ etc

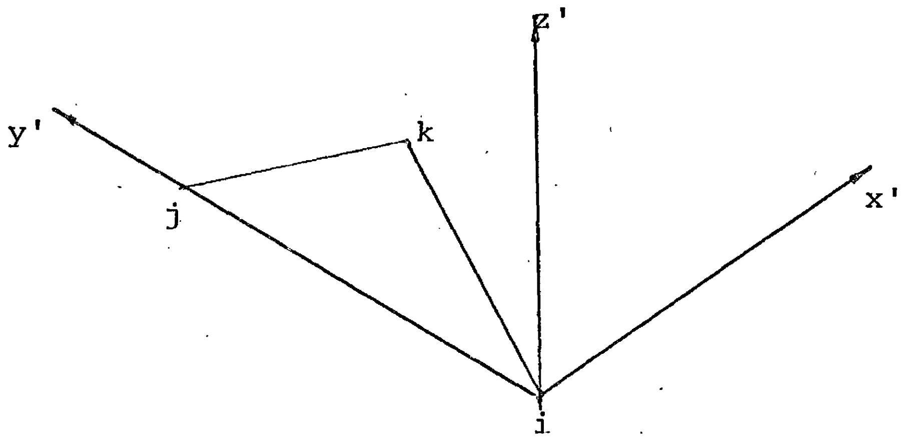


Fig. 21

APPENDIX IV - DERIVATION OF ELASTIC PROPERTIES OF TRIANGULAR ELEMENT IN BENDING USING THE COWPER DISPLACEMENT FUNCTION

DERIVATION OF ELASTIC PROPERTIES OF TRIANGULAR ELEMENT IN BENDING, USING THE COWPER DISPLACEMENT FUNCTION.

The method of approach using the above displacement function is given in reference [7] in detail.

DISPLACEMENT FUNCTION

The transverse deflection $w'(x',y')$ within a triangular element is taken as a quintic polynomial.

$$\begin{aligned} w'(x',y') = {} & \alpha_1 + \alpha_2 x' + \alpha_3 y' + \alpha_4 x'^2 + \alpha_5 x'y' + \alpha_6 y'^2 \\ & + \alpha_7 x'^3 + \alpha_8 x'^2 y' + \alpha_9 x'y'^2 + \alpha_{10} y'^3 + \alpha_{11} x'^4 \\ & + \alpha_{12} x'^3 y' + \alpha_{13} x'^2 y'^2 + \alpha_{14} x'y'^3 + \alpha_{15} y'^4 \\ & + \alpha_{16} x'^5 + \alpha_{17} x'^3 y'^2 + \alpha_{19} x'y'^4 + \alpha_{20} y'^5 \end{aligned} \tag{A.4.1}$$

COORDINATE SYSTEM

The rectangular coordinate system is exclusively used as they are completely adequate and have the merit of great simplicity.

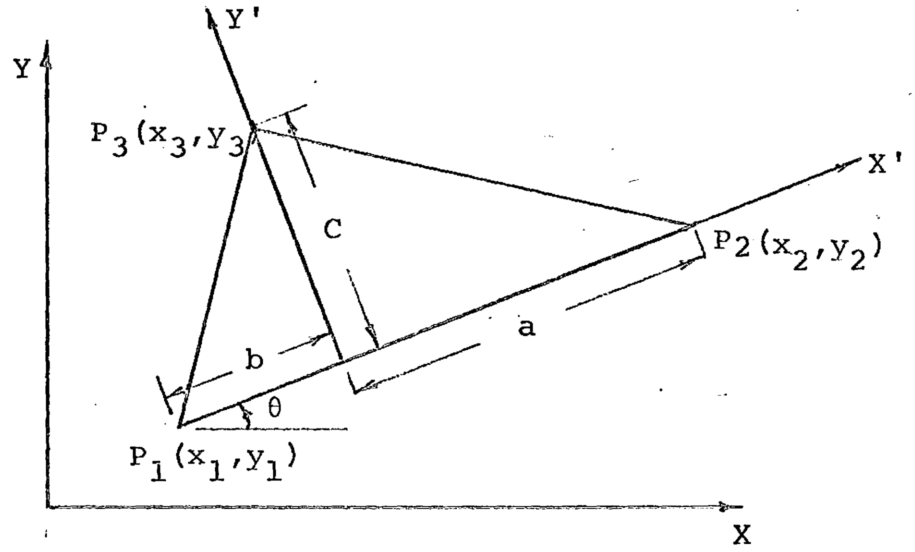


Fig. 22

$$\left.\begin{aligned} \cos\theta &= (x_2 - x_1)/r \\ \sin\theta &= (y_2 - y_1)/r \end{aligned}\right\} \tag{A.4.2}$$

where $$r = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{A.4.3}$$

The dimension

$$\begin{aligned} a &= (x_2 - x_3)\cos\theta - (y_3 - y_2)\sin\theta \\ &= \{(x_2 - x_3)(x_2 - x_1) + (y_2 - y_3)(y_2 - y_1)\}/r \end{aligned} \tag{A.4.4}$$

Similarly

$$b = \{(x_3 - x_1)(x_2 - x_1) + (y_3 - y_1)(y_2 - y_1)\}/r \tag{A.4.5}$$

$$c = \{(x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_3 - y_1)\}/r \tag{A.4.6}$$

NUMBER OF DEGREES OF FREEDOM

The eighteen generalized displacement for the finite element (six at each node) are the transverse deflection and its first and second derivatives at each node. They may be expressed in a column vector in local coordinates as $\{W^e\}$ and whose transpose is

$$\{W^e\}^T = [w_1', w_{x'}, w_{y'}, w_{x'x'}, w_{x'y'}, w_{y'y'}, w_2' \ldots\ldots\ldots, w_3' \ldots\ldots] \tag{A.2.7}$$

It can also be written as a function of nodal coordinates in matrix form

$$\{W^e\} = [T_1]\{\alpha\} \tag{A.4.8}$$

When equation (A.4.8) is augmented by the condition of cubic variation of nodal slope along the edge P_1P_3 and P_2P_3, it can be written as

$$\begin{Bmatrix} W^e \\ 0 \\ 0 \end{Bmatrix} = [T_2]\{\alpha\} \tag{A.4.9}$$

$$\{\alpha\} = [T_2]^{-1} \begin{Bmatrix} W^e \\ 0 \\ 0 \end{Bmatrix} \tag{A.4.10}$$

which is equivalent to

$$\{\alpha\} = [T_3]\{W^e\} \tag{A.4.11}$$

BENDING STIFFNESS MATRIX

The stiffness matrix of the finite element may be dedrived from strain energy which is given as

$$U_e = \frac{1}{2} D \iint \{W^2_{x'x'} + W^2_{y'y'} + 2\mu W_{x'x'} W_{y'y'} + 2(1-\mu) w^2_{x'y'}\} \, dx'dy' \tag{A.4.12}$$

for classical bending of uniform isotropic plate. When equation (A.4.1) is substituted in equation (A.4.12) and carried out the necessary integrations, the strain-energy becomes

$$U_e = \frac{1}{2} D\{\alpha\}^T [K]\{\alpha\} \tag{A.4.13}$$

Consider a typical term from equation (A.4.12), let

$$U'_e = \frac{1}{2} D \iint W^2_{x'x'} \, dx'dy' \tag{A.4.14}$$

Writing equation (A.4.1) in abbreviated form again

$$W = \sum_{i=1}^{20} \alpha_i x'^{m_i} y'^{n_i} \tag{A.4.15}$$

Thus

$$W_{x'x'} = \sum_{i=1} \alpha_i m_i (m_i - 1) x'^{m_i - 2} y'^{n_i} \tag{A.4.16}$$

$$W^2_{x'x'} = \sum_i \sum_j \alpha_i \alpha_j m_i m_j (m_i - 1)(m_j - 1) x'^{mi+mj-4} y'^{ni+nj} \tag{A.4.17}$$

Integrating equation (A.4.17) w.r.t. x' and y'

$$\iint W^2_{x'x'} dx'dy' = \sum_i \sum_j \alpha_i \alpha_j m_i m_j (m_i - 1)(m_j - 1) \iint x'^{m_i + m_j - 4} y'^{n_i + n_j} dx'dy' \tag{A.4.18}$$

Using Euler's beta function [22], it can be shown that

$$\iint x'^{m}\, y'^{n}\, dx'dy' = F(m,n) \tag{A.4.19}$$

where

$$F(m,n) = c^{m+1}\{a^{m+1} - (-b)^{m+1}\}\,\frac{m!n!}{(m+n+2)!} \tag{A.4.20}$$

Thus equation (A.4.18) becomes

$$\iint W^2_{x'x'}\, dx'dy' = \sum_i \sum_j \alpha_i \alpha_j \{m_i m_j (m_i-1)(m_j-1) F(m_i+m_j-4, n_i+n_j)\} \tag{A.4.21}$$

Other terms in equation (A.4.12) can be evaluated similarly and the element matrix [k] can be deduced as follows

$$\begin{aligned} k_{ij} = {} & m_i m_j (m_i-1)(m_j-1) F(m_i+m_j-4, n_i+n_j) + n_i n_j (n_i-1)(n_j-1) \\ & F(m_i+m_j, n_i+n_j-4) + \{2(1-\mu) m_i m_j n_i n_j + \mu m_i n_j (m_i-1)(n_j-1) \\ & + \mu m_j n_i (m_j-1)(N_i-1)\} F(m_i+m_j-2, n_i+n_j-2) \end{aligned} \tag{A.4.22}$$

Substituting equation (A.4.11) in equation (A.4.13), the strain energy can be expressed in $\{W^e\}$

$$U_e = \frac{1}{2} D \{W^e\}^T [K'_e] \{W^e\} \tag{A.4.23}$$

where $$[K'_e] = [T_3]^T [k] [T_3] \tag{A.4.24}$$

let {W} be the generalized displacements in global coordinates.

$$\{W^e\} = [R]\{W\} \tag{A.4.25}$$

Hence

$$U_e = \frac{1}{2} D\{W\}^T [K_e] \{W\} \tag{A.4.26}$$

where

$$\begin{aligned} [K_e] &= [R]^T [K'_e] [R] \\ &= [R]^T [T_3] [k] [T_3] [R] \end{aligned} \tag{A.4.27}$$

where

$$[R] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 & 0 & 0 \\ 0 & -\sin\theta & \cos\theta & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos^2\theta & 2\sin\theta\cos\theta & \sin\theta \\ 0 & 0 & 0 & -\sin\theta\cos\theta & \cos^2\theta-\sin^2\theta & \sin\theta\cos\theta \\ 0 & 0 & 0 & \sin^2\theta & -2\sin\theta\cos\theta & \cos\theta \end{bmatrix}$$

TRANSFORMATION MATRIX $[T_2]$

$$
\begin{array}{cccccccccccccccccccc}
1 & -b & 0 & b^2 & 0 & 0 & -b^3 & 0 & 0 & 0 & b^4 & 0 & 0 & 0 & 0 & -b^5 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & -2b & 0 & 0 & 3b^2 & 0 & 0 & 0 & -4b^3 & 0 & 0 & 0 & 0 & 5b^4 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & -b & 0 & 0 & b^2 & 0 & 0 & 0 & -b^3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 2 & 0 & 0 & -6b & 0 & 0 & 0 & 12b^2 & 0 & 0 & 0 & 0 & -20b^3 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & -2b & 0 & 0 & 0 & 3b^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & -2b & 0 & 0 & 0 & 2b^2 & 0 & 0 & 0 & -2b^3 & 0 & 0 & 0 \\
1 & a & 0 & a^2 & 0 & 0 & a^3 & 0 & 0 & 0 & a^4 & 0 & 0 & 0 & 0 & a^5 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 2a & 0 & 0 & 3a^2 & 0 & 0 & 0 & 4a^3 & 0 & 0 & 0 & 0 & 5a^4 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & a & 0 & 0 & a^2 & 0 & 0 & 0 & a^3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 2 & 0 & 0 & 6a & 0 & 0 & 0 & 12a^2 & 0 & 0 & 0 & 0 & 20a^3 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 2a & 0 & 0 & 0 & 3a^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 2a & 0 & 0 & 0 & 2a^2 & 0 & 0 & 0 & 2a^3 & 0 & 0 & 0 \\
1 & 0 & c & 0 & 0 & c^2 & 0 & 0 & 0 & c^3 & 0 & 0 & 0 & 0 & c^4 & 0 & 0 & 0 & 0 & c^5 \\
0 & 1 & 0 & 0 & c & 0 & 0 & 0 & c^2 & 0 & 0 & 0 & 0 & c^3 & 0 & 0 & 0 & 0 & c^4 & 0 \\
0 & 0 & 1 & 0 & 0 & 2c & 0 & 0 & 0 & 3c^2 & 0 & 0 & 0 & 0 & 4c^3 & 0 & 0 & 0 & 0 & 5c^4 \\
0 & 0 & 0 & 2 & 0 & 0 & 0 & 2c & 0 & 0 & 0 & 0 & 2c^2 & 0 & 0 & 0 & 0 & 2c^3 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 2c & 0 & 0 & 0 & 0 & 3c^2 & 0 & 0 & 0 & 0 & 4c^3 & 0 \\
0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 6c & 0 & 0 & 0 & 0 & 12c^2 & 0 & 0 & 0 & 0 & 20c^3 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5a^4c & 3a^2c^3-2a^4c & -2ac^4+3a^3c^2 & c^5-4a^2c^3 & 5ac^4 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5b^4c & 3b^2c^3-2b^4c & 2bc^4-3b^3c^2 & c^5-4b^2c^3 & -5bc^4
\end{array}
$$

APPENDIX V - COMPUTER PROGRAM DOCUMENTATION

COMPUTER PROGRAM DOCUMENTATION

The following data are required for a general arbitrary shell structure.

1. The total number of elements representing the structure and the total number of nodal points interconnected between elements.

2. Number of degrees of freedom for each node. For example, zero if the node is fixed, otherwise 5 or 6 for the Tocher and Rawtani displacement functions; appropriate curvatures if the node is fixed, otherwise 8 or 9 for the Cowper displacement function.

