A FINITE DIFFERENCE BASED FINITE STRIP METHOD FOR THE ANALYSIS OF TRANSLATIONAL SHELL STRUCTURES

by

R. BARKER

A thesis submitted in partial fulfilment of the requirements for the Degree of Master of Science in the Faculty of Engineering, University of Cape Town.

Department of Civil Engineering,
UNIVERSITY OF CAPE TOWN.

September, 1976.

SYNOPSIS

A numerical method for the analysis of translational shell structures is presented. The finite strip concept is utilised together with finite difference approximations to the differential equations along nodal lines. Numerical examples include open and closed translational shells, with various end conditions and continuity over intermediate supports.

DECLARATION OF CANDIDATE

I, Roger Barker, hereby declare that this thesis is my own work and that it has not been submitted for a degree at another University.

Signed by candidate

September, 1976.

This is my commandment, That ye love one another, as I have loved you.

John 15:12

ACKNOWLEDGEMENTS

The writer wishes to express his appreciation to:

Mr. W.S. Doyle*, under whose guidance this research was conducted, for his enthusiasm, help and interest throughout the thesis period.

Professor J.B. Martin†, for allocating teaching duties in such a manner as to make personal research possible.

Mackintosh, Bergh and Sturgess, Consulting Engineers for financial support.

Scott and de Waal, Consulting Engineers, for financial assistance.

Bridget Spalding for typing the manuscript, and to Charles Basson for producing copies of this thesis.

*Senior Lecturer, Department of Civil Engineering, University of Cape Town.

†Professor and Head of Department of Civil Engineering, University of Cape Town.

NOTATION

UPPER CASE CHARACTERS

A, B, C, D	Bending stiffness matrices.
D""	Fourth derivative of displacement vector D.
D_x, D_y, D_{xy}	Flexural rigidities for orthotropic plates.
E	Load vector for bending forces.
E_x, E_y	Young's modulus in x and y directions.
G	Modulus of elasticity in shear.
J	Load vector for in-plane forces.
K, L, M	In-plane stiffness matrices.
M_x, M_y	Bending moments per unit length perpendicular to x and y axes respectively.
M_{xy}	Twisting moment per unit length perpendicular to x axis.
M'_x, M'_y, M'_{xy}	Bending and twisting forces in local co-ordinate system.
T	Transformation matrix.
Up, Ub	Total potential energy for in-plane and bending conditions respectively.
δUp, δUb	First variation of Total Potential Energy for in-plane and bending conditions respectively.
$\Delta\delta$Up, $\Delta\delta$Ub	"Part of" expressions given above.
X, Y, Z	Rectangular co-ordinate axes.
X_i, Y_i	Edge forces on strip in x and y directions respectively.

LOWER CASE CHARACTERS

d_x, d_y, d_{xy}	Modified Young's modulus values for in-plane forces.
h	Strip thickness.
r_x, r_y, r_{xy}	Radii of curvature in x and y directions and for twisting moment respectively.
u, v, w	Global displacements in x, y and z directions respectively.

GREEK CHARACTERS

γ_{xy}	Shear strain.
ϵ_x, ϵ_y	Strains in x and y directions respectively.
θ	Global rotation about x axis.
σ_x, σ_y	Direct stresses in x and y direction respectively in global co-ordinates.
σ'_x, σ'_y	Direct stresses in x and y directions respectively in local co-ordinates.
τ_{xy}	Shear stress in global co-ordinates.
τ'_{xy}	Shear stress in local co-ordinates.
ν_x, ν_y	Poisson's ratio in x and y directions respectively.
ϕ	Angle strip makes with global axis system.

CONTENTS

CHAPTER I

INTRODUCTION

The literature on the analysis of folded plate structures may be considered to fall into seven principal categories with regard to method of analysis. They are:-

(1) Beam method.
(2) Folded plate theory, neglecting relative joint displacements.
(3) Folded plate theory considering relative joint displacements.
(4) "Elasticity" method.
(5) Finite difference method.
(6) Finite element method.
(7) Finite strip method.

The Beam Method uses conventional beam theory, which requires all cross-sections to remain the same under load. If intermediate transverse stiffening ribs are frequent, this method yields good results, but more rigorous analyses and experiments have shown that this method cannot be applied to more intricate structures.

Methods (2) and (3) both consider the longitudinal supporting action of each plate to be governed by beam theory and the transverse supporting action to be that of a continuous one-way slab. Method (2) assumes that the changes in transverse bending moment and in longitudinal stresses due to relative joint displacements are negligible in comparison with the values of these moments and stresses computed on the basis of no relative joint displacements. For the general case however, these stresses are not negligible and the method is not recommended.

Folded Plate Theory Considering Relative Joint Displacements (3) takes into account the effect of relative displacement of the joints or the transverse moments and membrane stresses. Practical methods of analysis include those by V.Z. Vlasov, which uses a Fourier Series approximation; the Portland Cement Association; Gaafar, which uses the principle of superposition; and Yitzhaki who has also included the application of plasticity to folded plate structures.

The "Elasticity" Method, developed by Goldberg and Leve combines the equations of classical plate theory, for loads normal to the plane of the plates, and the elasticity equations defining the plane stress problems, for loads in the plane of the plates. Applied loading is approximated by a Fourier Series.

Methods (1) to (4) are generally restricted to single spans with simple loading and edge conditions. (Reference [1] contains a detailed discussion on these methods.)

The application of Finite Difference Equations to shell analysis was performed in the 1950's, the advent of high speed digital computers making it possible to solve the resulting linear algebraic equations.

The method of Finite Elements was developed by many authors, and folded plate problems have been solved using triangular, rectangular, quadrilateral, isoparametric and thick-shell elements. With this method all possible boundary conditions are soluble [14].

A Finite Strip method of analysis was developed by Y.K. Cheung, first for flat plates [20] and then for folded plate structures [4]. In this method the structure is sub-divided into longitudinal strips which extend from one boundary to the other, making the method ideal for the solution of constant cross-section shell structures. Linear displacement functions are used for the in-plane displacements and cubic polynomials for the displacements normal to the strip surface. The displacements along the length of the strip are approximated by Fourier Series harmonics, likewise the strip loading. The Principle of Minimum Potential Energy is utilised to give the strip stiffness matrix in explicit form. The solution of the displacements is achieved in the normal manner.

Since Fourier Series utilisation becomes impractical for structures with more than three spans, and free boundary conditions cannot be used, du Preez [7] developed a general finite strip method which is capable of solving continuous structures with no boundary condition restriction. The interconnection of strips and other elements is also claimed.

Louw [8] subsequently developed a finite strip method of analysis for flat plates using finite difference approximations to the differential equations along the strip edges. Clamped and simple supported boundaries were considered.

This thesis is an extension of Louw's work and includes in-plane forces enabling constant cross-section folded plates, closed box structures and translational shells to be analysed. Support conditions include simple, fixed and guided ends as well as continuity over interior supports.

C H A P T E R   2

GOVERNING DIFFERENTIAL EQUATIONS FOR FINITE STRIP ANALYSIS

Figure 2.1 shows a typical finite strip element. The strip is of constant cross-sectional shape and has four degrees of freedom per edge, (i.e. u_i, v_i, w_i, θ_i). The right-hand axis system, and the corresponding displacements are also shown.

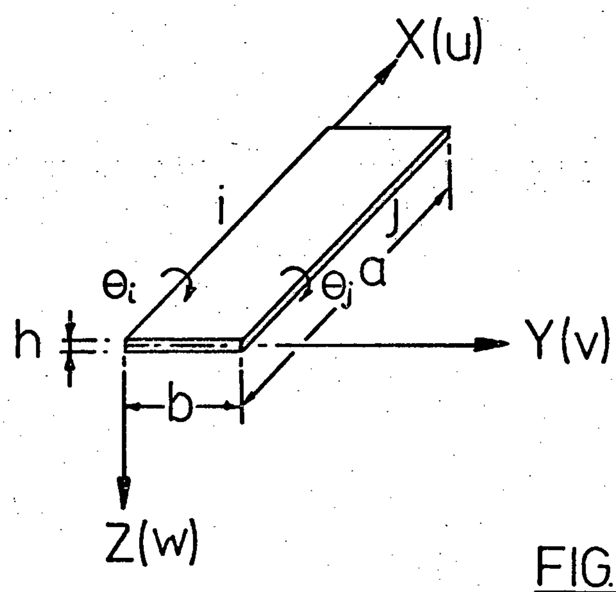


FIG. 2.1

Since in-plane, or membrane stresses, and bending stresses are not coupled, it is convenient to derive the equations for the two cases separately.

2.1 In-plane stiffness of strip

Consider the plate element in Figure 2.2 subjected to in-plane or membrane stresses as shown.

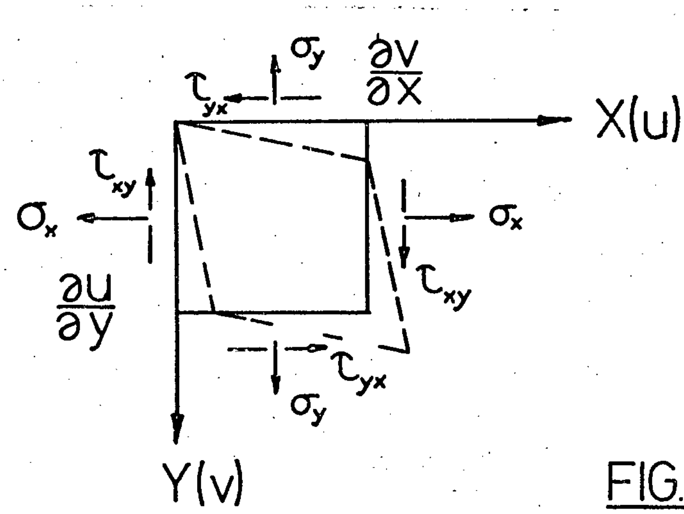


FIG. 2.2

Strains in the X, Y directions and the shear strain can be given by the following linear functions of the displacement gradients:-

$$\epsilon_x = \frac{\partial u}{\partial x}$$
$$\epsilon_y = \frac{\partial v}{\partial y} \qquad (2.1)$$
$$\gamma_{xy} = \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}$$

Writing these equations in terms of stresses and including the effect of Poisson's Ratio:-

$$\epsilon_x = \frac{\sigma_x}{Ex} - \nu_y \frac{\sigma_y}{Ey}$$
$$\epsilon_y = \frac{\sigma_y}{Ey} - \nu_x \frac{\sigma_x}{Ex}$$

The stress terms can be made the dependent variables, hence:-

$$\sigma_x = \frac{Ex}{(1 - \nu_x \nu_y)} [\epsilon_x + \nu_y \epsilon_y]$$
$$\sigma_y = \frac{Ey}{(1 - \nu_x \nu_y)} [\epsilon_y + \nu_x \epsilon_x] \qquad (2.2)$$

Also $\tau_{xy} = G.\gamma_{xy}$

Consider the strip as shown in Figure 2.1. Let the in-plane displacements at edge i be u_i and v_i, and those at edge j be u_j and v_j. The displacement at any point on the strip surface in the u direction can be made up of linear combinations of u_i and u_j.

Thus $u = f_1(u_i) + f_2(u_j)$,

where f_1 and f_2 are linear functions. Since u is a function of x and y, it is convenient to separate their dependence so that u_i and u_j are functions of x only and f_1 and f_2, functions of y only.

Hence $u(x,y) = f_1(y).u_i(x) + f_2(y).u_j(x)$.

At this stage, only y dependent functions need be given any specific form as the x dependent functions are delt with using finite difference approximations.

Since y increases from edge i to edge j, and $u(x,y)$ must have the value of $u_i(x)$ at edge i and $u_j(x)$ at edge j, $f_1(y)$ can be replaced by $(1 - y/b)$ and $f_2(y)$ by (y/b),

$$\text{i.e.}\quad u(x,y) = (1 - \tfrac{y}{b}).u_i(x) + (\tfrac{y}{b}).u_j(x).$$

By similar reasoning:-

$$v(x,y) = (1 - \tfrac{y}{b}).v_i(x) + (\tfrac{y}{b}).v_j(x).$$

Dropping the (x,y) post-script for u and v and expressing the displacements in matrix form:-

$$\begin{bmatrix} u \\ v \end{bmatrix} = C_p.\psi_p \tag{2.3}$$

$$\text{where}\quad C_p = \begin{bmatrix} (1 - \frac{y}{b}) & 0 & (\frac{y}{b}) & 0 \\ 0 & (1 - \frac{y}{b}) & 0 & (\frac{y}{b}) \end{bmatrix} \tag{2.4}$$

$$\text{and}\quad \psi_p = [u_i(x)\ v_i(x)\ u_j(x)\ v_j(x)]^T \tag{2.5}$$

(The subscript p indicates in-plane consideration).

Let the equations in (2.1) be represented by one strain matrix:-

$$[\epsilon] = \begin{bmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{bmatrix} = \begin{bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \end{bmatrix} \tag{2.6}$$

Calculating partial derivatives and substituting into (2.6):-

$$\epsilon = \begin{bmatrix} (1 - \frac{y}{b}).u_i'(x) + (\frac{y}{b}).u_j'(x) \\ (-\frac{1}{b}).v_i(x) + (\frac{1}{b}).v_j(x) \\ \left((-\frac{1}{b}).u_i(x) + (\frac{1}{b}).u_j(x)\right) + \left((1 - \frac{y}{b}).v_i'(x) + (\frac{y}{b}).v_j'(x)\right) \end{bmatrix} \tag{2.7}$$

The stresses in (2.2) can be represented by one stress matrix:-

$$[\sigma] = \begin{bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{bmatrix} = [D_p][\epsilon] \qquad (2.8)$$

Where

$$[D_p] = \begin{bmatrix} \frac{E_x}{(1-\nu_x\nu_y)} & \frac{\nu_x E_y}{(1-\nu_x\nu_y)} & 0 \\ \frac{\nu_x E_y}{(1-\nu_x\nu_y)} & \frac{E_y}{(1-\nu_x\nu_y)} & 0 \\ 0 & 0 & G \end{bmatrix} \qquad (2.9)$$

and $[\epsilon]$ is as given in (2.6).

The Total Potential Energy of a finite strip for in-plane forces can be written:-

$$U_p = \frac{h}{2}\int_X\int_Y (\sigma_x.\epsilon_x + \sigma_y.\epsilon_y + \tau_{xy}.\gamma_{xy})dy.dx \quad \text{(Internal Strain Energy)}$$

$$- \int_X (X_i.u_i + Y_i.v_i + X_i.u_j + Y_j.v_j)dx \quad \text{(Edge Forces)}$$

$$- \int_X\int_Y (X.u + Y.v)dy.dx \quad \text{(Surface Forces)} \qquad (2.10)$$

In matrix notation:-

$$U_p = \frac{h}{2}\int_X\int_Y \epsilon^T.D_p.\epsilon.dy.dx - \int_X (u_i v_i u_j v_j) \begin{bmatrix} X_i \\ Y_i \\ X_j \\ Y_j \end{bmatrix} dx$$

$$- \int_X\int_Y (u\ v).\begin{bmatrix} X \\ Y \end{bmatrix} dx.dy \qquad (2.11)$$

Equation (2.11) is expanded term by term in its three natural subdivisions. The following substitutions are also made to avoid unnecessary complication:-

$$d_x = \frac{E_x}{(1 - \nu_x\nu_y)}$$
$$d_y = \frac{E_y}{(1 - \nu_x\nu_y)} \qquad (2.12)$$
$$d_{xy} = \frac{\nu_x E_y}{(1 - \nu_x\nu_y)}$$

Expansion of Internal Strain Energy Term

Only an outline of the process is given as it involves excessive arithmetic.

(a) Expansion of $D_p.\epsilon$.

(b) Evaluation of ϵ^T.

(c) Multiplication of $\epsilon^T.D_p.\epsilon$.

(d) Integrating with respect to y, across the strip.

(e) Using the Principle of Minimum Potential Energy, the first variation of U_p with respect to the displacements involved (u_i, v_i, u_j, v_j; u_i', v_i', u_j', v_j') must be calculated and equated to zero.

The first variation of U_p with respect to u_i will be a partial differentiation of U_p with respect to u_i, which must be multiplied by the first variation of u_i itself, i.e. δu_i.

The process is repeated for each of the displacements mentioned which have first variations δv_i, δu_j, δv_j; $\delta u_i'$, $\delta v_i'$, $\delta u_j'$ and $\delta v_j'$.

(f) The result of the process oultined in (e) above is then sorted into three matrices; a row vector involving the first variations of the displacements (1 × 8), a square non-symmetrical matrix involving the elements of equation (2.12), G, and b as well as the constant h (8 × 8) and a column vector involving the displacements (8 × 1).

(g) The non-symmetrical (8 × 8) matrix is subdivided into four (4 × 4) matrices (K, L, M and M^T) as given in Appendix 'A'.

(h) Substituting (2.5) and grouping gives: (The Δ prefix to δU_p is used to indicate a part of the total δU_p).

$$\Delta\delta U_p = \int_X \left\{ \delta\psi_p^T [K.\psi_p + M.\psi_p'] + \delta\psi_p'^T [M^T.\psi_p + L.\psi_p'] \right\} dx \qquad (2.13)$$

Expansion of Edge and Surface Force Terms

From equation (2.11) the edge force term can be written:

$$\Delta U_p = - \int_X \psi_p^T . \begin{bmatrix} X_i \\ Y_i \\ X_j \\ Y_j \end{bmatrix} . \ dx.$$

Let the load vector above be represented by [J]. When using finite difference approximations to the differential equations [J] becomes:

$$[J] = \begin{bmatrix} \frac{X_i}{2} & \frac{Y_i}{SA} & \frac{X_j}{2} & \frac{Y_i}{SA} \end{bmatrix}^T ,$$

where SA is the distance between nodes in the longitudinal direction.

The surface force terms are required, for analysis purposes, to be proportioned and placed on the adjoining strip edges. Thus all in-plane surface loads can be included in the J matrix above. Thus edge and surface forces may be represented by:

$$\Delta U_p = - \int_X \psi_p^T . J . dx. \qquad (2.14)$$

The first variation of ΔU_p with respect to the displacements contained in ψ_p^T is:

$$\delta\Delta U_p = - \int_X \delta\psi_p^T . J . dx. \qquad (2.15)$$

Integration with Respect to the Variable X

Equations (2.13) and (2.15) are added giving the total expression for the first variation of Total Potential Energy for one strip, i.e.

$$\delta U_p = \int_X \left\{ \delta\psi_p^T [K.\psi_p + M.\psi_p'] + \delta\psi_p'^T [M^T.\psi_p + L.\psi_p'] - \delta\psi_p^T . J \right\} dx \qquad (2.16)$$

Integration of term involving $\delta\psi_p'^T$ by parts

$$\delta U_p = \delta\Psi_p^T[M^T.\Psi_p + L.\Psi_p']\Big|_o^a + \int_X \delta\Psi_p^T[-(M^T.\Psi_p' + L\Psi_p'') + (K.\Psi_p + M\Psi_p') - J]dx$$

$\delta\Psi_p^T$ is an arbitrary displacement vector which can be zero or any small finite quantity depending on the boundary conditions prescribed. For the Total Potential Energy to be a minimum, U_p is differentiated with respect to the generalised displacements, one at a time and each must be zero, hence:

$$\delta\Psi_p^T[M^T.\Psi_p + L.\Psi_p']\Big|_o^a = 0 \tag{2.17}$$

and

$$\int_X \delta\Psi_p^T[- (M^T.\Psi_p' + L.\Psi_p'') + (K.\Psi_p + M.\Psi_p') - J]dx = 0 \tag{2.18}$$

Equation (2.17) is a boundary condition equation which must be satisfied at $x = 0$ or $x = a$. Equation (2.18) represents the strip section away from the boundaries where the displacement vector, $\delta\Psi_p^T$ is generally non-zero, and hence for equation (2.18) to be satisfied, the integrand must be zero, giving:

$$\left.\begin{aligned} &\delta\Psi_p^T[M^T.\Psi_p + L.\Psi_p']\Big|_o^a = 0 \\ &- L.\Psi_p'' + (M - M^T).\Psi_p' + K.\Psi_p - J = 0 \end{aligned}\right\} \tag{2.19}$$

2.2 BENDING STIFFNESS OF STRIP

Consider a strip cross-section as shown in Figure 2.3 below:

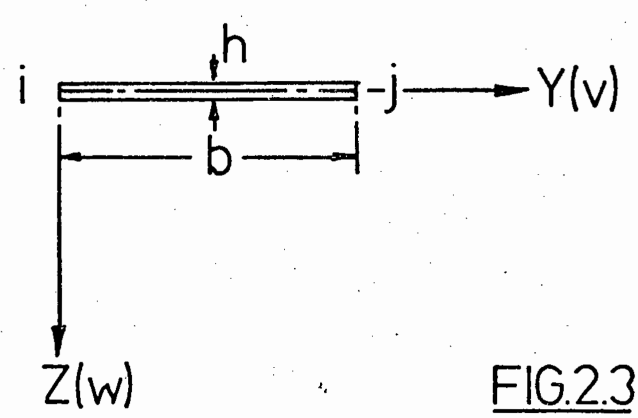
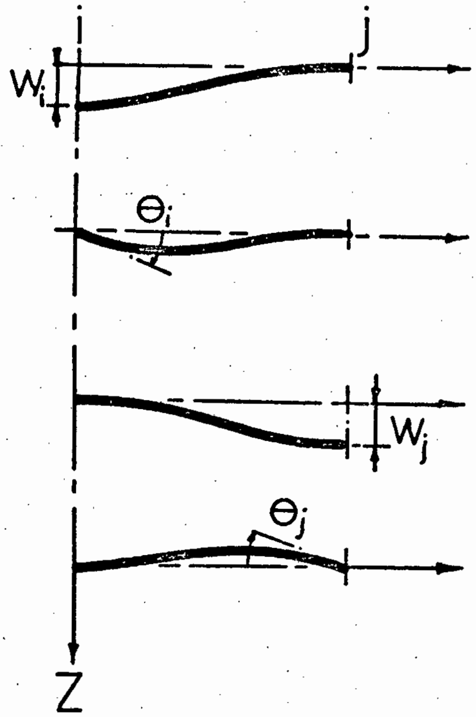


FIG.2.3

All possible displacements of this cross-section which cause bending or twisting stresses can be represented by combinations of the following displacements. The appropriate cubic polynomial displacement functions are also given: (Φ)



$$\Phi^{(1)} = [1 + 2(\tfrac{y}{b})^3 - 3(\tfrac{y}{b})^2]\, w_i$$

$$\Phi^{(2)} = [(\tfrac{y}{b})^3 - 2(\tfrac{y}{b})^2 + (\tfrac{y}{b})]\, b.\theta_i$$

$$\Phi^{(3)} = [3(\tfrac{y}{b})^2 - 2(\tfrac{y}{b})^3]\, w_j$$

$$\Phi^{(4)} = [(\tfrac{y}{b})^3 - (\tfrac{y}{b})^2]\, b.\theta_j$$

Figure 2.4

As with the in-plane displacements, it is convenient to separate the x and y dependence and make the edge displacements functions of x alone. The displacement of any point on the strip surface is then given by the summation of the four displacements. In matrix form:

$$w = C_b . u_b, \tag{2.20}$$

where

$$C_b = \left[(1 - \frac{3y^2}{b^2} + \frac{2y^3}{b^3})(y - \frac{2y^2}{b} + \frac{y^3}{b^2})(\frac{3y^2}{b^2} - \frac{2y^3}{b^3})(-\frac{y^2}{b} + \frac{y^3}{b^2}) \right] \tag{2.21}$$

and

$$u_b = [w_i(x)\ \theta_i(x)\ w_j(x)\ \theta_j(x)]^T \tag{2.22}$$

The curvature displacement relationship for thin plates may be written:

$$\chi = \left[-\frac{\partial^2 w}{\partial x^2} \;\middle|\; -\frac{\partial^2 w}{\partial y^2} \;\middle|\; \frac{2\partial^2 w}{\partial x . \partial y} \right]^T \tag{2.23}$$

and the relationship between moment and curvature may be expressed as

$$M = D\,\chi \tag{2.24}$$

where

$$D = \begin{bmatrix} D_x & D_1 & 0 \\ D_1 & D_y & 0 \\ 0 & 0 & D_{xy} \end{bmatrix} \tag{2.25}$$

and has individual elements:

$$\begin{aligned} D_x &= \frac{E_x h^3}{12(1 - \nu_x \nu_y)} \\ D_y &= \frac{E_y h^3}{12(1 - \nu_x \nu_y)} \\ D_1 &= \frac{\nu_x E_y h^3}{12(1 - \nu_x \nu_y)} \\ D_{xy} &= \frac{Gh^3}{12}. \end{aligned} \tag{2.26}$$

The Total Potential Energy for bending forces acting on a finite strip can be written: (the subscript b indicates bending consideration)

$$\begin{aligned} U_b = {} & \tfrac{1}{2} \int_X \int_Y (M_x . \frac{1}{r_x} + M_y . \frac{1}{r_y} + 2M_{xy} . \frac{1}{r_{xy}}) \, dy \, dx - \int_X \int_Y w.p.dy \, dx \\ & - \int_o^a (M_{xx} . \theta_y + M_{yy} . \theta_x + Q.w) dx \end{aligned} \tag{2.27}$$

Rewriting equation 2.27 with the surface load expression included in the nodal load expression; and using matrix notation (Shear Strains have been neglected)

$$U_b = \tfrac{1}{2} \int_X \int_Y \chi^T . D . \chi \, dy \, dx - \int_X \int_Y C_b . u_b . p.dy \, dx$$

or

$$U_b = \tfrac{1}{2} \int_X \int_Y \chi^T . D . \chi \, dy \, dx - \int_X \int_Y u_b^T . C_b^T . p.dy \, dx \tag{2.28}$$

When use is made of finite differences, loading terms become simple manipulations which are performed when filling out the load matrix. Thus $C_b^T.p.$ can be replaced by a matrix E, which at present requires no further definition.

Expansion of Matrices

As with the "in-plane" matrix expansion, the calculations involved in expanding

(2.28) are both tedious and voluminous. An outline of the process is:-

(a) Multiplication of D and χ.
(b) Evaluation of χ^T.
(c) Pre-multiplying the result from (a) by that from (b).
(d) Integrating with respect to y and evaluating the intergrand from 0 to b.
(e) Differentiating with respect to each displacement present (w_i, θ_i, w_j, θ_j; w_i', θ_i', w_j', θ_j'; w_i'', θ_i'', w_j'', θ_j'').
(f) Grouping the result to give:

$$\delta U_b = \int_X (\delta u''^T[A.u'' + B.u] + \delta u'^T[C.u'] + \delta u^T[B^T.u'' + D.u - E])\,dx \qquad (2.29)$$

The matrices A, B, C and D are given in Appendix B.

