Biparabolic Isoparametric Shell Element
(A Version of Ahmad Element)

by

Riyadh Ahmed Al-Mustafa

A Thesis Presented to the

FACULTY OF THE COLLEGE OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

CIVIL ENGINEERING

March, 1979

BIPARABOLIC ISOPARAMETRIC SHELL ELEMENT

(A VERSION OF AHMAD ELEMENT)

BY

RIYADH AHMED AL-MUSTAFA

A Thesis Presented to the

FACULTY OF THE GRADUATE SCHOOL

UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the

Requirements for the Degree

MASTER OF SCIENCE IN CIVIL ENGINEERING

MARCH 1979

UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN, SAUDI ARABIA

THE GRADUATE SCHOOL

This thesis, written by Riyadh Ahmed Al-Mustafa under the direction of his Thesis Committee, and approved by all its members, has been presented to and accepted by the Dean of the Graduate School, in partial fulfilment of the requirements for the degree of Master of Science in Civil Engineering.

Dean of the Graduate School

Date 30/1/79

Department Chairman

Thesis Committee

Dr. Moneer F. Tewfik
Chairman

Dr. Baluch, M. H.
Member

Dr. Voyiadjis, G.Z.
Member

THIS THESIS IS DEDICATED TO

MY WIFE AND MY COUNTRY

ACKNOWLEDGEMENTS

Acknowledgement is due to the University of Petroleum and Minerals for support of this work.

I wish to express my appreciation to Dr. Moneer F. Tewfik who served as my major advisor. I also wish to thank the other members of my Thesis Committee, Dr. Baluch, M.H. and Dr. Voyiadjis,G.Z.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF FIGURES (CONTD.)

ABSTRACT

The element is developed based on the degeneration concept, in which the displacements and slopes of the shell mid-surface are independent variables with a penalty function imposition. Biparabolic interpolation is employed in conjuction with a reduced integration for evaluating the element properties. With the six degrees of freedom formulation, this element can be used for kinked shell problems. The element is tested to demonstrate its accuracy and versatility. The numerical examples indicate that the developed element performs accurately for both thick and thin shell structures. This element is capable of analyzing kinked shell structures and also representing deep beam and membrane structures.

INTRODUCTION

The mathematical solutions for the shell problems have serious limitations in practice because of the unusual geometries and/or boundary conditions of the shell problems. Various numerical procedures have been developed and employed to deal with such complexities.

More recently, the finite element method has come to the fore as another approach to the solution of plate and shell problems. The attemps to develop a 'finite' element for arbitrary shell structure have generally followed three distinct approaches.

In the first approach, the actual surface of the shell is replaced by an assemblege of flat faceted plate element which are either triangular or quadrelateral in shape. The flat element matrices is constructed by adding plane stress and plate bending element matrice. This approach has been successfully applied for cylindrical shells by Hrennikoff [1], and for general shells by Zienkiewicz and Cheung [2]; Clough and Johnson [3] and Carr [4]. However, this approach has the disadvantage that there is no coupling between membrane and bending stresses within each element, and consequently a large number of elements must be used to achieve satisfying accuracy.

In the second approach, the actual surface is replaced by an assemblege of curved elements formulated on the basis of classical shell theories. This approach of using curved shell element was impeded by certain difficulties. Gallagher [5] has outlined these difficulties as (1) the choice of appropriate shell theory; (2) description of the geometry of the element; (3) representation of rigid body modes of behavior; and (4) satisfaction of requirements related to the continuity of displacements. Many curved shell elements were developed which had some of these difficulties. A survey of some of those elements is represented in References [6] and [7].

In the third approach, the actual surface is replaced by an assemblege of three-dimensional isoparametric (solid) elements as shown in Figure (1). This element can be used in representing thin shell structures with certain assumptions. The basic assumptions are that no strain occurs across the thickness and that the local (r, s directions) stresses vary linearly also along the thickness. These assumptions enable the elimination of the nodes in the middle surface, Figure 2. This approach was also impeded by certain difficulties. Firstly, the retention of three degrees of freedom at each node leads to large stiffness coefficients for relative displacements along the thickness which may lead to ill-conditioned equations for small thicknesses. Secondly, using several nodes through the thickness ignores the well-known fact that even for thick shell the normals to the midsurface before deformation remain straight after deformation and consequently wastes computer time.

On the basis of the third approach, special elements have been degenerated by implying basically two constraints. These are the zero normal stress constraint and normals constraints. Consequently, the transverse shear strain energy is retained which permits the use of the element for thin and thick shell analysis. This is first developed by Ahmad [8]. Numerical integration is employed to evaluate the matrices defining element properties.

Early tests on Ahmad plate and shell elements showed that when full Gauss integration (3 X 3 X 2) was adopted overstiff results were obtained [8] in thin shell structures. Improvements were obtained when reduced integration (2 X 2 X 2) was used [9]. The eight noded reduced integration plate element of Ahmad was later found, in certain cases, to suffer from 'locking' phenomenon as thickness to span ratio decreases.

With the previous defects of Ahmad element in mind, and based on another approach of adopting six degrees of freedom [10] which an essential requirement for the analysis of kinked shells, a nine noded element will be developed and tested based on the same original formulation of Ahmad shell element.

Thus, it is the thesis theme trying to improve the original Ahmad shell element.

2. Biparabolic Isoparametric Shell Element

2.1 General

The basic concept of the finite element method is the idealization of the actual continuum as an assemblege of discrete structural 'finite' elements. In the present shell analysis, the actual surface of the shell is approximated (replaced) by a system of quadrilateral curved shell elements. The solution requires first the evaluation of individual element properties, i.e., element stiffness and load properties, etc. As a result, the complete assemblege properties of the system are derived by superposition of each element properties. Finally the analysis is accomplished by solving the simultaneous equations of the system.

It is important to state that the success of any finite element idealization which is actually a numerical approximation, depends on its convergence properties. For this to occur two criteria must be met by the assumed displacement function :

1. Completeness of the displacement field within the element. This completeness is equivalently expressed by stating that the displacement function should include 'rigid body' and 'constant strain' states.
2. Conformity between elements must be met. This is the compatibility requirement.

2.2 Isoparametric Formulation

The basic concepts of this formulation were first introduced by Irons [11] and were later developed and popularized by Zienkiewicz and his group at Swansea [12,13,14]. The name 'isoparametric' is derived from the use of the same interpolation function to define the geometry of the element and to define the displacements within the element. Many of these element have been widely used for two- and three-dimensional analysis due to their versatility, efficiency, and more important, their remarkable ease of programming.

Isoparametric coordinates are of the type of local curvilinear coordinates r,s, and t. They are established such that each coordinate varies from -1 and +1 between opposite faces of the element, Fig.1. The relation between the global coordinates at any point and curvilinear coordinates is defined by

$$\begin{Bmatrix} x \\ y \\ z \end{Bmatrix} = \Sigma N_i \begin{Bmatrix} x_i \\ y_i \\ z_i \end{Bmatrix} \qquad (2.1)$$

Where N_i are the shape functions for node i, and i ranges over all nodes in the element, and x_i, y_i, and z_i are global coordinates of node i. These shape functions take a value of unity at node i and zero at all other nodes :

$$\Sigma N_i = 1 \qquad (2.2)$$

Using the same shape functions as used in defining the coordinates, the displacement field in the element is now defined as

$$\begin{Bmatrix} u \\ v \\ w \end{Bmatrix} = \Sigma N_i \begin{Bmatrix} u_i \\ v_i \\ w_i \end{Bmatrix} \qquad (2.3)$$

That the isoparametric elements satisfy the criteria of convergence can be shown as follows.

To show the possibility of rigid body and constant strain states [15] we begin by considering u displacement given by

$$u = a_o + a_1 x + a_2 y + a_3 z \qquad (2.4)$$

where the a_i are constants. If the v- and w- displacement are given by similar linear expressions, we see that by suitable choice of the constants, we can have rigid body translation or rotation, or constant strains ε_x or γ_{xy}, etc. Because u,v, and w all use the same interpolation function, it is sufficient to show that u- displacement satisfy the completeness criteria. At node i, Eq. (2.4) yields

$$u_i = a_o + a_1 x_i + a_2 y_i + a_3 z_i \qquad (2.5)$$

By definition

$u = \Sigma N_i u_i$, so Eq.2.4 becomes

$$u = a_o \Sigma N_i + a_1 \Sigma N_i x_i + a_2 \Sigma N_i y_i + a_3 \Sigma N_i z_i \qquad (2.6)$$

But also by definition

$$x = \Sigma N_i x_i, \quad y = \Sigma N_i y_i, \quad z = \Sigma N_i z_i, \quad \Sigma N_i = 1$$

so that Eq. (2.6) reduces to Eq. (2.4). This shows that the isoparametric elements satisfy the completeness criteria.

To show that conformity criteria is peserved during deformation, we note that displacements along any edge are uniquely determined by displacements of nodes on that edge and by no other nodes. This may be seen by the substitution of r and s equal to -1 and +1 into the shape functions. Also, along a common edge (S_1 = +1, and S_2 = -1 as in Fig. 3), the shape functions of the two elements are identical. Since $r_1 = r_2$ along this edge, it follows that conformity is preserved.

2.3 Ahmed Shell Element (8S)

2.3.1 Geometric Definition and Displacement Field

Ahmed shell element evolves from 16-node three dimensional element shown in Fig.2. The parent element, i.e. the element from which 8S is derived has its nodes on the top and bottom surfaces, while the derived element nodes are located in mid-surface, Fig. 4. With normal constrained to be straight, Eq. 2.1, becomes

$$\begin{Bmatrix} x \\ y \\ z \end{Bmatrix} = \sum_{i=1}^{8} N_i(r,s)\frac{1+t}{2} \begin{Bmatrix} x_i \\ y_i \\ z_i \end{Bmatrix}_{top} + \sum_{i=1}^{8} N_i(r,s)\frac{1-t}{2} \begin{Bmatrix} x_i \\ y_i \\ z_i \end{Bmatrix}_{bottom} \tag{2.7}$$

where t having a value of -1, and +1 and N_i are shape function of the eight - node element in terms of r and s (Appendix A).
Eq. (2.7) defines coordinates x, y, and z on t=0 with a vector connecting

the top and bottom nodes which is equal to the shell thickness. Eq. 2.7 can be rewritten as

$$\begin{Bmatrix} x \\ y \\ z \end{Bmatrix} = \sum_{i=1}^{8} N_i(r,s) \begin{Bmatrix} x_i \\ y_i \\ z_i \end{Bmatrix}_{mid} + \Sigma N_i(r,s) \frac{t}{2} \overline{V}_{3i} \qquad (2.8)$$

where

$$\overline{V}_{3_i} = \begin{Bmatrix} x_i \\ y_i \\ z_i \end{Bmatrix}_{top} - \begin{Bmatrix} x_i \\ y_i \\ z_i \end{Bmatrix}_{bottom} \qquad (2.9)$$

defining a vector normal to shell mid-surface, Fig. 5.

At any point on the mid-surface,an orthogonal set of local coordinates $\overline{V}_{1_i}$, $\overline{V}_{2_i}$ and $\overline{V}_{3_i}$ is constructed. V_{1_i} and V_{2_i} are constructed in the following manner

$$\overline{V}_{1_i} = \overline{i} \times \overline{V}_{3_i} \qquad (2.10)$$

$$V_{2_i} = V_{1_i} \times V_{3_i} \qquad (2.11)$$

where

$$\overline{i} = \begin{Bmatrix} 1 \\ 0 \\ 0 \end{Bmatrix}$$

is a unit vector in x-direction. $\overline{v}_{1_i}$, $\overline{v}_{2_i}$ and v_{3_i} unit vectors are obtained by dividing $\overline{V}_{1_i}$, $\overline{V}_{2_i}$ and $\overline{V}_{3_i}$ by their lengths.

Element displacements are completely defined by mid-surface nodal displacements u_i, v_i and w_i in the x, y, and z global directions, and by rotations α'_{1_i} and α'_{2_i} of a line originally normal to the mid-surface about two orthogonal axes parallel to the mid-surface, $\overline{V}_{1_i}$ and $\overline{V}_{2_i}$ as shown in Fig. 5.

Because of rotations α'_{1_i} and α'_{2_i}, the relative displacements of a point lying on $\overline{V}_{3_i}$ are

$$u'_i = \frac{th_i\alpha'_{2i}}{2}, \quad v'_i = -\frac{th_i\alpha'_{1i}}{2} \tag{2.12}$$

where h_i is the shell thickness at node i. These displacements are in directions $\overline{V}_{1_i}$ and $\overline{V}_{2_i}$, respectively. Their components in global directions are

$$\begin{matrix} u_i \\ v_i \\ w_i \end{matrix}_{\text{relative}} = \frac{1}{2} th_i \left[v_{1_i} , v_{2_i} \right] \begin{Bmatrix} \alpha'_{2_i} \\ -\alpha'_{1_i} \end{Bmatrix} \tag{2.13}$$

Thus, the total displacements of mid-surface is given in the form

$$\begin{Bmatrix} u \\ v \\ w \end{Bmatrix} = \sum_{i=1}^{8} N_i(r,s) \begin{Bmatrix} u_i \\ v_i \\ w_i \end{Bmatrix} + \sum_{i=1}^{8} N_i(r,s) \begin{Bmatrix} u_i \\ v_i \\ w_i \end{Bmatrix}_{\text{relative}} \tag{2.14}$$

2.3.2 Strain and Stresses

The strains and stresses have to be defined in order to derive the basic element properties.

The shell formulation is based on the assumptions that (i) the deflections are small, (ii) normals to the shell mid-surface before deformation remain straight but not necessarily normal after deformation ('normals' constraint) and (iii) normal stresses are negligible (zero-normal stress constraint). With these basic assumptions in mind, the strain components in local directions at any point are

$$\varepsilon' = \left\{ \begin{array}{c} \varepsilon'_x \\ \varepsilon'_y \\ \gamma'_{x'y'} \\ \gamma'_{x'y'} \\ \gamma'_{y'z'} \end{array} \right\} = \left\{ \begin{array}{l} \partial u'/\partial x' \\ \partial v'/\partial y' \\ \partial u'/\partial y' + \partial v'/\partial x' \\ \partial w'/\partial x' + \partial u'/\partial z' \\ \partial w'/\partial z' + \partial v'/\partial y \end{array} \right\} \qquad (2.15)$$

where the local directions are erected so that the z' direction coincide with the normal to the shell mid-surface with the other two orthogonal directions x' and y' tangent to it.

By standard transformation, the global derivatives of displacements u, v, and w are now transformed to the local derivatives of displacements u', v', and w' by

$$\begin{bmatrix} \partial u'/\partial x' & \partial v'/\partial x' & \partial w'/\partial x' \\ \partial u'/\partial y' & \partial v'/\partial y' & \partial w'/\partial y' \\ \partial u'/\partial z' & \partial v'/\partial z' & \partial w'/\partial z' \end{bmatrix} = \theta^T \begin{bmatrix} \partial u/\partial x & \partial v/\partial x & \partial w/\partial x \\ \partial u/\partial y & \partial v/\partial y & \partial w/\partial y \\ \partial u/\partial z & \partial v/\partial z & \partial w/\partial z \end{bmatrix} \theta \qquad (2.16)$$

where $\theta = [\theta_{ij}]$ is the transformation matrix defined by $\theta = \bar{v}_1, \bar{v}_2, \bar{v}_3$. This transformation matrix, θ, can be also constructed from the Jacobian matrix as follows.

As Eq. 2.14 relates the global displacements u, v, w to the curvilinear coordinates, the derivatives of these displacements with respect to global coordinates are given by

$$\begin{bmatrix} \partial u/\partial x & \partial v/\partial x & \partial w/\partial x \\ \partial u/\partial y & \partial v/\partial y & \partial w/\partial y \\ \partial u/\partial z & \partial v/\partial z & \partial w/\partial z \end{bmatrix} = J^{-1} \begin{bmatrix} \partial u/\partial r & \partial v/\partial r & \partial w/\partial r \\ \partial u/\partial s & \partial v/\partial s & \partial w/\partial s \\ \partial u/\partial t & \partial v/\partial t & \partial w/\partial t \end{bmatrix} \qquad (2.17)$$

In this, J^{-1} is the inverse of the Jacobian matrix which is defined as

$$J = \begin{bmatrix} \partial x/\partial r & \partial y/\partial r & \partial z/\partial r \\ \partial x/\partial s & \partial y/\partial s & \partial z/\partial s \\ \partial x/\partial t & \partial y/\partial t & \partial z/\partial t \end{bmatrix} \qquad (2.18)$$

and is calculated from the coordinate definitions in Eq. (2.8).

A vector normal to the mid-surface can be formed as a vector product of two tangents to the mid-surface as

$$\overline{V}_3 = \begin{Bmatrix} \partial x(r,s)/\partial r \\ \partial y(r,s)/\partial r \\ \partial z(r,s)/\partial r \end{Bmatrix} \times \begin{Bmatrix} \partial x(r,s)/\partial s \\ \partial y(r,s)/\partial s \\ \partial z(r,s)/\partial s \end{Bmatrix} \tag{2.19}$$

Following the previous process of defining uniquely two perpendicular vectors and reducing them to unit magnitudes, θ transformation matrix is constructed

$$\theta = [\ \overline{v}_1,\ \overline{v}_2,\ \overline{v}_3] \tag{2.20}$$

For linear isotropic elastic material, the local stresses components corresponding to local strains, Eq. 2.15, are obtained from the usual elasticity matrix D. Thus

$$\sigma' = D\varepsilon' \tag{2.21}$$

where $\sigma = [\ \sigma_{x'}\ \ \sigma_{y'}\ \ \tau_{x'y'}\ \gamma_{x'z'}\ \ \gamma_{y'z'}]$ and

$$D = \begin{bmatrix} \lambda + 2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda+2\mu & 0 & 0 & 0 \\ 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & \mu/k_s & 0 \\ 0 & 0 & 0 & 0 & \mu/k_s \end{bmatrix} \tag{2.22}$$

in which λ is the plane-stress reduced Lame constant, i.e. $\lambda = vE/(1-v^2)$, where E is the elasticity modulus and ν is Poisson ratio,

μ is the shear modulus, and k_s is a shear displacement correction factor (taken as 1.2 in Eq. 2.22).

2.3.3 Element Properties

The standard form of the element stiffness matrix, which involves integrals over the volume of the element, as derived by the general finite element procedure [14] is

$$K^{ij} = \int_V (B^i)^T D (B^j)\, dV \tag{2.23}$$

in which B^i is the standard strain matrix relating the local strain components to nodal displacements, $\delta_i = [u_i \ v_i \ w_i \ \alpha_{1i}' \quad \alpha_{2i}']^T$, such that

$$\varepsilon' = B\delta = \sum_{i=1}^{8} B_i \delta_i \tag{2.24}$$

For the integral in Eq. 2.23, the change of coordinates is

$$\int_V (\quad)dV = \iiint(\quad)dxdydz = \int_{-1}^{+1}\int_{-1}^{+1}\int_{-1}^{+1} (\quad)|J|drdsdt$$

where $|J|$ is the determinant of the Jacobian matrix. Thus Eq.2.23 becomes

$$K^{ij} = \int_{-1}^{+1} \int_{-1}^{+1} \int_{-1}^{+1} (B^i(r,s))^T D(B^j(r,s)) \left|J(r,s,t)\right| drdsdt \tag{2.25}$$

and by this expression the basic formulation is completed.

2.4 Biparabolic Shell Element (9L)

2.4.1 Geometric Definitions and Displacements Field

The 9L element evolves from an eighteen-node three dimensional element. And as in Ahmad element, the 9L element is defined by the curvilinear coordinates (r,s,t). Consequently, the relationship between the global and local coordinates can be written in the form

$$\begin{Bmatrix} x \\ y \\ z \end{Bmatrix} = \sum_{i=1}^{9} N_i(r,s) \begin{Bmatrix} x_i \\ y_i \\ z_i \end{Bmatrix}_{mid} + \sum_{i=1}^{9} N_i(r,s) \frac{t}{2} \overline{V}_{3_i} \qquad (2.26)$$

Now different scheme will be used to generate the orthogonal vectors $\overline{V}_1$, $\overline{V}_2$, and $\overline{V}_3$. Eq. 2.19 defines the normal vector $\overline{V}_3$ as a vector product of two tangents to the mid-surface as

$$\overline{V}_3 = \begin{Bmatrix} \partial x(r,s)/\partial r \\ \partial y(r,s)/\partial r \\ \partial z(r,s)/\partial r \end{Bmatrix} X \begin{Bmatrix} \partial x(r,s)/\partial s \\ \partial y(r,s)/\partial s \\ \partial z(r,s)/\partial s \end{Bmatrix} \qquad (2.27)$$

Then

$$\overline{V}_2 = \overline{V}_3 \; X \begin{Bmatrix} \partial x(0,0)/\partial r \\ \partial y(0,0)/\partial r \\ \partial z(0,0)/\partial r \end{Bmatrix}; \qquad (2.28)$$

and finally

$$\overline{V}_1 = \overline{V}_2 X \; V_3 \qquad (2.29)$$

Dividing $\overline{V}_1$, $\overline{V}_2$, $\overline{V}_3$ by their lengths to get the unit vectors $\overline{v}_1$, $\overline{v}_2$, and $\overline{v}_3$.

Just as before the displacements field is given in the form

$$\begin{Bmatrix} u \\ v \\ w \end{Bmatrix} = \sum_{i=1}^{9} N_i(r,s) \begin{Bmatrix} u_i \\ v_i \\ w_i \end{Bmatrix} + \sum_{i=1}^{9} N_i(r,s) \begin{Bmatrix} u_i \\ v_i \\ w_i \end{Bmatrix}_{relative} \tag{2.30}$$

One of the objective of this thesis is to adopt a six-degree of freedom formulation to analyze kinked shell problems. Thus, the relative displacements, in Eq. 2.13, can be rewritten as

$$\begin{Bmatrix} u_i \\ v_i \\ w_i \end{Bmatrix}_{rel} = \tfrac{1}{2} th_i \, [\overline{v}_{1_i}, \overline{v}_{2_i}, \overline{v}_{3_i}] \begin{Bmatrix} \alpha_2' \\ -\alpha_1' \\ \alpha_3' \end{Bmatrix} \tag{2.31}$$

where α_3' has a value of zero.

For small rotations, the usual transformation from local α_i' to global α_i lead to [10]

$$\begin{Bmatrix} u_i \\ v_i \\ w_i \end{Bmatrix}_{rel} = \tfrac{1}{2} th_i \, \phi_i \, \alpha_i \tag{2.32}$$

where

$$\phi_i = \begin{bmatrix} 0 & \theta_{33} & -\theta_{23} \\ \theta_{33} & 0 & \theta_{13} \\ \theta_{23} & -\theta_{13} & 0 \end{bmatrix} \tag{2.33}$$

in which θ_{ij} denotes the direction cosine from global to local coordinates.

Substituting Eq. 2.32 in Eq. 2.30 yields

$$\begin{Bmatrix} u \\ v \\ w \end{Bmatrix} = \sum_{i=1}^{9} N_i(r,s) \begin{Bmatrix} u_i \\ v_i \\ w_i \end{Bmatrix} + \sum_{i=1}^{9} N_i(r,s)\tfrac{1}{2}th_i\phi_i \begin{Bmatrix} \alpha_{1_i} \\ \alpha_{2_i} \\ \alpha_{3_i} \end{Bmatrix} \tag{2.34}$$

where u_i, v_i, w_i, α_{1_i},α_{2_i},α_{3_i} are the displacements and rotations in the global direction in the mid-surface.

2.4.2 Element Properties

As discussed in sections 2.3.2 and 2.3.3, the element stiffness matrix has the form

$$K^{ij} = \int_{-1}^{+1} \int_{-1}^{+1} \int_{-1}^{+1} (B^i(r,s))^T D(B^j(r,s)) |J(r,s,t)| \, drdsdt \tag{2.35}$$

Noting that certain economy [10] in programming can be achieved by splitting the stiffness into two parts, the bending and membrane part, K_m, and the transverse shear part, K_s, Eq. 2.35 becomes

$$K^{ij} = K_m^{ij} + K_s^{ij}$$

where

$$K_m^{ij} = \int_{-1}^{+1}\int_{-1}^{+1}\int_{-1}^{+1} (B_m^i)^T D_m (B_m^j) \left|J(r,s,t)\right| drdsdt \qquad (2.36)$$

and

$$K_s^{ij} = \int_{-1}^{+1}\int_{-1}^{+1}\int_{-1}^{+1} (B_s^i)^T D_s (B_s^j) \left|J(r,s,t)\right| drdsdt \qquad (2.37)$$

and B_m, B_s, D_m, D_s are defined according to

$$\varepsilon_m' = [\varepsilon_{x'} \quad \varepsilon_{y'} \quad \gamma_{x'y'}]^T = \sum_{i=1}^{9} B_m^i \, \delta_i \qquad (2.38)$$

$$\varepsilon_s' = [\gamma_{x'z'} \quad \gamma_{y'z'}] = \sum_{i=1}^{9} B_s^i \, \delta_i \qquad (2.39)$$

$$D_m = \begin{bmatrix} \lambda + 2\mu & \lambda & 0 \\ \lambda & \lambda + 2\mu & 0 \\ 0 & 0 & \mu \end{bmatrix} \qquad (2.40)$$

and

$$D_s = \begin{bmatrix} \mu/K_s & 0 \\ 0 & \mu/K_s \end{bmatrix} \qquad (2.41)$$

Upon the formulation of the matrices B_m^i and B_s^i the element stiffness matrix is completely formulated. Substituting Eq.2.34 in 2.16 and the result in 2.15 to yield

$$\begin{Bmatrix} \varepsilon_m(r,s) \\ \varepsilon_s(r,s) \end{Bmatrix} = \sum_{i=1}^{9} \begin{bmatrix} B^i_{1_m} & B^i_{2_m} + t \; B^i_{3_m} \\ B^i_{1_s} & B^i_{2_s} + t \; B^i_{3_s} \end{bmatrix} \begin{Bmatrix} u_i \\ \alpha_i \end{Bmatrix} \tag{2.42}$$

in which

$$B^i_{1_m} = L_m \, (N_i)\theta^T \tag{2.43}$$

$$B_{1_s} = L_s \, (N_i)\theta^T \tag{2.44}$$

$$B_{2_m} = \tfrac{1}{2} \, h_i \; N_i \; L_m \; (t) \; \theta^T\phi_i \tag{2.45}$$

$$B_{2_s} = \tfrac{1}{2} \, h_i \; N_i \; L_s \; (t) \; \theta^T\phi_i \tag{2.46}$$

$$B_{3_m} = \tfrac{1}{2} \, h_i \; L_m \; (N_i) \; \theta^T\phi_i \tag{2.47}$$

and

$$B_{3_s} = \tfrac{1}{2} \, h_i \; L_s \; (N_i) \; \theta^T{}_{\phi i} \tag{2.48}$$

The operator L_m ( ) and L_s ( ) are gradient operators with respect to local coordinates which are defined as

$$L_m(f) = \begin{bmatrix} L_1 & 0 & 0 \\ 0 & L_2 & 0 \\ L_2 & L_1 & 0 \end{bmatrix} \tag{2.49}$$

and

$$L_s(f) = \begin{bmatrix} L_3 & 0 & L_1 \\ 0 & L_3 & L_2 \end{bmatrix} \tag{2.50}$$

in which

$$L_1 = \theta_{11} \frac{\partial f}{\partial x} + \theta_{21} \frac{\partial f}{\partial y} + \theta_{31} \frac{\partial f}{\partial z} \tag{2.51}$$

$$L_2 = \theta_{12} \frac{\partial f}{\partial x} + \theta_{22} \frac{\partial f}{\partial y} + \theta_{32} \frac{\partial f}{\partial z} \tag{2.52}$$

and

$$L_3 = \theta_{13} \frac{\partial f}{\partial x} + \theta_{23} \frac{\partial f}{\partial y} + \theta_{33} \frac{\partial f}{\partial z} \tag{2.53}$$

Physically B_1^i corresponds to the strain contribution of the inplane displacements at node i, u_i, and $B_2^i + tB_3^i$ corresponds to the strain contribution of rotations at node i, α_i.