3. Material properties of the structure; Young's modulus and Poisson's ratio.

4. Thickness of each finite element.

5. The coordinate of the nodal points of each finite element in local coordinate system.

6. Transformation matrix for each finite element.

7. External loads (load-vector).

It is important to determine the material properties before the actual computation. The above items 5 and 6 constitute enormous amount of data to be supplied, thus it is easy to make errors in punching the data cards. This kind of error can be avoided for this particular structure under consideration by writing a subroutine to generate the data automatically. Also it is possible to use the same programme repeatedly for any numbers of times when the structure is idealised into more and more refined subdivisions. If the plate coordinate system is employed in this particular case for the Tocher displacement function, only four different types of transformation matrices will be required. Similarly a different local coordinate system is used for each finite element (as in the Rawtani and Cowper displacement functions (Figs. 17 and 18) it is possible to generate the necessary transformation matrices as the coordinates of the nodal points are known.

As the solution by direct stiffness assembly is limited to about forty nodes for CDC 6400 computer [6], convergence to the true results is not likely. Hence method of recursion, taking advantage of sparse nature of stiffness matrix, is adopted. The structure is divided into a number of partitions. The section in each face is again subdivided into triangular elements into three different patterns, namely two triangles, three triangles and four triangles. As the four triangular pattern (Fig. 19) gives better results than the other two, the former pattern is used throughout the analysis for all three displacement functions. In this case, the structure is completely defined by the inclinations of the four faces, the height, the thickness and the base width. Hence they are the necessary parameters for the computation.

SUBROUTINE FORT

This subroutine generates coordinates of finite elements (with respect to local plate coordinate system for the Tocher and Cowper displacement functions and with respect to global coordinate system for the Rawtani displacement function) for this particular problem.

SUBROUTINE PLAN

This subroutine calculates in-plane stiffness matrix (6x6) for each triangular element in terms of nodal coordinates and material properties. Expressions for the in-plane stiffness matrix are obtained from reference [10].

SUBROUTINE BEND

This subroutine calculates the bending stiffness matrix (9x9) for each triangular element for the Tocher and Rawtani displacement functions. For the Cowper displacement function this subroutine calculates the bending stiffness matrix (18x18) for each triangular element, using quintic displacement function.

SUBROUTINE GROUP

For shell problems, both in-plane and bending forces have to be considered. Hence in-plane stiffness and bending stiffness matrices are grouped before solving the problem. This subroutine combines in-plane stiffness matrix (6x6) and bending stiffness matrix (9x9) to form the combined element stiffness matrix (18x18) of the triangular element for the Tocher and Rawtani displacement functions. When the Cowper displacement function is used, in-plane stiffness matrix (6x6) and bending stiffness matrix (18x18) are combined to form the combined element stiffness matrix (27x27) of the triangular element.

SUBROUTINE ASSEBL

The combined stiffness matrix (18x18) or (27x27) is divided into nine submatrices corresponding to the three vertices. Since a partitioning technique is employed, only the elements involved in a particular partition are considered at a time. The submatrices are taken up one by one and transformed to global coordinates as described previously, taking special care of the nodal points where θ_z is missing. The submatrices $[K_i]$ and $[C_i]$ of the assembled stiffness matrix are written on magnetic tape 2 as soon as they are generated. Partitions are taken up one by one in DO 36.

SUBROUTINE PRDMAT

This subroutine finds the multiplication of two matrices.

SUBROUTINE PRTMAT

This subroutine finds the multiplication of transposed and ordinary matrices.

SUBROUTINE ALMD

This subroutine calculates the transformation matrix of each triangular element in its local coordinate system.

SUBROUTINE TRANF1

This subroutine brings back the stiffness matrix calculated in local coordinates to plate coordinates before assembly of overall stiffness matrix in each partition.

SUBROUTINE TRANF2

This subroutine transforms (9x9) submatrix of each node from plate coordinate to global coordinate system.

SUBROUTINE DEFL

This subroutine separates element nodal displacements from system displacements, obtained in global coordinates and transforms them back to element or local coordinate system.

SUBROUTINE STRESS

This subroutine calculates the stress matrices for each element. The element nodal displacements obtained from DEFL are used to calculate the stresses. There are two stages in calculation corresponding to in-plane and bending deformation. Stresses are calculated at the centroid of each triangular element.

SUBROUTINE INVMAT

This is a library subroutine used for the inversion of any matrix.

FUNCTION FACT

This function calculates the product of the factorial of an integer.

FUNCTION G

This function calculates the function value of the double integral $\iint x^m y^n dxdy$ by Euler's beta function.

FUNCTION NSUM

This function calculates the total number of degrees of freedom from the beginning of the node-system to a particular node.

MAIN PROGRAM

The main program utilises the above subprograms suitable for different displacement functions namely Tocher, Rawtani and Cowper. The subroutines PLAN, BEND, GROUP and ASSEBL are called in a do-loop for each element one by one. Since a partitioning technique is employed only the elements involved in a particular partition are considered at a time. The elements are taken up one by one in DO 36. The submatrices $[K_i]$ and $[C_i]$ of the assembled stiffness matrix are written on magnetic tape 2 as soon as they are generated. Partitions are taken up one by one in DO 35.

SOLUTION OF EQUATIONS

The method of solution of tridiagonalization is outlined in section 3.9 and in reference [1] in detail. The solution of equation is obtained from subroutine RECUR. The submatrices $[K_i]$ and $[C_i]$ are read from tape 2 and forward elimination is done in DO 40. The process is reversed for backward substitution in DO 350.

The residuals are calculated in order to check the errors introduced in the solutions due to rounding off and truncation as follows.

$$\{R\} = \{P\} - [K]\{\alpha\}$$

The residuals $\{R\}$ are compared with $\{P\}$ which serves as a check for the accuracy of the method of solution.

CALCULATION OF STRESSES

Subroutine RECUR calculates the nodal displacements of the triangular elements in global coordinates. These nodal displacements are written on magnetic tape 1, partition by partition, the order of sequence reversed. They are now read in the proper order and subroutine DEFL calculates the element nodal displacements and transforms them back to local coordinates. Once all the nodal displacements in their respective local coordinates are known, the stresses at the centroid of the element are calculated in subroutine STRESS.

