A SHEAR FLEXIBLE FACET SHELL ELEMENT FOR LARGE DEFLECTION AND INSTABILITY ANALYSIS

A MASTER'S THESIS

In

Engineering Sciences

Middle East Technical University

by

ATILA BARUT

September, 1990

Approval of Graduate School of Natural and Applied Sciences

Prof. Dr. Alpay ANKARA
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science in Engineering Sciences.

Prof.Dr. Nuri AKKAŞ
Chairman of the Department

We certify that we have read this thesis and in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science in Engineering Sciences.

Asst.Prof.Dr. Süha ORAL
Supervisor

Examining Commitee in Charge

Prof.Dr. Nuri AKKAŞ (Chairman) . . . . . .
Assoc.Prof.Dr. Mehmet UTKU (C.E) . . . . . .
Assoc.Prof.Dr. Turgut TOKDEMİR (G.U) . . . . . .
Assoc.Prof.Dr. Haluk AKSEL (M.E) . . . . . .
Asst.Prof.Dr. Süha ORAL (M.E) . . . . . .

To my family

ABSTRACT

A SHEAR FLEXIBLE FACET SHELL ELEMENT FOR LARGE DEFLECTION AND INSTABILITY ANALYSIS

BARUT, Atila

M.Sc. Thesis in Engineering Sciences

Supervisor : Asst. Prof. Dr. Süha Oral

September 1990, 137 pages

A facet shell element based on the superposition of a C^0 type anisoparametric plate bending element and a quadratic plane stress element with vertex rotations is formulated for geometrically nonlinear and instability analysis of plates and shells. The incremental equilibrium equations are derived using the updated Lagrangian formulation which proves to be effective for three-node elements.

The restriction of small rotation between succesive increments is removed by using the Hsiao's method which eliminates the large rigid body motions from the total displacements.

A displacement incrementation strategy is used to alleviate the singularity of the tangential stiffness matrix in the limit point type problems. The solution algorithms are based on the Newton-Raphson incremental-iterative method.

Several numerical examples for beams, plates and shells under conservative or non-conservative loadings are solved to assess the performance of the element.

Key words : updated Lagrangian formulation, finite rotation, displacement control, shells, plates, nonlinear analysis, finite element.

Science Code: Engineering Sciences (620)
620.01.00. Applied Mechanics 620.01.01.
Mechanics Strength.

ÖZET

BÜYÜK YER DEĞİŞTİRME VE KARARSIZLIK ANALİZLERİ İÇİN KAYMA ŞEKİL DEĞİŞTİRMELİ ÜÇ BAĞLANTI NOKTALI BİR DÜZLEM SONLU ELEMAN

BARUT, Atila

Master Tezi, Mühendislik Bilimleri Anabilim Dalı

Tez Yöneticisi: Yard. Doç. Dr. Süha Oral

Eylül 1990, 137 sayfa

Bu çalışmada, plak ve kabukların büyük yer değiştirme ve kararsızlık analizleri için, C^0 devamlılık tipinde olan, anizoparametrik bir plak elemanı ile ikinci dereceden bir düzlem-gerilme elemanının birleşmesinden meydana gelen üç bağlantı noktalı bir düzlem kabuk elemanı sunulacaktır. Denge durumlarının tespitinde koordinat düzeltmeli Lagrange algoritması kullanılmıştır.

Yük artırımı sırasında meydana gelen dönmelerin küçük kalması zorunluluğu, Hsiao'nun sonlu dönmeler metodu ile ortadan kaldırılmıştır.

Rijitlik matrisinin tekil davranış gösterdiği bazı kararsızlık problemlerinde yer değiştirme artırımlı algoritma kullanılmıştır. Çözüm algoritmaları Newton-Raphson tekniğine dayalıdır.

Çözüm problemlerinde yönü sabit veya değişebilen yükler göz önüne alınmış, cözümler eleman performansının yüksek olduğunu göstermiştir.

Anahtar kelimeler : Koordinat düzeltmeli Lagrange algoritması, sonlu dönme, yer değiştirme kontrol, kabuklar, plaklar, nonlineer analiz, sonlu eleman.

Bilim Kodu: Mühendislik Bilimleri (620)
620.01.00 Uygulamalı Mekanik 620.01.00
Mekanik Mukavemet

ACKNOWLEDGEMENTS

I would like to express my deepest thanks to Asst. Prof. Dr. Süha Oral who inspired and encouraged me to study in the field of Finite Elements. His initial suggestion of this topic and his constant ideas and suggestions were invaluable.

I also wish to express my gratitude to my colleague, Mr. Levent Ileri, and my brother, Fatih Barut, for their help in the preparation of this manuscript.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 General

A large number of structural systems in civil, mechanical and aerospace engineering is composed of plate and shell components. The reliable and economical design of these structures requires an investigation of their nonlinear behaviour. Such a structure may undergo large deflections that alter its shape so that geometric nonlinearity has to be considered. It is also possible that the structure may admit the possibility of buckling due to its doubly-curved geometry. Another possibility is that the stresses due to membrane action, usually neglected in plate flexure, may cause a considerable decrease of displacements as compared with the linear solution even though the displacements are quite small. In this case, a design using nonlinear analysis will be more economical than the linear analysis. Therefore, the nonlinear finite element analysis has gained increased importance in recent years. This is partly due to the development of high-speed computers.

The evaluation of nonlinear response of shell structures necessitates the use of efficient finite element models. The natural choice for such models seems to be doubly curved shell elements [1-8]. However, the satisfaction of interelement continuity conditions and proper representation of rigid body and constant strain modes in doubly-curved elements can be achieved by higher order field

assumptions. Moreover, the element stiffness matrix is relatively large in size and numerical integration over the thickness increases the computational time. Hence, the use of these sophisticated elements presents practical difficulties in the non-linear analysis of large engineering structures where the cost effectiveness becomes a prime factor.

A simple but efficient alternative is the facet shell element which proves to be accurate enough when used with the updated Lagrangian formulation [12-16]. Although a more refined mesh becomes necessary compared with the doubly-curved elements, the overall cost of the analysis may still be less. Moreover, the modelling is considerably simplified due to the simple nodal configurations of facet elements.

In nonlinear finite element analysis involving large displacements and instability, it is necessary to resort to an incremental formulation for the equations of motion. There are basically two approaches for the description of motion. In the first approach the initial configuration of the structure is selected as reference when describing the motion, stresses, strains etc. The formulation based on this approach is called the total lagrangian formulation. In the second approach, the current configuration is referred to in the description of motion. Since the reference configuration is continuously updated during deformation, this approach is called the updated Lagrangian or moving coordinate formulation.

Theoretically, there is no difference in the results obtained by using any of

the methods. However, the updated Lagrangian approach requires less computational time in the calculation of element stiffness matrices if simple type elements are used [15]. Moreover, the formulation can be extended to deal with non-conservative loading . This can be achieved by treating nonconservative loads as piecewise conservative loads whose directions are continuously updated together with the reference configuration [14].

In most of the element formulations the displacement field is linearized with respect to nodal rotations and the nodal rotations are considered to be vector quantities. However, it is a known fact that finite rotations (large rotations) are not true vector quantities, i.e., they do not comply with the rules of vector operations [14]. In such situations the formulations based on linearized element approximation require small increments, and thus the nodal rotations are restricted in magnitude.

In developing solution algorithms for nonlinear shell problems, one has to take into account all factors governing the complex behaviour of such structures. Sudden instabilities may occur accompanied by singular and indefinite incremental stiffness matrices. In this case, the load incrementation strategy fails and it becomes necessary to change the solution strategy to proceed beyond the limit point.

1.2 Survey of Past Research

One of the earliest attempts to develop finite element models for geometrically nonlinear analysis is reported in [9]. In this study, the well known constant strain membrane element is employed with the use of Green Lagrangian strain tensor. This model, however, does not include plate bending behaviour. The first developments of plate bending elements for nonlinear analysis are presented in [1,2]. These models are based on the Von Karman nonlinear plate assumptions [19] and they are formulated using the Kirchhoff plate theory in which C^1 continuity is imposed on the displacement field. The nonlinear finite element analysis of such models using Mindlin plate theory which includes shear deformations has been reported in [3]. Different types of higher order isoparametric elements have been compared in this study. However, the applicability of such elements based on Von Karman assumptions are limited to moderate rotations only.

Chang and Sawamiphakdi [4] developed a 9-node Lagrange shell element for large deflection, and post-buckling analysis . The formulation is based on updated Lagrangian description of motion. The degenerated isoparametric concept [10] has been used in the derivation of the element displacement field. The formulation of this element has also been discussed by Hsiao and Chen [5] using finite rotation vectors to define the orientation of the surface normal of the element. The use of degenerated concept, however, requires much computational effort to form the element matrices and the numerical integration over element

thickness increases the computational time as stated in [16].

Fafard, Dhatt and Batoz [6] presented a six node triangular shell element for geometrical and material nonlinear analysis. This element is obtained by superposing a discrete Kirchhoff model for bending and a linear strain triangular element for membrane. The updated Lagrangian formulation has been adopted to study the oriented structures undergoing large displacements and large rotations. It has been pointed out that the choice of transverse displacement field is very important to obtain an efficient nonlinear element. After a number of numerical examples, it has been concluded that the best choice for such models is to assume linear interpolations for transverse displacements.

There are also quadrilateral models available in the literature. These models differ in their shape, nodal variables and variational methods used. In [7] a 48 d.o.f. quadrilateral shell finite element including effect of geometric nonlinearity has been developed. The displacement functions are assumed to consist of bicubic Hermitian polynomials defined by curvilinear coordinates. The element is free from membrane locking; however, the introduction of second derivative displacement as a nodal degree of freedom may present difficulties when imposing boundary conditions.

In [8] a quadrilateral C^0 model based on the Hellinger - Reissner principle has been developed. The displacement, strain and stress fields are assumed independently. It has been observed that the element requires a careful selection of polynomial functions for the assumed strain field to avoid shear and membrane

locking and the results in some of the problems are diverging.

The most popular models in nonlinear finite element analysis of shell structures have been the simple flat triangular elements due to their computational advantage and low cost of stiffness matrix formulations. Various triangular elements have been proposed, each of which contains different interpolation models.

One of the early attempts on efficient nonlinear triangular models has been reported in [12]. In this study a hybrid stress flat triangular element with 15 d.o.f. has been developed for analyzing the large deflection and static behaviour of shells. Two models called as consistent (Total Lagrangian) and inconsistent (Updated Lagrangian) have been compared using this element. The inplane displacements along boundaries are interpolated linearly, while the transverse displacements cubically. In-plane and bending stress fields are assumed in terms of constant and linear functions, respectively. It was reported that much less computational effort was required using inconsistent model.

Argyris et al. [13] developed a facet triangular shell element called TRUMP which is applicable to thin or thick shells with shear deformations. The element is based on physical lumping ideas described in [17]. In this study various possible cases for the elastic stiffness of the element including thickness variation were considered and their associated matrices were derived. The in-plane rotational stiffness matrix of the element was established by introducing a triangular model which consists of rotational springs at its nodes and rigid sliding bars along

its vertices. Thus the singularity of the global stiffness matrix due to the coplanar assembly of elements of a common node was avoided using the above model. Various numerical examples showed that although the element membrane behaviour was improved by in-plane rotational stiffness, the results were not accurate enough for problems where the main load is carried by varying membrane stress.

Horrigmoe and Bergan [14] formulated a facet triangular shell element for large deflection and instability analysis of free form shells. The element is of C^1 type and derived by utilizing the free formulation concept [18]. In this study a base coordinate system which is fixed to the shell surface has been introduced to remove the in-plane rotational d.o.f. The element was tested in several numerical examples and it was observed that the performance of the element is superior for thin plate regimes. However, less accuracy is expected for thick plates since the model does not include shear deformations.

Recently, Bathe and Ho [15] developed a three node, shear flexible, triangular finite element for geometrically nonlinear analysis of general shells. The element displacement field is defined by different order of polynomials. In this model, the in-plane displacements are linear, the normal rotations are in quadratic form, and thus the transverse displacements are in cubic form which is one degree higher than the rotations. Although the element is simple and accurate it is restricted to small rotations between load increments and the use of higher order interpolations increases the computational time to

form the required matrices. The element performs satisfactorily with the use of finite rotation strategies which alleviates the small rotation restriction between load increments [16].

It is a well known fact that finite rotation components are not vectors and that the sum of finite rotations depends on the order of rotations[14]. In geometric nonlinear analysis this has led to the adoption of many different type of rotation variables and different rotation strategies to accomodate large rotation capability during the large deformation process.

The finite rotation problem was first discussed in detail by Wempner [20] in 1969 and a tensorial formulation for large deflection analysis has been developed in this study. It has been pointed out that the motion of a body is decomposed into a large rigid-body motion and a small deformation. So if the rigid-body part is eliminated then the infinitesimal approximations can be employed for the deformational part of the motion. This study can be considered as a good tool serving further studies.

Argyris [21] has also discussed this problem and developed a novel matrix formulation for finite rotations in space. In this study a rotation vector $\vec{\theta}$ referred to as pseudo vector has been introduced to define a finite rotation about an axis whose unit normal is $\tilde{n}$. Hence, a vector can be rotated to a new position with the use of this rotation vector.

Using the rotation vector described in [21], several rotation strategies have been proposed to determine the orientation of the shell normal for degenerated

isoparametric elements [22-24]. In each method, different type of rotation axis has been chosen and the rotation of the shell normal has been determined incrementally by the application of rotation vector associated with this axis.

Recently, Hsiao [11] developed a finite rotation method to remove the restriction of small rotations between succesive load increments for triangular shell elements. In this study two approaches termed as direct and incremental methods have been presented and compared. It has been observed that both methods yield identical results. However, the in-plane rotations are not included in the determination of element deformational rotations. In many cases this is appropriate because most of the triangular flat shell elements neglect the in-plane bending deformations. In a later work, Hsiao [25] also developed a corotational finite rotation strategy for straight beam elements.

The development of numerical techniques for the solution of nonlinear structural problems in recent years has been studied extensively. Early methods for solving incremental equilibrium equations were the classical load incrementation techniques which are well suited for the prediction of nonlinear response prior to the limit point. However, when these techniques are utilized for predicting the response beyond the limit point, they yield unsatisfactory results due to the singularity of the tangential stiffness matrix at the limit point in the load deflection curve. In order to overcome this difficulty, several different numerical techniques have been developed. Bergan et al [27], and Bergan [28], utilizing the current stress parameter, suppress the equilibrium iterations in

the critical zone until the limit point has been traversed. Alternatively Wright and Gaylord [29] developed the technique of adding fictitious springs to the incremental stiffness matrix to enable it to remain positive definite in a snap-through problem. The main disadvantage of this method is that trial and error is required in the selection of appropriate springs.

Another commonly used strategy has been to increment a characteristic displacement component by a prescribed amount and evaluate the corresponding load level. This technique was first described by Argyris [30]. However, this approach does not preserve the symmetry of the incremental stiffness matrix. The nonsymmetric matrix restriction was overcome by Pian and Tong [31], but their method did not perform iterations to equilibrium convergence. A scheme which preserved the symmetry and allowed for equilibrium iterations was developed by Haisler et al [32]. In this study an unknown load factor which controls the external load vector has been introduced. The incremental equilibrium equations are rearranged in a partitioned form to keep the unknowns on one side of the equation. The partitioned equations are then expanded to obtain two sets of symmetric equations to be solved individually for the load factor and the unspecified displacement components. Thus, the corresponding load level can be easily calculated by multiplying the load factor with the external load vector. This method yields accurate results.

1.3 Objective and Outline of the Present Study

The objective of the present study is to develop a simple and efficient nonlinear finite element model for large deflection and instability analysis of plates and shells. Particular emphasis is given here to the development of effective solution procedures to deal with large incremental rotations and snap-through.

The starting point for the present model is the MIN3 plate bending and Allman's plane-stress elements which have already been employed in linear analysis [38,39]. MIN3 [38] is an efficient Mindlin element in which only C^0 continuity requirement is imposed on the kinematic field. The performance of the element is improved by an in-plane shear correction factor. The element is of correct rank, does not lock in the thin plate limit and rapidly converges. Due to its relability, economy and good stress recovery, MIN3 has been considered to constitute the bending part of the present facet element. As for the membrane part, the plane stress element developed by Allman [39] has been used. In this element, the in-plane bending response is improved by introducing in-plane rotational d.o.f. at the vertex nodes only. The introduction of the rotational d.o.f does not increase the cost of the element since the transformation yields six d.o.f. per node. One disadvantage of this element is the existence of spurious zero energy mode, however this is easily suppressed by prescribing an arbitrary value for any one of in-plane rotations in the entire mesh.

An updated Lagrangian approach given by [35] has been used to derive the

governing incremental equilibrium equations for the present shell model. In this derivation the principle of virtual displacements, which is valid for nonlinear analysis, is used .

The restriction of small rotations between load increments during deformation process has been removed by using the motion process proposed by Hsiao [16] for flat triangular elements. Via the use of this motion process, the element rigid-body rotations are subtracted from the the nodal rotation increments of the individual elements. Hence, only the deformational part of the rotations are considered in the calculation of the geometric stiffness matrix and element internal load vector.

The numerical algorithms used here are incremental iterative procedures based on the Newton-Raphson method. The difficulty of singular tangential stiffness matrix encountered in stability problems of the limit point type has been circumvented by adopting a displacement incrementation strategy described in [32].

The validity of the element has been assessed by large deflection and instability problems of wide beams, plates and shells under conservative and nonconservative loads. The results indicate that the present element is reliable and accurate.

CHAPTER 2

VARIATIONAL FORMULATION

2.1 General

An incremental updated Lagrangian formulation [35] for the equations of motion of a body undergoing large deflections is derived using the principles of virtual displacements. The most general approach to incremental Lagrangian formulation is to consider the nonlinear kinematic relations within a linear increment. Hence, all higher order incremental terms in displacements and in strains can be discarded. This leads to the well known expression for a single geometric stiffness matrix.

Different stress and strain measures referred to the updated configuration are given with their incremental decompositions. Then the generalized stress and strain components of a plate based on the Mindlin plate theory are defined.

2.2 Description of Motion

Consider the motion of a body in a Cartesian coordinate system as shown in Figure 2.1. The objective is to evaluate the equilibrium position of the body in the deformed configuration at time t_2.

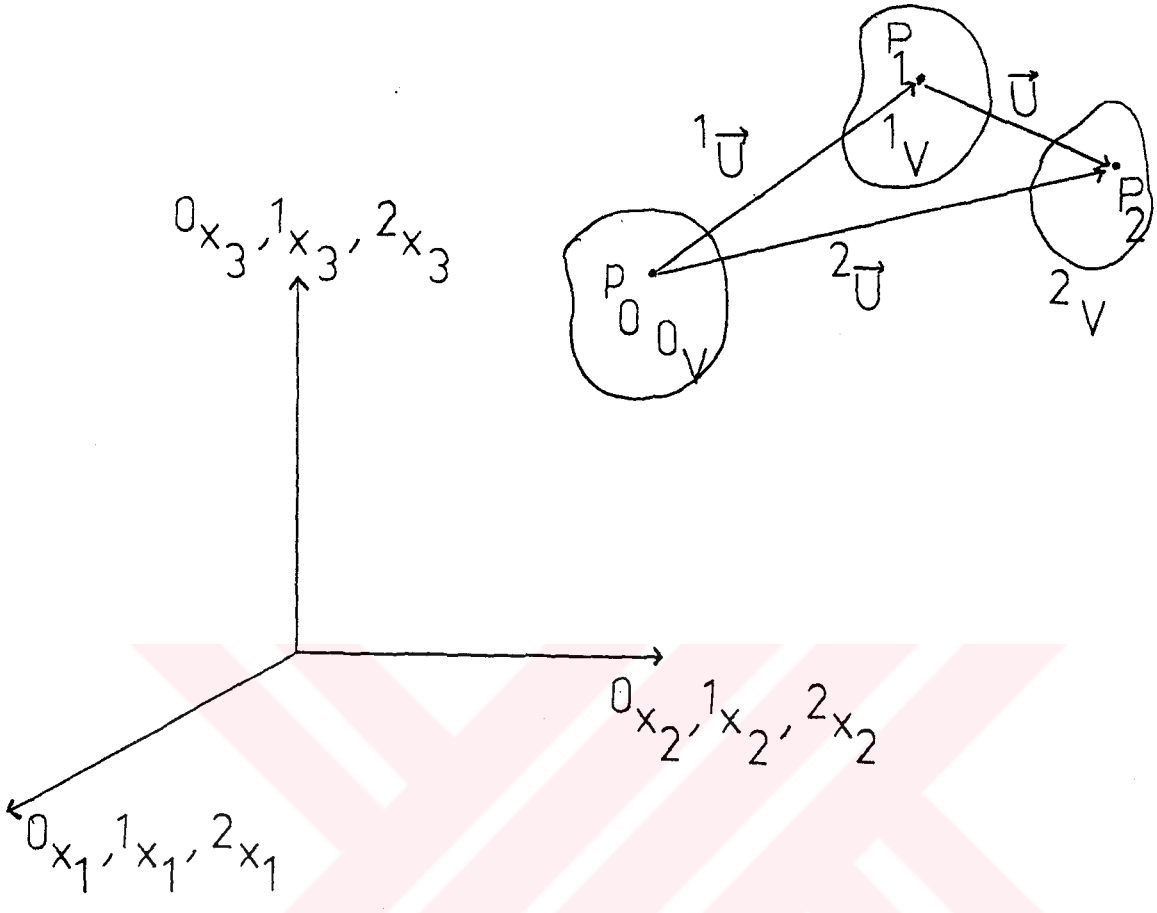


Figure 2.1. Motion of a Body in a Cartesian Coordinate System

In Figure 2.1, ${}^0x_1, {}^0x_2$ and ${}^0x_3$ are the coordinates of the initial configuration, ${}^1x_1, {}^1x_2$ and ${}^1x_3$ are the coordinates of the current or updated configuration and the coordinates describing the deformed configuration of the body are denoted by ${}^2x_1, {}^2x_2$ and ${}^2x_3$ where the left superscript of each coordinate is replaced by the number 2. The notation for the displacements of the body is similar to that for the coordinates. ${}^0V, {}^1V$ and ${}^2V$ are volumes at different configurations. The relation between the deformed and current configurations is given as follows.

$$^2x_i = {}^1x_i + u_i \qquad (i = 1,2,3) \tag{2.1}$$

where u_i are the unknown increments measured from time t_1 to t_2 and defined as,

$$u_i = {}^2u_i - {}^1u_i \tag{2.2}$$

2.3 Variational Formulation

In order to solve the state variables defined in the deformed configuration, the principle of virtual displacements is employed;

$$\int_{^2V} {}^2\tau_{ij}\ \delta_2 e_{ij} d^2V = \delta^2 W \tag{2.3}$$

where $^2\tau_{ij}$ are the components of the true Cauchy stress tensor measured in the deformed configuration , $\delta^2 W$ is the work done by external forces that the body is subjected and finally $\delta_2 e_{ij}$ are the infinitesimal virtual strains defined as,

$$_2e_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial {}^2x_j} + \frac{\partial u_j}{\partial {}^2x_i}\right). \tag{2.4}$$

(2.3) can not be solved directly since the configuration at time $t_2 (= t_1 + \Delta t)$ is unknown. A solution can be obtained by referring all variables in the deformed configuration to a previously calculated equilibrium configuration. Hence, in

the updated Lagrangian formulation, the variables in (2.3) will be referred to the current configuration denoted by $t = t_1$. The stress and strain measures used in a Lagrangian formulation are the 2^{nd} Piola Kirchhoff stresses and Green Lagrangian strain tensors, respectively. They are not true quantities, but related to the true Cauchy stress and variation of infinitesimal strain tensors through the following transformations.

$$ {}^{2}\tau_{ij} = \frac{{}^{2}\rho}{{}_{1}\rho} \frac{\partial^2 x_i}{\partial^1 x_m} {}^{2}_{1}S_{mn} \frac{\partial^2 x_j}{\partial^1 x_n} \tag{2.5.a} $$

$$ \delta_2 e_{ij} = \frac{\partial^1 x_m}{\partial^2 x_i} \frac{\partial^1 x_n}{\partial^2 x_j} {}^{2}_{1}\epsilon_{mn} \tag{2.5.b} $$

where $\frac{{}^{2}\rho}{{}_{1}\rho}$ represents the ratio of the mass densities at time t_1 and t_2, ${}^{2}_{1}S_{mn}$ and ${}^{2}_{1}\epsilon_{mn}$ are the Cartesian components of the 2^{nd} Piola Kirchhoff stress and Green Lagrangian strain tensors corresponding to time t_2, but referred to the configuration at t_1.

We also have,

$$ \frac{{}^{2}\rho}{{}_{1}\rho} = det({}^{1}_{2}J) \tag{2.6} $$

and

$$\frac{^2V}{^1V} = det(^2_1J) = \frac{1}{det(^1_2J)} \tag{2.7}$$

where ^2_1J is the deformation gradient matrix given by,

$$^2_1J = \begin{bmatrix} \frac{\partial^2 x_1}{\partial^1 x_1} & \frac{\partial^2 x_1}{\partial^1 x_2} & \frac{\partial^2 x_1}{\partial^1 x_3} \\ \frac{\partial^2 x_2}{\partial^1 x_1} & \frac{\partial^2 x_2}{\partial^1 x_2} & \frac{\partial^2 x_2}{\partial^1 x_3} \\ \frac{\partial^2 x_3}{\partial^1 x_1} & \frac{\partial^2 x_3}{\partial^1 x_2} & \frac{\partial^2 x_3}{\partial^1 x_3} \end{bmatrix} \tag{2.8}$$

Substituting [(2.5)-(2.8)] into (2.3), one obtains the virtual work expression used in the updated Lagrangian formulation as,

$$\int_{^1V} {}^2_1S_{ij}\ \delta^2_1\epsilon_{ij} d^1V = \delta^2W \tag{2.9}$$

The stresses $^2_1S_{ij}$ may be defined by the following incremental decomposition [33-35].

$$^2_1S_{ij} = ^1_1 S_{ij} + S_{ij} \tag{2.10}$$

where $^1_1S_{ij}$ are the 2^{nd} Piola Kirchhoff stresses both measured and referred to the same configuration. It is clear from (2.5.a) that we can prove

$$^1_1S_{ij} = ^1 \tau_{ij} \tag{2.11}$$

in which ${}^1\tau_{ij}$ are the Cauchy stresses measured in the current configuration. The second term S_{ij} on the right hand side of (2.10) is the tensor representing the incremental form of the Kirchhoff stresses.The Cauchy stress tensor in (2.11) is calculated as,

$$ {}^1\tau_{ij} = C_{ijkl}\ {}^1E_{kl} \tag{2.12} $$

where

$$ {}^1E_{kl} = \frac{1}{2}\left(\frac{\partial^1 u_i}{\partial^1 x_j} + \frac{\partial^1 u_j}{\partial^1 x_i} - \frac{\partial^1 u_k}{\partial^1 x_i}\frac{\partial^1 u_k}{\partial^1 x_j}\right) \tag{2.13} $$

is the Almansi strain tensor at $t = t_1$.

In the updated Lagrangian formulation the Green-Lagrangian strain tensor takes an incremental form and is defined by its linear and nonlinear parts as,

$$ \begin{aligned} {}^2_1\epsilon_{ij} &= \epsilon_{ij} = e_{ij} + \eta_{ij} \\ e_{ij} &= \frac{1}{2}\left(\frac{\partial u_i}{\partial^1 x_j} + \frac{\partial u_j}{\partial^1 x_i}\right) \\ \eta_{ij} &= \frac{1}{2}\left(\frac{\partial u_k}{\partial^1 x_i}\frac{\partial u_k}{\partial^1 x_j}\right) \end{aligned} \tag{2.14} $$

where e_{ij} and η_{ij} are the incremental forms of the linear and nonlinear parts of the Green Lagrangian strain tensor,respectively.

Having determined the incremental tensors S_{ij} and ϵ_{ij}, they can be related to each other through the constituve relation as,

$$S_{ij} = C_{ijkl}\epsilon_{kl} \tag{2.15}$$

where C_{ijkl} are the components of the material property tensor.