Integrating by parts:-

$$\delta U_b = \{\delta u'^T(A.u'' + B.u)\Big|_0^a - [\delta u^T[A.u''' + B.u')\Big|_0^a - \int_X \delta u^T(A.u'''' + B.u'')dx]\}$$
$$+ \{\delta u^T.C.u'\Big|_0^a - \int_X \delta u^T.C.u''.dx\}$$
$$+ \int_X \delta u^T(B^T.u'' + D.u - E)\,dx \qquad (2.30)$$

Grouping (2.30) as products of $\delta u'^T$, and δu^T

$$\delta U_b = \delta u'^T[Au'' + Bu]\Big|_0^a - \delta u^T[Au''' + (B - C)\,u']\Big|_0^a$$
$$+ \int_X \delta u^T[Au'''' + (B - C + B^T)\,u'' + Du - E]\,dx \qquad (2.31)$$

Using the same reasoning as in formulating equations (2.19), to satisfy the requirements that Total Potential Energy be a minimum:-

$$\delta u'^T[A.u'' + B.u]\Big|_0^a = 0$$
$$\delta u^T[A.u''' + (B - C)\,u']\Big|_0^a = 0 \qquad (2.32)$$
$$A.u'''' + (B - C + B^T)\,u'' + D.u - E = 0$$

2.3 COMBINATION OF IN-PLANE AND BENDING FORCES

The in-plane and bending stiffness matrices have been developed separately as there is no connection between the two systems of forces. In the combination of these two sets of forces, matrices are simply enlarged with a separate space for each system. This is shown in Figure 2.4 below.

	u_i v_i	w_i θ_i	u_j v_j	w_j θ_j
u_i v_i	In-Plane	0	In-Plane	0
w_i θ_i	0	Bending	0	Bending
u_j v_j	In-Plane	0	In-Plane	0
w_j θ_j	0	Bending	0	Bending

Figure 2.4

All strip stiffness matrices are now (8 × 8), compared with (4 × 4) previously.

The governing differential equation for a single finite strip is composed of the summation of the last equations of (2.19) and (2.32).

In general terms, this summation can be expressed

$$\alpha D'''' + \beta D''' + \gamma D'' + \varepsilon D' + \zeta D = \lambda \tag{2.33}$$

where:

$$\begin{aligned} \alpha &= A \\ \beta &= 0 \\ \gamma &= (B - C + B^T) - (L) \\ \varepsilon &= (M - M^T) \\ \zeta &= (D + K) \\ \lambda &= (E + J) \quad \text{(Appendix B)} \end{aligned} \tag{2.34}$$

The full α, γ, ε, ζ and λ matrices are simply an addition, as indicated in Figure 2.4, and are given in Appendix C.

With $\beta = 0$, (2.33) becomes:-

$$\alpha D'''' + \gamma D'' + \varepsilon D' + \zeta D = \lambda \tag{2.35}$$

2.4 TRANSFORMATION MATRICES

Equation (2.35) is only suitable for solving flat plate problems. The Transformation Matrix derived below enables folded plate problems to be solved as well.

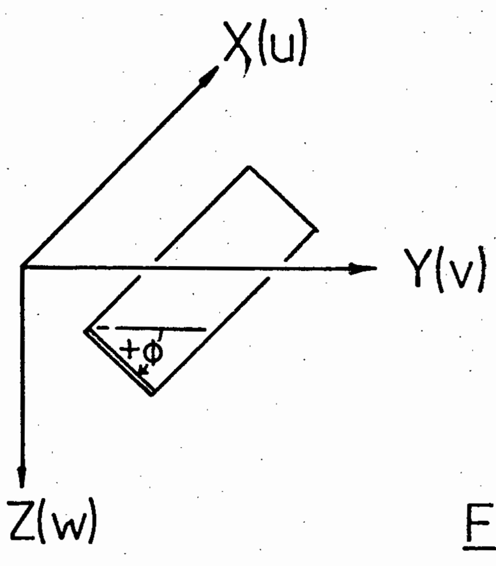


FIG.2.5

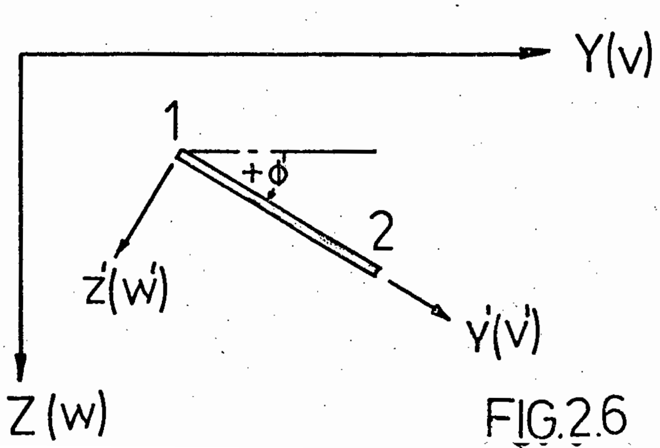


FIG.2.6

STUDENTE AAN OPLEIDINGSKOLE EN SENTRA, INSTITUTE EN KOLLEGES

STUDENTS AT TRAINING SCHOOLS AND CENTRES, INSTITUTES AND COLLEGES

S.31

A. Inrigting / Institution: PENINSULA COLLEGE FOR A.T.E.

B. Adres en poskode / Address and postal code: PRIVATE BAG X, KASSELSVLEI.

C. Kursus en kursusdeel voor ingeskryf / Course and part of course entered for: NATIONAL CERTIFICATE for TECHNICIANS

D. Voorletters en van / Initials and surname — Mnr/Mev/Mej. / Mr/Mrs/Miss: C. P. PERMALL

UITERS BELANGRIK/MOST IMPORTANT

ALLEEN SYFERS WAT SOOS VOLG LYK IS AANVAARBAAR / ONLY NUMERALS LIKE THE FOLLOWING ARE ACCEPTABLE

1 2 3 4 5 6 7 8 9 0

Reeks Nr./Serial No.

711229 — 1a Nommer / Number: — 1b Prov. 1 — 1c Dept. 18 — 31

2a S — 2b D — 3a Volksgroep / Population group: 01 — 9 Vakke / Subjects: A 0982 7 — B — C

3b H/D F/P: 3 — 4 Vandag se datum / Today's date: 19 Y M D — D — E — F

5 Ouderdom / Age: 21 — 6 Geslag / Sex: 1 — G — H — I

7 Aantal weke / Number of weeks: 13 — 8 Aantal ure / Number of hours: 06 — J — K — L

10 Persoonsnommer / Identity Number: 56 M 7 D 1 4 5 1 4 0 0 1 5 — 11 Tipe Kursus / Type of course: 1 — 12 Sert./Dipl. / Cert./Dipl.: 1 9 2

13 Watter deel / Which part: 1 — 14 Soort student / Type of student: 9 — 15 Studierigting / Field of Study: 0906

16 Woonplek / Address: 356 — 17 Loseis / Boarding: 8 — 18 Huistaal / Home Language: 2 — 19 Burgerskap / Citizenship: 1 — 20 Medium van onderrig / Medium of instruction: 2

21 Waarmee besig / How occupied: 22 — 22 Hoogste kwalifikasie (i) / Highest qualification (i): 20 — (ii) Wanneer / When: 4 — 23 Ekonomiese Aktiwiteit / Economic Activity: 1 — 24 Kerkverband / Religious Denomination: 2

The angle ϕ is measured from the main-axis direction to the local-axis direction. This angle is considered positive if clockwise and negative if anti-clockwise. The Y' axis is parallel to the strip and in the direction of increasing node number.

Consider Figure 2.6; the local axis displacements can be derived from global displacements: (Let local axis displacements be indicated by a prime)

$$\begin{aligned} u' &= u \\ v' &= v \cos\phi + w \sin\phi \\ w' &= -v \sin\phi + w \cos\phi \\ \theta' &= \theta. \end{aligned} \qquad (2.36)$$

Or in matrix form:

$$\begin{matrix} u' \\ v' \\ w' \\ \theta' \end{matrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \text{Cos}\,\phi & \text{Sin}\,\phi & 0 \\ 0 & -\sin\phi & \text{Cos}\,\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} u \\ v \\ w \\ \theta \end{bmatrix} = R.D$$

This is the Transformation Matrix for one edge of the strip; for both edges, the matrix is duplicated thus:-

$$T = \left[\begin{array}{c|c} R & 0 \\ \hline 0 & R \end{array}\right] \qquad (2.37)$$

(2.37)

Following the normal transformation procedures, equation (2.35) is written:

$$T^T \alpha\, T\, D'''' + T^T \gamma\, T\, D'' + T^T \varepsilon\, T\, D' + T^T \zeta\, T\, D = \lambda \qquad (2.38)$$

This equation makes possible the solution of folded plate problems. λ is the load matrix where nodal-global loads are substituted. The matrices D'''', D'' and D' have global finite difference approximations to the differential equation along the length of the strip.

2.5 FINITE DIFFERENCE APPROXIMATIONS

The finite difference approximations to the differential equation along a nodal line are derived as for beam elements. All the boundary conditions required in equations (2.19) and (2.32) are satisfied by the use of standard Operator Patterns as given in Appendix D.

The three boundary conditions which occur in this type of structure are; rigidly fixed, simply supported and guided. Each of the four displacements (u, v, w, θ) must be given an operator pattern for each of the above cases.

The displacements v and w become interchanged for a strip on edge, and for any strip which makes an angle to the global co-ordinate system there is the corresponding reciprocation between these two displacements. Thus v and w displacements must always have the same operator patterns. The operator pattern for the θ displacement has the same form as the w displacement.

The u displacement may be either fixed or free. The free operator pattern is derived for a simple rod element having the same values of displacement beyond the boundary as at the boundary node.

The three cases are as follows:-

Rigidly Fixed:	u	Fixed
	v	Fixed
	w	Fixed
	θ	Fixed
Simply Supported:	u	Free
	v	Simply Supported
	w	Simply Supported
	θ	Simply Supported
Guided Support:	u	Fixed
	v	Guided
	w	Guided
	θ	Guided

All nodes at the end of a structure are usually given the same boundary condition.

2.6 EXTENSION OF EQUATION (2.39) FOR COMPUTER USE

Consider the following folded plate structure shown in Figure 2.7.

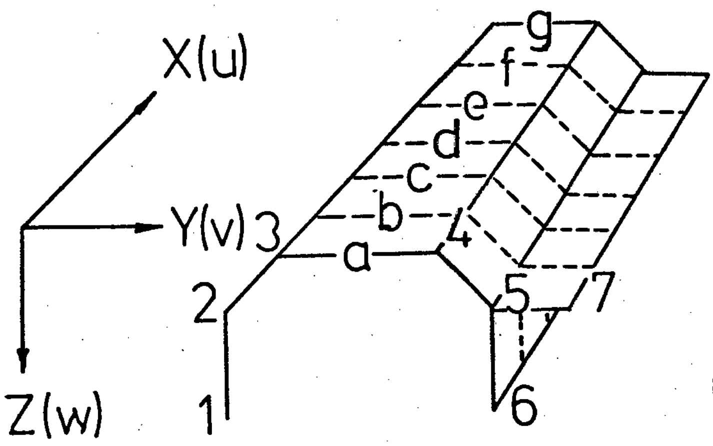


FIG.2.7

Steps in the solution of nodal displacements are:-

(a) Matrices α, γ, ε and ζ in equation (2.38) are evaluated.

(b) The transformation matrices and their transposes are calculated for each strip.

(c) Matrix multiplication yields the following condensed form of equation (2.38) for each strip:-

$$(AL)\ D'''' + (FA)\ D'' + (EP)\ D' + (ZE)\ D = \lambda$$

(d) To compile this equation for the complete cross-section the matrices are added where strip boundaries coincide as shown in Figure 2.8.

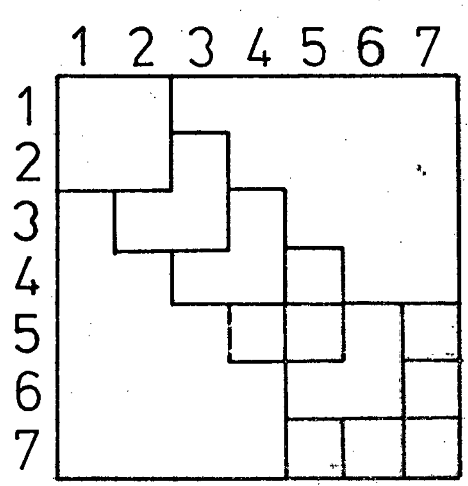


FIG.2.8

Notes:
1. Each small block represents a (4 × 4) matrix; with corresponding u, v, w and θ displacements.
2. The (8 × 8) matrix covering nodes 1 and 2 represents the first strip going from node 1 to 2, the second (8 × 8) block represents the second strip going from node 2 to 3, and so on.
3. The strip going from node 5 to 7 is broken into (4 × 4) matrices and is placed as shown by the shaded squares.

(e) The Finite Difference approximations run along the strip boundaries and always represent global displacements. The total equation to be solved is shown in Figure 2.9.

(f) The combined stiffness matrix is found by the addition of all components in the equation under step (c).

(g) The nodal loads are represented in vector form (λ).

(h) When compiling the stiffness matrix [K], the cross-section matrices are derived for the unit strip length, only a portion of these matrices must be considered. Thus the value S/(Number of cross-sections) must multiply all the matrices (AL, FA, EP and ZE), where S is the nodal spacing.

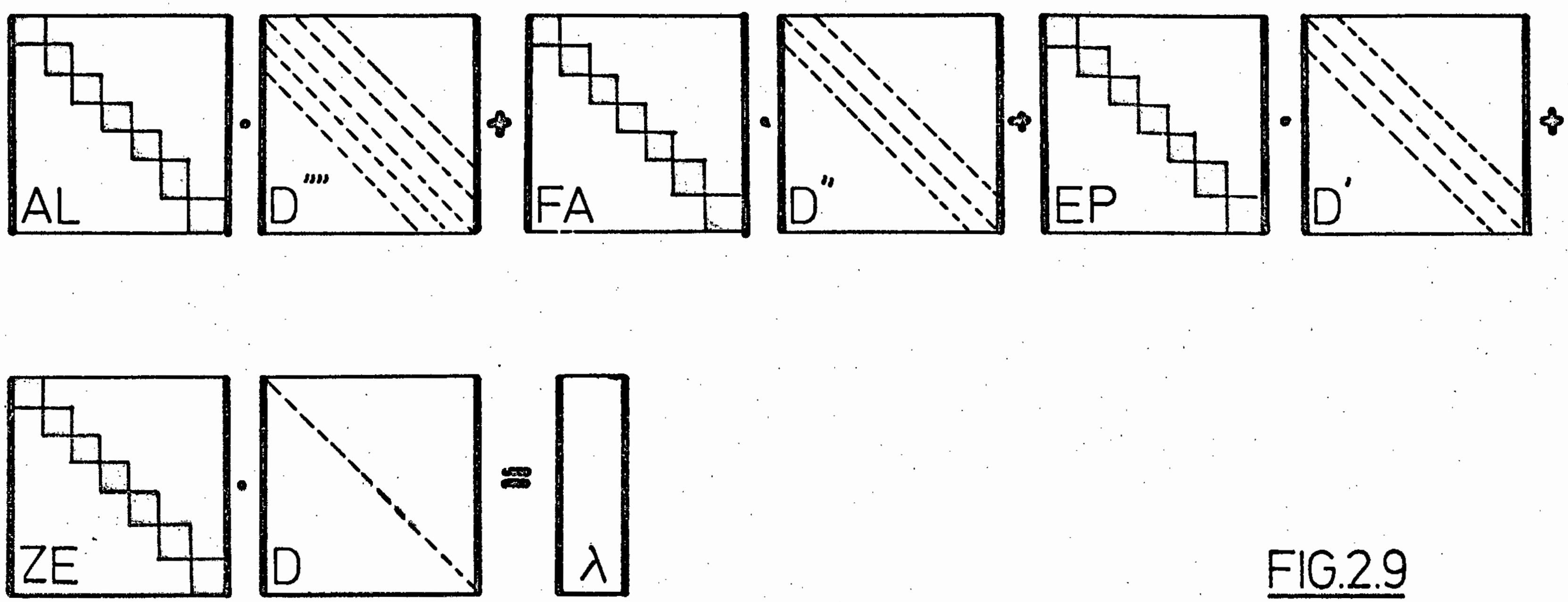


FIG.2.9

<u>Note</u>: 1. Each shaded block in the AL, FA, EP, and ZE matrices are compiled as shown in Fig. 2.8.

2. The Derivative Matrices are sparsely populated and hence are more compactly stored in the computer analysis. The D matrix is a scalar matrix.

3. When matrix arithmetic is done the form of this equation is $[K][D] = [\lambda]$, where K is a symmetric stiffness matrix.

2.7 STEPS IN SOLUTION OF STRESSES AND MOMENTS FROM DISPLACEMENTS

(a) All displacements are transformed into their respective local axis system:-

$$\begin{bmatrix} D_1' \\ D_2' \end{bmatrix} = \left[\begin{array}{c|c} T_{12} & 0 \\ \hline 0 & T_{12} \end{array}\right] \cdot \begin{bmatrix} D_1 \\ D_2 \end{bmatrix} \qquad (2.39)$$

where the strip is envisaged as going from node 1 to node 2.

(b) The stresses and moments are then calculated by the following formulae:-

$$\begin{aligned} \sigma_x' &= d_x \frac{\partial u}{\partial x} + d_{xy} \frac{\partial v}{\partial y} \\ \sigma_y' &= d_{xy} \frac{\partial u}{\partial x} + d_y \frac{\partial v}{\partial y} \\ \tau_{xy}' &= G[\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}] \\ M_x' &= -[D_x \frac{\partial^2 w}{\partial x^2} + D_1 \frac{\partial^2 w}{\partial y^2}] \\ M_y' &= -[D_1 \frac{\partial^2 w}{\partial x^2} + D_y \frac{\partial^2 w}{\partial y^2}] \\ M_{xy}' &= D_{xy}\, 2 \frac{\partial^2 w}{\partial x \partial y} \end{aligned} \qquad (2.40)$$

CHAPTER 3

NUMERICAL EXAMPLES

3.1 Numerical Examples

This chapter contains various numerical examples to illustrate the various displacement, stress and force results as obtained from the computer program.

The results of the first example are compared with those obtained by using conventional beam theory, while the more complicated examples are compared with the results obtained by other researchers in the same field.

EXAMPLE NO. 1:

This example was chosen to illustrate the three different boundary conditions discussed previously, as well as relevant stress and deflection curves.

Ex. 1a

Cross-section : Thick walled box section as shown.
Span : 12,000 m
Boundary conditions : Simply supported - simply supported.
Loading : Uniformly distributed load of 9,96 kN/m.
Young's modulus (X and Y): 31,0E09
Poisson's ratio (X and Y): 0,00
Program C.P.U. time : 1 min 6 sec.

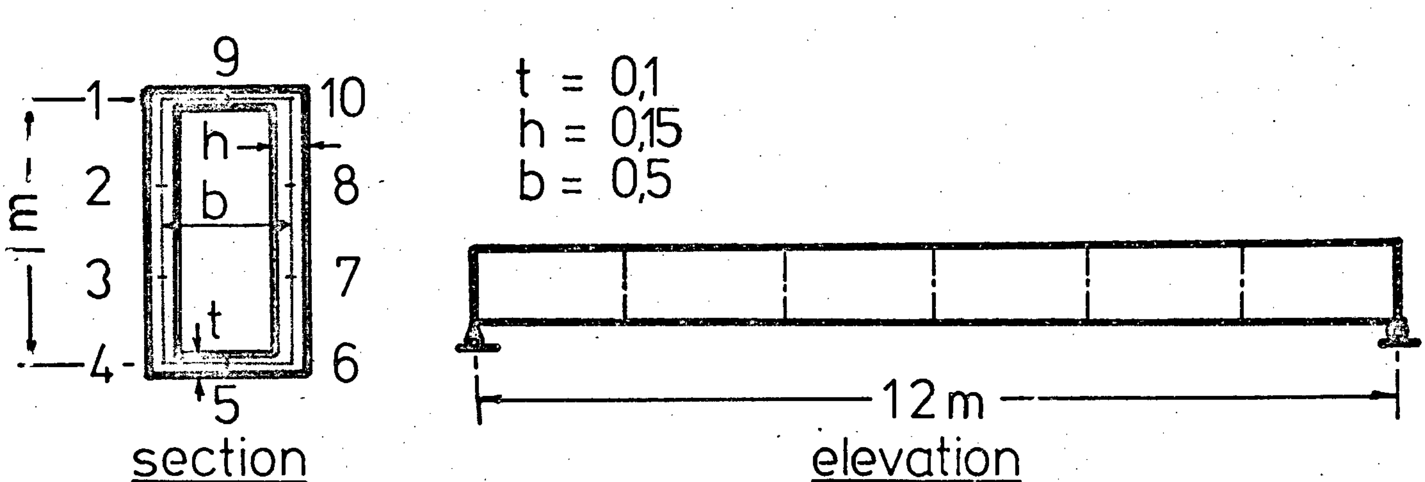


FIG.3.1

Shear lag at beam ends

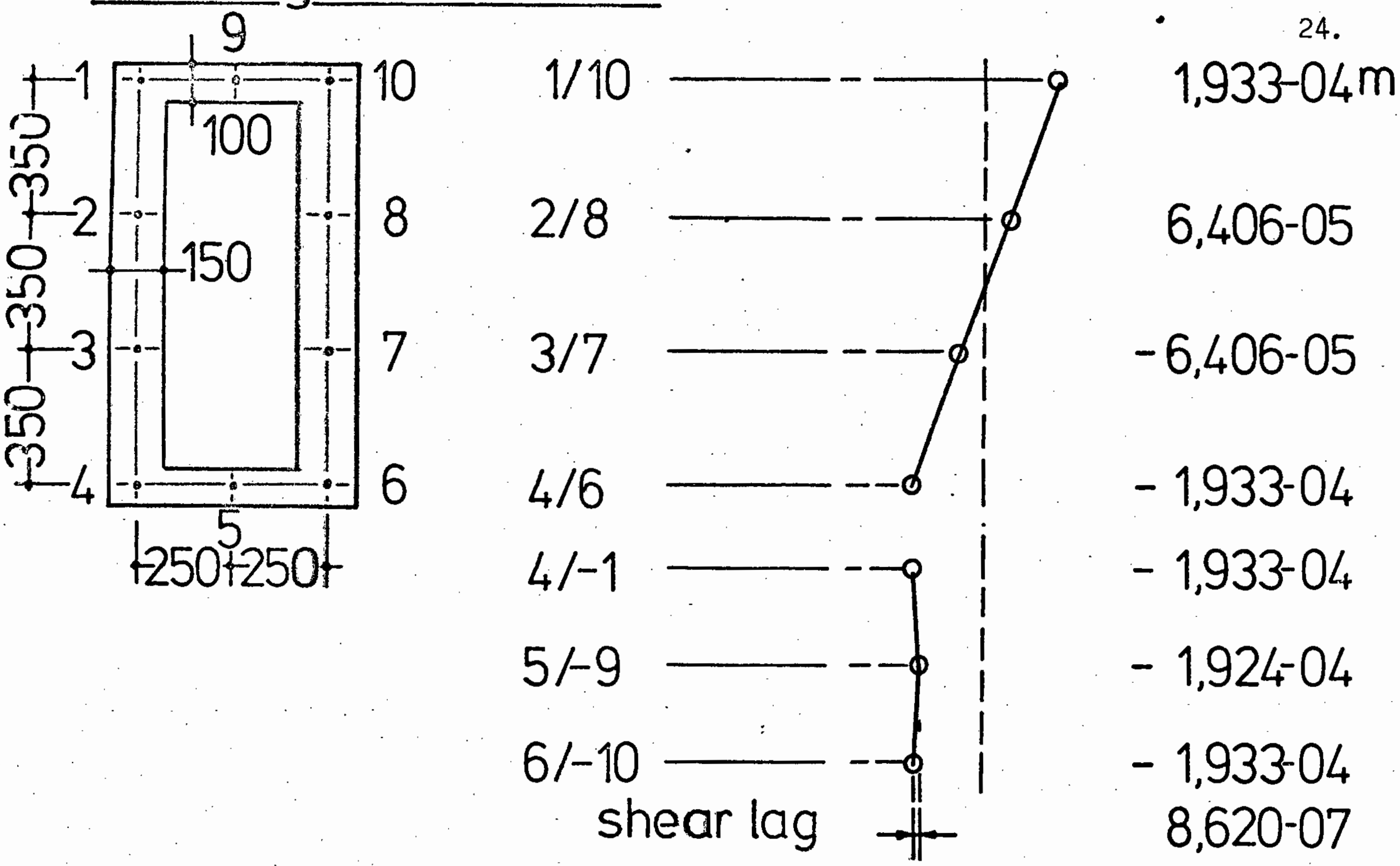


FIG.3.4

Ex. 1b

Cross-section	:	As in example 1a.
Span	:	12,000 m
Boundary conditions	:	Simply supported - Guided.
Loading	:	Uniformly distributed load of 9,96 kN/m.
Young's Modulus (X and Y)	:	31,0E09.
Poisson's Ratio (X and Y)	:	0,00.
Program C.P.U. time	:	58 sec.