APPENDIX VI - COMPUTER PROGRAMS

```
A4463,LC7000,T2000,CM75000.                                          WIN M.M
RUN(S)
SETINDF.
LOADER(PPLOADR)
REDUCE.
LGO.
'          6400 END OF RECORD
      PROGRAM TST (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE1,TAPE2,
     1TAPE4)
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     PROGRAM TO CALCULATE DEFLECTIONS AND STRESSES IN A SHELL BY
C     THE METHOD TRIDIAGONALIZATION
C     RAWTANI POLYNOMIAL IS USED AS THE DISPLACEMENT FUNCTION.
C     MAIN PROGRAM
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      DIMENSION NSTART(15),NEND(15),NFIRST(15),NLAST(15),NFREE(100)
      DIMENSION THICK(200),NI(200,3),X(200,3),Y(200,3)
      DIMENSION PP(500,2),TNF(200,3,3)
      DIMENSION HH(15)
       DIMENSION Z(200,3),ALMD(3,3)
      DIMENSION A(55,55),C(55,55),XEL(18,2)
      DIMENSION PK(6,6),BK(9,9),ST(18,18)
      COMMON NOD,NELEM,EM,V,NSTART,NEND,NFREE,TNF,PP,X,Y,Z,ALMD
      READ(5,14) NPROB
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     READING DATA
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      DO 1500 NTL=1,NPROB
      WRITE(6,9) NTL
 9    FORMAT(1H1,10X,*PROBLEM NO.*,I5,//)
C     * * * * * * * * * * * * * * * * * * * * * * * * * * *   * * * * *
      READ(5,10) NPART,NOD,NELEM,NCOLN,N6,NXL
C     * * * * * * * * * * * * * * * * * * * * * * * * * * *   * * * * *
 10   FORMAT(20I4)
C     NR IS THE TOTAL NUMBER OF DEGREES OF FREEDOM FOR THE STRUCTURE
      NR=N6*6+(NOD-N6)*5
C     * * * * * * * * * * * * * * * * * * * * * * * * * * *   * * * * *
      READ(5,13) EM,V
 13   FORMAT(2E15.3)
      READ(5,10) ((NSTART(I),NEND(I),NFIRST(I),NLAST(I)),I=1,NPART)
      READ(5,14) (NFREE(I),I=1,NOD)
 14   FORMAT(40I2)
      DO 12 I=1,NR
      DO 12 J=1,NCOLN
 12   PP(I,J)=0.
      DO 8 II=1,NXL
      READ(5,11) (I,(PP(I,J),J=1,NCOLN))
 11   FORMAT(I4,5F10.0)
 8    CONTINUE
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     INPUT SIMPLIFIED FOR THIS STRUCTURE
C     IN THE GENERAL CASE THE NODE NUMBERS NI AND THE TRANSFORMATION
C     MATRICES MUST BE DIRECTLY READ IN
```

```
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
       NLN=NPART*16
C     NLN IS THE TOTAL NUMBER OF ELEMENTS ON THE LATERAL SURFACE OF THE
C     PYRAMID.
       NLNN=NLN+1
C     READ THE NODE NUMBERS FOR THE ELEMENTS IN THE FIRST TWO PARTITIONS
C     FROM BOTTOM.
       READ(5,16)((NI(I,J),J=1,3),I=1,32)
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     RAWTANI POLYNOMIAL IS USED AS DEFLECTION FUNCTION.
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 16    FORMAT(40I2)
C      CALCULATE THE NODE NUMBERS FOR THE REMAINING ELEMENTS ON THE
C      OUTER SURFACE
       DO 17 NPT=3,NPART
       DO 17 I=1,3
       DO 17 J=1,16
       KK=J+(NPT-2)*16
       K=J+(NPT-1)*16
 17    NI(K,I)=NI(KK,I)+8
C      READ THE NODE NUMBERS FOR THE LAST SIX ELEMENTS.
       READ(5,16) ((NI(I,J),J=1,3),I=NLNN,NELEM)
       DO 18 I=1,NLN
       THICK(I)=0.125
 18    CONTINUE
C      THICKNESS OF THE LAST SIX   ELEMENTS IS 0.5 INCHES
       DO 19 I=NLNN,NELEM
       THICK(I)=0.5
 19    CONTINUE
C      READ THE ANGLES OF INCLINATIONS OF THE FOUR FACES, THE BASE WIDTH
C      AND THE TOTAL HEIGHT
       READ(5,20)AT,BT,CT,DT,H,AA
 20    FORMAT(8F10.0)
C      CONVERT ANGLES INTO RADIANS
       ATR=AT*3.1415926/180.
       BTR=BT*3.1415926/180.
       CTR=CT*3.1415926/180.
       DTR=DT*3.1415926/180.
       ATR=ATAN(48./31.5)
       BTR=ATAN(48./4.)
       CTR=3.1416-ATAN(48./12.)
       DTR=ATAN(48./15.5)
C      READ THE HEIGHTS OF THE PARTITIONS
       READ(5,20)(HH(I),I=1,NPART)
C      CALCULATION OF ELEMENT COORDINATES
       BB=AA
       DO 21 NPT=1,NPART
C      NST IS THE FIRST ELEMENT IN EACH PARTITION
       NST=1+(NPT-1)*16
        NN=NST+15
        HI=0.0
       HHH=HH(NPT)
       IF(NPT.EQ.1) GO TO 22
       HP=HH(NPT-1)
       AA=AA-HP*COS(DTR)/SIN(DTR)-HP*COS(BTR)/SIN(BTR)
       BB=BB-HP*COS(ATR)/SIN(ATR)-HP*COS(CTR)/SIN(CTR)
C      CALL SUBROUTINE TO CALCULATE THE NODAL CO-ORDINATES FOR EACH TRIANGLE
  22   CALL FORT(AA,BB,ATR,BTR,CTR,DTR,HHH,NST)
```

```
      IF(NPT.EQ.1) GO TO 21
      HI=HI+HH(NPT-1)
      DO 112 I=NST,NN
      DO 112 J=1,3
      XN=X(I,J)+HI*COS(ATR)/SIN(ATR)
      YN=Y(I,J)+HI*COS(BTR)/SIN(BTR)
      ZN=Z(I,J)+HI
      X(I,J)=XN
      Y(I,J)=YN
      Z(I,J)=ZN
  112 CONTINUE
 21   CONTINUE
      DO 118 I=NLNN,NELEM
      DO 118 J=1,3
 118  Z(I,J)=0.0
C     READ THE NODAL CO-ORDINATES FOR THE LAST  SIX  ELEMENTS
      READ(5,25)((X(I,J),J=1,3),I=NLNN,NELEM)
      READ(5,25)((Y(I,J),J=1,3),I=NLNN,NELEM)
 25   FORMAT(8F10.0)
      DO 119 I=NLNN,NELEM
      DO 119 J=1,3
      XN=X(I,J)+(HI+HH(NPART))*COS(ATR)/SIN(ATR)
      YN=Y(I,J)+(HI+HH(NPART))*COS(BTR)/SIN(BTR)
      ZN=Z(I,J)+(HI+HH(NPART))
      X(I,J)=XN
      Y(I,J)=YN
      Z(I,J)=ZN
  119 CONTINUE
C     INITIALIZE THE ARRAY
      DO 31 NM=1,NELEM
      DO 31 I=1,3
      DO 31 J=1,3
 31   TNF(NM,I,J)=0.
C     TRANSFORMATION MATRIX FOR FACE 1
      DO 27  I=1,4
      DO 27 NM=I,NLN,16
      TNF(NM,1,2)=-1.
      TNF(NM,2,1)=COS(ATR)
      TNF(NM,2,3)=SIN(ATR)
 27   CONTINUE
C     TRANSFORMATION MATRIX FOR FACE  2
      DO 28 I=5,8
      DO 28 NM=I,NLN,16
      TNF(NM,1,1)=1.
      TNF(NM,2,2)=COS(BTR)
 28   TNF(NM,2,3)=SIN(BTR)
C     TRANSFORMATION MATRIX FOR FACE 3
      DO 29 I=9,12
      DO 29 NM=I,NLN,16
      TNF(NM,1,2)=1.
      TNF(NM,2,1)=-COS(CTR)
 29   TNF(NM,2,3)=SIN(CTR)
C     TRANSFORMATION MATRIX FOR FACE 4
      DO 30 I=13,16
      DO 30 NM=I,NLN,16
      TNF(NM,1,1)=-1.
      TNF(NM,2,2)=-COS(DTR)
 30   TNF(NM,2,3)=SIN(DTR)
```

```
      DO 32 NM=NLNN,NELEM
      DO 32 I=1,3
      DO 32 J=1,3
       TNF(NM,I,J)=0.0
       IF(I.EQ.J) TNF(NM,I,J)=1.0
   32 CONTINUE
C     CALCULATE THE THIRD ROW OF EACH TRANSFORMATION MATRIX IN TERMS
C     OF THE ELEMENTS OF THE FIRST TWO ROWS
      DO 33 NM=1,NELEM
      TNF(NM,3,1)=TNF(NM,1,2)*TNF(NM,2,3)-TNF(NM,2,2)*TNF(NM,1,3)
      TNF(NM,3,2)=TNF(NM,2,1)*TNF(NM,1,3)-TNF(NM,1,1)*TNF(NM,2,3)
 33   TNF(NM,3,3)=TNF(NM,1,1)*TNF(NM,2,2)-TNF(NM,2,1)*TNF(NM,1,2)
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     PRINTING DATA
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      WRITE(6,400) H
 400  FORMAT(10X,20HHEIGHT OF STRUCTURE=,F10.3)
      WRITE(6,401) EM,V
 401  FORMAT(/,10X,19HMATERIAL PROPERTIES,//,10X,2HE=,E12.4,5X,2HV=,
     1E12.4)
      WRITE(6,403) NPART,NOD,NELEM,NCOLN,NXL
 403  FORMAT(//,10X,25HNPART,NOD,NELEM,NCOLN,NXL,/,5I20)
      WRITE(6,402)(HH(I),I=1,NPART)
 402  FORMAT(/,10X,25HHEIGHTS OF PARTITIONS ARE,/,(6F15.3))
      DO 442 J=1,NCOLN
      DO 442 NL=1,NR
      IF(PP(NL,J).LE.1.0E-8) GO TO 442
      WRITE(6,443) J
 443  FORMAT(/,5X,*EXTERNAL LOAD VECTOR NO.*,I6)
      WRITE(6,446) PP(NL,J),NL
 446  FORMAT(5X,F15.4,*LBS AT*,I6)
 442  CONTINUE
      IF(NTL.NE.1) GO TO 8888
      WRITE(6,410)
 410  FORMAT(1H1,5X,29HNODAL PATTERN AND COORDINATES,//,5X,11HELEMENT NO
     1.,10X,2HNI,32X,1HX,45X,1HY,//)
      DO 406 I=1,NELEM
      WRITE(6,405) I,(NI(I,J),J=1,3),(X(I,J),J=1,3),(Y(I,J),J=1,3)
 406  CONTINUE
 405  FORMAT(I6,10X,3I8,6F15.3)
      WRITE(6,430) (NFREE(I),I=1,NOD)
 430  FORMAT(//,20I5)
      WRITE(6,431) ((NSTART(I),NEND(I),NFIRST(I),NLAST(I)),I=1,NPART)
 431  FORMAT(//,4I25)
      WRITE(6,425) ATR,BTR,CTR,DTR
 425  FORMAT(//,5X,10HTHE ANGLES,//,4E20.3)
      DO 436 NM=1,NELEM
      WRITE(6,437)((TNF(NM,I,J),J=1,3),I=1,3)
 437  FORMAT(9F13.3,/)
 436  CONTINUE
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     FORMATION AND ASSEMBLY OF MATRICES
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 8888 REWIND 1
      REWIND 2
      REWIND 4
      DO 35 NPT=1,NPART
      NA=NSTART(NPT)
```

```
      NB=NEND(NPT)
      MA=NSUM(NFREE,NA-1)
      MB=NSUM(NFREE,NB)
      MI=MA+1
      M1=MB-MA
      IF (NPT.GE.NPART) GO TO 39
      NC=NEND(NPT+1)
      MC=NSUM(NFREE,NC)
      M2=MC-MB
      GO TO 5
 39   M2=M1
 5    DO 37 I=1,M1
      DO 38 J=1,M1
 38   A(I,J)=0.
      DO 37 J=1,M2
      C(I,J)=0.
 37   CONTINUE
      LA=NFIRST(NPT)
      LB=NLAST(NPT)
      DO 36 NM=LA,LB
      P=SQRT((X(NM,2)-X(NM,1))**2+(Y(NM,2)-Y(NM,1))**2+(Z(NM,2)-Z(NM,1))
     1**2)
      Q=SQRT((X(NM,3)-X(NM,2))**2+(Y(NM,3)-Y(NM,2))**2+(Z(NM,3)-Z(NM,2))
     1**2)
      R=SQRT((X(NM,1)-X(NM,3))**2+(Y(NM,1)-Y(NM,3))**2+(Z(NM,1)-Z(NM,3))
     1**2)
      S=(P**2+R**2-Q**2)/(2.0*P)
      Y2=P
      Y3=S
      X3=SQRT(R**2-S**2)
       X1=0.0
       X2=0.0
       Y1=0.0
       CALL LAMDA (NM)
      T=THICK(NM)
      CALL PLAN(X1,X2,X3,Y1,Y2,Y3,EM,V,PK,T)
      T=THICK(NM)
      CALL BEND(X1,X2,X3,Y1,Y2,Y3,EM,V,T,BK)
      CALL GROUP(ST,PK,BK)
       CALL TRANF(NM,ST)
      CALL ASSEBL(NM,A,C,ST,NI,NPT,NPART)
 36   CONTINUE
      WRITE(2)M1,M2,((A(I,J),I=1,M1),J=1,M1),((C(I,J),I=1,M1),J=1,M2),
     1((PP(I,J),I=MI,MB),J=1,NCOLN)
 35   CONTINUE
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     THE MATRICES ARE FORMED AND WRITTEN IN TAPE 2
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      REWIND 1
      REWIND 2
      REWIND 4
C     CALL SUBROUTINE TO SOLVE THE TRIDIAGONAL EQUATIONS
      CALL RECUR(A,C,NPART,NCOLN)
 1500 CONTINUE
      STOP
      END
```

```
      SUBROUTINE RECUR(NPART,NCOLN)
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     SUBROUTINE FOR SOLUTION OF EQUATIONS. CALCULATION AND PRINTING
C     OF RESIDUALS
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      DIMENSION A(72,72),C(72,72),BB(72,72),RS(72,2),F(72,2),TF(72,2)
       DIMENSION X(166,3),Y(166,3)
       DIMENSION PP(688,2)
      DIMENSION NT(72),NI(166,3),RTF(9,9)