The equations of motion with incremental decompositions are obtained upon substitution of (2.10),(2.11),(2.14) and (2.15) into (2.9) as,

$$\int_{^1V} C_{ijkl}\epsilon_{kl}\delta\epsilon_{ij}d^1V + \int_{^1V} {}^1\tau_{ij}\delta\eta_{ij}d^1V = \delta^2W - \int_{^1V} {}^1\tau_{ij}\delta e_{ij}d^1V \tag{2.16}$$

The solution of (2.16) can not be calculated directly since they are non-linear in displacement increments. However, approximate solutions can be obtained by assuming in (2.16) that $\epsilon_{kl} = e_{kl}$ and $\delta\epsilon_{ij} = \delta e_{ij}$. Therefore the incremental constituve relations become

$$S_{ij} = C_{ijkl}\ e_{kl} \tag{2.17}$$

Upon linearization we obtain the approximate equilibrium equations solved in the updated Lagrangian formulation as,

$$\int_{^1V} C_{ijkl} e_{kl} \delta e_{ij} d^1V + \int_{^1V} {}^1\tau_{ij} \delta \eta_{ij} d^1V = \delta^2 W - \int_{^1V} {}^1\tau_{ij} \delta e_{ij} d^1V \qquad (2.18)$$

2.4 Mindlin Theory

2.4.1 The Displacement Field

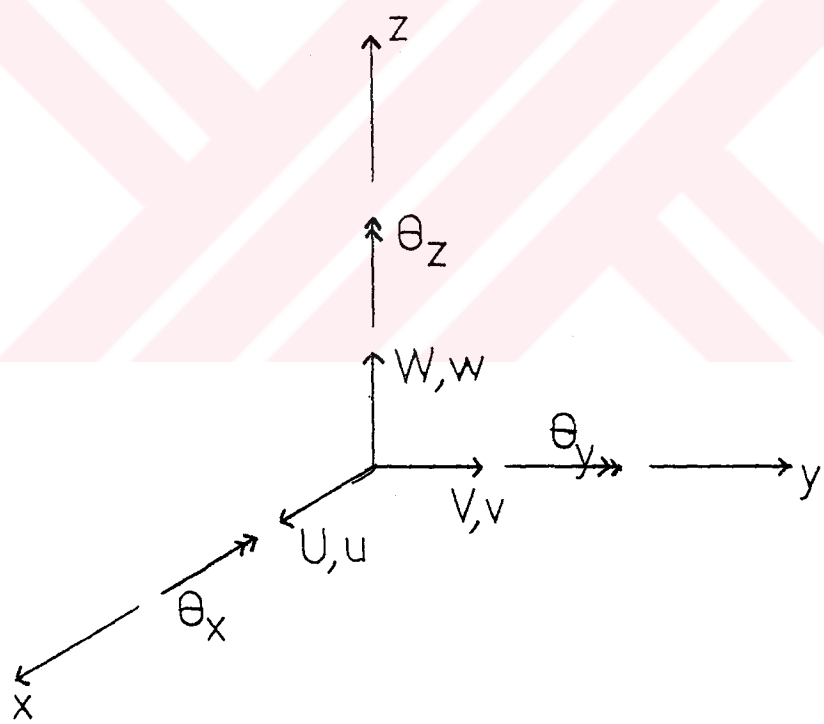


Figure 2.2. Sign Conventions for Displacements and Rotations

The displacement field based on Mindlin theory[36] is given as,

$$\begin{aligned} U(x,y,z) &= u(x,y) + z\theta_y(x,y) \\ V(x,y,z) &= v(x,y) + z\theta_x(x,y) \\ W(x,y,z) &= w(x,y) \end{aligned} \tag{2.19}$$

where U,V and W are the displacements in x,y and z directions, respectively, and they are defined in terms of mid-surface translations u,v,w and normal rotations θ_x and θ_y about x and y axes, respectively. The sign convention for the displacement field is as shown in Figure (2.2).

2.4.2 The Strain Field

The Green Lagrangian strains of a plate including transverse shear is given in as,

$$\begin{Bmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \\ \gamma_{xz} \\ \gamma_{yz} \end{Bmatrix} = \begin{Bmatrix} U_{,x} + \frac{1}{2}U_{,x}^2 + \frac{1}{2}V_{,x}^2 + \frac{1}{2}W_{,x}^2 \\ V_{,y} + \frac{1}{2}U_{,y}^2 + \frac{1}{2}V_{,y}^2 + \frac{1}{2}W_{,y}^2 \\ U_{,y} + V_{,x} + U_{,y}U_{,x} + V_{,y}V_{,x} + W_{,y}W_{,x} \\ U_{,z} + W_{,x} + U_{,z}U_{,x} + V_{,z}V_{,x} + W_{,z}W_{,x} \\ V_{,z} + W_{,y} + U_{,z}U_{,y} + V_{,z}V_{,y} + W_{,z}W_{,y} \end{Bmatrix} \tag{2.20}$$

where "," represents differentiation.

We can express the strain components in terms of mid-surface deformations in partitioned form as follows;

$$\begin{Bmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \\ \cdots \\ \gamma_{xz} \\ \gamma_{yz} \end{Bmatrix} = \begin{Bmatrix} \{\epsilon_m\} \\ \cdots \\ \{\epsilon_s\} \end{Bmatrix} + \begin{Bmatrix} z\{\epsilon_b\} \\ \cdots \\ \{0\} \end{Bmatrix} + \begin{Bmatrix} \{\eta_m\} \\ \cdots \\ \{0\} \end{Bmatrix} \tag{2.21}$$

where

$$\{\epsilon_m\} = \begin{Bmatrix} u_{,x} \\ v_{,y} \\ u_{,y} + v_{,x} \end{Bmatrix}, \tag{2.22.a}$$

are the in-plane strain components,

$$\{\epsilon_b\} = \begin{Bmatrix} \theta_{y,x} \\ \theta_{x,y} \\ \theta_{y,y} + \theta_{x,x} \end{Bmatrix}, \tag{2.22.b}$$

are the curvatures and twist, and

$$\{\epsilon_s\} = \begin{Bmatrix} \gamma_{xz} \\ \gamma_{yz} \end{Bmatrix}, \tag{2.22.c}$$

are the transverse shear strains. $\{\eta_m\}$ are the the nonlinear components of in-plane strains :

$$\{\eta_m\} = \begin{Bmatrix} \frac{1}{2}u_{,x}^2 + \frac{1}{2}v_{,x}^2 + \frac{1}{2}w_{,x}^2 \\ \frac{1}{2}u_{,y}^2 + \frac{1}{2}v_{,y}^2 + \frac{1}{2}w_{,y}^2 \\ u_{,x}u_{,y} + v_{,x}v_{,y} + w_{,x}w_{,y} \end{Bmatrix} \tag{2.22.d}$$

In (2.22) the nonlinear components of bending and shear strains are neglected. This is appropriate because the effect of bending and shear deformations on the geometric stiffness matrix is very small as compared with the effect of in-plane deformations [15]. Therefore it is justifiable to neglect these effects.The components defined in equations 2.22 form the generalized Green Lagrangian strain vector denoted by $\{\epsilon\}$ and expressed as follows.

$$\{\epsilon\} = \begin{Bmatrix} \epsilon_x^0 \\ \epsilon_y^0 \\ \gamma_{xy}^0 \\ \cdots \\ \kappa_x \\ \kappa_y \\ \kappa_{xy} \\ \cdots \\ \gamma_{xz} \\ \gamma_{yz} \end{Bmatrix} = \begin{Bmatrix} \{\epsilon_m\} \\ \{\epsilon_b\} \\ \{\epsilon_s\} \end{Bmatrix} + \begin{Bmatrix} \{\eta_m\} \\ \{0\} \\ \{0\} \end{Bmatrix} \tag{2.23}$$

Finally,

$$\{\epsilon\} = \{e\} + \{\eta\} \tag{2.24}$$

2.4.3 The Stress Field

The stress resultants are

$$\{\tau\}^T = \lfloor N_x \quad N_y \quad N_{xy} \quad M_x \quad M_y \quad M_{xy} \quad Q_x \quad Q_y \rfloor \tag{2.25}$$

where N_x, N_y and N_{xy} are the in-plane stress resultants, M_x, M_y and M_{xy} bending and twisting moments, Q_x and Q_y transverse shear forces.

The stress resultants are determined by integrating the stresses through the thickness h of the plate.

$$\begin{aligned}
(N_x, N_y, N_{xy}) &= \int_{-h/2}^{h/2} (\sigma_x, \sigma_y, \sigma_{xy}) dz \\
(M_x, M_y, M_{xy}) &= \int_{-h/2}^{h/2} (\sigma_x, \sigma_y, \sigma_{xy}) z dz \\
(Q_x, Q_y) &= \int_{-h/2}^{h/2} (\sigma_{xz}, \sigma_{yz}) dz
\end{aligned} \tag{2.26}$$

The positive convention for the stress resultants is shown in Figure 2.3.

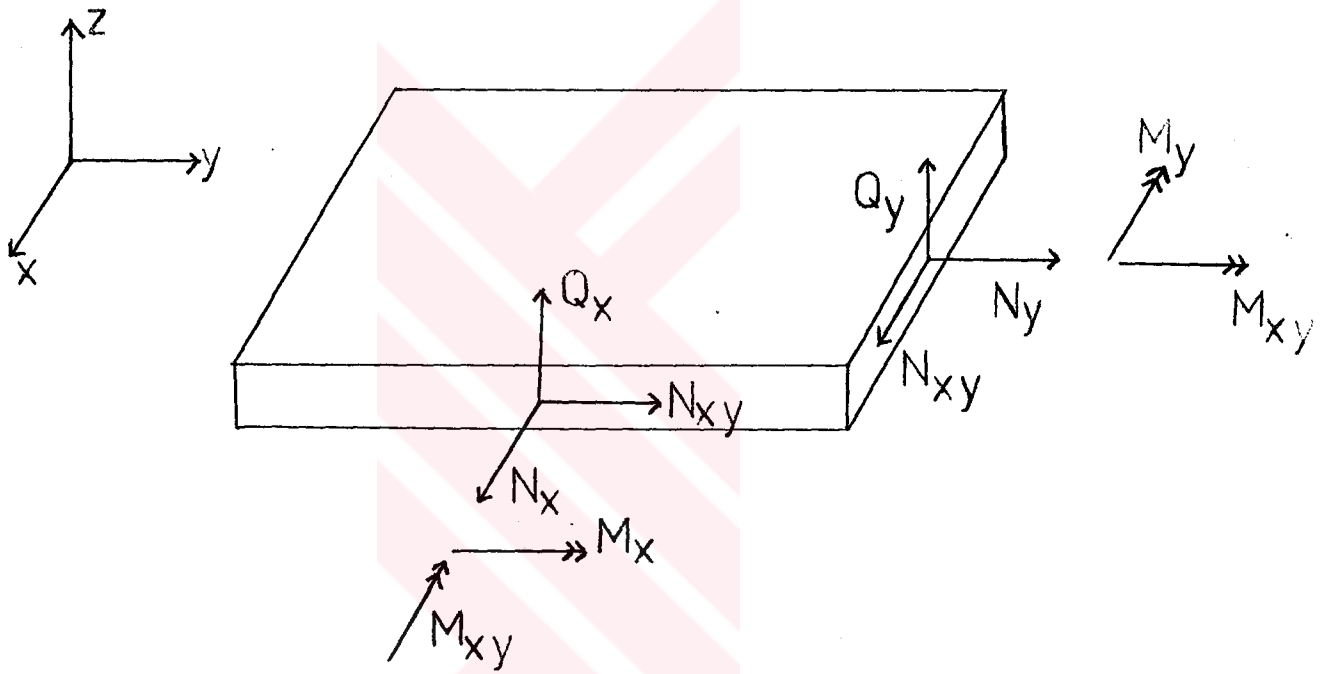


Figure 2.3. Positive Conventions for the Stress Resultants

CHAPTER 3

FINITE ELEMENT MODEL

3.1 General

A three node, shear flexible, flat shell element will be presented. This element is obtained by the superposition of a Mindlin plate element MIN3 for bending part [38], and a plane stress element including vertex rotations for membrane part[39]. The inclusion of vertex rotations improves the element inplane bending behaviour. The nodal degrees-of-freedom are three translations and three rotations. Therefore, there are totally 18 d.o.f for the present model. The displacement field will be expressed in terms of area coordinates. The locking phenomenon which is significant for shear flexible elements will be discussed.

The variational formulation given in Chapter 2 will be implemented to a facet shell element. In updated Lagrangian formulation there are two element matrices referred to as linear and geometric stiffness matrices as a result of the linearization of equations of motion. These matrices will be derived explicitly in the following sections.

3.2 Finite Element Implementation

In the following the finite element equilibrium equations of a shell element undergoing large deflection will be derived.

Consider a facet shell element under three different states as shown in Figure 3.1. Similar to the linearized form of the equations of motion given in (2.18), one can write the following matrix relationship for such an element.

$$\int_{^1A} \delta\{e\}^T [D]\{e\} d^1A + \int_{^1A} \delta\{\eta\}^T \{^1\tau_m\} d^1A = \delta\{q\}^T \{R\} - \int_{^1A} \delta\{e\}^T \{^1\tau\}\, d^1A \qquad (3.1)$$

where

$$\{q\}^T = \lfloor u_1 \quad u_2 \quad u_3 \quad \theta_{x1} \quad \theta_{y1} \quad \theta_{z1} \quad \ldots \quad \theta_{z3} \rfloor \qquad (3.2.a)$$

is the vector of incremental nodal displacements,

$$\{^1\tau\}^T = \lfloor {}^1N_x \quad {}^1N_y \quad {}^1N_{xy} \quad {}^1M_x \quad {}^1M_y \quad {}^1M_{xy} \quad {}^1Q_x \quad {}^1Q_y \rfloor \qquad (3.2.b)$$

is the Cauchy stress vector referred to the current configuration at $t = t_1$, [D] is the material property matrix, which is given in Appendix C, $\{R\}$ is the vector of externally applied forces in the absence of body forces, and finally

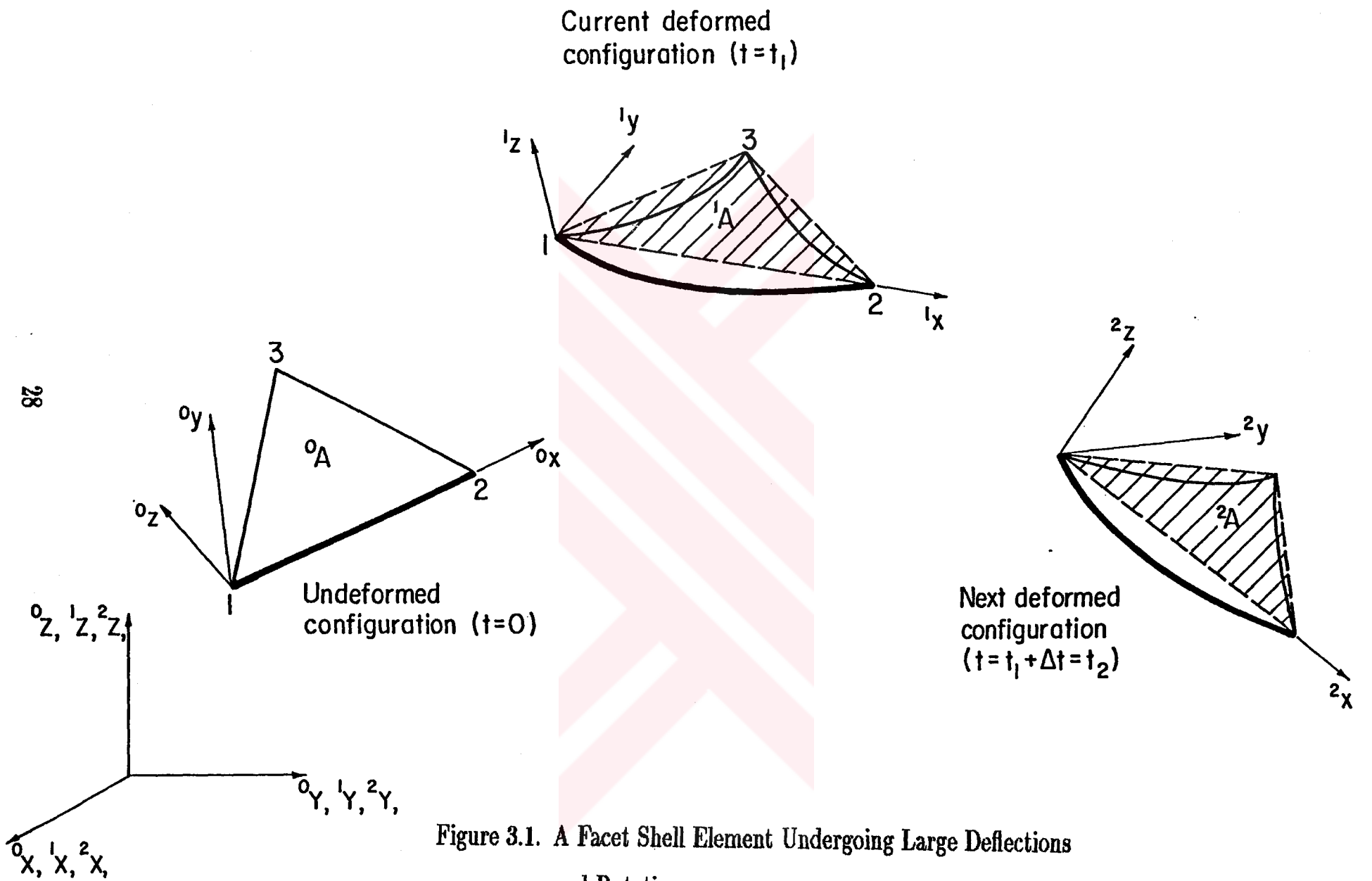


Figure 3.1. A Facet Shell Element Undergoing Large Deflections and Rotations

$$\{^1\tau_m\}^T = \lfloor {}^1N_x \quad {}^1N_y \quad {}^1N_{xy} \rfloor \tag{3.2.c}$$

- is the membrane part of $\{^1\tau\}$. This is due to the negligible effect of shear and bending deformations in the calculation of geometric stiffness matrix as described in section 2.2.2.

The first integral on the left hand side of (3.1) can be expressed as

$$\begin{aligned}\int_{^1A} \delta\{e\}^T[D]\{e\}d^1A &= \delta\{q\}^T\left(\int_{^1A}[N]^T[B]^T[D][B][N]d^1A\right)\{q\} \\ &= \delta\{q\}^T[K_L]\{q\}\end{aligned} \tag{3.3}$$

where [B] is the linear strain displacement matrix given in Appendix C, [N] is the shape matrix of the element model which will be defined later in this chapter, $[K_L]$ is the linear stiffness matrix.

The second integral on the left hand side of (3.1) can be evaluated as [37],

$$\begin{aligned}\int_{^1A} \delta\{\eta\}^T\{^1\tau\}d^1A &= \delta\{q\}^T\left(\int_{^1A}[N]^T[G]^T[^1S][G][N]d^1A\right)\{q\} \\ &= \delta\{q\}^T[K_G]\{q\}\end{aligned} \tag{3.4}$$

where [G] is the nonlinear strain displacement matrix given in Appendix C, $[K_G]$ is the geometric stiffness matrix. $[^1S]$ is a 6x6 matrix containing the membrane stresses as follows.

$$[{}^1S] = \begin{bmatrix} [{}^1S_x] & [{}^1S_{xy}] \\ [{}^1S_{xy}] & [{}^1S_y] \end{bmatrix} \tag{3.5}$$

where

$$\begin{aligned} [{}^1S_x] &= {}^1N_x[I_3] \\ [{}^1S_y] &= {}^1N_y[I_3] \\ [{}^1S_{xy}] &= {}^1N_{xy}[I_3] \end{aligned} \tag{3.6}$$

with $[I_3]$ being 3x3 identity matrix.

The Cauchy stresses $[{}^1\tau]$ can be calculated as,

$$[{}^1\tau] = [D]\{{}^1E\} \tag{3.7}$$

where

$$\{{}^1E\} = [B][N]\{q_d\} \tag{3.8}$$

is the vector of Almansi strains which can be defined linearly in terms of the deformational part of the nodal displacements represented by vector $\{q_d\}$. The deformational nodal displacements are obtained by subtracting the rigid-body motions from total displacements. Then the integral on the right hand side of (3.1) can be written as,

$$\int_{^1A} \delta\{e\}^T \{^1\tau\} d^1A = \delta\{q\}^T \left(\int_{^1A} [N]^T [B]^T [D][B][N] d^1A \right) \{q_d\}$$
$$= \delta\{q\}^T [K_L]\{q_d\} \tag{3.9}$$

Substituting (3.3),(3.4) and (3.9) into (3.1), one obtains

$$[K_T]\{q\} = \{F\} \tag{3.10}$$

where

$$[K_T] = [K_L] + [K_G] \tag{3.11}$$

is the tangential stiffness matrix, and

$$\{F\} = \{R\} - \{f\} \tag{3.12}$$

is the unbalanced load vector where

$$\{f\} = [K_L]\{q_d\} \tag{3.13}$$

3.3 Area Coordinate System for a Triangle

In this study, area coordinates are used as described in Figure 3.2 In this figure sides1, 2 and 3 are identified by the opposite vertices 1,2 and 3, respectively. The area coordinates ξ_i $(i = 1,2,3)$ for an interior point are defined as the ratios of the areas A_i to the total area A.

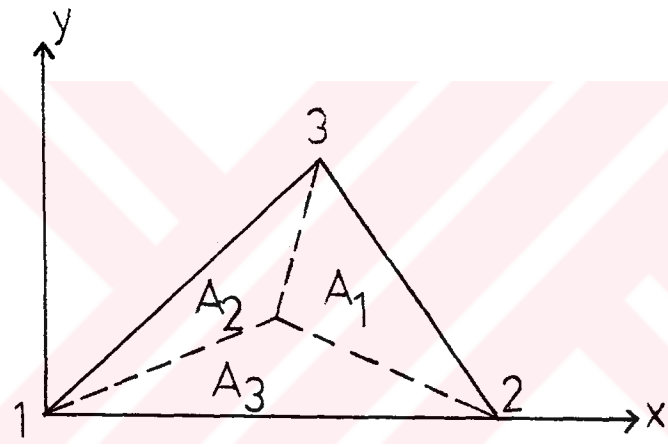


Figure 3.2. The Area Coordinates

$$\xi_1 = \frac{A_1}{A}, \quad \xi_2 = \frac{A_2}{A}, \quad \xi_3 = \frac{A_3}{A} \tag{3.14}$$

since

$$A_1 + A_2 + A_3 = A \tag{3.15a}$$

then

$$\xi_1 + \xi_2 + \xi_3 = 1 \tag{3.15b}$$

The local coordinates x and y of the element are related to the area coordinates in matrix form as,

$$\begin{Bmatrix} 1 \\ x \\ y \end{Bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{bmatrix} \begin{Bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{Bmatrix} \tag{3.16}$$

inverting (3.16), one obtains

$$\xi_i = \frac{1}{2A}(c_i + b_i x + a_i y) \quad (i = 1, 2, 3) \tag{3.17}$$

where

$$A = \frac{1}{2}\begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix} \tag{3.18}$$

and

$$c_i = x_j y_k - x_k y_j$$

$$b_i = y_j - y_k$$

$$a_i = x_k - x_j$$

$$(i = 1, 2, 3; j = 2, 3, 1; k = 3, 1, 2) \tag{3.19}$$

The integral of a polynomial over the area is given as,

$$\int_A \xi_1^\alpha \xi_2^\beta \xi_3^\gamma dA = \frac{\alpha!\beta!\gamma!}{(\alpha+\beta+\gamma+2)!} 2A \tag{3.20}$$

3.4 Membrane Part

A possible choice for a triangular model to constitute membrane part of the present shell element would be the well known constant strain triangle in which the displacement field is approximated by linear interpolations. However, this model yields inaccurate result for in-plane bending problems where the strain field varies linearly.

In order to improve the in-plane bending response, Allman[39] developed a three node plane stress element based on a compatible quadratic displacement field includinq vertex rotations as shown in Figure3.3.

The displacement field is approximated as follows.

$$u = \xi_i u_i + \frac{1}{2}(b_j \xi_k \xi_i - b_k \xi_i \xi_j)\theta_{zi}$$
$$v = \xi_i v_i + \frac{1}{2}(a_j \xi_k \xi_i - a_k \xi_i \xi_j)\theta_{zi}$$

$$(i = 1,2,3; j = 2,3,1; k = 3,1,2). \tag{3.21}$$

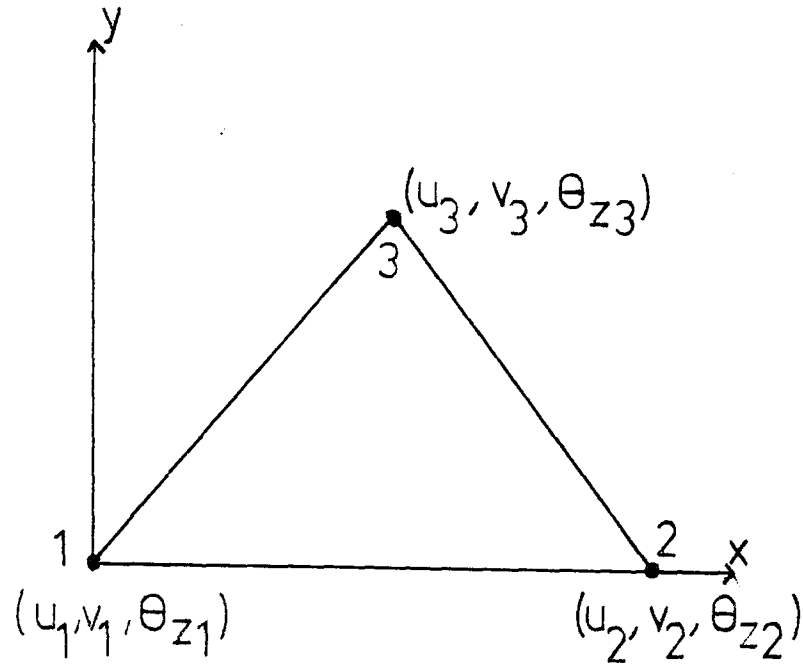


Figure 3.3. Nodal Variables for the Quadratic model

The inclusion of in-plane rotation θ_{zi} as a degree-of-freedom enhances the performance of the element since the quadratic displacement field provides linear variation in membrane strains ϵ_x, ϵ_y and γ_{xy}.

3.5 Bending Part

One major simplification associated with the utilization of Mindlin theory is that only C_0 continuity requirement is imposed upon interpolations of the kinematic field (i.e the derivatives of the deflection and rotation kinematic variables encountered in a variational theorem are no higher thanthe first order). This enables the Mindlin elements to be generated from low-order

interpolations. Regrettably, the use of standard isoparametric approach for low-order interpolations may yield stiffer solutions in displacements due to the lost of bending strain energy. This phenomenon is often called " locking ", and it is discussed in [38,40]. To alleviate this difficulty, Tessler and Hughes [38] proposed the anisoparametric approach in which different order of polynomials have been used for the kinematic field. Using this approach they developed a three-node triangular Mindlin plate element (MIN3) with 9 d.o.f. as shown in Figure 3.4.