Vertical deflection

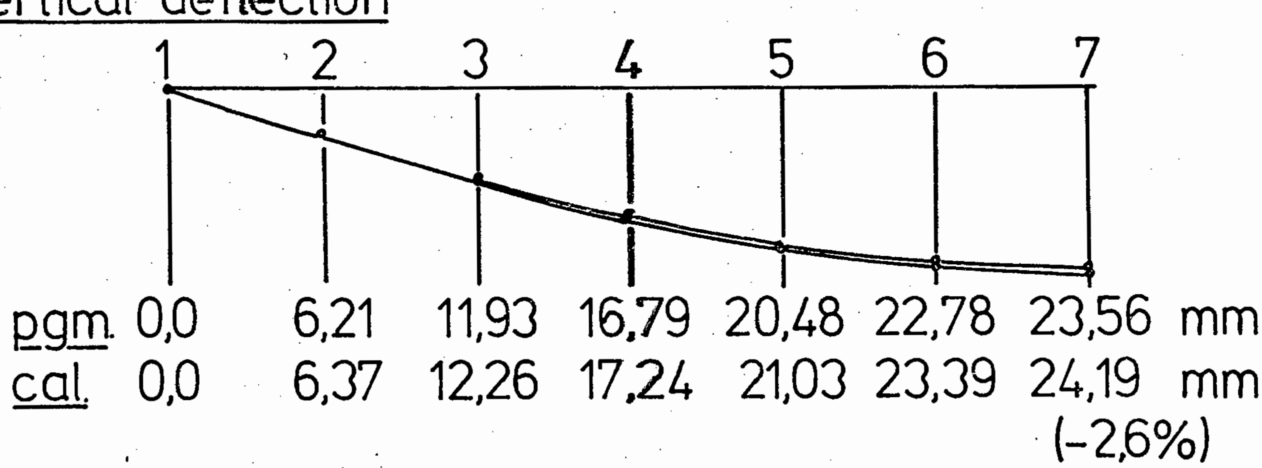
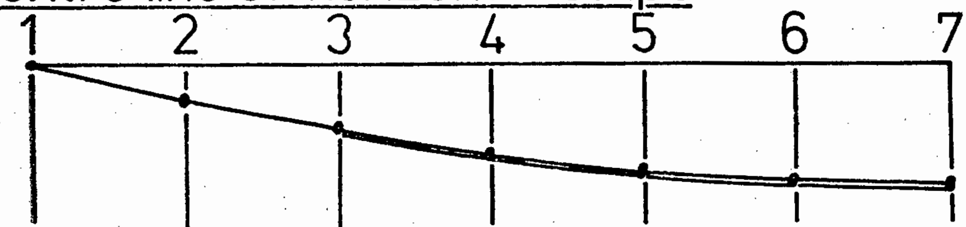


FIG.3.5

σ_x on centre line of horizontal strips



	1	2	3	4	5	6	7	
pgm.	0,08	1,83	3,41	4,64	5,53	6,05	6,42	MPa
cal.	0,00	2,01	3,65	4,92	5,83	6,38	6,56	MPa
							(-2,13%)	

FIG.3.6

Shear stress distribution 2m from l.h. end

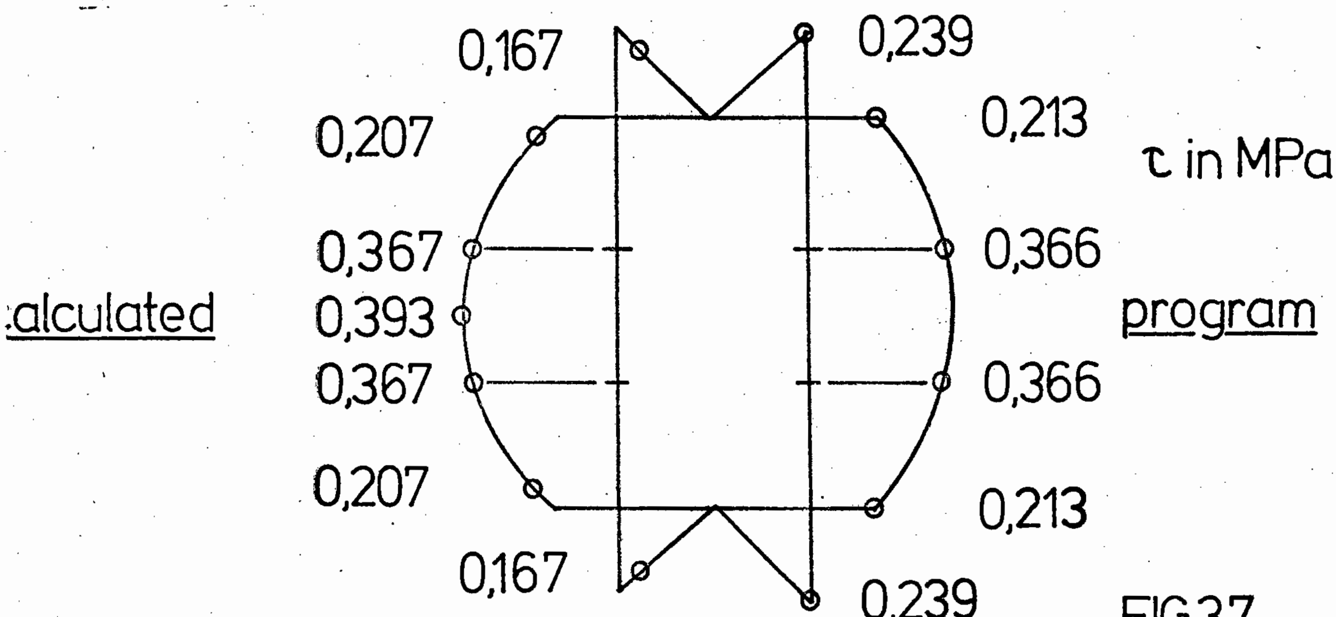
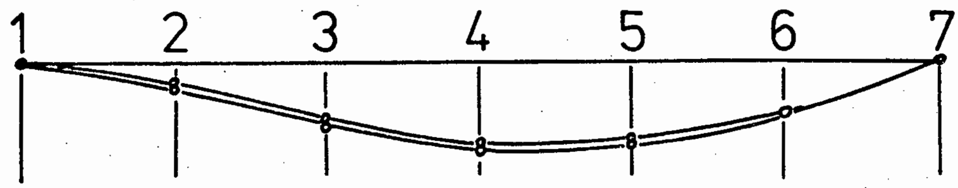


FIG.3.7

Ex. 1c

Cross-section	:	As in example 1a
Span	:	12,000 m.
Boundary conditions	:	Fixed - Simply supported.
Loading	:	Uniformly distributed loading of 9,96 kN/m.
Young's Modulus (X and Y)	:	31,0E09.
Poisson's Ratio (X and Y)	:	0,00
Program C.P.U. time	:	1 min 8 sec.

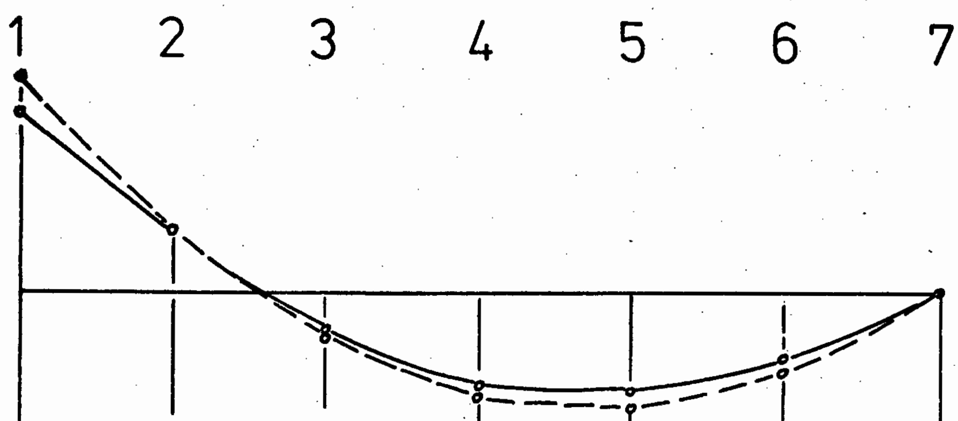
Vertical deflection



pgm.	0,00	0,204	0,491	0,674	0,651	0,404	0,00	mm
cal.	0,00	0,134	0,418	0,642	0,597	0,402	0,00	mm
				(+4,98%)	(+9,05%)			

FIG.3.8

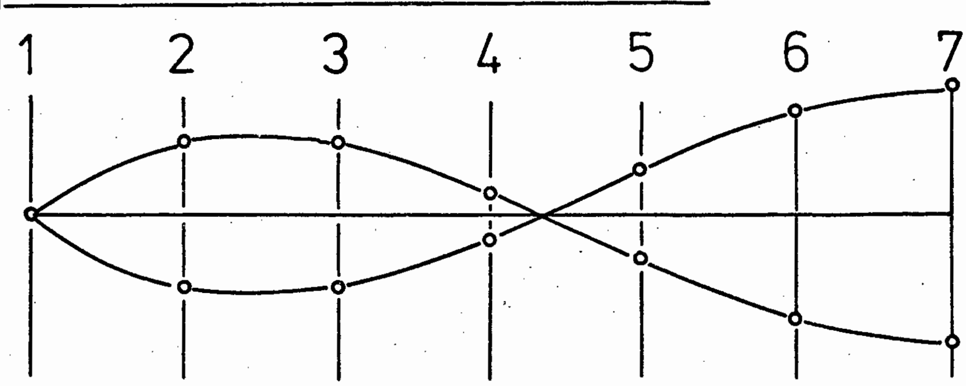
σ_x on centre line of horizontal strips



—pgm.	-1,382	-0,451	0,306	0,721	0,795	0,526	0,085	MPa
--cal.	-1,640	-0,455	0,364	0,820	0,911	0,637	0,000	MPa
				(-12,07%)	(-12,73%)			

FIG.3.9

u displacement curves (x direction)



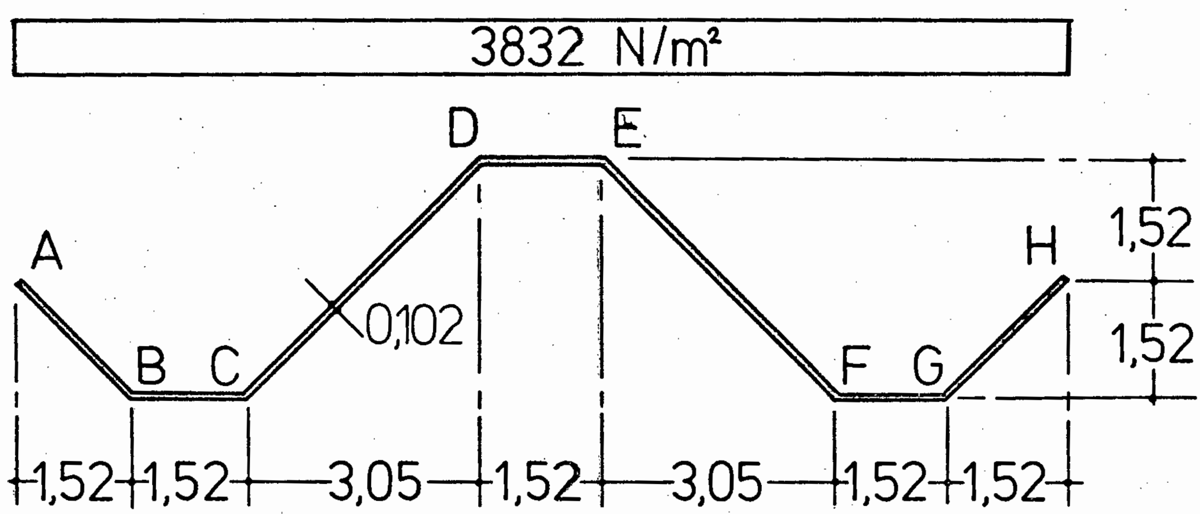
top.	0,00	5,916	5,831	1,961	-3,484	-8,298	-1,027-05	m
bot.	0,00	-5,916	-5831	-1961	3484	8298	1027-05	m

FIG.3.10

EXAMPLE NO. 2:

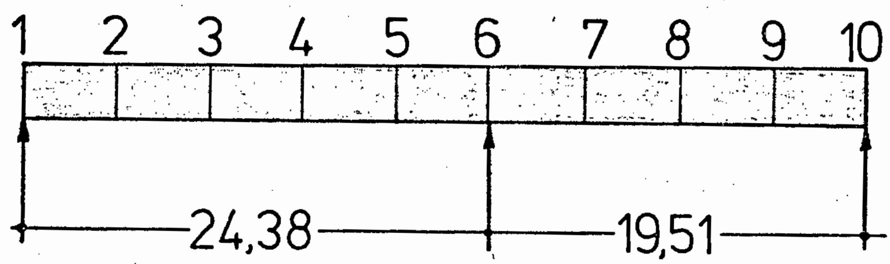
This example is an open cell, thin walled continuous structure. The ability of the theory and program to cater for an interior support is illustrated. Longitudinal stresses and transverse bending moments are compared with other published work.

Reference	:	F.W. Beaufait; Analysis of Continuous Folded Plate Surface, A.S.C.E., Dec. 1965. A.C. Scordelis and K. Lo; Analysis of Continuous Folded Plate Surface, A.S.C.E., June 1966.
Cross-section:	:	As shown in Fig. 3.11.
Span	:	24,38 m and 19,507.m.
Boundary conditions	:	Simply supported - Simply supported with interior support.
Loading	:	Uniformly distributed load of 3832 N/m^2.
Young's modulus (X and Y)	:	2,52443 E 10 N/m^2.
Poisson's ratio (X and Y)	:	0,00.
Program C.P.U. time	:	2 min 21 sec.



Cross-section

FIG.3.11



Elevation

FIG.3.12

Section AB has 2 strips

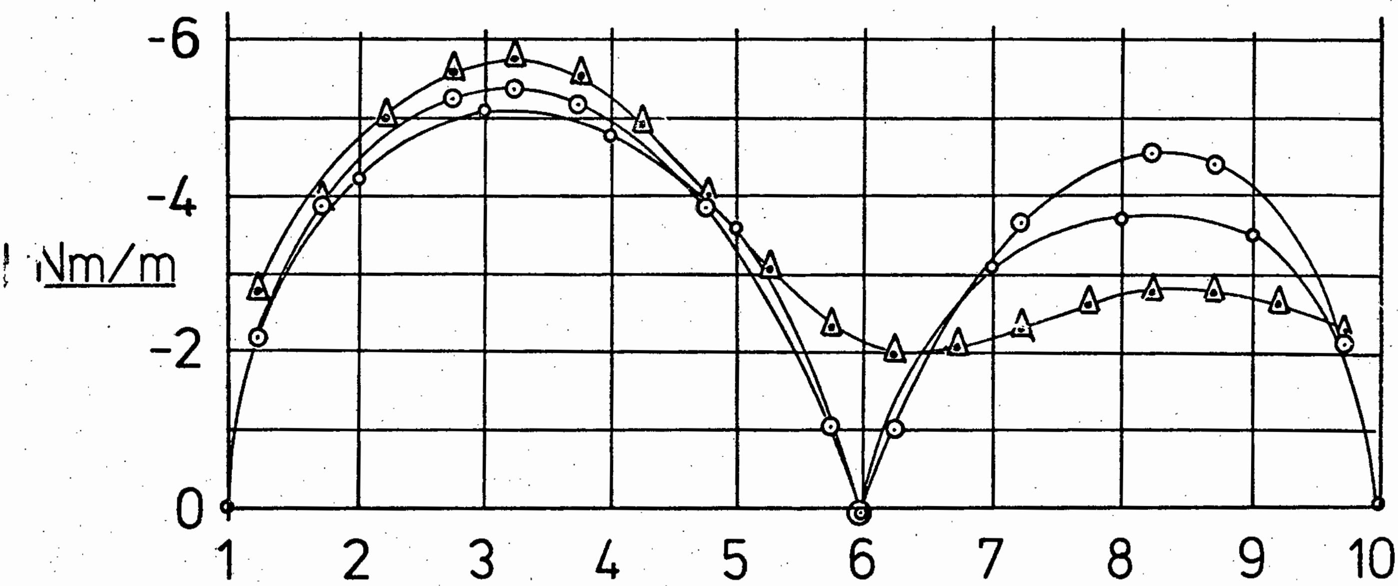
Section BC has 2 strips

Section CD has 4 strips

Section D to centre line has 2 strips

Only half the cross-section was analysed, the rotations and lateral displacements at the node on the centre line were zeroed.

Longitudinal distribution of transverse moment on fold line C



∘ Program ∆ Beaufait ⊙ Scordelis & Lo

FIG.3.13.

Transverse bending moment at section 3

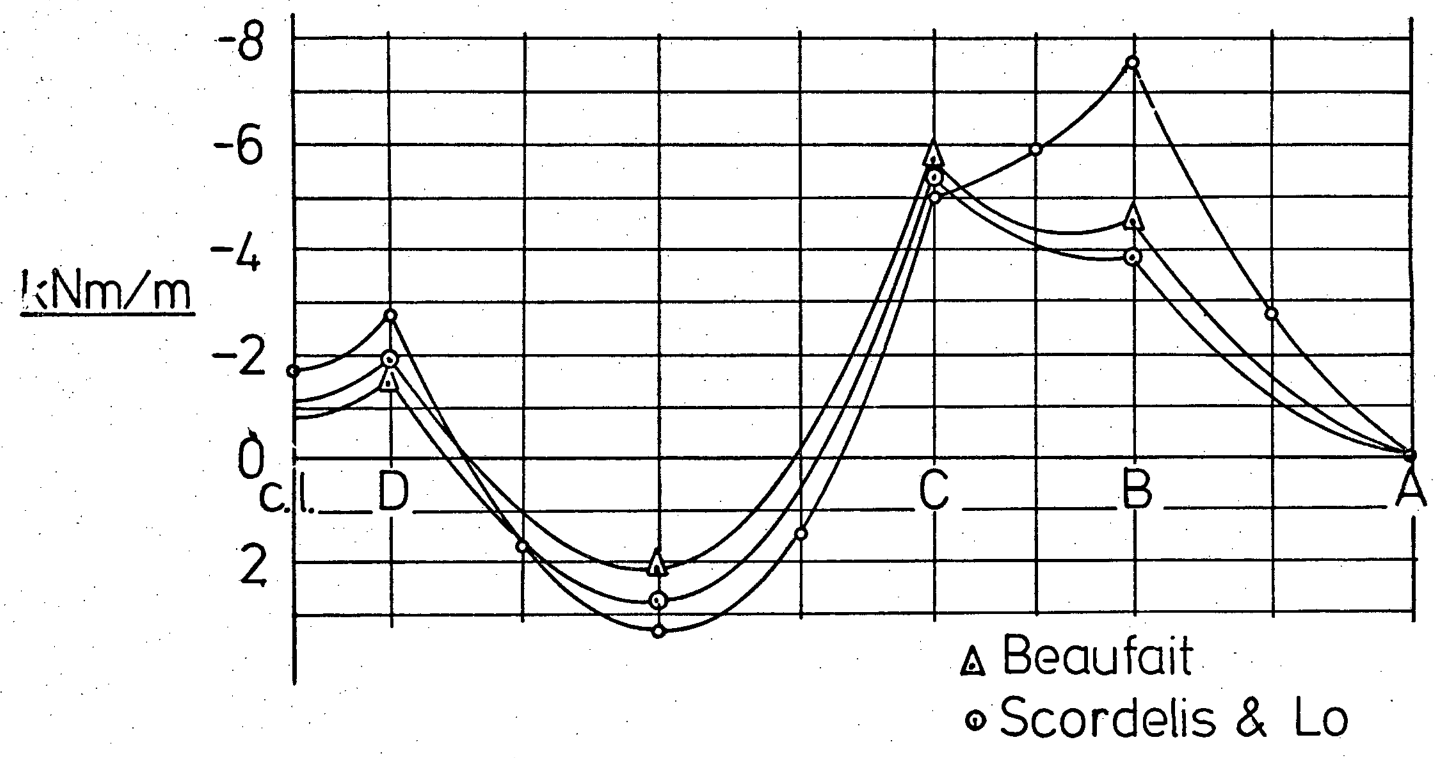


FIG 3.14.

σ_x on foldline D

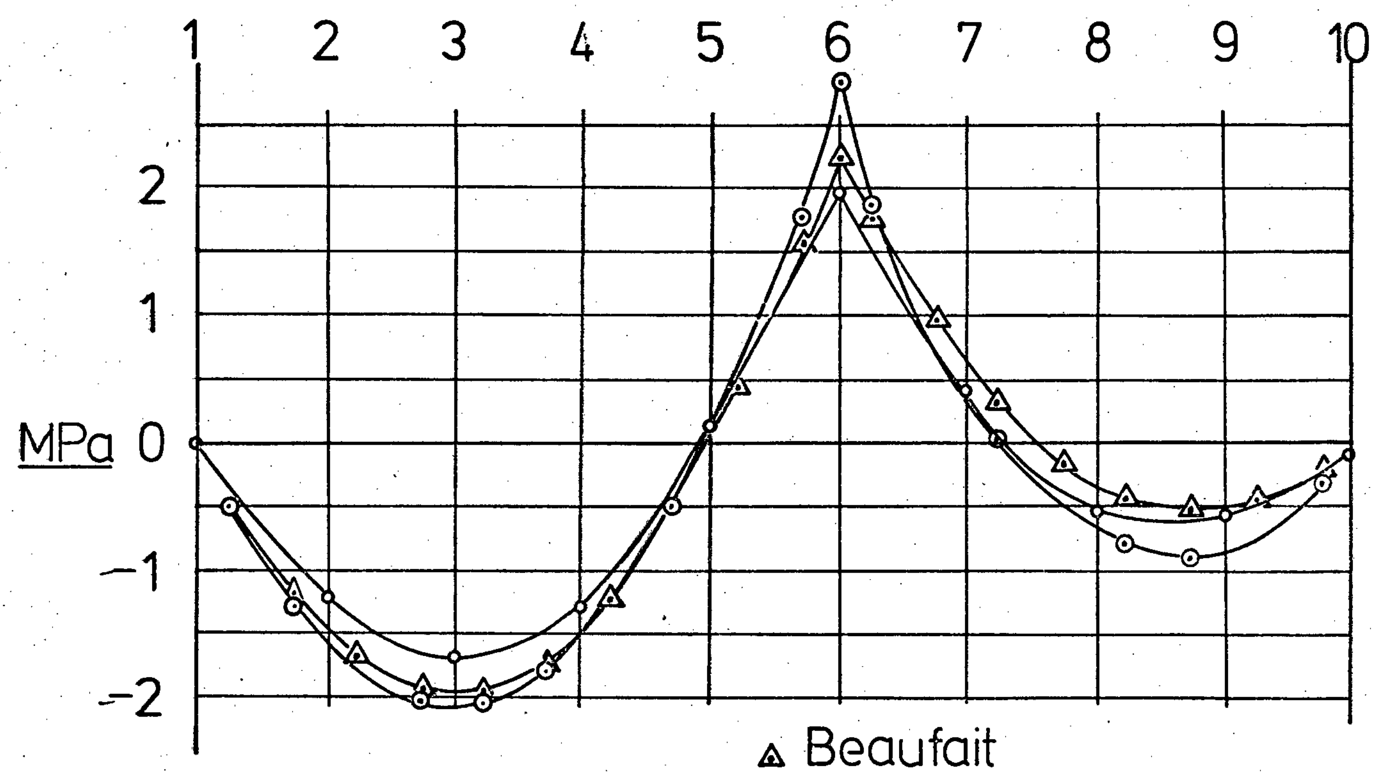


FIG 3.15.

Transverse distribution of longitudinal stress

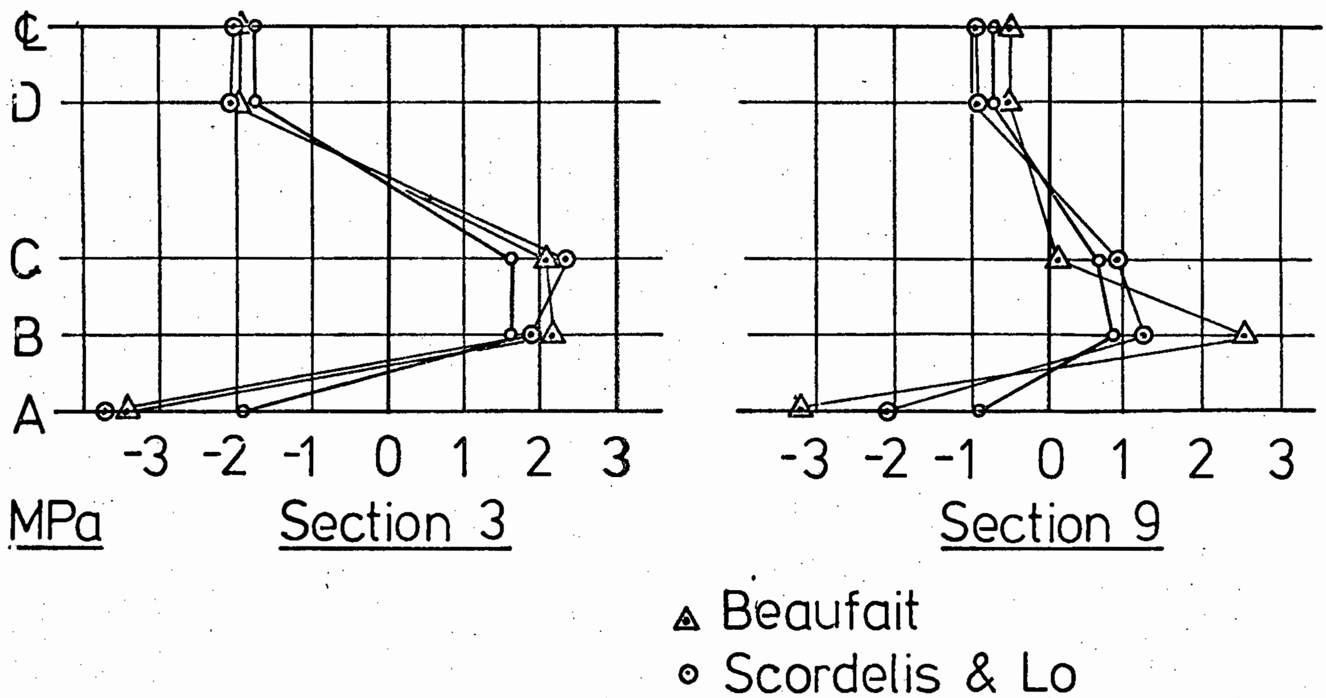
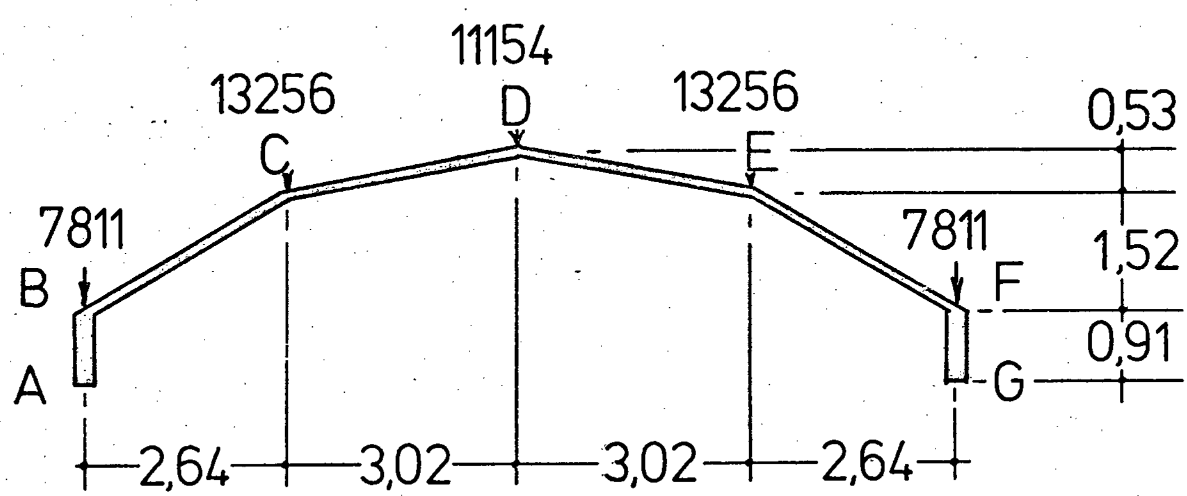


FIG.3.16

EXAMPLE NO. 3:

The structure shown is an open cell, thin walled translational shell which has members of two different thickneses.