      DIMENSION DIS(72,2),NSTART(15),NEND(15),NFREE(100),TNF(166,3,3)
      COMMON NOD,NELEM,EM,V,NSTART,NEND,NFREE,TNF,PP
      COMMON A,C,NI,RTF,X,Y
      EQUIVALENCE(F(1,1),PP(1,1)),(DIS(1,1),PP(1,2))
      DO 140 I=1,72
      DO 141 J=1,NCOLN
      TF(I,J)=0.
  141 RS(I,J)=0.
      DO 140 J=1,72
  140 BB(I,J)=0.
      DO 40 NPT=1,NPART
      READ(2) M1,M2,((A(I,J),I=1,M1),J=1,M1),((C(I,J),I=1,M1),J=1,M2),
     1((F(I,J),I=1,M1),J=1,NCOLN)
  150 DO 44 I=1,M1
      DO 45 J=1,NCOLN
      F(I,J)=F(I,J)-TF(I,J)
   45 CONTINUE
      DO 44 J=1,M1
   44 A(I,J)=A(I,J)-BB(I,J)
      CALL INVMAT(A,72,M1,1.0E-8,IERR,NT)
      IF(IERR.NE.0) WRITE(6,385) IERR,NPT
  385 FORMAT(5X,5HIERR=,I5,5X,4HNPT=,I5,///)
      WRITE(4) M1,M2,((A(I,J),I=1,M1),J=1,M1),((C(I,J),I=1,M1),J=1,M2),
     1((F(I,J),I=1,M1),J=1,NCOLN)
      IF(NPT.EQ.NPART) GO TO 50
      DO 46 I=1,M1
      DO 46 J=1,NCOLN
      DIS(I,J)=0.
      DO 46 K=1,M1
   46 DIS(I,J)=DIS(I,J)+A(I,K)*F(K,J)
      DO 47 I=1,M2
      DO 47 J=1,NCOLN
      TF(I,J)=0.
      DO 47 K=1,M1
   47 TF(I,J)=TF(I,J)+C(K,I)*DIS(K,J)
      DO 48 I=1,M1
      DO 48 J=1,M2
      BB(I,J)=0.
      DO 48 K=1,M1
   48 BB(I,J)=BB(I,J)+A(I,K)*C(K,J)
      DO 49 I=1,M2
      DO 49 J=1,M2
      A(I,J)=0.
      DO 49 K=1,M1
```

```
49    A(I,J)=A(I,J)+C(K,I)*BB(K,J)
      DO 51 I=1,M2
      DO 51 J=1,M2
51    BB(I,J)=A(I,J)
40    CONTINUE
50    REWIND 2
      DO 55 I=1,M1
      DO 55   J=1,NCOLN
      DIS(I,J)=0.
      DO 55 K=1,M1
55    DIS(I,J)=DIS(I,J)+A(I,K)*F(K,J)
      WRITE(6,325) NPART
325   FORMAT(10X,13HPARTITION NO=,I6,//,5X,19HTHE DEFLECTIONS ARE,//)
      MA=NSTART(NPART)
      MB=NEND(NPART)
      DO 320 J=1,NCOLN
      WRITE(6,322) J
      K=1
      DO 320 I=MA,MB
      KK=NFREE(I)
      KK=KK+K-1
      WRITE(6,321) I, (DIS(II,J),II=K,KK)
462   K=KK+1
320   CONTINUE
 321  FORMAT(I3,9E14.6)
322   FORMAT(/,5X,16HLOAD VECTOR NO.=,I6,/)
      WRITE(1) ((DIS(I,J),I=1,M1),J=1,NCOLN)
      NA=NPART-1
      DO 350 LL=1,NA
      BACKSPACE 4
      BACKSPACE 4
      READ(4) M1,M2,((A(I,J),I=1,M1),J=1,M1),((C(I,J),I=1,M1),J=1,M2),
     1((F(I,J),I=1,M1),J=1,NCOLN)
      DO 110 I=1,M1
      DO 110 J=1,NCOLN
      TF(I,J)=0.
      DO 110 K=1,M2
110   TF(I,J)=TF(I,J)+C(I,K)*DIS(K,J)
      DO 144 I=1,M1
      DO 144 J=1,NCOLN
144   F(I,J)=F(I,J)-TF(I,J)
      DO 60 I=1,M1
      DO 60 J=1,NCOLN
      DIS(I,J)=0.
      DO 60 K=1,M1
60    DIS(I,J)=DIS(I,J)+A(I,K)*F(K,J)
      NPT=NPART-LL
      WRITE(6,325) NPT
       IF(NPT.EQ.1) GO TO 500
      MA=NSTART(NPT)
      MB=NEND(NPT)
500    DO 326 J=1,NCOLN
      WRITE(6,322) J
       IF(NPT.NE.1) GO TO 501
       WRITE(6,502) (DIS(II,J),II=1,4)
       GO TO 326
 501   K=1
      DO 326 I=MA,MB
```

```
      IF(NPT.EQ.1) GO TO 326
     KK=NFREE(I)
     KK=KK+K-1
     WRITE(6,321) I,(DIS(II,J),II=K,KK)
 442 K=KK+1
502   FORMAT(5X,6E15.5)
326  CONTINUE
350  WRITE(1) ((DIS(I,J),I=1,M1),J=1,NCOLN)
     WRITE(6,115)
115  FORMAT(1H1,17HTHE RESIDUALS ARE,//)
     DO 200 NPT=1,NPART
     READ(2)M1,M2,((A(I,J),I=1,M1),J=1,M1),((C(I,J),I=1,M1),J=1,M2) ,
    1((F(I,J),I=1,M1),J=1,NCOLN)
     BACKSPACE 1
     READ(1) ((DIS(I,J),I=1,M1),J=1,NCOLN)
     BACKSPACE 1
     BACKSPACE 1
     READ(1)((TF(I,J),I=1,M2),J=1,NCOLN)
     DO 250 I=1,M1
     DO 250 J=1,NCOLN
     F(I,J)=F(I,J)-RS(I,J)
     DO 260 K=1,M1
260  F(I,J)=F(I,J)-A(I,K)*DIS(K,J)
     DO 250 L=1,M2
250  F(I,J)=F(I,J)-C(I,L)*TF(L,J)
     DO 265 I=1,M2
     DO 265 J=1,NCOLN
     RS(I,J)=0.
     DO 265 K=1,M1
265  RS(I,J)=RS(I,J)+C(K,I)*DIS(K,J)
     WRITE(6,300) ((F(I,J),I=1,M1),J=1,NCOLN)
300  FORMAT(8E15.3)
200  CONTINUE
     RETURN
     END



      SUBROUTINE TRANF2(NM,TL)
C    TRANSFORMATION OF ELEMENT STIFFNESS MATRIX FROM PLATE COORDINATE
C    TO GLOBAL COORDINATE SYSTEMS.
      DIMENSION TL(9,9),RTF(9,9)
     DIMENSION A(72,72),C(72,72),NI(166,3)
     DIMENSION PP(688,2),NSTART(15),NEND(15),NFREE(100),TNF(166,3,3)
      DIMENSION X(166,3),Y(166,3)
     COMMON NOD,NELEM,EM,V,NSTART,NEND,NFREE,TNF,PP
     COMMON A,C,NI,RTF,X,Y
      DO 11 I=1,9
      DO 11 J=1,9
      TL(I,J)=0.0
 11   CONTINUE
      DO 12 I=1,3
      DO 12 J=1,3
      TL(I,J)=TNF(NM,I,J)
      IA=I+6
      JA=J+6
```

```
      TL(IA,JA)=TNF(NM,I,J)
   12 CONTINUE
      TL(4,4)=TNF(NM,1,1)**2*TNF(NM,3,3)
      TL(4,5)=2.0*TNF(NM,1,1)*TNF(NM,1,2)*TNF(NM,3,3)
      TL(4,6)=TNF(NM,3,3)*TNF(NM,1,2)**2
      TL(5,4)=TNF(NM,1,1)*TNF(NM,2,1)*TNF(NM,3,3)
      TL(5,5)=(TNF(NM,1,2)*TNF(NM,2,1)+TNF(NM,2,2)*TNF(NM,1,1))*TNF(NM,
     1 3,3)
      TL(5,6)=TNF(NM,3,3)*TNF(NM,1,2)*TNF(NM,2,2)
      TL(6,4)=TNF(NM,3,3)*TNF(NM,2,1)**2
      TL(6,5)=2.0*TNF(NM,3,3)*TNF(NM,2,1)*TNF(NM,2,2)
      TL(6,6)=TNF(NM,3,3)*TNF(NM,2,2)**2
      RETURN
      END




      SUBROUTINE TRANF1(ALPHA,ST)
C     TRANSFORMATION OF ELEMENT STIFFNESS MATRIX FROM ELEMENT COORDINATE
C     TO PLATE COORDINATE SYSTEMS.
      DIMENSION TED(27,27),ST(27,27),TST(27,27),RTF(9,9)
      DIMENSION X(166,3),Y(166,3)
     DIMENSION PP(688,2),NSTART(15),NEND(15),NFREE(100),TNF(166,3,3)
     DIMENSION A(72,72),C(72,72),NI(166,3)
     COMMON NOD,NELEM,EM,V,NSTART,NEND,NFREE,TNF,PP
     COMMON A,C,NI,RTF,X,Y
      DO 1I=1,27
      DO 1J=1,27
    1 TED(I,J)=0.0
      CALL ROTMAT(ALPHA)
      DO 2 I=1,9
      DO 2 J=1,9
      DO 2 K=1,3
      N=(K-1)*9
      TED(I+N,J+N)=RTF(I,J)
    2 CONTINUE
      CALL PRTMAT(TED,ST,TST,27,27,27)
      CALL PRDMAT(TST,TED,ST,27,27,27)
      RETURN
      END




      SUBROUTINE ROTMAT(THITA)
C     THE ROTATORY MATRIX FOR ROTATION OF ELEMENT.
     DIMENSION RTF(9,9)
      DIMENSION X(166,3),Y(166,3)
      DIMENSION PP(688,2)
     DIMENSION NSTART(15),NEND(15),NFREE(100),TNF(166,3,3)
      DIMENSION A(72,72),C(72,72),NI(166,3)
     COMMON NOD,NELEM,EM,V,NSTART,NEND,NFREE,TNF,PP
     COMMON A,C,NI,RTF,X,Y
      DO 1 I=1,9
     DO 1 J=1,9
```

```
1     RTF(I,J)=0.0
       IF(ABS(THITA).LT.1.0E-6) GO TO 11
      RTF(1,1)=COS(THITA)
      RTF(1,2)=SIN(THITA)
      RTF(2,1)=-SIN(THITA)
      RTF(2,2)=COS(THITA)
      RTF(3,3)=1.0
       RTF(4,4)=COS(THITA)**2
       RTF(4,5)=2.0*COS(THITA)*SIN(THITA)
       RTF(4,6)=SIN(THITA)**2
       RTF(5,4)=-SIN(THITA)*COS(THITA)
       RTF(5,5)=COS(THITA)**2-SIN(THITA)**2
       RTF(5,6)=-RTF(5,4)
       RTF(6,4)=RTF(4,6)
       RTF(6,5)=-RTF(4,5)
       RTF(6,6)=RTF(4,4)
       DO 2 I=1,3
       DO 2 J=1,3
  2    RTF(I+6,J+6)=RTF(I,J)
      RETURN
 11    DO 10 I=1,9
       DO 10 J=1,9
      RTF(I,J)=0.0
      IF(I.EQ.J) RTF(I,J)=1.0
 10    CONTINUE
       RETURN
      END


      SUBROUTINE GROUP(STIFF,PSTIFF,BSTIFF)
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     SUBROUTINE TO COBINE ELEMENT STIFFNESS MATRICES IN PLANE FORCES
C     AND IN BENDING TO GET 27,27 STIFFNESS MATRIX FOR THE
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
       DIMENSION STIFF(27,27),PSTIFF(6,6),BSTIFF(18,18)
C     ASSEMBLE THE ELEMENT STIFFNESS MATRIX
C     CLEAR ARRAY
      DO 134 I=1,27
      DO 134 J=1,27
 134  STIFF(I,J)=0.0
      DO 60 KI=1,3
      DO 60 KJ=1,3
      DO 60 I=1,2
      DO 60 J=1,2
      II=I+(KI-1)*9
      JJ=J+(KJ-1)*9
      III=I+(KI-1)*2
      JJJ=J+(KJ-1)*2
 60   STIFF(II,JJ)=PSTIFF(III,JJJ)
      DO 65 KI=1,3
      DO 65 KJ=1,3
       DO 65 I=1,6
       DO 65 J=1,6
      II=2+I+(KI-1)*9
      JJ=2+J+(KJ-1)*9
```

```
      III=I+(KI-1)*6
      JJJ=J+(KJ-1)*6
     STIFF(II,JJ)=BSTIFF(III,JJJ)
  65  CONTINUE
     RETURN
     END
```

```
      SUBROUTINE BEND(A,B,C,M,N,V,E,T,BG)
C     CALCULATION OF BENDING STIFFNESS MATRIX OF EACH ELEMENT,
C     IN TERMS OF SIDES ,POWER OF X AND Y VARIBLES.
      DIMENSION BIG(20,20),M(20),N(20),TM(20,20),N1(20),TBIG(18,20)
      DIMENSION TN(20,18),AB(2,18),BG(18,18)
      DIMENSION BA(18,2)
      DO 299 I=1,20
      DO 299 J=1,20
 299  BIG(I,J)=0.0
     DO 300 I=1,20
     DO 300 J=1,20
     I1=M(I)
     J1=M(J)
     I2=N(I)
     J2=N(J)
      BIG(I,J)=I1*J1*(I1-1)*(J1-1)*G((I1+J1-4),(I2+J2),A,B,C)+I2*J2*(I2
    1-1)*(J2-1)*G((I1+J1),(I2+J2-4),A,B,C)+(2.0*(1.-V)*I1*J1*I2*J2+V*I1
    2*J2*(I1-1)*(J2-1)+V*J1*I2*(J1-1)*(I2-1))*G((I1+J1-2),(I2+J2-2),A,B
    3,C)
 300 CONTINUE
     FM=(E*T**3)/(12.0*(1.-V**2))
     DO 288 I=1,20
     DO 288 J=1,20
 288 BIG(I,J)=FM*BIG(I,J)
     CALL TMAT(A,B,C,TM)
     CALL INVMAT(TM,20,20,1.0E-12,IERR,N1)
     IF(IERR.NE.0) GO TO 21
      DO 19 I=1,20
      DO 19 J=1,18
      TN(I,J)=TM(I,J)
 19   CONTINUE
      CALL PRTMAT(TN,BIG,TBIG,18,20,20)
      CALL PRDMAT(TBIG,TN,BG,18,20,18)
      DO 306 K=1,3
      L=(K-1)*6
      DO 30 I=1,18
      DO 30 J=1,2
  30  BA(I,J)=BG(I,J+L+1)
      DO 31 I=1,18
      DO 31 J=4,6
  31  BG(I,J+L-2)=BG(I,J+L)
      DO 32 I=1,18
      DO 32 J=1,2
  32  BG(I,J+L+4)=BA(I,J)
 306  CONTINUE
      DO 305 K=1,3
      L=(K-1)*6
```

```
      DO 28 I=1,2
      DO 28 J=1,18
   28 AB(I,J)=BG(I+L+1,J)
      DO 10 I=4,6
      DO 10 J=1,18
   10 BG(I+L-2,J)=BG(I+L,J)
      DO 11 I=1,2
      DO 11 J=1,18
   11 BG(I+L+4,J)=AB(I,J)
  305 CONTINUE
      RETURN
   21 WRITE(6,22) IERR
   22 FORMAT(/,2X,*IERR=*,I5,/)
   25 RETURN
      END
```

```
      SUBROUTINE ASSEBL(NM,ST,NPT,NPART)
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     SUBROUTINE FOR THE ASSEMBLY OF ELEMENT STIFFNESS MATRICES
C     PARTITION BY PARTITION
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      DIMENSION ST(27,27)
      DIMENSION SX(9,9),TL(9,9),SXT(9,9),RTF(9,9)
      DIMENSION X(166,3),Y(166,3)
      DIMENSION NSTART(15),NEND(15),NFREE(100),TNF(166,3,3)
      DIMENSION A(72,72),C(72,72),NI(166,3)
      DIMENSION PP(688,2)
      COMMON NOD,NELEM,EM,V,NSTART,NEND,NFREE,TNF,PP
      COMMON A,C,NI,RTF,X,Y
      NA=NSTART(NPT)
      NB=NEND(NPT)