Because of the orthogonality condition, $L_m(t)$ can be easily shown to be

$$L_m(t) = 0$$

which results in

$$B^i_{2_m} = 0$$

Substituting Eq. 2.42 into Eqs. 2.36 and 2.37 yields

$$K_m^{ij} = \int_{-1}^{+1} \int_{-1}^{+1} \int_{-1}^{+1} \begin{bmatrix} (B^i_{1_m})^T D_m B^j_{1_m} & t(B^i_{3_m})^T D_m B^j_{1_m} \\ t(B^i_{1_m}) D_m B^j_{3_m} & t^2 (B^i_{3_m})^T D_m B^j_{3_m} \end{bmatrix} |J(r,s,t)| \, drdsdt \tag{2.56}$$

and

$$K_s^{ij} = \int_{-1}^{+1} \int_{-1}^{+1} \int_{-1}^{+1} \left[\begin{array}{c:c} (B^i_{rs})^T D_s B^j_{is} & (B^i_{1_s})^T D_s B^j_{2_s} + t(B^i_{1_s})^T D_s B^j_{3_s} \\ \hdashline (B^i_{2_s})^T D_s B^j_{1_s}) + t(B^i_{3_s})^T D_s B^j_{1_s} & \begin{array}{c} (B^i_{2_s})^T D_s B^j_{2_s} + t(B^j_{3_s}) D_s B^j_{2_s} \\ + u \\ t(B_{2_s})^T D_s B^j_{3_s} + t^2 (B^j_{3_s})^T D_s B^j_{3_s} \end{array} \end{array} \right] |J(r,s,t)| \, drdsdt \tag{2.57}$$

2.4.2.1 Explicit and reduced integrations

Approximate explicit integration through the thickness can be performed if we neglect the variation of $J(r,s,t)$ with respect to t at each integration point to get $J(r,s,o)$. This is a consequent result from the shell assumption of straight normals[9]. (This can be stated also as neglecting the variation of transformation matrix θ with respect to t). Thus, Eq. 2.56 and 2.57 can be explicitly integrated with respect to get

$$K_m^{ij} = 2 \int_{-1}^{+1} \int_{-1}^{+1} \begin{bmatrix} (B_{1_m}^i)^T D_m B_{1_m}^j & 0 \\ 0 & \frac{1}{3}(B_{3_m}^i)^T D_m B_{3_m}^j \end{bmatrix} |J(r,s,o)|\, drds \quad (2.58)$$

and

$$K_s^{ij} = 2 \int_{-1}^{+1} \int_{-1}^{+1} \left[\begin{array}{c:c} (B_{1_s}^i)^T D_s B_{1_s}^j & (B_{1_s}^i)^T D_s B_{2_s}^j \\ \hdashline (B_{2_s}^i)^T D_s B_{1_s}^j & (B_{2_s}^i)^T D_s B_{2_s}^j + \frac{1}{3}(B_{3_s}^i)^T D_s B_{3_s}^j \end{array}\right] |J(r,s,o)|\, drd \quad (2.59)$$

The numerical integration is then used to evaluate Eq. 2.58 and Eq. 2.59. This integration was carried out originally [8] using two Gaus points in the transverse direction t and three-by-three mesh of Gauss points in direction r, and s. In this original form excellent results are obtained for thick plates and shells but completely unacceptable results obtained for thin plates and shells.

Several investigators have shown the superior results obtained by using reduced integration of two-by-two mesh of Gauss points in direction r and s [9,16,17]

2.4.4 Generalized Loads

The generalized loads corresponding to the body and surface forces are computed from equating the external virtual work of them with the external work of the nodal forces.

The general expression for these forces is [14]

$$P = \int_V N^T \, bdV + \int_S N^T pdS \tag{2.60}$$

where the two terms in the right hand of equation represents body and surface forces respectively.

Employing the change in coordinates, the first term becomes

$$P_b = \int_V N^T \, bdV = \int_V N^T \begin{Bmatrix} b_x \\ b_y \\ b_z \end{Bmatrix} dV$$

$$= \int_{-1}^{+1} \int_{-1}^{+1} \int_{-1}^{+1} N^T \begin{Bmatrix} b_x \\ b_y \\ b_z \end{Bmatrix} |J| \, drdsdt \tag{2.61}$$

Similarly, the second term becomes

$$P_p = \int \int N^T Pds \tag{2.62}$$

and knowing that the elemental surface area can be written as [14]

$$ds = \bar{v}_1 \times \bar{v}_2 = \begin{Bmatrix} \partial_x/\partial_r \\ \partial_y/\partial_r \\ \partial_z/\partial_r \end{Bmatrix} \times \begin{Bmatrix} \partial_x/\partial_s \\ \partial_y/\partial_s \\ \partial_z/\partial_s \end{Bmatrix} drds \qquad (2.63)$$

$$= |J_2| \; drds$$

where

$$|J_2| = \sqrt{\left(\frac{\partial x}{\partial r}\frac{\partial y}{\partial s} - \frac{\partial x}{\partial s}\frac{\partial y}{\partial r}\right)^2 + \left(\frac{\partial x}{\partial r}\frac{\partial z}{\partial s} - \frac{\partial x}{\partial s}\frac{\partial z}{\partial r}\right)^2 + \left(\frac{\partial y}{\partial r}\frac{\partial z}{\partial s} - \frac{\partial y}{\partial s}\frac{\partial z}{\partial r}\right)^2} \qquad (2.64)$$

After performing explicit integration with respect to t, Eq. 2.60 becomes

$$P = 2 \int_{-1}^{+1} \int_{-1}^{+1} N^T b \, |J_2| \, drds + \int_{-1}^{+1} \int_{-1}^{+1} N^T P |J_2| drds \qquad (2.65)$$

2.5 Shear-Locking Problem

As before, the element stiffness matrix can be split into two parts

$$K^{ij} = K^{ij}_m + K^{ij}_s$$

The transverse shear matrix, K^{ij}_s, is of order $(h/L)^2$ higher than the remaining terms. Thus, as thickness/span ratio decreases the computed shear stiffness, K^{ij}_s, completely dominates and no effect of the bending stiffness remains with the finite length of the computer word. This actually causes the element stiffress K^{ij} to over-stiff. This problem was partially solved by using reduced or selective integrations [9,17].

Very recently several investigator studied this problem. Pugh et al [8] employed selective and reduced integrations in computing the stiffness matrix of linear and both parabolic and biparabolic plate elements, respectively. They also showed that the bilinear interpolation and biparabolic interpolation perform better than those using 8-node parabolic elements.

2.6 Kinked Shell and Torsional Effect

The most useful development of the 9L element is the possibility of analysing kinked shells. This is achieved by using six degrees of freedom in the form of adding a fictitious torsional rotation about the normal of the mid-surface at each node.

In any usual shell structure,this fictitious torsional rotation may cause a difficulty if all the element meeting at a node are co-planar. This is due to the assignment of a zero-stiffness to the torsional rotation about the normal,$\alpha_3^{'}$. In non-planer elements meeting at a node, the resistence to this rotation comes directly from the surrounding rotations. As the angles of the kinks between these elements becomes close to 2π, the rotational resistance reduces which spoils the convergence of the solution [10] as a result of the badly-behaved equations. This problem is common to all shell elements which adopt six-global degrees of freedom.

In the past, this problem was remedied by adding a fictitious torsional spring in the normal direction which led to well-behaved equations. In the present, a technique of employing a penalty function is suggested by Kanoknukulchai [10] is used.

This alternative suggests the use of additional constraint between the local torsional rotation of the normal, α_3' , and the rotation of the midsurface, $\frac{1}{2}(\frac{\partial v'}{\partial x'} - \frac{\partial u'}{\partial y'})$, Fig.6. Thus, the suggested penalty function constraint is

$$C \left(\alpha_3' \quad \frac{\partial v'}{\partial x'} , \frac{\partial u'}{\partial y'} \right) = \frac{1}{2} \left(\frac{\partial v'}{\partial x'} - \frac{\partial u'}{\partial y'} \right) - \alpha_3' = 0 \qquad (2.66)$$

Before employing the penalty function procedure, a brief introduction to it is now presented [14]. The idea of penalty function approach is basically an alternative to introduce constrained equations in any system of equations which will, at the same time, introduce any additional equation (as the case in Lagrangian multipliers). Zienkiewicz [14] states this as trying to obtain the stationarity of the potential with a set of equations $C(u) = 0$ in domain Ω (the domain in our case is the volume of the element).

Upon introducing the penalty function, the total potential energy is now

$$\pi^* = \pi + \alpha \int C^T C \, dV \qquad (2.67)$$

in which α is a penalty number. This penalty number α should have a large value enough to satisfy the constraint equations $C(u) = 0$ when the stationarity of the problem is seeked.

This new constraint approach has been first applied in a beam element [10, 14, 31]. Usually, the beam element is formulated based on C_1 continuity from the strain energy

$$U = \int_0^{\ell} EI \left(\frac{d^2w}{dx^2} \right)^2 dx \qquad (2.68)$$

The idea of introducing the penalty function is to formulate the beam element based on C_o continuity where w and θ are assumed to be independant variables. The strain energy expression becomes

$$U = \tfrac{1}{2} \int_0^{\ell} EI \left(\frac{d\theta}{dx} \right)^2 dx + \alpha \int_0^{\ell} C^T C dx \qquad (2.69)$$

or

$$U = \tfrac{1}{2} \int_0^{\ell} EI \left(\frac{d\theta}{dx} \right)^2 dx + \alpha \int_0^{\ell} \left(\frac{dw}{dx} - \theta \right)^2 dx \qquad (2.70)$$

Obviously the physical meaning of α is that of the shear rigidity.

$$\alpha = \tfrac{1}{2} GA \qquad (2.71)$$

For rectangular cross-section beam, Eq. 6.70 becomes

$$U = \tfrac{1}{2} \frac{Et^3}{12} \left[\int_0^{\ell} \left(\frac{d\theta}{dx} \right)^2 dx + \frac{12G}{Et^2} \int_0^{\ell} \left(\frac{dw}{dx} - \theta \right)^2 dx \right] \qquad (2.72)$$

Consequently upon discritization and minimization of Eqs.2.67 or 2.72, a system of equations of the form

$$(K_1 + \alpha K_2)\ \delta + f = 0 \qquad (2.73)$$

is obtained, where K_1 and K_2 represent, respectively, the stiffness contributions of the first and third terms of 2.72, δ represents nodal variables, and f represents the nodal forces.

Eq. 2.73 is a standard form which evolves from using penalty function procedure. This procedure encounters two basic difficulties.

1. As α increases (to infinity) the above equation degenerates to

$$K_2 a = -f/\alpha \qquad (2.74)$$

Unless the matrix K_2 is singular such a solution will be over-constrained and trivial answer.

2. With large but finite value of α numerical difficulties will be arisen as a result of the finite length of the computer word.

The remedy for such over-constraint (as $\alpha - \infty$) is to impose singularity on the matrix K_2. It is found [19] that reduced integration is an effective way of providing such a singularity of matrix K_2 without at the same time introducing any singularity of the total matrix $K_1 + \alpha K_2$.

Now, introducing the constraint $[\ \alpha_3' - \frac{1}{2}\ (\frac{\partial v'}{\partial x'} - \frac{\partial u'}{\partial y'})]$ in the formulation of 9L element is presented.

The governing strain energy results from the above constraint is

$$U_t = \kappa_t uh \int_s [\alpha_3' - \tfrac{1}{2}(\frac{\partial v'}{\partial x'} - \frac{\partial v}{\partial y'})]^2 \, ds \qquad (2.74)$$

where κ_t is a number to be determined. The κ_t number should be chosen such that the desired constrained :

$$a_3' \quad \tfrac{1}{2} [\frac{\partial v'}{\partial x'} - \frac{\partial u'}{\partial y'}]$$

is achieved at the integration points.

To be consistent with previous formulation of Eq. 2.59, the local variables should be expressed in terms of the global variables. This process will give Eq. 2.74 the form [10].

$$U_t = (\delta_i)^T K_t^{ij} \delta_j \qquad (2.75)$$

where the torsional stiffness K_t^{ij} is

$$K_2 = K_t^{ij} = K_t uh \int_{-1}^{+1} \int_{-1}^{+1} \int_{-1}^{+1} \begin{bmatrix} (R_m^i)^T R_m^j & (R_m^i)^T R_n^j \\ (R_n^i)^T R_m^j & (R_n^i)^T R_n^j \end{bmatrix} |J| dr ds \qquad (2.76)$$

and

$$R_m^i = \tfrac{1}{2} [R_2^i \quad -R_1^i \quad 0] \theta^T \qquad (2.77)$$

$$R_m^i = N_i [\theta_{13} \quad \theta_{23} \quad \theta_{33}] \qquad (2.78)$$

where

$$R_1^i = \theta_{11} \frac{\partial N_i}{\partial x} + \theta_{21} \frac{\partial N_i}{\partial y} + \theta_{31} \frac{\partial N_i}{\partial z} \qquad (2.79)$$

and

$$R_2^i = \theta_{12} \frac{\partial N_i}{\partial x} + \theta_{22} \frac{\partial N_i}{\partial y} + \theta_{32} \frac{\partial N_i}{\partial z} \qquad (2.80)$$

A two-by-two integration should be used in evaluating Eq. 2.76 in order to provide the singularity of matrix K_t.

3. Numerical Results

A number of structures of different types have been chosen to demonstrate the 9L element performance, accuracy and reliability. The selected examples are chosen to illustrate the following groups of different structures :

1. thin and thick square plate to demonstrate mesh size convergence and to show the absence or the presence of shear-locking problem,

2. thin cylinderical and hyperbolic paraboloid shells to demonstrate the reliability of 9L element in analyzing thin shell structures,

3. kinked shells (folded plate) to show the success of adopting six-degrees of freedom approach in analyzing such structures where defining the normal at the kink (junction) causes major difficulty,

4. Pinched cylindrical shells to show the effect of central concentrated load in both the torsional penalty number and the reduced integration.

3.1 Thin and Thick Square Plate

A series of square plates with different loadings and boundary conditions are analyzed using three isoparametric elements, namely, the bilinear shell element (4L), Ahmad shell element (8S) and the biparabolic shell element (9L). Reduced integration (2X2X2) is employed in evaluating element properties for elements 8S and 9L, but selective integration(3X3X2) for element 4L. Figures 7-9 show the convergence tests for thin square plate (t/L = 0.01) and thick square plate (t/L = 0.1) for two loadings and two boundary conditions. Because of symmetry only one quarter of the plate is analyzed. As expected, the 9L element performs better than 4L and 8S.

Investigating the 'locking' problem in such isoparametric is followed. For a mixed mesh size (4 X 4, quarter of plate) Figures 9 and 10 show the behavior of uniformly square plate with simply supported and clamped boundary conditions as thickness/span (t/L) ratio decreases. As shown in Figures 9 and 10, Ahmad element locks for both simply supported and clamped boundary conditions, but it is more obvious for simply supported case. Both 4L and 9L elements show no locking as t/L ratio decreases from 0.1 to 0.0001.

3.2 Thin Shells

3.2.1 Thin Cylindrical Shell

This particular shell structure has become a standard shell structure for testing the performance of any shell finite element. The shell is simply supported by rigid diaphragms (u = w = 0) in the transverse edges and free in the longitudinal edges.
The only loading is the gravity load of the shell. Because of

symmetry only one quarter of the shell needs be analyzed. This structure is of interest because both membrane and bending effects are of importance in it.

The results of this example are shown in Figures 12-15. Different mesh-size elements are plotted. The present element results are very accurate with even 2X2 mesh-size giving good agreement with the exact solutions. Convergence study of 9L element along with different finite elements is shown in Figure 15. The stress-resultants of this example have been computed at the nodal points rather than the integration points by smoothing technique [20, 21].

3.2.2 Clamped Hyperbolic Paraboloid Shell

This is another shell structure that is frequently used to test new shell elements. The geometry and material properties are shown in Figure 16. The shell is subjected to a uniform normal pressure. Because of the antisymmetry only one quarter of the shell needs be analyzed. The boundary of antisymmetry axes are also shown in Figure 16. The deflected shape of the center line of the shell is plotted for different mesh sizes and compared to the exact solution of Ref.[23]. The present element results are in good agreement with the exact solution.

It should be noted that the element assembleges of the two previous shell structures represent exactly the real shell geometry, and identical results are obtained with or without the torsional stiffness.

3.3 Simply Supported Folded Plate

This example is chosen to show the performance of 9L element to analyze kinked structures. When using the usual five degrees of freedom (global u, v, w, and local rotation α_1' and α_2'), two major difficulties arise in modelling kinked shells, namely, the difficulty of defining the normal at the kink and the constraining of the rotation about the normal. This example was analyzed by Bakhrebah[24] by using three different elements as shown in Figure 18(b).

The geometry and material properties are shown in Figure 18. The structure is simply supported by rigid diaphragms in the transverse edges and free in the longitudinal edges. Only quarter of the structure is analyzed by 5 elements in transverse direction and 4 in the longitudinal direction.

The 9L element results are in good agreement with Bakhrebah results (which are not shown) and the elasticity and lumped parameter method reported in Reference [25].

3.4 Pinched Cylindrical Shell

The thin and thick forms of the pinched cylindrical shell which demonstrate different load and boundary conditions from those considered previously is chosen to show the efficiency of 9L element in solving such extremes. The pinched cylindrical shell with free edges, Figure 22, has been solved by many investigators [26, 27, 28].

The geometry and material properties of the thin form is shown in Figure 22. The thin form has t = 0.01548 in. with a load P = 0.1 lb, while the thick form has t = 0.094 in. and P = 100 lb.

Initially the example is solved without the addition of a torsional stiffness. The solution is completely travial and unacceptable. This is because of zero-torsional mode. The adding of the torsional stiffness with κ_t large enough to provide singularity of K_t assures the restraining of this mode. Figure 23 shows the study carried out using 4X4 mesh to determine κ_t number.

Although 9L element gives accurate results a small oscillation in the longitudinal displacement occurs, Figure 24, for both thin and thick forms (the thick form is not shown).

Employing a selective integration (3X3X2 mesh) eliminate the oscillation but gives low value for displacements.

The same pinched cylindrical shell problem is then solved with different data and boundary condition. Now, the pinched cylindrical shell is simply supported on diaphragms(u = w = 0) and has the following data : L/R = 2, R/t = 100, ν = 0.3. The exact solution for vertical displacement at C is $W_c = -164.24 \times \frac{Et}{P}$ [30]. The problem is solved using 4X4 mesh and both reduced and selective integrations, and yields

$$(W_c)_{reduced} = 213.69 \frac{Et}{P}$$

and

$$(W_c)_{selective} = 156.90 \frac{Et}{P}$$

and more important, no oscillation occurs.

4. Discussion and Conclusions

The thin and thick square plate tests show the good performance of the biparabolic shell element. The biparabolic shell element has no locking problem which exists in the parabolic shell element of Ahmad. Its performance in analyzing shell probelms is also shown. Reduced integration is an essential requirement for the success of 9L element.

Although the 9L element seems to give accurate results when applied to the pinched cylindrical shell with free edges, a small oscillation in the longitudinal displacement occurs. The use of selective integration in both pinched cylindrical shell (free edges and simply supported edges) proves the possibility of using such scheme when reduced integrated stiffness matrix permits the assembled structure to have zero-energy modes. This is happened when the boundary condition is not enough constrained and thus will not suppress such modes [15].

Even though the oscillation problem is remedied by using selective integration, an alternative element combined the simplicity of the bilinear shell element and the accuracy of biparabolic element (or parabolic element) is suggested. The suggested element uses the bilinear interpolation (4 nodes) for the definition of displacement while retaining the biparabolic interpolation (9 nodes) for the definition of geometry.

The suggested element will be developed and tested by the author in the near future.

REFERENCES

1. A. Hrennikoff, 'The Finite Element Method in Application to Plane Stress, Publication Int. Ass. Bridge Struct. Engng. 28, Part II, 49-68, 1968.

2. O.C.Zienkiewicz and Y. K. Cheung, 'Finite Element Procedures in the Solution of Plate and Shell Problems', Stress Analysis, Chapter 8, eds. O.C.Zienkiewicz and G. S. Holister, Wiley,1965.

3. R.W. Clough and C. P. Johnson, 'A Finite Element Approximation for the Analysis of Thin Shells', Int. J. Solids and Structures, 4, 43-60, 1968.

4. A. J. Carr, A Refined Element Analysis of Thin Shell Structures Including Dynamic Loading, SEL Report No.67-9, Univ. of California, Berkeley, 1967.

5. R. H. Gallagher, 'Analysis of Plate and Shell Structures', Proc. Symp. Application of Finite Element Method in Civil Engineering, Vanderbilt Univ., Nashville, Tenn., 1969.

6. E.L.Wilson R. L. Taylor, W.P. Doherty and J. Gahaboussi, 'Incompatible Displacement Models', Numerical and Computer Methods in Structural Mechanics, Ed. S. J. Fenues et. al., Academic Press, New York, 1973.

7. S. W. Key and Z. E. Beisinger, 'The Analysis of Thin Shells with Transverse Shear Strains by the Finite Element Method', Proc. Conf. Matrix Meth. Struct. Mech. Wright-Patterson Air Force Base, Ohio, AFFDL - TR - 68 - 150, 667-710, 1968.

8. S. Ahmad, B. M. Irons and O.C.Zienkiewicz, 'Analysis of Thick and Thin Shell Structures by Curved Finite Elements', Int. J. Num. Meth. Engng. 2, 419-451, 1970.

9. O. C. Zienkiewicz, R. L. Taylor and J. M. Too, 'Reduced Integration Technique in General Analysis of Plates and Shells', Int. J. Num. Meth. Engng., 3, 275-290, 1971.

10. W. Kanoknukulchai, A Simple and Efficient Finite Element for General Shell Analysis, SESM 78-1, Univ. of California, Berkeley, January, 1978.

11. B. M. Irons, ' Numerical Integration Applied to Finite Element Methods', Conf. Use of Digital Computers in Struct. Engng., Univ. of Newcastle, 1966.

12. O. C. Zienkiewicz, B. M. Irons, J. Ergatoudis, S. Ahmad and F. C. Scott, 'Isoparametric and Associated Element Families for Two and Three Dimensional Analysis', Proc. Cource on Finite Element Methods in Stress Analysis, ed. I. Holland and K. Bell, Trondheim Tech. Univ., Norway, 1969.

13. O. C. Zienkiewicz, 'Isoparametric and Allied Numerically Integrated Elements - A Review', ONR Symp., Univ. of Illinois at Urbana - Champaign, 8-10 Sept., 1971.

14. O. C. Zienkiewicz, The Finite Element Method, McGraw-Hill,London,1977.

15. R. D. Cook, Concepts and Applications of Finite Element Analysis, Wiley, 1974.

16. S. F. Pawsey, 'The Analysis of Moderately Thick and Thin Shells by the Finite Element Method', Ph.D. Thesis, Univ. of California, Berkeley, 1969.

17. S. F. Pawsey and R. W. Clough, 'Improved Numerical Integration of Thick Shell Finite Element', Int. J. Num. Meth. Engng, 3, 575-586,1971.

18. E. D. Pugh, E. Hinton and O. C. Zienkiewicz, 'A Study of Quadrilateral Plate Bending Elements with Reduced Integration', Int. J. Num. Meth. Engng., 12, 1059-1079, 1978.

19. O. C. Zienkiewicz and E. Hinton, 'Reduced Integration, Fuction Smoothing and Non-Conformity in Finite Element Analysis', J. Franklin Inst., 302, 443-61, 1974.

20. E. Hinton and J. S. Campbell, 'Local and Global Smoothing of Discontinuous Finite Element, Function Using a Least Square Method', Int. J. Num. Meth. Engng, 8, 411-480, 1974.

21. E. Hinton, and D.R.J.Owen, Finite Element Programming, Academic Press, 1977.

22. A. C. Scordelis and K.S.LO, 'Computer Analysis of Cylindrical Shells', ACI. J., 61, 539-561, 1964.

23. S. Chetty and H. Tohenham, 'Investigation into the Bending Analysis of Hyperbolic Paraboloid Shells', Indian Concrete Journal, July,1964.

24. S. A. Bakhrebah, Finite Element Analysis of Intersecting Cylinders, Ph.D.Thesis, University of Illinois at Urbana-Champaign, 1973.

25. A Conrado and W.C.Schnobrich, Discrete Analysis of Continuous Folded Plate, SES No.321, Univ. of Illinois at Urbana - Champaign, 1967.

26. D. G. Ashwell and A. B. Sabir, 'A New Cylindrical Shell Finite Element Based on Simple Independent Strain Functions', Int. J. Mech. Sci., 18, 555, 1971.

27. G. Cantin, 'Rigid Body Motions of Curved Finite Element', AIAA,8, 1252, 1970.

28. D. G. Ashwell, 'Strain Element, With Application to Arches, Rings and Cylindrical Shells', Finite Element for Thin Shells and Curved Members,ed. D.G.Ashwell and R.H.Gallagher,Wiley, 1976.

29. G. R. Cowper, G. M. Lindberg and M.D.Olson, 'A Shallow Shell Finite Element of Triangular Shape', Int. J. Solids Struct., 6, 1133-1156 (1970).

30. G. M. Lindberg, M. D. Olson and G.R.Cowper, 'New Development in the Finite Analysis of Shells', DME/NAE Quarterly Bulletin No. 1969, (4).