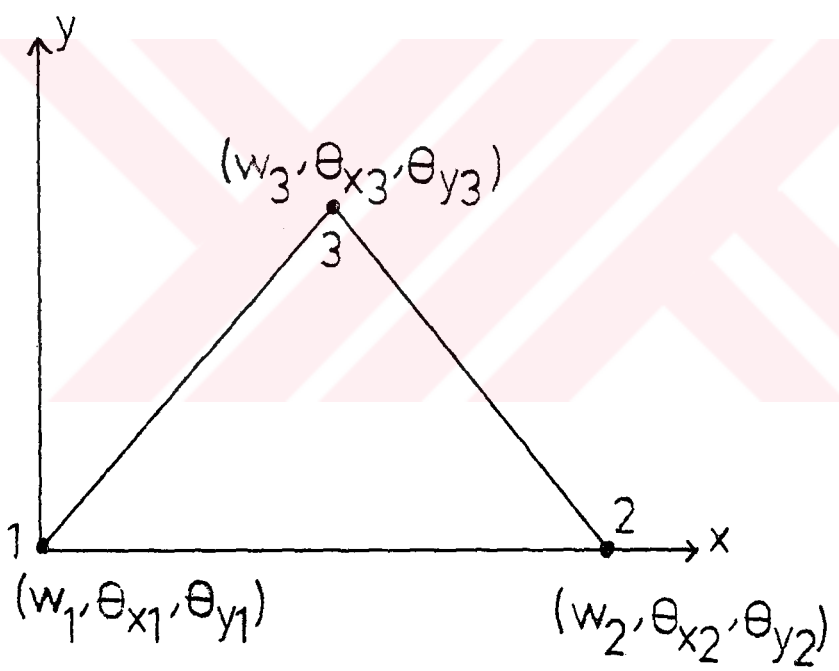


Figure 3.4. Nodal Variables for the Anisoparametric Model

According to this methodology the transverse displacement field w is one degree higher than the rotation fields θ_x and θ_y. The displacement field is given in terms of area coordinates as follows,

$$w = \xi_i w_i + \frac{1}{2}(b_k \xi_i \xi_j - b_j \xi_i \xi_k)\theta_{xi} + \frac{1}{2}(a_j \xi_i \xi_k - a_k \xi_i \xi_j)\theta_{yi}$$
$$\theta_x = \xi_i \theta_{xi} \tag{3.22}$$
$$\theta_y = \xi_i \theta_{yi}$$

A characteristic feature of Mindlin elements based on the displacement formulation is that they become numerically ill-conditioned when the plates are extremely thin. This numerical deficiency is due to the differences in the order of bending and shear stiffnesses[41]. The ratio of shear to bending stiffness is approximately A/h^2. For very thin elements A/h^2 becomes very large and bending stiffness coefficients can not be accurately represented because of the finite precision in digital computations. Tessler and Hughes [38] investigated the locking phenomenon and proposed an element appropriate shear correction factor by making use of energy principles. This factor is given by,

$$k = k^* \phi \tag{3.23}$$

where $k^* = \frac{1}{12}\pi^2$ is the classical shear correction factor, ϕ is the in-plane correction factor given as,

$$\phi = \frac{1}{1 + \alpha/2} \tag{3.24}$$

and

$$\alpha = \frac{\sum k_{sii}}{\sum k_{bii}} \cong \frac{A}{h^2} \tag{3.25}$$

is a generalized parameter defined as a ratio of the diagonal shear and bending stiffness coefficients.

3.6 Element Shape Matrix

(3.21) and (3.22) yield the shape matrix as

$$[N] = [[N_1][N_2][N_3]] \tag{3.26}$$

where

$$[N_i] = \begin{bmatrix} \xi_i & 0 & 0 & 0 & 0 & R_i \\ 0 & \xi_i & 0 & 0 & 0 & S_i \\ 0 & 0 & \xi_i & L_i & M_i & 0 \\ 0 & 0 & 0 & \xi_i & 0 & 0 \\ 0 & 0 & 0 & 0 & \xi_i & 0 \end{bmatrix} \tag{3.27}$$

where

$$
\begin{aligned}
R_i &= \frac{1}{2}(b_j\xi_k\xi_i - b_k\xi_i\xi_j) \\
S_i &= \frac{1}{2}(a_j\xi_k\xi_i - a_k\xi_i\xi_j) \\
L_i &= \frac{1}{2}(b_k\xi_i\xi_j - b_j\xi_i\xi_k) \\
M_i &= \frac{1}{2}(a_j\xi_i\xi_k - a_k\xi_i\xi_j)
\end{aligned}
\tag{3.28}
$$

The linear and geometric stiffness matrices can be calculated by substituting (3.26) into (3.3) and (3.4), respectively.

CHAPTER 4

FINITE ROTATIONS AND THE DEFORMATIONAL PART OF THE MOTION

4.1 General

A simple and practical formulation and procedure to remove the restriction of small rotations between succesive increments for geometrically nonlinear analysis of shells using flat triangular elements are presented. The rigid-body rotation of the coordinate system is assumed to be accomplished by an out-of-plane rotation followed by an in-plane rotation. The element internal stresses and nodal forces are calculated using the total deformational in-plane displacements and deformational nodal rotations which are obtained by subtracting the rigid-body motions from total displacements. The direct method proposed by Hsiao is used in the calculation of the deformational nodal rotations.

4.2 Finite Rotation of a Vector

If a vector is transported to a new position by the application of finite rotation vector about an axis whose normal is denoted by $\tilde{n}$, the relation between initial and rotated position of a vector is given as [16].

$$\vec{R}' = \cos\theta\vec{R} + (1-\cos\theta)(\tilde{n}.\vec{R})\tilde{n} + \sin\theta(\tilde{n}\times\vec{R}) \qquad (4.1)$$

where $\vec{R}$ and $\vec{R}'$ are the vectors of initial and rotated positions, respectively. θ is the angle of finite rotation. Figure 4.1 illustrates the rotation of a vector around an axis whose normal is $\tilde{n}$.

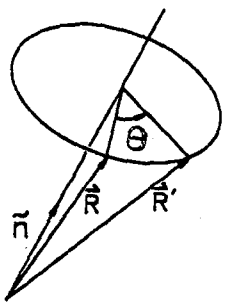


Figure 4.1. Finite Rotation of a Vector

4.3 Element Rigid Body Motions

Assume that the incremental-iterative method is used for the solution of nonlinear equilibrium equations and the configuration of the I^{th} increment is known. Then the rigid body motions are accomplished by the succesive application of the following motions.

1. A rigid body translation by ΔU_1 (global displacement of node 1): The element coordinate system is translated parallel to itself to the new position of node 1 as shown in Figure 4.2.

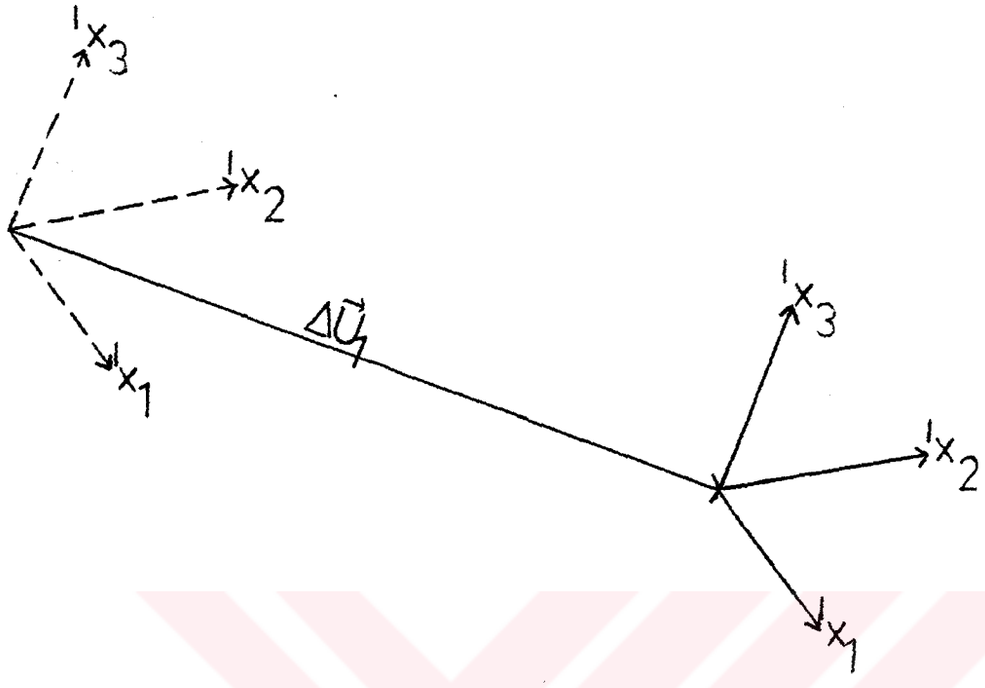


Figure 4.2. Rigid Body Translation of the Element Coordinate System

2. An out-of-plane rotation by the rotation vector $\vec{\alpha}$: The $^I x_i$ axes are tated by an angle $|\vec{\alpha}|$ about the axis perpendicular to $^I x_3$ and x_3 axes. This indicated in Figure 4.3.a. In this figure the resultant coordinate system is elled as x_i' and referred to as intermediate coordinate system. The out-of-ıe rigid body rotation vector $\vec{\alpha}$ is determined as

$$\vec{\alpha} = \cos^{-1}(^I\tilde{e}_3.\tilde{e}_3)\frac{^I\tilde{e}_3 \times \tilde{e}_3}{|^I\tilde{e}_3 \times \tilde{e}_3|} \tag{4.2}$$

$^I\tilde{e}_3$ and $\tilde{e}_3$ are the unit vectors associated with the $^I x_3$ and x_3 axes, ively, and they are referred to global coordinate system.

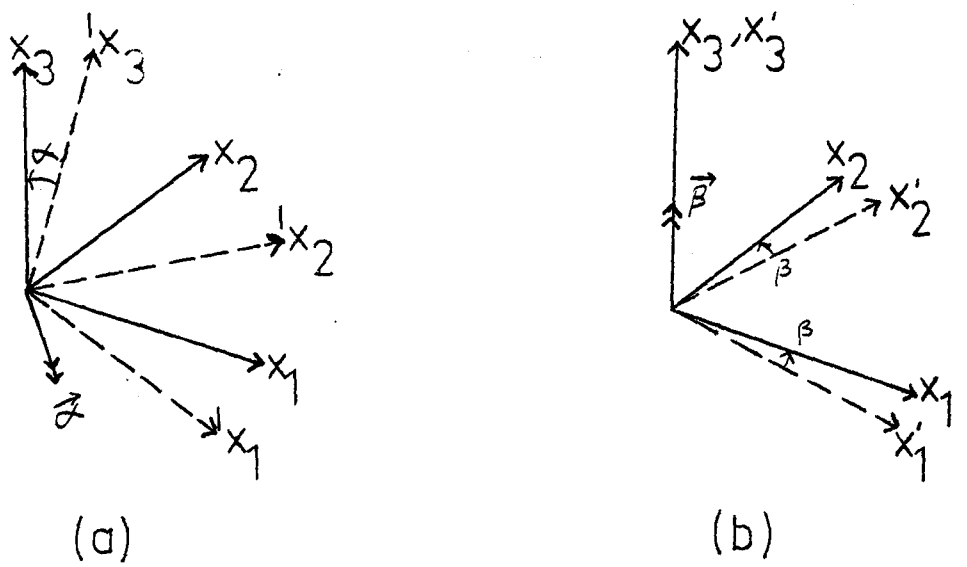


Figure 4.3. (a) Out-of-Plane Rigid-Body Rotation
(b) In-Plane Rigid-Body Rotation

3. An in-plane rotation by the rotation vector $\vec{\beta}$: The intermediate axes x_i' are rotated by an angle $|\vec{\beta}|$ about the axis perpendicular to x_1' and x_1 axes to coincide with x_i axes as shown in Figure 4.3.b. This rotation vector is determined by

$$\vec{\beta} = \cos^{-1}({}^I\tilde{e}_1 . \tilde{e}_1) \frac{{}^I\tilde{e}_1 \times \tilde{e}_1}{|{}^I\tilde{e}_1 \times \tilde{e}_1|} \tag{4.3}$$

4.4 Determination of Element Deformations

4.4.1 In-Plane Deformational Displacements

The in-plane relative nodal displacements are represented by vector $\{q_1\}$ and are calculated from the difference in the local coordinates at the initial and current configurations as illustrated in Figure 4.4. Then the displacement vector contains the following

$$\{q_1\}^T = \lfloor 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; \bar{u}_1 \;\; 0 \;\; 0$$
$$0 \;\; 0 \;\; 0 \;\; \bar{u}_3 \;\; \bar{v}_3 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \rfloor \qquad (4.4)$$

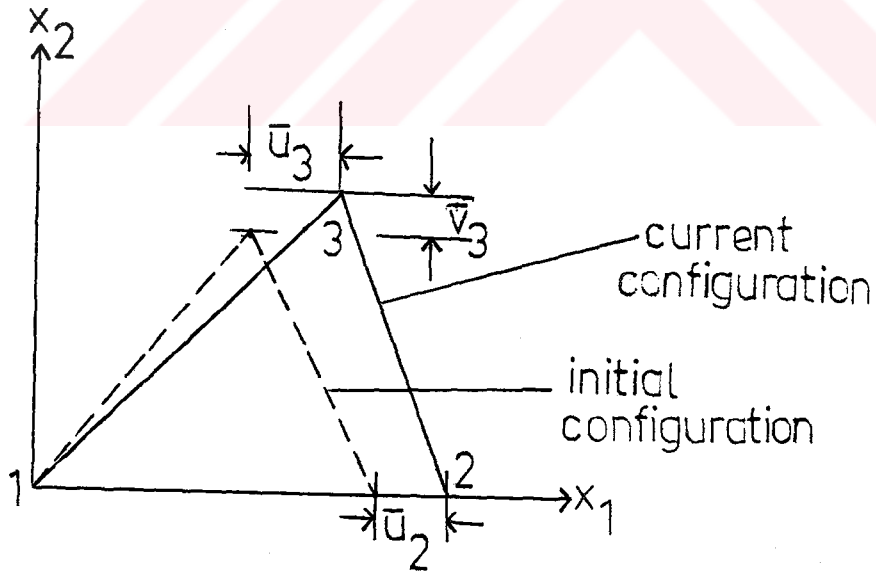


Figure 4.4. Element Membrane Deformational Displacements

4.4.2 Out-of-plane Deformational Rotations

The representation of the deformational nodal rotations are based on the assumption that these rotations are small enough to assure validity of vector operations. On the basis of this assumption the deformational nodal rotations of the element shown in Figure 4.5 is determined from the angles measured between the deformed normals $\tilde{n}_{dj}$ and undeformed normal $\tilde{n}_u$ of the element. Thus,the components associated with the flexural deformations are represented in a matrix $\{q_2\}$ as

$$\{q_2\}^T = \lfloor 0 \;\; 0 \;\; 0 \;\; \bar{\theta}_{x1} \;\; \bar{\theta}_{y1} \;\; 0 \;\; 0 \;\; 0 \;\; 0$$
$$\bar{\theta}_{x2} \;\; \bar{\theta}_{y2} \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; \bar{\theta}_{x3} \;\; \bar{\theta}_{y3} \;\; 0 \rfloor \qquad (4.5)$$

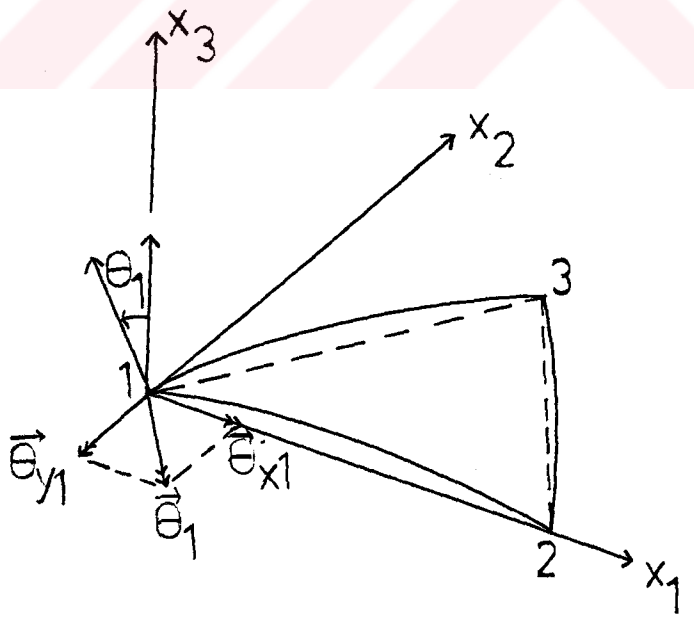


Figure 4.5. Deformational Nodal Rotations

In order to determine these normals, Hsiao [16] proposed two methods termed as direct and incremental methods. In that study the performance of each method has been tested with a number of numerical examples and it was concluded that both methods give identical results. In the present study the direct method has been selected for the determination of deformed normals. The procedure associated with this method is as follows

1. The whole element is translated by ΔU_1 (the global displacement of node 1 as illustrated in Figure 4.2).

2. The whole element is then rotated by the rotation vector $\vec{\alpha}$ passing through node 1 except ${}^{I}\tilde{n}_{dj}$, the deformed normals of the element surface in the I^{th} equilibrium configuration. See Figure 4.3.

3. The deformed normals ${}^{I}\tilde{n}_{dj}$ are rotated to n'_{dj} by the application of finite rotation vectors $\Delta\theta'_{tj}$ which are the projections of $\Delta\vec{\theta}_j$ in the $x'_1 - x'_2$ plane as shown in Figure 4.6.

4. The rotation vector $\vec{\beta}$ passing through node 1 is applied to the whole element including the unit vector $\tilde{n}'_{dj}$. In Figure 4.7 vector $\tilde{n}_{dj}$ are rotated by this vector to reach their final position.

Finally, the out-of-plane deformational nodal rotations, referred to the current element coordinate system can be expressed as

$$\vec{\theta}_j = \begin{Bmatrix} \bar{\theta}_{xj} \\ \bar{\theta}_{yj} \end{Bmatrix} = \cos^{-1}(\tilde{n}_u . \tilde{n}_{dj}) \frac{\tilde{n}_u \times \tilde{n}_{dj}}{|\tilde{n}_u \times \tilde{n}_{dj}|} (i = 1, 2, 3) \qquad (4.6)$$

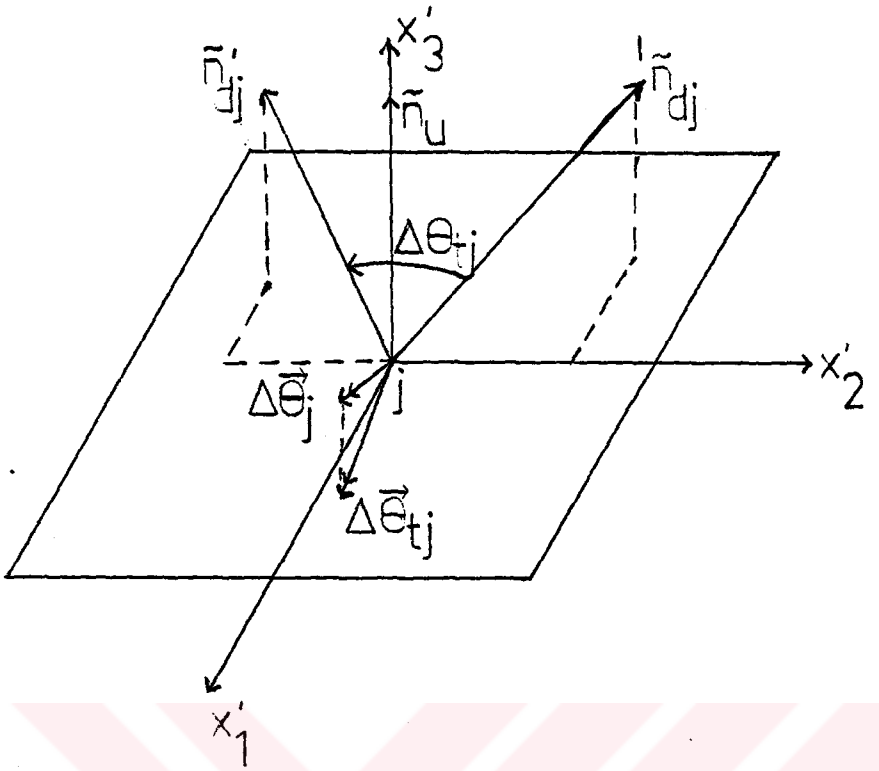


Figure 4.6. Step 3 of the Deformational Process of the Direct Method

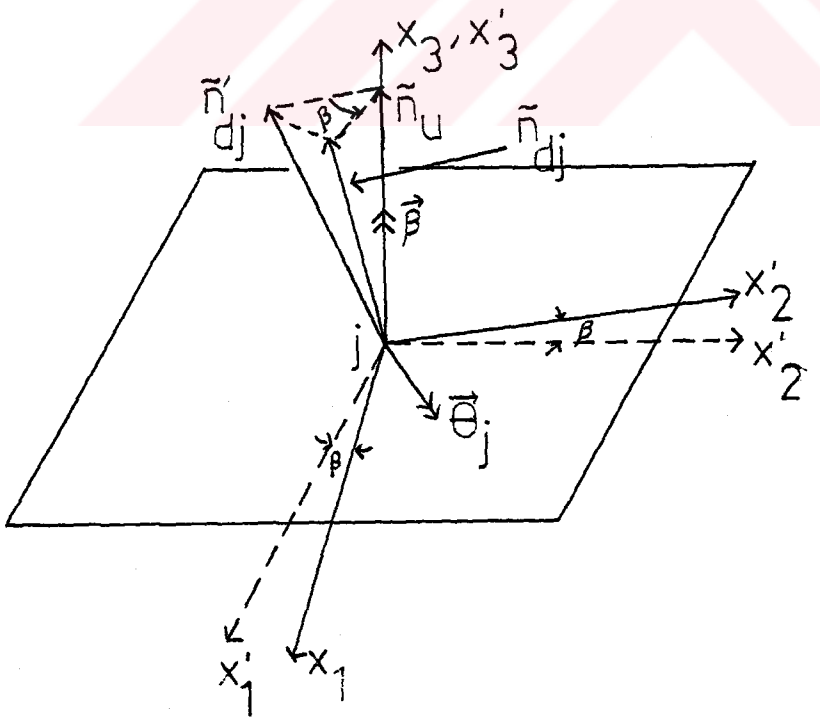


Figure 4.7 Step 4 of the Deformational Process of the Direct Method

4.4.3. In-Plane Deformational Rotations

The element in-plane deformational rotations defined in the current equilibrium configuration can be determined, incrementally, as

$$\bar{\theta}_{zj} = {}^{I}\bar{\theta}_{zj} + \Delta\bar{\theta}_{zj} \tag{4.7}$$

where ${}^{I}\bar{\theta}_{zj}$ are the in-plane deformational rotations calculated in the previous equilibrium configuration, $\Delta\bar{\theta}_{zj}$ denote the in-plane incremental deformational rotations to be defined as

$$\Delta\bar{\theta}_{zj} = \Delta\bar{\theta}pj - \beta \tag{4.8}$$

where $\Delta\bar{\theta}_{pj}$ is the component of the finite rotation vector $\Delta\vec{\theta}_j$ in local z direction. Finally, the in-plane deformational nodal rotations are represented in vector form as

$$\{q_3\}^T = \lfloor 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; \bar{\theta}_{z1} \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; \bar{\theta}_{z2} \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; \bar{\theta}_{z1} \rfloor \tag{4.9}$$

4.5. Element Stresses and Nodal Force Vector

The vector storing the deformational part of the total element motion as discussed in chapter 2 is obtained by superposing $\{q_1\}, \{q_2\}$ and $\{q_3\}$ as

$$\{q_d\}^T = \lfloor 0 \;\; 0 \;\; 0 \;\; \bar{\theta}_{x1} \;\; \bar{\theta}_{y1} \;\; 0 \;\; \bar{u}_2 \;\; 0 \;\; 0$$
$$\bar{\theta}_{x2} \;\; \bar{\theta}_{y2} \;\; 0 \;\; \bar{u}_3 \;\; \bar{v}_3 \;\; 0 \;\; \bar{\theta}_{x3} \;\; \bar{\theta}_{y3} \;\; 0 \rfloor \qquad (4.10).$$

This vector is then used to find the stress and load vectors in (3.7) and (3.13), respectively.

CHAPTER 5

SOLUTION ALGORITHM

5.1 General

The incremental-iterative strategies described here are based on a full Newton-Raphson approach in which the tangential stiffness matrix is updated throughout the iterative cycles. The matrix implementations associated with both load and displacement incrementation strategies are given with their algorithms.The convergence creation required to achieve a succesive accuracy also discussed here.

5.2 Load Control

The load incrementation process is very simple to implement computer and it requires the following incremental equilibrium equations.

$$[K_T]^i\{U\}^{i+1} = \lambda^i\{R\} - \{f\}^i \tag{5.1}$$

where $\{U\}^{i+1}$ is the vector of incremental displacements required for the next iteration, λ^i is the load parameter varying from zero to one and it determines the current load level on the structure when multiplied by the external load vector as seen in (5.1). The true equilibrium configuration is then achieved by means of the pure Newton Raphson iteration method in which the tangent stiffness matrix is updated at each iteration.

The iterations required between two successive load increments should be terminated when sufficient accuracy is obtained. There are a few convergence criteria proposed to achieve the desired accuracy. The most common criterion employed is the displacement norm criterion [35] .

$$\frac{\|\{U\}\|}{\|\{U_T\}\|} \leq \rho_{tol} \tag{5.2}$$

where $\{U\}$ and $\{U_T\}$ are the incremental and total displacements respectively, ρ_{tol} is a tolerance value for the given criterion, $\|.\|$ denotes the norm of a vector and it is defined as

$$\|\{X\}\| = \sum_{i=1}^{n} |X_i| \tag{5.3}$$

where

$$\{X\} = \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{Bmatrix} \tag{5.4}$$

The algorithm dealing with the load control strategy is shown in Figure 5.1.

i) Newton-Raphson algoritm for force incrementation:

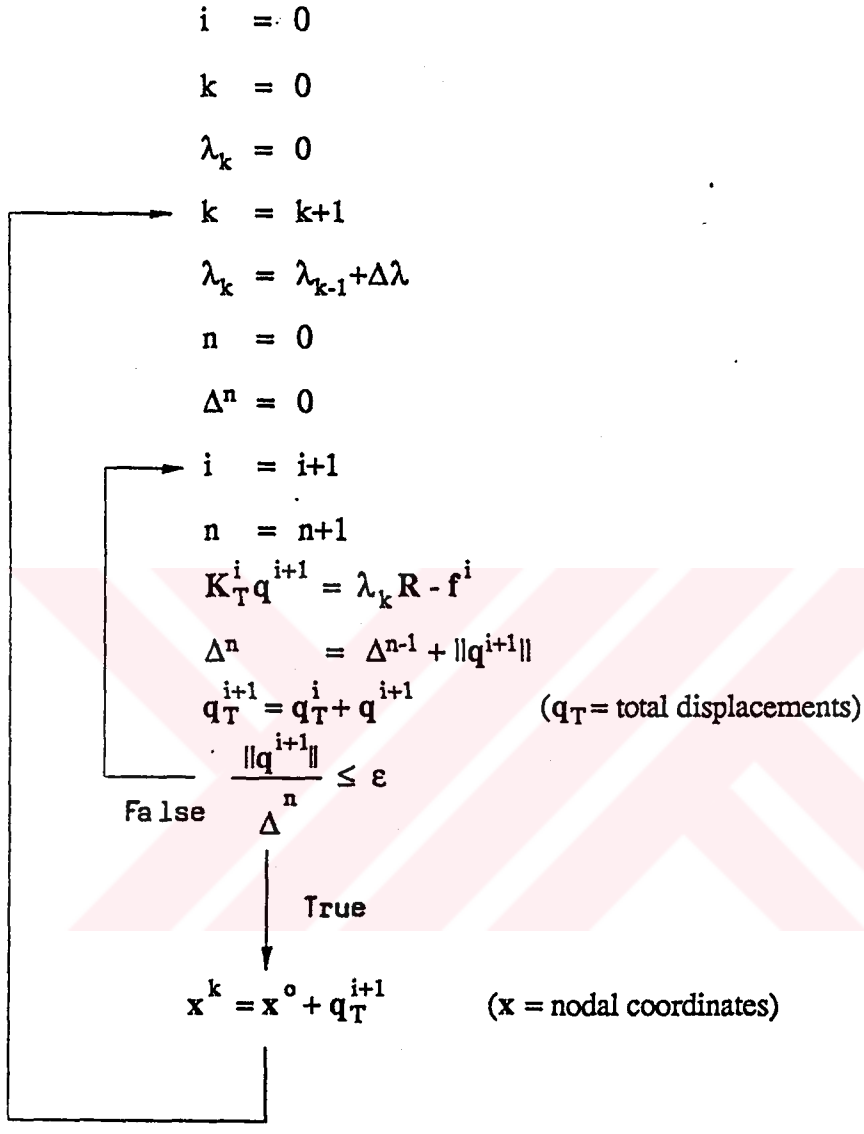


Figure 5.1. Solution Algorithm of the Force Control Strategy

5.3 Displacement Control

As pointed out in section 5.1, the tangent stiffness matrix of a doubly curved structure approaches singularity as the load deflection curve is near the limit point. In such cases the analysis procedure should be switched from load control to another strategy which avoids the singularity of the tangent stiffness matrix. The common way of improving the stiffness matrix is to increment displacement components rather than the loads. The procedure given in [32] is shown to be efficient, reliable and easy to implement.