      IF(NPT.GE.NPART) GO TO 50
      ND=NEND(NPT+1)
      GO TO 60
   50 ND=NB
   60 MINUS=NSUM(NFREE,NA-1,NPT)
      MIN=NSUM(NFREE,NB,NPT)
      DO 100 I=1,3
      LL=NI(NM,I)
      IF(LL.EQ.0) GO TO 100
      IF(LL.LT.NA) GO TO 100
      IF(LL.GT.NB) GO TO 100
      DO 110 J=1,3
      MM=NI(NM,J)
      IF(MM.EQ.0) GO TO 110
      IF(MM.GT.ND) GO TO 110
      IF(MM.LT.NA) GO TO 110
      NF1=NFREE(LL)
      NF2=NFREE(MM)
      DO 10 II=1,NF1
      DO 10 JJ=1,NF2
      IN=II+(I-1)*9
      JN=JJ+(J-1)*9
   10 SX(II,JJ)=ST(IN,JN)
      CALL TRANF2(NM,TL)
```

```
 51   IL=NSUM(NFREE,LL-1,NPT)
      JL=NSUM(NFREE,MM-1,NPT)
      IF(NF1.EQ.9.AND.NF2.EQ.9) NC=1
      IF(NF1.EQ.8.AND.NF2.EQ.9) NC=2
      IF(NF1.EQ.9.AND.NF2.EQ.8) NC=3
      IF(NF1.EQ.8.AND.NF2.EQ.8) NC=4
      GO TO (20,21,22,24),NC
 20   CALL PRTMAT(TL,SX,SXT,9,9,9)
       CALL PRDMAT(SXT,TL,SX,9,9,9)
      GO TO 24
 21   DO 27 II=1,8
      DO 27 JJ=1,9
      SXT(II,JJ)=0.
      DO 27 KK=1,9
      SXT(II,JJ)=SXT(II,JJ)+SX(II,KK)*TL(KK,JJ)
 27   CONTINUE
      DO 28 II=1,8
      DO 28 JJ=1,9
      SX(II,JJ)=SXT(II,JJ)
 28   CONTINUE
      GO TO 24
 22   DO 29 II=1,9
      DO 29 JJ=1,8
      SXT(II,JJ)=0.
      DO 29 KK=1,9
      SXT(II,JJ)=SXT(II,JJ)+TL(KK,II)*SX(KK,JJ)
 29   CONTINUE
      DO 30 II=1,9
      DO 30 JJ=1,8
      SX(II,JJ)=SXT(II,JJ)
 30   CONTINUE
 24   DO 25 II=1,NF1
      DO 25 JJ=1,NF2
      ILL=IL+II
      JLL=JL+JJ
      ILM=ILL-MINUS
      JLM=JLL-MINUS
      JLN=JLL-MIN
      IF(JLL.GT.MIN) GO TO 26
      A(ILM,JLM)=A(ILM,JLM)+SX(II,JJ)
      GO TO 25
 26   C(ILM,JLN)=C(ILM,JLN)+SX(II,JJ)
 25   CONTINUE
 110  CONTINUE
 100  CONTINUE
      RETURN
      END


      FUNCTION NSUM (NFREE,NN,NPARD)
      DIMENSION NFREE(100)
      NSUM=0
      IF(NPARD.NE.1) GO TO 20
      IF(NN.EQ.0) RETURN
      DO 10 I=1,NN
```

```
      NSUM=NSUM+NFREE(I)
   10 CONTINUE
      RETURN
   20 DO 30 I=1,NN
      NSUM=NSUM+NFREE(I)
   30 CONTINUE
      NSUM=NSUM-64
      RETURN
      END
```

```
      SUBROUTINE ELIM (M1,M2)
     DIMENSION A(72,72),C(72,72),NI(166,3),RTF(9,9)
      DIMENSION X(166,3),Y(166,3)
     DIMENSION PP(688,2),NSTART(15),NEND(15),NFREE(100),TNF(166,3,3)
     COMMON NOD,NELEM,EM,V,NSTART,NEND,NFREE,TNF,PP
     COMMON A,C,NI,RTF,X,Y
C    THIS SUBROUTINE ELIMINATES ALL DEGREES OF FREEDOM EXCEPT
C    FOUR CURVATURES AT THE BASE.
      DO 1 I=1,14
      DO 2 J=1,M1
    2 A(I,J)=0.0
      DO 1 J=1,M2
      C(I,J)=0.0
    1 CONTINUE
      A(1,1)=A(15,15)
      A(1,2)=A(15,30)
      A(1,3)=A(15,49)
      A(1,4)=A(15,64)
      DO 3 J=1,M2
    3 C(1,J)=C(15,J)
      DO 4 I=15,29
      DO 5 J=1,M1
    5 A(I,J)=0.0
      DO 4 J=1,M2
      C(I,J)=0.0
    4 CONTINUE
      A(2,1)=A(30,15)
      A(2,2)=A(30,30)
      A(2,3)=A(30,49)
      A(2,4)=A(30,64)
      DO 6 J=1,M2
    6 C(2,J)=C(30,J)
      DO 7 I=30,48
      DO 8 J=1,M1
    8 A(I,J)=0.0
      DO 7 J=1,M2
      C(I,J)=0.0
    7 CONTINUE
      A(3,1)=A(49,15)
      A(3,2)=A(49,30)
      A(3,3)=A(49,49)
      A(3,4)=A(49,64)
      DO 9 J=1,M2
    9 C(3,J)=C(49,J)
```

```
      DO 10 I=49,63
      DO 11 J=1,M1
11    A(I,J)=0.0
      DO 10 J=1,M2
      C(I,J)=0.0
10    CONTINUE
      A(4,1)=A(64,15)
      A(4,2)=A(64,30)
      A(4,3)=A(64,49)
      A(4,4)=A(64,64)
      DO 12 J=1,M2
12    C(4,J)=C(64,J)
      DO 13 I=64,M1
      DO 14 J=1,M1
14    A(I,J)=0.0
      DO 13 J=1,M2
      C(I,J)=0.0
13    CONTINUE
      RETURN
      END


      SUBROUTINE PLAN(X1,X2,X3,Y1,Y2,Y3,EM,V,PK,THICK)
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     SUBROUTINE FOR CALCULATING ELEMENT STIFFNESS MATRIX IN PLANE FORCES
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      DIMENSION PK(6,6)
      AREA=0.5*(X2*Y3-Y2*X3-X1*Y3+Y1*X3+X1*Y2-Y1*X2)
      AREA=ABS(AREA)
      C=EM*THICK/(4.*AREA*(1.0-V**2))
      D=EM*THICK/(8.0*AREA*(1.0+V))
      Y32=Y3-Y2
      X32=X3-X2
      Y31=Y3-Y1
      X31=X3-X1
      Y21=Y2-Y1
      X21=X2-X1
      PK(1,1)=C*Y32**2+D*X32**2
      PK(2,1)=-C*V*Y32*X32-D*X32*Y32
      PK(3,1)=-C*Y32*Y31-D*X32*X31
      PK(4,1)=C*V*Y32*X31+D*X32*Y31
      PK(5,1)=C*Y32*Y21+D*X32*X21
      PK(6,1)=-C*V*Y32*X21-D*X32*Y21
      PK(2,2)=C*X32**2+D*Y32**2
      PK(3,2)=C*V*X32*Y31+D*Y32*X31
      PK(4,2)=-C*X32*X31-D*Y32*Y31
      PK(5,2)=-C*V*X32*Y21-D*Y32*X21
      PK(6,2)=C*X32*X21+D*Y32*Y21
      PK(3,3)=Y31**2*C+D*X31**2
      PK(4,3)=-C*V*Y31*X31-D*X31*Y31
      PK(5,3)=-C*Y31*Y21-D*X31*X21
      PK(6,3)=C*V*Y31*X21+D*X31*Y21
      PK(4,4)=C*X31**2+D*Y31**2
      PK(5,4)=C*V*X31*Y21+D*Y31*X21
      PK(6,4)=-C*X31*X21-D*Y31*Y21
```

```
      PK(5,5)=C*Y21**2+D*X21**2
      PK(6,5)=-C*V*Y21*X21-D*X21*Y21
      PK(6,6)=C*X21**2+D*Y21**2
      DO 10 I=1,5
      IP=I+1
      DO 10 J=IP,6
 10   PK(I,J)=PK(J,I)
      RETURN
      END
```

```
      SUBROUTINE PRDMAT(A,B,C,M,N,L)
C     SUBROUTINE TWO MULTIPLY TWO MATRICES C=A*B
      DIMENSION A(M,N),B(N,L),C(M,L)
      DO 10 I=1,M
      DO 10 J=1,L
      C(I,J)=0.
      DO 10 K=1,N
      C(I,J)=C(I,J)+A(I,K)*B(K,J)
 10   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PRTMAT(A,B,C,M,N,L)
C     SUBROUTINE TO MULTIPY TWO MATRICES C=(TRANSPOSE A)*B
      DIMENSION A(N,M),B(N,L),C(M,L)
      DO 10 I=1,M
      DO 10 J=1,L
      C(I,J)=0.
      DO 10 K=1,N
      C(I,J)=C(I,J)+A(K,I)*B(K,J)
 10   CONTINUE
      RETURN
      END
```

```
      FUNCTION G(M,N,A,B,C)
C    EVALUATING THE FUNCTION VALUE OF THE INTEGRAL.
       G=FACT(M+N+2)
       IF(G.EQ.0.0) RETURN
       IF(ABS(B).LT.1.0E-6)GO TO    1
       IF(ABS(A).LT.1.0E-6)GO TO 2
       G1=G
       G=C**(N+1)*(A**(M+1)-(-B)**(M+1))*FACT(M)*FACT(N)/G1
       RETURN
   1   G1=G
       G=C**(N+1)*A**(M+1)*FACT(M)*FACT(N)/G1
       RETURN
   2   G1=G
```

```
      G=-C**(N+1)*(-B)**(M+1)*FACT(M)*FACT(N)/G1
      RETURN
      END


      FUNCTION FACT(N)
C     EVALUATING THE FACTORIAL OF AN INTEGER.
      IF(N.LT.1) GO TO 601
      M=1
      DO 600 I=1,N
      M=M*I
  600 CONTINUE
      FACT=M
      RETURN
  601 IF(N.EQ.0) FACT=1.0
      IF(N.LT.0) FACT=0.0
      RETURN
      END


      SUBROUTINE TMAT(A,B,C,TM)
      DIMENSION TM(20,20)
      DO 1 I=1,20
      DO 1 J=1,20
      TM(I,J)=0.0
    1 CONTINUE
      TM(1,1)=1.0
      TM(1,2)=-B
      TM(1,4)=B**2
      TM(1,7)=-B**3
      TM(1,11)=B**4
      TM(1,16)=-B**5
      TM(2,2)=1.0
      TM(2,4)=-2.0*B
      TM(2,7)=3.0*B**2
      TM(2,11)=-4.0*B**3
      TM(2,16)=5.0*B**4
      TM(3,3)=1.0
      TM(3,5)=-B
      TM(3,8)=B**2
      TM(3,12)=-B**3
      TM(4,4)=2.0
      TM(4,7)=-6.0*B
      TM(4,11)=12.0*B**2
      TM(4,16)=-20.0*B**3
      TM(5,5)=1.0
      TM(5,8)=-2.0*B
      TM(5,12)=3.0*B**2
      TM(6,6)=2.0
      TM(6,9)=-2.0*B
      TM(6,13)=2.0*B**2
      TM(6,17)=-2.0*B**3
```

```
TM(7,1)=1.0
TM(7,2)=A
TM(7,4)=A**2
TM(7,7)=A**3
TM(7,11)=A**4
TM(7,16)=A**5
TM(8,2)=1.0
TM(8,4)=2.0*A
TM(8,7)=3.0*A**2
TM(8,11)=4.0*A**3
TM(8,16)=5.0*A**4
TM(9,3)=1.0
TM(9,5)=A
TM(9,8)=A**2
TM(9,12)=A**3
TM(10,4)=2.0
TM(10,7)=6.0*A
TM(10,11)=12.0*A**2
TM(10,16)=20.0*A**3
TM(11,5)=1.0
TM(11,8)=2.0*A
TM(11,12)=3.0*A**2
TM(12,6)=2.0
TM(12,9)=2.0*A
TM(12,13)=2.0*A**2
TM(12,17)=2.0*A**3
TM(13,1)=1.0
TM(13,3)=C
TM(13,6)=C**2
TM(13,10)=C**3
TM(13,15)=C**4
TM(13,20)=C**5
TM(14,2)=1.0
TM(14,5)=C
TM(14,9)=C**2
TM(14,14)=C**3
TM(14,19)=C**4
TM(15,3)=1.0
TM(15,6)=2.0*C
TM(15,10)=3.0*C**2
TM(15,15)=4.0*C**3
TM(15,20)=5.0*C**4
TM(16,4)=2.0
TM(16,8)=2.0*C
TM(16,13)=2.0*C**2
TM(16,18)=2.0*C**3
TM(17,5)=1.0
TM(17,9)=2.0*C
TM(17,14)=3.0*C**2
TM(17,19)=4.0*C**3
TM(18,6)=2.0
TM(18,10)=6.0*C
TM(18,15)=12.0*C**2
TM(18,20)=20.0*C**3
TM(19,16)=5.0*A**4*C
TM(19,17)=3.0*A**2*C**3-2.0*A**4*C
TM(19,18)=-2.0*A*C**4+3.0*A**3*C**2
TM(19,19)=C**5-4.0*A**2*C**3
```

```
      TM(19,20)=5.0*A*C**4
      TM(20,16)=5.0*B**4*C
      TM(20,17)=3.0*B**2*C**3-2.0*B**4*C
      TM(20,18)=2.0*B*C**4-3.0*B**3*C**2
      TM(20,19)=C**5-4.*B**2*C**3
      TM(20,20)=-5.0*B*C**4
      RETURN
      END
```



```
       SUBROUTINE FORT(AA,BB,ATR,BTR,CTR,DTR,H,NST)
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     SUBROUTINE FOR CALCULATING THE ELEMENT NODE COORDINATES
C     OF EACH PARTITION IN TERMS OF THE PARTITION HEIGHT AND THE FACE
C     INCLINATIONS
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
       DIMENSION X(166,3),Y(166,3)
       DIMENSION A(72,72),C(72,72),NI(166,3),RTF(9,9)
       DIMENSION NSTART(15),NEND(15),NFREE(100),TNF(166,3,3),PP(688,2)
      COMMON NOD,NELEM,EM,V,NSTART,NEND,NFREE,TNF,PP
      COMMON A,C,NI,RTF,X,Y
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
       N=NST
       NN=N+15
      DO 10 I=N,NN
      DO 10 J=1,3
      X(I,J)=0.
 10   Y(I,J)=0.
      AB=AA-H*COS(DTR)/SIN(DTR)-H*COS(BTR)/SIN(BTR)
      AC=BB-H*COS(ATR)/SIN(ATR)-H*COS(CTR)/SIN(CTR)
C    NEW COORDINATE SYSTEM SUITABLE FOR COWPER POLYNOMIAL.
       X(N,1)=AA/2.0
       X(N,2)=H*COS(DTR)/SIN(DTR)
       Y(N,2)=H/SIN(ATR)
       X(N+1,1)=X(N,2)
       X(N+1,2)=X(N,1)
       X(N+1,3)=X(N+1,1)+AB/2.0
       Y(N+1,1)=Y(N,2)
       Y(N+1,3)=Y(N,2)
       X(N+2,2)=AA/2.0-H*COS(BTR)/SIN(BTR)
       X(N+2,3)=X(N,2)+AB/2.0-AA/2.0
       Y(N+2,2)=Y(N,2)
       Y(N+2,3)=Y(N,2)
       X(N+3,1)=X(N+2,2)
       X(N+3,3)=AA/2.0
       Y(N+3,1)=Y(N,2)
       X(N+4,1)=BB/2.0
       X(N+4,2)=H*COS(ATR)/SIN(ATR)
       Y(N+4,2)=H/SIN(BTR)
       X(N+5,1) =X(N+4,2)
       X(N+5,2)=X(N+4,1)
       X(N+5,3)=X(N+5,1)+AC/2.0
       Y(N+5,1)=Y(N+4,2)
       Y(N+5,3)=Y(N+4,2)
```

```
      X(N+6,1)=BB/2.0
      X(N+6,2)=X(N+5,1)+AC/2.0-BB/2.0
      Y(N+6,2)=Y(N+4,2)
      X(N+7,1)=X(N+6,2)
      X(N+7,2)=X(N+6,1)
      X(N+7,3)=X(N+7,1)+AC/2.0
      Y(N+7,1)=Y(N+4,2)