31. R. H. Gallagher, Finite Element Analysis Fundamentals, Prentice-Hall, 1975.

FIGURES

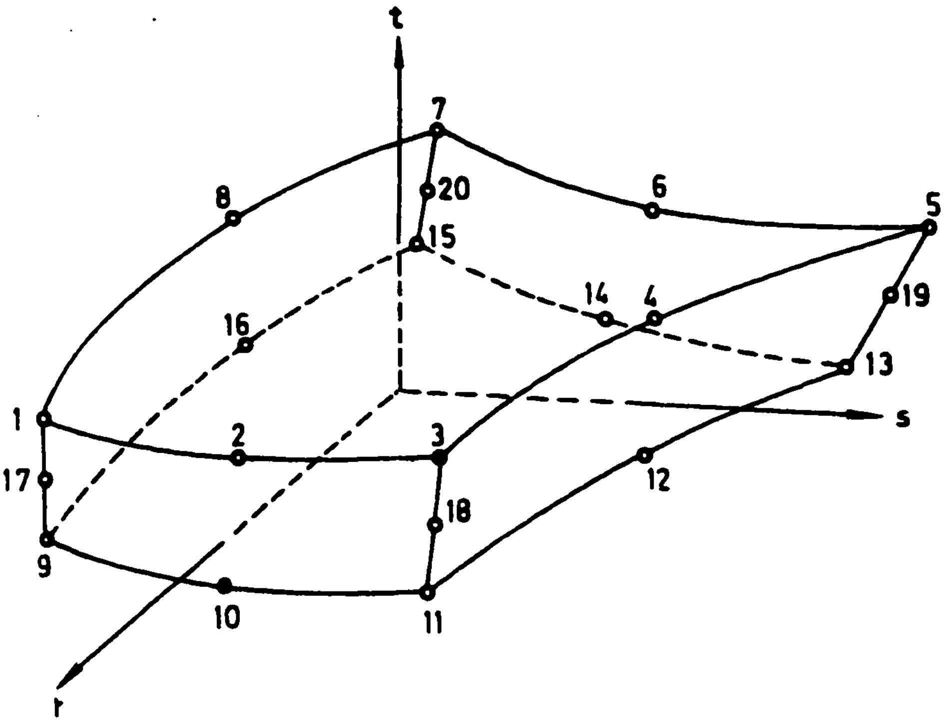


Fig. 1. Three Dimensional Thick-Shell Elements (20 Nodes)

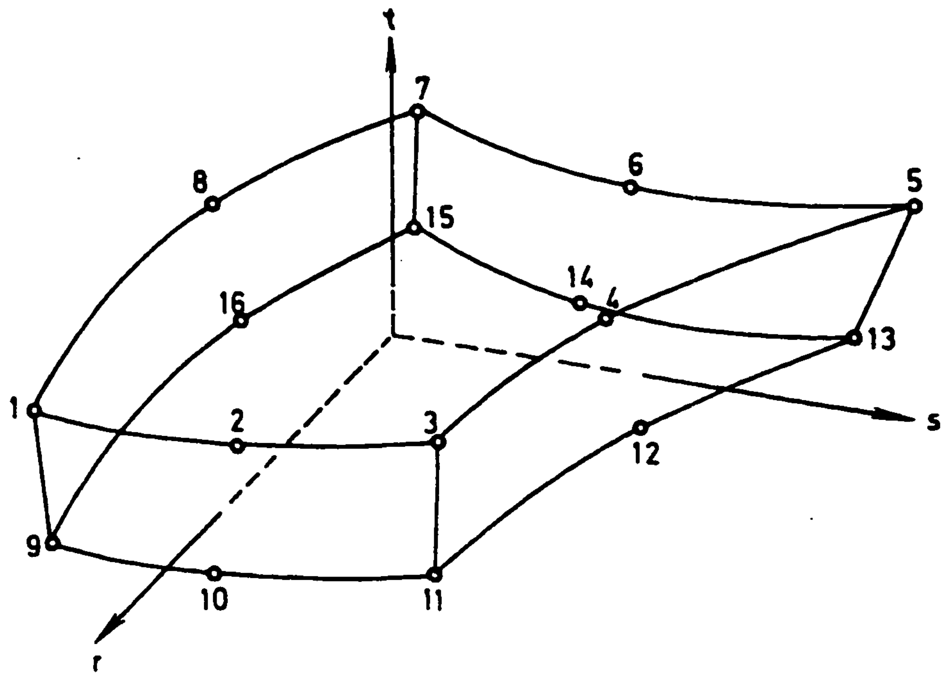


Fig. 2. Three Dimensional Shell Element (16 Nodes)

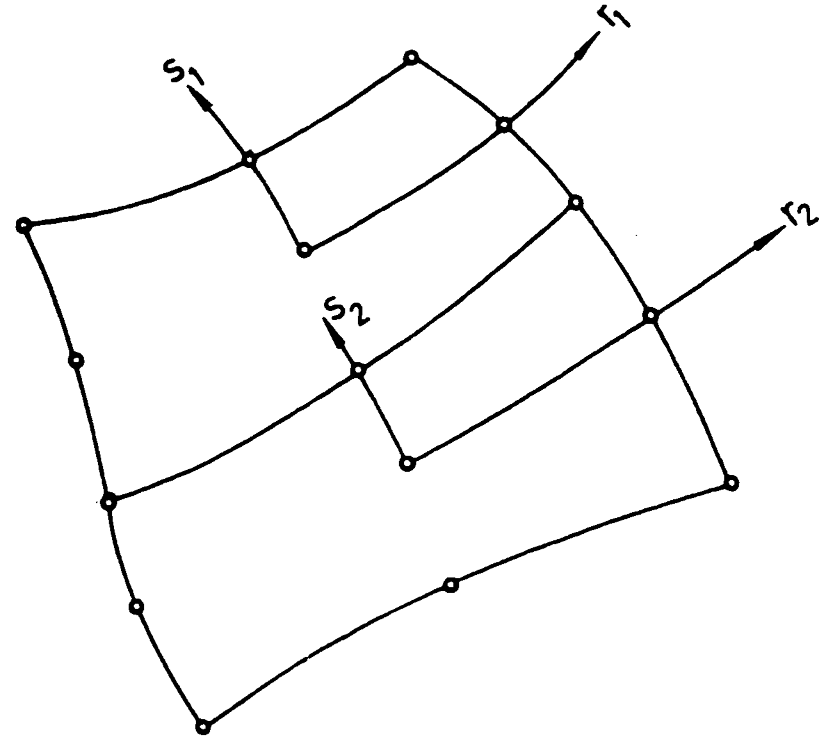


Fig. 3 . Adjacent Elements .

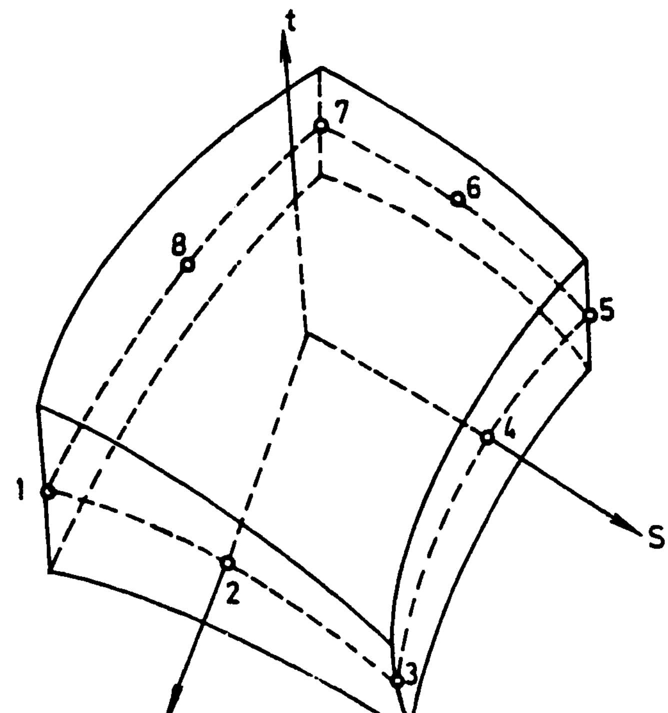


Fig. 4. Ahmad Shell Element (85)

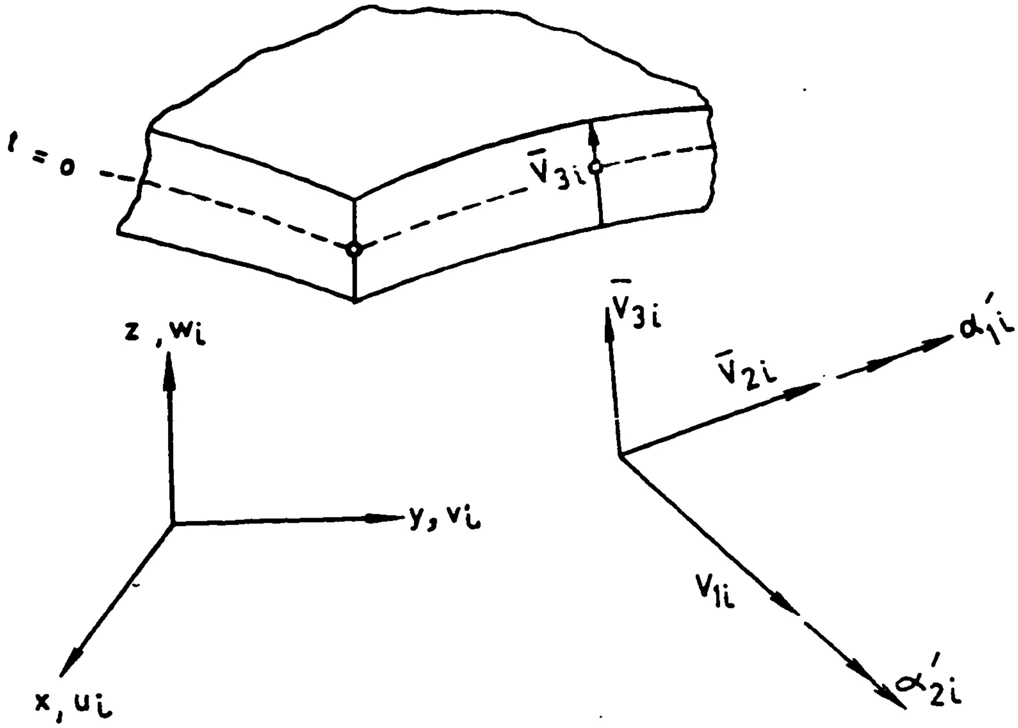


Fig. 5. Global and Local Coordinates and Displacements.

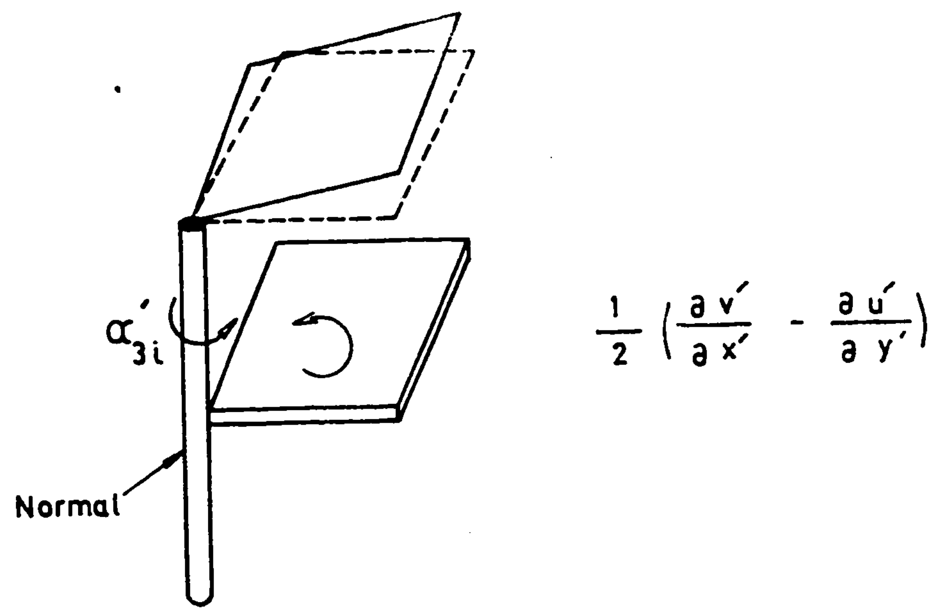


Fig. 6. Torstional Penalty Function

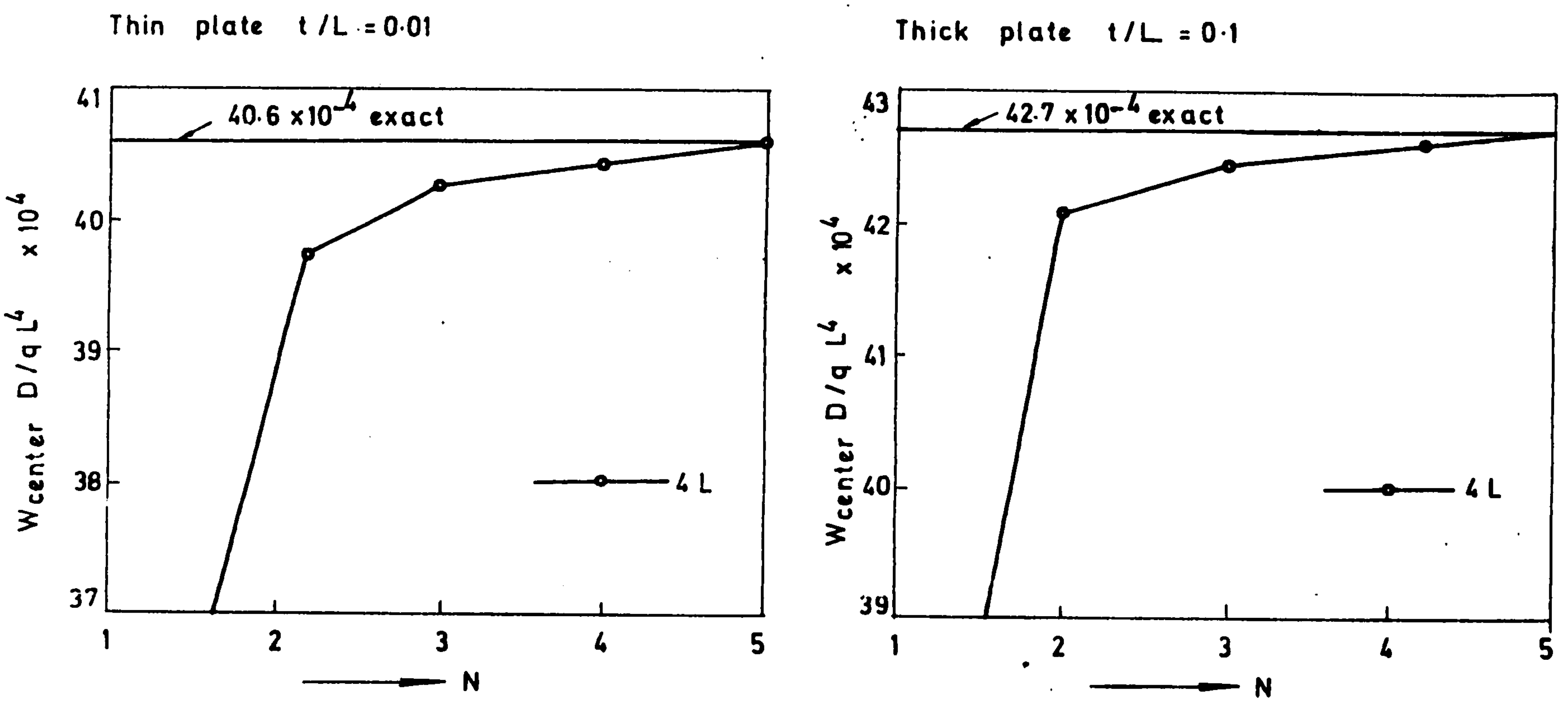


Fig7A Simply Supported square plate under uniform load q (N is the number of elements / side quarter of plate analyzed)

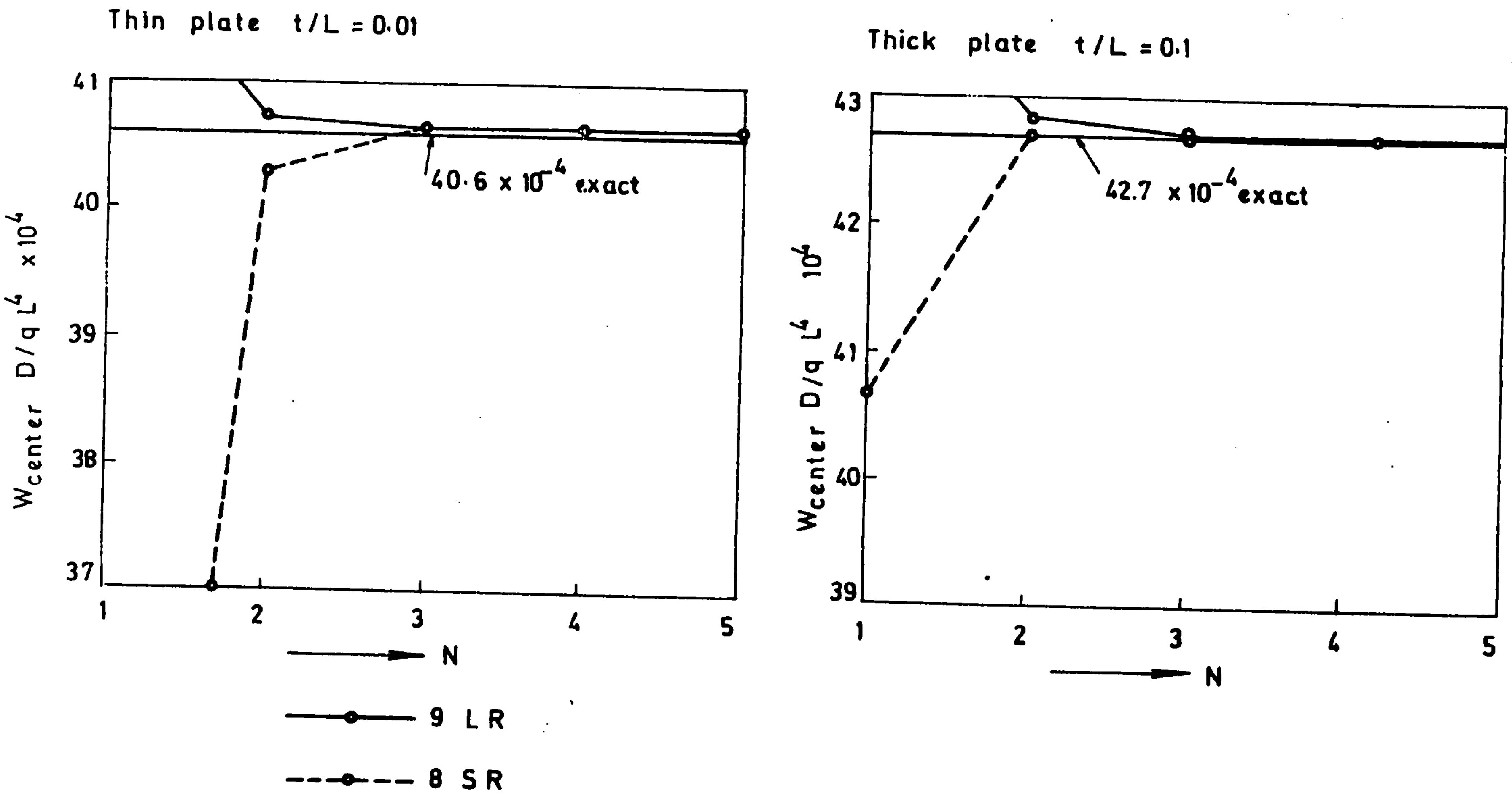


Fig 7B Simply Supported Square Plate Under Uniform Load q (N is the Number of Elements / Side Quarter of Plate Analyzed.

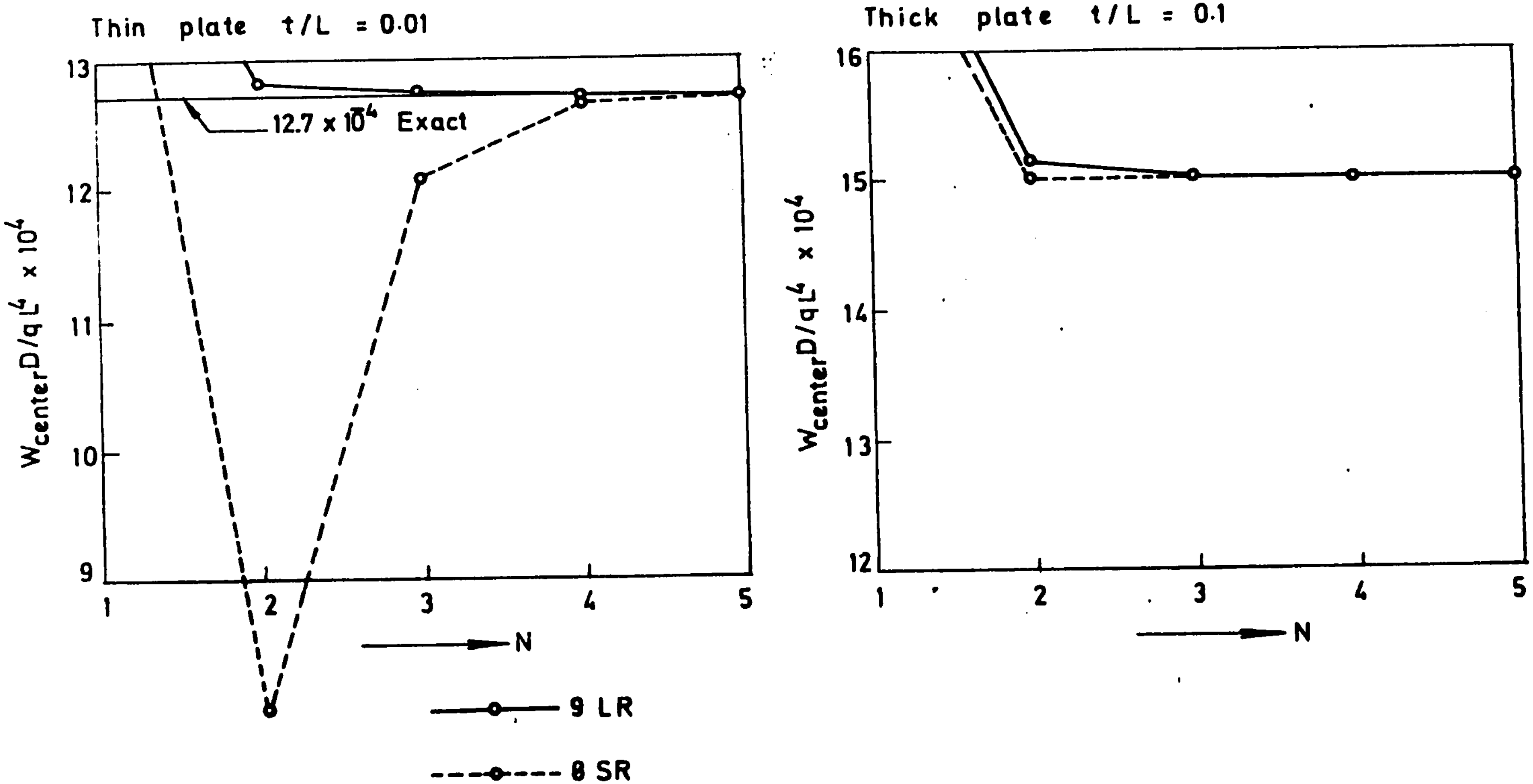


Fig 8 B Clamped Square Plate Under Uniform Load q. (N is The Number of Element / Side of Quarter of Plate analyzed.)

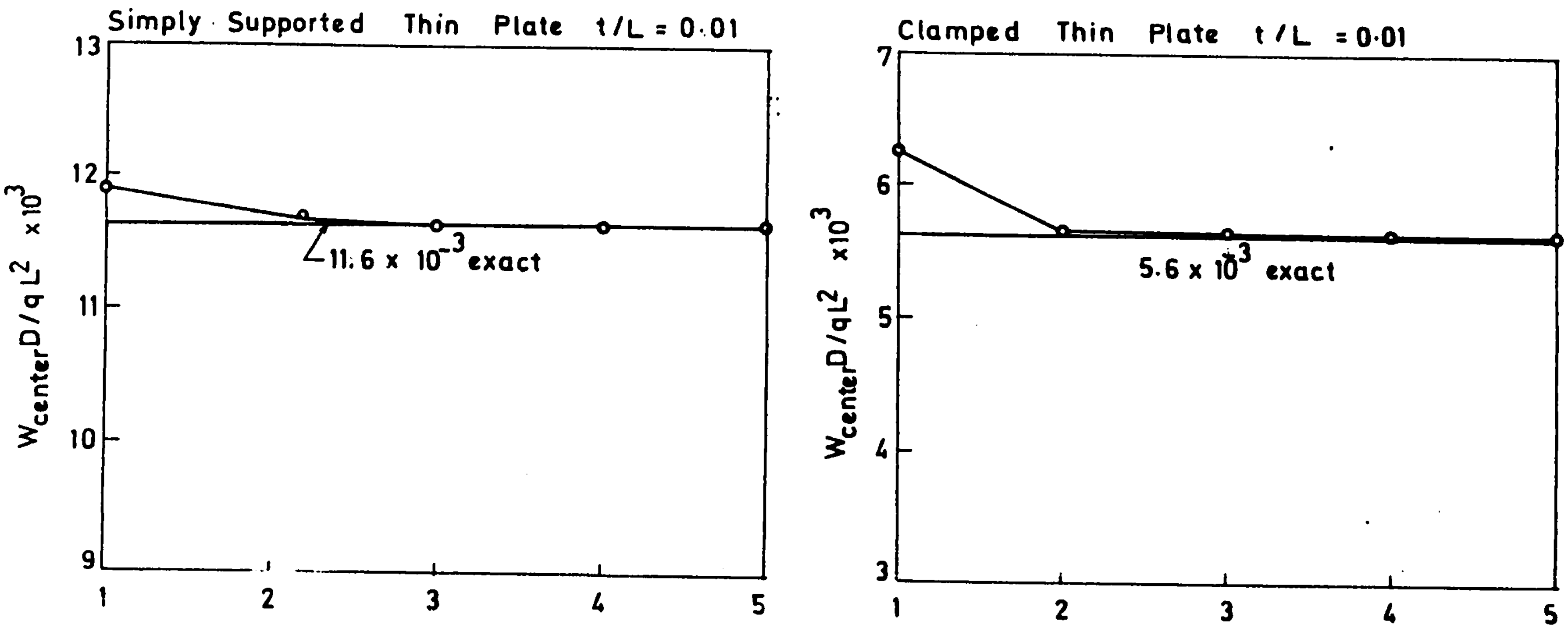


Fig. 9 Thin Square Plate Under Central Load
9 LR Only.