Reference	:	A. de Fries-Skene and A.C. Scordelis; Direct Stiffness Solution for Folded Plates, A.S.C.E., August, 1964. Y.K. Cheung; Folded Plate Structures by Finite Strip Method, A.S.C.E., Dec. 1969.
Cross-section	:	As shown in Fig. 3.
Span	:	30,480 m
Boundary conditions	:	Simply supported - Simply supported.
Loading	:	Ridge line loading as shown in Fig. 3.17.
Young's modulus (X and Y)	:	13,795 E0 9 N/m^2.
Poisson's ratio (X and Y)	:	0,00.
Program C.P.U. time	:	2 min 38 sec.



Cross-section

FIG.3.17

All plate elements were divided into 2 strips each. The C.P.U. time given is for the analysis of the complete structure. The program, however, is quite capable of analysing one quarter of such "two way symmetrical" structures. A time of 1 min 29 sec was recorded for such an analysis.

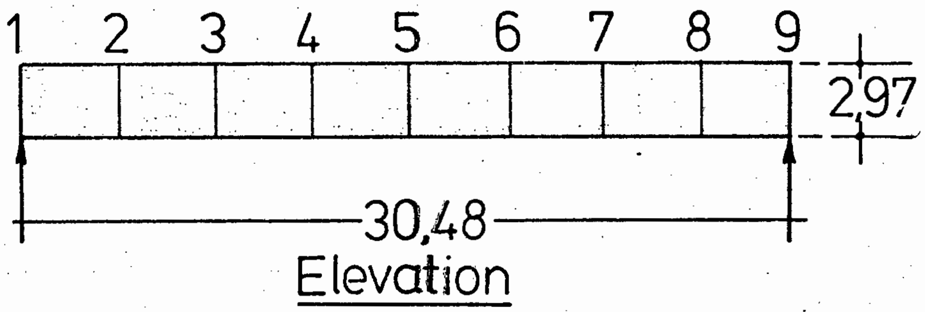


FIG.3.18

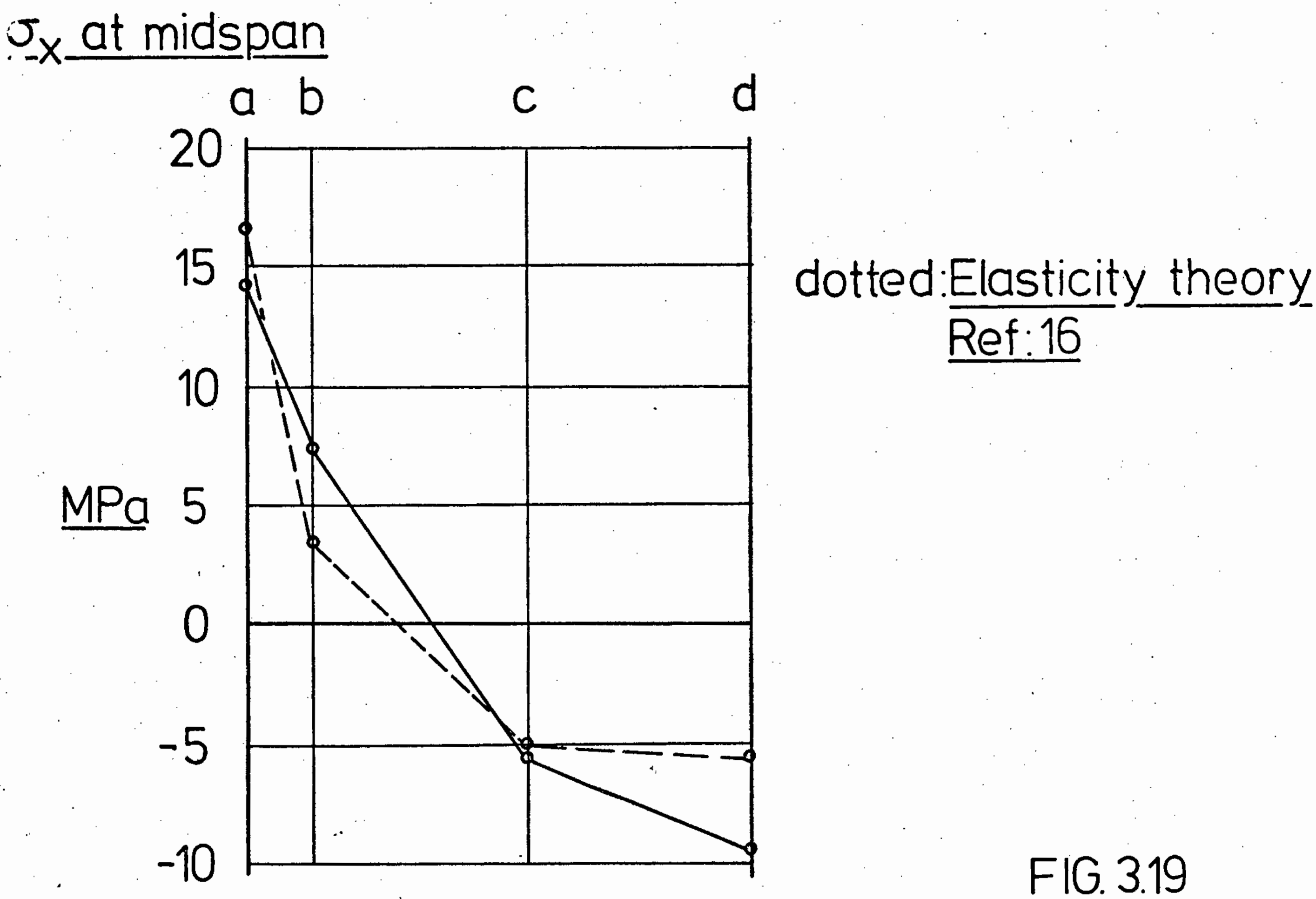


FIG. 3.19

Transverse moment at midspan

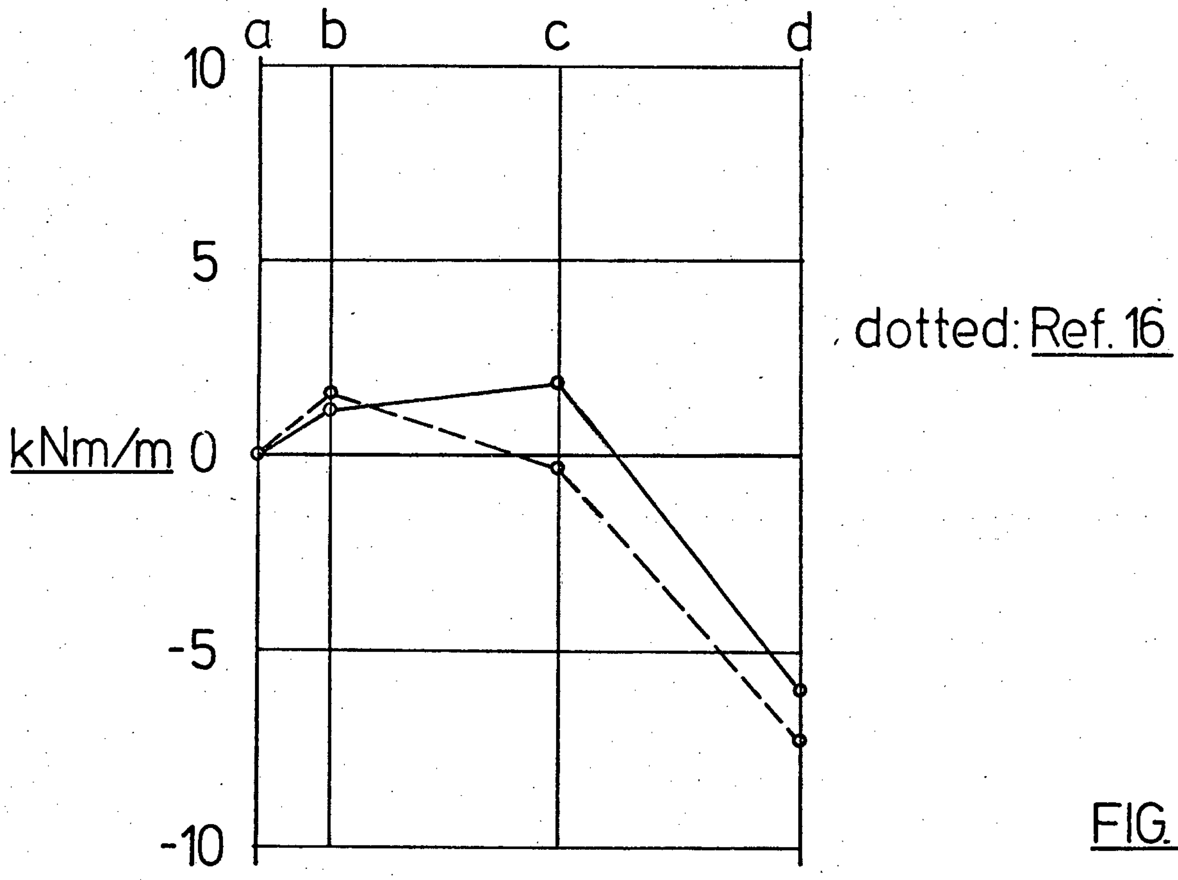


FIG. 3.20

Plate membrane shear at support

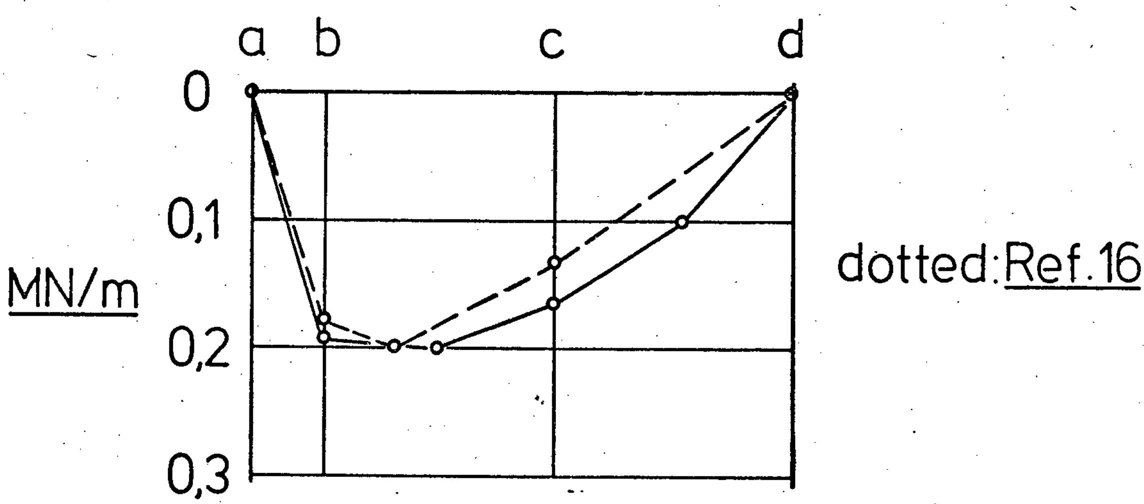


FIG. 3.21

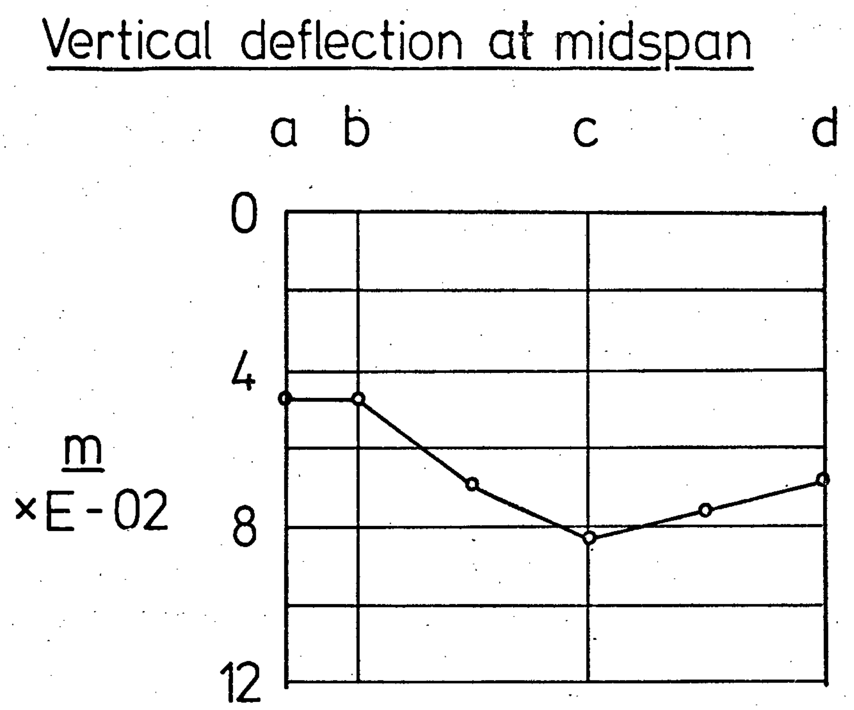


FIG.3.22

3.2 Discussion of Results

In example 1, the box beam webs and flanges were kept relatively thick so as to reduce the effect of secondary stresses and so make the comparison with simple beam theory realistic.

In all cases the deflection and bending stress curves were within acceptable limits. Fig. 3.4 shows that the method of analysis is capable of detecting shear lag, which is important in the analysis of thin walled box structures. Fig. 3.7 shows shear stresses in the beam near the supports. The computer analysis overestimates these stresses at the ends of the structure and it was found that extrapolating the results for nodes near the ends gives acceptable answers. Fig. 3.10 shows the form of the in-plane u displacement for the propped cantilever case. It can be seen that the slope of this displacement with respect to the longitudinal axis is zero at the right hand support, fulfilling the requirement of zero stress at the boundary ($\sigma_x \doteq d_x \, \partial u/\partial x$).

Example 2 was compared with published work. It shows that the method is quite capable of dealing with internal supports and gives reasonably accurate results. The values, as plotted, indicate that there is a fairly large range between results obtained by Beaufait and those obtained by Scordelis and Lo.

Fewer nodes were used in the longitudinal direction than in the two references and thus it was not possible to compare results at exactly the same cross-sections. The number of cross-sections was limited by practical considerations and the maximum array size handled by the computer.

Fig. 3.13 shows that Beaufait's solution has a relatively high transverse moment over the internal support. (Cross-section 6). The exact nature of the support is not described in the literature, but Scordelis gives a very small bending moment at this section, indicating a diaphragm type support. The present analysis also makes use of a diaphragm type support, and is capable of simulating any type of internal fixity.

Fig. 3.14 shows the transverse bending moment at Section 3. The results agree reasonably well except at fold line B, where the bending moment is substantially more but still within acceptable design limits.

Example 3 shows essentially the same variables as example 2. The accuracy of the results, compared with published work, is also approximately the same.

CHAPTER 4

CONCLUSIONS

This chapter contains general conclusions in respect of the method of analysis and its applications. The main points of interest which arose with the application of finite differences are discussed, as well as computer implications and advantages over the more usual Fourier Series Method are listed.

4.1 Finite Difference Operator Patterns

The patterns employed in the computer program are given in Appendix D. The following facts governed the choice of patterns and the resulting accuracy.

(a) The half-band width of the stiffness matrix is three times the number of degrees of freedom per cross-section. This is because five point central difference operator patterns were used for the fourth derivative, of which only the central and right hand values were actually used in the analysis. Thus these three values, which determine the stiffness matrix band-width, are the minimum required for the fourth derivative.

(b) The various strip stiffness matrices and their multipliers are as follows:-

Matrix	Symmetry	Multiplier	Symmetry
α	Symmetrical	D""	Symmetrical
γ	Symmetrical	D"	Symmetrical
ε	Skew-Symmetrical	D'	Skew-Symmetrical
ζ	Symmetrical	D	Symmetrical

It is noticed that every matrix that is symmetrical must be multiplied by a symmetrical multiplier and vice-versa. This dictates that no odd pattern may be inserted at the end of the structure to increase accuracy, as it would not be compatible with the rest of the multiplier matrix. The maximum number of values that are actually used in these multiplier matrices are limited to three, as more would give values outside the already rather broad band-width.

(c) The values in the scalar matrix (D) may only be different from 1.0 at the extreme ends of the structure. This is because boundary conditions need only be satisfied at the ends of the structure.

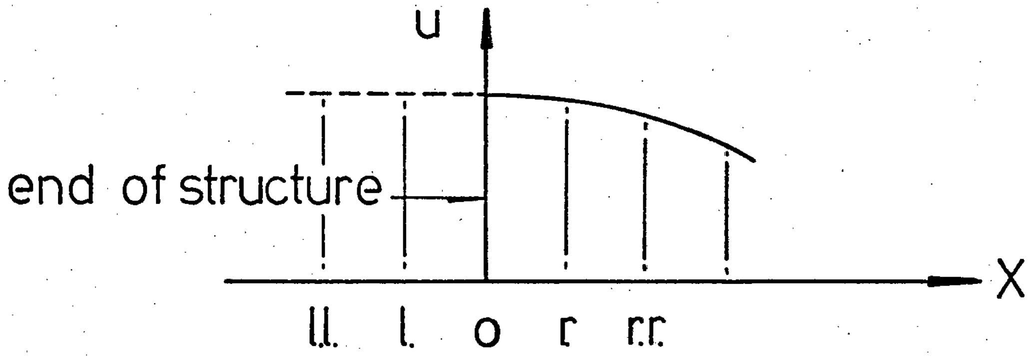
(d) The in-plane u displacement boundary condition is satisfied by using the fact that the longitudinal stress is zero at the free edge. The longitudinal stress is given by:

$$\sigma_x = d_x \frac{\partial u}{\partial x} + d_{xy} \frac{\partial v}{\partial y}.$$

Since the second term makes only a small contribution, and none at all when $\nu = 0$, the stress may be approximated by:

$$\sigma_x \doteqdot d_x \frac{\partial u}{\partial x}.$$

At the free edge, $\sigma_x = 0$, which implies that $\partial u/\partial x$ must be zero. This is satisfied by the following displacement model;



(e) The final stresses are calculated from the displacements obtained, and it was found that small differences in corresponding displacements resulted in unacceptably large differences in stresses. Thus it was absolutely necessary to use operator patterns which gave very good symmetrical displacements.

(f) External reactions and other forces not given in the computer analysis can be readily calculated using the correct finite difference operator pattern together with the displacements given.

4.2 Computer Program Implications

(a) The computer program, as written, uses a large amount of core storage because of the necessarily broad band width of the stiffness matrix. However, the solution time compares very favourably with those given in the references listed. (This may be because the computer used, UNIVAC 1106, is a relatively high speed machine).

(b) A small amount of core storage is required for the CA, CG, CE and CZ matrices, and the finite difference values are stored as efficiently as possible. This method can hence be used on a fairly small core machine if the formation of the stiffness matrix is altered so that blocks of figures are transferred to disc storage and are later recalled as required.

(c) Tapering members can be given an equivalent thickness which corresponds to an identical second moment of area.

4.3 Advantages over Fourier Series Method

The method of using finite differences together with the normal finite strip concept has distinct advantages over the use of Fourier Series. These are:

(a) The global displacements are solved directly, as compared with the Fourier Series Method where substantially smaller calculations are done repeatedly and added, giving the required displacements. Up to seventy, and more, harmonics are used in the papers referred to.

(b) Point loadings are distributed over one nodal length, this will give more accurate answers than using a number of Fourier Series harmonics.

(c) Different boundary conditions are employed without any difficulty and are relatively easily understood.

(d) Interior supports along the length of a structure may be as numerous as desired. Fourier Series becomes impractical for more than three spans.

(e) This method may also be used for strips of non-uniform thickness in the longitudinal direction, i.e. cut-outs and the change of thickness of members over supports. This simply leads to the necessity of calculating and storing CA, CG, CE and CZ matrices for as many different cross-sections as are required.

BIBLIOGRAPHY

1. A.S.C.E., Report of the Task Committee on Folded Plate Construction. Dec. 1963.

2. F.W. Beaufait, "Analysis of Continuous Folded Plate Surfaces", A.S.C.E. Struct. Dec. 1965.

3. A.C. Scordelis and Kam-Shing Lo, "Analysis of Continuous Folded Plate Surfaces", Discussion A.S.C.E. Struct., June 1966.

4. Y.K. Cheung, "Folded Plate Structures by the Finite Strip Method", A.S.C.E., Dec. 1969.

5. P.V. Thangam Babu and D.V. Reddy, "Frequency Analysis of Orthotropic Circular Cylindrical Panels by the Finite Strip Method", Build. Sci., Vol. 8., 1973.

6. T.G. Brown and A. Ghali, "Semi-Analytic Solution of Skew Box Girder Bridges", Proc. I.C.E., Part 2, Sept. 1975.

7. S.A.I.C.E. and C.S.I.R., "Symposium on Bridge Deck Analysis", Kyalami Ranch, May 1973.

8. J.G. van H. Louw, "Finite Strip Analysis of Flat Plates", M.Sc.(Eng.) Thesis, U.C.T., 1975.

9. S.P. Timoshenko and J.N. Goodier, "Theory of Elasticity", 3rd Edition McGraw-Hill, 1970.

10. K. Washizu, "Variational Methods in Elasticity and Plasticity", Pergamon Press, 1968.

11. Prof. P.S. Symonds, "Structural Dynamics", Postgraduate Course given at U.C.T., 1975.

12. Prof. M.O. de Kock, "Skeletal Structures", Postgraduate Course given at U.C.T., 1975.

13. H.C. Martin, "Introduction to Finite Element Analysis", McGraw Hill, March, 1973.

14. A. Ghali and A.M. Neville, "Structural Analysis", Intext 1972.

15. Prof. A.D.W. Sparks, "Introduction to Numerical Methods", Course given at U.C.T., 1967.

16. A. de Fries-Skene and A.C. Scordelis, "Direct Stiffness Solution for Folded Plates", A.S.C.E., Struct., Aug. 1964.

17. B.I. Maisel and F. Roll, "Methods of Analysis and Design of Concrete Boxbeams with side Cantilevers", C. and C.A. (U.K.), Technical Report, Nov. 1974.

19. Y.K. Cheung, "Finite Strip Method of Analysis of Elastic Slabs", A.S.C.E. Struct., Dec. 1968.

APPENDIX A:

STIFFNESS SUB-MATRICES (In-Plane Forces)

$$[K] = \begin{bmatrix} G\frac{h}{b} & 0 & -G\frac{h}{b} & 0 \\ 0 & d_y\frac{h}{b} & 0 & -d_y\frac{h}{b} \\ -G\frac{h}{b} & 0 & G\frac{h}{b} & 0 \\ 0 & -d_y\frac{h}{b} & 0 & d_y\frac{h}{b} \end{bmatrix}$$

$$[L] = \begin{bmatrix} d_x\frac{bh}{3} & 0 & d_x\frac{bh}{6} & 0 \\ 0 & G\frac{bh}{3} & 0 & G\frac{bh}{6} \\ d_x\frac{bh}{6} & 0 & d_x\frac{bh}{3} & 0 \\ 0 & G\frac{bh}{6} & 0 & G\frac{bh}{3} \end{bmatrix}$$

$$[M] = \begin{bmatrix} 0 & -G\frac{h}{2} & 0 & -G\frac{h}{2} \\ -d_{xy}\frac{h}{2} & 0 & -d_{xy}\frac{h}{2} & 0 \\ 0 & G\frac{h}{2} & 0 & G\frac{h}{2} \\ d_{xy}\frac{h}{2} & 0 & d_{xy}\frac{h}{2} & 0 \end{bmatrix}$$

APPENDIX B:

STIFFNESS SUB-MATRICES (Bending Forces)

$$[A] = \begin{bmatrix} \frac{13b\,D_x}{35} & \frac{11b^2D_x}{210} & \frac{9b\,D_x}{70} & -\frac{13b^2D_x}{420} \\ \frac{11b^2D_x}{210} & \frac{b^3D_x}{105} & \frac{13b^2D_x}{420} & -\frac{b^3D_x}{140} \\ \frac{9b\,D_x}{70} & \frac{13b^2D_x}{420} & \frac{13b\,D_x}{35} & -\frac{11b^2D_x}{210} \\ -\frac{13b^2D_x}{420} & -\frac{b^3D_x}{140} & -\frac{11b^2D_x}{210} & \frac{b^3D_x}{105} \end{bmatrix}$$

$$[B] = \begin{bmatrix} -\frac{6D_1}{5b} & -\frac{11D_1}{10} & \frac{6D_1}{5b} & -\frac{D_1}{10} \\ -\frac{D_1}{10} & -\frac{2b\,D_1}{15} & \frac{D_1}{10} & \frac{b\,D_1}{30} \\ \frac{6D_1}{5b} & \frac{D_1}{10} & -\frac{6D_1}{5b} & \frac{11D_1}{10} \\ -\frac{D_1}{10} & \frac{b\,D_1}{30} & \frac{D_1}{10} & -\frac{2b\,D_1}{15} \end{bmatrix}$$

$$[C] = \begin{bmatrix} \frac{24D_{xy}}{5b} & \frac{4D_{xy}}{10} & -\frac{24D_{xy}}{5b} & \frac{4D_{xy}}{10} \\ \frac{4D_{xy}}{10} & \frac{8bD_{xy}}{15} & -\frac{4D_{xy}}{10} & -\frac{4bD_{xy}}{30} \\ -\frac{24D_{xy}}{5b} & -\frac{4D_{xy}}{10} & \frac{24D_{xy}}{5b} & -\frac{4D_{xy}}{10} \\ \frac{4D_{xy}}{10} & -\frac{4bD_{xy}}{30} & -\frac{4D_{xy}}{10} & \frac{8bD_{xy}}{15} \end{bmatrix}$$

$$[D] = \begin{bmatrix} \frac{12D_y}{b^3} & \frac{6D_y}{b^2} & -\frac{12D_y}{b^3} & \frac{6D_y}{b^2} \\ \frac{6D_y}{b^2} & \frac{4D_y}{b} & -\frac{6D_y}{b^2} & \frac{2D_y}{b} \\ -\frac{12D_y}{b^3} & -\frac{6D_y}{b^2} & \frac{12D_y}{b^3} & -\frac{6D_y}{b^2} \\ \frac{6D_y}{b^2} & \frac{2D_y}{b} & -\frac{6D_y}{b^2} & \frac{4D_y}{b} \end{bmatrix}$$

$$[\lambda] = \begin{bmatrix} N \\ N/m \\ N/m \\ Nm/m \end{bmatrix} \begin{matrix} u & \text{(Force)} \\ v & \text{(Force per unit length)} \\ w & \text{(Force per unit length)} \\ \theta & \text{(Moment per unit length)} \end{matrix}$$

APPENDIX C:

α, γ, ε and ζ MATRICES

THE α MATRIX

$$
\left[\begin{array}{cccc|cccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{13b\,D_x}{35} & \frac{11b^2D_x}{210} & 0 & 0 & \frac{9b\,D_x}{70} & -\frac{13b^2D_x}{420} \\
0 & 0 & \frac{11b^2D_x}{210} & \frac{b^3D_x}{105} & 0 & 0 & \frac{13b^2D_x}{420} & -\frac{b^3D_x}{140} \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{9b\,D_x}{70} & \frac{13b^2D_x}{420} & 0 & 0 & \frac{13b\,D_x}{35} & -\frac{11b^2D_x}{210} \\
0 & 0 & -\frac{13b^2D_x}{420} & -\frac{b^3D_x}{140} & 0 & 0 & -\frac{11b^2D_x}{210} & \frac{b^3D_x}{105}
\end{array}\right]
$$

THE γ MATRIX

$-\frac{h\,d_x\,b}{3}$	0	0	0	$-\frac{h\,d_x\,b}{6}$	0	0	0
0	$-\frac{h\,G\,b}{3}$	0	0	0	$-\frac{h\,G\,b}{6}$	0	0
0	0	$-\frac{6D_1}{5b} - \frac{24D_{xy}}{5b}$ $-\frac{6D_1}{5b}$	$-\frac{11D_1}{10} - \frac{4D_{xy}}{10}$ $-\frac{D_1}{10}$	0	0	$\frac{6D_1}{5b} + \frac{24D_{xy}}{5b}$ $+\frac{6D_1}{5b}$	$-\frac{D_1}{10} - \frac{4D_{xy}}{10}$ $-\frac{D_1}{10}$
0	0	$-\frac{D_1}{10} - \frac{4D_{xy}}{10}$ $-\frac{11D_1}{10}$	$-\frac{2bD_1}{15} - \frac{8bD_{xy}}{15}$ $-\frac{2bD_1}{15}$	0	0	$\frac{D_1}{10} + \frac{4D_{xy}}{10}$ $\frac{D_1}{10}$	$\frac{bD_1}{30} + \frac{4bD_{xy}}{30}$ $\frac{bD_1}{30}$
$-\frac{h\,d_x\,b}{6}$	0	0	0	$-\frac{h\,d_x\,b}{3}$	0	0	0
0	$-\frac{h\,G\,b}{6}$	0	0	0	$-\frac{h\,G\,b}{3}$	0	0
0	0	$\frac{6D_1}{5b} + \frac{24D_{xy}}{5b}$ $+\frac{6D_1}{5b}$	$\frac{D_1}{10} + \frac{4D_{xy}}{10}$ $+\frac{D_1}{10}$	0	0	$-\frac{6D_1}{5b} - \frac{24D_{xy}}{5b}$ $-\frac{6D_1}{5b}$	$\frac{11D_1}{10} + \frac{4D_{xy}}{10}$ $+\frac{D_1}{10}$
0	0	$-\frac{D_1}{10} - \frac{4D_{xy}}{10}$ $-\frac{D_1}{10}$	$\frac{bD_1}{30} + \frac{4bD_{xy}}{30}$ $+\frac{bD_1}{30}$	0	0	$\frac{D_1}{10} + \frac{4D_{xy}}{10}$ $+\frac{11D_1}{10}$	$-\frac{2bD_1}{15} - \frac{8bD_{xy}}{15}$ $-\frac{2bD_1}{15}$

THE ε MATRIX

0	$-\frac{hG}{2} + \frac{hd_{xy}}{2}$	0	0	0	$-\frac{hG}{2} - \frac{hd_{xy}}{2}$	0	0
$+\frac{hG}{2} - \frac{hd_{xy}}{2}$	0	0	0	$-\frac{hG}{2} - \frac{hd_{xy}}{2}$	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	$\frac{hG}{2} + \frac{hd_{xy}}{2}$	0	0	0	$\frac{hG}{2} - \frac{hd_{xy}}{2}$	0	0
$+\frac{hG}{2} + \frac{hd_{xy}}{2}$	0	0	0	$-\frac{hG}{2} + \frac{hd_{xy}}{2}$	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

THE ζ MATRIX

$\frac{Gh}{b}$	0	0	0	$-\frac{Gh}{b}$	0	0	0
0	$\frac{d_y h}{b}$	0	0	0	$-\frac{d_y h}{6}$	0	0
0	0	$\frac{12D_y}{b^3}$	$\frac{6D_y}{b^2}$	0	0	$-\frac{12D_y}{b^3}$	$\frac{6D_y}{b^2}$
0	0	$\frac{6D_y}{b^2}$	$\frac{4D_y}{b}$	0	0	$-\frac{6D_y}{b^2}$	$\frac{2D_y}{b}$
$-\frac{Gh}{b}$	0	0	0	$\frac{Gh}{b}$	0	0	0
0	$-\frac{d_y h}{b}$	0	0	0	$\frac{d_y h}{b}$	0	0
0	0	$-\frac{12D_y}{b^3}$	$-\frac{6D_y}{b^2}$	0	0	$\frac{12D_y}{b^3}$	$-\frac{6D_y}{b^2}$
0	0	$\frac{6D_y}{b^2}$	$\frac{2D_y}{b}$	0	0	$-\frac{6D_y}{b^2}$	$\frac{4D_y}{b}$

APPENDIX D:

FINITE DIFFERENCE OPERATOR PATTERNS

Note: 1. SA is the spacing between cross-section nodes.

2. Central difference formulae are used.

i.e.
$$\frac{dy}{dx} = \frac{1}{2SA}\left[-1\, y_L + 0\, y_o + 1\, y_R\right]$$

$$\frac{d^2y}{dx^2} = \frac{1}{SA^2}\left[1\, y_L - 2\, y_o + 1\, y_R\right]$$

$$\frac{d^4y}{dx^4} = \frac{1}{SA^4}\left[1\, y_{LL} - 4\, y_L + 6\, y_o - 4\, y_R + 1\, y_{RR}\right]$$

4th DERIVATIVE ($\div$ SA^4)

Simply Supported Ends

0	0	0					
0	5	-4	1				
0	-4	6	-4	1			
	1	-4	6	-4	1		
		1	-4	6	-4	1	
			1	-4	6	-4	0
				1	-4	5	0
					0	0	0

Fixed Ends

0	0	0					
0	7	-4	1				
0	-4	6	-4	1			
	1	-4	6	-4	1		
		1	-4	6	-4	1	
			1	-4	6	-4	0
				1	-4	7	0
					0	0	0

Guided Ends

3	-4	1					
-4	7	-4	1				
1	-4	6	-4	1			
	1	-4	6	-4	1		
		1	-4	6	-4	1	
			1	-4	6	-4	1
				1	-4	7	-4
					1	-4	3

2nd DERIVATIVE (÷ SA^2)

Simply Supported Ends

$$\begin{array}{cccccccc}
(0) & 0 & & & & & & \\
0 & (-2) & 1 & & & & & \\
 & 1 & (-2) & 1 & & & & \\
 & & & & & & & \\
 & & & & 1 & (-2) & 1 & \\
 & & & & & 1 & (-2) & 0 \\
 & & & & & & 0 & (0)
\end{array}$$

Fixed Ends

$$\begin{array}{cccccccc}
(0) & 0 & & & & & & \\
0 & (-2) & 1 & & & & & \\
 & 1 & (-2) & 1 & & & & \\
 & & & & & & & \\
 & & & & 1 & (-2) & 1 & \\
 & & & & & 1 & (-2) & 0 \\
 & & & & & & 0 & (0)
\end{array}$$

Guided Ends

$$\begin{array}{cccccccc}
(-1) & 1 & & & & & & \\
1 & (-2) & 1 & & & & & \\
 & 1 & (-2) & 1 & & & & \\
 & & & & & & & \\
 & & & & 1 & (-2) & 1 & \\
 & & & & & 1 & (-2) & 1 \\
 & & & & & & 1 & (-1)
\end{array}$$

Function u: Fixed Ends

$$\begin{array}{cccccccc}
(0) & 0 & & & & & & \\
0 & (-2) & 1 & & & & & \\
 & 1 & (-2) & 1 & & & & \\
 & & & & & & & \\
 & & & & 1 & (-2) & 1 & \\
 & & & & & 1 & (-2) & 0 \\
 & & & & & & 0 & (0)
\end{array}$$

Function u: Free Ends:

$$\begin{array}{cccccccc}
(-1) & 1 & & & & & & \\
1 & (-2) & 1 & & & & & \\
 & 1 & (-2) & 1 & & & & \\
 & & & & & & & \\
 & & & & 1 & (-2) & 1 & \\
 & & & & & 1 & (-2) & 1 \\
 & & & & & & 1 & (-1)
\end{array}$$

1st DERIVATIVE (÷ 2 SA)

Simply Supported Ends

```
(0)  1
-1  (0)  1
     -1  (0)  1

                    -1  (0)  1
                         -1  (0)  1
                              -1  (0)
```

Fixed Ends

```
(0)  0
 0  (0)  1
     -1  (0)  1

                    -1  (0)  1
                         -1  (0)  0
                               0  (0)
```

Guided Ends

```
(0)  0
-1  (0)  1
     -1  (0)  1

                     1  (0)  1
                         -1  (0)  1
                               0  (0)
```

Function u: Fixed Ends

```
(0)  0
 0  (0)  1
     -1  (0)  1

                    -1  (0)  1
                         -1  (0)  0
                               0  (0)
```

Function u: Free Ends

```
(-1)  1
-1  (0)  1
     -1  (0)  1

                    -1  (0)  1
                         -1  (0)  1
                              -1  (1)
```

The following Finite Difference Operator patterns are used in Sub-routine ECHO to determine stresses from deflections.

2nd DERIVATIVE (÷ SA^2)

			(2)	-5	4	-1	Forward Difference
		1	(-2)	1			Central Difference
-1	4	-5	(2)				Backward Difference

1st DERIVATIVE (÷ 2 SA)

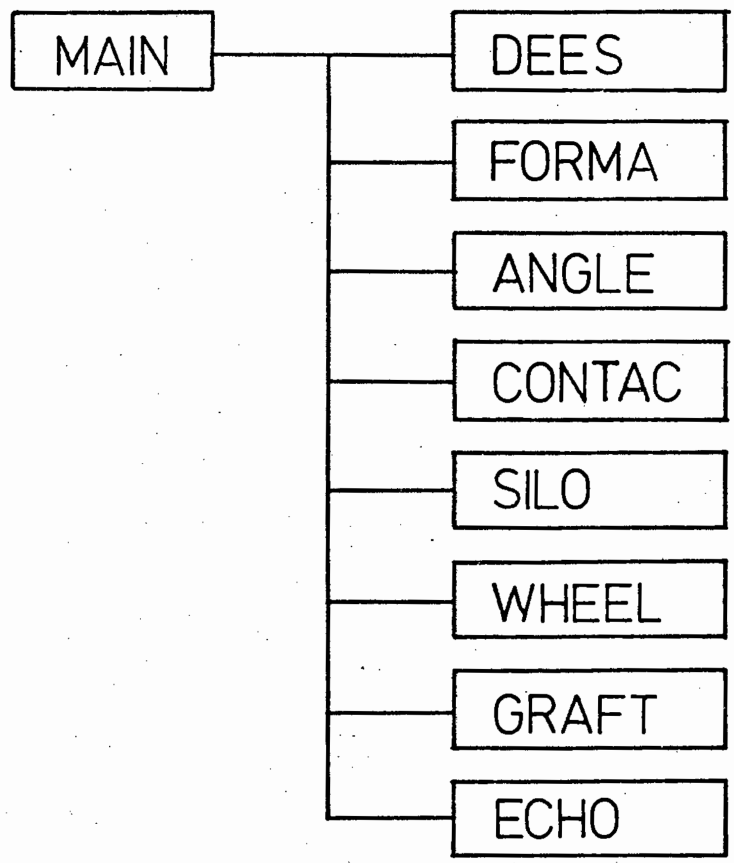
		(-3)	4	-1	Forward Difference
	-1	(0)	1		Central Difference
1	-4	(3)			Backward Difference

FORTRAN V COMPUTER PROGRAM

MENAI*

*The strait over which Robert Stephenson built the Britannia Tubular Bridge in 1850.

MENAI SUBROUTINE SEQUENCE



```
PROJECT*MENAI(1).MAIN
            COMPILER(XM=3)
      C     ****************************************************************
      C     'MENAI' PROGRAM TO SOLVE TRANSLATIONAL SHELL STRUCTURES
      C     THE FINITE STRIP METHOD OF ANALYSIS IS USED
      C     CUBIC POLYNOMIAL DISPLACEMENT FUNCTIONS ARE USED ACROSS THE STRIPS
      C     FOR THE NORMAL DISPLACEMENTS,AND THE IN-PLANE DISPLACEMENTS ARE
      C     VARIED LINEARLY;
      C     FINITE DIFFERENCE APPROXIMATIONS TO THE DIFFERENTIAL EQUATIONS
      C     ARE USED ALONG THE LENGTH OF THE STRIPS * UNITS ARE ALWAYS N.M.S.*
      C     ****************************************************************
            COMMON/EXT/SKU(440,133),SD4CA(440,5),SD2C3(440,3),SDOZE(440),
           .SD1CE(440,3),CA(44,44),CG(44,44),CE(44,44),
           .CZ(44,44),SIGMAX(180),SIGMAY(180),TAUXY(180),
           .AMX(180),AMY(180),AMXY(180)
            COMMON DDX(30),DDY(30),DD1(30),DDXY(30),D4(30),DB(30)
            COMMON FA(8,8,30),FG(8,8,30),FE(8,8,30),FZ(8,8,30)
            COMMON AFA(8,8,30),AFG(8,8,30),AFE(8,8,30),AFZ(8,8,30)
            COMMON SD4(6,5,6),SD2(6,3,6),SD1(6,3,6),SD(6,6)
            COMMON DEX,DEY,DPX,DPY,DG,IDNUM,DLDX,DLDXY,DLDY
            COMMON MHC,MSKD,MSKB,MS31C,MSZERO(250)
            COMMON NW27,JENOX,JENUM,MS(31),ME(31)
            COMMON ANG(30),SA
            COMMON /B2/ATRANS
            DOUBLE PRECISION ATRANS(8,8,30)
            MHC=44
            MSKD=440
            MSKB=132
            NW27=133
            PARAMETER JE1=720
            DIMENSION EDV(JE1)
            PRINT 541
        541 FORMAT(1H1,6(/),1H0,30X,60('*'),/,
           .31X,'*',13X,'MENAI: TRANSLATIONAL SHELL PROGRAM',11X,'*',/,
           .31X,'*',58X,'*',/,
           .31X,'*',3X,'DATE:.......................',28X,'*',/,
           .31X,'*',58X,'*',/,
           .31X,'*',3X,'PROJECT:....................',27X,'*',/,
           .31X,'*',58X,'*',/,
           .31X,'*',3X,'CLIENT:.....................',28X,'*')
            PRINT 567
        567 FORMAT(1H ,30X,'*',58X,'*',/,
           .31X,'*',3X,'GENERAL DESCRIPTION OF STRUCTURE:....................
           .',1X,'*',/,
           .31X,'*',58X,'*',/,
           .31X,'*',3X,52('.'),3X,'*',/,
           .31X,'*',58X,'*')
            PRINT 566
        566 FORMAT(1H ,30X,'*',3X,'LOADING CASE CONSIDERED',30('.'),2X,'*',/,
           .31X,'*',58X,'*',/,
           .31X,60('*'))
      C
      C     EVALUATION OF ELASTIC CONSTANTS
      C
            CALL DEES
      C
            PRINT 542
        542 FORMAT(1H1,'**ALL UNITS USED ARE N.M.S.**')
            WRITE(6,501) DEX,DEY,DPX,DPY,DG
        501 FORMAT(1H0,'YOUNGS MODULUS X:',8X,1PE16.6,/,
```

```
         .' YOUNGS MODULUS Y:',8X,E16.6,/,
         .' POISSON RATIO  X:',12X,0PF6.3,/,
         .' POISSON RATIO  Y:',12X,F6.3,/,
         .' SHEAR MODULUS  G:',8X,1PE16.6,/,)
      C
      C     FORMULATION OF THE BASIC STIFFNESS MATRICES
      C
            CALL FORMA
      C
      C     TO COMPILE THE TRANSFORMATION MATRICES FOR INDIVIDUAL STRIPS
      C     AND TO COMPILE GLOBAL STRIP STIFFNESS MATRICES
      C
            CALL ANGLE
      C
            PRINT 543
      543   FORMAT(1H0,'STRIP NO     STRIP WIDTH     STRIP THICKNESS    STRIP
           .ANGLE(RAD)',/,
           .62('-'))
            DO 544 I=1,IDNUM
      544   WRITE(5,545) I,DB(I),DH(I),ANG(I)
      545    FORMAT(1H0,4X,I2,11X,F6.3,10X,F6.3,9X,F10.7)
      C
      C     TO ESTABLISH THE COMBINED STIFFNESS MATRICES FOR THE TOTAL
      C     CROSS-SECTION
      C
            CALL CONTAC
      C
      C     TO COMPILE THE FINITE DIFFERENCE EQUATIONS, MULTIPLY WITH CA,CG,
      C     CE,CZ MATRICES AND COMBINE TO FORM UPPER TRIANGULAR MATRIX SKU
      C
            CALL SILO
      C
            IF(MS(1).EQ.1) PRINT 570
            IF(MS(1).EQ.3) PRINT 571
            IF(MS(1).EQ.5) PRINT 572
            IF(MS(1).EQ.7) PRINT 573
      570   FORMAT(1H0,'BOUNDARY CONDITION AT START IS FIXED')
      571   FORMAT(1H0,'BOUNDARY CONDITION AT START IS GUIDED')
      572   FORMAT(1H0,'BOUNDARY CONDITION AT START IS FREE')
      573   FORMAT(1H0,'BOUNDARY CONDITION AT START IS SIMPLY SUPPORTED')
            IF(ME(1).EQ.1) PRINT 575
            IF(ME(1).EQ.3) PRINT 576
            IF(ME(1).EQ.5) PRINT 577
            IF(ME(1).EQ.7) PRINT 578
      575   FORMAT(1H0,'BOUNDARY CONDITION AT END IS FIXED')
      576   FORMAT(1H0,'BOUNDARY CONDITION AT END IS GUIDED')
      577   FORMAT (1H0,'BOUNDARY CONDITION AT END IS FREE')
      578   FORMAT(1H0,'BOUNDARY CONDITION AT END IS SIMPLY SUPPORTED')
            WRITE(5,580) SA
      580   FORMAT(1H0,'NODAL SPACING =',F6.3)
      C     FORMULATION OF LOAD MATRIX
      C
            CALL WHEEL
      C
      C     TO SOLVE DISPLACEMENTS
      C
            CALL GRAFT
      C
            PRINT 565
      565   FORMAT(1H0,'CROSS-SECTION',6X,'NODE',29X,'GLOBAL DISPLACEMENTS
```

```
               .')
               PRINT 546
        546    FORMAT(1H0,35X,'U',18X,'V',14X,'W',13X,'THETA(RAD)')
               PRINT 568
        568    FORMAT(1H ,96('-'))
        C      NUMBER OF CROSS-SECTIONS
               KA10=3*MSKD/MSKB
        C      NUMBER OF NODES PER SECTION
               KA11=MSKB/12
        C      DEGREES OF FREEDOM PER SECTION
               KA16=MSKB/3
               DO 549 K=1,KA10
               WRITE(5,547) K
        547    FORMAT(1H ,6X,I3)
               DO 549 L=1,KA11
               KA12=(K-1)*KA16+L*4-3
               KA13=KA12+1
               KA14=KA13+1
               KA15=KA14+1
               WRITE(5,548) L,SKU(KA12,NW27),SKU(KA13,NW27),
              .SKU(KA14,NW27),SKU(KA15,NW27)
        548    FORMAT(1H ,20X,I2,7X,1PE12.6,6X,E12.6,6X,E12.6,6X,E12.6)
        549    CONTINUE
        C
        C      TO SOLVE STRESSES IN LOCAL AXES CO-ORDINATES
        C
               CALL ECHO (EDV,JE1)
        C
               PRINT 550
        550    FORMAT(1H1,'CROSS-SECTION  SUB-STRUCTURE  NODE',35X,'LOCAL STRESSE
              .S')
               PRINT 551
        551    FORMAT(1H ,41X,'SIGMA-X',7X,'SIGMA-Y',6X,'TAU-XY',
              .12X,'M-X',9X,'M-Y',8X,'M-XY')
               PRINT 569
        569    FORMAT(1H ,111('-'))
               DO 555 I=1,JENOX
               WRITE(5,552) I
        552    FORMAT(1H0,6X,I2)
               DO 555 J=1,JENUM
               WRITE(5,553) J
        553    FORMAT(1H ,21X,I2)
               DO 555 K=1,3
               JE201=(I-1)*3*JENUM+J*3-2+K-1
               WRITE(5,554) K,SIGMAX(JE201),SIGMAY(JE201),TAUXY(JE201),
              .AMX(JE201),AMY(JE201),AMXY(JE201)
        554    FORMAT(1H ,32X,I2,6X,1PE11.5,2X,E11.5,2X,E11.5,
              .2X,E11.5,2X,E11.5,2X,E11.5)
        555    CONTINUE
               STOP
               END
@PRT,S MENAI.DEES
```

```
PROJECT*MENAI(1).DEES
            COMPILER(XM=3)
            SUBROUTINE DEES
            COMMON/EXT/SKU(440,133),SD4CA(440,5),SD2CG(440,3),SDCZE(440),
           .SD1CE(440,3),CA(44,44),CG(44,44),CE(44,44),
           .CZ(44,44),SIGMAX(180),SIGMAY(180),TAUXY(180),
           .AMX(180),AMY(180),AMXY(180)
            COMMON DDX(30),DDY(30),DD1(30),DDXY(30),DH(30),DB(30)
            COMMON FA(8,8,30),FG(8,8,30),FE(8,8,30),FZ(8,8,30)
            COMMON AFA(8,8,30),AFG(8,8,30),AFE(8,3,30),AFZ(8,3,30)
            COMMON SD4(6,5,6),SD2(5,3,6),SD1(6,3,6),SD(6,6)
            COMMON DEX,DEY,DPX,DPY,DG,IDNUM,DLDX,DLDXY,DLDY
            COMMON MHC,MSKD,MSKB,MS31D,MSZERO(250)
            COMMON NW27,JENOX,JENUM,MS(31),ME(31)
            COMMON ANG(30),SA
      C     READ EX , EY , POISSON X , POISSON Y
            READ (8,100) DEX , DEY , DPX , DPY
      100   FORMAT (       )
      C     READ NUMBER OF STRIPS IN STRUCTURE IDNUM
            READ(8,101) IDNUM
      101   FORMAT(      )
      C     READ STRIP THICKNESS AND WIDTH
            DO 102 I = 1,IDNUM
      102   READ (8,103) DH(I) , DB(I)
      103   FORMAT(      )
      C     IN ORDER OF CALCULATION;SHEAR MODULUS,LITTLE(DX,DXY,DY),
      C     CAPITAL(DX,DY,D1,DXY)
            DG = (DEX+DEY)/(4.*(1.+(DPX+DPY)/2.))
            DLDX = DEX/(1.-DPX*DPY)
            DLDXY = DPX*DEY/(1.-DPX*DPY)
            DLDY = DEY/(1.-DPX*DPY)
            DO 104 I = 1,IDNUM
            DDX(I) = (DLDX*DH(I)**3.)/12.
            DDY(I) = (DLDY*DH(I)**3.)/12.
            DD1(I) = (DLDXY*DH(I)**3.)/12.
            DDXY(I) = (DG*DH(I)**3.)/12.
      104   CONTINUE
            RETURN
            END

@PRT,S MENAI.FORMA
```

```
PROJECT*MENAI(1).FORMA
              COMPILER(XM=3)
              SUBROUTINE FORMA
              COMMON/EXT/SKU(440,133),SD4CA(440,5),SD2CG(440,3),SDCZE(440),
             .SD1CE(440,3),CA(44,44),CG(44,44),CE(44,44),
             .CZ(44,44),SIGMAX(180),SIGMAY(180),TAUXY(180),
             .AMX(180),AMY(180),AMXY(180)
              COMMON DDX(30),DDY(30),DD1(30),DDXY(30),DH(30),DB(30)
              COMMON FA(8,8,30),FG(8,8,30),FE(8,8,30),FZ(8,8,30)
              COMMON AFA(8,8,30),AFG(8,8,30),AFE(8,8,30),AFZ(8,8,30)
              COMMON SD4(6,5,6),SD2(6,3,6),SD1(6,3,6),SD(6,6)
              COMMON DEX,DEY,DPX,DPY,DG,IDNUM,DLDX,DLDXY,DLDY
              COMMON MHC,MSKD,MSKB,MS31D,MSZERO(250)
              COMMON NW27,JENOX,JENUM,MS(31),ME(31)
              COMMON ANG(30),SA
        C
        C     FORMULATION OF STIFFNESS MATRICES. THE FA,FG,FE,FZ MATRICES ARE
        C     COMPILED FOR ONE STRIP AT A TIME IN ORDER GIVEN
              DO 1000 I=1,IDNUM
        C     THE ALPHA MATRIX ************************************************************
              FA(3,3,I) = 13.*DB(I)*DDX(I)/35.
              FA(4,4,I) = DB(I)**3.*DDX(I)/105.
              FA(7,7,I) = 13.*DB(I)*DDX(I)/35.
              FA(8,8,I) = DB(I)**3.*DDX(I)/105.
              FA(4,3,I) = 11.*DB(I)**2.*DDX(I)/210.
              FA(3,4,I) = FA(4,3,I)
              FA(8,7,I) = -11.*DB(I)**2.*DDX(I)/210.
              FA(7,8,I) = FA(8,7,I)
              FA(7,4,I) = 13.*DB(I)**2.*DDX(I)/420.
              FA(4,7,I) = FA(7,4,I)
              FA(7,3,I) = 9.*DB(I)*DDX(I)/70.
              FA(3,7,I) = FA(7,3,I)
              FA(8,4,I) = -DB(I)**3.*DDX(I)/140.
              FA(4,8,I) = FA(8,4,I)
              FA(8,3,I) = -13.*DB(I)**2.*DDX(I)/420.
              FA(3,8,I) = FA(8,3,I)
        C     THE GAMMA MATRIX ************************************************************
              FG(1,1,I) = -DH(I)*DLDX*DB(I)/3.
              FG(2,2,I) = -DH(I)*DG*DB(I)/3.
              FG(3,3,I) = -6.*DD1(I)/(5.*DB(I))-24.*DDXY(I)/(5.*DB(I))
             .-6.*DD1(I)/(5.*DB(I))
              FG(4,4,I) = -2.*DB(I)*DD1(I)/15.-8.*DB(I)*DDXY(I)/15.
             .-2.*DB(I)*DD1(I)/15.
              FG(5,5,I) = FG(1,1,I)
              FG(6,6,I) = FG(2,2,I)
              FG(7,7,I) = FG(3,3,I)
              FG(8,8,I) = FG(4,4,I)
              FG(4,3,I) = -DD1(I)/10.-4.*DDXY(I)/10.-11.*DD1(I)/10.
              FG(3,4,I) = FG(4,3,I)
              FG(8,7,I) = DD1(I)/10.+4.*DDXY(I)/10.+11.*DD1(I)/10.
              FG(7,8,I) = FG(8,7,I)
              FG(5,1,I) = -DH(I)*DLDX*DB(I)/6.
              FG(1,5,I) = FG(5,1,I)
              FG(6,2,I) = -DH(I)*DG*DB(I)/6.
              FG(2,6,I) = FG(6,2,I)
              FG(7,3,I) = 6.*DD1(I)/(5.*DB(I))+24.*DDXY(I)/(5.*DB(I))
             .+6.*DD1(I)/(5.*DB(I))
              FG(3,7,I) = FG(7,3,I)
              FG(7,4,I) = DD1(I)/10.+4.*DDXY(I)/10.+DD1(I)/10.
              FG(4,7,I) = FG(7,4,I)
```

```
      FG(8,4,I) = DB(I)*DD1(I)/30.+4.*DB(I)*DDXY(I)/30.
     .+DB(I)*DD1(I)/30.
      FG(4,8,I) = FG(8,4,I)
      FG(8,3,I) = -DD1(I)/10.-4.*DDXY(I)/10.-DD1(I)/10.
      FG(3,8,I) = FG(8,3,I)
C     THE EPSILON MATRIX ************************************************
      FE(2,1,I) = DH(I)*DLDXY*(-1./2.)-DH(I)*DG*(-1./2.)
      FE(1,2,I) = -FE(2,1,I)
      FE(5,2,I) = DG*DH(I)/2.-DLDXY*(-DH(I)/2.)
      FE(2,5,I) = -FE(5,2,I)
      FE(6,1,I) = DLDXY*DH(I)/2.-DG*(-DH(I)/2.)
      FE(1,6,I) = -FE(6,1,I)
      FE(6,5,I) = DLDXY*(DH(I)/2.)-DG*(DH(I)/2.)
      FE(5,6,I) = -FE(6,5,I)
C     THE ZETA MATRIX ***************************************************
      FZ(1,1,I)=DG*DH(I)/DB(I)
      FZ(2,2,I) = DLDY*DH(I)/DB(I)
      FZ(3,3,I) = 12.*DDY(I)/DB(I)**3.
      FZ(4,4,I) = 4.*DDY(I)/DB(I)
      FZ(5,5,I) = FZ(1,1,I)
      FZ(6,6,I) = FZ(2,2,I)
      FZ(7,7,I) = FZ(3,3,I)
      FZ(8,8,I) = FZ(4,4,I)
      FZ(4,3,I) = 6.*DDY(I)/DB(I)**2.
      FZ(3,4,I) = FZ(4,3,I)
      FZ(8,7,I) = -6.*DDY(I)/DB(I)**2.
      FZ(7,8,I) = FZ(8,7,I)
      FZ(7,4,I) = -6.*DDY(I)/DB(I)**2.
      FZ(4,7,I) = FZ(7,4,I)
      FZ(5,1,I)=-DG*DH(I)/DB(I)
      FZ(1,5,I) = FZ(5,1,I)
      FZ(6,2,I)=-DLDY*DH(I)/DB(I)
      FZ(2,6,I) = FZ(6,2,I)
      FZ(7,3,I) = -12.*DDY(I)/DB(I)**3.
      FZ(3,7,I) = FZ(7,3,I)
      FZ(8,4,I) = 2.*DDY(I)/DB(I)
      FZ(4,8,I) = FZ(8,4,I)
      FZ(8,3,I) = 6.*DDY(I)/DB(I)**2.
      FZ(3,8,I) = FZ(8,3,I)
 1000 CONTINUE
      RETURN
      END
@PRT,S MENAI.ANGLE
```

```
?RUJECT*MENAI(1).ANGLE
            COMPILER(XM=3)
            SUBROUTINE ANGLE
            COMMON/EXT/SKU(440,133),SD4CA(440,5),SD2CG(440,3),SDOZE(440),
           .SD1CE(440,3),CA(44,44),CG(44,44),CE(44,44),
           .CZ(44,44),SIGMAX(180),SIGMAY(180),TAUXY(180),
           .AMX(180),AMY(180),AMXY(180)
            COMMON DDX(30),DDY(30),DD1(30),DDXY(30),DH(30),DB(30)
            COMMON FA(8,8,30),FG(8,8,30),FE(8,8,30),FZ(8,8,30)
            COMMON AFA(8,8,30),AFG(8,8,30),AFE(8,8,30),AFZ(8,8,30)
            COMMON SD4(6,5,6),SD2(6,3,6),SD1(6,3,6),SD(6,6)
            COMMON DEX,DEY,DPX,DPY,DG,IDNUM,DLDX,DLDXY,DLDY
            COMMON MHC,MSKD,MSKB,MSZID,MSZERO(250)
            COMMON NW27,JENOX,JENUM,MS(31),ME(31)
            COMMON ANG(30),SA
            COMMON/B2/ATRANS
      C
            DIMENSION ASUM(8,8)
            DOUBLE PRECISION ATRANS(8,8,30)
            DOUBLE PRECISION ATTRAN(8,8,30)
      C     READ ANGLE OF STRIP:ANGLE IS MEASURED FROM GLOBAL
      C     Y AXIS TO LOCAL Y AXISFOR STRIP,CLOCKWISE ANGLE IS
      C     CONSIDERED POSITIVE AND COUNTERCLOCKWISE ANGLE IS CONSIDERED
      C     NEGATIVE.ANGLE IS ENTERED IN DEGREES AND FRACTIONS OF A
      C     DEGREE.ORDER OF ENTRY IS AS FOR THICKNESS AND WIDTH CARDS
      C     ONE ANGLE PER CARD
            DO 2000 I=1,IDNUM
      2000  READ (3,2001) ANG(I)
      2001  FORMAT(     )
      C     CHANGE ANGLE TO RADIANS
            DO 2002 I=1,IDNUM
            ANG(I) = ANG(I)/57.29577951
      2002  CONTINUE
      C     FORMULATE TRANSFORMATION MATRICES
            DO 2003 I = 1,IDNUM
            ATRANS(1,1,I) = 1.
            ATRANS(2,2,I) = COS(ANG(I))
            ATRANS(3,3,I) = ATRANS(2,2,I)
            ATRANS(4,4,I) = 1.
            ATRANS(5,5,I) = 1.
            ATRANS(6,6,I) = ATRANS(2,2,I)
            ATRANS(7,7,I) = ATRANS(2,2,I)
            ATRANS(8,8,I) = 1.
            ATRANS(2,3,I) = SIN(ANG(I))
            ATRANS(3,2,I) = -ATRANS(2,3,I)
            ATRANS(6,7,I) = ATRANS(2,3,I)
            ATRANS(7,6,I) = -ATRANS(2,3,I)
      C     FORMULATION OF THE TRANSFORMED TRANSFORMATION MATRICES
            ATTRAN(1,1,I) = 1.
            ATTRAN(2,2,I) = COS(ANG(I))
            ATTRAN(3,3,I) = ATTRAN(2,2,I)
            ATTRAN(4,4,I) = 1.
            ATTRAN(5,5,I) = 1.
            ATTRAN(6,6,I) = ATTRAN(2,2,I)
            ATTRAN(7,7,I) = ATTRAN(2,2,I)
            ATTRAN(8,8,I) = 1.
            ATTRAN(3,2,I) = SIN(ANG(I))
            ATTRAN(2,3,I) = -ATTRAN(3,2,I)
            ATTRAN(6,7,I) = -ATTRAN(3,2,I)
            ATTRAN(7,6,I) = ATTRAN(3,2,I)
```

```
 2003 CONTINUE
C
C     TO ESTABLISH THE TRANSFORMED ALFA MATRICES (ATTRAN*FA*ATRANS)
C
      DO 2023 I1 = 1,IDNUM
C     TO CLEAR TEMP. MATRIX ASUM
      DO 2004 I = 1,8
      DO 2004 J=1,8
      ASUM(I,J) = 0.
 2004 CONTINUE
C     TO DO FA*ATRANS=ASUM
      DO 2005 I2 = 1,8
      DO 2005 I3 = 1,8
      DO 2005 I4 = 1,8
      ASUM(I2,I3)=ASUM(I2,I3)+FA(I2,I4,I1)*ATRANS(I4,I3,I1)
 2005 CONTINUE
C     TO CLEAR FA( , ,I1) SO SAME STORAGE MAY BE USED FOR AFA( , ,I1)
      DO 2006 L = 1,8
      DO 2006 M = 1,8
      FA(L,M,I1) = 0.
 2006 CONTINUE
C     TO DO ATTRAN*ASUM=AFA
      DO 2007 K2 = 1,8
      DO 2007 K3 = 1,8
      DO 2007 K4 = 1,8
      AFA(K2,K3,I1) = AFA(K2,K3,I1)+ATTRAN(K2,K4,I1)*ASUM(K4,K3)
 2007 CONTINUE
 2023 CONTINUE
C
C     TO ESTABLISH THE TRANSFORMED GAMMA MATRICES(ATTRAN*FG*ATRANS)
C
      DO 2008 I1 = 1,IDNUM
C     TO CLEAR TEMP. MATRIX ASUM
      DO 2009 I = 1,8
      DO 2009 J = 1,8
      ASUM(I,J) = 0.
 2009 CONTINUE
C     TO DO FG*ATRANS=ASUM
      DO 2010 I2 = 1,8
      DO 2010 I3 = 1,8
      DO 2010 I4 = 1,8
      ASUM(I2,I3) = ASUM(I2,I3)+FG(I2,I4,I1)*ATRANS(I4,I3,I1)
 2010 CONTINUE
C     TO CLEAR FG( , ,I1) SO SAME STORAGE MAY BE USED FOR AFG( , ,I1)
      DO 2011 L = 1,8
      DO 2011 M = 1,8
      FG(L,M,I1) = 0.
 2011 CONTINUE
C     TO DO ATTRAN*ASUM = AFG
      DO 2012 K2 = 1,8
      DO 2012 K3 = 1,8
      DO 2012 K4 = 1,8
      AFG(K2,K3,I1) = AFG(K2,K3,I1)+ATTRAN(K2,K4,I1)*ASUM(K4,K3)
 2012 CONTINUE
 2008 CONTINUE
C
C     TO ESTABLISH THE TRANSFORMED EPSILON MATRICES(ATTRAN*FE*ATRANS)
      DO 2013 I1 = 1,IDNUM
C
C     TO CLEAR TEMP. MATRIX ASUM
```