Suppose that one of the displacement components q_j is specified. Removing q_j from the left side of (5.1) and partitioning the remaining equations, one obtains

$$\begin{bmatrix} [\bar{K}] & -\{\bar{R}\} \\ \{K_j\}^T & R_j \end{bmatrix} \begin{Bmatrix} \{\bar{q}\} \\ \lambda \end{Bmatrix} = - \begin{Bmatrix} \{\bar{f}\} \\ f_j \end{Bmatrix} - q_j \begin{Bmatrix} \{K_j\} \\ K_{jj} \end{Bmatrix} \tag{5.5}$$

where $[\bar{K}]$ is the $[K_T]$ matrix with j^{th} row deleted, $\{K_j\}$ is the j^{th} column of $[K_T]$ with j^{th} row deleted, $\{\bar{R}\}$ is the external load vector with j^{th} component removed, R_j is the j^{th} component of $\{R\}$, $\{\bar{q}\}$ is the vector of unspecified displacements, q_j is the specified displacement component, λ is the load factor to be determined, $\{\bar{f}\}$ is the internal load vector with j^{th} component deleted and finally f_j is the j^{th} component of $\{f\}$.

The partitioned form of the coefficient matrix in (5.5) seems to be unsymmetric. Hence, the direct solution of this equation would be difficult.

However, it is possible to expand (5.5) as follows,

$$[\bar{K}]\{\bar{q}\} = \lambda\{\bar{R}\} + \{\bar{f}\} - q_j\{K_j\} \qquad (5.6.a)$$

$$\{K_j\}^T\{\bar{q}\} = f_j - q_j K_{jj}. \qquad (5.6.b)$$

then the solution for $b f \bar{q}$ may be written

$$\{\bar{q}\} = \{A\} + \lambda\{B\} \qquad (5.7)$$

where $\{A\}$ and $\{B\}$ are the solutions of

$$\{\bar{K}\}\{A\} = \{\bar{f}\} - q_j\{K_j\} \qquad (5.8.a)$$

$$\{\bar{K}\}\{B\} = \{\bar{R}\} \qquad (5.8.b)$$

When (5.8) are substituted into (5.6.b), one obtains

$$\lambda = \frac{f_j - \{K_j\}^T\{A\} - K_{jj}q_j}{\{K_j\}^T\{B\} - R_j} \qquad (5.9)$$

Having determined the load parameter λ with the solution vectors $\{A\}$ and $\{B\}$, the vector of unspecified displacements $\{\bar{q}\}$ can be computed directly using (5.6). The procedure given after expanding (5.5) avoids the solution of

nonsymmetric equations. Moreover the solution of the two sets of equations ((5.6.a) and (5.6.b)) is shown to be computationally efficient due to the existence of the same coefficient matrices which may be solved simultaneously.

The convergence criterion chosen for the present strategy is the same as that used in the previous case.

The algorithm based on displacement incrementation strategy is given in Figure 5.2.

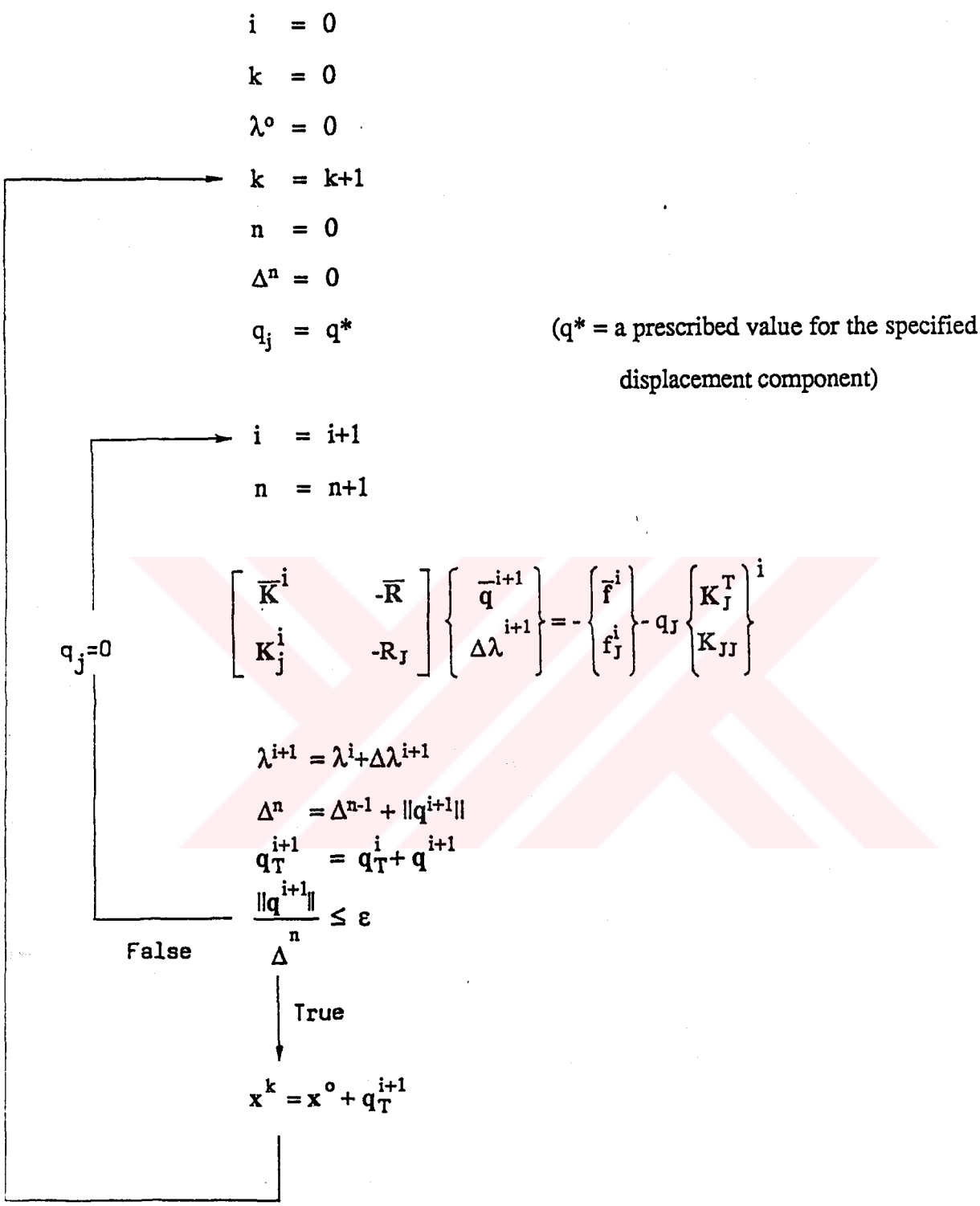


Figure 5.2. Solution Algorithm of the Displacement Control Strategy

CHAPTER 6

NUMERICAL STUDIES

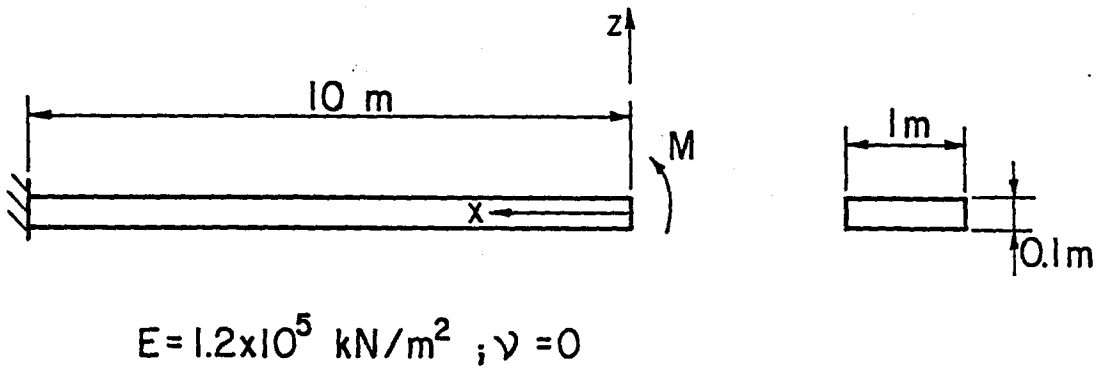
6.1 General

In this chapter the performance of the present element is to be assessed by means of a number of numerical examples consisting of wide beams, plates and shells undergoing large deflections, large rotations and instability. Some of the problems are studied under non-conservative loading where the direction of applied loading remains normal to the deformed surface of structure. The results obtained by the present model are compared with either analytical or other numerical solutions. The numbers enclosed by paranthesis in each load deflection curve indicate the number of iterations between two succesive increments. The deformed configurations at various load levels are also plotted to demonstrate the applicability of the present formulation. In all problems considered here, the error tolerance is taken as $\rho = 0.005$ for convergence at each equilibrium configuration.

6.2 Wide Beams

6.2.1 Pure Bending of a Cantilever Beam

The first example considered in our numerical analysis is a cantilever beam subjected to a concentraded end moment as shown in Figure 6.1. This problem has been widely used by many investigators as a test for large deflection and large rotation analysis. The beam is discretized by 20 elements as shown in



$E = 1.2 \times 10^5\ kN/m^2\ ;\ \nu = 0$

Figure 6.1. Cantilever Beam with End Moment

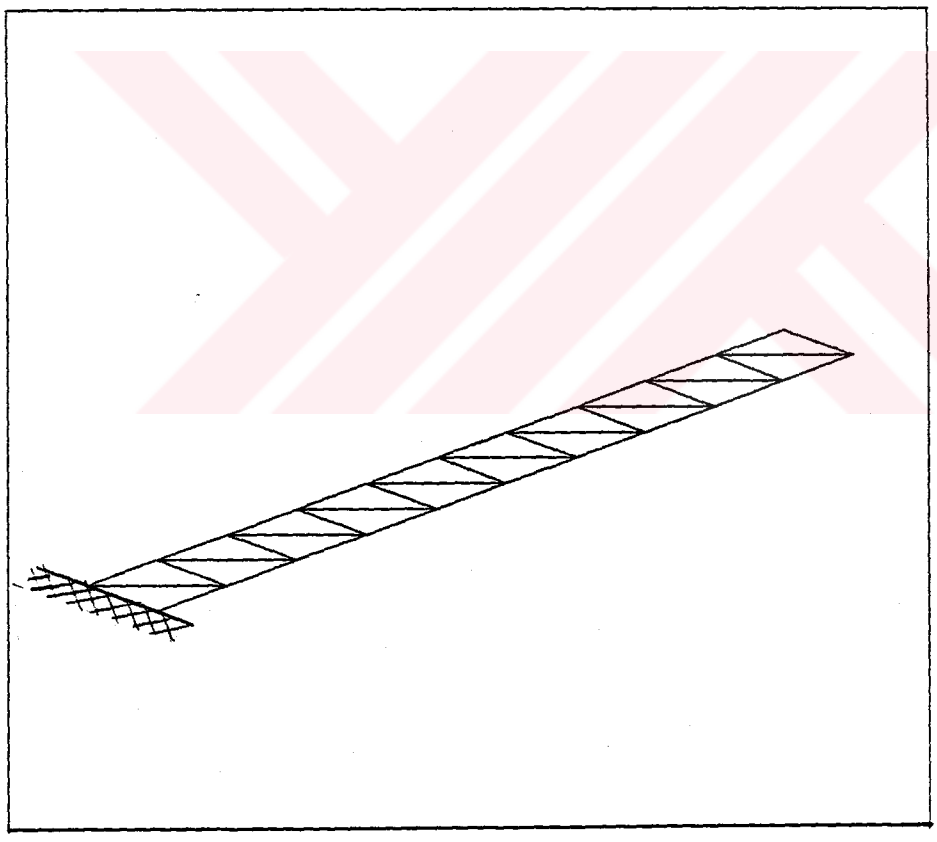


Figure 6.2. The Finite Element Discretization of the Beam Using 20 Elements

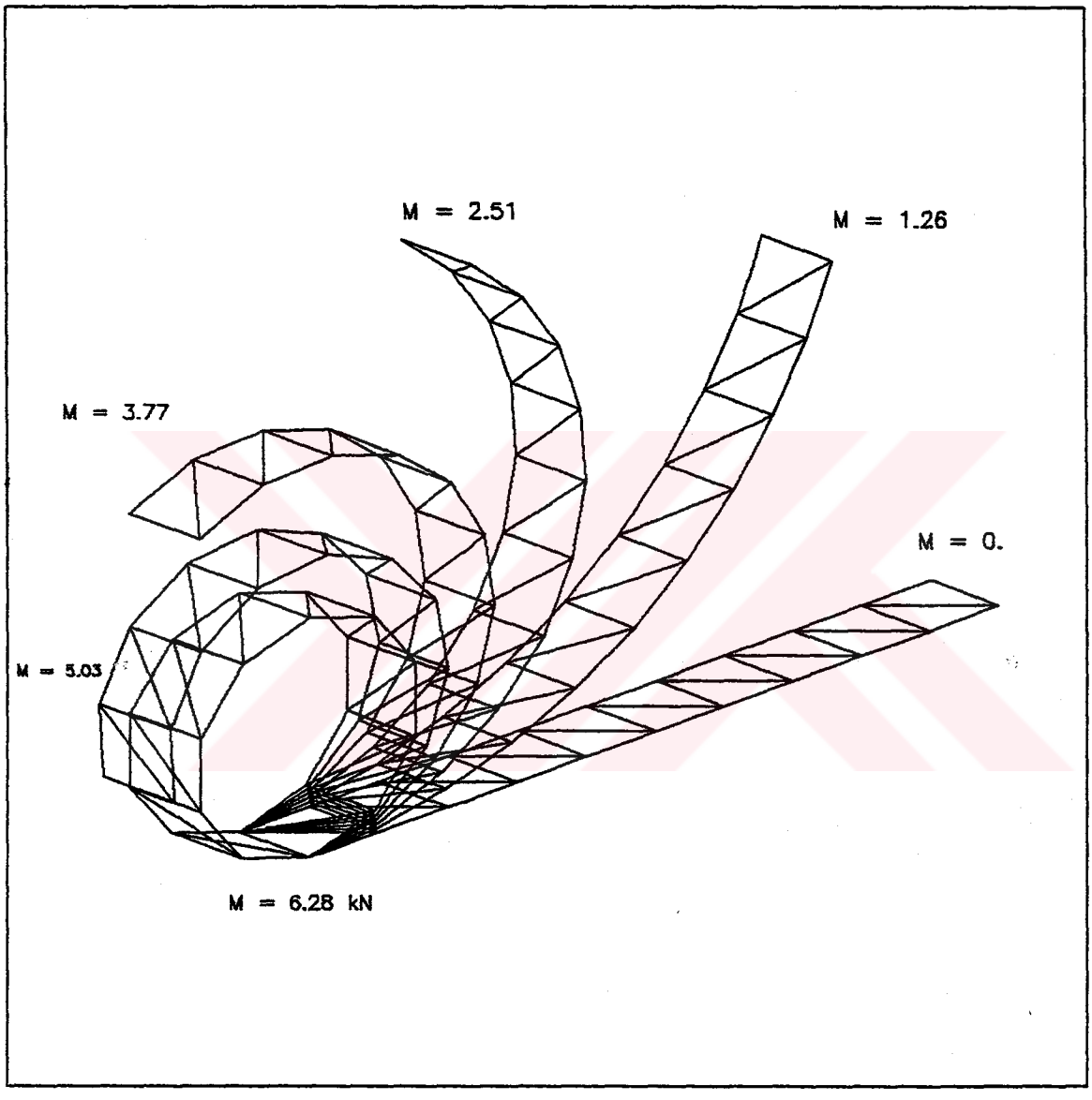


Figure 6.3. Deformed Configuration of the Cantilever Beam with End Moment

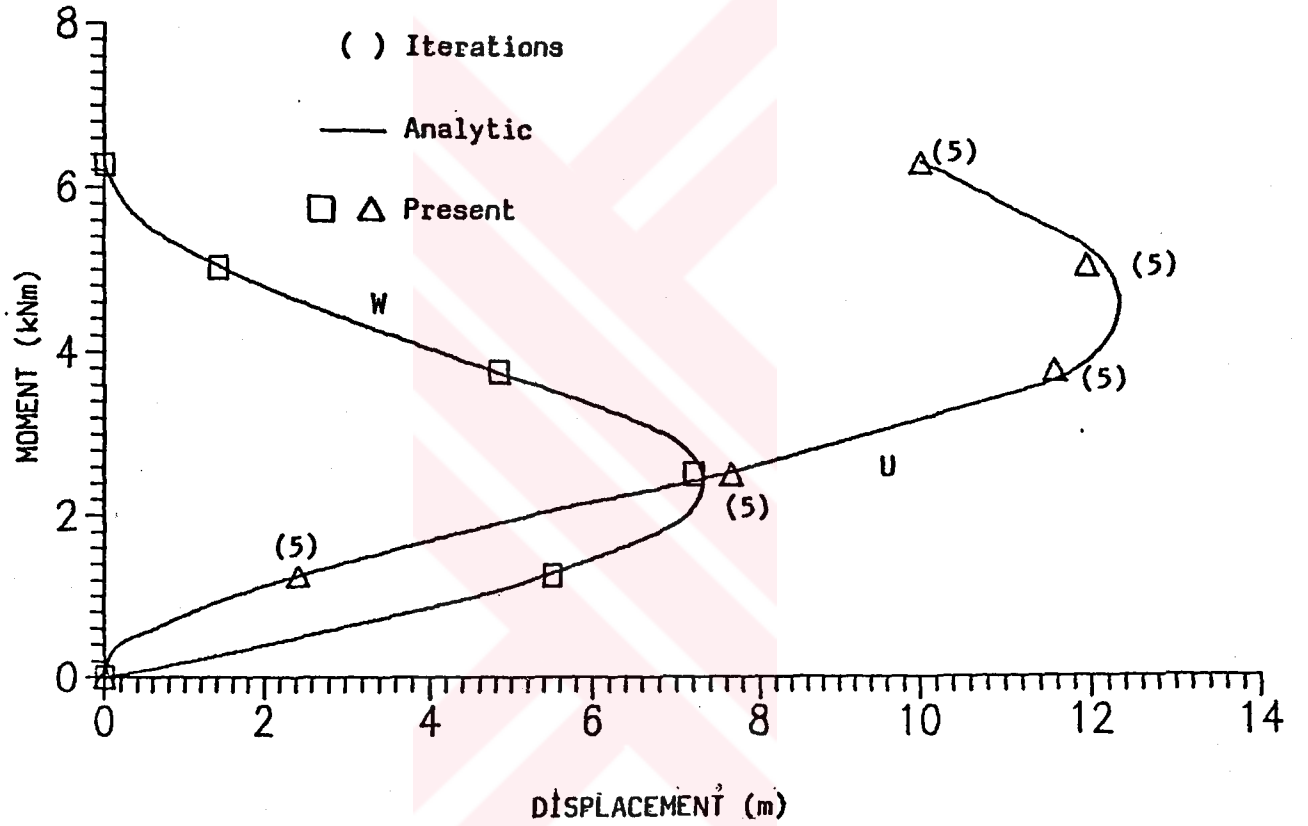


Figure 6.4. Load-Deflection Curve for the Cantilever Beam with End Moment

Figure 6.2. The solution is obtained in five equal load increments until the end point is rotated 2π radians. The deformed configurations at each increment are illustrated in Figure 6.3. This problem has also been solved by Horrigmoe and Bergan [14] with 12 increments (using 10 elements), by Surana and Sorem [50] using a curved beam element with 10 load increments and by Hsiao[16] by 3 increments. The load-deflection curve is shown in Figure 6.4.

It is seen that the results are in very good agreement with the analytic solution given in [42].

6.2.2 Cantilever Beam Subjected to End Shear

The beam discussed in the previous example is selected to study the effect of non-conservative loading. To this end, two loading situations were considered, the first of these being that of a standard conservative load P_c with fixed direction normal to the initial straight beam. The second one is considered as nonconservative loading P_{nc} which follows the direction of the normal to the element at the tip. The beam is discretized by 20 elements. The final configuration is achieved in four equal load increments. The deformed configurations for both cases are shown in Figure 6.5.a and Figure 6.5.b, respectively. As can be observed from Figures 6.5, there is a considerable difference in the responses for conservative and nonconservative loading at the higher load levels.

In Figure 6.6. the present solution for conservative loading case is compared with the solution given by Mattiasson [43]. Also shown in Figure 6.7 are the

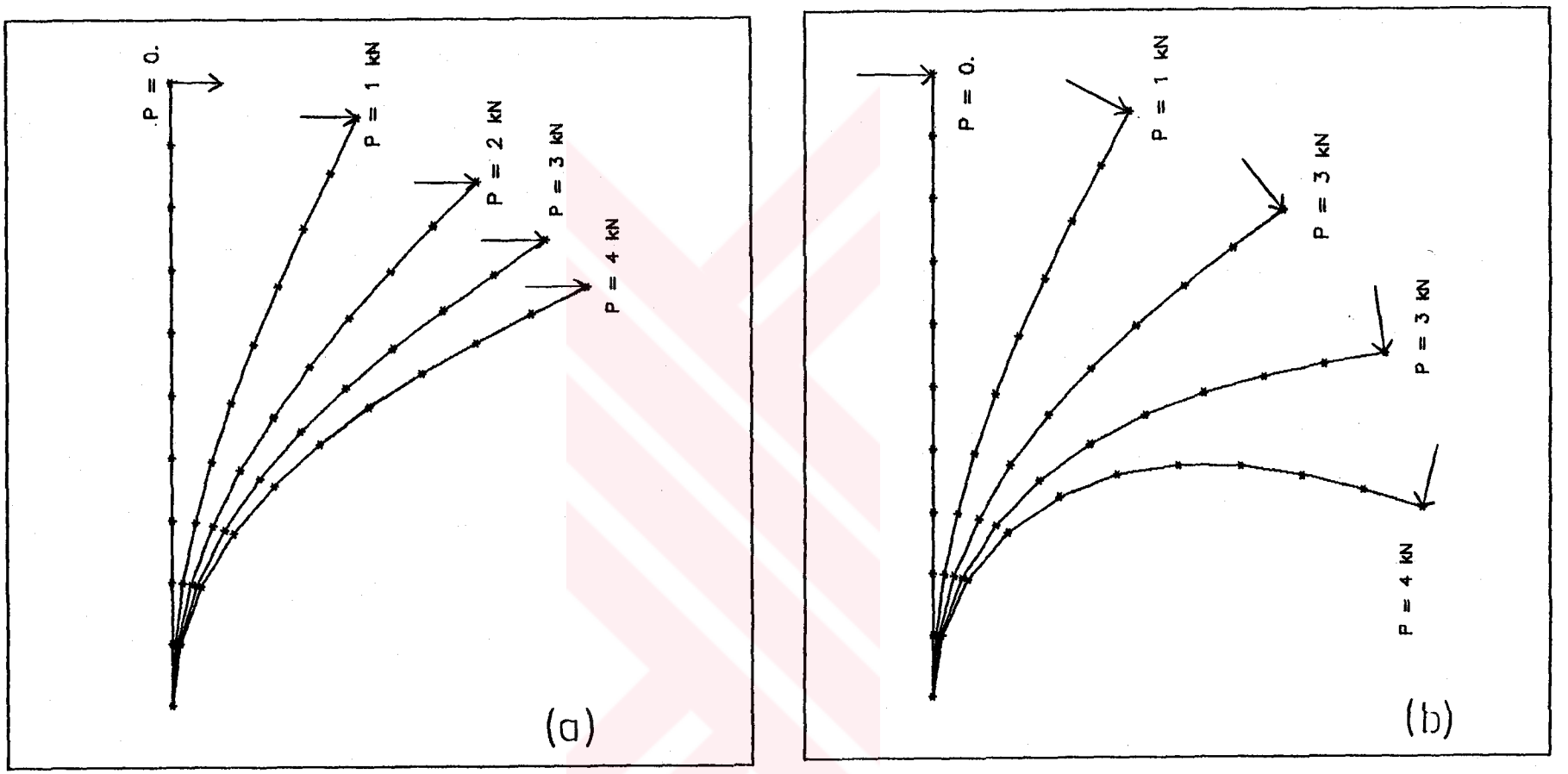


Figure 6.5. Deformed Configurations of the Cantilever Beam under (a) Conservative and (b) Non-conservative Tip Load

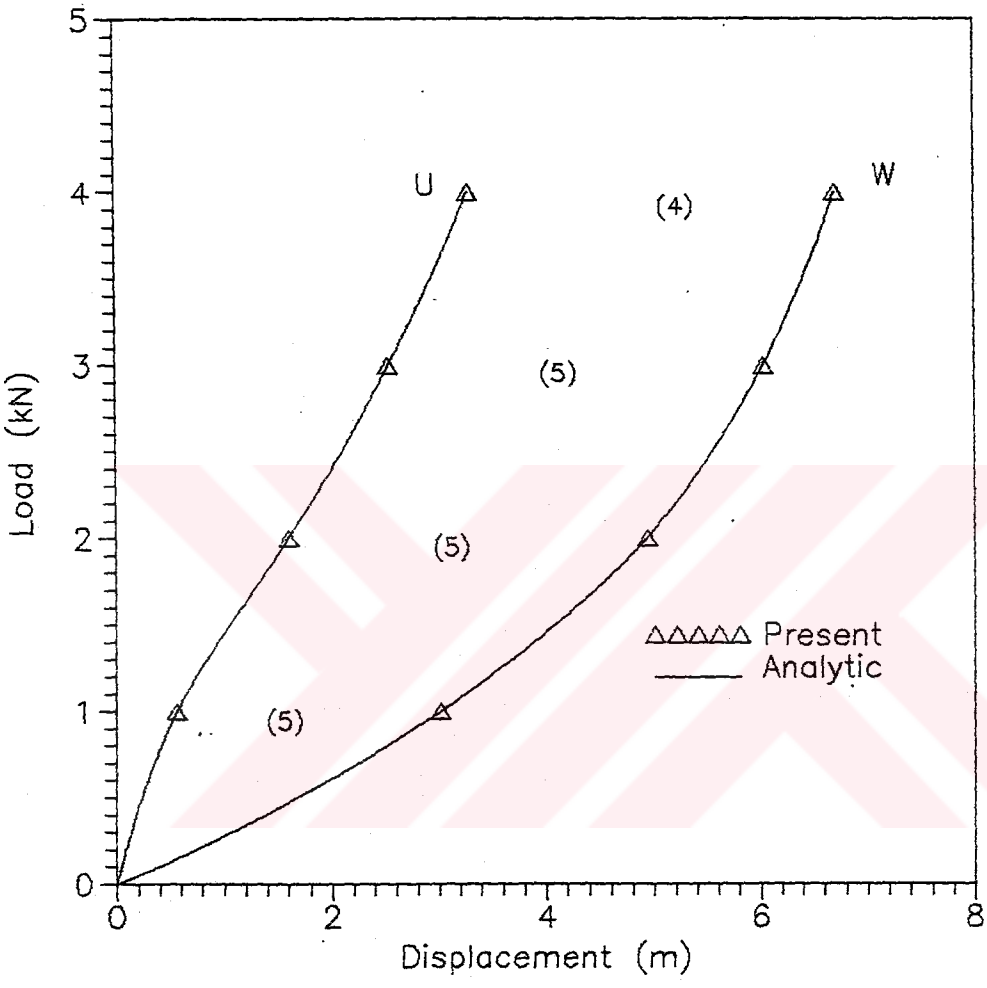


Figure 6.6. Load-deflection Curve for the Cantilever Beam with Conservative Tip Load

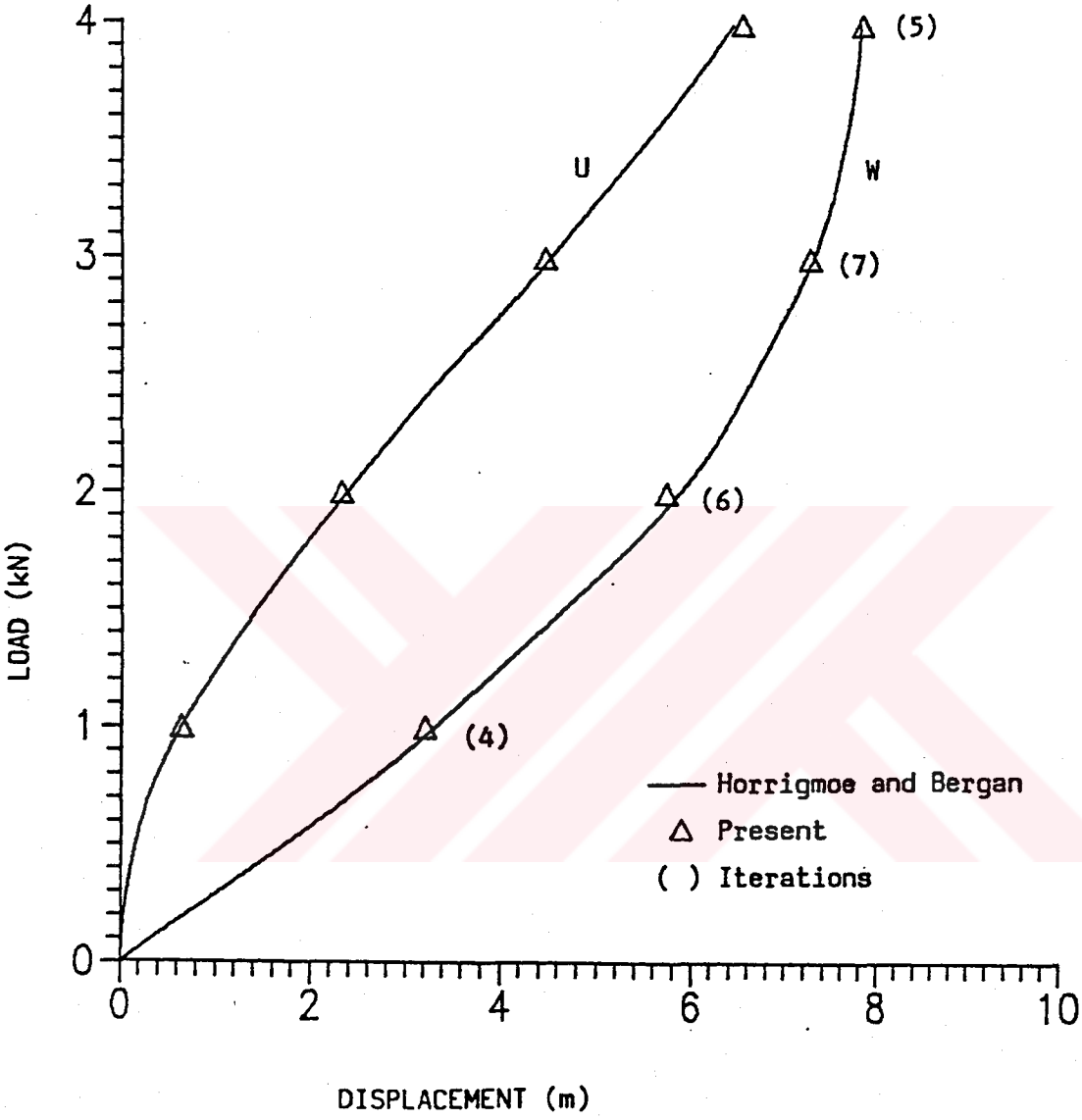


Figure 6.7. Load-deflection Curve for the Cantilever Beam with Non-Conservative Tip Load

solutions for non-conservative case obtained by the present model and Horrigmoe and Bergan [14] in which 7 rectangular elements are used . Again the results show that the present model is highly accurate under both conservative and nonconservative loading.