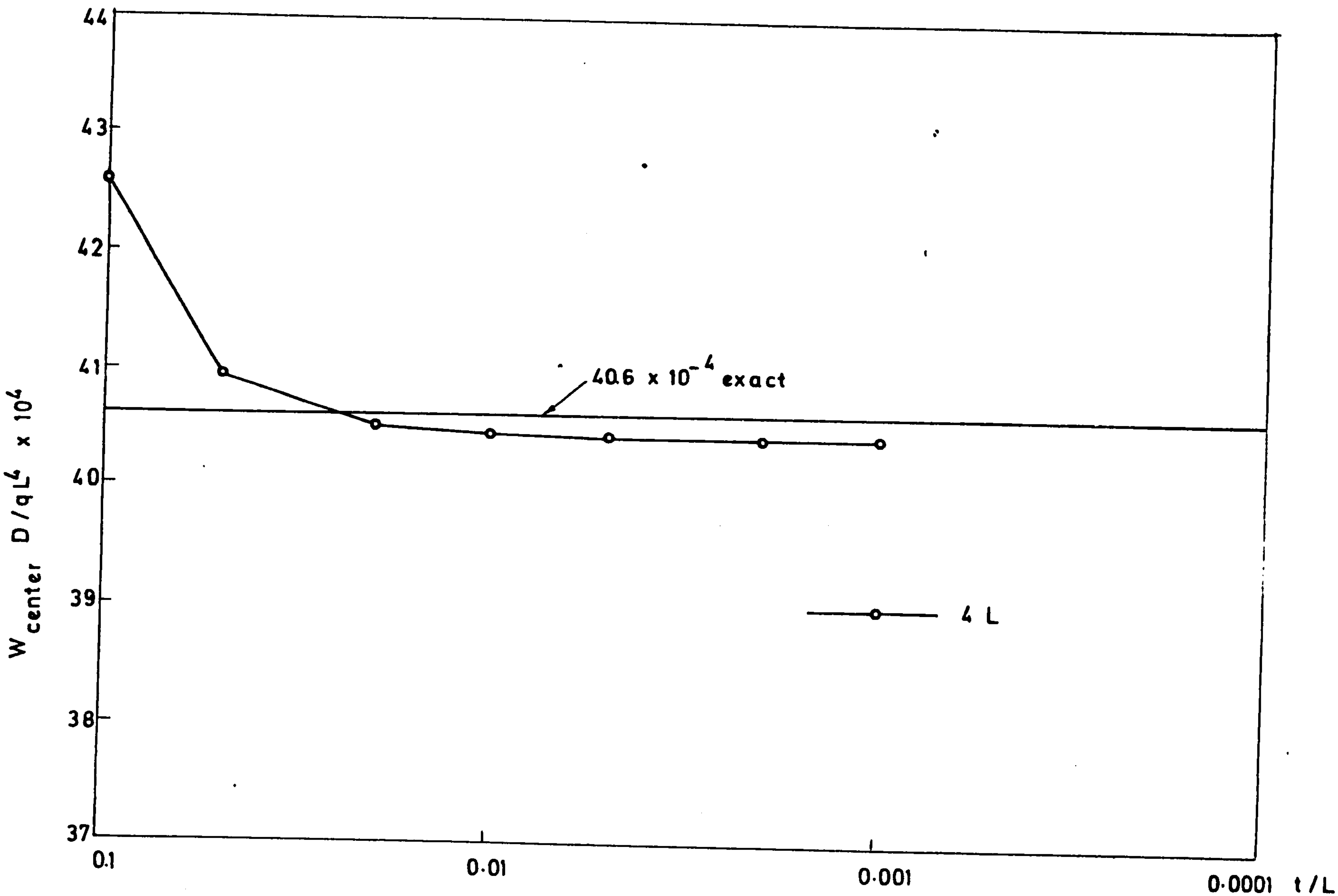


Fig.10A1 Simply Supported Square Plate Behavior of Solution as t / L decreases, N = 4

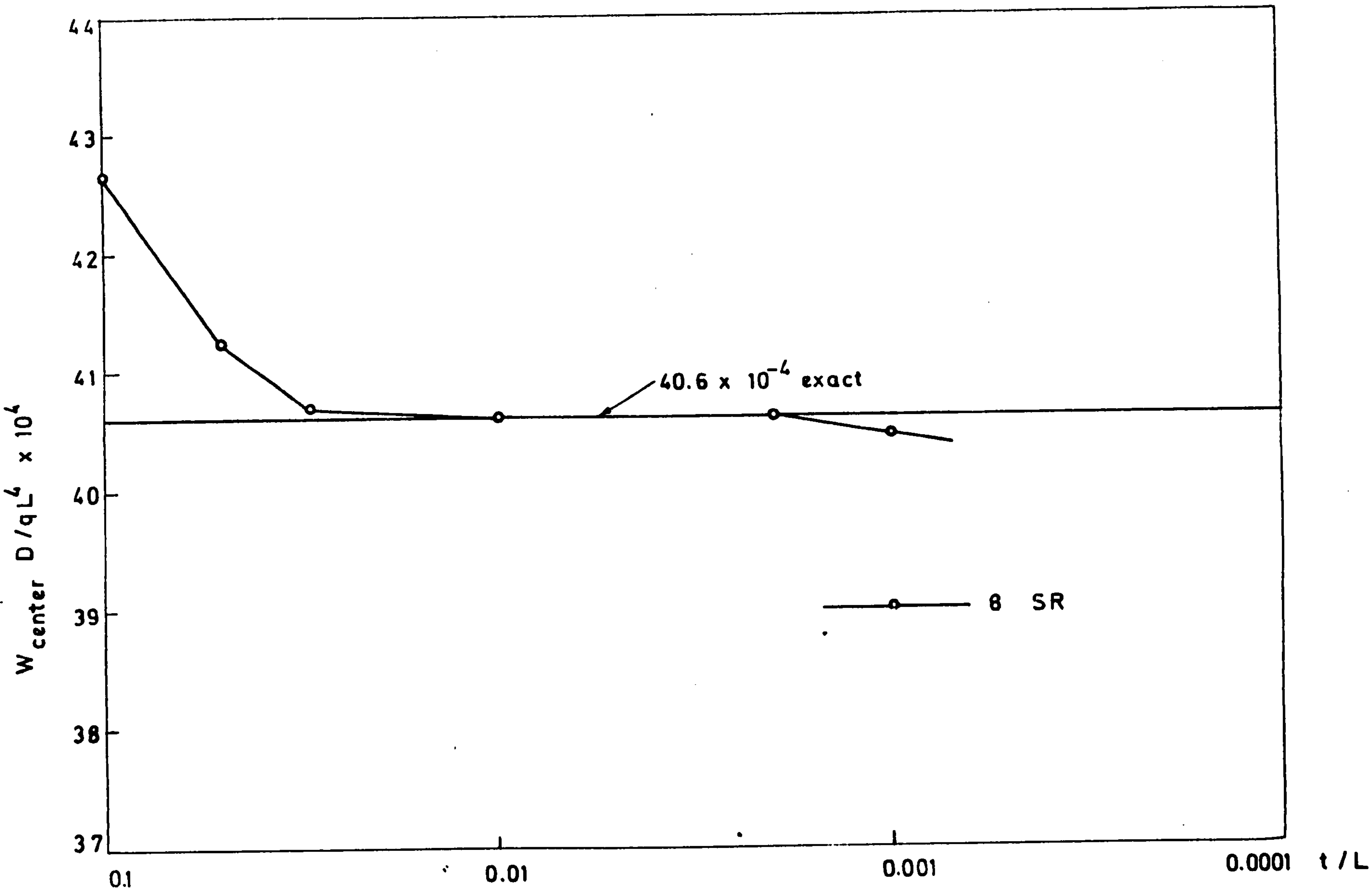


Fig.10 A2 Simply Supported Square Plate Behavior of Solution as t / L decreases , N = 4

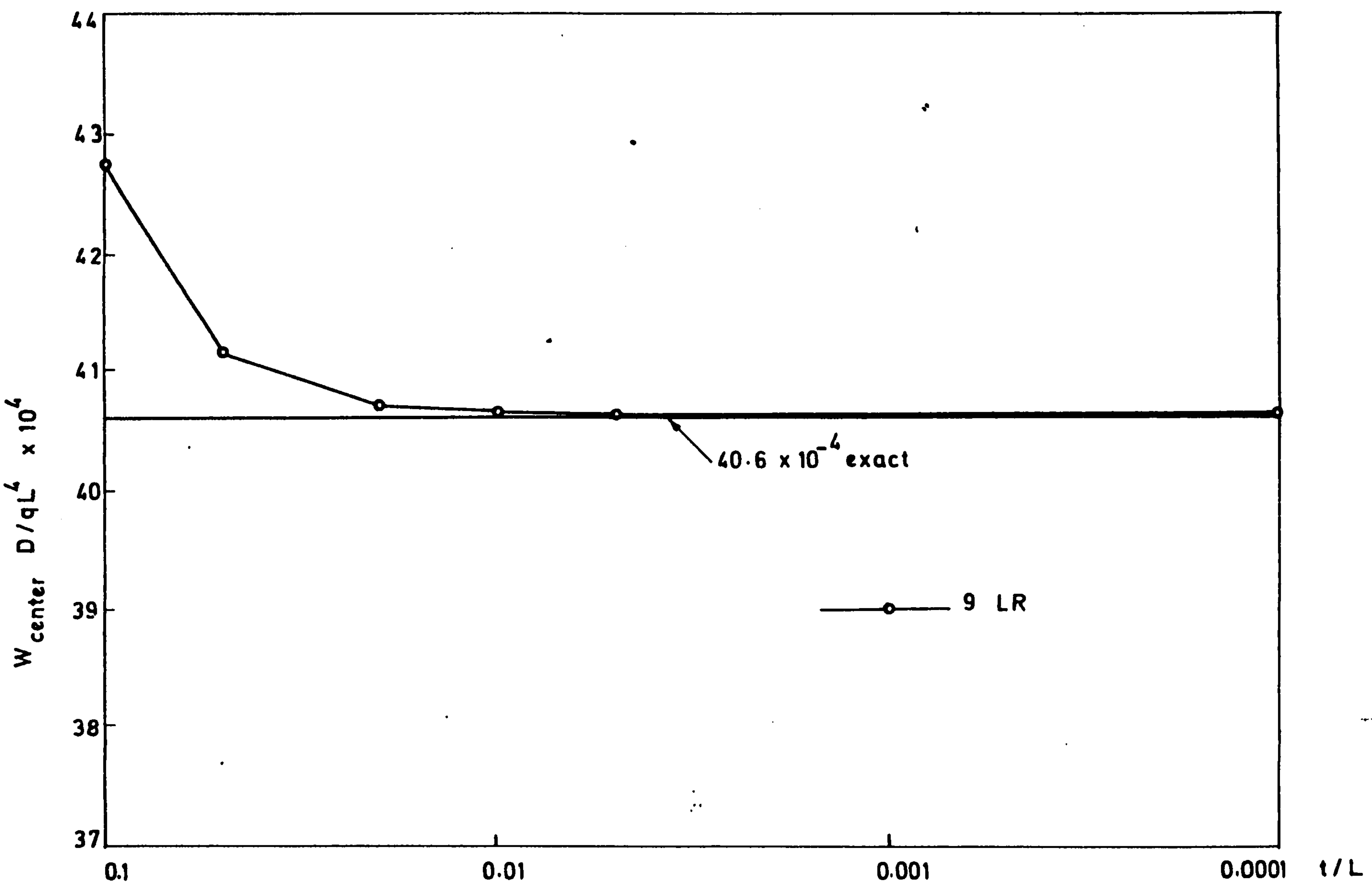


Fig.10A3 Simply Supported Square Plate Behavior of Solution as t / L decreases, N = 4

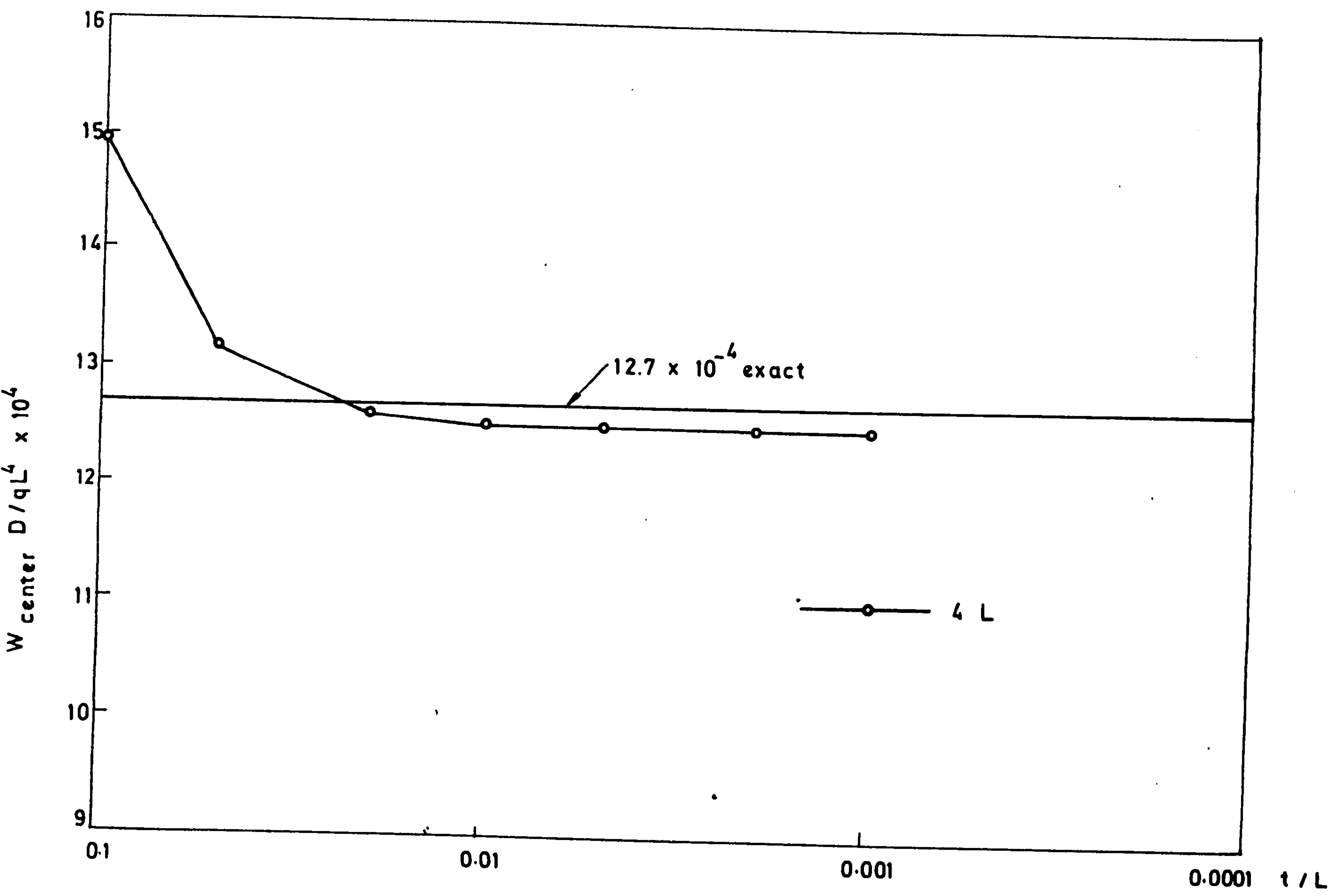


Fig.10B1 Clamped Square Plate Behavior of Solution as t / L decreases, N = 4

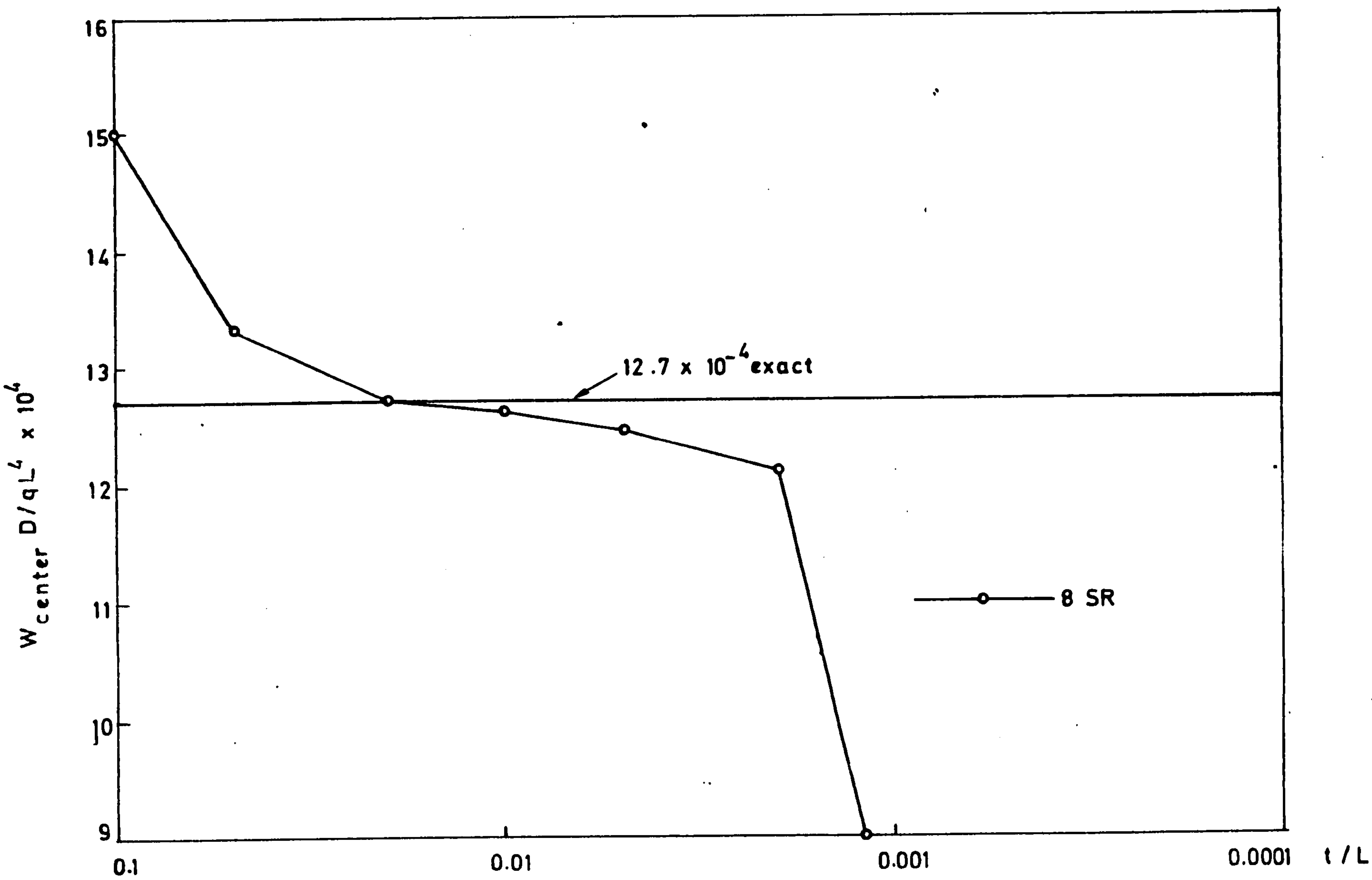


Fig.10B2 Clamped Square Plate Behavior of Solution as t / L decreases, N = 4

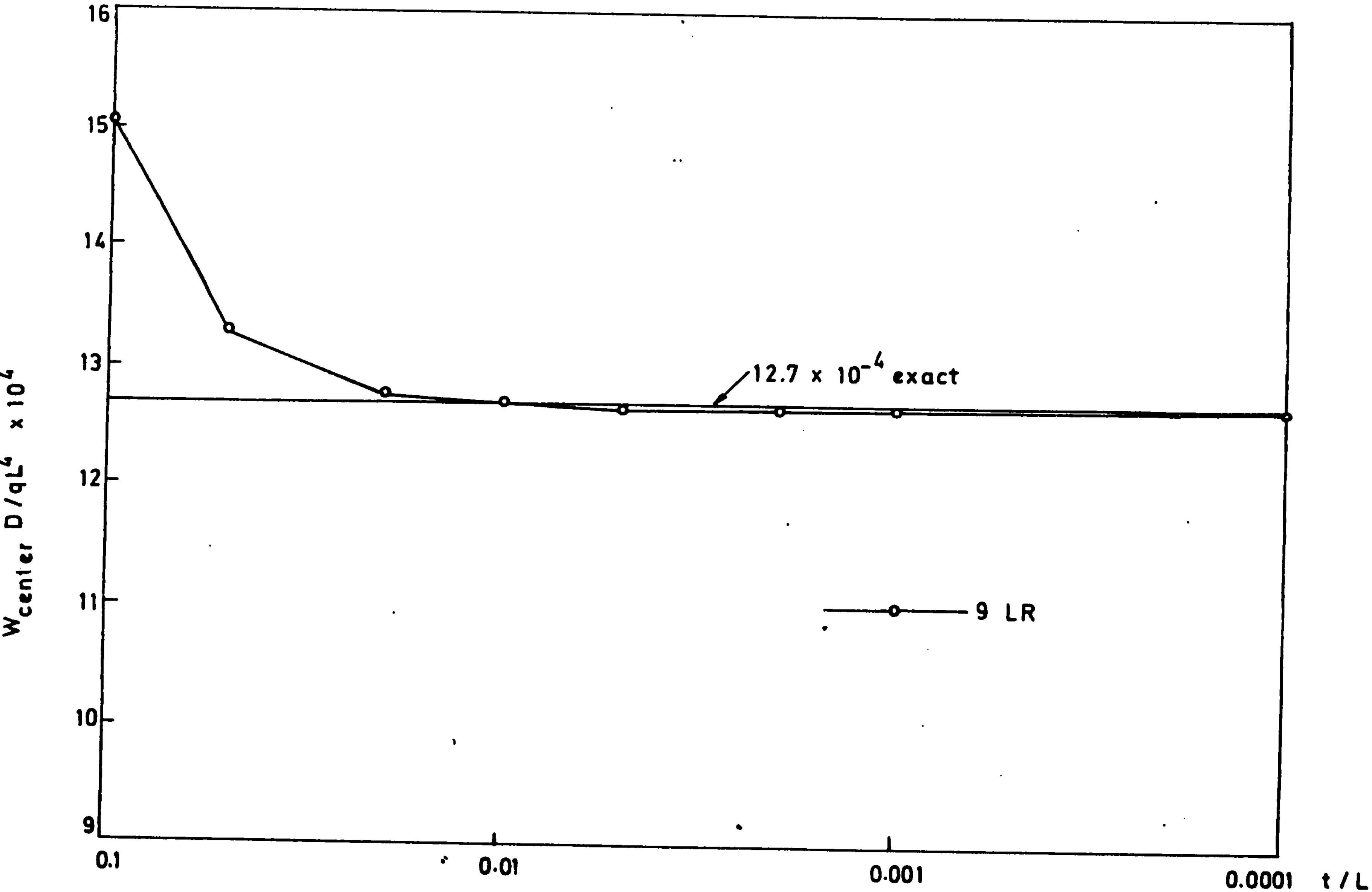


Fig.10B3 Clamped Square Plate Behavior of Solution as t / L decreases, N = 4

$E = 3{\cdot}0 \times 10^6$ Ksi

$= 0{\cdot}0$

$q = 0{\cdot}09$ K/ft^2

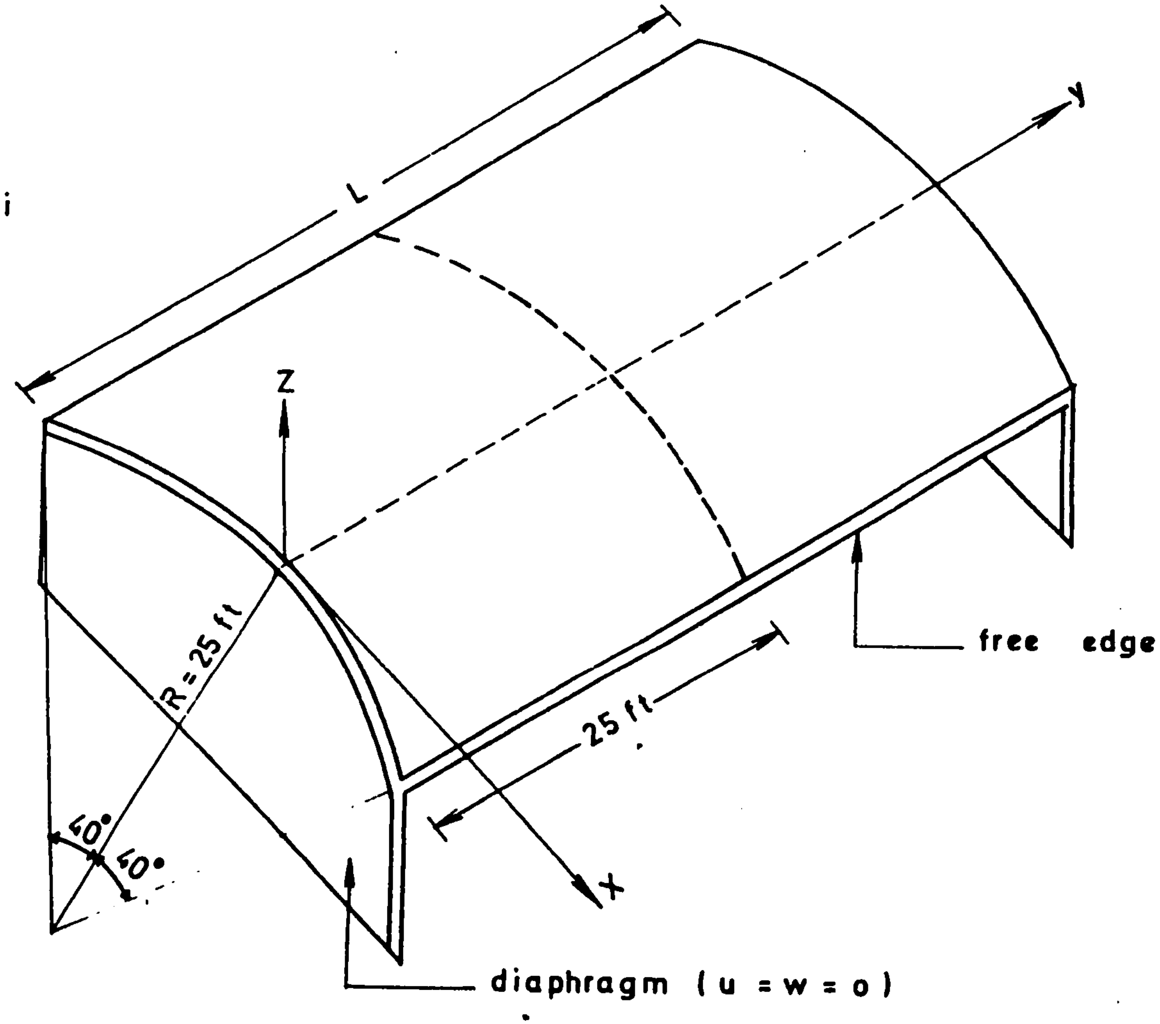


Fig 11 Geometry and loading of Cylindrical Thin Shell.