      Y(N+7,3)=Y(N+4,2)
      X(N+8,1)=AA/2.0
      X(N+8,2)=H*COS(BTR)/SIN(BTR)
      Y(N+8,2)=H/SIN(CTR)
      X(N+9,1)=X(N+8,2)
      X(N+9,2)=X(N+8,1)
      X(N+9,3)=X(N+9,1)+AB/2.0
      Y(N+9,1)=Y(N+8,2)
      Y(N+9,3)=Y(N+8,2)
      X(N+10,2)=AA/2.0-H*COS(DTR)/SIN(DTR)
     X(N+10,3)=X(N+9,1)+AB/2.-AA/2.
      Y(N+10,2)=Y(N+8,2)
      Y(N+10,3)=Y(N+8,2)
      X(N+11,1)=X(N+10,2)
      X(N+11,3)=AA/2.0
      Y(N+11,1)=Y(N+8,2)
      X(N+12,2)=H*COS(CTR)/SIN(CTR)+AC/2.0
      X(N+12,3)=H*COS(CTR)/SIN(CTR)
      Y(N+12,2)=H/SIN(DTR)
      Y(N+12,3)=Y(N+12,2)
      X(N+13,1)=X(N+12,2)
      X(N+13,3)=BB/2.0
      Y(N+13,1)=Y(N+12,2)
      X(N+14,2)=BB/2.0-H*COS(ATR)/SIN(ATR)
      X(N+14,3)=X(N+13,1)-BB/2.0
      Y(N+14,2)=Y(N+12,2)
      Y(N+14,3)=Y(N+12,2)
      X(N+15,1)=X(N+14,2)
      X(N+15,3)=BB/2.0
      Y(N+15,1)=Y(N+12,2)
      RETURN
      END
```

```
      SUBROUTINE CNC (CC,X3,Y2,Y3)
      DIMENSION CC(9,9)
      DIMENSION XX(3),YY(3)
C     SUBROUTINE FOR CALCULATING C MATRIX.
      XX(1)=0.0
      YY(1)=0.0
      XX(2)=0.0
      YY(2)=Y2
      XX(3)=X3
```

```
      YY(3)=Y3
      DO 10 I=1,9
      DO 10 J=1,9
   10 CC(I,J)=0.0
      DO 15 I=1,3
      J=1+(I-1)*3
      CC(J,1)=1.0
      CC(J,2)=XX(I)
      CC(J,3)=YY(I)
      CC(J,4)=XX(I)**2
      CC(J,5)=XX(I)*YY(I)
      CC(J,6)=YY(I)**2
      CC(J,7)=XX(I)**3
      CC(J,8)=XX(I)**2*YY(I)
      CC(J,9)=YY(I)**3
      K=2+(I-1)*3
      CC(K,3)=1.0
      CC(K,5)=XX(I)
      CC(K,6)=2.0*YY(I)
      CC(K,8)=XX(I)**2
      CC(K,9)=3.0*YY(I)**2
      L=3+(I-1)*3
      CC(L,2)=-1.0
      CC(L,4)=-2.0*XX(I)
      CC(L,5)=-YY(I)
      CC(L,7)=-3.0*XX(I)**2
      CC(L,8)=-2.0*XX(I)*YY(I)
   15 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE LAMDA(NM)
      DIMENSION NEND(15),NSTART(15),PP(500,2)
      DIMENSION X(200,3),Y(200,3),Z(200,3),NFREE(100),TNF(200,3,3)
      DIMENSION ALMD(3,3)
      COMMON NOD,NELEM,EM,V,NSTART,NEND,NFREE,TNF,PP,X,Y,Z,ALMD
C     THIS SUBROUTINE FINDS THE DIRECTION COSINES OF LOCAL AXES.
C     X,Y,Z-ARE THE GLOBAL COORDINATES.
C     ALMD-MATRIX OF DIRECTION COSINES RETURNED.
      YJI=Y(NM,2)-Y(NM,1)
      XJI=X(NM,2)-X(NM,1)
      ZJI=Z(NM,2)-Z(NM,1)
      YMI=Y(NM,3)-Y(NM,1)
      XMI=X(NM,3)-X(NM,1)
      ZMI=Z(NM,3)-Z(NM,1)
      A=YJI*ZMI-YMI*ZJI
      B=-XJI*ZMI+XMI*ZJI
      C=XJI*YMI-XMI*YJI
      F=SQRT(XJI**2+YJI**2+ZJI**2)
      G=-SQRT(A**2+B**2+C**2)
      ALMD(2,1)=XJI/F
      ALMD(2,2)=YJI/F
      ALMD(2,3)=ZJI/F
      ALMD(3,1)=A/G
```

```
      ALMD(3,2)=B/G
      ALMD(3,3)=C/G
      ALMD(1,1)=ALMD(2,2)*ALMD(3,3)-ALMD(3,2)*ALMD(2,3)
      ALMD(1,2)=ALMD(3,1)*ALMD(2,3)-ALMD(3,3)*ALMD(2,1)
      ALMD(1,3)=ALMD(2,1)*ALMD(3,2)-ALMD(3,1)*ALMD(2,2)
     RETURN
     END




      SUBROUTINE BEND(X3,Y2,Y3,EM,V,THICK,BK)
      DIMENSION BIG(9,9),CC(9,9),DD(9,9),N1(9),BK(9,9)
C
C     SUBROUTINE FOR CALCULATING THE ELEMENT STIFFNESS MATRIX
C     IN BENDING OF PLATE.
C
      C11=1.0/2.0*X3*Y2
      C21=1./6.*X3**2*Y2
      C31=1./12.*X3**3*Y2
      C12=1./6.*X3*Y2*(Y2+Y3)
      C13=1./12.*X3*Y2*(Y2**2+Y2*Y3+Y3**2)
      C22=1./24.*X3**2*Y2*(Y2+2.*Y3)
      DO15 I=1,9
      DO15 J=1,9
   15 BIG(I,J)=0.0
      BIG(4,4)=4.*C11
      BIG(6,4)=4.*V*C11
      BIG(7,4)=12.*C21
      BIG(8,4)=4.*C12
      BIG(9,4)=12.*V*C12
      BIG(5,5)=2.*(1.-V)*C11
      BIG(8,5)=4.*(1.-V)*C21
      BIG(6,6)=4.*C11
      BIG(7,6)=12.*V*C21
      BIG(8,6)=4.*V*C12
      BIG(9,6)=12.*C12
      BIG(7,7)=36.*C31
      BIG(8,7)=12.*C22
      BIG(9,7)=36.*V*C22
      BIG(8,8)=4.*C13+8.*(1.-V)*C31
      BIG(9,8)=12.*V*C13
      BIG(9,9)=36.*C13
C     FILLING THE VALUE THAT REPEAT IN THE MATRIX.
      DO 77 I=1,8
      IP=I+1
      DO 77 J=IP,9
   77 BIG(I,J)=BIG(J,I)
      FM=EM*THICK**3/(12.*(1.-V**2))
      DO 16 I=1,9
      DO 16 J=1,9
   16 BIG(I,J)=FM*BIG(I,J)
      CALL CNC (CC,X3,Y2,Y3)
      CALL INVMAT(CC,9,9,1.0E-12,IERR,N1)
  100 FORMAT(9E12.3)
      IF(IERR.NE.0) GO TO 20
      CALL PRTMAT(CC,BIG,DD,9,9,9)
```

```
      CALL PRDMAT(DD,CC,BK,9,9,9)
C     BK IS THE BENDING STIFFNESS MATRIX.
      GO TO 25
   20 WRITE(6,30) IERR
   30 FORMAT(/,2X,5HIERR=,I5,/)
   25 RETURN
      END
```

```
      SUBROUTINE STRESS(NCOLN,XEL,T,NM)
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     SUBROUTINE FOR CALCULATING THE STRESSES AT THE CENTROID OF
C     EACH TRIANGLE
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      DIMENSION SXX(2),SYY(2),SXY(2),SMAX(2),SMIN(2)
      DIMENSION X(200,3),Y(200,3)
      DIMENSION XP(9,2),XEL(18,2)
      DIMENSION BP(3,6),CP(3,6),DE(3,3),CC(9,9),DB(3,9),DC(3,9),N1(9)
      DIMENSION NSTART(15),NEND(15),NFREE(100),TNF(200,3,3),PP(500,2)
      COMMON NOD,NELEM,EM,V,NSTART,NEND,NFREE,TNF,PP,X,Y
      X1=X(NM,1)
      X2=X(NM,2)
      X3=X(NM,3)
      Y1=Y(NM,1)
      Y2=Y(NM,2)
      Y3=Y(NM,3)
      AREA=0.5*(X2*Y3-Y2*X3-X1*Y3+Y1*X3+X1*Y2-Y1*X2)
      DO 10 I=1,3
      DO 31 J=1,3
   31 DE(I,J)=0.
      DO 32 J=1,6
      BP(I,J)=0.
   32 CP(I,J)=0.
      DO 10 J=1,9
      DB(I,J)=0.
   10 DC(I,J)=0.
      BP(1,1)=Y2-Y3
      BP(1,3)=Y3-Y1
      BP(1,5)=Y1-Y2
      BP(2,2)=X3-X2
      BP(2,4)=X1-X3
      BP(2,6)=X2-X1
      BP(3,1)=X3-X2
      BP(3,2)=Y2-Y3
      BP(3,3)=X1-X3
      BP(3,4)=Y3-Y1
      BP(3,5)=X2-X1
      BP(3,6)=Y1-Y2
      DO 41 I=1,3
      DO 41 J=1,6
      BP(I,J)=BP(I,J)/(2.*AREA)
   41 CONTINUE
      DE(1,1)=1.
      DE(1,2)=V
      DE(2,1)=V
```

```
      DE(2,2)=1.
      DE(3,3)=(1.-V)/2.
      FM=EM/(1.-V**2)
      DO 35 I=1,3
      DO 35 J=1,3
   35 DE(I,J)=FM*DE(I,J)
      CALL PRDMAT(DE,BP,CP,3,3,6)
      DO 11 KI=1,3
      DO 11 II=1,2
      K=II+(KI-1)*6
      III=II+(KI-1)*2
      DO 11 JJ=1,NCOLN
   11 XP(III,JJ)=XEL(K,JJ)
      DO 12 J=1,NCOLN
      SXX(J)=0.
      SYY(J)=0.
      SXY(J)=0.
      DO 12 K=1,6
      SXX(J)=SXX(J)+CP(1,K)*XP(K,J)
      SYY(J)=SYY(J)+CP(2,K)*XP(K,J)
   12 SXY(J)=SXY(J)+CP(3,K)*XP(K,J)
C     STRESSES CORRESPONDING TO IN PLANE DEFLECTIONS CALCULATED
C     CALCULATE STRESSES COR OUT OF PLANE DISPLACEMENTS
      DO 13 KI=1,3
      DO 13 I=1,3
      II=I+(KI-1)*3
      IL=2+I+(KI-1)*6
      DO 13 JJ=1,NCOLN
   13 XP(II,JJ)=XEL(IL,JJ)
C     CALCULATE THE COORD OF CENTROID
      XX=(X1+X2+X3)/3.
      YY=(Y1+Y2+Y3)/3.
      DB(1,4)=2.
      DB(1,7)=6.*XX
      DB(1,8)=2.*YY
      DB(2,6)=2.
      DB(2,8)=2.*XX
      DB(2,9)=6.*YY
      DB(3,5)=2.
      DB(3,8)=4.*(XX+YY)
      CALL CNC(CC,X1,X2,X3,Y1,Y2,Y3)
      CALL INVMAT(CC,9,9,1.0E-12,IERR,N1)
      CALL PRDMAT(DB,CC,DC,3,9,9)
      DO 14 I=1,3
      DO 14 J=1,9
   14 DC(I,J)=-DC(I,J)*T/2.
      CALL PRDMAT(DE,DC,DB,3,3,9)
      DO 43 I=1,3
      DO 43 J=1,9
   43 DC(I,J)=DB(I,J)
C     DC IS THE STRESS MATRIX
      DO 15 J=1,NCOLN
      DO 15 K=1,9
      SXX(J)=SXX(J)+DC(1,K)*XP(K,J)
      SYY(J)=SYY(J)+DC(2,K)*XP(K,J)
   15 SXY(J)=SXY(J)+DC(3,K)*XP(K,J)
C     THE X,Y NORMAL STRESSES AND THE SHEAR STRESS CALCULATED
C     CALCULATE THE PRINCIPAL STRESSES
```

```
      DO 16 J=1,NCOLN
      PA=(SXX(J)+SYY(J))/2.
      PB=SQRT((SXX(J)-SYY(J))**2/4.+SXY(J)**2)
      SMAX(J)=PA+PB
 16   SMIN(J)=PA-PB
      WRITE (6,40) NM
 40   FORMAT(//,5X,11HELEMENT NO=,I5)
      DO 20 J=1,NCOLN
      WRITE(6,30) J,SXX(J),SYY(J),SXY(J),SMAX(J),SMIN(J)
 30   FORMAT(//,5X,I5,5E22.5)
 20   CONTINUE
      RETURN
      END


      SUBROUTINE DEFL(NI,NCOLN,XEL,NM)
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     SUBROUTINE FOR CALCULATING THE ELEMENT NODE DEFLECTIONS  IN ELEMENT
C     COORDINATES FROM THE SYSTEM DEFLECTIONS
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      DIMENSION NI(200,3),XEL(18,2),TL(6,6),DX(6,2),DXT(6,2)
      DIMENSION NSTART(15),NEND(15),NFREE(100),TNF(200,3,3),PP(500,2)
      DIMENSION X(200,3),Y(200,3)
      COMMON NOD,NELEM,EM,V,NSTART,NEND,NFREE,TNF,PP,X,Y
      DO 5 I=1,18
      DO 5 J=1,NCOLN
 5    XEL(I,J)=0.
      DO 15 I=1,6
      DO 15 J=1,NCOLN
 15   DX(I,J)=0.
      DO 20 I=1,6
      DO 20 J=1,6
 20   TL(I,J)=0.
      DO 26 II=1,3
      DO 26 JJ=1,3
      TL(II,JJ)=TNF(NM,II,JJ)
      IA=II+3
      JA=JJ+3
 26   TL(IA,JA)=TNF(NM,II,JJ)
      DO 25 I=1,3
      LL=NI(NM,I)
      IF(LL.EQ.0) GO TO 25
      NF1=NFREE(LL)
      MA=NSUM(NFREE,LL-1)
      DO 27 II=1,NF1
      IL=II+MA
      DO 27 KK=1,NCOLN
 27   DX(II,KK)=PP(IL,KK)
      IF(NF1.EQ.5) GO TO 30
      DO 28 II=1,NF1
      DO 28 JJ=1,NCOLN
      DXT(II,JJ)=0.
      DO 28 KK=1,NF1
 28   DXT(II,JJ)=DXT(II,JJ)+TL(II,KK)*DX(KK,JJ)
      DO 29 II=1,NF1
```

```
      DO 29 JJ=1,NCOLN
 29   DX(II,JJ)=DXT(II,JJ)
 30   DO 31 II=1,NF1
      IN=II+(I-1)*6
      DO 31 JJ=1,NCOLN
 31   XEL(IN,JJ)=DX(II,JJ)
 25   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE TRANF(NM,ST)
      DIMENSION ST(18,18),TED(18,18),TST(18,18),TN(3,3),AL(3,3),TA(3,3)
      DIMENSION NSTART(15),NEND(15),NFREE(100),TNF(200,3,3),PP(500,2)
      DIMENSION X(200,3),Y(200,3),Z(200,3),ALMD(3,3)
      COMMON NOD,NELEM,EM,V,NSTART,NEND,NFREE,TNF,PP,X,Y,Z,ALMD
C     TRANSFORMATION OF ELEMENT STIFFNESS MATRIX FROM LOCAL
C     TO PLATE COORDINATE SYSTEMS.
      DO 1 I=1,18
      DO 1 J=1,18
 1    TED(I,J)=0.0
      DO 2 I=1,3
      DO 2 J=1,3
      TN(I,J)=0.0
      AL(I,J)=0.0
 2    CONTINUE
      DO 3 I=1,3
      DO 3 J=1,3
      AL(I,J)=ALMD(J,I)
      TN(I,J)=TNF(NM,I,J)
 3    CONTINUE
      CALL PRDMAT(TN,AL,TA,3,3,3)
      DO 4 I=1,3
      DO 4 J=1,3
      DO 4 K=1,6
      N=(K-1)*3
      TED(I+N,J+N)=TA(I,J)
 4    CONTINUE
      CALL PRDMAT(TED,ST,TST,18,18,18)
      DO 7 I=1,18
      DO 7 J=1,18
      ST(I,J)=0.0
      DO 7 K=1,18
      ST(I,J)=ST(I,J)+TST(I,K)*TED(J,K)
 7    CONTINUE
      RETURN
      END
```