6.3 Plate Bending Problems

6.3.1 Simply Supported Plate Subjected to Lateral Pressure

This example is chosen to study the effect of membrane stresses on bending action of a plate. The plate is subjected to a uniform pressure loading as shown in Figure 6.8. Dimensions and physical properties are also illustrated in Figure 6.8. Due to symmetry,only one-quarter of the plate is studied using a 3x3 mesh. The number of load increments to obtain the solution is five. This problem has been solved analytically by Levy [44]. The comparison between the present and Levy's results are shown in Figure 6.9. As can be seen that the results are in agreement with the analytic solution although there is a slight difference at the higher load levels. It is observed from Figure 6.9 that there is a decrease of displacements in the load deflection curve due to the stiffening effect eventhough the largest deflection is of the order of the thickness of the plate. In Figure 6.10, the final deformed configuration of one quarter of the plate is demonstrated.

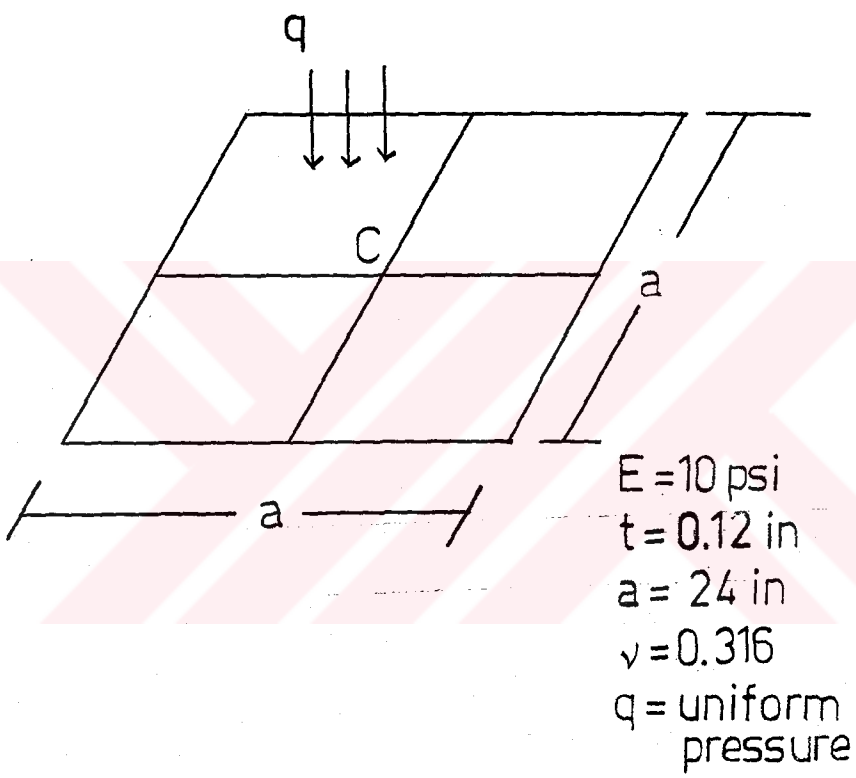


Figure 6.8. Simply Supported Plate Subjected to Lateral Pressure

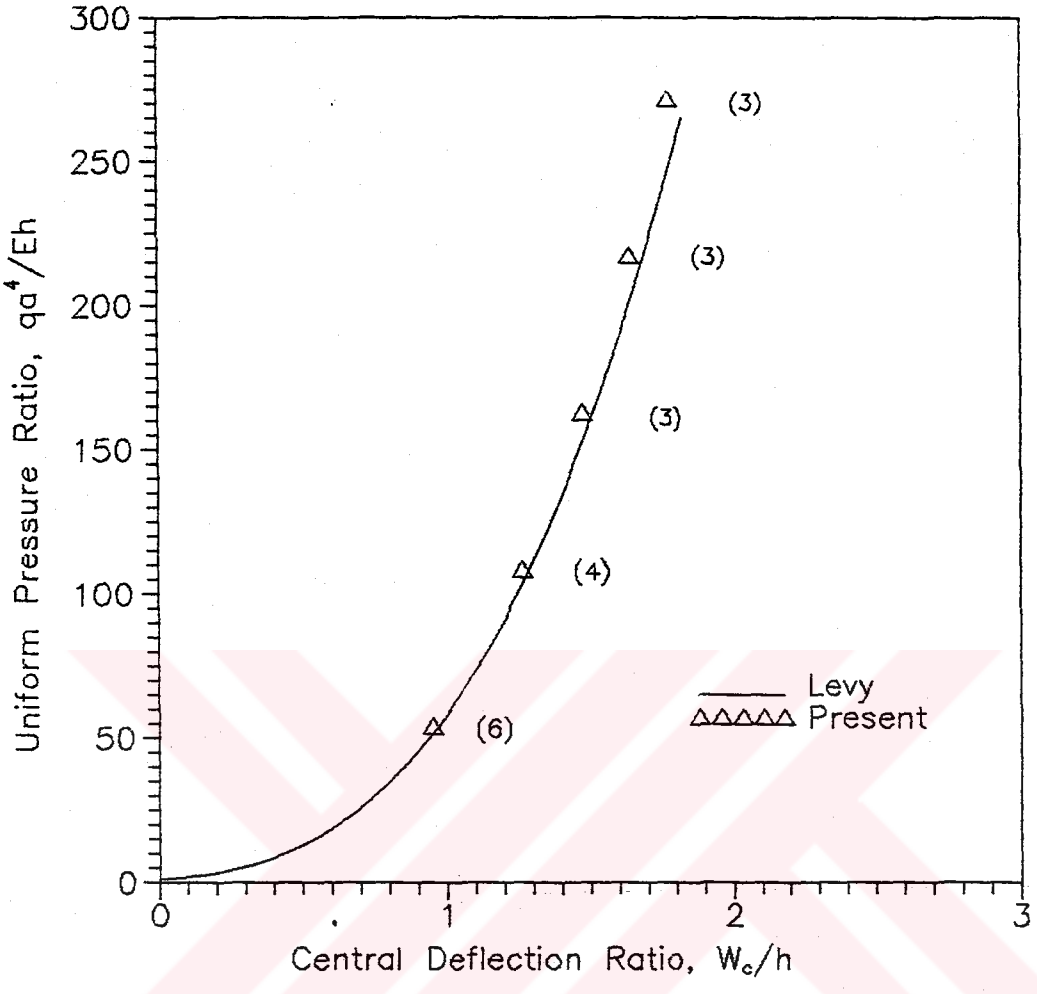


Figure 6.9. Load-deflection Curve for the Simply Supported Plate

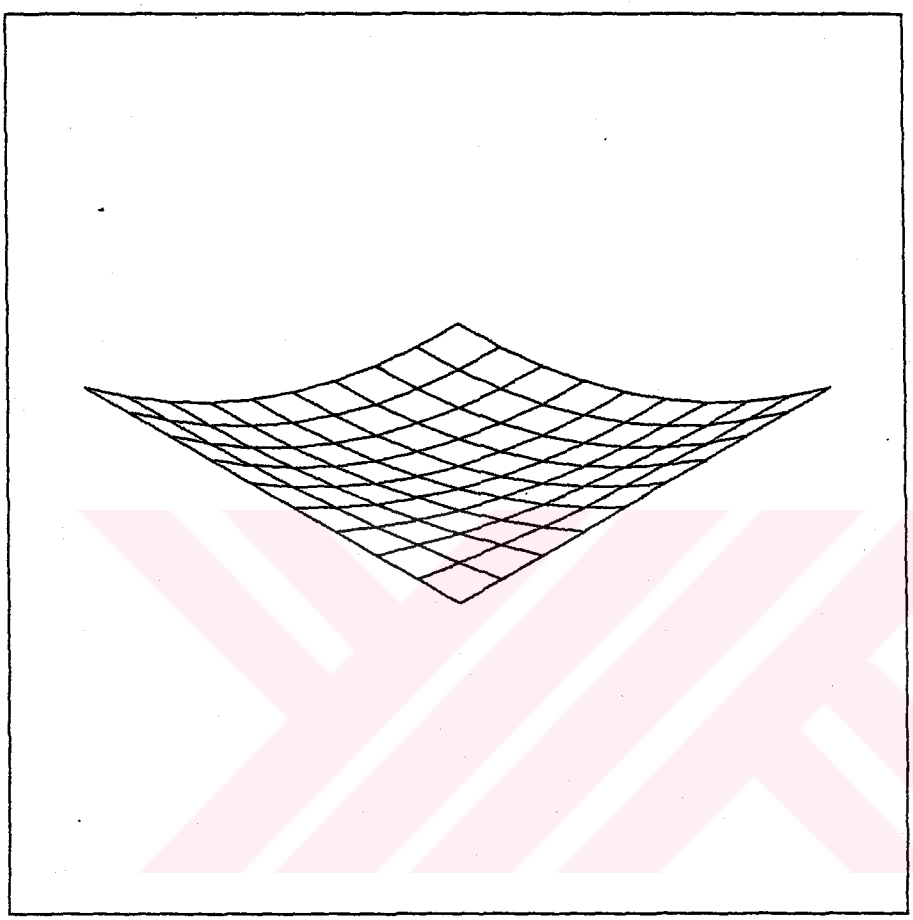
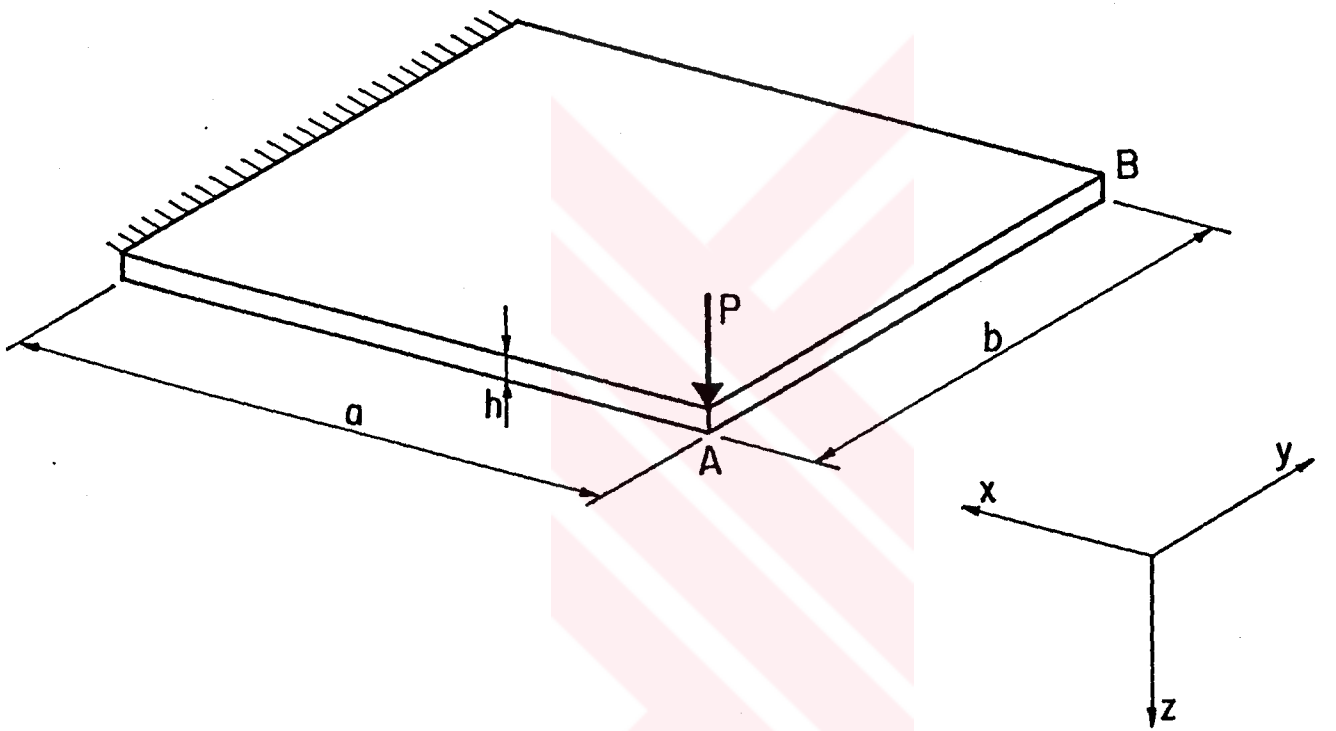


Figure 6.10. The Final Deformed Configuration of One-Quarter of the Plate

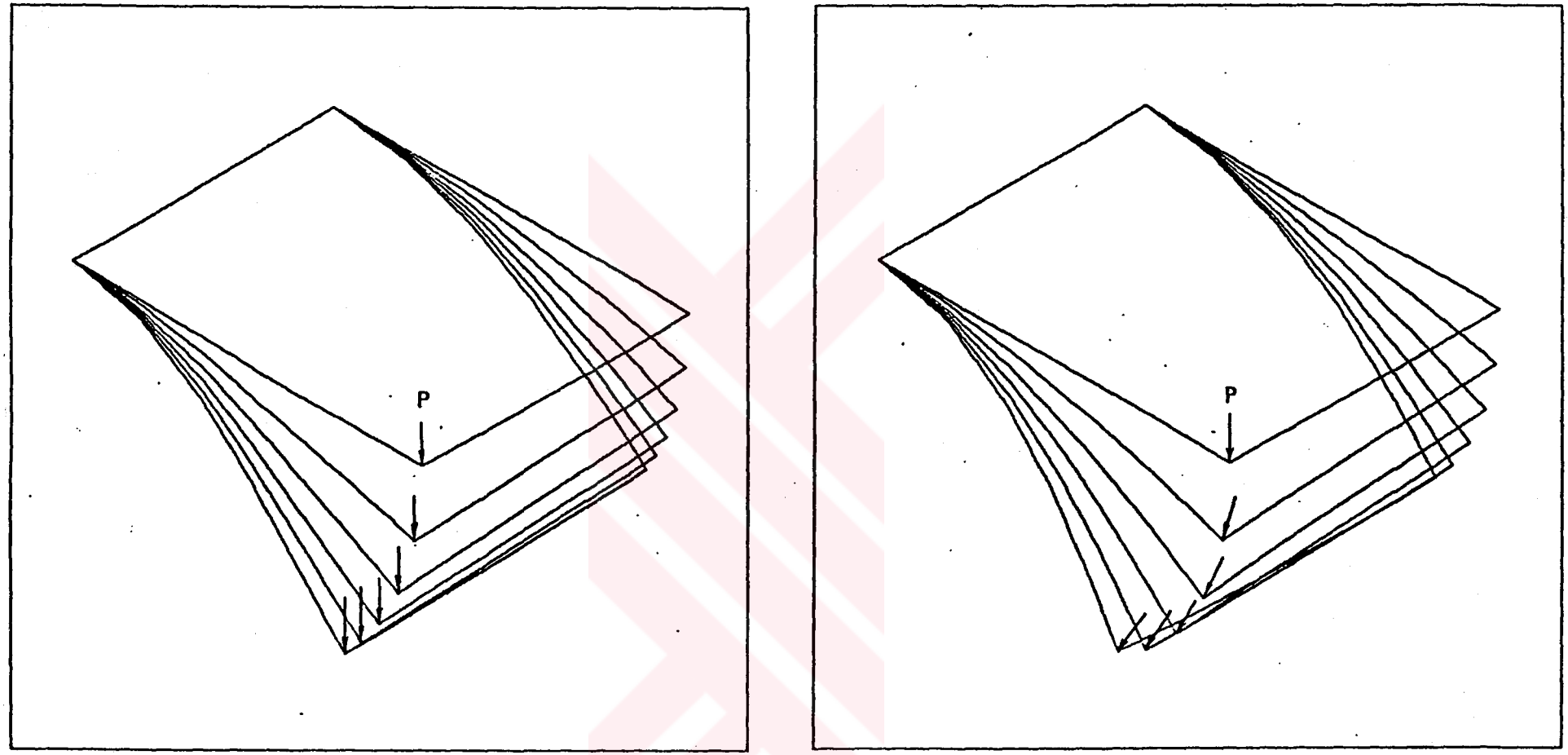
6.3.2 Cantilever Plate with Conservative and Non-conservative End Load

A cantilever plate subjected to a concentrated load at its one end has been studied. Figure 6.11 shows the geometry and physical properties of the plate. A 5x5 mesh model is used to analyze the large deflection and rotation of the plate considering two loading cases : (a) load with a constant vertical direction (conservative loading) and (b) load always normal to the deformed middle surface (non-conservative loading). Five equal load increments have been used in this analysis. The deformed configurations at each increment upto 34 kN for the two loading cases are shown in Figure 6.12.a and Figure 6.12.b, respectively. The conservative loading case has also been solved by Hsiao [16] using a 6x8 mesh discretization. the load-deflection curves for this case are shown in Figure 6.13 and very good agreement is observed with respect to Hsiao's solution. There is however, no 'benchmark' solution available for nonlinear response of the non-conservative loading case. In this study,the non-conservative case has also been solved and the load deflection curves are given in Figure 6.14. From Figure 6.13 and Figure 6.14, it can be seen that non-conservative loading yields very large displacements at the higher load levels.



a = 40 m
b = 30 m
h = 0.4 m
$E = 1.2 \times 10^8$ N/m^2
$\nu = 0.3$

Figure 6.11. Cantilever Plate with End Load



a) Conservative load

b) Nonconservative load

Figure 6.12. Deformed Configurations of the Cantilever Plate with End Load

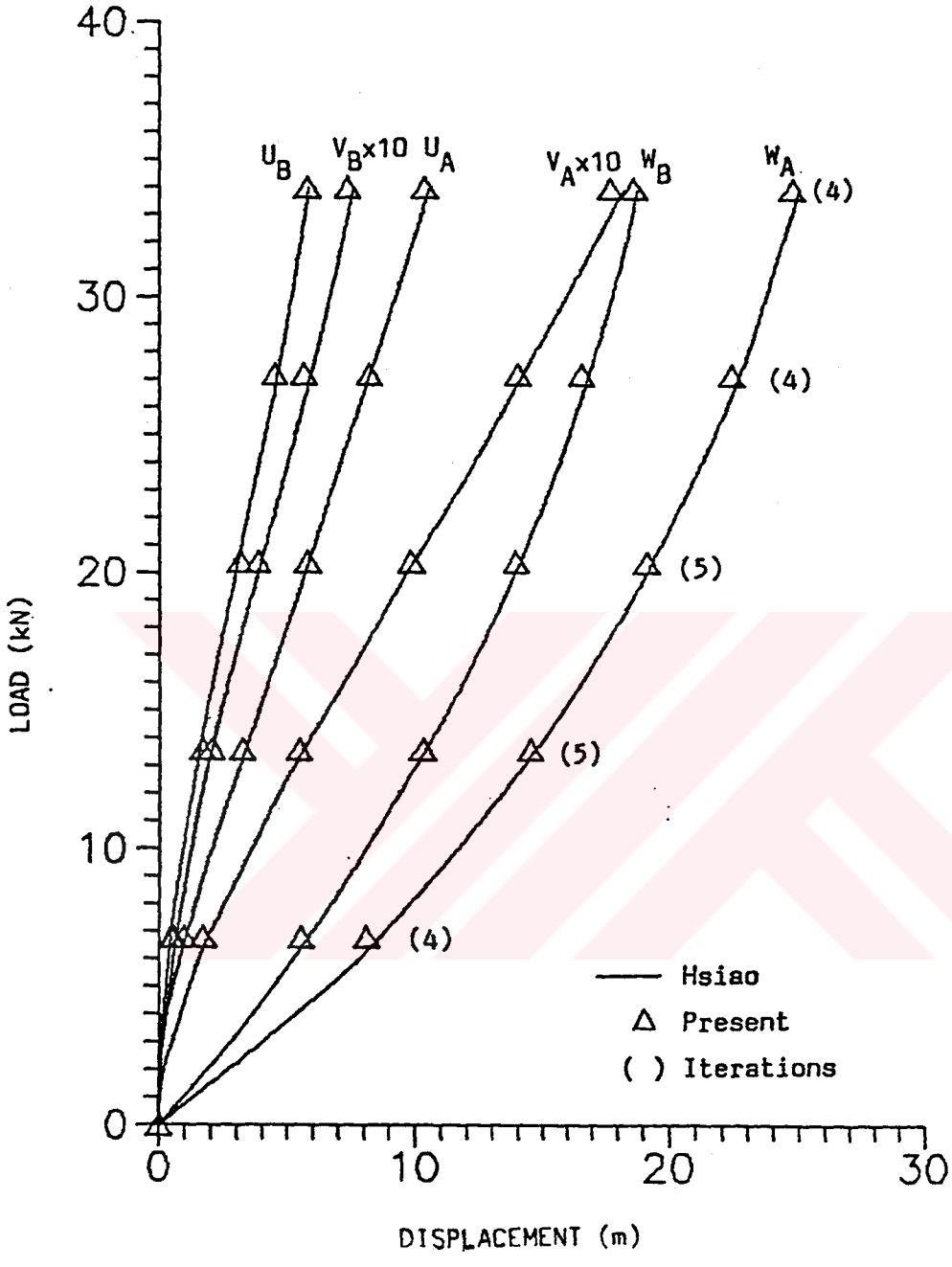


Figure 6.13. Load-Deflection Curve for the Cantilever Plate with Conservative End Load

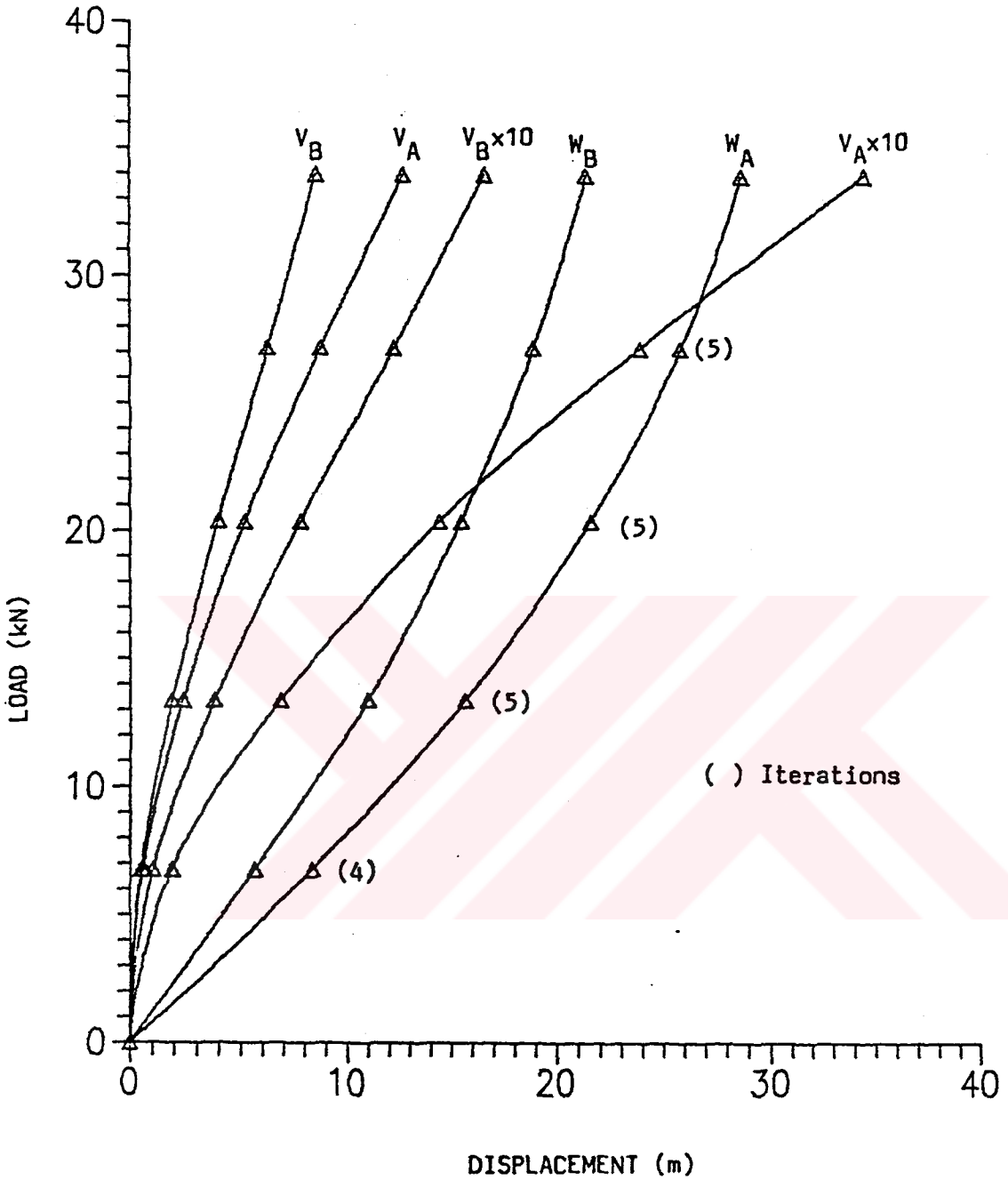
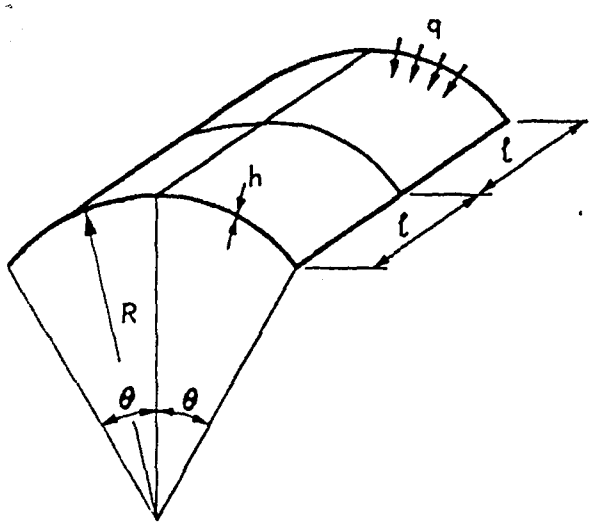


Figure 6.14. Load-Deflection Curve for the Cantilever Plate with Non-Conservative End Load

6.4 Shell Problems

6.4.1 Clamped Cylindrical Shell

A circular cylindrical shell as shown in Figure 6.15is considered.The shell is clamped along all four boundaries and subjected to uniform inward radial loading. This example has been extensively studied by a number of finite element investigators. Due to symmetry, only one-quarter of the panel has been discretized using a 3x3 mesh model. Five equal load increments have been used in this study. Fig 6.16 shows the final deformed configuration of the shell. The results are shown in Figure 6.17 with the solutions reported by Dhatt [45] and Sabir and Lock[46] using curved elements and by Horrigmoe and Bergan[14] with triangular elements. It is observed that the present results are in good agreement with the other results. The discrepancies at the higher load levels my be due to the large membrane forces that develop when the load is increased.