PROJECT*MENAI(1).CONTAC

```
      COMPILER(XM=3)
      SUBROUTINE CONTAC
      COMMON/EXT/SKU(440,133),SD4CA(440,5),SD2CG(440,3),SD0ZE(440),
     .SD1CE(440,3),CA(44,44),CG(44,44),CE(44,44),
     .CZ(44,44),SIGMAX(180),SIGMAY(180),TAUXY(180),
     .AMX(180),AMY(180),AMXY(180)
      COMMON DDX(30),DDY(30),DD1(30),DDXY(30),DH(30),DB(30)
      COMMON FA(8,8,30),FG(8,8,30),FE(8,8,30),FZ(8,8,30)
      COMMON AFA(8,8,30),AFG(8,8,30),AFE(8,8,30),AFZ(8,8,30)
      COMMON SD4(6,5,6),SD2(6,3,6),SD1(6,3,6),SD(6,6)
      COMMON DEX,DEY,DPY,DPY,DG,IDNUM,DLDX,DLDXY,DLDY
      COMMON MHC,MSKD,MSK3,MS31D,MSZERO(250)
      COMMON NW27,JENOX,JENUM,MS(31),ME(31)
      COMMON ANG(30),SA
C
C     READ A CARD TELLING WHICH AF(A,G,E OR Z)(IDNUM) TO PICK(LC3);AND
C     ROW OR COLUMN NUMBER WHERE IT MUST START IN C(A,G,E OR Z)
C     MATRIX(LC4);ONE CARD PER STRIP.THE MATRICES THAT
C     DONT REQUIRE SUBDIVISION ARE DONE FIRST
3005  READ(8,3002) LC3,LC4
3002  FORMAT(      )
C     TEST FOR LAST CARD ; MUST BE 0   ,  0 TO STOP PROCESS
      IF(LC3.EQ.0 .AND. LC4 .EQ. 0 ) GO TO 3006
      DO 3003 LC1 = 1,8
      DO 3004 LC2 = 1,8
      LC13 = (LC4-1+LC1)
      LC14 = (LC4-1+LC2)
      CA(LC13,LC14) = CA(LC13,LC14)+AFA(LC1,LC2,LC3)
      CG(LC13,LC14) = CG(LC13,LC14)+AFG(LC1,LC2,LC3)
      CE(LC13,LC14) = CE(LC13,LC14)+AFE(LC1,LC2,LC3)
      CZ(LC13,LC14) = CZ(LC13,LC14)+AFZ(LC1,LC2,LC3)
3004  CONTINUE
3003  CONTINUE
      GO TO 3005
3006  CONTINUE
C     PROCEED WITH *ODD-BALL* PLACEMENT
C     READ NUMBER OF *ODD-BALLS*
      READ(8,3011) LC5
3011  FORMAT(      )
      IF(LC5 .EQ. 0 ) GO TO 3007
C     THE ODD-BALLS ARE PLACED IN QUARTERS:1ST LH TOP QUARTER;
C     2ND RH TOP QUARTER;3RD LH BOTTOM QUARTER;4TH RH BOTTOM QUARTER
C
C     READ A CARD TELLING WHICH AF(A,G,E OR Z) (IDNUM) TO PICK
C     (LC6);AND ROW(LC7) AND COLUMN NUMBER(LC8) WHERE IT IS
C     TO START IN COMBINED MATRIX;THE ROW NUMBER WHERE TO START READING
C     IN SUB-MATRIX(LC11)AND THE COLUMN NUMBER WHERE TO START IN SUB-
C     MATRIX(LC12)
3010  READ(8,3008)LC6,LC7,LC8,LC11,LC12
C     TEST FOR LAST CARD ;MUST BE 0,0,0,0,0TO STOP PROCESS
      IF(LC6 .EQ. 0 ) GO TO 3007
3008  FORMAT(      )
C
      DO 3009 LC9 = 1,4
      DO 3009 LC10 = 1,4
      LC15 = (LC7-1+LC9)
      LC16 = (LC8-1+LC10)
      LC17 = (LC9-1+LC11)
      LC18 = (LC10-1+LC12)
```

```
           CA(LC15,LC16) = CA(LC15,LC16)+AFA(LC17,LC18,LC6)
           CG(LC15,LC16) = CG(LC15,LC16)+AFG(LC17,LC18,LC6)
           CE(LC15,LC16) = CE(LC15,LC16)+AFE(LC17,LC18,LC6)
           CZ(LC15,LC16) = CZ(LC15,LC16)+AFZ(LC17,LC18,LC6)
     3009  CONTINUE
           GO TO 3010
     3007  CONTINUE
           RETURN
           END
@PRT,S MENAI.SILO
```

```
      SD4(5,5,3)=1.
      SD4(6,3,3)=6.
      SD4(6,4,3)=-4.
      SD4(6,5,3)=1.
C     FOURTH FACE
      SD4(3,3,4)=6.
      SD4(3,4,4)=-4.
      SD4(4,3,4)=6.
      SD4(4,4,4)=-4.
      SD4(4,5,4)=1.
      SD4(5,3,4)=6.
      SD4(5,4,4)=-4.
      SD4(5,5,4)=1.
      SD4(6,3,4)=6.
      SD4(6,4,4)=-4.
C     FIFTH FACE
      SD4(3,3,5)=7.
      SD4(4,3,5)=5.
      SD4(4,4,5)=-2.
      SD4(5,3,5)=7.
      SD4(5,4,5)=-4.
      SD4(6,3,5)=5.
C     SIXTH FACE
      SD4(4,3,6)=1.
      SD4(5,3,6)=3.
C
C     SECOND DERIVATIVE
C
C     FIRST FACE
      SD2(2,2,1)=-1.
      SD2(2,3,1)=1.
      SD2(5,2,1)=-1.
      SD2(5,3,1)=1.
C     SECOND FACE
      SD2(1,2,2)=-2.
      SD2(1,3,2)=1.
      SD2(2,2,2)=-2.
      SD2(2,3,2)=1.
      SD2(3,2,2)=-2.
      SD2(3,3,2)=1.
      SD2(4,2,2)=-2.
      SD2(4,3,2)=1.
      SD2(5,2,2)=-2.
      SD2(5,3,2)=1.
      SD2(6,2,2)=-2.
      SD2(6,3,2)=1.
C     THIRD FACE
      SD2(1,2,3)=-2.
      SD2(1,3,3)=1.
      SD2(2,2,3)=-2.
      SD2(2,3,3)=1.
      SD2(3,2,3)=-2.
      SD2(3,3,3)=1.
      SD2(4,2,3)=-2.
      SD2(4,3,3)=1.
      SD2(5,2,3)=-2.
      SD2(5,3,3)=1.
      SD2(6,2,3)=-2.
      SD2(6,3,3)=1.
C     FOURTH FACE
```

```
      SD2(1,2,4)=-2.
      SD2(1,3,4)=1.
      SD2(2,2,4)=-2.
      SD2(2,3,4)=1.
      SD2(3,2,4)=-2.
      SD2(3,3,4)=1.
      SD2(4,2,4)=-2.
      SD2(4,3,4)=1.
      SD2(5,2,4)=-2.
      SD2(5,3,4)=1.
      SD2(6,2,4)=-2.
      SD2(6,3,4)=1.
C     FIFTH FACE
      SD2(1,2,5)=-2.
      SD2(2,2,5)=-2.
      SD2(2,3,5)=1.
      SD2(3,2,5)=-2.
      SD2(4,2,5)=-2.
      SD2(4,3,5)=1.
      SD2(5,2,5)=-2.
      SD2(5,3,5)=1.
      SD2(6,2,5)=-2.
      SD2(6,3,5)=1.
C     SIXTH FACE
      SD2(2,2,6)=-1.
      SD2(5,2,6)=-1.
C
C     FIRST DERIVATIVE
C
C     FIRST FACE
      SD1(4,2,1)=-1.
      SD1(4,3,1)=1.
      SD1(6,3,1)=1.
C     SECOND FACE
      SD1(1,3,2)=1.
      SD1(2,3,2)=1.
      SD1(3,3,2)=1.
      SD1(4,3,2)=1.
      SD1(5,3,2)=1.
      SD1(6,3,2)=1.
C     THIRD FACE
      SD1(1,3,3)=1.
      SD1(2,3,3)=1.
      SD1(3,3,3)=1.
      SD1(4,3,3)=1.
      SD1(5,3,3)=1.
      SD1(6,3,3)=1.
C     FOURTH FACE
      SD1(1,3,4)=1.
      SD1(2,3,4)=1.
      SD1(3,3,4)=1.
      SD1(4,3,4)=1.
      SD1(5,3,4)=1.
      SD1(6,3,4)=1.
C     FIFTH FACE
      SD1(2,3,5)=1.
      SD1(4,3,5)=1.
      SD1(5,3,5)=1.
      SD1(6,3,5)=1.
C     SIXTH FACE
```

```
      SD2CG(MS5,L)=SD2(MRS,L,J)
      SD1CE(MS5,L)=SD1(MRS,L,J)
 4009 CONTINUE
      SD0ZE(MS5)=SD(MRS,J)
 4007 CONTINUE
C     COPY IN V VALUES
      DO 4018 J=1,3
      MS7=((I*4-2)+(J-1)*MS4)
      DO 4019 K=1,5
      SD4CA(MS7,K)=SD4(MS6,K,J)
 4019 CONTINUE
      DO 4020 L=1,3
      SD2CG(MS7,L)=SD2(MS6,L,J)
      SD1CE(MS7,L)=SD1(MS6,L,J)
 4020 CONTINUE
      SD0ZE(MS7)=SD(MS6,J)
 4018 CONTINUE
C     COPY IN W VALUES
      DO 4028 J=1,3
      MS10=((I*4-1)+(J-1)*MS4)
      DO 4029 K=1,5
      SD4CA(MS10,K)=SD4(MS9,K,J)
 4029 CONTINUE
      DO 4030 L=1,3
      SD2CG(MS10,L)=SD2(MS9,L,J)
      SD1CE(MS10,L)=SD1(MS9,L,J)
 4030 CONTINUE
      SD0ZE(MS10)=SD(MS9,J)
 4028 CONTINUE
C     COPY IN THETA VALUES
      DO 4033 J=1,3
      MS13=((I*4)+(J-1)*MS4)
      DO 4034 K=1,5
      SD4CA(MS13,K)=SD4(MS12,K,J)
 4034 CONTINUE
      DO 4035 L=1,3
      SD2CG(MS13,L)=SD2(MS12,L,J)
      SD1CE(MS13,L)=SD1(MS12,L,J)
 4035 CONTINUE
      SD0ZE(MS13)=SD(MS12,J)
 4033 CONTINUE
 4003 CONTINUE
C     END OF STARTING CONDITIONS
C
C     END CONDITIONS
      DO 4036 I=1,31
C     TEST FOR LAST ENTRY
      IF (ME(I).EQ.0) GO TO 4036
C
      IF(ME(I).EQ.1) GO TO  4037
      IF(ME(I).EQ.3) GO TO 4039
      IF(ME(I).EQ.5) GO TO 4044
      IF(ME(I).EQ.7) GO TO 4046
 4037 MS14=1
      MS18=3
      MS22=3
      MS26=3
      GO TO 4048
 4039 MS14=1
      MS18=5
```

```
          SD4CA(MS27,K)=SD4(MS26,K,MS28)
     4067 CONTINUE
          DO 4068 L=1,3
          SD2CG(MS27,L)=SD2(MS26,L,MS28)
          SD1CE(MS27,L)=SD1(MS26,L,MS28)
     4068 CONTINUE
          SDDZE(MS27)=SD(MS26,MS28)
     4066 CONTINUE
     4030 CONTINUE
     C    END OF END CONDITIONS
     C    STRUCTURE MUST HAVE AT LEAST 7 NODES (6 DIVISIONS) ALONG LENGTH.
     C    READ NODAL SPACING ,AND DIVIDE BY SA**4,SA**2,&2*SA.
          READ(8,4069) SA
     4069 FORMAT(    )
          SA4=SA**4
          SA2=SA**2
          SA3=SA*2.
          DO 4070 I=1,MSKD
          DO 4071 J=1,5
          SD4CA(I,J)=SD4CA(I,J)/SA4
     4071 CONTINUE
          DO 4072 K=1,3
          SD2CG(I,K)=SD2CG(I,K)/SA2
     4072 CONTINUE
          DO 4073 L=1,3
          SD1CE(I,L)=SD1CE(I,L)/(SA3)
     4073 CONTINUE
     4070 CONTINUE
     C
     C    ROWS AND COLUMNS ARE REDUCED TO ZERO IN THE SKU
     C    MATRIX TO ALLOW FOR SUPPORT CONDITIONS.
     C
     C    READ NUMBER OF ROWS TO BE REDUCED TO ZERO.
          READ(8,4400) MS310
     4400 FORMAT(     )
     C    ROW NUMBERS TO BE ZEROED ARE ENTERED 10 TO A CARD IN FREE FORMAT
     C    FIRST INDICATE HOW MANY CARDS TO BE READ. (ZEROES TO BE USED
     C    TO FILL FIELD)
     C    READ NUMBER OF CARDS TO BE READ(MS401)
          READ(8,4400) MS401
          NN=1
     4401 IF(NN.GT.MS401) GO TO 4402
          MS403=10*NN-9
          MS402=10*NN
          READ(8,4400) (MSZERO(I),I=MS403,MS402)
          NN=NN+1
          GO TO 4401
     4402 CONTINUE
     C
     C    TO FACILITATE MATRIX MULTIPLICATION,THE SD4CA,SD2CG,&SD1CE
     C    MATRICES HAVE COLUMNS SHIFTED UP ON L.H. SIDE OF MAIN
     C    DIAGONAL AND DOWN ON RH SIDE OF MAIN DIAGONAL.
     C
     C    4TH DERIVATIVE
          MS35=2*MS4
          MS36=1*MS4
          MS38=MSKD-MS35
          DO 4074 I=1,MS38
          MS37=I+MS35
          SD4CA(I,1)=SD4CA(MS37,1)
```

```
4317  CONTINUE
      DO 4323 K=1,MS310
      MS324=MSZERO(K)
      SKU(MS324,1)=1.
4323  CONTINUE
      RETURN
      END
```