CD TOT   0050

```
A4463,LC6000,T300,CM75000.                                        WIN M.M.
RUN(S)
SETINDF.
LOADER(PPLOADR)
REDUCE.
LGO.
'          6400 END RECORD
      PROGRAM TST (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE1,TAPE2,
     1TAPE4)
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     PROGRAM TO CALCULATE DEFLECTIONS AND STRESSES IN A SHELL BY
C     THE METHOD TRIDIAGONALIZATION
C     MAIN PROGRAM
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      DIMENSION NSTART(15),NEND(15),NFIRST(15),NLAST(15),NFREE(100)
      DIMENSION PK(6,6),BK(9,9),ST(18,18)
       DIMENSION XEL(18,2)
      DIMENSION A(55,55),C(55,55)
      DIMENSION THICK(200),NI(200,3),X(200,3),Y(200,3)
      DIMENSION PP(500,2),TNF(200,3,3)
      DIMENSION HH(15)
      COMMON NOD,NELEM,EM,V,NSTART,NEND,NFREE,TNF,PP,X,Y
      READ(5,14) NPROB
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     TOCHER  POLYNOMIAL IS USED AS THE DISPLACEMENT FUNCTION.
C     READING DATA
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      DO 1500 NTL=1,NPROB
      WRITE(6,9) NTL
 9    FORMAT(1H1,10X,*PROBLEM NO.*,I5,//)
      READ(5,10) NPART,NOD,NELEM,NCOLN,N6,NXL
 10   FORMAT(20I4)
C     NR IS THE TOTAL NUMBER OF DEGREES OF FREEDOM FOR THE STRUCTURE
      NR=N6*6+(NOD-N6)*5
      READ(5,13) EM,V
 13   FORMAT(2E15.3)
      READ(5,10) ((NSTART(I),NEND(I),NFIRST(I),NLAST(I)),I=1,NPART)
      READ(5,14) (NFREE(I),I=1,NOD)
 14   FORMAT(40I2)
      DO 12 I=1,NR
      DO 12 J=1,NCOLN
 12   PP(I,J)=0.
      DO 8 II=1,NXL
      READ(5,11) (I,(PP(I,J),J=1,NCOLN))
 11   FORMAT(I4,5F10.0)
 8    CONTINUE
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     INPUT SIMPLIFIED FOR THIS STRUCTURE
C     IN THE GENERAL CASE THE NODE NUMBERS NI AND THE TRANSFORMATION
C     MATRICES MUST BE DIRECTLY READ IN
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      NLN=NPART*16
C     NLN IS THE TOTAL NUMBER OF ELEMENTS ON THE LATERAL SURFAE OF THE
C     PYRAMID
      NLNN=NLN+1
C     READ THE NODE NUMBERS FOR THE ELEMENTS IN THE FIRST TWO PARTITION
C     FROM BOTTOM
      READ(5,16)((NI(I,J),J=1,3),I=1,32)
```

```
 16    FORMAT(40I2)
C      CALCULATE THE NODE NUMBERS FOR THE REMAINING ELEMENTS ON THE
C      OUTER SURFACE
       DO 17 NPT=3,NPART
       DO 17 I=1,3
       DO 17 J=1,16
       KK=J+(NPT-2)*16
       K=J+(NPT-1)*16
 17    NI(K,I)=NI(KK,I)+8
C      READ THE NODE NUMBERS FOR THE LAST E SIX ELEMENTS
       READ(5,16) ((NI(I,J),J=1,3),I=NLNN,NELEM)
C      THICKNESS OF ALL THE ELEMENTS EXCEPT THE LAST   SIX ELEMENTS .125
       DO 18 I=1,NLN
       THICK(I)=0.125
 18    CONTINUE
C      THICKNESS OF THE LAST  SIX  ELEMENTS IS 0.5 INCHES
       DO 19 I=NLNN,NELEM
       THICK(I)=0.5
 19    CONTINUE
C      READ THE ANGLES OF INCLINATIONS OF THE FOUR FACES, THE BASE WIDTH
C      AND THE TOTAL HEIGHT
       READ(5,20)AT,BT,CT,DT,H,AA
 20    FORMAT(8F10.0)
C      CONVERT ANGLES INTO RADIANS
       ATR=AT*3.1415926/180.
       BTR=BT*3.1415926/180.
       CTR=CT*3.1415926/180.
       DTR=DT*3.1415926/180.
       ATR=ATAN(48./31.5)
       BTR=ATAN(48./4.)
       CTR=3.1416-ATAN(48./12.)
       DTR=ATAN(48./15.5)
C      READ THE HEIGHTS OF THE PARTITIONS
       READ(5,20)(HH(I),I=1,NPART)
C      CALCULATION OF ELEMENT COORDINATES
       BB=AA
       DO 21 NPT=1,NPART
C      NST IS THE FIRST ELEMENT IN EACH PARTITION
       NST=1+(NPT-1)*16
       HHH=HH(NPT)
       IF(NPT.EQ.1) GO TO 22
       HP=HH(NPT-1)
       AA=AA-HP*COS(DTR)/SIN(DTR)-HP*COS(BTR)/SIN(BTR)
       BB=BB-HP*COS(ATR)/SIN(ATR)-HP*COS(CTR)/SIN(CTR)
C      CALL SUBROUTINE TO CALCULATE THE NODAL CO-ORDINATES FOR EACH TRI-
   22   CALL FORT(AA,BB,ATR,BTR,CTR,DTR,HHH,NST)
 21    CONTINUE
       READ THE NODAL CO-ORDINATES FOR THE LAST   SIX ELEMENTS.
       READ(5,25)((X(I,J),J=1,3),I=NLNN,NELEM)
       READ(5,25)((Y(I,J),J=1,3),I=NLNN,NELEM)
 25    FORMAT(8F10.0)
     ROTATE THE TOP PLANE BY 30 DEG.
       TT=3.1415296/6.0
       DO 26 NM=NLNN,NELEM
       DO 26 I=1,3
       XN=X(NM,I)*COS(TT)+Y(NM,I)*SIN(TT)
       YN=-X(NM,I)*SIN(TT)+Y(NM,I)*COS(TT)
       X(NM,I)=XN
```

```
      Y(NM,I)=YN
 26   CONTINUE
C     INITIALIZE THE ARRAY
      DO 31 NM=1,NELEM
      DO 31 I=1,3
      DO 31 J=1,3
 31   TNF(NM,I,J)=0.
C     TRANSFORMATION MATRIX FOR FACE 1
      DO 27  I=1,4
      DO 27 NM=I,NLN,16
      TNF(NM,1,2)=-1.
      TNF(NM,2,1)=COS(ATR)
      TNF(NM,2,3)=SIN(ATR)
 27   CONTINUE
C     TRANSFORMATION MATRIX FOR FACE  2
      DO 28 I=5,8
      DO 28 NM=I,NLN,16
      TNF(NM,1,1)=1.
      TNF(NM,2,2)=COS(BTR)
 28   TNF(NM,2,3)=SIN(BTR)
C     TRANSFORMATION MATRIX FOR FACE 3
      DO 29 I=9,12
      DO 29 NM=I,NLN,16
      TNF(NM,1,2)=1.
      TNF(NM,2,1)=-COS(CTR)
 29   TNF(NM,2,3)=SIN(CTR)
C     TRANSFORMATION MATRIX FOR FACE 4
      DO 30 I=13,16
      DO 30 NM=I,NLN,16
      TNF(NM,1,1)=-1.
      TNF(NM,2,2)=-COS(DTR)
 30   TNF(NM,2,3)=SIN(DTR)
C     TRANSFORMATION MATRIX FOR THE LAST  SIX  ELEMENTS
      DO 32 NM=NLNN,NELEM
      TNF(NM,1,1)=COS(TT)
      TNF(NM,1,2)=SIN(TT)
      TNF(NM,2,1)=-SIN(TT)
      TNF(NM,2,2)=COS(TT)
 32   TNF(NM,3,3)=1.
C     CALCULATE THE THIRD ROW OF EACH TRANSFORMATION MATRIX IN TERMS
C     OF THE ELEMENTS OF THE FIRST TWO ROWS
      DO 33 NM=1,NELEM
      TNF(NM,3,1)=TNF(NM,1,2)*TNF(NM,2,3)-TNF(NM,2,2)*TNF(NM,1,3)
      TNF(NM,3,2)=TNF(NM,2,1)*TNF(NM,1,3)-TNF(NM,1,1)*TNF(NM,2,3)
 33   TNF(NM,3,3)=TNF(NM,1,1)*TNF(NM,2,2)-TNF(NM,2,1)*TNF(NM,1,2)
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     PRINTING DATA
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      WRITE(6,400) H
 400  FORMAT(10X,20HHEIGHT OF STRUCTURE=,F10.3)
      WRITE(6,401) EM,V
 401  FORMAT(/,10X,19HMATERIAL PROPERTIES,//,10X,2HE=,E12.4,5X,2HV=,
     1E12.4)
      WRITE(6,403) NPART,NOD,NELEM,NCOLN,NXL
 403  FORMAT(//,10X,25HNPART,NOD,NELEM,NCOLN,NXL,/,5I20)
      WRITE(6,402)(HH(I),I=1,NPART)
 402  FORMAT(/,10X,25HHEIGHTS OF PARTITIONS ARE,/,(6F15.3))
      DO 442 J=1,NCOLN
```

```
      DO 442 NL=1,NR
      IF(PP(NL,J).LE.1.0E-8) GO TO 442
      WRITE(6,443) J
  443 FORMAT(/,5X,*EXTERNAL LOAD VECTOR NO.*,I6)
      WRITE(6,446) PP(NL,J),NL
  446 FORMAT(5X,F15.4,*LBS AT*,I6)
  442 CONTINUE
      IF(NTL.NE.1) GO TO 8888
      WRITE(6,410)
  410 FORMAT(1H1,5X,29HNODAL PATTERN AND COORDINATES,//,5X,11HELEMENT NO
     1.,10X,2HNI,32X,1HX,45X,1HY,//)
      DO 406 I=1,NELEM
      WRITE(6,405) I,(NI(I,J),J=1,3),(X(I,J),J=1,3),(Y(I,J),J=1,3)
  406 CONTINUE
  405 FORMAT(I6,10X,3I8,6F15.3)
      WRITE(6,430) (NFREE(I),I=1,NOD)
  430 FORMAT(//,20I5)
      WRITE(6,431) ((NSTART(I),NEND(I),NFIRST(I),NLAST(I)),I=1,NPART)
  431 FORMAT(//,4I25)
      WRITE(6,425) ATR,BTR,CTR,DTR
  425 FORMAT(//,5X,10HTHE ANGLES,//,4E20.3)
      DO 436 NM=1,NELEM
      WRITE(6,437)((TNF(NM,I,J),J=1,3),I=1,3)
  437 FORMAT(9F13.3,/)
  436 CONTINUE
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     FORMATION AND ASSEMBLY OF MATRICES
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 8888 REWIND 1
      REWIND 2
      REWIND 4
      DO 35 NPT=1,NPART
      NA=NSTART(NPT)
      NB=NEND(NPT)
      MA=NSUM(NFREE,NA-1)
      MB=NSUM(NFREE,NB)
      MI=MA+1
      M1=MB-MA
      IF (NPT.GE.NPART) GO TO 39
      NC=NEND(NPT+1)
      MC=NSUM(NFREE,NC)
      M2=MC-MB
      GO TO 5
   39 M2=M1
    5 DO 37 I=1,M1
      DO 38 J=1,M1
   38 A(I,J)=0.
      DO 37 J=1,M2
      C(I,J)=0.
   37 CONTINUE
      LA=NFIRST(NPT)
      LB=NLAST(NPT)
      DO 36 NM=LA,LB
      X1=X(NM,1)
      X2=X(NM,2)
      X3=X(NM,3)
      Y1=Y(NM,1)
```

```
      Y3=Y(NM,3)
      T=THICK(NM)
      CALL PLAN(X1,X2,X3,Y1,Y2,Y3,EM,V,PK,T)
      T=THICK(NM)
      CALL BEND(X1,X2,X3,Y1,Y2,Y3,EM,V,T,BK)
      CALL GROUP(ST,PK,BK)
      CALL ASSEBL(NM,A,C,ST,NI,NPT,NPART)
 36   CONTINUE
      WRITE(2)M1,M2,((A(I,J),I=1,M1),J=1,M1),((C(I,J),I=1,M1),J=1,M2),
     1((PP(I,J),I=MI,MB),J=1,NCOLN)
 35   CONTINUE
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     THE MATRICES ARE FORMED AND WRITTEN IN TAPE 2
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      REWIND 1
      REWIND 2
      REWIND 4
C     CALL SUBROUTINE TO SOLVE THE TRIDIAGONAL EQUATIONS
      CALL RECUR(A,C,NPART,NCOLN)
      REWIND 1
      DO 200 MM=1,NPART
      NPT=NPART+1-MM
      NA=NSTART(NPT)
      NB=NEND(NPT)
      MA=NSUM(NFREE,NA-1)
      MB=NSUM(NFREE,NB)
      MA=MA+1
C     READ SYSTEM DISPLACEMENTS FROM TAPE 1
      READ(1) ((PP(I,J),I=MA,MB),J=1,NCOLN)
 200  CONTINUE
      WRITE(6,415)
  415 FORMAT(1H1,5X,38HTHE STRESSES SXX,SYY,SXY,SMAX,SMI ARE,//)
C    ******************************************************************
C     CALL SUBROUTINE TO CALCULATE ELEMENT DISPLACEMENTS AND STRESSES
C    ******************************************************************
      DO 215 NM=1,NELEM
      CALL DEFL(NI,NCOLN,XEL,NM)
      T=THICK(NM)
      CALL STRESS(NCOLN,XEL,T,NM)
  215  CONTINUE
 1500 CONTINUE
      STOP
      END


      FUNCTION NSUM(NFREE,NN)
C     SUBPROGRAM TO FOR CALCULATING THE TOTAL NO. OF DEGREES OF FREEDOM
C     UPTO A PARTICULAR NODE
      DIMENSION NFREE(100)
C
      NSUM=0
      IF (NN.EQ.0) RETURN
      DO 10 I=1,NN
      NSUM=NSUM+NFREE(I)
 10   CONTINUE
      RETURN
      END
```

```
A4463,CM120000,T2000.                                          WIN M.M.
RUN(S)

SETINDF.
LOADER(PPLOADR)
REDUCE.

'          6400 END OF RECORD
      1TAPE4)
       PROGRAM TST (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE1,TAPE2,
C      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     COWPER POLYNOMIAL IS USED AS THE DEFLECTION FUNCTION.
C      PROGRAM TO CALCULATE DEFLECTIONS IN A SHELL BY
C      THE METHOD TRIDIAGONALIZATION
C      MAIN PROGRAM
C      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