R = 2540 mm
ℓ = 254 mm
h = 3.175 mm
θ = 0.1 rad.
E = 3.10275 kN/mm^2
ν = 0.3

Figure 6.15. Clamped Cylindrical Shell Subjected to Inward Radial Loading

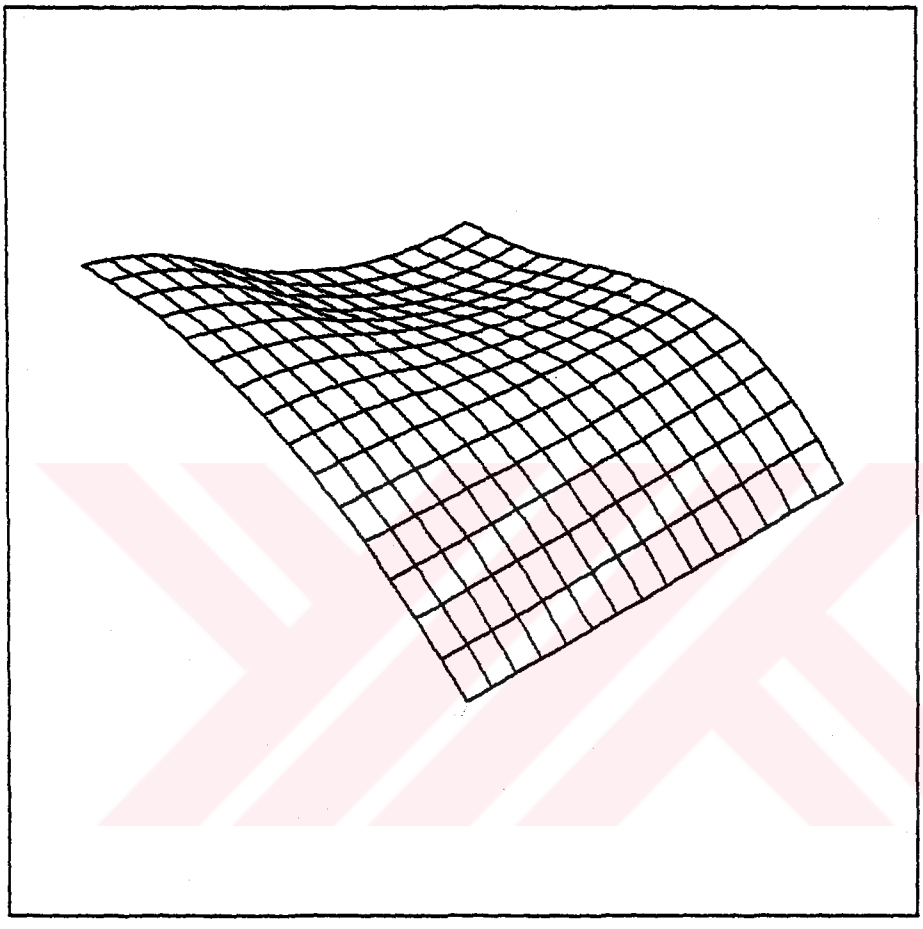


Figure 6.16. The Final Deformed Configuration of the Clamped Cylindrical Shell.

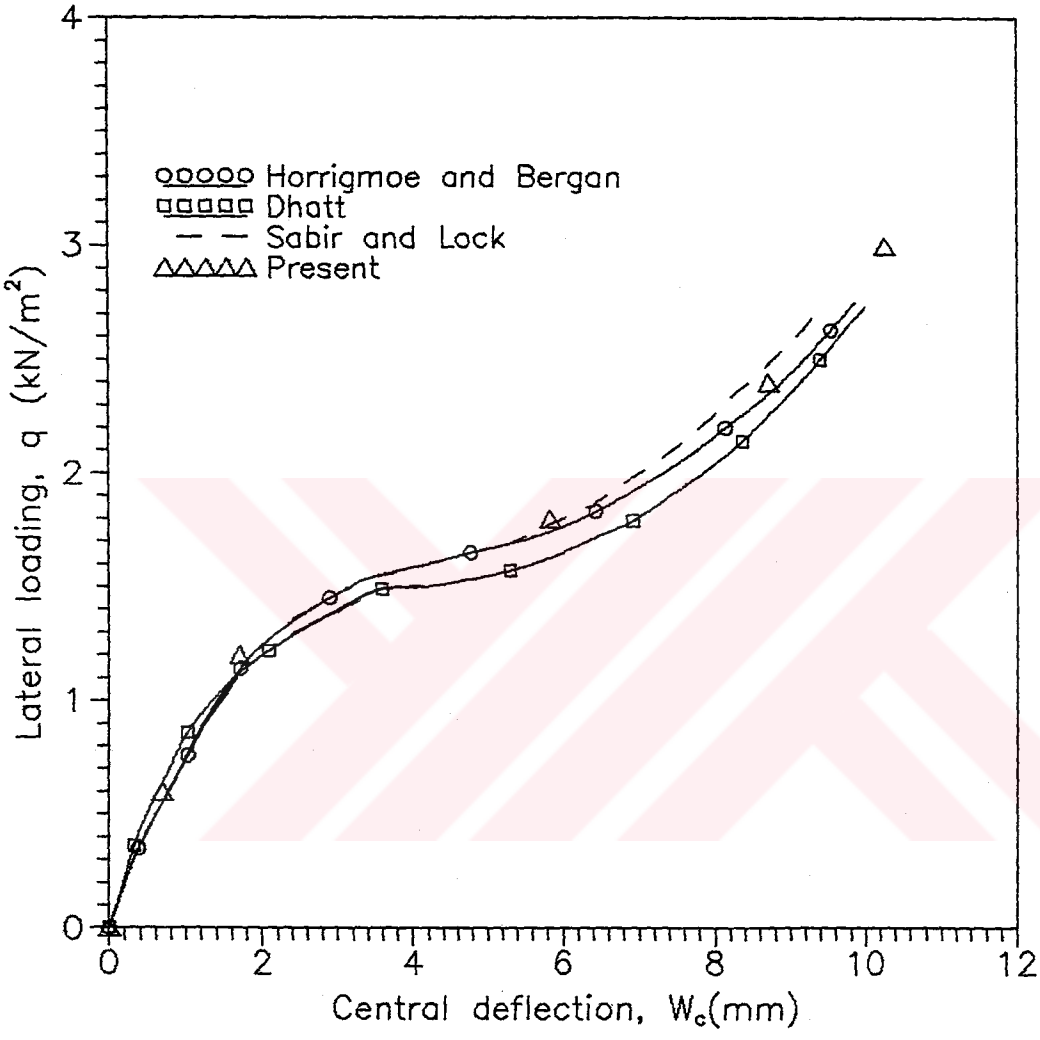


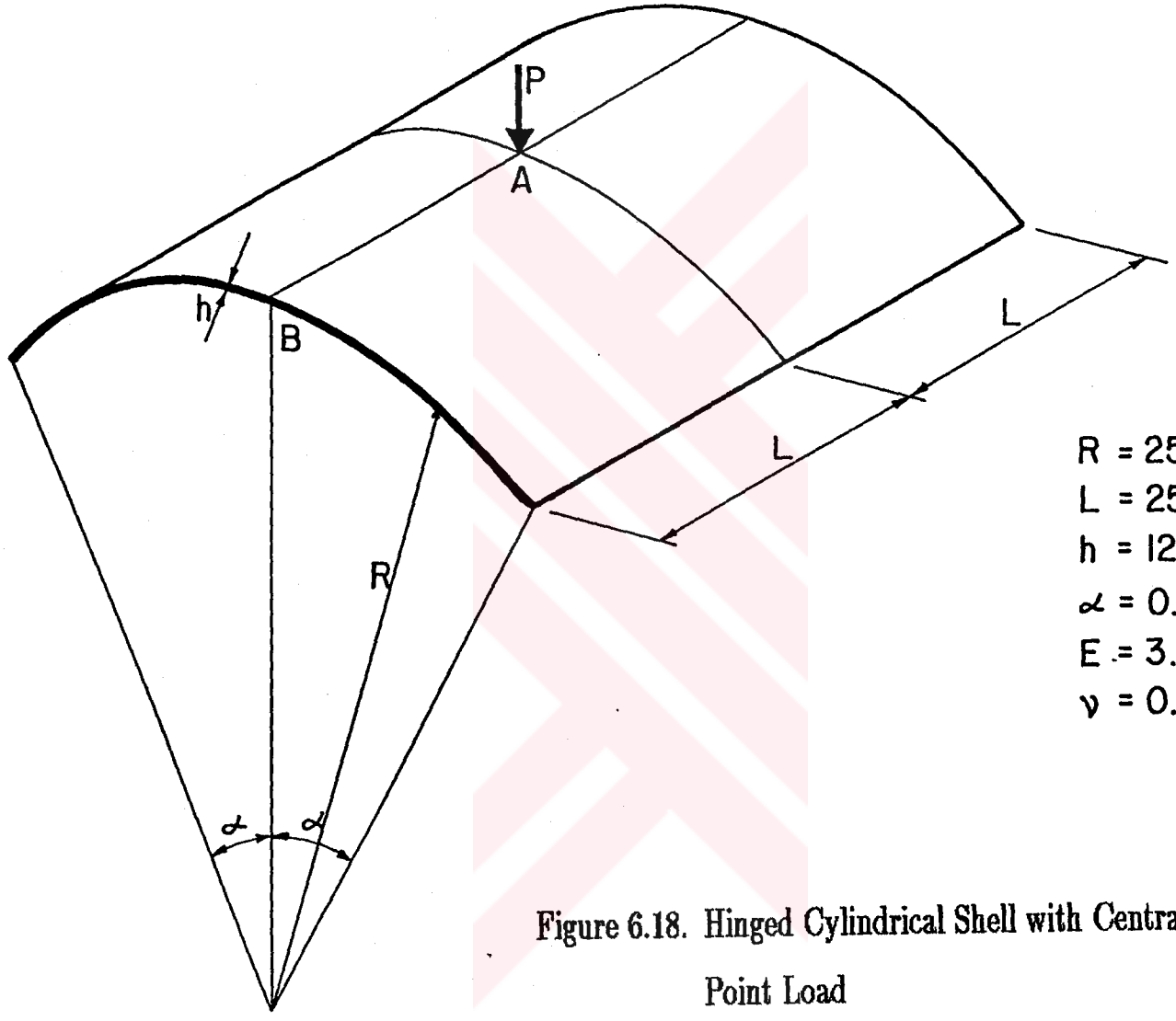
Figure 6.17. Load-deflection Curve for the Clamped Cylindrical Shell

6.4.2 Snap-Through of a Hinged Cylindrical Shell

The first example for snap-through analysis is a thin cylindrical shell which has hinged supports along the longitidunal edges and free along the curved ends. All dimensions and material properties are illustrated in Figure 6.18. Due to symmetry, the shell is modelled by a 3x3 mesh discretization. 5 levels of equal displacement increments for the vertical deflection at the crown(W_c) are used in this example. the final deformed configuration corresponding to $W_c = 300mm$ is shown in Figure 6.19. The load v.s. central deflections at the crown and at the middle of one end are shown in Figure 6.20. Also shown in Figure 6.20 are the results reported by Surana [48] and Oliver [49] using degenerated isoparametric elemets and by Hsiao [16] using facet shell elements. It is seen from Figure 6.20 that almost identical results are obtained.

6.4.3 Hinged Spherical Shell with Central Point Load

The final numerical example is the instability analysis of a hinged spherical shell subjected to a concentrated load at the crown as shown in Figure 6.21. All four edges are hinged and immovable. Due to the symmetry, only one-quarter of the shell is discretized by using a 5x5 mesh model. 5 levels of equal displacement increments upto 300 mm for the central deflection at the crawn are used in this analysis. The final deformed configuration corresponding to $W_c = 300mm$ is plotted in Figure 6.22. The load deflection curve of the vertical displacement component W_c at the crawn compared with the solutions by Horrigmoe and Bergan [14], Hsiao and Chen [16], Pica and Wood [47] and Surana [48]. In Figure 6.23 it is seen that all five solutions are agree rather well.



R = 2540 mm
L = 254 mm
h = 12.7 mm
α = 0.1 rad
E = 3.10275 kN/mm^2
ν = 0.3

Figure 6.18. Hinged Cylindrical Shell with Central Point Load

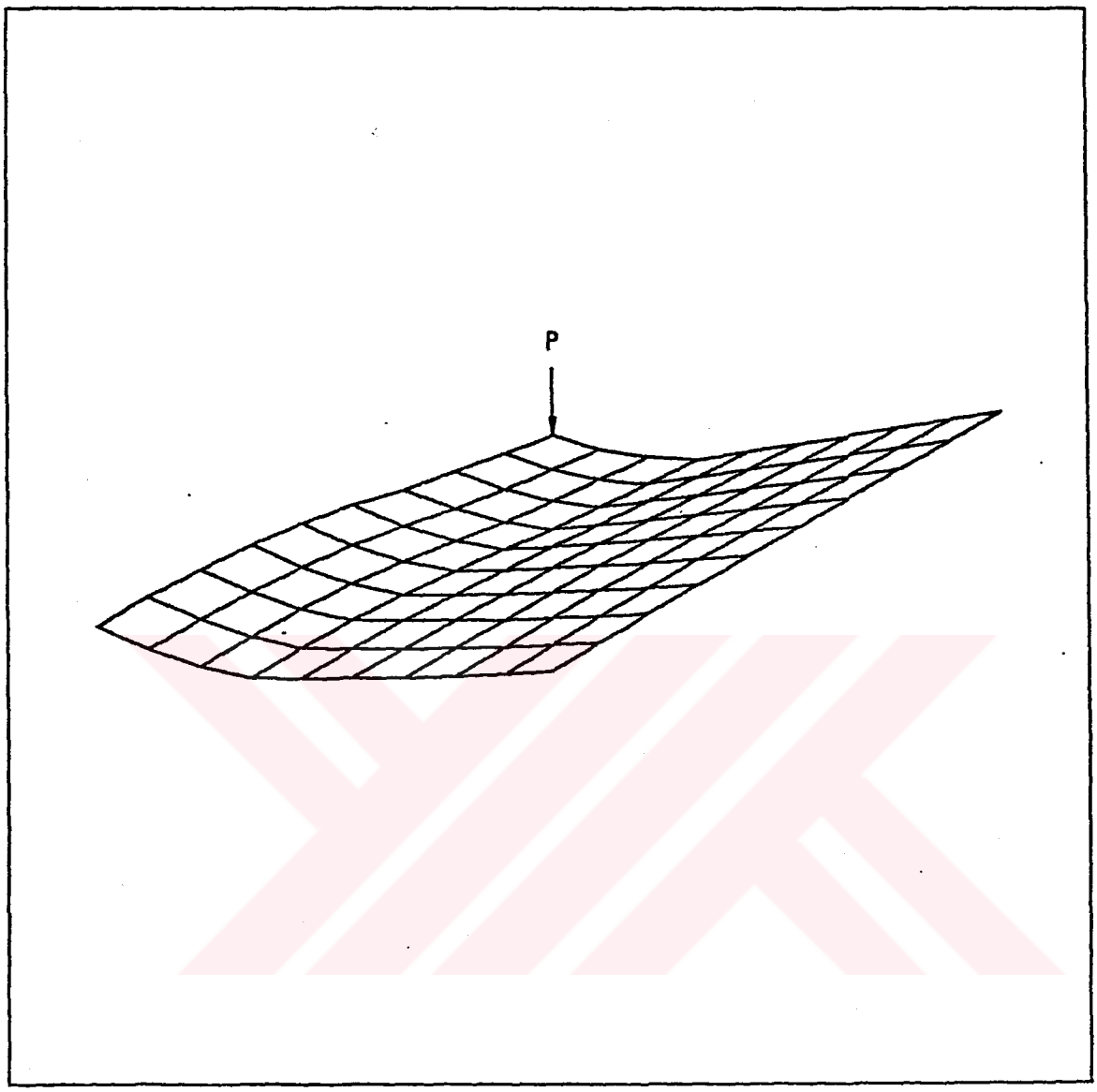


Figure 6.19. The Deformed Configuration of Hinged Cylindrical Shell Corresponding to $W_A = 300$ mm.

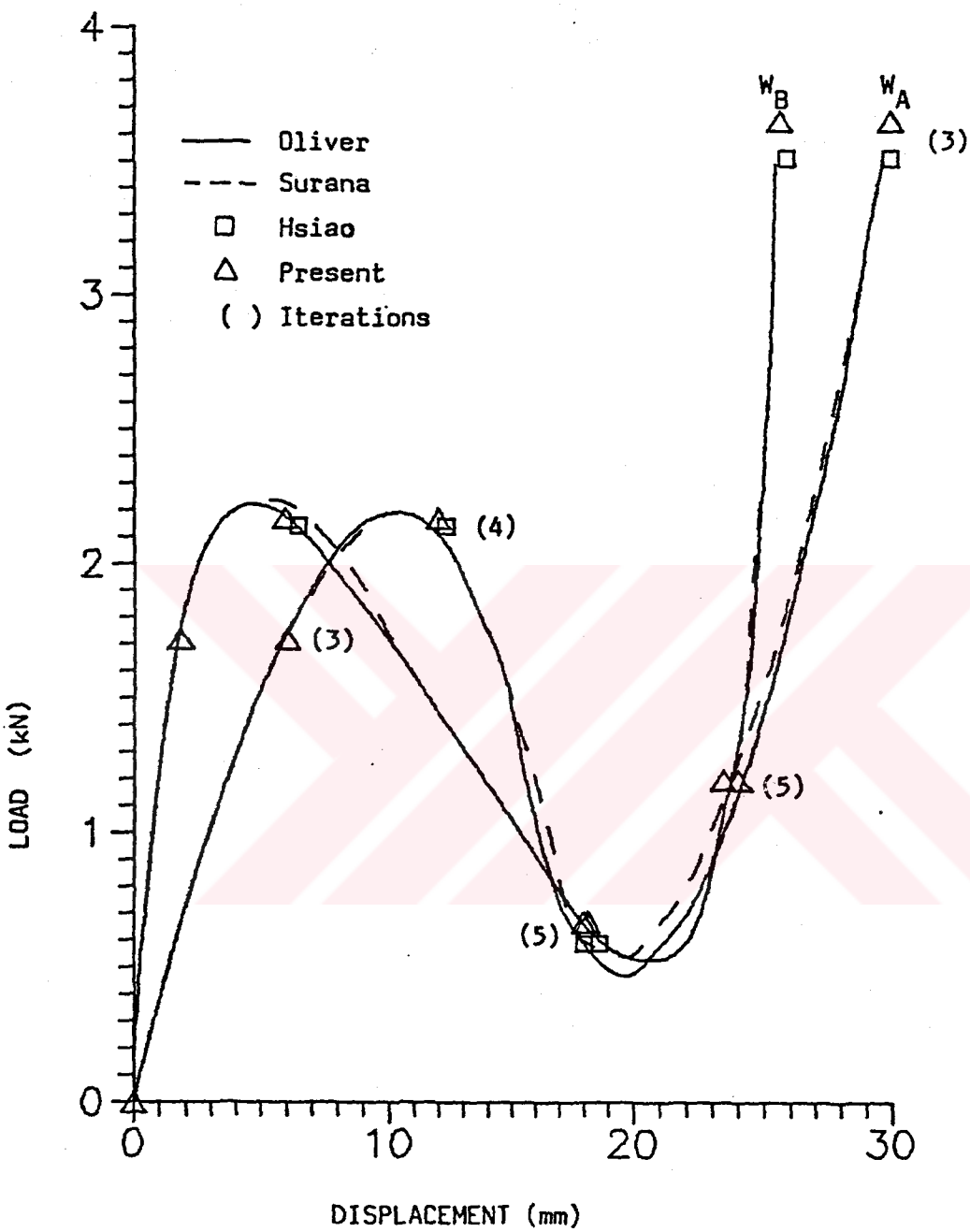
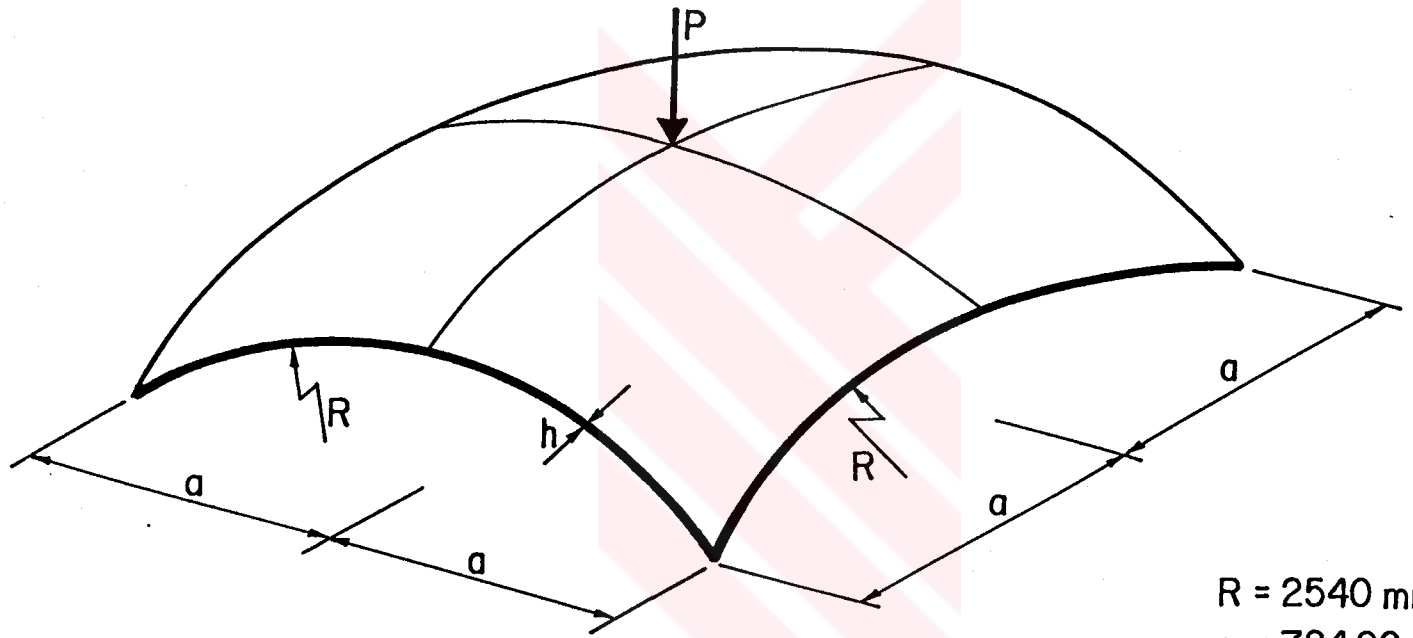


Figure 6.20. The Load v.s.Central Deflections at the Crown (W_A) and at the Middle of One End (W_B)



R = 2540 mm
a = 784.90 mm
h = 99.45 mm
E = 68.95 N / mm^2
ν = 0.3

Figure 6.21. Hinged Spherical Shell with Central Point Load

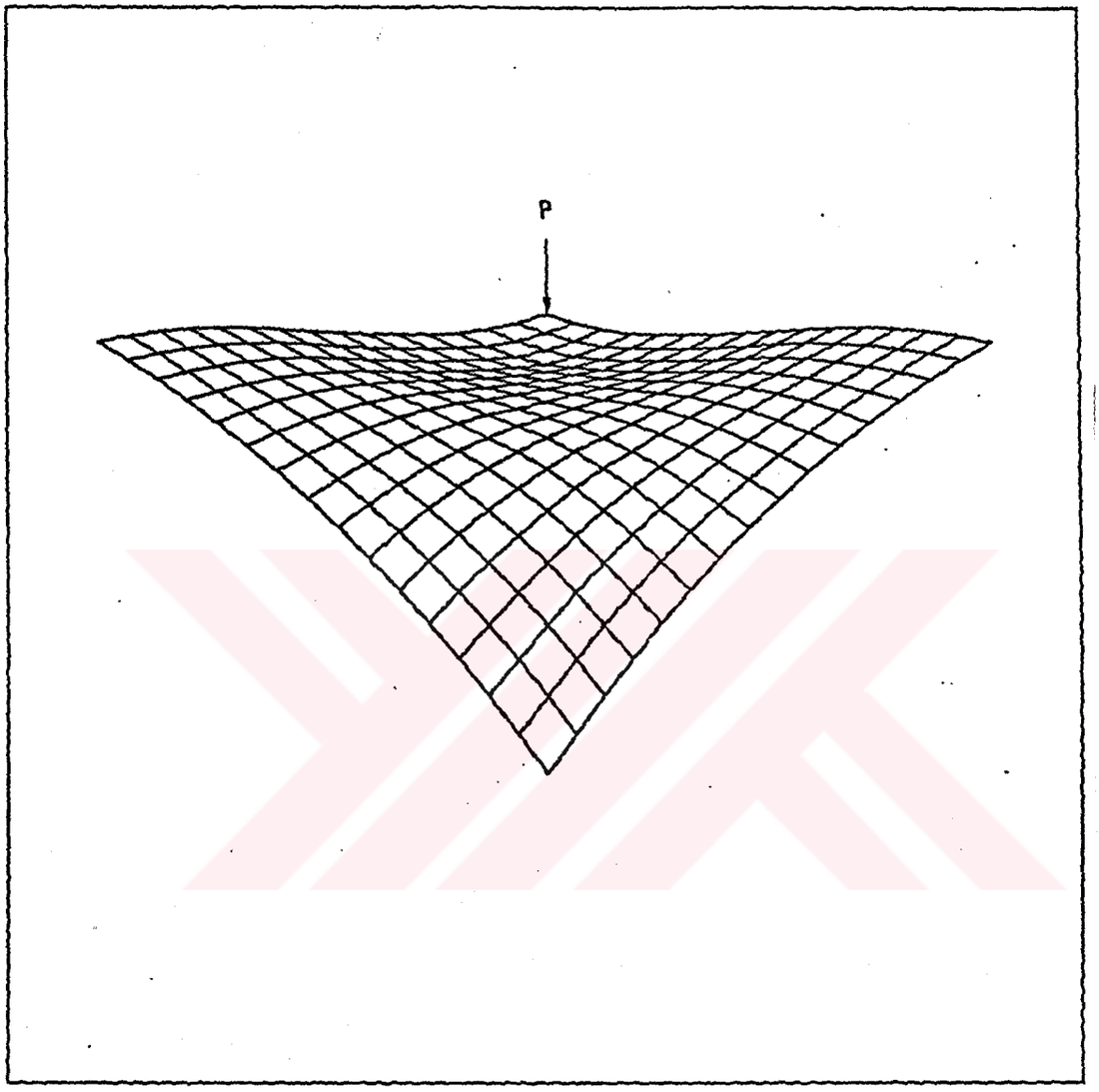


Figure 6.22. The Final Deformed Configuration of the Hinged Spherical Shell Corresponding to $W_C = 300$ mm.