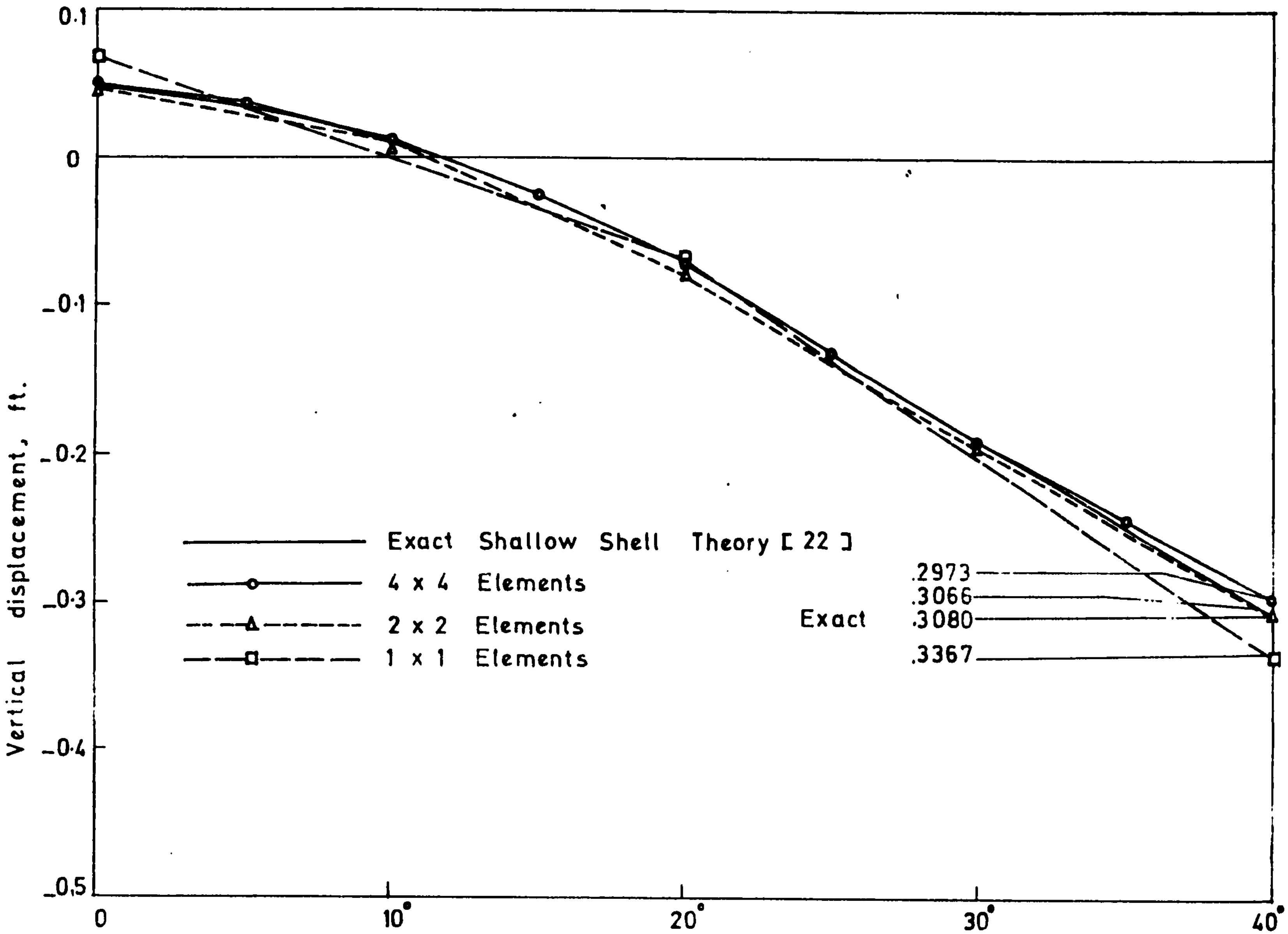


Fig. 12 Cylindrical shell root, vertical displacement at y = L/2

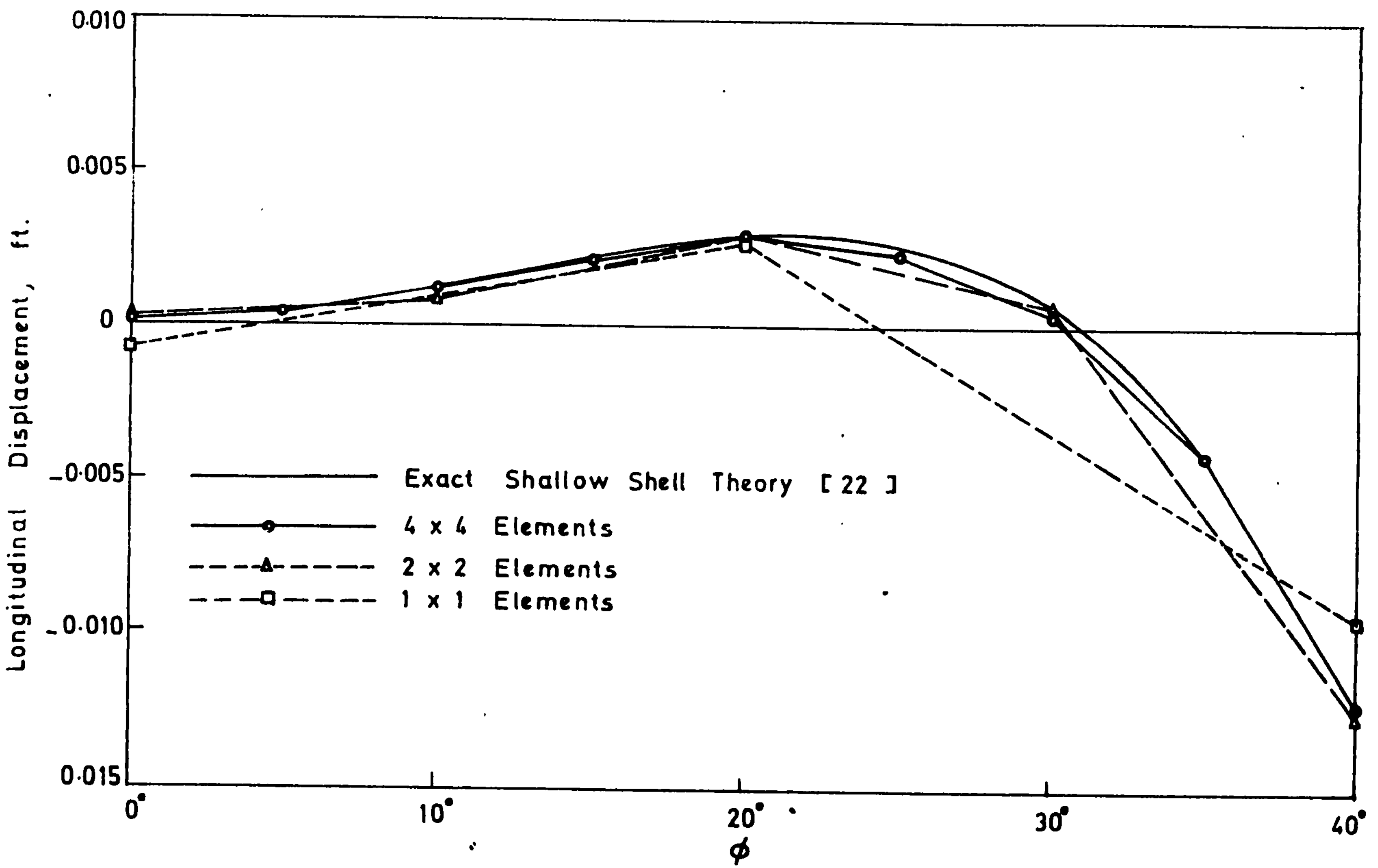
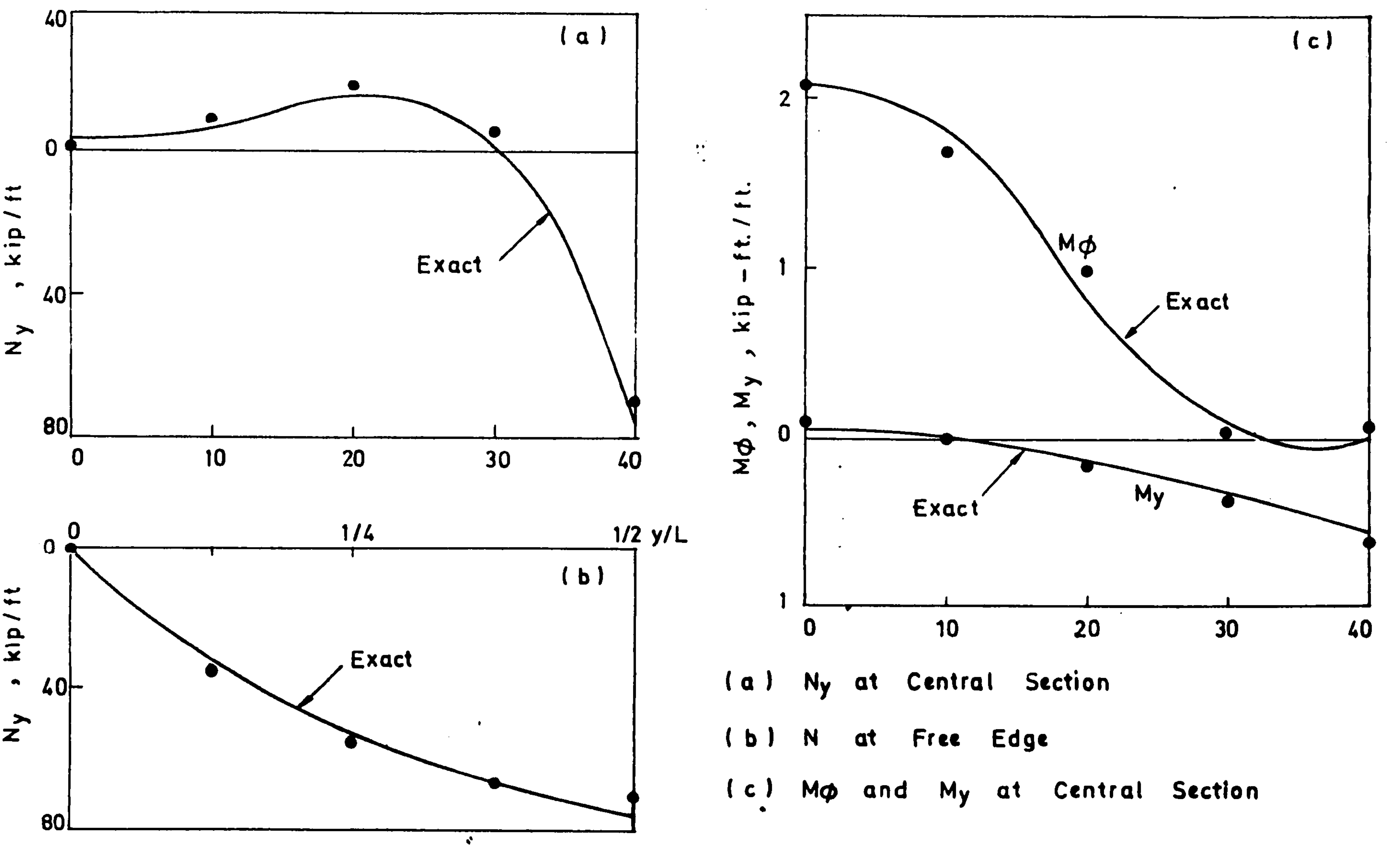


Fig. 13 Cylindrical shell roof, longitudinal displacement at diaphragm (y = o).



(a) N_y at Central Section

(b) N at Free Edge

(c) Mϕ and M_y at Central Section

Fig. 14 Cylindrical Shell Roof, Stress Resultants for 4 x 4 Elements, Obtained at Nodal Points by Extrpolation Of Guass Point Stresses.

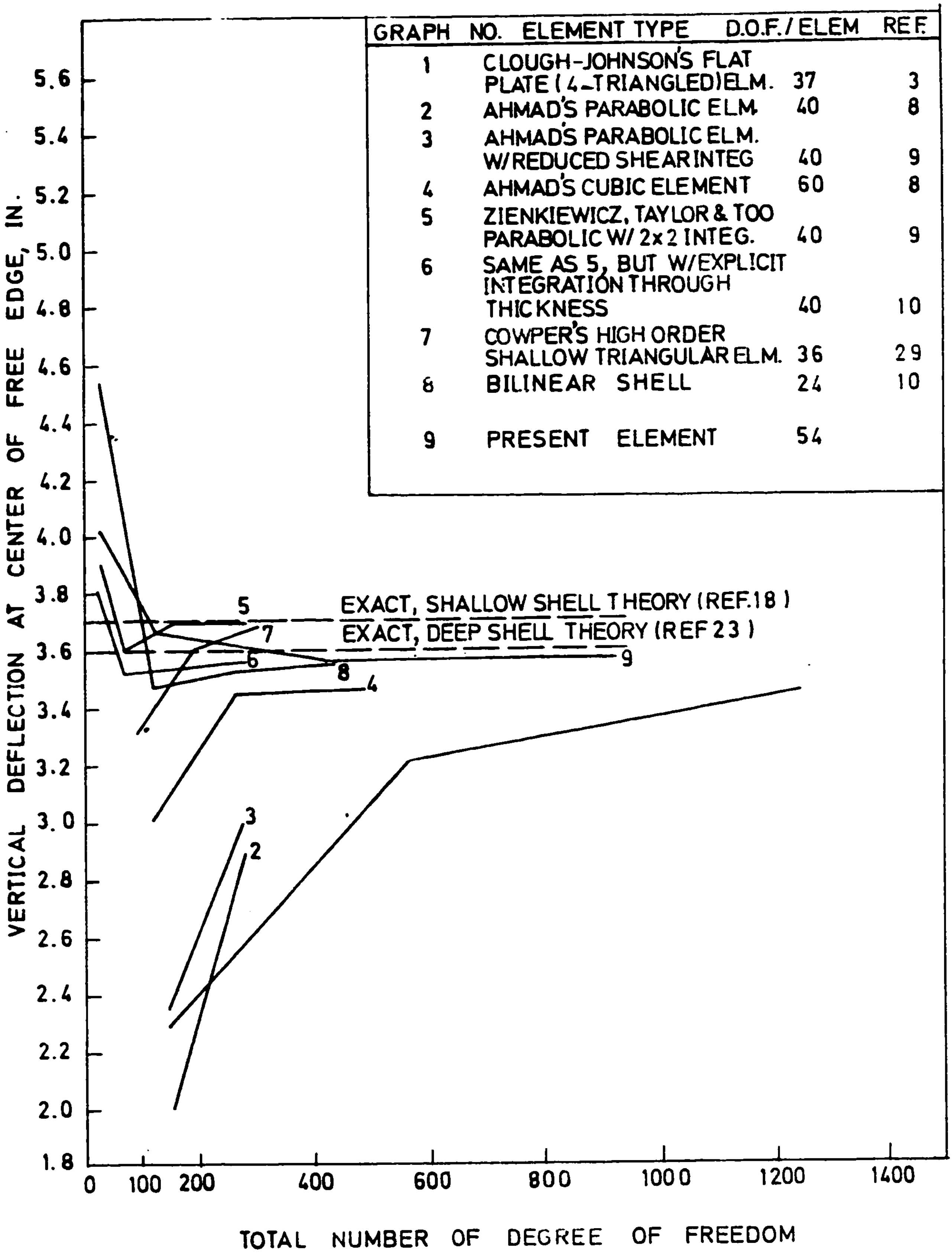


Fig. 15 Gylindrical Shell Roof, Converqence Test.

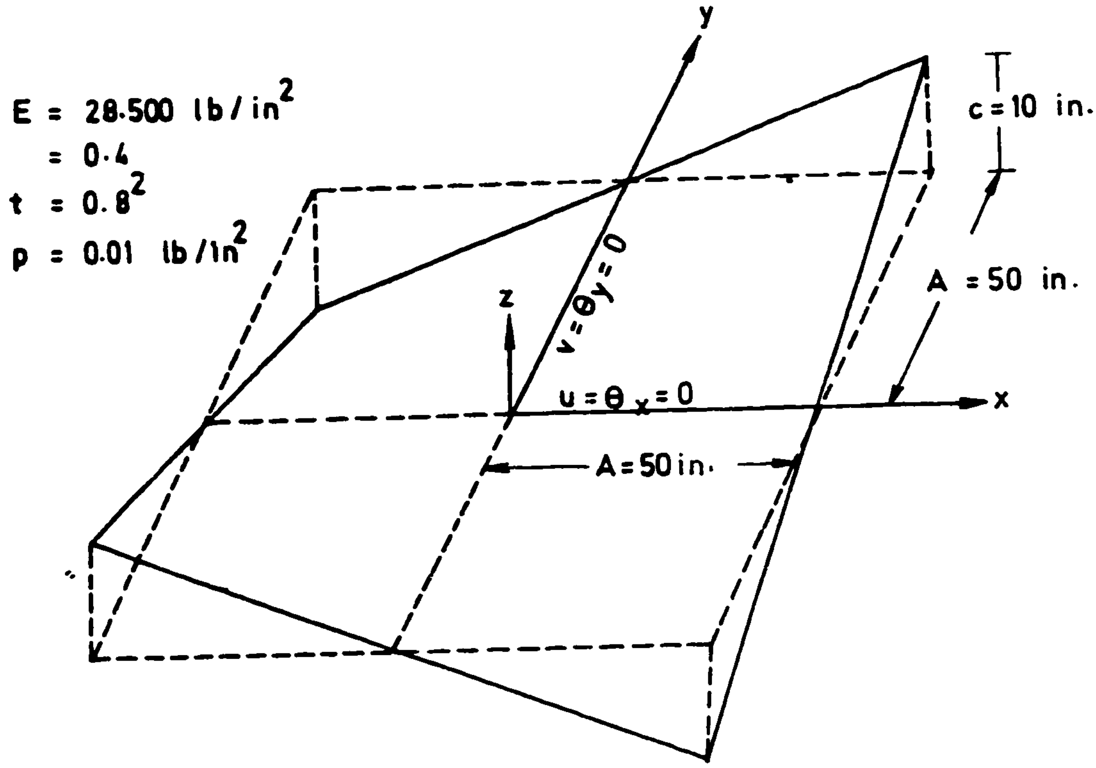


Fig. 16 Clamped hyperbolic shell

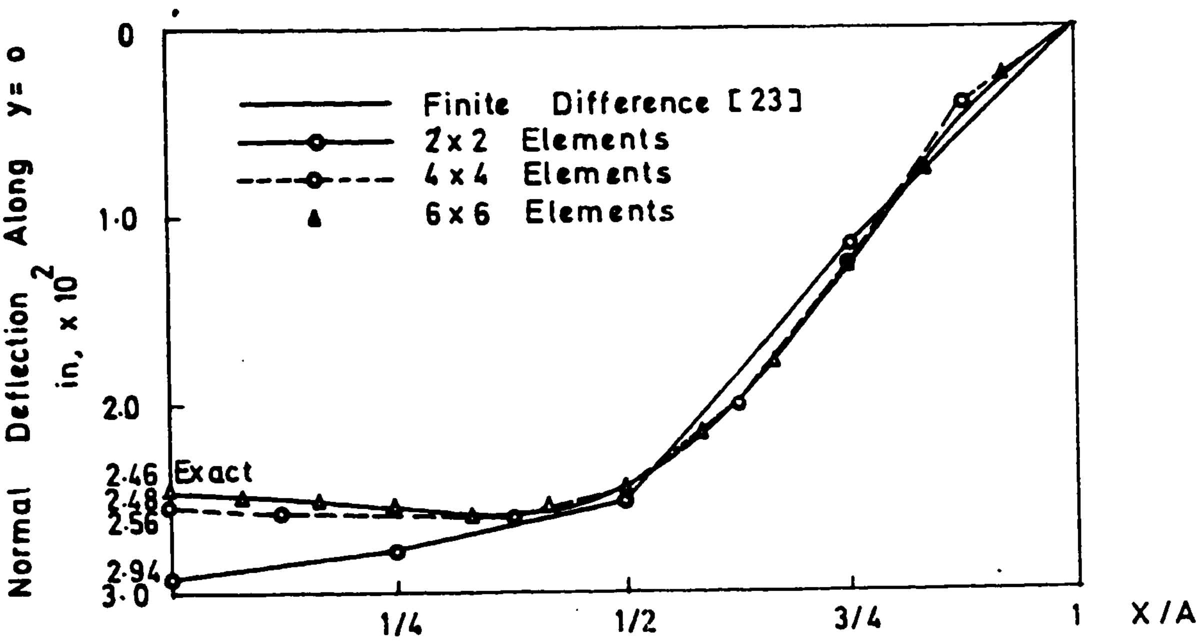
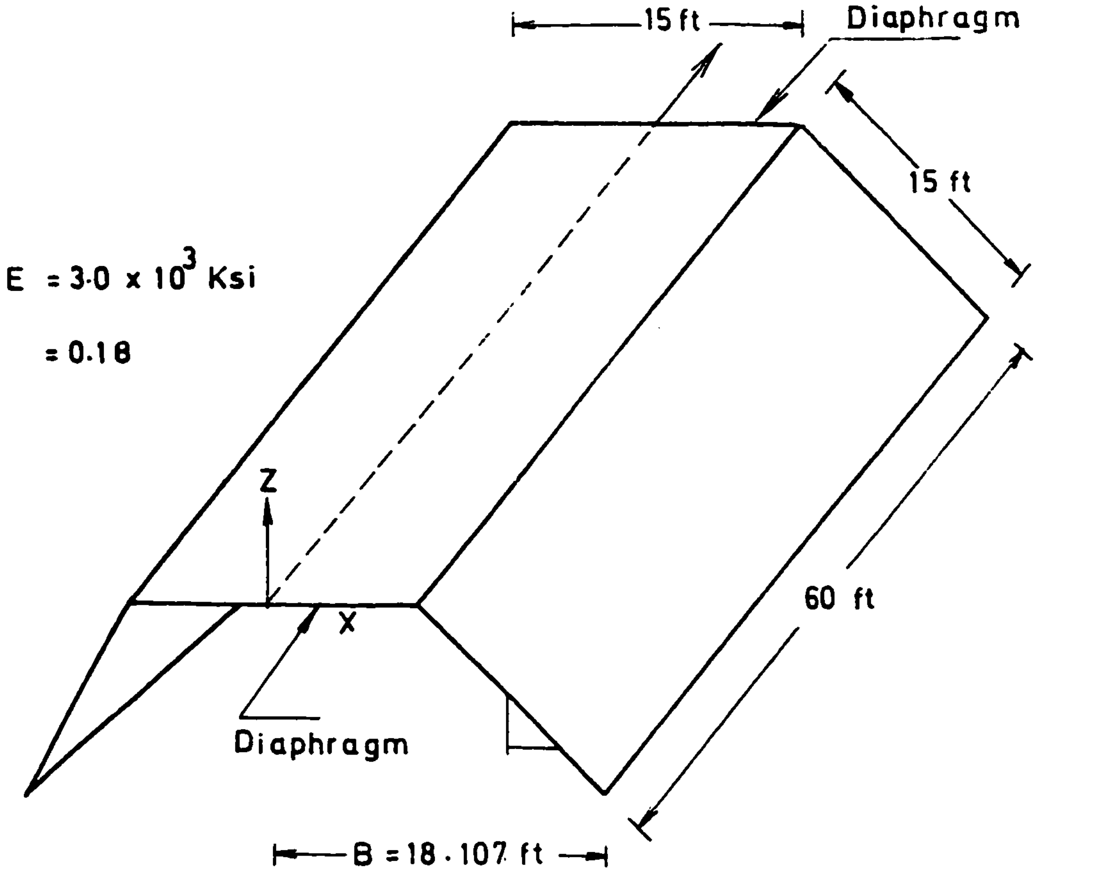
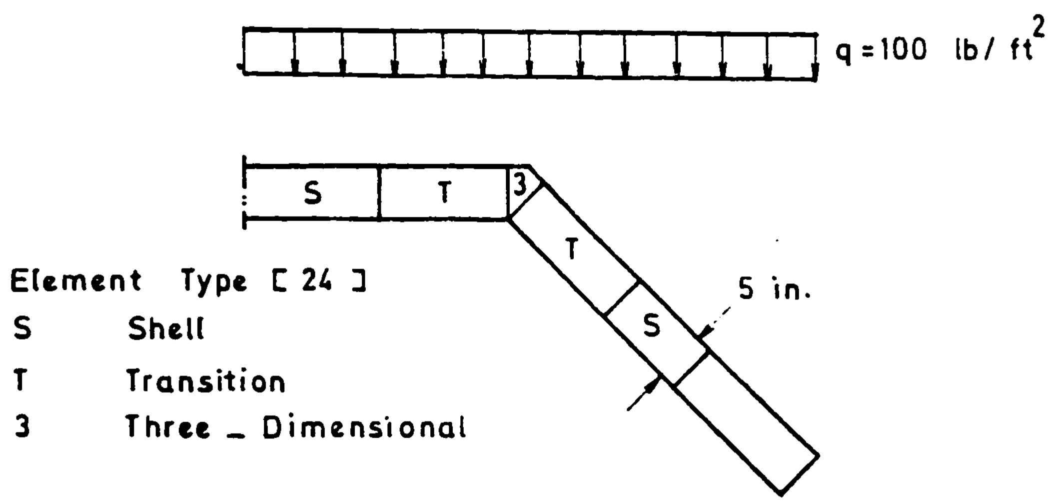


Fig. 17 Clamped Hyper Shell, Deflection at Center Line



(a) Geometry



(b) Finite Element Idealization as uud in Ref [24]

Fig. 18 Geometry and Loading of Folded Plate.