       DIMENSION NSTART(15),NEND(15),NFIRST(15),NLAST(15),NFREE(100)
       DIMENSION PK(6,6),BK(18,18),ST(27,27),M(20),N(20),RTF(9,9)
       DIMENSION HH(15)
        DIMENSION PP(685,2),TNF(166,3,3),THICK(166),NI(166,3),X(166,3)
       DIMENSION Y(166,3)
       DIMENSION A(72,72),C(72,72),THITA(166)
       COMMON NOD,NELEM,EM,V,NSTART,NEND,NFREE,TNF,PP
       COMMON A,C,NI,RTF,X,Y
       READ(5,14) NPROB
C      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     THE DEFLECTION OF THE FOLDED PLATE STRUCTURE IS FOUND
        PI=3.1415926
C     BY THE COWPER POLYNOMIAL .
C      READING DATA
C      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
       DO 1500 NTL=1,NPROB
       WRITE(6,9) NTL
 9     FORMAT(1H1,10X,*PROBLEM NO.*,I5,//)
       READ(5,14) (M(I),I=1,20),(N(I),I=1,20)
       READ(5,10) NPART,NOD,NELEM,NCOLN,N9,NXL
 10    FORMAT(20I4)
C      NR IS THE TOTAL NUMBER OF DEGREES OF FREEDOM FOR THE STRUCTURE
        NODE=NOD-8
C     NO. OF EFFECTIVE NODES,EXCLUDING THE ORIGINAL ASSUMED NODES
C     AT THE BASE, MAKING THE BASE RIGIDLY FIXED.
        NE9=N9-4
        NR=NE9*9+(NODE-NE9)*8+4
        NPARD=NPART+1
C     NO. OF PARTITIONS, STARTING FROM THE BASE WHICH IS ASSUMED
C     NOT FIXED.
       READ(5,13) EM,V
 13    FORMAT(2E15.3)

        READ(5,10)((NSTART(I),NEND(I),NFIRST(I),NLAST(I)),I=1,NPARD)
       READ(5,14) (NFREE(I),I=1,NOD)
 14    FORMAT(40I2)
       DO 12 I=1,NR
       DO 12 J=1,NCOLN
 12    PP(I,J)=0.
       DO 8 II=1,NXL
       READ(5,11) (I,(PP(I,J),J=1,NCOLN))
 11    FORMAT(I4,5F10.0)
 8     CONTINUE
C      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C      INPUT SIMPLIFIED FOR THIS STRUCTURE
```

```
C     IN THE GENERAL CASE THE NODE NUMBERS NI AND THE TRANSFORMATION
C     MATRICES MUST BE DIRECTLY READ IN
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      NLN=NPART*16
C     NLN IS THE TOTAL NUMBER OF ELEMENTS ON THE LATERAL SURFAE OF THE
C     PYRAMID
      NLNN=NLN+1
C     READ THE NODE NUMBERS FOR THE ELEMENTS IN THE FIRST TWO PARTITIONS
C     FROM BOTTOM
       READ(5,16) ((NI(I,J),J=1,3),I=1,16)
 16   FORMAT(40I2)
C     CALCULATE THE NODE NUMBERS FOR THE REMAINING ELEMENTS ON THE
C     OUTER SURFACE
       DO 17 NPT=2,NPART
      DO 17 I=1,3
      DO 17 J=1,16
      KK=J+(NPT-2)*16
      K=J+(NPT-1)*16
 17   NI(K,I)=NI(KK,I)+8
C      READ THE NODE NUMBERS FOR THE LAST SIX   ELEMENTS
      READ(5,16) ((NI(I,J),J=1,3),I=NLNN,NELEM)
C     THICKNESS OF ALL THE ELEMENTS EXCEPT THE LAST SIX ELEMENTS IS .125
      DO 18 I=1,NLN
      THICK(I)=0.125
 18   CONTINUE
C     THICKNESS OF THE LAST  SIX  ELEMENTS IS 0.5 INCHES
      DO 19 I=NLNN,NELEM
      THICK(I)=0.5
 19   CONTINUE
C     READ THE ANGLES OF INCLINATIONS OF THE FOUR FACES, THE BASE WIDTH
C     AND THE TOTAL HEIGHT
      READ(5,20)AT,BT,CT,DT,H,AA
 20   FORMAT(8F10.0)
       DO 1115 NM=1,NELEM
 1115  THITA(NM)=0.0
C     CONVERT ANGLES INTO RADIANS
      ATR=AT*3.1415926/180.
      BTR=BT*3.1415926/180.
      CTR=CT*3.1415926/180.
      DTR=DT*3.1415926/180.
      ATR=ATAN(48./31.5)
      BTR=ATAN(48./4.)
      CTR=3.1416-ATAN(48%/12.)
      DTR=ATAN(48./15.5)
C     READ THE HEIGHTS OF THE PARTITIONS
      READ(5,20)(HH(I),I=1,NPART)
C     CALCULATION OF ELEMENT COORDINATES
      BB=AA
      DO 21 NPT=1,NPART
C     NST IS THE FIRST ELEMENT IN EACH PARTITION
      NST=1+(NPT-1)*16
      HHH=HH(NPT)
      IF(NPT.EQ.1) GO TO 22
      HP=HH(NPT-1)
      AA=AA-HP*COS(DTR)/SIN(DTR)-HP*COS(BTR)/SIN(BTR)
      BB=BB-HP*COS(ATR)/SIN(ATR)-HP*COS(CTR)/SIN(CTR)
C     CALL SUBROUTINE TO CALCULATE THE NODAL CO-ORDINATES FOR EACH TRIANGLE
  22   CALL FORT(AA,BB,ATR,BTR,CTR,DTR,HHH,NST)
 21   CONTINUE
C     READ THE NODAL CO-ORDINATES FOR THE LAST  SIX  ELEMENTS
```

```
      READ(5,25)((X(I,J),J=1,3),I=NLNN,NELEM)
      READ(5,25)((Y(I,J),J=1,3),I=NLNN,NELEM)
 25   FORMAT(8F10.0)
C     INITIALIZE THE ARRAY
      DO 31 NM=1,NELEM
      DO 31 I=1,3
      DO 31 J=1,3
 31   TNF(NM,I,J)=0.
C     TRANSFORMATION MATRIX FOR FACE 1
      DO 27  I=1,4
      DO 27 NM=I,NLN,16
      TNF(NM,1,2)=-1.
      TNF(NM,2,1)=COS(ATR)
      TNF(NM,2,3)=SIN(ATR)
 27   CONTINUE
C     TRANSFORMATION MATRIX FOR FACE  2
      DO 28 I=5,8
      DO 28 NM=I,NLN,16
      TNF(NM,1,1)=1.
      TNF(NM,2,2)=COS(BTR)
 28   TNF(NM,2,3)=SIN(BTR)
C     TRANSFORMATION MATRIX FOR FACE 3
      DO 29 I=9,12
      DO 29 NM=I,NLN,16
      TNF(NM,1,2)=1.
      TNF(NM,2,1)=-COS(CTR)
 29   TNF(NM,2,3)=SIN(CTR)
C     TRANSFORMATION MATRIX FOR FACE 4
      DO 30 I=13,16
      DO 30 NM=I,NLN,16
      TNF(NM,1,1)=-1.
      TNF(NM,2,2)=-COS(DTR)
 30   TNF(NM,2,3)=SIN(DTR)
C     TRANSFORMATION MATRIX FOR THE LAST EIGHT ELEMENTS
      DO 32 NM=NLNN,NELEM
       DO32 I=1,3
       DO32 J=1,3
       TNF(NM,I,J)=0.0
       IF(I.EQ.J) TNF(NM,I,J)=1.0
 32    CONTINUE
C     CALCULATE THE THIRD ROW OF EACH TRANSFORMATION MATRIX IN TERMS
C     OF THE ELEMENTS OF THE FIRST TWO ROWS
      DO 33 NM=1,NELEM
      TNF(NM,3,1)=TNF(NM,1,2)*TNF(NM,2,3)-TNF(NM,2,2)*TNF(NM,1,3)
      TNF(NM,3,2)=TNF(NM,2,1)*TNF(NM,1,3)-TNF(NM,1,1)*TNF(NM,2,3)
 33   TNF(NM,3,3)=TNF(NM,1,1)*TNF(NM,2,2)-TNF(NM,2,1)*TNF(NM,1,2)
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     PRINTING DATA
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      WRITE(6,400) H
 400  FORMAT(10X,20HHEIGHT OF STRUCTURE=,F10.3)
      WRITE(6,401) EM,V
 401  FORMAT(/,10X,19HMATERIAL PROPERTIES,//,10X,2HE=,E12.4,5X,2HV=,
     1E12.4)
      WRITE(6,403) NPART,NOD,NELEM,NCOLN,NXL
 403  FORMAT(//,10X,25HNPART,NOD,NELEM,NCOLN,NXL,/,5I20)
      WRITE(6,402)(HH(I),I=1,NPART)
 402  FORMAT(/,10X,25HHEIGHTS OF PARTITIONS ARE,/,(6F15.3))
      DO 442 J=1,NCOLN
      DO 442 NL=1,NR
```

```
      IF(PP(NL,J).LE.1.0E-8) GO TO 442
      WRITE(6,443) J
  443 FORMAT(/,5X,*EXTERNAL LOAD VECTOR NO.*,I6)
      WRITE(6,446) PP(NL,J),NL
  446 FORMAT(5X,F15.4,*LBS AT*,I6)
  442 CONTINUE
      IF(NTL.NE.1) GO TO 8888
      WRITE(6,410)
  410 FORMAT(1H1,5X,29HNODAL PATTERN AND COORDINATES,//,5X,11HELEMENT NO
     1.,10X,2HNI,32X,1HX,45X,1HY,//)
      DO 406 I=1,NELEM
      WRITE(6,405) I,(NI(I,J),J=1,3),(X(I,J),J=1,3),(Y(I,J),J=1,3)
  406 CONTINUE
  405 FORMAT(I6,10X,3I8,6F15.3)
      WRITE(6,430) (NFREE(I),I=1,NOD)
       WRITE(6,430) (NFREE(I),I=1,NODE)
  430 FORMAT(//,20I5)
       WRITE(6,431) ((NSTART(I),NEND(I),NFIRST(I),NLAST(I)),I=1,NPARD)
  431 FORMAT(//,4I25)
      WRITE(6,425) ATR,BTR,CTR,DTR
  425 FORMAT(//,5X,10HTHE ANGLES,//,4E20.3)
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     FORMATION AND ASSEMBLY OF MATRICES
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 8888 REWIND 1
      REWIND 2
      REWIND 4
       DO 35 NPT=1,NPARD
      NA=NSTART(NPT)
      NB=NEND(NPT)
       MA=NSUM(NFREE,NA-1,NPT)
       MB=NSUM(NFREE,NB,NPT)
      MI=MA+1
      M1=MB-MA
       IF(NPT.GE.NPARD) GO TO 39
      NC=NEND(NPT+1)
       MC=NSUM(NFREE,NC,NPT)
      M2=MC-MB
      GO TO 5
   39 M2=M1
    5 DO 37 I=1,M1
      DO 38 J=1,M1
   38 A(I,J)=0.
      DO 37 J=1,M2
      C(I,J)=0.
   37 CONTINUE
      LA=NFIRST(NPT)
      LB=NLAST(NPT)
      DO 36 NM=LA,LB
       IF(ABS(Y(NM,2)-Y(NM,1)).LT.1.0E-8) GO TO 1113
       IF(ABS(X(NM,2)-X(NM,1)).LT.1.0E-8) GO TO 1112
      THITA(NM)=ATAN((Y(NM,2)-Y(NM,1))/(X(NM,2)-X(NM,1)))
      S=SQRT((X(NM,2)-X(NM,1))**2+(Y(NM,2)-Y(NM,1))**2)
      P=((X(NM,2)-X(NM,3))*(X(NM,2)-X(NM,1))+(Y(NM,2)-Y(NM,3))*(Y(NM,2)-
     1Y(NM,1)))/S
       P=ABS(P)
      Q=((X(NM,3)-X(NM,1))*(X(NM,2)-X(NM,1))+(Y(NM,3)-Y(NM,1))*(Y(NM,2)-
     1Y(NM,1)))/S
       Q=ABS(Q)
      R=((X(NM,2)-X(NM,1))*(Y(NM,3)-Y(NM,1))-(X(NM,3)-X(NM,1))*(Y(NM,2)-
     1Y(NM,1)))/S
```

```
      R=ABS(R)
C     P,Q,R-THE MAGNITUTE OF THE ELEMENT,HENCE THE ABS. VALUES ARE
C     CONSIDERED.
      GO TO 1114
 1113 P=X(NM,2)-X(NM,3)
      P=ABS(P)
      Q=X(NM,3)-X(NM,1)
      Q=ABS(Q)
      R=Y(NM,3)
      R=ABS(R)
      S=ABS(P+Q)
      GO TO 1114
 1112 P=Y(NM,2)-Y(NM,3)
      P=ABS(P)
      Q=Y(NM,3)-Y(NM,1)
      Q=ABS(Q)
      R=X(NM,1)-X(NM,3)
      R=ABS(R)
      S=ABS(P+Q)
C     TRANSLATION AND ROTATION OF AXES ARE CONSIDERED.
C     STIFFNESS MATRIX OF THE ELEMENT IS TRANSFERRED TO THE PLATE COORDINATE
C     SYSTEM BEFORE ASSEMBLY.
C     THE COORDINATES OF THE LOCAL  SYSTEM ARE OBTAINED BY TAKING THE APPROP
C     VALUES OF THE VARIBLES P,Q, AND R.
 1114 X1=-Q
      X2=P
      X3=0.
      Y1=0.
      Y2=0.
      Y3=R
      T=THICK(NM)
      CALL PLAN(X1,X2,X3,Y1,Y2,Y3,EM,V,PK,T)
      T=THICK(NM)
      CALL BEND(P,Q,R,M,N,V,EM,T,BK)
      CALL GROUP(ST,PK,BK)
      ALPHA=THITA(NM)
      CALL TRANF1(ALPHA,ST)
      CALL ASSEBL(NM,ST,NPT,NPARD)
   36 CONTINUE
      IF(NPT.NE.1) GO TO 565
      CALL ELIM(M1,M2 )
      M1=4
      MB=4
  565 WRITE(2)M1,M2,((A(I,J),I=1,M1),J=1,M1),((C(I,J),I=1,M1),J=1,M2),
     1((PP(I,J),I=MI,MB),J=1,NCOLN)
   35 CONTINUE
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     THE MATRICES ARE FORMED AND WRITTEN IN TAPE 2
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      REWIND 1
      REWIND 2
      REWIND 4
C     CALL SUBROUTINE TO SOLVE THE TRIDIAGONAL EQUATIONS
      CALL RECUR (NPARD,NCOLN)
 1500 CONTINUE
      STOP
      END
```

CD TOT   0300

APPENDIX VII - LIST OF EQUIPMENT

LIST OF EQUIPMENTS

1. OBLIQUE TRUNCATED PYRAMID LIKE STRUCTURE.
2. STORAGE OSCILLOSCOPE (TYPE 564).
3. OSCILLATOR (TYPE DISA 51EO2 555).
4. REACTANCE CONVERTER (TYPE DISA 51E01).
5. STRAIN INDICATOR (TYPE BUDD MODEL P-350).
6. MICROMETER PROXIMITY TRANSDUCER (TYPE DISA 51D11).
7. STRAIN GAGES AND ALLIED EQUIPMENTS.
8. SWITCH AND BALANCE UNITS (SERIAL NO. 005520).
9. TURNBUCKLE.
10. LOAD CELL WITH WIRE CORD.
11. TINIUS OLSEN TESTING MACHINE (NO. 66712).
12. CDC 6400 DIGITAL COMPUTER.