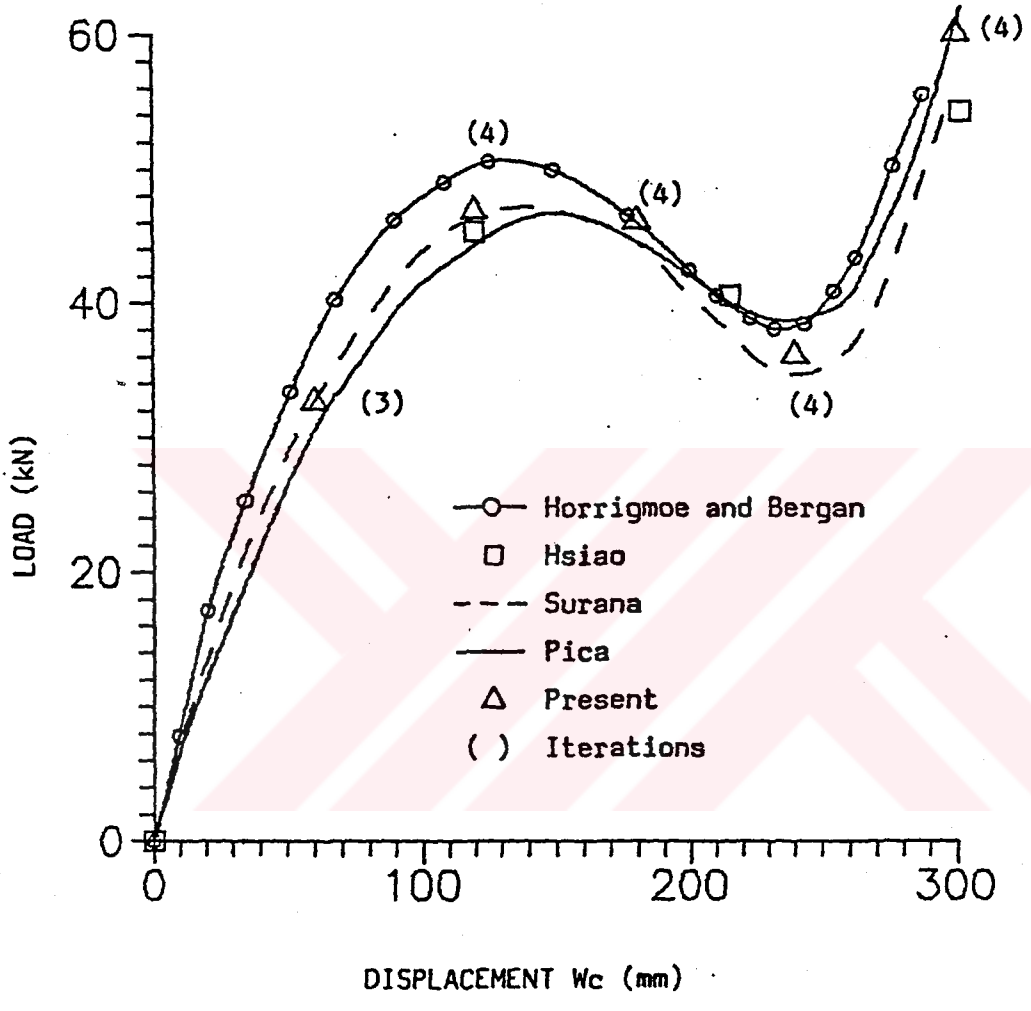


Figure 6.23. The Load-Deflection Curve for the Hinged Spherical Shell

CHAPTER 7

CONCLUSIONS

A three-node facet shell element which is the combination of the anisoparametric plate bending element of Tessler and Hughes [38] and the plane-stress element with vertex rotations developed by Allman [39] has been presented. The element has six d.o.f. per node. The nodal variables are three translations and three rotations of C^0 continuity type. An incremental updated Lagrangian formulation has been used. The vector operations have been applied to the rotations by subtracting the rigid-body part of the motion from the total displacements. Newton-Raphson algorithms for force and displacement incrementation have been used in the incremental iterative solutions. The singularity of the tangential stiffness matrix has been alleviated by displacement incrementation. Accurate results have been obtained for beam, plate and singly or doubly curved shells under conservative or non-conservative loads. The present element proves to be an alternative for large deflection and instability problems.

REFERENCES

1. D.W. Murrey and E.L.Wilson, Finite element large deflection analysis of plates, A.S.C.E,95(EM1),143-165 (1969).
2. P.G. Bergan and R.W.Clough, Large deflection analysis of plates and shallow shells using the finite element method, Int. J. Num. Meth. Eng, 5, 543-556 (1973).
3. A. Pica, R.D. Wood and E.Hinton, Finite element analysis of geometrically nonlinear plate behaviour using a Mindlin formulation, Comput. Structures, 11, 203-215 (1980).
4. T.C. Chang and K. Sawamiphakdi, Large deflection and postbuckling analysis of shell structures, Comp. Meth. Appl. Mech. Eng, 32, 311-326 (1982).
5. K.M. Hsiao and Y.R. Chen, Nonlinear analysis of shell structures by degenerated isoparametric shell element, Comput. Structures, 31, 427 - 438 (1989).
6. M. Fafard, G. Dhatt and L. Batoz, A new discrete Kirchhoff plate/shell element with updated procedures, Comput. Structures, 31, 591-606 (1989).
7. T.Y. Yang and Y.C. Wu, A geometrically nonlinear tensorial formulation of a skewed quadrilateral thin shell finite element, Int. J. Numer. Methods Eng., 29, 407-446 (1990).
8. A.F. Saaleb, T.Y. Yang, W. Graf and S. Yinggeungyong, A hybrid/mixed model for nonlinear shell analysis and its applications to large rotation

problems, Int. J. numer. meth. Eng., 28, 2855-2875 (1989).

9. J.T. Oden and T. Sato, Finite strains of elastic membranes by the finite element method. Int. J. Solids Struct., 3, 471 (1967).
10. S. Ahmad, B.M. Iron and O.C. Zienkiewicz, Analysis of thick and thin shell structures by curved finite elements. Int. J. numer. Meth. Eng., 2, 419-451 (1970).
11. K.M. Hsiao and H.C. Hung, large deflection analysis of shell structures by using corotational total lagrangian formulation, Comp. Meth. Appl. Mech. Eng.,73,209-225 (1989).
12. P.L. Boland and T.H.Pian, Large deflection analysis of thin elastic structures by the assumed stress hybrid finite element methods, Comput. Structures, 7,1-12 (1977).
13. J.H.Argyris, P.C. Dunne, G.A. Malejannakis and E. Schelkle, A simple triangular facet shell element with applications to linear and nonlinear equilibrium and elastic stability problems, Comp. Meth. Appl. Mech. Eng. 1, 10,371-403 (1977).
14. G. Horrigmoe and P.G. Bergan, Nonlinear analysis of free-form shells by flat finite elements, Comp. Meth. Appl. Mech. Eng.,16,11-35 (1978)
15. K.J. Bathe and L.W. Ho, A simple and effective element for analysis of general shell structures, Comput. Structures, 13,673-681 (1981).
16. K.M. Hsiao, Nonlinear analysis of general shell structures by flat triangular element, Comput. Structures, 25,665-675 (1987).

17. P.C. Dunne, Metodo dos elementos finitos, Instituto de Pesquisas Tecnologicas, Sao Paulo (1969).

18. P.G. Bergan, M.K. Nygard, Finite elements with increased freedom in choosing shape functions, Int. J. Numer. Methods. Eng.,20,643-663 (1984).

19. Y.C. Fung, Foundations of solid mechanics, Prentice-Hall (1965).

20. G. Wempner, Finite elements, finite rotations and small strains of flexible shells, Int. J. Solids Structures, 5, 117-153 (1969).

21. J. Argyris, An excursion into large rotations, Comp. Meth. Appl. Mech. Eng.,32,85-155 (1982).

22. R. Ramm, A plate /shell element for large deflections and rotations. In formulations and computational algorithms in finite element analysis, U.S. - Germany Symp., 264-293, Cambridge (1977).

23. K.J. Bathe and S. Bolourchi, A geometric and material nonlinear plate and shell element. Comput. Structures., 11,23-48 (1980).

24. T.J.R. Hughes and W.K. Liu, Nonlinear finite element analysis of shells : Part I. Three dimensional shells, Comp. Meth. Appl. Mech. Eng., 26, 331-362 (1981).

25. K.M. Hsiao, H.J. Hung and Y.R. Chen, A corrotational procedure that handles large rotations of spatial beam structures, Comput. Structures, 27, 769-781 (1987).

26. K.J. Bathe and S. Bolourchi, Large displacement analysis of three dimensional beam structures, Int. J. Numer. Meth. Eng., 14,961-986 (1979).

27. P.G. Bergan, G. Horrigmoe, B. Krakeland and T.H. Soleide, Solution techniques for nonlinear finite element problems, Int. J. Numer. Methods Eng., 12, 1677-1696 (1980).

28. P.G. Bergan, Solution algorithms for nonlinear problems, Comput Structures, 12,497-509 (1980).

29. E.W. Wright and E.H. Gaylord, Analysis of unbraced multistorey steel rigid frames, J.Struct. Div. A.S.C.E., 94, 1143-1163 (1968).

30. J.H. Argyris, Continua and discontinua, Proc. 1st Conf. Matrix methods Struct. Mech., Wright-Patterson A.F.B., Ohio, 11-189, (1965).

31. T.H.H. Pian and P.Tong, Variational formulation of finite displacement analysis, in B.F de Veubeke, High Speed Computing of Elastic Structures, University of Liegge, 43 - 63 (1971).

32. W.E. Haisler, J.A. Stricklin, Displacement incrementation in nonlinear structural analysis by the self correcting method, Int. J. Numer. Meth. Eng., 11,3-10 (1977).

33. K.J. Bathe et al, finite element formulations for large deformation dynamic analysis, Int. J. Numer. Meth. Eng., 9, 353-386 (1975).

34. K.J. Bathe and H. Ozdemir, Elastic-plastic large deformation static and dynamic analysis, Comput. Structures, 6, 81-92 (1976).

35. K.J. Bathe, Finite Element Procedures in Engineering Analysis, Prentice-Hall (1982).

36. R.D. Mindlin, Influence of rotary inertia and shear on flexural motions of isotropic elastic plates, J. Appl. Mech., 18, 31-38, (1951).

37. O.C. Zienkiewicz, The Finite Element Method, 3rd Ed., Mc Graw Hill Book Company (UK) Ltd. (1977).

38. A Tessler and T.J.R. Hughes, A three node Mindlin plate element with improved transverse shear, Comp. Meth. Appl. Mech., 50,71-101 (1985).

39. D.J. Allman, A compatible triangular element including vertex rotations for plane elasticity analysis, Comput. Structures, 19, 1-8 (1984).

40. D. Briassoulis, On the basics of the shear locking problem of C^0 isoparametric plate elements, Comput. Structures, 33, 169-185 (1989).

41. A. Tessler and T.J.R. Hughes, An improved treatment of transverse shear in the Mindlin type four node quadrilateral element, Comp. Meth. Appl. Mech. Eng., 39, 311-335 (1983).

42. M. Saje and S. Srpčič, Large deformations of in-plane beam, Int. J. Solids Structures, 21,1181-1195 (1985).

43. K. Mattiasson, Numerical results from large deflection beam and frame problems analysed by means of elliptic integrals, Int. J. Numer. Meth. Eng., 16,145-153 (1981).

44. S. Levy, bending of rectangular plates with large deflection. NACA Tech. Note 846 (1942).

45. G.S. Dhatt, Instability of thin shells by the finite element method, IASS Syposium for Folded Plates and Prismatic structures, Vienna (1970).

46. A.B. Sabir and A.C. Lock, The application of finite elements to the large deflection geometrically nonlinear behaviour of cylindrical shells in : C.A. Brebien and H. Tottenham(eds.),Variational methods in engineering, Southampton Univ-Press.,7/66 - 7/75 (1973).

47. A. Pica and R.D. Wood, Post-buckling behaviour of plates and shells using a Mindlin shallow shell formulation, Comput. Structures, 12, 759-768 (1980).

48. K.S. Surana, Geometrically nonlinear formulation for the curved shell elements, Int. J. Numer. Meth. Eng., 19,581-615 (1983).

49. J. Oliver and E. Onate, A total lagrangian formulation for the geometrically nonlinear analysis of structures using finite elements, Int. J. Numer. Meth. Eng. 1, 20,2253-2281 (1984).

50. K.S. Surana and R.M.Sorem, Geometrically nonlinear formulation for three dimensional curved beam elements with large rotations,Int .J .Numer. Meth. Eng., 28, 43-73 (1989)

APPENDIX A

USER'S MANUAL FOR PROGRAM NLFEM

A.1 Introduction

The program NLFEM performs the large deflection and instability analysis of plates and shells subjected to either conservative or nonconservative loading. The general structure of the program is shown in fig A.1. In this figure INPUT is a user created file which contains the type of incrementation, type of loading, nodal and element data, connectivities, restraints as well as nodal forces. DISOUT and CONOUT are output files created by execution of NLFEM. These files list the force-displacement relations of pre-selected nodes and the coordinates of deformed configurations corresponding to each increment.

Data preperation for INPUT is explained below. Typical input and output files for large deflection analysis of a sample problem are shown in APPENDIX-B.

A.2 Creation of the File INPUT

INPUT is the data file required to execute the program NLFEM. There format specification for the variables in each data block since they are free formatted. These variables will be explained in detail.

ISTR : Type of control strategy

(1) .....Force Control

(2) .....Displacement Control

ILOAD : Type of loading

(0) .....Conservative

(1) .....Non-conservative

NINC : Number of increments

TOL : Error tolerance to stop iterations

ND, ID : A critical node and its degree of freedom associated with the prescribed displacement component.

DTOT : A prescribed value for the displacement component required in the displacement incrementation strategy.

NNOD : Number of nodes to be studied

INOD : An array containing the node numbers selected to study their load-deflection relatios.

NOEL : Number of elements.

NODES : Number of nodes.

NEDOF : Number of d.o.f. per element.

NDOF : number of d.o.f. per node.

NPE : Number of nodes per element.

NELOD : Number of nodal forces.

NOD : Node number.

X,Y,Z : The global cartesien coordinates of a node.

NEL : Element number.

ICON : defines the element connectivity, i.e. the number of nodes belonging to that element. The sequence determines the number of element.

IDOF : shows the constrained degrees of freedom at a node.

0 : unconstrained.

1 : constrained.

for the order $U, V, W, \Theta_X, \Theta_Y, \Theta_Z$.

YM : Modulus of elasticity.

RN : Poissons ratio.

H : Thickness of the shell.

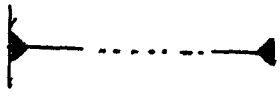
LG : An integer value indicating whether the forces will be given in local or global coordinates.

0 ..... local

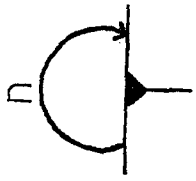
1 ..... global

F : force components of a node either in local or global coordinates.

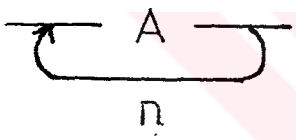
In the folloving, the syntax diagram of the file INPUT will be given. It is useful here to explain some of the important shapes used in the diagram.



: Start and end of a data block.



: The data block will be input n times.



: The variable will be input n times.

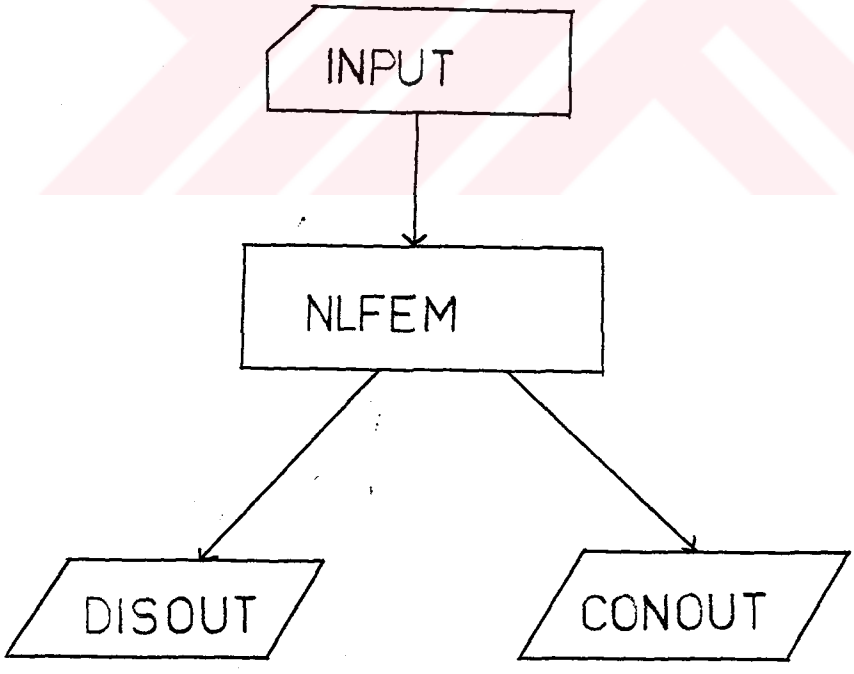


Figure A.1. The General Structure of the Program NLFEM

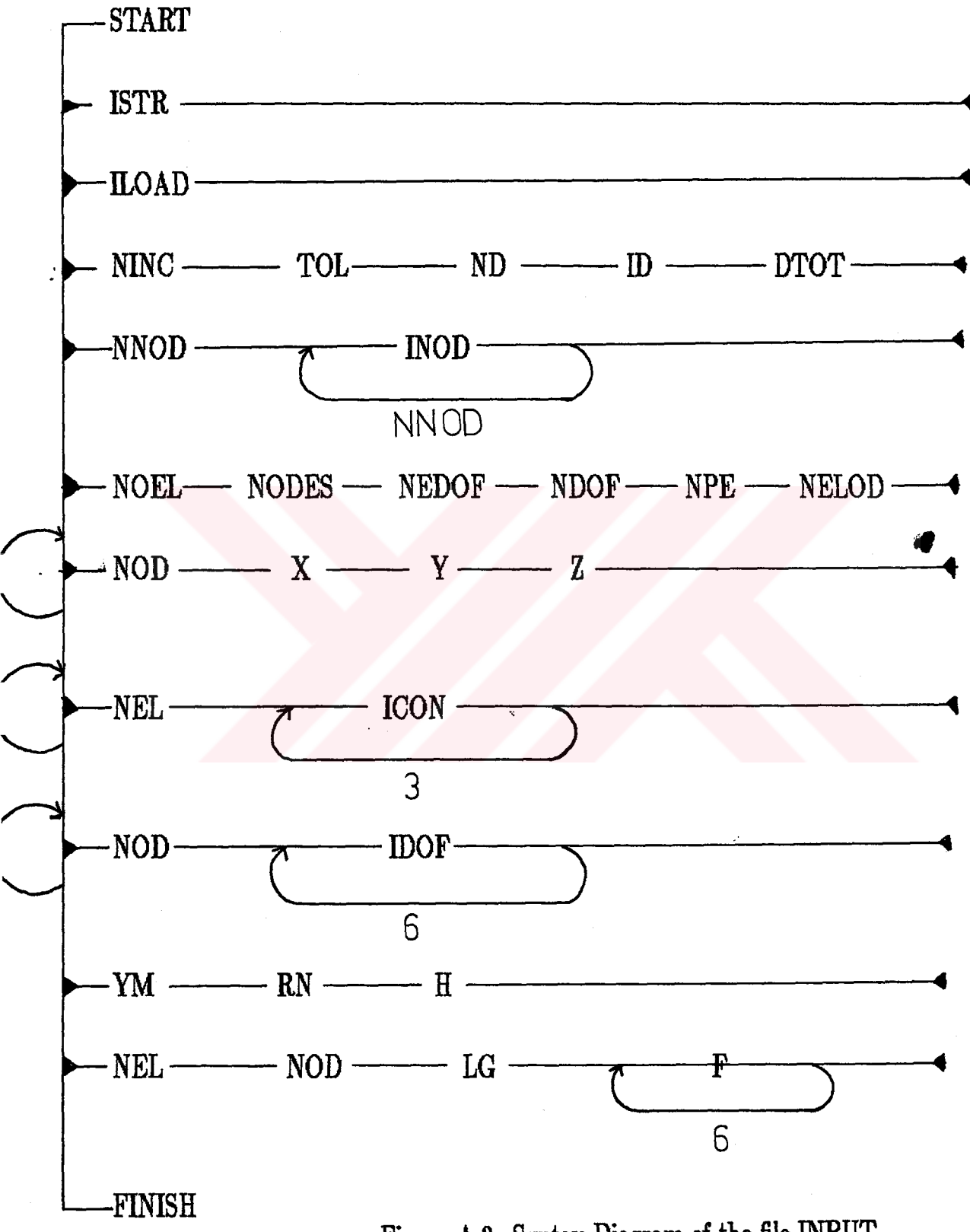


Figure A.2. Syntax Diagram of the file INPUT

APPENDIX B

A SAMPLE RUN

A diamond structure composed of four equal bars shown in fig B.1 is analysed as a sample problem. A finite element model of a quarter of the structure using six triangular elements is shown in fig B.2. The load is applied by incrementing displacements in 5 equal increments of 18m.The data file INPUT is given in fig B.3, the output files DISOUT and ONOUT are given in figures B.4 and B.5. Figures B.6 and B.7 show the deformed configurations at each dislplacement level.

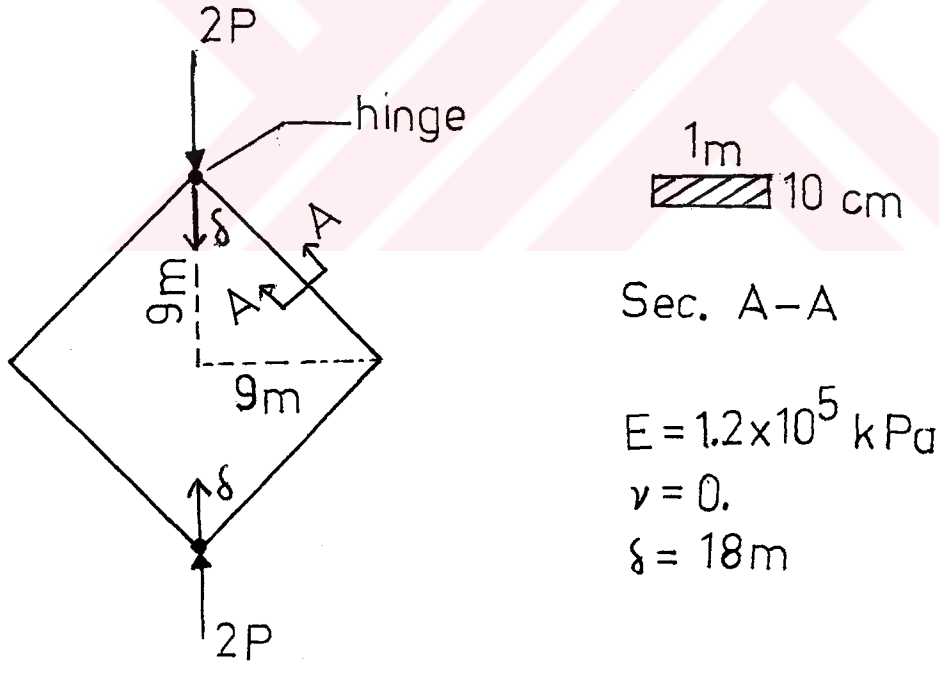


Figure B.1. Sample Problem for Large Deflection Analysis

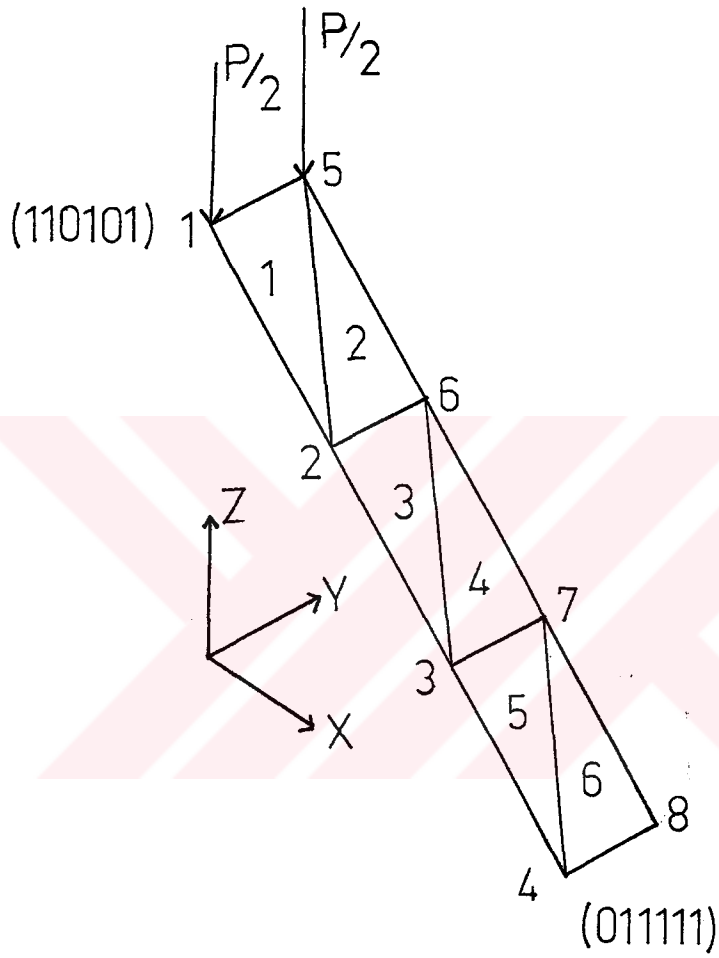


Figure B.2. Discretization and Boundary Conditions of the Sample Problem

```
2

0

5    0.005    1   3   -18.00

1    1

6    8    18   6   3   2

1    0.00     0.00     9.00
2    3.00     0.00     6.00
3    6.00     0.00     3.00
4    9.00     0.00     0.00
5    0.00     1.00     9.00
6    3.00     1.00     6.00
7    6.00     1.00     3.00
8    9.00     1.00     0.00

1    1   2   5
2    2   6   5
3    2   3   6
4    3   7   6
5    3   4   7
6    4   8   7

1     1 1 0 1 0 1
2     0 1 0 1 0 1
3     0 1 0 1 0 1
4     0 1 1 1 1 1
5     1 1 0 1 0 1
6     0 1 0 1 0 1
7     0 1 0 1 0 1
8     0 1 1 1 1 1

1.2E5    0.00    0.1

1   1   1   0.00    0.00    1.00   0.00   0.00    0.00
1   5   1   0.00    0.00    1.00   0.00   0.00    0.00
```