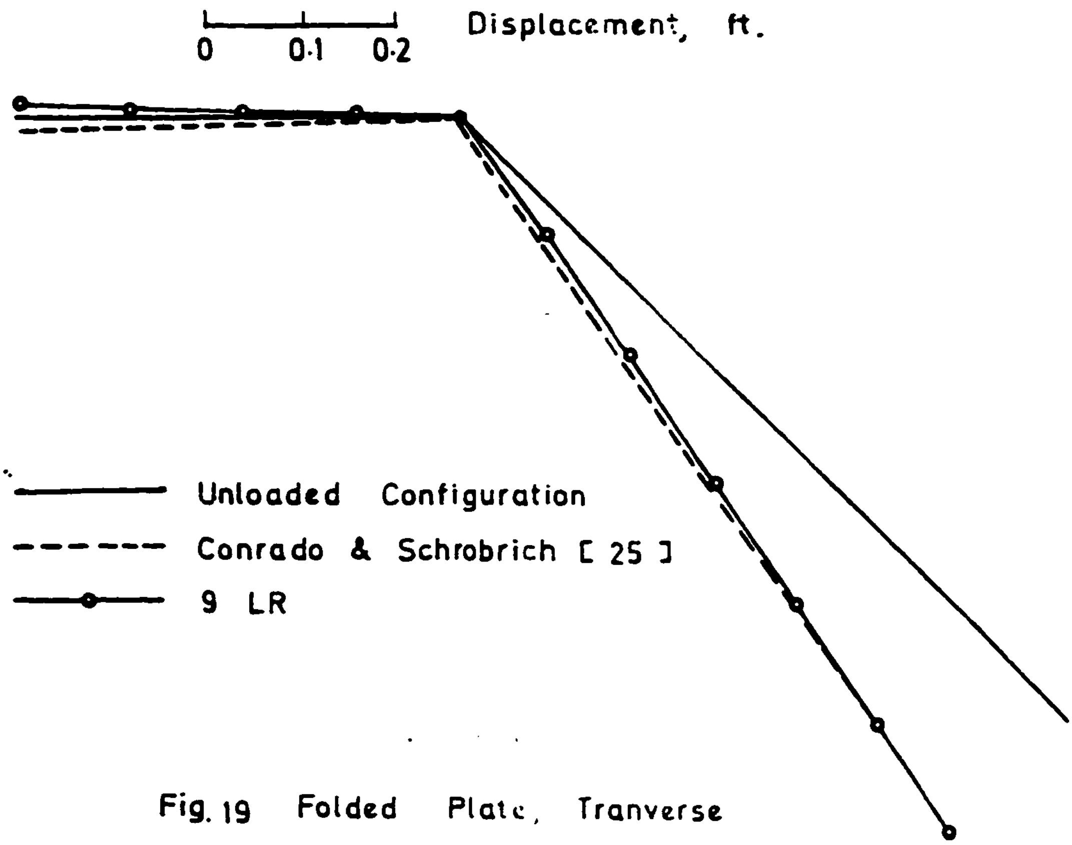


Fig. 19 Folded Plate, Tranverse Displacement Midspan

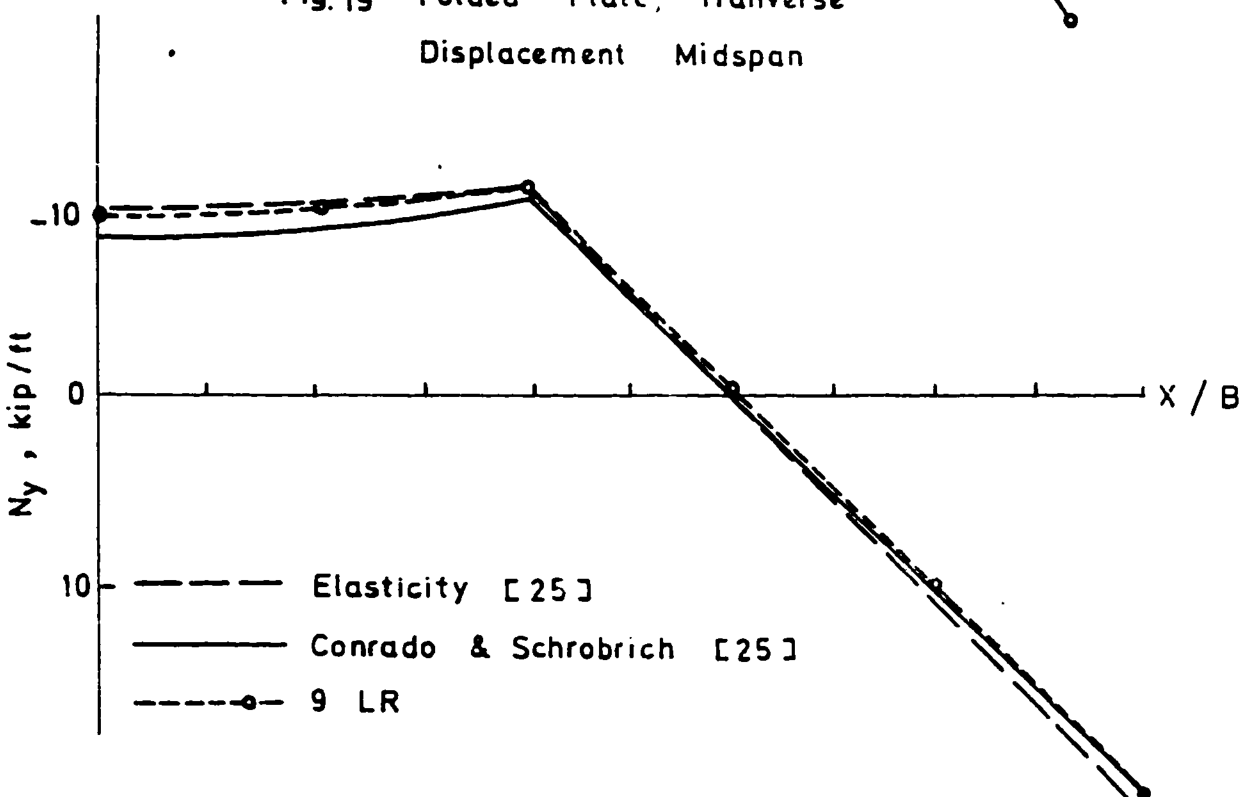


Fig. 20 Folded Plate, N_y at Mispan

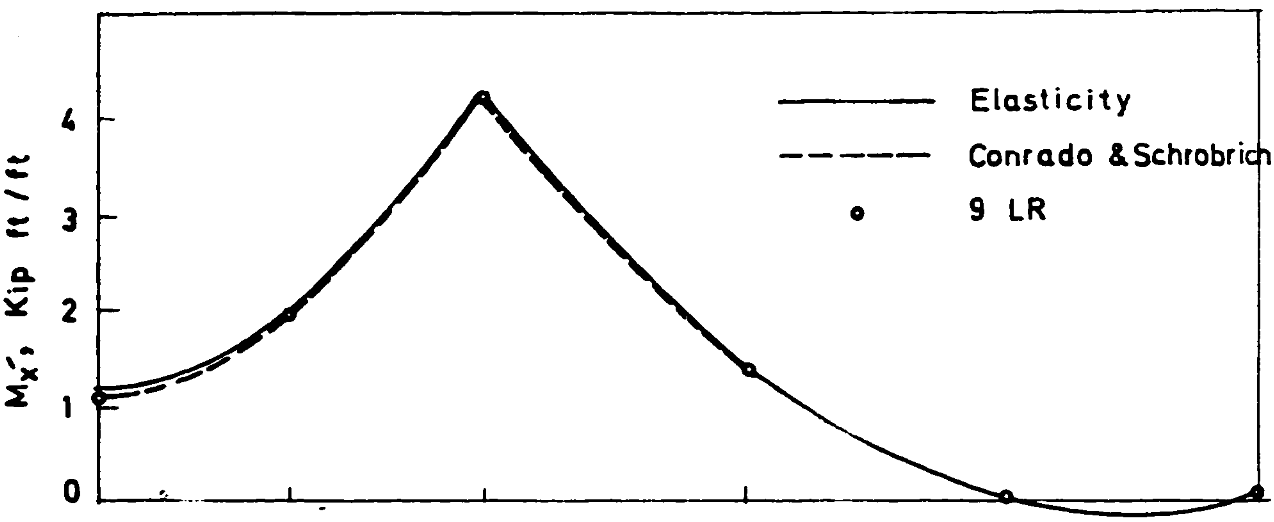


Fig. 21 Folded Plate, M_x' at Midspan

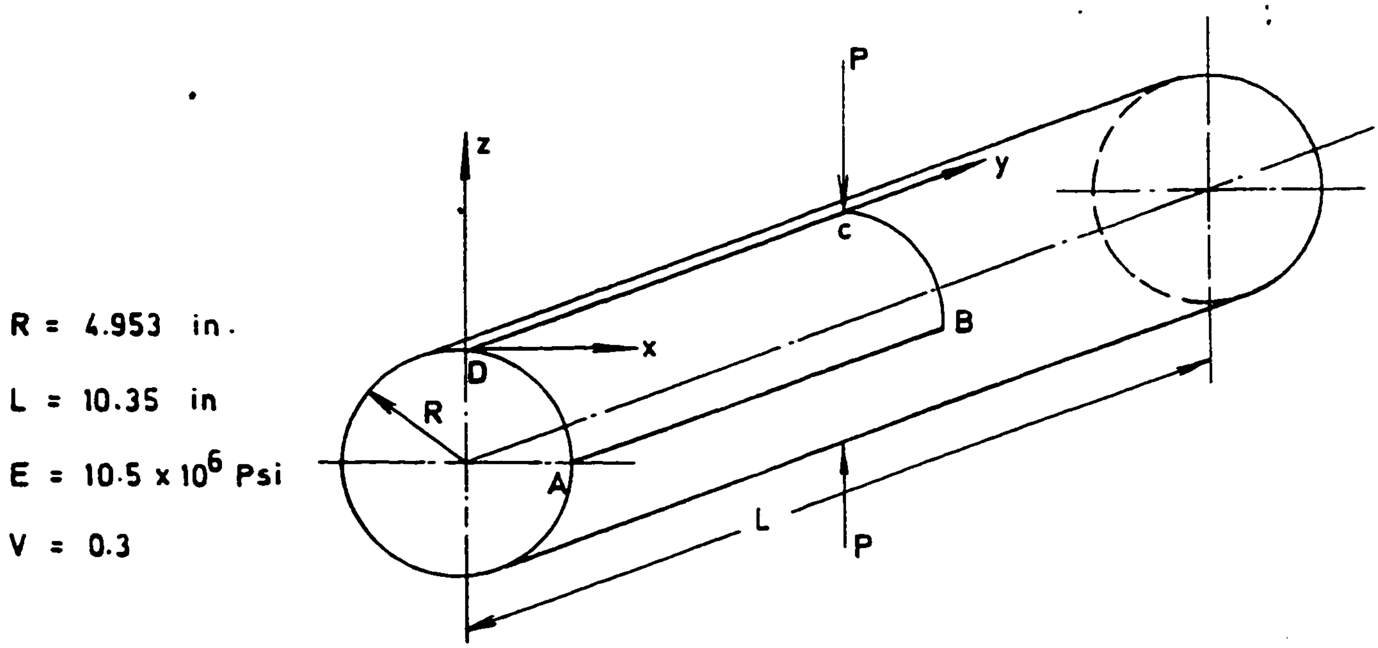


Fig. 22. Geometry and Loading, Pinched Cylindrical Shell (free ends)

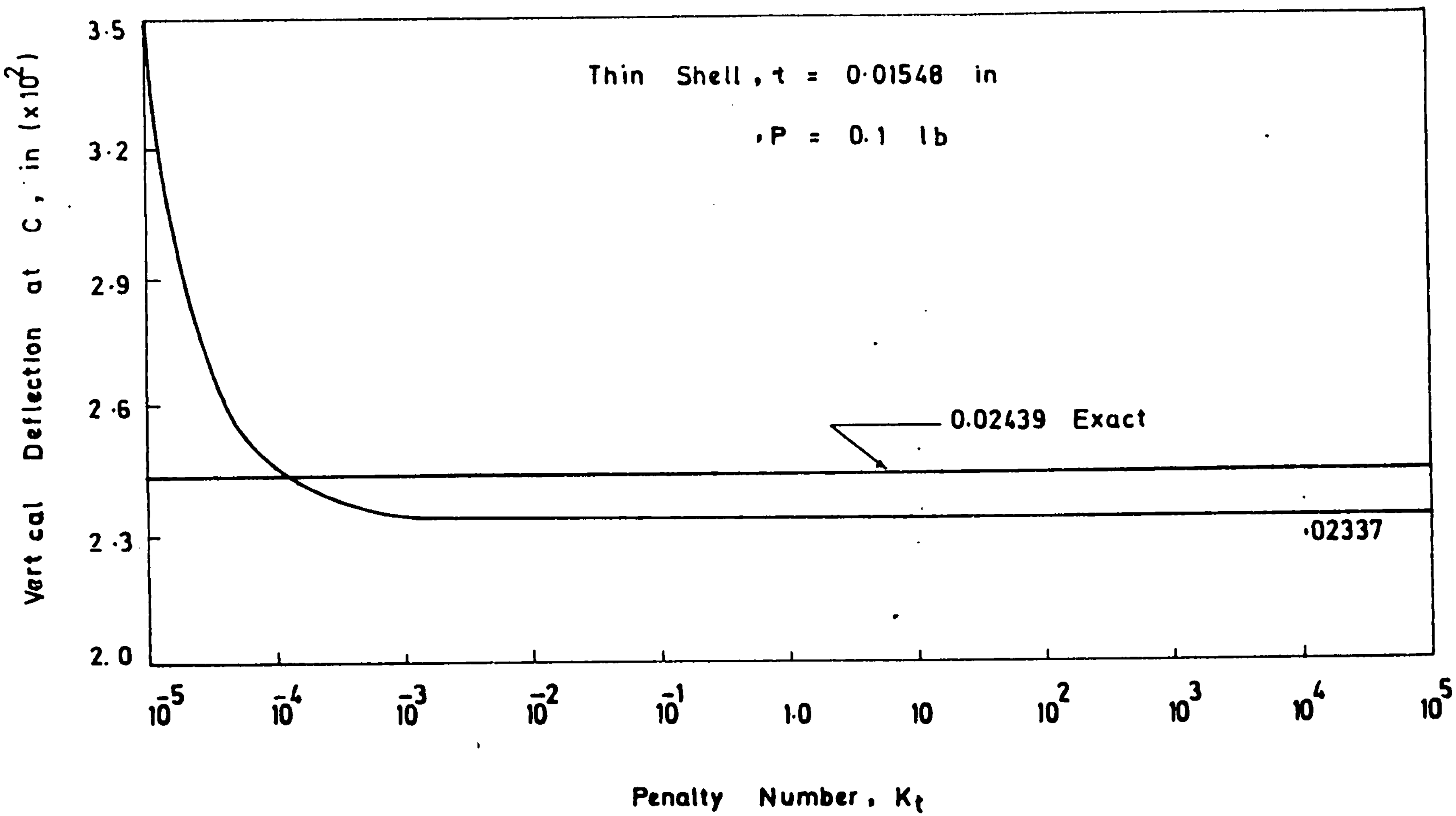


Fig.23 .Effect of Torsional Stiffness to Solution Pinched Cylindrical Shell.

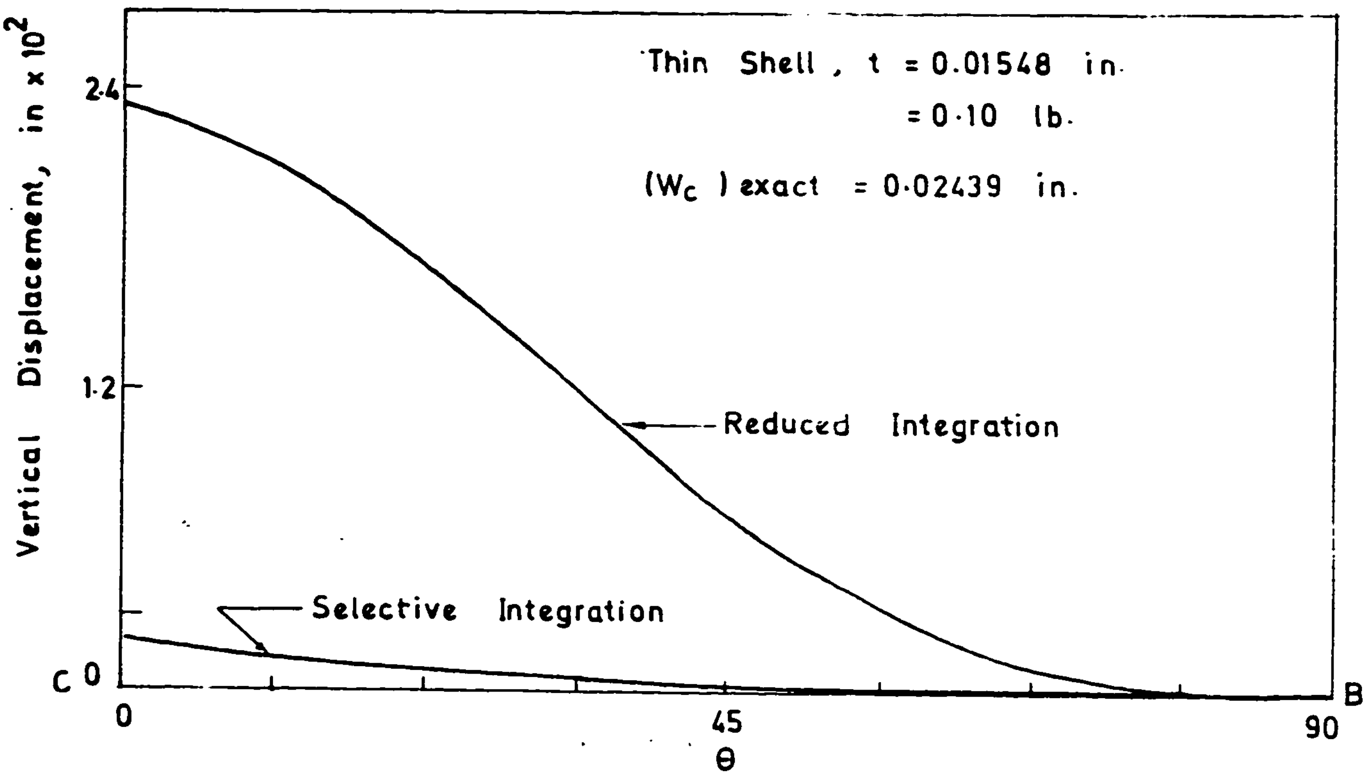


Fig. 24 Pinched Cylinder, Vertical Displacement along CB

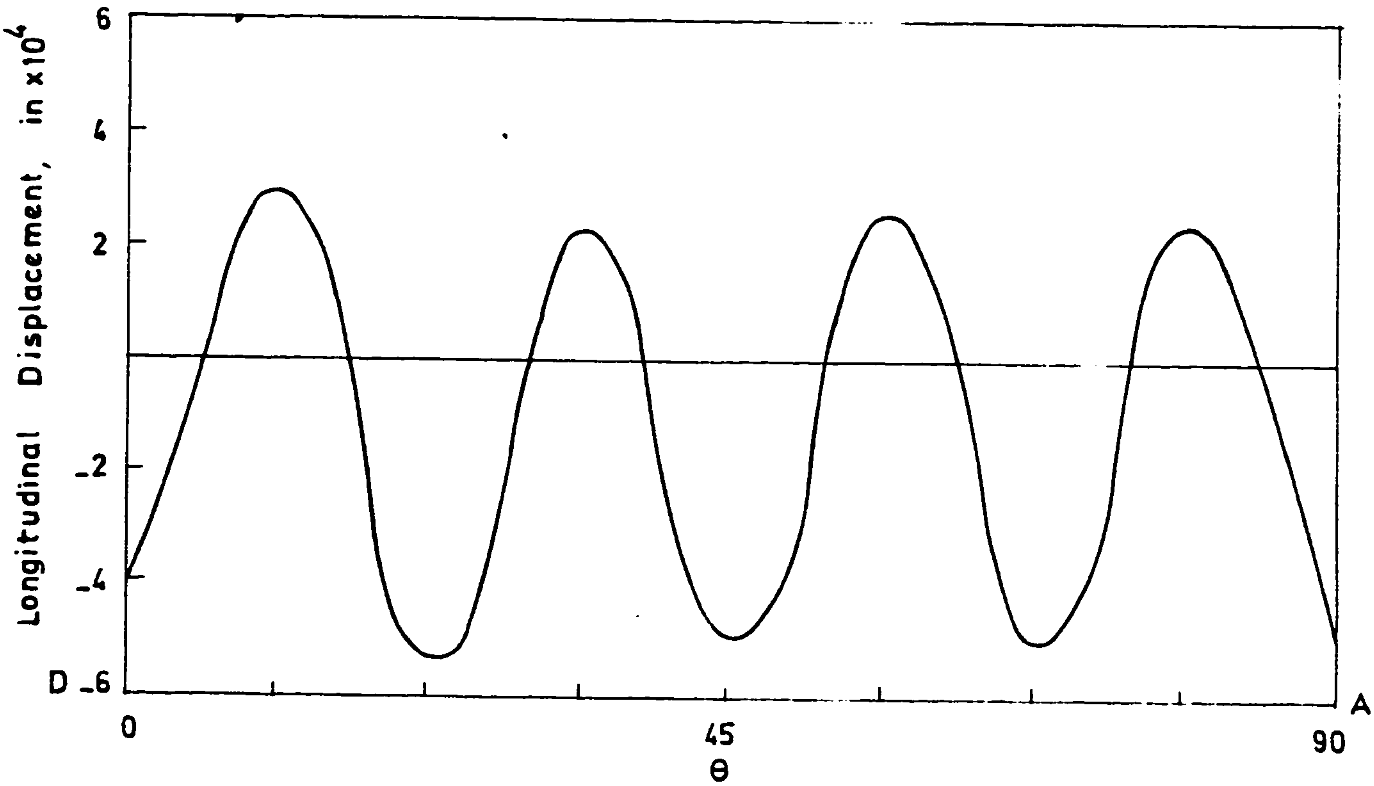


Fig 25 Pinched Cylinder, Longitudinal Displacement along DA

APPENDICES

Appendix A

The generation of the shape function polynomials of the isoparametric elements are classified into two families; the Lagrange family and the Serendipity family. Basically, the Lagrange family elements are derived from Lagrange interpolation polynomial.

$$N_I = \frac{(r - r_1)\ (r - r_2)\ \text{---}\ (r - r_{i-1})}{(r_2 - r_1)\ (r_i - r_2)\ \text{---}\ (r_i - r_{i-1})} \tag{A1}$$

N_I are (n - 1) degree polynomials. Similar Eq. (A1) is obtained in s direction. Thus, the shape function polynomials become

$$N_i\ (r,S) = \sum_I \sum_J N_I\ (r)\ N_J\ (S)$$

The Serendipity family elements are generated from the boundary nodes only rather than the boundary and internal nodes as in Lagrange family elements. The generation of such polynomials are described in references [14, 15].

As previously, these functions N_i have a value of unity at node i and zero at other nodes. The shape functions for linear, parabolic and biparabolic are shown in Figs. A1 - A2. Their polynomials are given below.

APPENDIX (CONTD.)

Shape Functions

Bilinear Element

$$N_1 = (1 - r)(1 - s)/4$$

$$N_2 = (1 + r)(1 - s)/4$$

$$N_3 = (1 + r)(1 + s)/4$$

$$N_4 = (1 - r)(1 + s)/4$$

Parabolic Element

$$N_1 = (1 - r)(1 - s)(-r-s-1)/4$$

$$N_2 = (1 - r)(1 - s)/2$$

$$N_3 = (1 + r)(1 - s)(r-s-1)/4$$

$$N_4 = (1 + r)(1 - s^2)/2$$

$$N_5 = (1 + r)(1 + s)(r+s-1)/4$$

$$N_6 = (1 - r^2)(1 + s)/2$$

$$N_7 = (1 - r)(1 + s)(-r+s-1)/4$$

$$N_8 = (1 - r)(1 - s^2)$$

Biparabolic Element

$$N_1 = rs(1 - r)(1 - s)/4$$

$$N_2 = -s(1 - r^2)(1 - s)/2$$

$$N_3 = -rs(1 + r)(1 - s)/4$$

$$N_4 = r(1 + r)(1 - s^2)/2$$

Appendix (contd.)

$$N_5 = rs(1 + r)(1 + s)/4$$

$$N_6 = s(1 - r^2)(1 + s)/2$$

$$N_7 = -rs(1 - r)(1 + s)/4$$

$$N_8 = -r(1 - r)(1 - s^2)/2$$

$$N_9 = (1 - r^2)(1 - s^2)$$

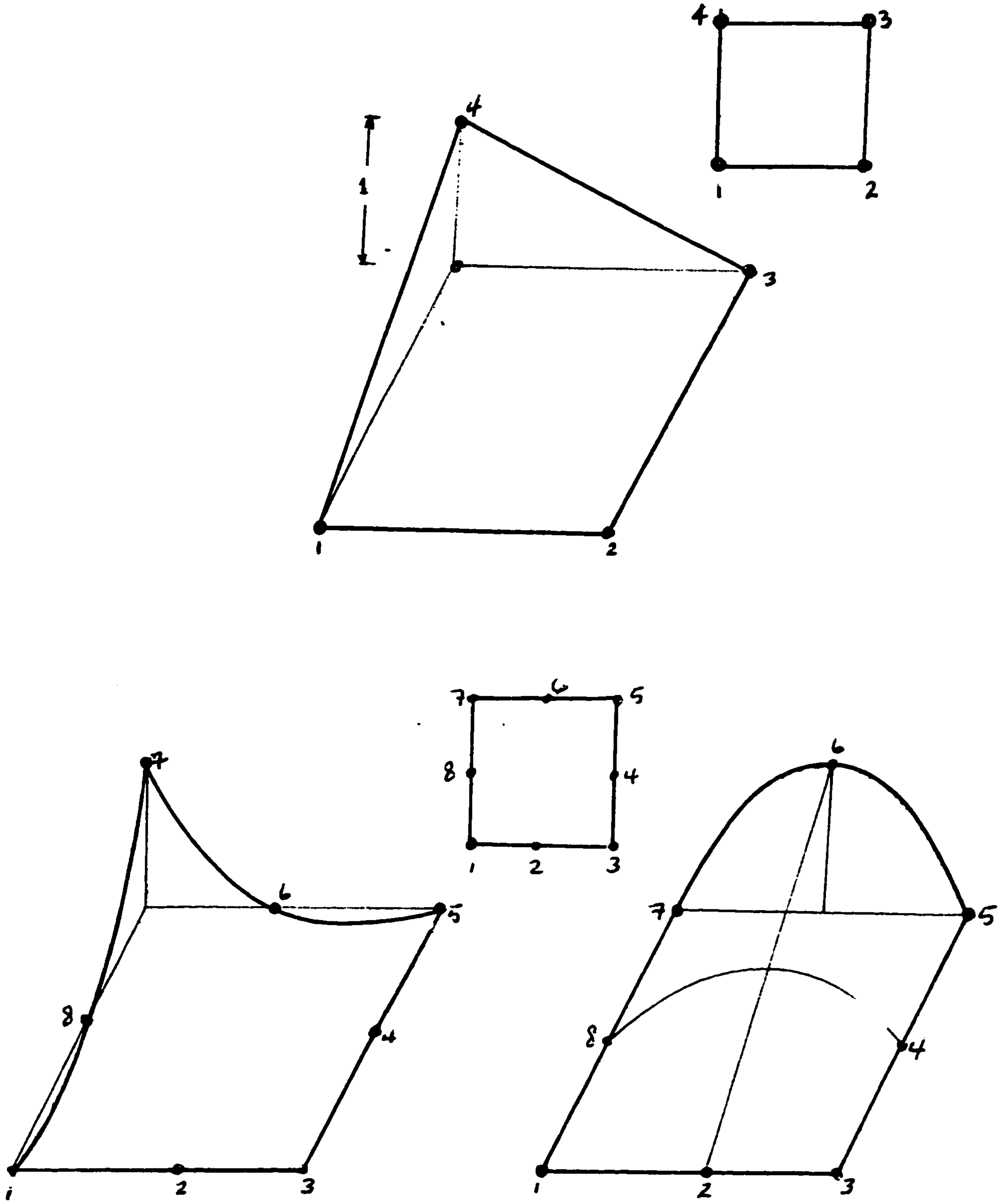


Fig. A1. Shape Functions for Bilinear and Parabolic Elements

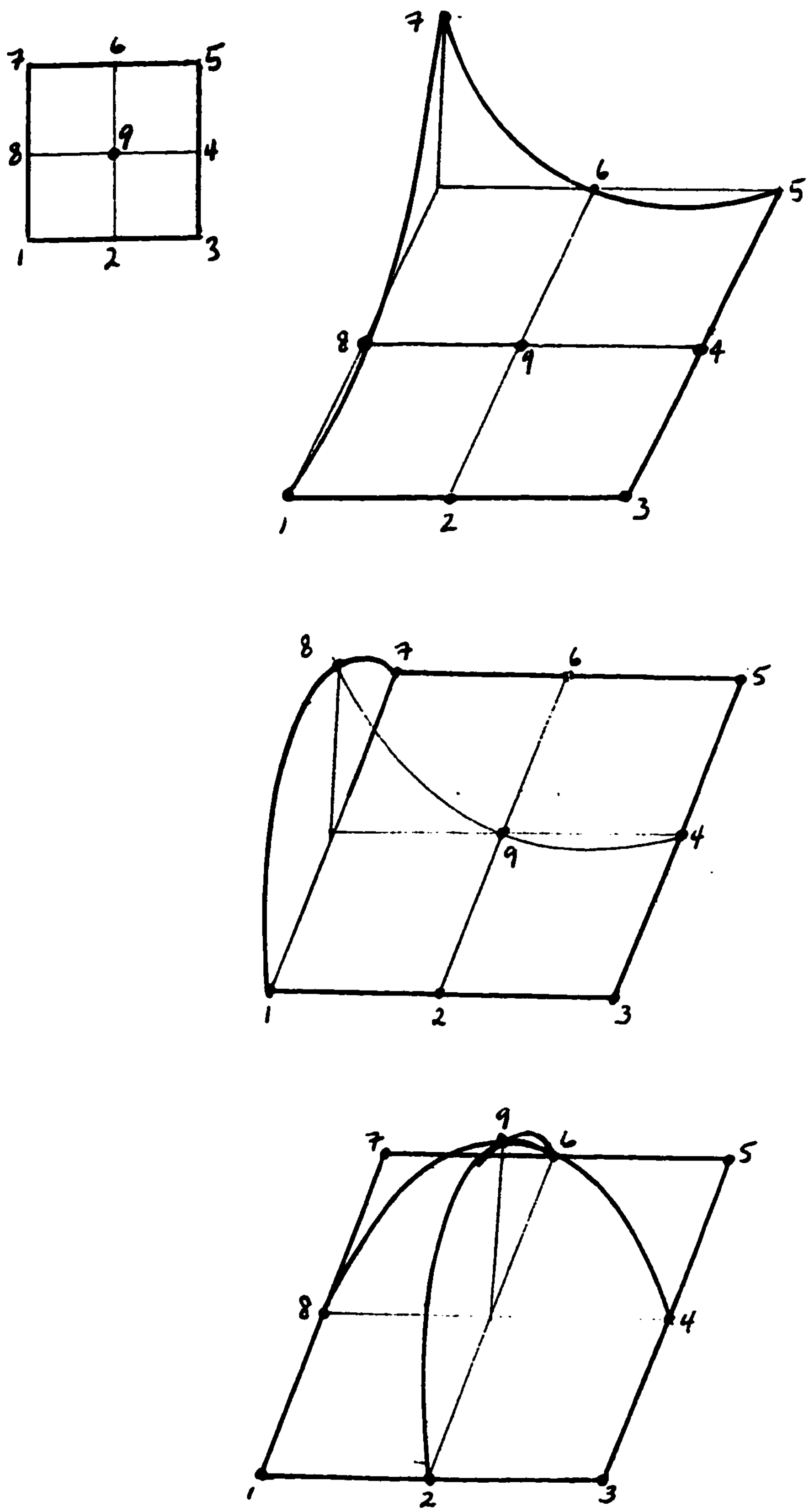


Fig. A2. For Biparabolic Elements

APPENDIX B

Shell Stress-Resultants

The shell stress-resultants which are the membrane force N_1, N_2, N_{12}, shear N_{13}, N_{23} and couples M_1, M_2 and M_{12} are defined as follows :

$$N(r,s,0) = \begin{Bmatrix} N_1 \\ N_2 \\ N_{12} \\ N_{13} \\ N_{23} \end{Bmatrix} = \int_{h/2}^{h/2} \sigma' \; (r,s,t) \; dz' \qquad \text{B1}$$

$$M(r,s,0) = \begin{Bmatrix} M_1 \\ M_2 \\ M_{12} \end{Bmatrix} = \int_{h/2}^{h/2} z'\sigma(r,s,t) \; dz' \qquad \text{B2}$$

Eqs. B1 and B2 define the membrane forces, shears and couples in the local coordinates. Substituting Eq. 2.42 in 2.21 and the result in B1 and B2 which are integrated explicitly to yield :

$$N_m = \begin{Bmatrix} N_1 \\ N_2 \\ N_{12} \end{Bmatrix} = \sum_{i=1}^{9} h_i(r,s) \; D_m B^i_{1m} \; u_i \qquad \text{B3}$$

APPENDIX B (CONTD.)

$$N_s = \begin{Bmatrix} N_{13} \\ N_{23} \end{Bmatrix} = \sum_{i=1}^{9} h_i(r,s)\ D_s \left[B_{1_s}^{i} \quad B_{2_s}^{i} \right] \begin{Bmatrix} u_i \\ \alpha_i \end{Bmatrix} \qquad \text{B4}$$

$$M = \begin{Bmatrix} M_1 \\ M_2 \\ M_{12} \end{Bmatrix} = \sum_{i=1}^{9} \frac{h^2(r,s)}{6}\ D_m\ B_{3_m}^{i} \left\{ \alpha_i \right\} \qquad \text{B5}$$

APPENDIX C

Program

ELMT03 routine is first written for the biparabolic isoparametric shell element and then modified to suit 'any' isoparametric shell element. In this version, the routine can suit 9-noded or less shell element provided its shape function and corresponding directives are in subroutine SHAPE9. The present elements :

1. Biparabolic shell element
2. Parabolic shell element
3. Bilinear shell element

Program Aspect

a) Main Program : FEAP (By Prof. R. L. Taylor, University of California, Berkeley) can be found in Chapter 24 of reference 14.

b) FEAP has been converted to double precision on IBM machine.

c) Modular Subroutines : BMTRX9, JACOB9, LOA9, LOCAL9, POINT9, SHAPE9, TORSN9.

d) Input material data (7I0.0/I5,2F10.0) E, ν, H, LCODE, κ_t, ZB, ρ, IBS, FLOAD, WTABL

where E = Elasticity modulus
ν = Possion ratio
H = Thickness (for non-uniform thickness,corresponding nodal thicknesses are input also)

APPENDIX C (CONTD.)