@PRT,S MENAI.WHEEL

```
      SKU(NW5,NW27)=WW(J)
      SKU(NW6,NW27)=WTHETA(J)
 5002 CONTINUE
C
C     FAR END
      NW7=MSKD-NW1+1
      DO 5003  K=1,NW1
      NW8=NW7+K-1
      SKU(NW8,NW27)=SKU(K,NW27)
 5003 CONTINUE
C
C     CENTRAL SECTION
C
C      DOUBLE END VALUES OF LOADING.
      DO 5004 L=1,NW2
      WU(L)=2.*WU(L)
      WV(L)=2.*WV(L)
      WW(L)=2.*WW(L)
      WTHETA(L)=2.*WTHETA(L)
 5004 CONTINUE
C     SUBSTITUTE INTO  CENTRAL  SECTION OF SKU( ,NW27)
      NW9=(MSKD-2*NW1)/NW1
      DO 5005 M=1,NW9
C     STARTING ADDRESS IN SKU( ,NW27)
      NW10=NW1+1
      DO 5006 N=1,NW2
      NW15=NW10+(M-1)*4*NW2+(N-1)*4
      NW16=NW15+1
      NW17=NW15+2
      NW18=NW15+3
      SKU(NW15,NW27)=WU(N)
      SKU(NW16,NW27)=WV(N)
      SKU(NW17,NW27)=WW(N)
      SKU(NW18,NW27)=WTHETA(N)
 5006 CONTINUE
 5005 CONTINUE
C
C     LIVE LOADS
C
C     READ NUMBER OF NODES HAVING LIVE LOADS.
      READ(8,5007) NW19
 5007 FORMAT(    )
C     NW20 IS NODE NUMBER,NW21 IS CROSS-SECTION NUMBER; OTHERS ARE
C     GLOBAL LIVE LOADS.
      DO 5011 I=1,NW19
      READ(8,5009)NW20(I),NW21(I),WLU(I),WLV(I),WLW(I),WLTHE(I)
 5009 FORMAT(     )
 5011 CONTINUE
C     COPY INTO SKU( ,NW27)
      I=1
 5016 NW22=(4*NW20(I)-3)+(NW21(I)-1)*NW1
      NW23=NW22+1
      NW24=NW22+2
      NW25=NW22+3
      SKU(NW22,NW27)=SKU(NW22,NW27)+WLU(I)
      SKU(NW23,NW27)=SKU(NW23,NW27)+WLV(I)
      SKU(NW24,NW27)=SKU(NW24,NW27)+WLW(I)
      SKU(NW25,NW27)=SKU(NW25,NW27)+WLTHE(I)
      I=I+1
      IF(I.GT.NW19) GO TO 5010
```

```
            GO TO 5016
      5010  CONTINUE
      C
      C     PUT LOADS AT SUPPORT TO ZERO.
      C
            DO 5015 K=1,MS310
            NW28=MSZERO(K)
            SKU(NW28,NW27)=0.
      5015  CONTINUE
            RETURN
            END
@PRT,S MENAI.GRAFT
```

```
ROJECT*MENAI(1).ECHO
              COMPILER(XM=3)
              SUBROUTINE ECHO (EDV,JE1)
              COMMON/EXT/SKU(440,133),SD4CA(440,5),SD2CG(440,3),SDCZE(440),
             .SD1CE(440,3),CA(44,44),CG(44,44),CE(44,44),
             .CZ(44,44),SIGMAX(180),SIGMAY(180),TAUXY(180),
             .AMX(180),AMY(180),AMXY(180)
              COMMON DDX(30),DDY(30),DD1(30),DDXY(30),DH(30),DB(30)
              COMMON FA(8,8,30),FG(8,8,30),FE(8,8,30),FZ(8,8,30)
              COMMON AFA(8,8,30),AFG(8,8,30),AFE(8,8,30),AFZ(8,8,30)
              COMMON SD4(6,5,6),SD2(6,3,6),SD1(6,3,6),SD(6,6)
              COMMON DEX,DEY,DPX,DPY,DG,IDNUM,DLDX,DLDXY,DLDY
              COMMON MHC,MSKD,MSKB,MS310,MSZERO(250)
              COMMON NW27,JENOX,JENUM,MS(31),ME(31)
              COMMON ANG(30),SA
              COMMON/B2/ATRANS
              DIMENSION JETRID(16),JEST(16),JEMID(16),JEEND(16)
              DIMENSION EDV(JE1)
              DOUBLE PRECISION ATRANS(8,8,30)
              DIMENSION EGDIS(8),ELDIS(8)
              DOUBLE PRECISION EDUX(3),EDVY(3),EDVX(3),EDUY(3),ED2WX2(3)
              DOUBLE PRECISION ED2WY2(3),ED2WXY(3)
      C       READ NUMBER OF TWO STRIP SUB-STRUCTURES:JENUM
              READ(6,7000) JENUM
      7000    FORMAT(    )
      C       FOR EACH SUB-STRUCTURE READ:ONE STRIP NUMBER WITH SAME
      C       ANGLE(JETRID); NUMBER IN SKU,ON FIRST CROSS-SECTION,WHERE
      C       FIRST NODE DISPLACEMENTS BEGIN (JEST);CONTINUE(JEMID);
      C       AND END(JEEND)
              DO 7001 I=1,JENUM
              READ(8,7000) JETRID(I),JEST(I),JEMID(I),JEEND(I)
      7001    CONTINUE
      C       NUMBER OF CROSS-SECTIONS IN STRUCTURE(JENOX)
              JENOX=3*MSKD/MSKB
      C
      C       IF FICTICOUS NODES USED,JENOX IS REDUCED
              IF(MS(1).EQ.6) JENOX=JENOX-1
              IF(ME(1).EQ.6) JENOX=JENOX-1
      C       SPACE TO ALLOW IN NEW DISPLACEMENT VECTOR EDV IS
      C       4*3*NUMBER OF SUB-STRUCTURES*NUMBER OF CROSS-SECTIONS
      C       IN COMPLETE STRUCTURE(THIS IS CALCULATED AND PLACED
      C       IN LIST IN MAIN PROGRAM).NUMBER IS JE1.
      C
      C       NUMBER OF DEGREES OF FREEDOM PER CROSS-SECTION IN NEW EDV
              JE2=JE1/JENOX
      C       NUMBER OF DEGREES OF FREEDOM IN CROSS-SECTION IN OLD SKU
              JE8=MSKB/3
              JA=0
              IF(MS(1).EQ.6) JA=1
      C
      C       PERFORM TRANSFORMATIONS (CALCULATE LOCAL DISPLACEMENTS)
      C
      C       FOR NUMBER OF CROSS-SECTIONS IN STRUCTURE(JENOX)
              DO 7002 J=1,JENOX
      C       FOR NUMBER OF TWO STRIP SUB-STRUCTURES(JENUM)
              DO 7003 K=1,JENUM
      C       LOAD GLOBAL DISPLACEMENTS INTO EGDIS
      C       FIRST FOUR
              DO 7004 L=1,4
              JE3=L
```

```
                JE4=JEST(K)+(J-1+JA)*JE9+L-1
                EGDIS(JE3)=SKU(JE4,NW27)
       7004     CONTINUE
       C        LAST FOUR
                DO 7005 M=1,4
                JE5=4+M
                JE6=JEMID(K)+(J-1+JA)*JE9+M-1
                EGDIS(JE5)=SKU(JE6,NW27)
       7005     CONTINUE
       C        MULTIPLY: ATRANS*EGDIS
                DO 7006 N=1,8
                DO 7006 I=1,8
                JE7=JETRID(K)
                ELDIS(N)=ELDIS(N)+ATRANS(N,I,JE7)*EGDIS(I)
       7006     CONTINUE
       C        COPY INTO EDV
                DO 7007 I2=1,8
                JE8=(J-1)*JE2+(12*K-11)+I2-1
                EDV(JE8)=ELDIS(I2)
       7007     CONTINUE
       C        CLEAR ELDIS AND EGDIS
                DO 7008 J2=1,8
                ELDIS(J2)=0.
                EGDIS(J2)=0.
       7008     CONTINUE
       C        LOAD LAST OF SET OF THE THREE INTO EGDIS
                DO 7009 K2=1,4
                JE10=JEEND(K)+(J-1+JA)*JE9+K2-1
                EGDIS(K2)=SKU(JE10,NW27)
       7009     CONTINUE
       C        MULTIPLY: ATRANS*EGDIS
                DO 7010 L2=1,4
                DO 7010 M2=1,4
                JE11=JETRID(K)
                ELDIS(L2)=ELDIS(L2)+ATRANS(L2,M2,JE11)*EGDIS(M2)
       7010     CONTINUE
       C        COPY INTO EDV
                DO 7011 N2=1,4
                JE12=(J-1)*JE2+12*K-3+N2-1
                EDV(JE12)=ELDIS(N2)
       7011     CONTINUE
       C        CLEAR ELDIS AND EGDIS
                DO 7012 I3=1,8
                ELDIS(I3)=0.
                EGDIS(I3)=0.
       7012     CONTINUE
       7003     CONTINUE
       7002     CONTINUE
       C        NUMBER OF CROSS-SECTIONS EXCLUDING FIRST AND LAST
                JE16=JENOX-2
       C        STRESS PACKAGES REQUIRED IN CENTRAL SECTION
                JE17=JE16*JENUM
       C
       C        STRESS CALCULATIONS FOR FIRST CROSS-SECTION
       C
                DO 7013 I=1,JENUM
       C        TO GET TO STARTING DISPLACEMENT IN EDV
                JESTDV=12*I-11
       C
       C        L.H.NODE
```

```
C     CALCULATE ED2WX2(2)
      JE272=JESTDV+6
      JE273=JESTDV+6+JE2
      JE274=JESTDV+6+2*JE2
      JE275=JESTDV+6+3*JE2
      ED2WX2(2)=(2.*EDV(JE272)-5.*EDV(JE273)+4.*EDV(JE274)
     .-1.*EDV(JE275))/(SA**2.)
C      CALCULATE ED2WY2(2)
      JE47=JESTDV+2
      JE48=JESTDV+6
      JE49=JESTDV+10
      ED2WY2(2)=(1.*EDV(JE47)-2.*EDV(JE48)+1.*EDV(JE49))/(DB(JE23)**2)
C     CALCULATE ED2WXY(2)
      JE50=JESTDV+7
      JE51=JESTDV+7+JE2
      JE52=JESTDV+7+2*JE2
      ED2WXY(2)=(-3.*EDV(JE50)+4.*EDV(JE51)-1.*EDV(JE52))/
     .(2.*SA)
C
C     R.H. NODE
C
C     CALCULATE EDUX(3)
      JE54=JESTDV+8
      JE55=JE54+JE2
      JE56=JE54+JE2*2
      EDUX(3)=(-3.*EDV(JE54)+4.*EDV(JE55)-1.*EDV(JE56))/(2.*SA)
C     CALCULATE EDVY(3)
      JE57=JESTDV+1
      JE58=JESTDV+5
      JE59=JESTDV+9
      EDVY(3)=(1.*EDV(JE57)-4.*EDV(JE58)+3.*EDV(JE59))/(2.*DB(JE23))
C     CALCULATE EDVX(3)
      JE60=JESTDV+9
      JE61=JE60+JE2
      JE62=JESTDV+9+2*JE2
      EDVX(3)=(-3.*EDV(JE60)+4.*EDV(JE61)-1.*EDV(JE62))/(2.*SA)
C      CALCULATE EDUY(3)
      JE63=JESTDV
      JE64=JESTDV+4
      JE65=JESTDV+8
      EDUY(3)=(1.*EDV(JE63)-4.*EDV(JE64)+3.*EDV(JE65))/(2.*DB(JE23))
C      CALCULATE ED2WX2(3)
      JE276=JESTDV+10
      JE277=JESTDV+10+JE2
      JE278=JESTDV+10+2*JE2
      JE279=JESTDV+10+3*JE2
      ED2WX2(3)=(2.*EDV(JE276)-5.*EDV(JE277)+4.*EDV(JE278)
     .-1.*EDV(JE279))/(SA**2.)
C     CALCULATE ED2WY2(3)
      JE253=JESTDV+3
      JE254=JESTDV+7
      JE255=JESTDV+11
      ED2WY2(3)=(1.*EDV(JE253)-4.*EDV(JE254)+3.*EDV(JE255))
     ./(DB(JE23)*2.)
C     CALCULATE ED2WXY(3)
      JE72=JESTDV+11
      JE73=JESTDV+11+JE2
      JE74=JESTDV+11+2*JE2
      ED2WXY(3)=(-3.*EDV(JE72)+4.*EDV(JE73)-1.*EDV(JE74))/
     .(2.*SA)
```

```
      JE122=JECDV+4
      JE123=JECDV+8
      EDUY(3)=(1.*EDV(JE121)-4.*EDV(JE122)+3.*EDV(JE123))/(2.*DB(JE82))
C     CALCULATE ED2WX2(3)
      JE124=JECDV+10-JE2
      JE125=JECDV+10
      JE126=JECDV+10+JE2
      ED2WX2(3)=(1.*EDV(JE124)-2.*EDV(JE125)+1.*EDV(JE126))/(SA**2)
C     CALCULATE ED2WY2(3)
      JE259=JECDV+3
      JE260=JECDV+7
      JE261=JECDV+11
      ED2WY2(3)=(1.*EDV(JE259)-4.*EDV(JE260)+3.*EDV(JE261))
     ./(DB(JE82)*2.)
C     CALCULATE ED2WXY(3)
      JE130=JECDV+11-JE2
      JE131=JECDV+11+JE2
      ED2WXY(3)=(-1.*EDV(JE130)+1.*EDV(JE131))/(2.*SA)
C
C     CALCULATE STRESSES
C
      DO 7017 M=1,3
      JE133=JENUM*3+(LS-1)*JENUM*3+3*(L-1)+M
      SIGMAX(JE133)=DLDY*EDUX(M)+DLDXY*EDVY(M)
      SIGMAY(JE133)=DLDXY*EDUX(M)+DLDY*EDVY(M)
      TAUXY(JE133)=DG*(EDVX(M)+EDUY(M))
C
      AMX(JE133)=-(DDX(JE82)*ED2WX2(M)+DD1(JE82)*ED2WY2(M))
      AMY(JE133)=-(DD1(JE82)*ED2WX2(M)+DDY(JE82)*ED2WY2(M))
      AMXY(JE133)=DDXY(JE82)*2.*ED2WXY(M)
 7017 CONTINUE
C      CLEAR DERIVATIVE VECTORS.
      DO 7018 N=1,3
      EDUX(N)=0.
      EDVY(N)=0.
      EDVX(N)=0.
      EDUY(N)=0.
      ED2WX2(N)=0.
      ED2WY2(N)=0.
      ED2WXY(N)=0.
 7018 CONTINUE
 7016 CONTINUE
 7022 CONTINUE
C
C     STRESS CALCULATIONS FOR LAST CROSS SECTION.
C
      DO 7019 I=1,JENUM
C     TO GET TO STARTING ADDRESS
      JEEDV=JE1-JE2+1+I*12-11-1
C
C      L.H. NODE.
C
C     CALCULATE EDUX(1)
      JE134=JEEDV-2*JE2
      JE135=JEEDV-JE2
      JE136=JEEDV
      EDUX(1)=(1.*EDV(JE134)-4.*EDV(JE135)+3.*EDV(JE136))/(2.*SA)
C     CALCULATE EDVY(1)
      JE137=JEEDV+1
      JE138=JEEDV+5
```

MENAI: TRANSLATIONAL SHELL PROGRAM

DATE: 23. IX. 76

PROJECT: EX. Nº 2

CLIENT: U.C.T.

GENERAL DESCRIPTION OF STRUCTURE: Thin walled roof structure.

LOADING CASE CONSIDERED. Uniformly distributed loading.

ALL UNITS USED ARE N.M.S.

YOUNGS MODULUS X:	2.524430+10
YOUNGS MODULUS Y:	2.524430+10
POISSON RATIO X:	.000
POISSON RATIO Y:	.000
SHEAR MODULUS G:	1.262215+10

STRIP NO	STRIP WIDTH	STRIP THICKNESS	STRIP ANGLE(RAD)
1	.381	.102	.0000000
2	.381	.102	.0000000
3	1.078	.102	.7853982
4	1.078	.102	.7853982
5	1.078	.102	.7853982
6	1.078	.102	.7853982
7	.762	.102	.0000000
8	.762	.102	.0000000
9	1.078	.102	-.7853982
10	1.078	.102	-.7853982

BOUNDARY CONDITION AT START IS SIMPLY SUPPORTED

BOUNDARY CONDITION AT END IS SIMPLY SUPPORTED

NODAL SPACING = 4.876

CROSS-SECTION	NODE	GLOBAL DISPLACEMENTS U	V	W	THETA(RAD)
1					
	1	6.713610-04	0.000000	0.000000	0.000000
	2	6.736274-04	0.000000	0.000000	0.000000
	3	6.780336-04	0.000000	0.000000	0.000000
	4	3.942237-04	0.000000	0.000000	0.000000
	5	1.244441-04	0.000000	0.000000	0.000000
	6	-1.434428-04	0.000000	0.000000	0.000000
	7	-4.223833-04	0.000000	0.000000	0.000000
	8	-4.204766-04	0.000000	0.000000	0.000000
	9	-4.308293-04	0.000000	0.000000	0.000000
	10	1.173533-04	0.000000	0.000000	0.000000
	11	5.679770-04	0.000000	0.000000	0.000000
2					
	1	5.360041-04	0.000000	1.713798-03	0.000000
	2	5.371815-04	-2.688409-06	1.781796-03	3.735671-04
	3	5.407186-04	-5.317646-06	2.023797-03	9.467559-04
	4	3.230387-04	-1.077437-03	3.088147-03	1.358996-03
	5	1.153687-04	-1.600011-03	3.607521-03	-1.344466-04

6	-3.092204-05	-9.148399-04	2.924230-03	-1.463429-03
7	-3.047033-04	6.769562-05	1.948491-03	-5.515919-04
8	-3.043383-04	7.297957-05	2.169852-03	1.065052-03
9	-3.122391-04	7.861925-05	3.674173-03	3.086015-03
10	1.020991-04	3.475441-03	7.062633-03	5.323881-03
11	5.179552-04	7.708732-03	1.129326-02	5.665945-03
4				
1	2.161337-04	0.000000	2.740214-03	0.000000
2	2.161803-04	-2.702747-05	2.809725-03	3.811459-04
3	2.162189-04	-5.318717-06	3.055387-03	9.571027-04
4	1.546967-04	-1.078331-03	4.120557-03	1.341143-03
5	9.352529-05	-1.562524-03	4.601414-03	-2.149166-04
6	3.250743-05	-8.186408-04	3.859541-03	-1.505669-03
7	-2.885323-05	1.062227-04	2.941575-03	-3.116564-04
8	-3.110686-05	1.110335-04	3.466108-03	1.629470-03
9	-3.361505-05	1.173849-04	5.535163-03	4.004687-03
10	6.704076-05	4.379402-03	9.783774-03	6.536443-03
11	1.676781-04	9.680850-03	1.508747-02	7.111088-03
5				
1	-1.112296-04	0.000000	2.496516-03	0.000000
2	-1.122719-04	-2.735372-05	2.568805-03	3.957494-04
3	-1.153928-04	-5.377217-06	2.822855-03	9.866008-04
4	-1.303382-05	-1.111512-03	3.921034-03	1.391016-03
5	7.027725-05	-1.027987-03	4.434011-03	-1.972275-04
6	1.575748-04	-8.835941-04	3.691461-03	-1.542335-03
7	2.516805-04	1.074070-04	2.707132-03	-4.570308-04
8	2.472406-04	1.127725-04	3.093359-03	1.422810-03
9	2.501421-04	1.185867-04	5.002349-03	3.802123-03
10	3.407303-05	4.241921-03	9.117022-03	6.469384-03
11	-1.843003-04	9.421050-03	1.429327-02	6.953793-03
6				
1	-2.685055-04	0.000000	1.241445-03	0.000000
2	-2.709534-04	-2.482995-05	1.314165-03	3.983864-04
3	-2.779759-04	-5.040679-06	1.570412-03	9.968251-04
4	-9.336151-05	-1.133907-03	2.692185-03	1.447604-03
5	5.792675-05	-1.716741-03	3.272541-03	-7.641179-05
6	2.134451-04	-1.038399-03	2.595548-03	-1.520252-03
7	3.844480-04	6.511901-05	1.497749-03	-8.350152-04
8	3.755572-04	6.997213-05	1.441305-03	6.416317-04
9	3.837785-04	7.539415-05	2.611291-03	2.653218-03
10	2.412792-05	3.167470-03	5.694932-03	4.951219-03
11	-3.364479-04	7.128214-03	9.653012-03	5.316571-03
7				
1	-9.687206-05	0.000000	0.000000	0.000000
2	-8.709126-05	1.168109-05	0.000000	2.646914-08
3	-8.774591-05	2.328058-05	0.000000	-2.657754-07
4	-9.307916-06	7.210835-05	0.000000	-2.904651-05
5	6.637952-05	8.966197-05	0.000000	-8.294943-08
6	1.424033-04	7.518926-05	0.000000	2.297917-05
7	2.200557-04	2.885482-05	0.000000	1.734194-05
8	2.110539-04	5.287794-06	0.000000	1.740755-06
9	2.034332-04	-2.065808-05	0.000000	-2.218002-05
10	5.761953-05	-7.618770-05	0.000000	-1.237761-05
11	-8.877468-05	-9.361292-05	0.000000	-1.185875-05
1	1.305823-04	0.000000	1.779773-04	0.000000
2	1.325089-04	-2.510152-06	2.547040-04	4.194171-04
3	1.383276-04	-5.095052-06	5.229632-04	1.038638-03
4	9.309613-05	-1.179551-03	1.690202-03	1.513097-03
5	7.581594-05	-1.798394-03	2.306359-03	-5.990875-05

	6	5.497119-05	-1.090598-03	1.599626-03	-1.026923-03
	7	2.176159-05	1.650049-04	3.494121-04	-1.140497-03
	8	1.739418-05	1.697676-04	6.391228-06	2.081192-04
	9	-1.003444-06	1.752176-04	8.222938-04	2.170722-03
	10	9.692964-05	2.895426-03	3.533882-03	4.458821-03
	11	1.947189-04	6.479163-03	7.114885-03	4.620774-03
8	1	7.395516-05	0.000000	5.921125-04	0.000000
	2	8.057816-05	-2.796535-06	6.726673-04	4.391561-04
	3	8.244223-05	-5.490869-06	9.515818-04	1.073939-03
	4	7.221228-05	-1.208018-03	2.145837-03	1.533066-03
	5	6.732393-05	-1.809234-03	2.743213-03	-1.314709-04
	6	6.306328-05	-1.027486-03	1.962777-03	-1.722522-03
	7	5.470070-05	2.417491-04	6.995927-04	-1.034909-03
	8	5.945087-05	2.469435-04	5.342406-04	5.768193-04
	9	5.775141-05	2.528171-04	1.742606-03	2.834215-03
	10	1.035688-04	7.615853-03	5.096596-03	5.445960-03
	11	1.508363-04	9.004011-03	9.481738-03	5.908505-03
9	1	-7.209888-05	0.000000	4.632338-04	0.000000
	2	-7.250740-05	-2.729806-06	5.372128-04	4.049672-04
	3	-7.405771-05	-5.401256-06	7.971692-04	1.009627-03
	4	-8.570914-06	-1.146247-03	1.930056-03	1.458784-03
	5	5.253231-05	-1.726191-03	2.505479-03	-1.012614-04
	6	1.128701-04	-1.008585-03	1.790371-03	-1.599344-03
	7	1.757097-04	1.744093-04	6.136251-04	-9.595465-04
	8	1.902561-04	1.795689-04	4.524039-04	4.892519-04
	9	2.039342-04	1.852685-04	1.492901-03	2.464910-03
	10	9.529712-05	3.115024-03	4.414315-03	4.717124-03
	11	-1.771573-05	6.887735-03	8.187890-03	5.062229-03
10	1	-1.515687-04	0.000000	0.000000	0.000000
	2	-1.526485-04	0.000000	0.000000	0.000000
	3	-1.553398-04	0.000000	0.000000	0.000000
	4	-5.055395-05	0.000000	0.000000	0.000000
	5	4.552607-05	0.000000	0.000000	0.000000
	6	1.400456-04	0.000000	0.000000	0.000000
	7	2.405376-04	0.000000	0.000000	0.000000
	8	2.594505-04	0.000000	0.000000	0.000000
	9	2.353690-04	0.000000	0.000000	0.000000
	10	9.942328-05	0.000000	0.000000	0.000000
	11	-1.073899-04	0.000000	0.000000	0.000000

CROSS-SECTION	SUB-STRUCTURE	NODE	SIGMA-X	SIGMA-Y	LOCAL STRESSES TAU-XY	M-X	M-Y	M-XY
1								
	1							
		1	-2.27023+05	0.00000	-2.00068+02	9.82639+00	0.00000	0.00000
		2	-2.28670+05	0.00000	1.05817+05	2.24309+01	0.00000	2.54313+02
		3	-2.33650+05	0.00000	2.14029+05	6.76338+01	0.00000	6.47834+02
	2							
		1	-2.33650+05	0.00000	1.17679+06	4.85342+01	0.00000	6.47834+02
		2	-1.17040+05	0.00000	1.33040+06	3.36289+02	0.00000	9.37402+02
		3	-1.39345+04	0.00000	1.49662+06	4.94244+02	0.00000	-7.39123+01
	3							
		1	-1.17040+05	0.00000	1.40676+06	3.36289+02	0.00000	9.37402+02
		2	-1.39345+04	0.00000	1.42026+06	4.94244+02	0.00000	-7.39123+01
		3	8.76323+04	0.00000	1.44765+06	3.35825+02	0.00000	-9.95366+02
	4							
		1	-1.39345+04	0.00000	1.49605+06	4.94244+02	0.00000	-7.39123+01
		2	8.76323+04	0.00000	1.37185+06	3.35925+02	0.00000	-9.95366+02
		3	1.99298+05	0.00000	1.26081+06	4.79734+01	0.00000	-4.33742+02
	5							
		1	1.99298+05	0.00000	3.46126+05	6.41591+01	0.00000	-4.33742+02
		2	1.04000+05	0.00000	1.63338+05	7.34140+00	0.00000	6.02235+02
		3	2.00227+05	0.00000	-1.79390+04	1.15735+02	0.00000	1.90907+03
	6							
		1	2.00227+05	0.00000	-1.79835+06	8.46350+01	0.00000	1.90907+03
		2	-2.77092+04	0.00000	-1.75055+06	6.22583+02	0.00000	3.35580+03
		3	-2.55985+05	0.00000	-1.71806+06	1.30474+03	0.00000	3.56036+03
2								
	1							
		1	-1.17971+06	-1.80089+05	-8.03886+01	6.45436+01	-1.60406+03	0.00000
		2	-1.18416+06	-1.76168+05	7.45955+04	7.07365+01	-2.67602+03	8.72529+01
		3	-1.19755+06	-1.72248+05	1.49384+05	9.31660+01	-3.94342+03	2.19103+02
	2							
		1	-1.19755+06	-1.66503+05	1.84092+05	6.62313+01	-2.82698+03	2.19103+02
		2	-5.20046+05	-9.03278+04	2.94119+05	2.07970+02	1.48580+03	3.07019+02
		3	-8.00374+04	-1.51525+04	4.08272+05	2.82257+02	5.06606+03	-4.91993+01
	3							
		1	-6.20046+05	-9.54659+04	3.45056+05	2.07970+02	3.26310+03	3.07018+02
		2	-8.00374+04	-1.05144+04	3.57334+05	2.32257+02	3.27425+03	-4.91993+01
		3	4.55470+05	7.44372+04	3.74503+05	1.99191+02	2.58193+03	-3.44682+02
	4							
		1	-8.00374+04	-7.91632+03	4.08073+05	2.82257+02	5.07250+03	-4.91993+01
		2	4.55470+05	7.19391+04	3.23805+05	1.93191+02	8.01341+02	-3.44682+02
		3	1.01922+06	1.51595+05	2.44051+05	6.14983+01	-4.20862+03	-7.13452+01
	5							
		1	1.01922+06	1.69160+05	2.10003+05	8.97105+01	-4.14405+03	-7.13452+01
		2	1.00793+06	1.80944+05	8.20761+04	8.20286+01	-4.93271+03	3.73023+02
		3	1.02772+06	1.92729+05	-4.54103+04	1.70254+02	-6.51315+03	9.16764+02
	5							
		1	1.02772+06	1.83956+05	-1.15908+05	1.23034+02	-6.59747+03	9.16764+02
		2	-1.30240+05	9.13139+04	-9.04414+04	4.58566+02	-2.28013+03	1.51923+03
		3	-1.29421+06	-1.31640+03	-7.01490+04	8.76791+02	1.25464+03	1.62789+03
3								
	1							
		1	-1.67545+06	-1.81953+05	2.03683+01	1.19261+02	-1.66257+03	0.00000
		2	-1.68119+06	-1.76204+05	3.00615+03	1.13142+02	-2.70906+03	5.07803+00