LCODE	=	Loading Code :
		1.0 dead load
		2.0 uniform load
		3.0 normal pressure
		4.0 water pressure
κ_t	=	torsional penalty number
ZB	=	ZB. NE. 0 means all out-of-plane stiffness neglected (membrane element)
ρ	=	material density
IBS	=	integration scheme
FLOAD	=	load intensity correspond to LCODE
WTABL	=	water table correspond to LCODE = 4.0 only

```
      SUBROUTINE ELMT03(D,UL,XL,IX,TL,S,P,NDF,NDM,NST,ISW)

 **************************************************************************
 ***                                                                    ***
 ***      ##################################################           ***
 ***      #                                                #           ***
 ***      # GENERAL ISOPARAMETRIC SHELL ELEMENT PROGRAM #              ***
 ***      #                                                #           ***
 ***      ##################################################           ***
 ***                                                                    ***
 ***                BY  RIYADH AHMED ALMUSTAFA                          ***
 ***           UNIVERSITY OF PETROLEUM AND MINERALS                     ***
 ***                DEPT. OF CIVIL ENGINEERING                          ***
 ***                     AUGUST 1978                                    ***
 ***                                                                    ***
 ***           THESIS SUBMITTED ON DECEMBER 26, 1978                    ***
 ***                                                                    ***
 **************************************************************************

( ACKNOLEDGEMENT -- THE CODE OF ELMT03 ROUTINE IS BASED ON
            FORTRAN CODE OF ELMT04 (BILINEAR SHELL ELEMENT)
            PREPARED BY WORSAK KANOKNUKULCHAI ,UNIVERSITY OF
            CALIFORNIA ,BERKELEY.)

 PROGRAM     - THIS ROUTINE IS FIRST WRITTEN FOR THE BIPARABOLIC
             - ISOPARAMETRIC SHELL ELEMENT AND THEN MODIFIED TO
             - SUIT ANY ISOPARAMETRIC SHELL ELEMENT. IN THIS VERSION
             - THE ROUTINE CAN SUIT A 9-NODED OR LESS SHELL ELEMENT.
             - THE PRESENT SHELL ELEMENTS ARE :
             -        1) BIPARABOLIC SHELL ELEMENT.
             -        2) PARABOLIC SHELL ELEMENT (AHMAD ELEMENT).
             -        3) BILINEAR SHELL ELEMENT.
             - THE ADDITION OF OTHER NEW ELEMENTS REQUIRE THE
             - FOLLOWING STEPS :
             -        1) THE ADDITION OF THEIR SHAPE FUNCTIONS AND
             -           THEIR DERIVATIVES IN SUBROUTINE SHAPE9.
             -        2) ANY SPECIAL MESH OF INTEGRATION POINTS CAN
             -           BE ADDED IN SUBROUTINE POINT9.
             -           (THE PRESENT MESHES : 1X1, 2X1, 2X2, 3X2,
             -            AND 3X3 )
             -        3) ANY EXTRA ELEMENT DATA CAN BE READ DIRECTLY
             -           THROUGH ROUTINE ELMT03.


GEOMETRY    - MAX. OF 9-NODED QUADRILATERAL IN 3-D SPACE .
VARIABLES   - 6 DOF/NODE,3 DISPLACEMENTS AND 3 ROTATIONS IN
            - GLOBAL COORDINATES.
INTERPOLATION FUNCTION
            - BILINEAR,  PARABOLIC OR BIPARABOLIC FUNCTIONS FOR
              GEOMETRY AND NODAL VARIABLES (ISOPARAMETRIC ELEMENTS)
CAPACITIES  - 1) LINEAR ELASTIC ISOTROPIC MATERIAL
            - 2) THICK/THIN SHELL WITH USER,S ASSIGNED QUADRATURES
              ON BENDING,INPLANE SHEAR AND TRANVERSE SHEAR ENERGIES
            - 3) TORSIONAL STIFFNESS CAN BE ADDED TO RESTRAIN THE
              WEAK TORSIONAL MODE DUE TO WEAK COUPLING IN FINE MESH
            - 4) 4 TYPES OF LOADS BUILT IN
              A)GRAVITY ACTION(OPPOSITE TO GLOBAL Z3 DIRECTION)
              B)UNIFORM LOAD (IN Z3 DIRECTION)
```

```
:                C) NORMAL PRESSURE(IN LOCAL X3 DIRECTION)
.                D) WATER PRESSURE (IN LOCAL X3 DIRECTION)
:                E) STRESSES WILL BE ALSO CALCULATED AT NOCAL POINTS
.                FOR PARABOLIC AND BIPARABOLIC ONLY .
.    INPUT  - FORMAT(7F10.0/I5,2F10.0)
.    D(1)-D(4),D(7),D(10),D(9),LBS/D(5),D(6) (ONLY D(4).NE.0)
.              - D(1)=E
.              - D(2)=POISSON RATIO
.              - D(3)=UNIFORM THICKNESS (IF=0,THICKNESS IS INPUT FOR
.                EACH NCDE THROUGH : FORMAT(I5,4F10.0) )
.              - D(4)=1.0 -GRAVITY ACTION, 2.0 -UNIFORM LCAC
.                3.0 -NORMAL PRESSURE, 4.C -WATER PRESSURE
.              - C(7) TORSIONAL DEFORMATICN CCEFFICIENT , WHERE
.                TCRS.ENERGY=D(7)*MU*TT*INT(RELATIVE TORS ROTATION)**2
.                RECOMMENDED VALUE = C.1 TC 1000.
.              - C(10).NE.0 CNLY INPLANE EFFECT CONSIDERED(DIAPHRAGM)
.              - D( 9)=MATERIAL DENSITY (FOR LUMPED MASS MATRIX)
.              - LBS=INTEGRATION SCHEME, E.G., LBS=444 MEANS USING
.                TOTAL 4,4,4 POINTS FCR BENCING, INPLANE SHEAR
.                PARTS AND TRANSVERSE SHEAR PART.
.                DEFAULT :
.                      441  --- BILINEAR ELEMENT
.                      444  --- PARABOLIC ANC BIPARABOLIC ELEMENTS
.              - D(5)=LCAD INTENSITY CORRESPONCING TC 1)SPECIFIC WEIGH
.                2) UNIFORM LOAD/PROJECTED AREA IN GLOBAL Z3 DIRECTION
.                3)NCRMAL PRESSURE  4) WATER SPECIFIC WEIGHT
.              - D(6)=WATER TABLE IN GLOBAL Z3 CIRECTION (CASE 4 ONLY)

     ***************************************************************
     IMPLICIT REAL*8 (A-H,C-Z)
     COMMON/SHAP9/VA(3,3),V(3),V3(3,9),BU(3,3,9),BR(3,3,9)
    1,CR(3,3,9),CT(3,3),LX(3)
     COMMON/CDATA/C,HEAD(20),NUMNP,NUMEL,NUMMAT,NEN,NEQ,IPR
     COMMON/ELDATA/ DM,N,MA,MCT,IEL,NEL
     COMMON/ELMO3/XJAC,RJAC(3,3),SHP(3,9),BN,B(2,9),C6,ISW6
     DIMENSION D(10,1),P(1),S(NST,1),XL(NDM,1),IX(1),OHM(3,3),TL(1),
    1   UL(NDF,1),EPS(5,3),XN(5,9),CHI(5),XM(3,9),XNN(5,4),XMM(3,4),
    1   SIG(5,3),RG(9),QG(9),R(9),Q(9),T(9)
     LOGICAL NPR,NPL,ISW6
     DATA R/-1.000,0.000,3*1.000,0.000,2*-1.000,0.000/
     DATA Q/3*-1.000,0.000,3*1.000,0.000,C.0C0/
...
     IF(ISW.LE.2)GC TO 999
...
...  COMMUN ROUTE
...
     LBS=D(8,MA)
     RO=D(11,MA)
...
...  IF OUT-OF-PLANE EFFECT NEGLECTED,BZERC=0.C
...
     BZERO=1.000
     IF(D(10,MA).NE.0.000)BZERO=0.000
...
...  COUNT FOR VARIABLE THICKNESSES.
...
     IF(D(3,MA).NE.0.000)GO TC 3010
     READ(5,2408)N,T
```

```
      GO TO 3030
C....
C....  UNIFORM THICKNESS CASE.
C....
 3010 DO 301 I=1,NEL
      T(I)=D(3,MA)
  301 CONTINUE
C....
C....  COMPUTE NORMALS AT EACH NODE AND STORE  IN V3
C....
 3030 II=1
      NELL=NEL
      IF(NEL.EQ.4) NELL=NEL+4
      IF(NEL.EQ.4) II=2
      DO 303 I=1,NELL,II
      RR=R(I)
      SS=Q(I)
      I4=I
      IF(NEL.EQ.4) I4=I/II + 1
      CALL LOCAL9(RR,SS,TT,NDM,NEL,OHM,R,Q,T,XL,IX)
      V3(1,I4)=OHM(1,3)
      V3(2,I4)=OHM(2,3)
      V3(3,I4)=OHM(3,3)
  303 CONTINUE
C....
  999 GO TO (1,2,3,4,5,6),ISW
C....
    1 READ(5,1100)(D(I,MA),I=1,4),D(7,MA),D(10,MA),D(11,MA)
      READ(5,1110)LBS,D(5,MA),D(6,MA)
      TEMP=1.2D0
      LB=LBS/100
      LPS=MOD(LBS,100)/10
      LS=MOD(LBS,10)
      IF(LB.EQ.0) LB=4
      IF(LS.EQ.0.AND.NEL.EQ.4) LS=1
      IF(LS.EQ.0.AND.NEL.EQ.8.OR.NEL.EQ.9) LS=4
      IF(LPS.EQ.0)LPS=LB
      WRITE(6,2100)(D(I,MA),I=1,2),D(11,MA),TEMP,LB,LPS,LS
      IF(D(10,MA).NE.0.0D0)WRITE(6,2101)
      IF(D(3,MA).NE.0.0D0) WRITE(6,2102) D(3,MA)
      IF(D(4,MA).NE.0.0D0)WRITE(6,2103)D(4,MA),D(5,MA),D(6,MA)
      D(6,MA)=LBS
      WRITE(6,2104)D(7,MA)
      C6=0.0D0
      ISW6=.FALSE.
C...
    2 RETURN
C...
    3 IF(ISW.EQ.3.AND.ISW6)GO TO 389
      LX(1)=LBS/100
      LX(2)=MOD(LBS,10)
      LX(3)=MOD(LBS,100)/10
      IF(LX(1).EQ.0)LX(1)=4
      IF(LX(2).EQ.0.AND.NEL.EQ.4)LX(2)=1
      IF(LX(2).EQ.0.AND.NEL.EQ.8.OR.NEL.EQ.9)LX(2)=4
      IF(LX(3).EQ.0)LX(3)=LX(1)
C...
C...  ENERGIES SPLIT LOOP - ISP=(1,3)=(VOLUMETRIC,INPLANE SHEAR INVARI.)
```

```
C...                       ISP= 2   =TRANSVERSE SHEAR ENERGY
C...
      ISPLT=3
      IF(LX(3).EQ.LX(1))ISPLT=2
      DO 399 ISP=1,ISPLT
      IBS=ISP
      IF(ISP.EQ.3)IBS=1
      LL=LX(ISP)
      CALL SPACKD(D(1,MA),CT,ISP,ISPLT)
C...
C...  FOR EACH GUASS POINT COMPUTE ITS CONTRIBUTION
C...  TO ELEMENT STIFFNESS MATRIX. LOOP ON GUASS POINTS.
C...
      DO 312 L=1,LL
C...
C...  LOCATE PROPER GUASS POINTS (RR,SS)
C...
      CALL POINT9(RR,SS,WW,L,LL)
      CALL LOCAL9(RR,SS,TT,NDM,NEL,OHM,R,Q,T,XL,IX)
      CALL JACOB9(RR,SS,NEL,NDM,OHM,XL,T)
      DO 304 I=1,NEL
  304 CALL BMTRX9(RR,SS,TT,NEL,NDM,OHM,I,T,IBS)
      DV=XJAC*WW*2.0D0
C...
C...  COMPUTE ELEMENTAL SURFACE AREA(SJAC) .
C...
      SJAC=DSQRT(RJAC(1,3)**2+RJAC(2,3)**2+RJAC(3,3)**2)*XJAC*WW
      DO 302 I=1,3
      DO 302 J=1,I
      RJAC(I,J)=CT(I,J)*DV
  302 RJAC(J,I)=RJAC(I,J)
      IS=4-IBS
      IF(ISW.EQ.3.AND.ISP.EQ.1.AND.D(4,MA).NE.0.0D0)
     1CALL LOAD9(NEL,NDM,NDF,D(4,MA),D(5,MA),D(6,MA),SJAC,TT,OHM,P,XL)
      J1=0
C...
C...  FOR EACH IJ NODE COMPUTE DB = D * B
C...
      DO 364 IJ=1,NEL
      K1=0
C...
C...  FOR EACH IK NODE COMPUTE  S = BT * DB
C...
      DO 362 IK=1,IJ
      CALL TRIMUL(BU(1,1,IJ),RJAC,BU(1,1,IK),S(J1+1,K1+1),1.0D0,IS,NST)
      IF(BZERO.NE.0.0D0)
     1CALL TRIMUL(BR(1,1,IJ),RJAC,BR(1,1,IK),S(J1+4,K1+4),3.0D0,IS,NST)
      IF(IBS.EQ.1)GO TO 362
      CALL TRIMUL(CR(1,1,IJ),RJAC,CR(1,1,IK),S(J1+4,K1+4),1.0D0,2,NST)
      CALL TRIMUL(BU(1,1,IJ),RJAC,CR(1,1,IK),S(J1+1,K1+4),1.0D0,2,NST)
      CALL TRIMUL(CR(1,1,IJ),RJAC,BU(1,1,IK),S(J1+4,K1+1),1.0D0,2,NST)
  362 K1=K1+NDF
  364 J1=J1+NDF
C...
C...  COMPUTE TORSIONAL ENERGY BY REDUCED INTEGRATION IF D(7) .GT.0
C...
      IF(ISP.NE.2)GO TO 312
      IF(D(7,MA).NE.0.D0)CALL TORSN9(D(1,MA),S,SJAC,TT,OHM,NST,NEL,NDF)
```

```
 312 CONTINUE
....
....  IF BZERO=0.0 THEN OUT-OF-PLANE EFFECT IS NEGLECTED.
....
      IF(BZERO.EQ.0.0D0)GO TO 398
 399 CONTINUE
....
....  COMPUTE LOWER TRIANGULAR PART BY SYMMETRY.
....
 398 DO 382 J=1,NST
     DO 382 K=1,J
 382 S(K,J)=S(J,K)
 389 RETURN
....
....  COMPUTE ELEMENT STRAINS, STRESSES, AND INTERNAL FORCES.
....  ALL ARE COMPUTED AT THE INTEGRATION POINTS.
....
   4 LX(2)=MOD(LBS,10)
     IF(LX(2).EQ.0.AND.NEL.EQ.4) LX(2)=1
     IF(LX(2).EQ.0.AND.NEL.EQ.4.OR.NEL.EQ.8) LX(2)=4
     LL=LX(2)
....
...   COMPUTE STRAINS AT INTEGRATION POINTS.
...
     DO 500 L=1,LL
     CALL POINT9(RR,SS,WW,L,LL)
     CALL LOCAL9(RR,SS,TT,NDM,NEL,CHM,R,Q,T,XL,IX)
     CALL JACOB9(RR,SS,NEL,NDM,OHM,XL,T)
     DO 450 J=1,5
     DO 449 JJ=1,3
 449 EPS(J,JJ)=0.0
 450 CHI(J)=0.0
     TEMP=2.0/TT
     XX=0.0
     YY=0.0
     ZZ=0.0
     DO 470 I=1,NEL
     SH3=SHP(3,I)
     XX=XX + XL(1,I)*SH3
     YY=YY + XL(2,I)*SH3
     ZZ=ZZ + XL(3,I)*SH3
     CALL BMTRX9(RR,SS,TT,NEL,NDM,CHM,I,T,1)
     DO 460 K=1,3
     TP=UL(K,I)
     TQ=UL(K+3,I)*TEMP
     DO 460 J=1,3
     EPS(J,1)=EPS(J,1)+BU(J,K,I)*TP+BR(J,K,I)*TQ/TEMP
     EPS(J,2)=EPS(J,2)+BU(J,K,I)*TP
     EPS(J,3)=EPS(J,3)+BU(J,K,I)*TP-BR(J,K,I)*TQ/TEMP
     IF(BZERO.NE.0.)CHI(J)=CHI(J)+BR(J,K,I)*TQ
 460 CONTINUE
     IF(BZERO.EQ.0.0)GO TO 470
     CALL BMTRX9(RR,SS,TT,NEL,NDM,OHM,I,T,2)
     DO 465 K=1,3
     TP=UL(K,I)
     TQ=UL(K+3,I)
     DO 465 J=1,2
     EPS(3+J,1)=EPS(3+J,1)+BU(J,K,I)*TP+CR(J,K,I)*TQ+BR(J,K,I)*TQ
```

```
      EPS(3+J,2)=EPS(3+J,2) + BU(J,K,I)*TP + CR(J,K,I)*TQ
      EPS(3+J,3)=EPS(3+J,3)+BU(J,K,I)*TP+CR(J,K,I)*TC-BR(J,K,I)*TQ
  465 CONTINUE
  470 CONTINUE
....
....  COMPUTE STRESSES
....
      CALL SPACKD(D(1,MA),CT,1,2)
      DO 472 I=1,3
      DO 472 K1=1,3
      SIG(K1,I)=0.0D0
      DO 471 K2=1,3
      SIG(K1,I)=SIG(K1,I)+CT(K1,K2)*EPS(K2,I)
  471 CONTINUE
  472 CONTINUE
      CALL SPACKD(D(1,MA),CT,2,2)
      DO 474 I=1,2
      DO 474 K1=1,2
      SIG(K1+3,I)=0.0D0
      DO 473 K2=1,3
      SIG(K1+3,I)=SIG(K1+3,I)+CT(K1,K2)*EPS(K2+3,I)
  473 CONTINUE
  474 CONTINUE
....
....  COMPUTE STRESS-RESULTANT
....
      CALL SPACKD(D(1,MA),CT,1,2)
      TEMP=TT**3/12.0
....
....  COMPUTE MEMBRANE FORCES (N1, N2, N12) AND COUPLES (M1, M2, M12).
....
      DO 476 J=1,3
      XN(J,L)=0.0
      XM(J,L)=0.0
      DO 475 K=1,3
      XN(J,L)=XN(J,L)+CT(J,K)*EPS(K,2)*TT
      XM(J,L)=XM(J,L)+CT(J,K)*CHI(K)*TEMP
  475 CONTINUE
  476 CONTINUE
....
....  COMPUTE SHEAR FORCES (N13, N23)
....
      CALL SPACKD(D(1,MA),CT,2,2)
      DO 480 J=1,2
      XN(3+J,L)=0.0
      DO 480 K=1,2
      XN(3+J,L)=XN(3+J,L)+TT*CT(J,K)*EPS(3+K,2)
  480 CONTINUE
      MCT=MCT-1
      IF(MCT.GT.0) GO TO 490
      MCT=4
      IF(NEL.EQ.4) WRITE(6,2411) O,HEAD
      IF(NEL.EQ.8) WRITE(6,2412) O,HEAD
      IF(NEL.EQ.9) WRITE(6,2413) O,HEAD
      WRITE(6,2401)
  490 WRITE(6,2402) N,L,XX,YY,ZZ,SIG,(EPS(J,2),J=1,5),(OHM(1,J),J=1,3),
     1               (CHI(J),J=1,3),(OHM(2,J),J=1,3),
     1               (XN(J,L),J=1,5),(OHM(3,J),J=1,3),(XM(J,L),J=1,3)
```

```
  500 CONTINUE
C...
C.....SMOOTH STRESSES BY EXTRAPOLATING GUASS POINT STRESSES TO
C.....OBTAIN NODAL STRESSES.
C...
      IF(LL.NE.4) GO TO 501
      SQ=DSQRT(3.0D0)/2.0
      FAC1=1.0 + SQ
      FAC2=-0.5
      FAC3=1.0 - SQ
      DO 503 I=1,5
      XNN(I,1)=FAC1*XN(I,1)+FAC2*(XN(I,2)+XN(I,3))+FAC3*XN(I,4)
      XNN(I,2)=FAC2*(XN(I,1)+XN(I,4))+FAC1*XN(I,2)+FAC3*XN(I,3)
      XNN(I,3)=FAC2*(XN(I,1)+XN(I,4))+FAC3*XN(I,2)+FAC1*XN(I,3)
      XNN(I,4)=FAC3*XN(I,1)+FAC2*(XN(I,2)+XN(I,3))+FAC1*XN(I,4)
  503 CONTINUE
      DO 505 I=1,3
      XMM(I,1)=FAC1*XM(I,1)+FAC2*(XM(I,2)+XM(I,3))+FAC3*XM(I,4)
      XMM(I,2)=FAC2*(XM(I,1)+XM(I,4))+FAC1*XM(I,2)+FAC3*XM(I,3)
      XMM(I,3)=FAC2*(XM(I,1)+XM(I,4))+FAC3*XM(I,2)+FAC1*XM(I,3)
      XMM(I,4)=FAC3*XM(I,1)+FAC2*(XM(I,2)+XM(I,3))+FAC1*XM(I,4)
  505 CONTINUE
      WRITE(6,2501)
      DO 507 L=1,4
      WRITE(6,2502) N,L,(XNN(J,L),J=1,5),(XMM(J,L),J=1,3)
  507 CONTINUE
  501 RETURN
    5 RETURN
    6 RETURN
C...
C... FORMATS
C...
 1100 FORMAT(7F10.0)
 1110 FORMAT(I5,7F10.0)
 2100 FORMAT(/5X,7HMODULUS,13X,1H=,G13.4
     1       /5X,13HPOISSON RATIO ,7X,1H=,G13.4
     3       /5X,20HMASS DENSITY        1H=,G13.4,17H LUMPED MASS ONLY
     2       /5X,17HCONSTRAINT FACTOR,3X,1H=,G13.4
     4       /5X,20HNO OF GAUSS POINT
     5       /5X,20H  BENDING           ,1H=,I5
     5       /5X,20H  INPLANE SHEAR     ,1H=,I5
     5       /5X,20H  TRANSVERSE SHEAR  ,1H=,I5    )
 2101 FORMAT(5X,50HOUT-OF-SURFACE EFFECT NEGLECTED(MEMBRANE ELEMENT)   )
 2102 FORMAT(5X,17HUNIFORM THICKNESS,3X,1H=,G13.4)
 2103 FORMAT(5X,12HLOADING CODE,8X,1H=,G13.4,/,
     1 36X,20H( 1-SPECIFIC WEIGHT,/,
     1 38X,24H2-UNIF. LOAD/PROJ. AREA,/,
     1 38X,18H3-NORMAL PRESSURE,/,
     1 38X,24H4-WATER PRESSURE       ),/,
     1 5X,14HLOAD INTENSITY,6X,1H=,G13.4,/,
     1 5X,20HWATER TABLE AT Z3   ,1H=,G13.4,13H(CODE=4 ONLY))
 2104 FORMAT(5X,35HCOEFFICIENT FOR TORSIONAL ENERGY =  ,G13.4)
 2411 FORMAT(A1,20A4,//5X,23HBILINEAR SHELL ELEMENT//)
 2412 FORMAT(A1,20A4,//5X,20HAHMAD SHELL ELEMENT//)
 2413 FORMAT(A1,20A4,//5X,26HBIPARABOLIC SHELL ELEMENT//)
 2401 FORMAT(9H ELM NODE,3X,
     1       7H1-COORD,4X,7H2-COORD,4X,7H3-COORD,4X,11H1-DIRECTION,4X,
     2       11H2-DIRECTION,7X,8H12-SHEAR,7X,8H13-SHEAR,7X,8H23-SHEAR/)
```

```
:402 FORMAT(2I4,3F11.3/
   1        30X,11H+1-STRESSES, 5E15.5/
   1        30X,11H 0-STRESSES, 5E15.5/
   1        30X,11H-1-STRESSES, 5E15.5/ 6X,18H DIRECTION COSINES,6X
   1           ,11HEPS-STRAINS, 5E15.5/6X,3F6.2,6X
   1           ,11HCHI-STRAINS, 3E15.5/6X,3F6.2,6X
   1           ,11HRESULTANTS , 5E15.5/6X,3F6.2,6X
   1           ,11HCOUPLES    , 3E15.5)
501 FORMAT(//5X,26HBIPARABOLIC SHELL ELEMENT//,
   1        5X,26HEXTRAPOLATED NODAL VALUES///,
   1        9H ELM NODE,36X,11H1-DIRECTION,4X,
   2        11H2-DIRECTION,7X,8H12-SHEAR,7X,8H13-SHEAR,7X,8H23-SHEAR/)
502 FORMAT(2I4,22X,11HRESULTANTS     ,5E15.5/30X,
   2        11HCOUPLES          3E15.5)
408 FORMAT(I5,4F10.0)
    END
```

```
      SUBROUTINE BMTRX9(RR,SS,TT,NEL,NDM,OHM,I,T,ISW)
...
...   SUBROUTINE TO COMPUTE BMU, BMR  WHERE
...                  EM= BMU*U+Z*BMR*ROTATION
...
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION OHM(3,3),T(1)
      COMMON/SHAP9/VA(3,3),VV(3),V3(3,9),BMU(3,3,9),BMR(3,3,9)
     1,CR(3,3,9),CT(3,3),LX(3)
      COMMON/ELMO3/XJAC,RJAC(3,3),SHP(3,9),BN,B(2,9),C6,ISW6
      IF(ISW.NE.1) GO TO 100
      DO 20 J=1,3
      DO 20 K=1,3
      IF(J.EQ.3) GO TO 10
      BMU(J,K,I)=B(J,I)*OHM(K,J)
      BMR(J,K,I)=0.0
      DO 5 L=1,3
      IF(L.EQ.K) GO TO 5
      SIGN=1.0D0
      IF(L.GT.K)SIGN=-1.0D0
      LK=6-L-K
      BC=V3(LK,I)
      IF(LK.EQ.2)BC=-BC
      BMR(J,K,I)=BMR(J,K,I)+OHM(L,J)*BC*SIGN*T(I)/2.0
    5 CONTINUE
      GO TO 20
   10 BMU(3,K,I)=B(2,I)*OHM(K,1)+B(1,I)*OHM(K,2)
      BMR(3,K,I)=B(1,I)*BMR(2,K,I)+B(2,I)*BMR(1,K,I)
   20 CONTINUE
      DO 30 J=1,2
      DO 30 K=1,3
   30 BMR(J,K,I)=BMR(J,K,I)*B(J,I)
      RETURN
..
..       ES=BSU*U+Z*BSR*R+CSR*ROTATION
..
  100 FAC=BN*SHP(3,I)
      DO 60 J=1,3
      DO 60 K=1,3
      CR(J,K,I)=0.0
      DO 65 L=1,3
      IF(L.EQ.K)GO TO 65
      SIGN=1.0D0
      IF(L.GT.K)SIGN=-1.0D0
      LK=6-L-K
      BC=V3(LK,I)
      IF(LK.EQ.2)BC=-BC
      CR(J,K,I)=CR(J,K,I)+GHM(L,J)*BC*SIGN*T(I)/2.0
   65 CONTINUE
      IF(J.LT.3)CR(J,K,I)=CR(J,K,I)*FAC
   60 BMU(J,K,I)=B(J,I)*OHM(K,3)
      DO 70 J=1,2
      DO 70 K=1,3
   70 BMR(J,K,I)=B(J,I)*CR(3,K,I)
      RETURN
      END
```

```
      SUBROUTINE JACOB9(RR,SS,NEL,NDM,OHM,XL,T)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/ELM03/XJAC,RJAC(3,3),SHP(3,9),BN,B(2,9),C6,ISW6
      COMMON/SHAP9/AJAC(3,3),VA(3),V3(3,9),BU(3,3,9),BR(3,3,9)
     1,CR(3,3,9),CT(3,3),LX(3)
      DIMENSION OHM(3,3),XL(NDM,1),T(1)
C...
C...  FIND JACUBIAN MATRIX, DETERMINANT AND INVERSE,AJAC,XJAC,RJAC
C...
      DO 35 K=1,NDM
      AJAC(K,3)=0.0
      DO 35 J=1,2
      AJAC(K,J)=0.0
      DO 30 I=1,NEL
   30 AJAC(K,J)=AJAC(K,J)+SHP(J,I)*XL(K,I)
   35 CONTINUE
      DO 20 I=1,NEL
      TP=SHP(3,I)*T(I)/2.0
      DO 25 K=1,3
   25 AJAC(K,3)=AJAC(K,3)+V3(K,I)*TP
   20 CONTINUE
      XJAC=0.0
      DO 40 J=1,NDM
      J2=J+1
      J3=J+2
      IF(J3.GT.3) J3=J3-3
      IF(J2.GT.3)J2=J2-3
   40 XJAC=XJAC+AJAC(1,J)*(AJAC(2,J2)*AJAC(3,J3)-AJAC(2,J3)*AJAC(3,J2))
      CALL CROSSP(AJAC(1,2),AJAC(1,3),RJAC(1,1),1)
      CALL CROSSP(AJAC(1,3),AJAC(1,1),RJAC(1,2),1)
      CALL CROSSP(AJAC(1,1),AJAC(1,2),RJAC(1,3),1)
      DO 50 J=1,NDM
      DO 50 K=1,NDM
   50 RJAC(J,K)=RJAC(J,K)/XJAC
C...
C...  TO DETERMINE B=L(N) AND BN=L(ZETA)
C...  B(1,I)=OHM(1).(N(I)),X
C...  B(2,I)=OHM(2).(N(I)),X
C...  BN=OHM(3).(ZETA),X
C...
      DO 14 J=1,2
      DO 14 I=1,NEL
      B(J,I)=0.0
      DO 12 K=1,3
      VA(K)=0.0
      DO 10  L=1,2
   10 VA(K)   =VA(K)+RJAC(K,L)*SHP(L,I)
   12 B(J,I)=B(J,I)+VA(K)*OHM(K,J)
   14 CONTINUE
      BN=0.0
      DO 15 J=1,3
   15 BN=BN+OHM(J,3)*RJAC(J,3)
      RETURN
      END
```

```
      SUBROUTINE LOAD9(NEL,NDM,NDF,TYPE,F,XR,WW,TT,OHM,P,XL)
C...
C...  SUBROUTINE TO COMPUTE GENERALIZED LOAD  CCRRESPONDING TO EACH LOAD
C...  CASES
C...
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/SHAP9/VA(3,3),V(3),V3(3,9),BU(3,3,9),BR(3,3,9)
     1,CR(3,3,9),CT(3,3),LX(3)
      COMMON/ELMO3/XJAC,RJAC(3,3),SHP(3,9),BN,B(2,9),C6,ISW6
      DIMENSION OHM(3,3) ,P(1),XL(NDM,1)
      NJ=1
      JJ=0
      IF(TYPE.GE.3.0)NJ=3
      H=1.0D0
      IF(TYPE.LE.3.0)GO TO 20
      H=0.0
      DO 25 I=1,NEL
   25 H=H+SHP(3,I)*XL(3,I)
      H=(XR-H)
      IF(H.LT.0.0)H=0.0
   20 DO 32 I=1,NEL
      DO 30 K=1,NJ
      J=4-K
      A=OHM(J,3)*H
      IF(TYPE.EQ.2.0)A=CHM(3,3)
      IF(TYPE.EQ. 1.0D0)A=-TT
   30 P(JJ+J)=P(JJ+J)+SHP(3,I)*F*WW*A
   32 JJ=JJ+NDF
      RETURN
      END
```

```
      SUBROUTINE LOCAL9(RR,SS,TT,NDM,NEL,OHM,S,R,T,XL,IX)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/SHAP9/VA(3,3),V(3),V3(3,9),BU(3,3,9),BR(3,3,9)
     1,CR(3,3,9),CT(3,3),LX(3)
      COMMON/ELM03/XJAC,RJAC(3,3),SHP(3,9),BN,B(2,9),C6,ISW6
      DIMENSION OHM(3,3),R(1),S(1),T(1),XL(NDM,1),IX(1)
      TT=0.0
      CALL SHAPE9(RR,SS,NEL)
      DO 1  I=1,NEL
    1 TT=TT+SHP(3,I)*T(I)
C...
C...  FIND DIRECTION COSINE OF LOCAL AXES BETWEEN ANY POINT(RR,SS) AND
C...  GLOBAL  AXIS
C...
      DO 10 J=1,NDM
      OHM(J,1)=0.0
      OHM(J,2)=0.0
      DO 10 I=1,NEL
      OHM(J,1)=OHM(J,1)+SHP(1,I)*XL(J,I)
   10 OHM(J,2)=OHM(J,2)+SHP(2,I)*XL(J,I)
      CALL CROSSP(OHM(1,1),OHM(1,2),OHM(1,3),0)
      DO 11 J=1,NDM
      AOHM=DABS(OHM(J,3))
      IF(AOHM.LE.1.0D-10) OHM(J,3)=0.0DO
   11 CONTINUE
C...
C...  FIND SHP(1,I) AT (0.0,0.0) AS IN EQ. (4)
C...
      CALL SHAPE9(0.0D0,0.0D0,NEL)
      DO 12 J=1,NDM
      OHM(J,1)=0.0D0
      DO 12 I=1,NEL
   12 OHM(J,1)=OHM(J,1)+SHP(1,I)*XL(J,I)
      CALL CROSSP(OHM(1,3),OHM(1,1),OHM(1,2),0)
      CALL CROSSP(OHM(1,2),OHM(1,3),OHM(1,1),0)
C...
C...  RESTORE SHAPES AND DERIVATIVES VALUES AT (RR,SS)
C...
      CALL SHAPE9(RR,SS,NEL)
      RETURN
      END
```

```
      SUBRUUTINE POINT9(RR,SS,WW,L,LL)
C...
C...  LOCATE PRCPER GUASS POINT (RR,SS) CONSISTENT INTEGRATION SCHEME .
C...
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION R9(9),S9(9),R6(6),S6(6),R4(4),S4(4),S2(2),L9(9),L6(6)
      DATA R9/3*-1.0D0,3*0.0D0,3*1.0D0/
      DATA S9/-1.0D0,0.0D0,1.0D0,-1.0D0,0.0D0,1.0D0,-1.0D0,0.0DC,1.0D0/
      DATA L9/25,40,25,40,64,40,25,40,25/
      DATA R6/-1.0D0,1.0D0,-1.0D0,1.0D0,-1.0D0,1.0D0/
      DATA S6/-1.0D0,-1.0D0,0.0D0,0.0D0,1.0D0,1.0D0/
      DATA L6/5,5,8,8,5,5/
      DATA R4/-1.0D0,1.0D0,-1.0D0,1.0D0/
      DATA S4/-1.0DC,-1.0D0,1.0D0,1.0D0/
      DATA S2/-1.0D0,1.0D0/
C...
C...  FIND WHICH GUASS POINT IS NEEDED .
C...
      IF(LL.NE.1) GO TO 50
      RR=0.0D0
      SS=0.0D0
      WW=4.0D0
      RETURN
   50 IF(LL.NE.2) GC TO 70
      SQ=DSQRT(3.0D0)
      RR=0.0D0
      SS=S2(L)/SQ
      WW=2.0D0
      RETURN
   70 IF(LL.NE.4) GO TO 100
      SQ=DSQRT(3.0D0)
      RR=R4(L)/SQ
      SS=S4(L)/SQ
      WW=1.0D0
      RETURN
  100 IF(LL.NE.6) GO TO 200
      SQR=DSQRT(3.0D0)
      SQS=DSQRT(0.6D0)
      H=1.0D0/9.0D0
      RR=R6(L)/SQR
      SS=S6(L)*SQS
      WW=H*L6(L)
      RETURN
  200 IF(LL.NE.9) GC TO 300
      SQ=DSQRT(0.6D0)
      H=1.0D0/81.0D0
      RR=R9(L)*SQ
      SS=S9(L)*SQ
      WW=H*L9(L)
      RETURN
  300 WRITE(6,3001)LL
 3001 FORMAT(5X,55H** FATAL ERROR ** THERE IS NC SUCH INTEGRATION SCHEME
     1 =,I5)
      STOP
      END
```

```
      SUBROUTINE SHAPE9(R,S,NEL)
C...
C...  THIS SUBROUTINE CALCULATES THE SHAPE FUNCTIONS AND
C...  DERIVATIVES FOR THE 4-NODE LAGRANGIAN ELEMENT,
C...               OR THE 8-NODE SERENDIPITY ELEMENT,
C...               OR THE 9-NODE LAGRANGIAN ELEMENT.
C...
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/ELM03/XJAC,RJAC(3,3),SHP(3,9),BN,B(2,9),C6,ISW6
      DIMENSION RV(4),SV(4)
      DATA RV/-1.0D0,1.0D0,1.0D0,-1.0D0/,SV/-1.0D0,-1.0D0,1.0D0,1.0D0/
C...
      R2=R*2.0D0
      S2=S*2.0D0
      RR=R*R
      SS=S*S
      RS=R*S
      RRS=RR*S
      RSS=R*SS
      RS2=R*S*2.0D0
      RRSS=RR*SS
C...
C...  SHAPE FUNCTIONS .4-NODE ELEMENT.
C...
      IF(NEL.NE.4) GO TO 100
      DO 50 I=1,4
      SHP(1,I)=0.25D0*RV(I)*(1.0D0+SV(I)*S)
      SHP(2,I)=0.25D0*SV(I)*(1.0D0+RV(I)*R)
      SHP(3,I)=0.25D0*(1.0D0+SV(I)*S)*(1.0D0+RV(I)*R)
   50 CONTINUE
      RETURN
  100 IF(NEL.NE.8) GO TO 200
C...
C...  SHAPE FUNCTIONS .8-NODE ELEMENT.
C...
      SHP(3,1)=(-1.0+RS+RR+SS-RRS-RSS)/4.0
      SHP(3,2)=(1.0-S-RR+RRS)/2.0
      SHP(3,3)=(-1.0-RS+RR+SS-RRS+RSS)/4.0
      SHP(3,4)=(1.0+R-SS-RSS)/2.0
      SHP(3,5)=(-1.0+RS+RR+SS+RRS+RSS)/4.0
      SHP(3,6)=(1.0+S-RR-RRS)/2.0
      SHP(3,7)=(-1.0-RS+RR+SS+RRS-RSS)/4.0
      SHP(3,8)=(1.0-R-SS+RSS)/2.0
C...
C... SHAPE FUNCTION DERIVATIVES
C...
      SHP(1,1)=(S+R2-RS2-SS)/4.0
      SHP(1,2)=-R+RS
      SHP(1,3)=(-S+R2-RS2+SS)/4.0
      SHP(1,4)=(1.0-SS)/2.0
      SHP(1,5)=(S+R2+RS2+SS)/4.0
      SHP(1,6)=-R-RS
      SHP(1,7)=(-S+R2+RS2-SS)/4.0
      SHP(1,8)=(-1.0+SS)/2.0
C...
C... SHAPE FUNCTION DERIVATIVES
C...
      SHP(2,1)=(R+S2-RR-RS2)/4.0
```

```
      SHP(2,2)=(-1.0+RR)/2.0
      SHP(2,3)=(-R+S2-RR+RS2)/4.0
      SHP(2,4)=-S-RS
      SHP(2,5)=(R+S2+RR+RS2)/4.0
      SHP(2,6)=(1.0-RR)/2.0
      SHP(2,7)=(-R+S2+RR-RS2)/4.0
      SHP(2,8)=-S+RS
      RETURN
  200 IF(NEL.NE.9) GO TO 300
...
...   SHAPE FUNCTIONS .9-NODE ELEMENT.
...
      SHP(3,1)=0.25D0*(RRSS-RRS-RSS+RS)
      SHP(3,2)=-0.5D0*(RRSS-SS-RRS+S)
      SHP(3,3)=-0.25D0*(RS+RRS-RSS-RRSS)
      SHP(3,4)=-0.5D0*(RSS+RRSS-R-RR)
      SHP(3,5)=0.25D0*(RS+RSS+RRS+RRSS)
      SHP(3,6)=0.5D0*(SS+S-RRS-RRSS)
      SHP(3,7)=0.25D0*(RRSS-RSS+RRS-RS)
      SHP(3,8)=0.5D0*(RSS-R-RRSS+RR)
      SHP(3,9)=RRSS-RR-SS+1.0D0
...
...
...   SHAPE FUNCTION DERIVATIVES
...
      SHP(1,1)=0.25D0*(2.0D0*RSS-SS-2.0D0*RS+S)
      SHP(1,2)=-0.5D0*(2.0D0*RSS-2.0D0*RS)
      SHP(1,3)=-0.25D0*(S+2.0D0*RS-SS-2.0D0*RSS)
      SHP(1,4)=-0.5D0*(SS+2.0D0*RSS-1.0D0-2.0D0*R)
      SHP(1,5)=0.25D0*(S+2.0D0*RS+SS+2.0D0*RSS)
      SHP(1,6)=0.5D0*(-2.0D0*RS-2.0D0*RSS)
      SHP(1,7)=-0.25D0*(S+SS-2.0D0*RS-2.0D0*RSS)
      SHP(1,8)=0.5D0*(SS-1.0D0-2.0D0*RSS+2.0D0*R)
      SHP(1,9)=2.0D0*RSS-2.0D0*R
...
...   SHAPE FUNCTION DERIVATIVES
...
      SHP(2,1)=0.25D0*(2.0D0*RRS-RR-2.0D0*RS+R)
      SHP(2,2)=-0.5D0*(2.0D0*RRS-2.0D0*S-RR+1.0D0)
      SHP(2,3)=-0.25D0*(R-2.0D0*RS+RR-2.0D0*RRS)
      SHP(2,4)=-0.5D0*(2.0D0*RS+2.0D0*RRS)
      SHP(2,5)=0.25D0*(R+2.0D0*RS+RR+2.0D0*RRS)
      SHP(2,6)=0.5D0*(2.0D0*S+1.0D0-RR-2.0D0*RRS)
      SHP(2,7)=-0.25D0*(R-RR+2.0D0*RS-2.0D0*RRS)
      SHP(2,8)=0.5D0*(2.0D0*RS-2.0D0*RRS)
      SHP(2,9)=2.0D0*RRS-2.0D0*S
      RETURN
  300 WRITE(6,3001)NEL
 3001 FORMAT(5X,55H** FATAL ERROR ** THERE IS NO SUCH ELEMENT WITH NODES
     1 =,I5)
      STOP
      END
```

```
      SUBROUTINE TORSN9(D,S,WS,TT,OHM,NST,NEL,NDF)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION D(1),S(NST,1),OHM(3,3),VA(3,9)
      COMMON/ELM03/XJAC,RJAC(3,3),SHP(3,9),BN,B(2,9),C6,ISW6
      COMMON/SHAP9/VV(3,4),V3(3,9),BU(3,3,9),BR(3,3,9)
     1,CR(3,3,9),CT(3,3),LX(3)
      GG=CT(1,1)*1.2
      GW=WS*GG*TT*D(7)
      DO 10  IJ=1,NEL
      DO 10 K=1,3
   10 VA(K,IJ)=0.5D0*(OHM(K,1)*B(2,IJ)-OHM(K,2)*B(1,IJ))
      J1=0
      DO 1 IJ=1,NEL
      K1=0
      DO 20 IK=1,IJ
      GW2=GW*SHP(3,IJ)*SHP(3,IK)
      DO 40 J=1,3
      DO 40 K=1,3
      S(J1+J,K1+K)=S(J1+J,K1+K)+VA(J,IJ)*VA(K,IK)*GW
      S(J1+3+J,K1+3+K)=S(J1+3+J,K1+3+K)+OHM(J,3)*OHM(K,3)*GW2
      S(J1+J  ,K1+3+K)=S(J1+J  ,K1+3+K)+VA(J,IJ)*SHP(3,IK)*OHM(K,3)*GW
   40 S(J1+J+3,K1+K  )=S(J1+J+3,K1+K  )+SHP(3,IJ)*OHM(J,3)*VA(K,IK)*GW
   20 K1=K1+NDF
    1 J1=J1+NDF
      RETURN
      END
```

```
      SUBROUTINE TRIMUL(A,B,C,ABC,FAC,IB,NST)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION A(3,3),B(3,3),C(3,3),ABC(NST,3),S(3)
      DO 250 L=1,3
      DO 150 J=1,IB
      TEMP=0.0
      DO 100 K=1,IB
  100 TEMP=TEMP+B(J,K)*C(K,L)
  150 S(J)=TEMP
      DO 250 I=1,3
      TEMP=0.0
      DO 200 J=1,IB
  200 TEMP=TEMP+A(J,I)*S(J)
  250 ABC(I,L)=ABC(I,L)+TEMP/FAC
      RETURN
      END
```

```
//FACRIYAD    JOB 418-14CI49EN01,'RIYACH MUSTAFA',CLASS=W,
//             MSGCLASS=T
//          EXEC  PGM=FEAP90
//STEPLIB  DD  DSN=ACS.FACLIB2,DISP=SHR
//FT06F001  DD  SYSOUT=T
//FT07F001  DD  SYSOUT=A
//FT05F001  DD  *
FEAP  ** CYLINDRICAL SHELL TEST **
    9    1    1    3    6    9
COOR
CART
    1    3
    7    0              25.0
    2    3   8.5505              -1.5077
    8    0   8.5505     25.0     -1.5077
    3    3  16.0697              -5.8489
    9    0  16.0697     25.0     -5.8489

MATERIAL
   1ELM03
432.0   +3             0.25       1.0
 444    0.3600
ELEMENT
    1    1    3    6    9    8    7    4    1    2    5
BOUNDARY
    5    0
    6    0
    3        -1    0   -1    0
    2        -1    0   -1    0
    1    1   -1    0   -1    0   -1   -1
    4        -1    0    0    0   -1   -1
    7    1   -1   -1    0   -1   -1   -1
    9         0   -1    0   -1    0   -1

ND
MACR
TANG
SOLV
DISP
STRE
ND
TOP
/*
//
```

FEAP ** CYLINDRICAL SHELL TEST ** 00000080

NODAL DISPLACEMENTS TIME 0.0

NODE	1 COORD	2 COORD	3 COORD	1 DISPL	2 DISPL	3 DISPL	4 DISPL	5 DISPL	6 DISPL
1	0.0	0.0	0.0	0.0	0.7746D-03	0.0	-0.7219D-02	0.0	0.0
2	8.5505	0.0	-1.5077	0.0	-0.2601D-02	0.0	0.8896D-02	-0.1447D-01	0.4304D 02
3	16.0697	0.0	-5.8489	0.0	0.9627D-02	0.0	-0.2095D-01	-0.2354D-01	-0.1030D+03
4	0.0	12.5000	0.0	0.0	-0.1322D-02	-0.6775D-01	-0.8632D-02	0.0	0.0
5	8.5505	12.5000	-1.5077	0.2684D-01	-0.8712D-03	0.5558D-01	0.6539D-02	-0.1930D-01	0.1330D 03
6	16.0697	12.5000	-5.8489	0.1669D+00	0.4807D-02	0.2940D+00	0.1389D-02	-0.2209D-01	-0.4782D-02
7	0.0	25.0000	0.0	0.0	0.0	-0.6877D-01	0.0	0.0	0.0
8	8.5505	25.0000	-1.5077	0.2971D-01	0.0	0.7180D-01	0.0	-0.4394D-01	0.0
9	16.0697	25.0000	-5.8489	0.1881D+00	0.0	0.3367D+00	0.0	-0.6264D-01	0.0

MACRO INSTRUCTION 4 EXECUTED STRE V1 = .0 , V2 = .0

FEAP ** CYLINDRICAL SHELL TEST ** 00000080

BIPARABOLIC SHELL ELEMENT/

ELM	NODE	1-COORD	2-COORD	3-COORD		1-DIRECTION	2-DIRECTION	12-SHEAR	13-SHEAR	23-SHEAR
1	1	13.018	5.283	-3.668						
					+1-STRESSES	0.70031D+02	-0.16594D+02	0.76601D+02	-0.36730D+00	-0.15123D+02
					0-STRESSES	-0.24547D+02	0.22841D+01	0.20805D+02	0.95399D+00	0.62998D-01
					-1-STRESSES	-0.11913D+03	0.21162D+02	-0.34992D+02	0.0	0.0
	DIRECTION COSINES				EPS-STRAINS	-0.54823D-04	0.52872D-05	0.96318D-04	0.52999D-05	0.34999D-06
	0.0	-0.85	0.52		CHI-STRAINS	0.17514D-02	-0.34959D-03	0.20665D-02		
	1.00	0.0	0.0		RESULTANTS	-0.61369D+01	0.57102D+00	0.52012D+01	0.23850D+00	0.15750D-01
	0.0	0.52	0.85		COUPLES	0.98519D+00	-0.19665D+00	0.58121D+00		
1	2	13.018	19.717	-3.668						
					+1-STRESSES	-0.37916D+02	-0.45066D+02	0.62375D+02	-0.17493D+01	-0.34875D+02
					0-STRESSES	-0.63774D+02	0.37152D+01	0.65785D+01	-0.42801D+00	-0.94442D+00
					-1-STRESSES	-0.89632D+02	0.52497D+02	-0.49218D+02	0.0	0.0
	DIRECTION COSINES				EPS-STRAINS	-0.14763D-03	0.86000D-05	0.30456D-04	-0.23778D-05	-0.52467D-05
	0.0	-0.85	0.52		CHI-STRAINS	0.47886D-03	-0.90336D-03	0.20665D-02		
	1.00	0.0	0.0		RESULTANTS	-0.15944D+02	0.92880D+00	0.16446D+01	-0.10700D+00	-0.23610D+00
	0.0	0.52	0.85		COUPLES	0.26936D+00	-0.50814D+00	0.58121D+00		
1	3	3.740	5.283	-0.292						
					+1-STRESSES	0.42003D+01	-0.82596D+02	0.76601D+02	0.75835D+00	-0.65998D+01
					0-STRESSES	0.24547D+02	0.87954D+01	0.20805D+02	-0.56295D+00	-0.51944D-01
					-1-STRESSES	0.44895D+02	0.10019D+03	-0.34992D+02	0.0	0.0
	DIRECTION COSINES				EPS-STRAINS	0.54823D-04	0.20360D-04	0.96318D-04	-0.31275D-05	-0.28858D-06
	0.0	-0.99	0.15		CHI-STRAINS	-0.37680D-03	-0.16924D-02	0.20665D-02		
	1.00	0.0	0.0		RESULTANTS	0.61369D+01	0.21989D+01	0.52012D+01	-0.14074D+00	-0.12986D-01
	0.0	0.15	0.99		COUPLES	-0.21195D+00	-0.95199D+00	0.58121D+00		
1	4	3.740	19.717	-0.292						
					+1-STRESSES	0.63823D+02	-0.18658D+03	0.62375D+02	0.10794D+01	-0.15386D+02
					0-STRESSES	0.63774D+02	0.14693D+02	0.65785D+01	-0.24192D+00	-0.10447D+01
					-1-STRESSES	0.63725D+02	0.21596D+03	-0.49218D+02	0.0	0.0
	DIRECTION COSINES				EPS-STRAINS	0.14763D-03	0.34012D-04	0.30456D-04	-0.13440D-05	-0.58037D-05
	0.0	-0.99	0.15		CHI-STRAINS	0.90841D-06	-0.37272D-02	0.20665D-02		
	1.00	0.0	0.0		RESULTANTS	0.15944D+02	0.36733D+01	0.16446D+01	-0.60480D-01	-0.26117D+00
	0.0	0.15	0.99		COUPLES	0.51098D-03	-0.20966D+01	0.58121D+00		

BIPARABOLIC SHELL ELEMENT/
EXTRAPOLATED NODAL VALUES/

ELM	NODE		1-DIRECTION	2-DIRECTION	12-SHEAR	13-SHEAR	23-SHEAR
1	1	RESULTANTS	-0.44122D+01	-0.61681D-02	0.65029D+01	0.56081D+00	0.11894D+00
		COUPLES	0.16058D+01	0.82230D-01	0.58121D+00		
1	2	RESULTANTS	-0.33832D+02	-0.94358D-01	0.34285D+00	-0.30753D+00	-0.31961D+00
		COUPLES	-0.18620D-01	0.70869D-01	0.58121D+00		
1	3	RESULTANTS	0.44122D+01	0.21054D+01	0.65029D+01	-0.36596D+00	0.66845D-01
		COUPLES	-0.85227D+00	-0.69791D+00	0.58121D+00		
1	4	RESULTANTS	0.33832D+02	0.53672D+01	0.34285D+00	0.42966D-01	-0.36069D+00
		COUPLES	0.16424D+00	-0.32085D+01	0.58121D+00		