		3	-1.69843+06	-1.70454+05	6.03641+03	1.13698+02	-3.94554+03	9.12113+00
	2	1	-1.69843+06	-1.57204+05	7.09242+03	8.39285+01	-2.80422+03	9.12113+00
		2	-8.83065+05	-9.25464+04	1.22015+04	7.96514+01	1.59359+03	7.33004+00
		3	-1.16725+05	-1.73995+04	1.72070+04	7.02697+01	5.23138+03	-1.20327+01
	3	1	-8.93065+05	-9.94593+04	1.39420+04	7.96514+01	3.49719+03	7.33004+00
		2	-1.16725+05	-1.09756+04	1.54664+04	7.02697+01	3.32919+03	-1.20827+01
		3	6.43447+05	7.75981+04	1.71843+04	6.25573+01	2.39833+03	-1.79947+01
	4	1	-1.16725+05	-7.20836+03	1.72012+04	7.02697+01	5.24593+03	-1.20827+01
		2	6.43447+05	7.37408+04	1.54495+04	6.25573+01	4.85051+02	-1.79947+01
		3	1.44027+06	1.54690+05	1.37092+04	7.90232+01	-5.04559+03	2.16472+01
	5	1	1.44027+06	1.73611+05	1.28642+04	1.15262+02	-5.05109+03	2.16472+01
		2	1.42799+06	1.84897+05	8.75306+03	1.56716+02	-5.93836+03	8.18991+01
		3	1.45580+06	1.96183+05	4.75230+03	2.24773+02	-7.59463+03	1.63933+02
	6	1	1.45580+06	1.86021+05	3.19323+03	1.61432+02	-7.68377+03	1.63933+02
		2	-1.76094+05	9.23985+04	9.55191+02	2.94750+02	-2.83162+03	2.62232+02
		3	-1.81788+06	-1.22406+03	-1.45782+03	4.52840+02	1.25065+03	2.94813+02
4	1	1	-1.25455+06	-1.84339+05	-9.99534+01	9.49657+01	-1.74728+03	0.00000
		2	-1.26077+06	-1.78142+05	-6.86763+04	9.51860+01	-2.79534+03	3.94675+00
		3	-1.27955+06	-1.71945+05	-1.37462+05	9.57674+01	-4.03367+03	9.03335+00
	2	1	-1.27955+06	-1.68535+05	-1.67295+05	6.77440+01	-2.89048+03	9.03335+00
		2	-6.58956+05	-9.48336+04	-2.71144+05	6.90590+01	1.59594+03	2.43712+01
		3	-9.21514+04	-2.10318+04	-3.74895+05	6.44554+01	5.32139+03	3.17069+01
	3	1	-6.58956+05	-1.02159+05	-3.17127+05	6.90590+01	3.49978+03	2.43712+01
		2	-9.21514+04	-1.37062+04	-3.28912+05	6.44554+01	3.41829+03	3.17069+01
		3	4.68380+05	7.47468+04	-3.42085+05	5.55384+01	2.57434+03	-3.33822+00
	4	1	-9.21514+04	-9.46019+03	-3.75054+05	6.44554+01	5.33203+03	3.17069+01
		2	4.68380+05	7.05007+04	-2.95943+05	5.58384+01	6.63429+02	-3.33822+00
		3	1.06936+06	1.50462+05	-2.17306+05	6.18455+01	-4.77329+03	-1.19903+02
	5	1	1.06936+06	1.70320+05	-1.87551+05	9.15447+01	-4.77577+03	-1.19903+02
		2	1.05299+06	1.85187+05	-6.66643+04	1.20124+02	-5.85470+03	-2.26127+02
		3	1.08099+06	2.00053+05	5.45177+04	1.74486+02	-7.70240+03	-3.09382+02
	6	1	1.08099+06	1.91324+05	1.21127+05	1.26328+02	-7.78395+03	-3.09382+02
		2	-1.11085+05	9.55220+04	9.41637+04	2.44822+02	-2.87626+03	-3.85786+02
		3	-1.30499+06	-2.80115+02	6.71020+04	3.90347+02	1.25708+03	-4.10806+02
5	1	1	6.30500+04	-1.62044+05	6.45097+02	-1.27945+03	-1.74823+03	0.00000
		2	6.51373+04	-1.66993+05	-1.38212+05	5.58922+00	-2.82250+03	-9.05900+01
		3	7.15675+04	-1.71341+05	-2.77297+05	2.39565+01	-4.09263+03	-2.25916+02
	2	1	7.15675+04	-1.55666+05	-3.52786+05	2.29698+01	-2.97835+03	-2.25916+02
		2	2.14494+04	-7.92388+04	-5.38813+05	1.78721+02	1.47720+03	-3.25085+02
		3	-1.01333+04	-2.81172+03	-7.57541+05	2.65996+02	5.20034+03	4.28415+01
	3	1	2.14494+04	-7.27047+04	-6.44521+05	1.78721+02	3.23914+03	-3.25085+02
		2	-1.01933+04	-9.34586+03	-6.51832+05	2.65996+02	3.42118+03	4.28416+01

		3	-3.92734+04	5.40130+04	-6.93343+05	1.83785+02	2.90706+03	3.58267+02
	4	1	-1.01973+04	-1.38497+04	-7.55763+05	2.65996+02	5.19466+03	4.28416+01
		2	-3.92734+04	5.85168+04	-5.89412+05	1.83785+02	1.14919+03	3.58267+02
		3	-8.17614+04	1.30883+05	-4.56641+05	1.95462+01	-3.62365+03	1.07679+02
	5	1	-8.17614+04	1.51355+05	-3.86959+05	2.70769+01	-3.54286+03	1.07679+02
		2	-9.36738+04	1.70202+05	-1.44664+05	-1.97889+01	-4.71533+03	-3.25315+02
		3	-1.20912+05	1.89049+05	9.39707+04	2.06775+01	-6.67672+03	-8.75470+02
	6	1	-1.20912+05	1.87220+05	2.45156+05	1.81309+01	-6.76017+03	-8.75470+02
		2	6.09531+04	3.18094+04	1.75550+05	2.94933+02	-2.36784+03	-1.48382+03
		3	2.47280+05	-3.60111+03	1.46113+05	6.60089+02	1.24457+03	-1.59460+03
3	1	1	1.03309+06	7.76669+05	8.13528-01	-1.33281+02	-1.08884+00	0.00000
		2	1.04418+06	7.71262+05	-1.44933+04	-1.47313+02	0.00000	4.81440+00
		3	1.07765+06	7.65856+05	-2.89883+04	-1.96563+02	2.64614+00	9.61763+00
	2	1	1.07765+06	1.06746+06	-3.56443+04	-1.42756+02	1.19395+02	9.61763+00
		2	5.13992+05	5.49599+05	-5.07245+04	-4.54151+02	-4.24830+01	1.49928+01
		3	4.63084+04	3.17358+04	-7.78839+04	-6.15711+02	-1.19773+02	3.77792+00
	3	1	5.13992+05	5.55827+05	-6.67192+04	-4.54151+02	-6.60920+01	1.49928+01
		2	4.63084+04	2.55081+04	-6.78092+04	-6.15711+02	-4.35040+01	3.77792+00
		3	-4.10230+05	-5.04810+05	-6.91286+04	-4.29882+02	-4.16491+01	-2.44194+01
	4	1	4.63084+04	2.41453+04	-7.79126+04	-6.15711+02	-8.16188+01	3.77792+00
		2	-4.10230+05	-5.03448+05	-5.91052+04	-4.29882+02	-4.32812+01	-2.44194+01
		3	-9.38855+05	-1.03104+06	-4.03383+04	-1.11196+02	5.38169+01	-6.99318+01
	5	1	-9.38855+05	-7.41348+05	-3.22584+04	-1.73444+02	1.59418+01	-6.99318+01
		2	-9.27436+05	-8.20156+05	-8.83614+03	-1.35935+02	0.00000	-9.92523+01
		3	-9.96065+05	-8.96965+05	1.47393+04	-7.22405+02	8.81280+01	-1.10454+02
	6	1	-9.96065+05	-1.23499+06	2.47609+04	-2.47358+02	-2.99128+01	-1.10454+02
		2	1.88457+05	-6.04023+05	1.81428+04	-1.32542+03	-5.17613+01	-1.12721+02
		3	1.37499+06	2.69409+04	1.14095+04	-2.02922+03	8.53825+00	-1.13499+02
7	1	1	4.31856+05	-1.63821+05	-6.43380+02	-2.21747+01	-1.87160+03	0.00000
		2	4.34034+05	-1.63814+05	1.09558+05	-1.93293+01	-2.94561+03	1.00527+02
		3	4.40554+05	-1.73806+05	2.19987+05	8.85875+00	-4.21543+03	2.45887+02
	2	1	4.40554+05	-1.56988+05	2.82459+05	8.12137+00	-3.10202+03	2.45887+02
		2	2.12320+05	-8.19928+04	4.26399+05	1.53181+02	1.50334+03	3.57603+02
		3	2.54815+03	-6.99738+03	6.02901+05	2.48766+02	5.37744+03	-3.00777+01
	3	1	2.12320+05	-7.55460+04	5.12550+05	1.63181+02	3.26378+03	3.57603+02
		2	2.54815+03	-1.34442+04	5.16749+05	2.48766+02	3.59914+03	-3.00777+01
		3	-2.05382+05	4.80576+04	5.55029+05	1.63690+02	3.23896+03	-3.99585+02
	4	1	2.54815+03	-1.82106+04	6.01055+05	2.48760+02	5.37142+03	-3.00777+01
		2	-2.05382+05	5.34240+04	4.70724+05	1.53690+02	1.48241+03	-3.99585+02
		3	-4.29146+05	1.25059+05	3.73889+05	-3.99533+00	-3.13361+03	-2.39969+02
	5	1	-4.28146+05	1.46401+05	3.19451+05	-7.21715-02	-3.05167+03	-2.39969+02
		2	-3.05033+05	1.69167+05	1.24191+05	-4.09637+01	-4.45502+03	1.31648+02

		3	-3.77116+05	1.91934+05	-6.71197+04	-9.20367+00	-5.64930+03	6.53894+02
	6							
		1	-3.77116+05	1.91504+05	-1.96961+05	1.34505+00	-6.73292+03	5.53894+02
		2	1.18946+05	9.40089+04	-1.39603+05	2.80346+02	-2.35404+03	1.24954+03
		3	6.20394+05	-3.48653+03	-1.22636+05	6.50410+02	1.24487+03	1.35533+03
8								
	1							
		1	-5.24588+05	-1.88414+05	8.17294+01	5.09378+01	-2.00038+03	0.00000
		2	-5.30806+05	-1.82172+05	4.09126+04	5.19639+01	-3.05073+03	-3.30790+00
		3	-5.49812+05	-1.75931+05	8.19156+04	5.47453+01	-4.29174+03	-6.68695+00
	2							
		1	-5.49812+05	-1.71494+05	9.96258+04	3.87440+01	-3.15010+03	-5.68695+00
		2	-2.77675+05	-1.00213+05	1.61489+05	5.05780+01	1.62778+03	-1.24335+01
		3	-6.01432+04	-2.89329+04	2.23270+05	5.09563+01	5.64620+03	-9.46657+00
	3							
		1	-2.77675+05	-1.07724+05	1.83035+05	5.05786+01	3.52322+03	-1.24335+01
		2	-6.01472+04	-2.14223+04	1.95674+05	5.05568+01	3.75018+03	-9.46657+00
		3	1.49379+05	6.48794+04	2.03424+05	3.26237+01	3.21895+03	5.31342+00
	4							
		1	-6.01432+04	-1.75035+04	2.23366+05	5.09568+01	5.65441+03	-9.46657+00
		2	1.49879+05	6.09606+04	1.75732+05	3.26233+01	1.31797+03	6.31342+00
		3	3.98514+05	1.39425+05	1.28436+05	1.93918+01	-3.78346+03	4.14237+01
	5							
		1	3.98514+05	1.60830+05	1.11145+05	4.09534+01	-3.77611+03	4.14237+01
		2	4.47734+05	1.83336+05	3.79501+04	5.72481+01	-5.28165+03	6.43577+01
		3	5.42428+05	2.05941+05	-3.54300+04	1.39862+02	-7.55938+03	6.73464+01
	6							
		1	5.42428+05	1.99687+05	-7.68678+04	8.73214+01	-7.63403+03	5.73464+01
		2	-4.22598+03	9.98633+04	-5.92802+04	2.30124+02	-2.79326+03	5.91314+01
		3	-5.49914+05	3.90185+01	-4.16172+04	4.18680+02	1.25721+03	5.52743+01
9								
	1							
		1	-5.99329+05	-1.82813+05	7.84903+01	3.13952+01	-1.78784+03	0.00000
		2	-6.07777+05	-1.78938+05	-2.94900+04	3.77246+01	-2.86018+03	-1.00533+02
		3	-6.15379+05	-1.75064+05	-5.91805+04	6.03534+01	-4.12803+03	-2.45926+02
	2							
		1	-6.15379+05	-1.68542+05	-7.35504+04	4.30284+01	-3.01062+03	-2.45326+02
		2	-3.17796+05	-9.50459+04	-1.16900+05	1.85821+02	1.51784+03	-3.50954+02
		3	-5.64254+04	-2.15501+04	-1.64139+05	2.59799+02	5.31118+03	3.00367+01
	3							
		1	-3.17796+05	-9.98529+04	-1.37263+05	1.35821+02	3.29489+03	-3.50354+02
		2	-5.64264+04	-1.67432+04	-1.43827+05	2.59799+02	3.51839+03	3.00967+01
		3	1.99279+05	6.63605+04	-1.55101+05	1.73137+02	3.03825+03	3.94324+02
	4							
		1	-5.64264+04	-1.45105+04	-1.63832+05	2.59799+02	5.31610+03	3.00367+01
		2	1.99279+05	6.41338+04	-1.35155+05	1.73137+02	1.25792+03	3.94324+02
		3	4.81218+05	1.42778+05	-1.10815+05	2.79252+01	-3.53866+03	2.36914+02
	5							
		1	4.81218+05	1.61354+05	-9.95385+04	4.95458+01	-3.47291+03	2.36914+02
		2	5.20313+05	1.79877+05	-4.77616+04	7.47954+01	-4.02035+03	-1.32047+02
		3	5.30771+05	1.97901+05	3.13605+03	1.16733+02	-6.55992+03	-5.43816+02
	6							
		1	5.90771+05	1.91149+05	3.62924+04	9.03590+01	-6.63890+03	-5.48316+02
		2	-3.66176+04	9.50172+04	3.06353+04	4.21402+02	-2.29641+03	-1.24670+03
		3	-6.68582+05	-1.11458+03	3.05036+04	8.40404+02	1.26005+03	-1.35261+03
10								
	1							

	1	-2.23853+05	0.00000	1.97838+02	1.18026+01	0.00000	0.00000
	2	-2.25464+05	0.00000	-6.12236+04	2.34852+01	0.00000	-2.70292+02
	3	-2.30353+05	0.00000	-1.22836+05	6.59615+01	0.00000	-6.78680+02
2							
	1	-2.30353+05	0.00000	-7.44984+05	4.73129+01	0.00000	-6.78680+02
	2	-1.16918+05	0.00000	-8.31827+05	3.21064+02	0.00000	-9.84341+02
	3	-1.66374+04	0.00000	-9.31000+05	4.68642+02	0.00000	6.26275+01
3							
	1	-1.16918+05	0.00000	-3.76979+05	3.21064+02	0.00000	-9.84841+02
	2	-1.66374+04	0.00000	-3.85888+05	4.68642+02	0.00000	6.26275+01
	3	8.21123+04	0.00000	-9.08442+05	3.13650+02	0.00000	1.07018+03
4							
	1	-1.66374+04	0.00000	-9.30337+05	4.68642+02	0.00000	6.26275+01
	2	3.21123+04	0.00000	-9.63393+05	3.13650+02	0.00000	1.07019+03
	3	1.90665+05	0.00000	-8.10700+05	3.64587+01	0.00000	6.41734+02
5							
	1	1.90665+05	0.00000	-3.41280+05	5.81383+01	0.00000	6.41734+02
	2	1.95126+05	0.00000	-2.34264+05	1.23427+01	0.00000	-3.15955+02
	3	2.11030+05	0.00000	-1.29204+05	1.23605+02	0.00000	-1.60828+03
5							
	1	2.11030+05	0.00000	1.12383+06	9.33966+01	0.00000	-1.60828+03
	2	-2.42032+04	0.00000	1.09735+06	6.12680+02	0.00000	-3.07272+03
	3	-2.59950+05	0.00000	1.08660+06	1.26213+03	0.00000	-3.28282+03

&FIN

```
PUNID: BARKER   ACCT: AC005-3033     PROJECT: PROJECT

TIME:     TOTAL: 00:02:31.799

            CPU:  00:02:27.125      I/O:  00:00:02.257

          CC/ER:  00:00:02.416     WAIT:  00:00:00.000

IMAGES READ:  76         PAGES:  11

START:   09:15:44 SEP 22,1976    FIN:  09:18:47 SEP 22,1976
```

APPENDIX F:

MENAI - USERS' MANUAL

The program is liberally documented and the READ statements are explained in the various sub-routines. All data is entered in FREE FORMAT, units being combinations of Newtons and metres.

MAIN PROGRAM: Calls the sub-routine in a specific sequence, and monitors all the output. No data is read by this program.

Dimension parameters indicating the size of large matrices must be changed to suit each structure:

Sequence No.	Parameter
25	MHC
26	MSKD
27	MSKB
28	NW27
29	JE1

where:

- MHC = Size of "Stiffness" matrix for 1 cross-section (= Degrees of freedom per section).
- MSKD = Size of Diagonal in total Stiffness matrix (= Degrees of freedom in complete structure).
- MSKB = Half Band Width in Stiffness matrix (= 3 × MHC)
- NW27 = Column in Stiffness matrix for placing the load vector. (= MSKB + 1).
- JE1 = Size of displacement vector to accommodate transformed global displacements. (= (Number of 2-strip sub-structures) × 12 × (Number of cross-sections in Structure)).

The common storage block "EXT" must be changed for each different structure as indicated:

```
COMMON/EXT/SKU(MSKD,NW27),SD4CA(MSKD,5),SD2CG(MSKD,3),SDOZE(MSKD),
.SD1CE(MSKD,3),CA(MHC,MHC),CG(MHC,MHC),CE(MHC,MHC),
```

.CZ(MHC,MHC),SIGMAX(JE1/4),SIGMAY(JE1/4),TAUXY(JE1/4),
.AMX(JE1/4),AMY(JE1/4),AMXY(JE1/4)

SUB-ROUTINE DEES: Formulates the elastic constants.

Sequence No.	Variables Read	
16	DEX,DEY,DPX,DPY	
19	IDNUM	
23	DH(I),DB(I)	(I = 1,IDUM)

where:

DEX	=	Young's Modulus in the X axis direction.
DEY	=	" " " " Y " " .
DPX	=	Poisson's Ratio in the X axis direction.
DPY	=	" " " " Y " " .
IDNUM	=	Number of strips in structure (Max. = 30).
DH(I)	=	Strip thickness.
DB(I)	=	Strip widths.

SUB-ROUTINE FORMA: Formulates the basic strip stiffness matrices. No data is read by this sub-program.

SUB-ROUTINE ANGLE: Compiles the transformation matrices for individual strips and produces global stiffness matrices.

Sequence No.	Variable Read	
27	ANG(I)	(I = 1,IDUM)

where:

ANG(I) = Strip angle to global co-ordinate directions. One strip angle per card, entry sequence same as thickness and width cards.

SUB-ROUTINE CONTAC: Formulates the combined cross-section stiffness matrices.

Sequence No.	Variable Read
20	LC3,LC4
38	LC5
49	LC6,LC7,LC8,LC11,LC12

where:

LC3 = Strip number.

LC4 = Row number where the L.H. top corner of the strip stiffness matrices belong in the combined matrices. Repeat for all'normal' strips. Place 0,0 after last entry to stop process.

LC5 = Number of strip stiffness matrices that require 'split' placing. (If split placing is not required, enter 0 and program automatically skips this process).

LC6 = Strip number.

LC7 = Row number in combined matrices where strip stiffness sub-matrices belong.

LC8 = Column number (ditto).

LC11 = Row number in sub-matrices where the copy process starts.

LC12 = Column number in sub-matrices where copy process starts.

Repeat step 49 four (4) times for every strip stiffness matrix that requires 'split' placing. Information given first for L.H. top section (4 × 4), then R.H. top section, then L.H. bottom section, and finally R.H. bottom section (4 × 4). Place 0,0,0,0,0 after last entry to stop process.

SUB-ROUTINE SILO: Formulates the Finite Difference equations, multiplies them with the combined cross-sectional stiffness matrices to give the upper half of the banded stiffness matrix.

Sequence No.	Variable Read
247	MS(I) (I = 1,16)
249	MS(I) (I = 17,31)
254	ME(I) (I = 1,16)
256	ME(I) (I = 17,31)
432	SA (Min No. nodes in longitudinal direction = 7)
453	MS 310
459	MS 401
464	MSZERO(I)

where:

MS(I) = Single digit numbers indicating boundary conditions at start of structure. 1 = Fixed, 3 = Guided, 7 = Simply Supported. (Zeroes to be used to fill field).

ME(I)	=	Ditto ... at end of structure.
SA	=	Longitudinal nodal spacing (metres).
MS310	=	Number of rows to be reduced to zero in stiffness and Load Matrix (Columns are automatically done as well)
MS401	=	Number of cards to be read with ten row numbers per card. (Zeroes are used to fill field).
MSZERO(I)	=	Vector of numbers entered on MS401 cards (Max. = 250)

SUB-ROUTINE WHEEL: Formulates the load matrix.

Sequence No.	Variable Read
46	WU(I), WV(I), WW(I), WTHETA(I) (I = 1, to number of nodes per cross-section)
100	NW19
105	NW10(I), NW21(I), WLU(I), WLV(I), WLW(I), WLTHE(I) (I = 1, NW19)

where:

WU(I)	=	Cross-section dead load in global X direction.
WV(I)	=	Half cross-section dead load (N/m) in global Y direction.
WW(I)	=	Half cross-section dead load (N/m) in global Z direction.
WTHETA(I)	=	Half cross-section twisting dead load about X axis (Nm/m).

One card is read for every nodal point on cross-section.

NW19	=	Number of nodes having global live loads. (Max. number = 50) If no live loads - use dummy.
NW20	=	Node number for load.
NW21	=	Cross-section number for load.
WLU(I)	=	Global live load in X direction (N).
WLV(I)	=	" " " " Y " (N/m).
WLW(I)	=	" " " " Z " (N/m).
WLTHE(I)	=	Global twisting live load about X axis (Nm/m).

One card read for each node having global live loads.

SUB-ROUTINE GRAFT: Solves the linear algebraic equations to give global displacements. No data is read by this sub-program.

SUB-ROUTINE ECHO: Solves stresses in local axes co-ordinates. The structure is now thought of as a combination of two-strip sub-structures to facilitate the application of the Finite Difference operator patterns. Sections may have an odd number of strips; this merely leads to a duplication of stress and moment calculations for a particular node or the cross-section. Strips belonging to the same sub-structure must have the same thickness and width.

Sequence No.	Variable Read
23	JENUM
30	JETRID(I), JEST(I), JEMID(I), JEEND(I) (I = 1, JENUM)

where:

JENUM	=	Number of two-strip sub-structures.
JETRID(I)	=	Strip number for transformation of displacements into local axis directions.
JEST(I)	=	Row number in SKU(,NW27) matrix, on first cross-section, where first node displacements begin.
JEMID(I)	=	Row number in SKU(,NW27) matrix, on first cross-section, where second node displacements begin.
JEEND(I)	=	Row number in SKU(,NW27) matrix, on first cross-section. where third node displacements begin.

SIGN CONVENTION FOR OUTPUT

STRESSES: (In local co-ordinate directions)

SIGMA - X:	Positive indicates tensile stress. (N/m^2)
SIGMA - Y:	Positive indicates tensile stress. (N/m^2)
TAU - XY :	Positive as indicated.
M - X :	Positive induces sagging of strip.
M - Y :	Positive induces sagging of strip.
M - XY :	Positive as indicated (= - (M - YX).

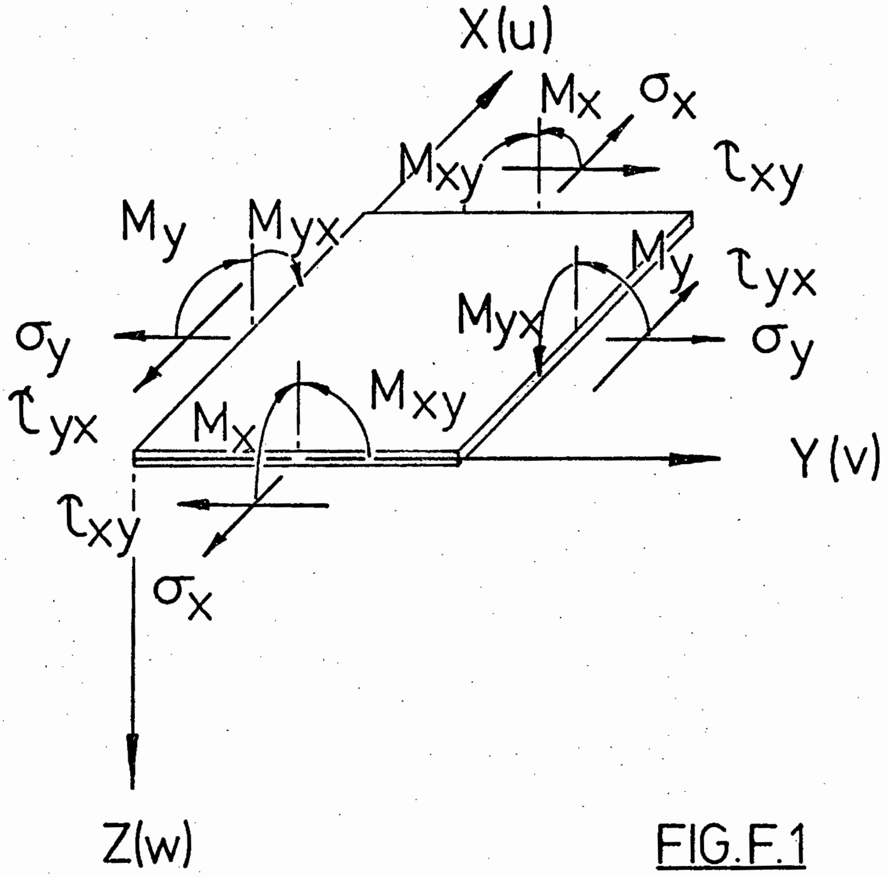


FIG.F.1