Figure B.3. INPUT for the Sample Problem

INCREMENT : 1
NUMBER OF ITERATIONS : 5
NODE NUMBER : 1

FORCE	DISPLACEMENT
0.00000000E+00	0.00000000E+00
0.00000000E+00	0.00000000E+00
-0.36664737E-01	-0.36000000E+01
0.00000000E+00	0.00000000E+00
0.00000000E+00	-0.53503787E+00
0.00000000E+00	0.00000000E+00

INCREMENT : 2
NUMBER OF ITERATIONS : 5
NODE NUMBER : 1

FORCE	DISPLACEMENT
0.00000000E+00	0.00000000E+00
0.00000000E+00	0.00000000E+00
-0.62274482E-01	-0.72000000E+01
0.00000000E+00	0.00000000E+00
0.00000000E+00	-0.98050118E+00
0.00000000E+00	0.00000000E+00

INCREMENT : 3
NUMBER OF ITERATIONS : 5
NODE NUMBER : 1

FORCE	DISPLACEMENT
0.00000000E+00	0.00000000E+00
0.00000000E+00	0.00000000E+00
-0.99212630E-01	-0.10800000E+02
0.00000000E+00	0.00000000E+00
0.00000000E+00	-0.14201780E+01
0.00000000E+00	0.00000000E+00

INCREMENT : 4
NUMBER OF ITERATIONS : 5
NODE NUMBER : 1

FORCE	DISPLACEMENT
0.00000000E+00	0.00000000E+00
0.00000000E+00	0.00000000E+00
-0.18307911E+00	-0.14400000E+02
0.00000000E+00	0.00000000E+00
0.00000000E+00	-0.18792454E+01
0.00000000E+00	0.00000000E+00

INCREMENT : 5
NUMBER OF ITERATIONS : 7
NODE NUMBER : 1

FORCE	DISPLACEMENT
0.00000000E+00	0.00000000E+00
0.00000000E+00	0.00000000E+00
-0.63779242E+00	-0.18000000E+02
0.00000000E+00	0.00000000E+00
0.00000000E+00	-0.23205227E+01
0.00000000E+00	0.00000000E+00

Figure B.4. DISOUT for the Sample Problem

```
1,  0.00000000E+00,  0.00000000E+00,  0.90000000E+01,
2,  0.30000000E+01,  0.00000000E+00,  0.60000000E+01,
3,  0.60000000E+01,  0.00000000E+00,  0.30000000E+01,
4,  0.90000000E+01,  0.00000000E+00,  0.00000000E+00,   D = 0.
5,  0.00000000E+00,  0.10000000E+01,  0.90000000E+01,   P = 0.
6,  0.30000000E+01,  0.10000000E+01,  0.60000000E+01,
7,  0.60000000E+01,  0.10000000E+01,  0.30000000E+01,
8,  0.90000000E+01,  0.10000000E+01,  0.00000000E+00,

1,  0.00000000E+00,  0.00000000E+00,  0.54000000E+01,
2,  0.40763763E+01,  0.00000000E+00,  0.42238111E+01,
3,  0.79722124E+01,  0.00000000E+00,  0.25438116E+01,
4,  0.11367662E+02,  0.00000000E+00,  0.00000000E+00,   D = 3.6 m
5,  0.00000000E+00,  0.10000000E+01,  0.53999818E+01,   P = 74 N
6,  0.40763781E+01,  0.10000000E+01,  0.42238159E+01,
7,  0.79722123E+01,  0.10000000E+01,  0.25438115E+01,
8,  0.11367662E+02,  0.10000000E+01,  0.00000000E+00,

1,  0.00000000E+00,  0.00000000E+00,  0.18000000E+01,
2,  0.41991307E+01,  0.00000000E+00,  0.24060893E+01,
3,  0.84293053E+01,  0.00000000E+00,  0.20811198E+01,
4,  0.12126436E+02,  0.00000000E+00,  0.00000000E+00,   D = 7.2 m
5,  0.00000000E+00,  0.10000000E+01,  0.17998911E+01,   P = 124 N
6,  0.41991303E+01,  0.10000000E+01,  0.24060910E+01,
7,  0.84293053E+01,  0.10000000E+01,  0.20811197E+01,
8,  0.12126436E+02,  0.10000000E+01,  0.00000000E+00,

1,  0.00000000E+00,  0.00000000E+00, -0.18000000E+01,
2,  0.35824354E+01,  0.00000000E+00,  0.47300703E+00,
3,  0.76827319E+01,  0.00000000E+00,  0.15628364E+01,
4,  0.11627015E+02,  0.00000000E+00,  0.00000000E+00,   D = 10.8 m
5,  0.00000000E+00,  0.10000000E+01, -0.18000098E+01,   P = 198 N
6,  0.35824342E+01,  0.10000000E+01,  0.47300932E+00,
7,  0.76827319E+01,  0.10000000E+01,  0.15628364E+01,
8,  0.11627015E+02,  0.10000000E+01,  0.00000000E+00,

1,  0.00000000E+00,  0.00000000E+00, -0.54000000E+01,
2,  0.22188103E+01,  0.00000000E+00, -0.17836548E+01,
3,  0.55615368E+01,  0.00000000E+00,  0.82917859E+00,
4,  0.97223480E+01,  0.00000000E+00,  0.00000000E+00,   D = 14.4
5,  0.00000000E+00,  0.10000000E+01, -0.54000026E+01,   P = 366 N
6,  0.22188088E+01,  0.10000000E+01, -0.17836538E+01,
7,  0.55615363E+01,  0.10000000E+01,  0.82917857E+00,
8,  0.97223480E+01,  0.10000000E+01,  0.00000000E+00,

1,  0.00000000E+00,  0.00000000E+00, -0.90000000E+01,
2,  0.26574322E+00,  0.00000000E+00, -0.47652137E+01,
3,  0.14906698E+01,  0.00000000E+00, -0.70274918E+00,
4,  0.56747778E+01,  0.00000000E+00,  0.00000000E+00,   D = 18.0 m
5,  0.00000000E+00,  0.10000000E+01, -0.90000002E+01,   P = 1.28 kN
6,  0.26575236E+00,  0.10000000E+01, -0.47652159E+01,
7,  0.14906643E+01,  0.10000000E+01, -0.70275365E+00,
8,  0.56747774E+01,  0.10000000E+01,  0.00000000E+00,
```

Figure B.5. CONOUT for the Sample Problem

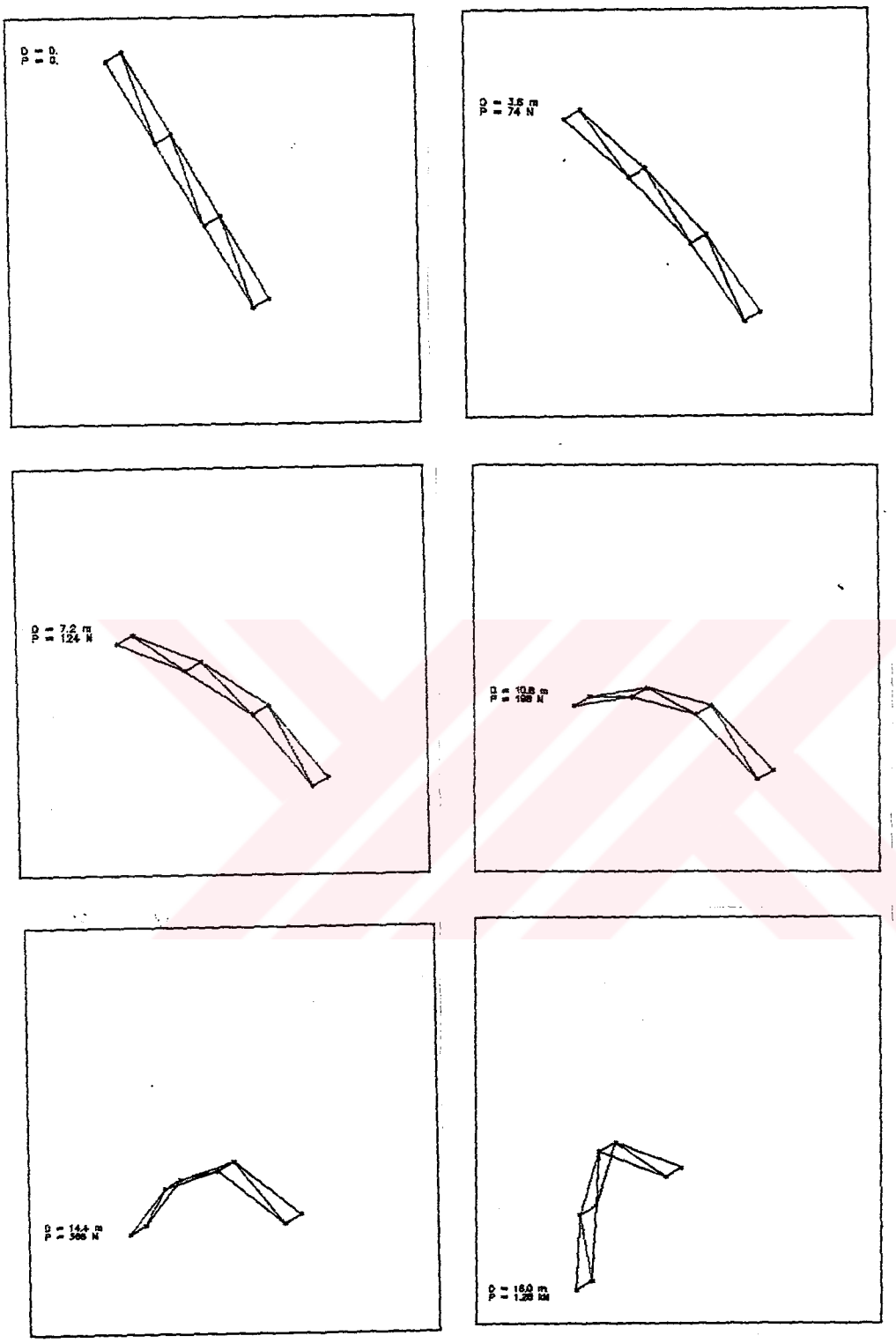


Figure B.6. Deformed Configurations at Each Displacement Level

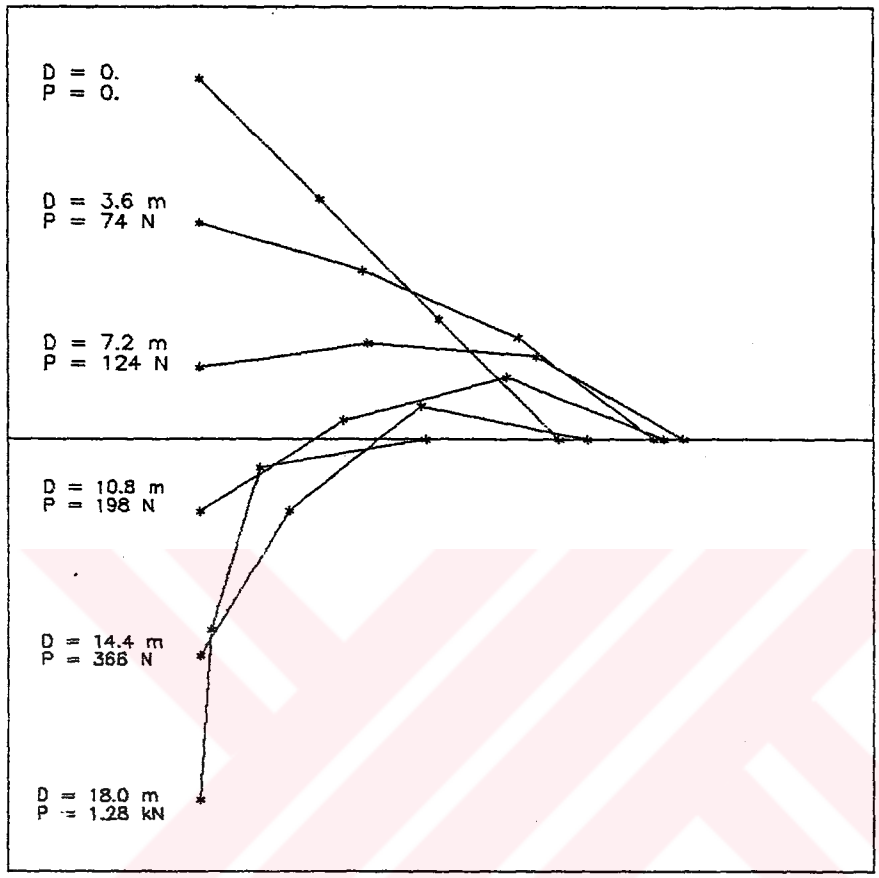


Figure B.7. Deformed Configurations from Another View

APPENDIX C

MATRICES DEFINED IN CHAPTER III

$$\mathbf{D} = \begin{bmatrix} \mathbf{D}_m & & \\ & \mathbf{D}_b & \\ & & \mathbf{D}_s \end{bmatrix}; \qquad \mathbf{D}_m = \frac{Eh}{1-\upsilon^2} \begin{bmatrix} 1 & \upsilon & 0 \\ & 1 & 0 \\ \text{sym.} & & 1/2\,(1-\upsilon) \end{bmatrix};$$

$$\mathbf{D}_b = \frac{h^2}{12}\mathbf{D}_m; \qquad \mathbf{D}_s = kGh \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

where E is the extensional modulus, G is the shear modulus, υ is the Poisson's ratio, h is the shell thickness and k is the classical shear correction factor.

$$B = \begin{bmatrix} \partial/\partial\,{}^1x & 0 & 0 & 0 & 0 \\ 0 & \partial/\partial\,{}^1y & 0 & 0 & 0 \\ \partial/\partial\,{}^1y & \partial/\partial\,{}^1x & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \partial/\partial\,{}^1x \\ 0 & 0 & 0 & \partial/\partial\,{}^1y & 0 \\ 0 & 0 & 0 & \partial/\partial\,{}^1x & \partial/\partial\,{}^1y \\ 0 & 0 & \partial/\partial\,{}^1x & 0 & 1 \\ 0 & 0 & \partial/\partial\,{}^1y & 1 & 0 \end{bmatrix}$$

$$G = \begin{bmatrix} \partial/\partial^1x & 0 & 0 & 0 & 0 \\ 0 & \partial/\partial^1x & 0 & 0 & 0 \\ 0 & 0 & \partial/\partial^1x & 0 & 0 \\ \partial/\partial^1y & 0 & 0 & 0 & 0 \\ 0 & \partial/\partial^1y & 0 & 0 & 0 \\ 0 & 0 & \partial/\partial^1y & 0 & 0 \end{bmatrix}$$

APPENDIX D

LISTING OF THE PROGRAM

```
      OPEN(1,FILE = 'INPUT')
      OPEN(2,FILE = 'CONOUT')
      OPEN(3,FILE = 'DISOUT')

      READ(1,*) OPT
      IF (OPT.EQ.1) CALL FORCE
      IF (OPT.EQ.2) CALL DIS
      STOP
      END


      SUBROUTINE ACTIVE(NEQ,NEQP)

      COMMON /ID/ IDOF(6,100)
      COMMON / IMFO/ NOEL,NODES,NEDOF,NDOF,NPE

      NEQ=0
      DO 10 N=1,NODES
      DO 20 I=1,NDOF
      IF (IDOF(I,N)) 30,40,30
40    NEQ=NEQ+1
      IDOF(I,N) = NEQ
      GOTO 20
30    IDOF(I,N)=0
20    CONTINUE
10    CONTINUE
      NEQP=NEQ+1
      RETURN
      END



      SUBROUTINE ADDRES(NEQ,NEQP,NWK)

      COMMON /MAX/ MAXA(450)
      COMMON /MHT/ MHT(450)
      COMMON / IMFO/ NOEL,NODES,NEDOF,NDOF,NPE

      DO 10 I=1,NEQP
10    MAXA(I)=0
      MAXA(1)=1
      MAXA(2)=2
```

```
      MK=0
      IF (NEQ.EQ.1) GOTO 20
      DO 30 I=2,NEQ
      IF (MHT(I).GT.MK) MK=MHT(I)
   30 MAXA(I+1) = MAXA(I) + MHT(I) +1
   20 MK=MK+1
      NWK=MAXA(NEQP) - 1
      RETURN
      END


      SUBROUTINE  ASMBLY (NEL)
      IMPLICIT REAL *8(A-H,O-Z)

      COMMON /ES/ ES(171)
      COMMON /S/  S(20000)
      COMMON /MAX/ MAXA(450)
      COMMON /LM/ LM(18)
      COMMON /IC/ ICON(150,3)
      COMMON /ID/ IDOF(6,100)
      COMMON / IMFO/ NOEL,NODES,NEDOF,NDOF,NPE

      DO 1 I=1,NEDOF
    1 LM(I) = 0
      DO 2 M=1,NPE
      I=ICON(NEL,M)
      MM=NDOF*(M-1) + 1
      MMM=MM+NDOF - 1
      JJ=0
      DO 3 J=MM,MMM
      JJ=JJ + 1
    3 LM(J)=IDOF(JJ,I)
    2 CONTINUE

      NDI=0
      DO 10 I=1,NEDOF
      II=LM(I)
      IF (II) 10,10,20
   20 MI=MAXA(II)
      KS=I
      DO 30 J=1,NEDOF
      JJ=LM(J)
      IF (JJ) 30,30,40
   40 IJ=II-JJ
      IF (IJ) 30,50,50
   50 KK=MI+IJ
      KSS=KS
      IF(J.GE.I) KSS=J+NDI
      S(KK)=S(KK)+ES(KSS)
   30 KS=KS+NEDOF-J
   10 NDI=NDI+NEDOF-I
      RETURN
      END
```

```
      SUBROUTINE COLHT(NEQ,NEL)

      COMMON /LM/ LM(18)
      COMMON /ID/ IDOF(6,100)
      COMMON /IC/ ICON(150,3)
      COMMON /MHT/ MHT(450)
      COMMON / IMFO/ NOEL,NODES,NEDOF,NDOF,NPE

      DO 10 I=1,NEDOF
   10 LM(I) = 0
      DO 20 M=1,NPE
      I=ICON(NEL,M)
      MM=NDOF*(M-1) + 1
      MMM=MM+NDOF - 1
      JJ=0
      DO 30 J=MM,MMM
      JJ=JJ + 1
   30 LM(J)=IDOF(JJ,I)
   20 CONTINUE
      DO 40 J=1,NEDOF
      IF (LM(J).EQ.0) GOTO 40
      DO 50 K=1,NEDOF
      IF (LM(K).EQ.0) GOTO 50
      IF (K.EQ.J) GOTO 50
      IF (LM(K).LT.LM(J)) GOTO 40
   50 CONTINUE
      MIN=LM(J)
      GOTO 90
   40 CONTINUE
      RETURN
   90 DO 70 J=1,NEDOF
      NNEQ = LM(J)
      IF (NNEQ.EQ.0) GOTO 70
      IF(MHT(NNEQ).LE.(NNEQ-MIN)) MHT(NNEQ) = NNEQ-MIN
   70 CONTINUE
      RETURN
      END



      SUBROUTINE DECOMP(NEQ)
      IMPLICIT REAL *8(A-H,O-Z)

      COMMON /S/ S(20000)
      COMMON /MAX/ MAXA(450)
      COMMON / IMFO/ NOEL,NODES,NEDOF,NDOF,NPE

      DO 10 N=1,NEQ
      KN=MAXA(N)
      KL=KN+1
      KU=MAXA(N+1)-1
```

```
      KH=KU-KL
      IF (KH) 20,30,40
   40 K=N-KH
      IC=0
      KLT=KU
      DO 50 J=1,KH
      IC=IC+1
      KLT=KLT-1
      KI=MAXA(K)
      ND=MAXA(K+1)-KI-1
      IF (ND) 50,50,60
   60 KK=MINO(IC,ND)
      C=0
      DO 70 L=1,KK
   70 C=C+S(KI+L)*S(KLT+L)
      S(KLT)=S(KLT)-C
   50 K=K+1
   30 K = N
      B=0
      DO 80 KK=KL,KU
      K=K-1
      KI = MAXA(K)
      C = S(KK)/S(KI)
      IF (ABS(C).LT.1E07) GOTO 90
      STOP
   90 B=B+C*S(KK)
   80 S(KK)=C
      S(KN)=S(KN)-B
   20 IF (S(KN)) 110,110,10
  110 WRITE(6,*) ' MATRIX NOT POSITIVE DEFINITE '
      STOP
   10 CONTINUE
      RETURN
      END



      SUBROUTINE REDBAK(NEQ)
      IMPLICIT REAL *8(A-H,O-Z)

      COMMON /S/ S(20000)
      COMMON /MAX/ MAXA(450)
      COMMON /V/ V(450)
      COMMON / IMFO/ NOEL,NODES,NEDOF,NDOF,NPE

      DO 10 N=1,NEQ
      KL=MAXA(N)+1
      KU=MAXA(N+1)-1
      IF (KU-KL) 10,20,20
   20 K=N
```

```
      C=0
      DO 30 KK=KL,KU
      K=K-1
   30 C=C+S(KK)*V(K)
      V(N)=V(N)-C
   10 CONTINUE

      DO 40 N=1,NEQ
      K=MAXA(N)
   40 V(N)=V(N)/S(K)

      N=NEQ
      DO 50 L=2,NEQ
      KL=MAXA(N)+1
      KU=MAXA(N+1)-1
      IF (KU-KL) 50,60,60
   60 K=N
      DO 70 KK=KL,KU
      K=K-1
   70 V(K)=V(K)-S(KK)*V(N)
   50 N=N-1
      RETURN
      END



      SUBROUTINE  UDLOAD (PP)
      IMPLICIT REAL *8(A-H,O-Z)

      COMMON / IMFO/ NOEL,NODES,NEDOF,NDOF,NPE
      COMMON / F/ FT(100,6),FI2(100,6),FIT(100,6)
      DO 10 I = 1,NODES
      DO 10 J = 1,NDOF
      FI2(I,J) = PP*FT(I,J)
   10 CONTINUE
      RETURN
      END


      SUBROUTINE  SETLD
      IMPLICIT REAL *8(A-H,O-Z)

      COMMON / IMFO/ NOEL,NODES,NEDOF,NDOF,NPE
      COMMON / V/ V(450)
      COMMON / F/ FT(100,6),FI2(100,6),FIT(100,6)
      COMMON / ID/ IDOF(6,100)
      DO 10 I = 1,NODES
      DO 10 J = 1,NDOF
      N = IDOF(J,I)
      IF (N.EQ.0) GOTO 10
      V(N) = FI2(I,J) - FIT(I,J)
   10 CONTINUE
```

```
      RETURN
      END



      SUBROUTINE  LLOAD (NEL)
      IMPLICIT REAL *8(A-H,O-Z)

      COMMON / IC/ ICON(150,3)
      COMMON / COOR/  X(100,3),Y(100,3),Z(100,3)
      COMMON / T/  T(3,3)
      COMMON / ESL/ ESL(18,18),GS(18,18),EL(18,18)
      COMMON / LF/ F(18)
      COMMON / D/ DIT(100,6),DT(100,6),D(18),ROT(150,3,3)
      COMMON / IMFO/ NOEL,NODES,NEDOF,NDOF,NPE

      DO 5 I = 1,18
      D(I) = 0.00
    5 CONTINUE
      I = ICON(NEL,1)
      J = ICON(NEL,2)
      K = ICON(NEL,3)

      CALL DIRCOS (NEL,1,T11,T12,T13,T21,T22,T23,T31,T32,T33)
      CALL DIRCOS (NEL,2,C11,C12,C13,C21,C22,C23,C31,C32,C33)
      CALL DIRCOS (NEL,3,S11,S12,S13,S21,S22,S23,S31,S32,S33)
      XS21 = X(J,3) - X(I,3)
      YS21 = Y(J,3) - Y(I,3)
      ZS21 = Z(J,3) - Z(I,3)

      XS31 = X(K,3) - X(I,3)
      YS31 = Y(K,3) - Y(I,3)
      ZS31 = Z(K,3) - Z(I,3)

      XT21 = X(J,1) - X(I,1)
      YT21 = Y(J,1) - Y(I,1)
      ZT21 = Z(J,1) - Z(I,1)

      XT31 = X(K,1) - X(I,1)
      YT31 = Y(K,1) - Y(I,1)
      ZT31 = Z(K,1) - Z(I,1)

      VS21 = SQRT(XS21*XS21 + YS21*YS21 + ZS21*ZS21)
      VT21 = SQRT(XT21*XT21 + YT21*YT21 + ZT21*ZT21)

      VSX = S11*XS31 + S12*YS31 + S13*ZS31
      VSY = S21*XS31 + S22*YS31 + S23*ZS31

      VTX = T11*XT31 + T12*YT31 + T13*ZT31
      VTY = T21*XT31 + T22*YT31 + T23*ZT31
```

```
U2X = VS21 - VT21
U3X = VSX  - VTX
U3Y = VSY  - VTY

D( 1) = 0.00
D( 2) = 0.00
D( 3) = 0.00
D( 7) = U2X
D( 8) = 0.00
D( 9) = 0.00
D(13) = U3X
D(14) = U3Y
D(15) = 0.00

AL1 = C32*S33 - S32*C33
AL2 = S31*C33 - C31*S33
AL3 = C31*S32 - S31*C32

AL = SQRT(AL1*AL1 + AL2*AL2 + AL3*AL3)
ALF = C31*S31 + C32*S32 + C33*S33
IF(ALF.GT.1) ALF = 1.
ALF = ACOS(ALF)
IF( AL.LT.1D-20) THEN
     AL1 = 0.00
     AL2 = 0.00
     AL3 = 0.00
ELSE
     ALL = 1./AL
     AL1 = AL1*ALL
     AL2 = AL2*ALL
     AL3 = AL3*ALL
ENDIF
CALL ROTAT(ALF,AL1,AL2,AL3,C11,C12,C13,E11,E12,E13)
CALL ROTAT(ALF,AL1,AL2,AL3,C21,C22,C23,E21,E22,E23)
CALL ROTAT(ALF,AL1,AL2,AL3,C31,C32,C33,E31,E32,E33)

BN1 = E12*S13 - S12*E13
BN2 = S11*E13 - E11*S13
BN3 = E11*S12 - S11*E12
BN = SQRT(BN1*BN1 + BN2*BN2 + BN3*BN3)
IF (BN.LT.1D-20) THEN
    BET1 = 0.00
    BET2 = 0.00
    BET3 = 0.00
ELSE
    BET1 = BN1/BN
    BET2 = BN2/BN
    BET3 = BN3/BN
ENDIF
BET = E11*S11 + E12*S12 + E13*S13
IF (BET.GT.1.) BET = 1.
BET = ACOS(BET)
```

```
DO 10 I = 1,NPE

J = ICON(NEL,I)

TJ1 = DIT(J,4)
TJ2 = DIT(J,5)
TJ3 = DIT(J,6)

TPL1 = E11*TJ1 + E12*TJ2 + E13*TJ3
TPL2 = E21*TJ1 + E22*TJ2 + E23*TJ3
TPL3 = E31*TJ1 + E32*TJ2 + E33*TJ3

TP1 = E11*TPL1 + E21*TPL2
TP2 = E12*TPL1 + E22*TPL2

TP3 = E13*TPL1 + E23*TPL2

TP = SQRT(TP1*TP1 + TP2*TP2 + TP3*TP3)
TPP = SQRT(TPL1*TPL1 + TPL2*TPL2)
IF (TP.LT.1D-20) THEN
    TP1 = 0.00
    TP2 = 0.00
    TP3 = 0.00
ELSE
    TP1 = TP1/TP
    TP2 = TP2/TP
    TP3 = TP3/TP
ENDIF


DNJ1 = ROT(NEL,I,1)
DNJ2 = ROT(NEL,I,2)
DNJ3 = ROT(NEL,I,3)

CALL ROTAT(TP,TP1,TP2,TP3,DNJ1,DNJ2,DNJ3,DNP1,DNP2,DNP3)
CALL ROTAT(BET,BET1,BET2,BET3,DNP1,DNP2,DNP3,DNJ1,DNJ2,DNJ3)


ROT(NEL,I,1) = DNJ1
ROT(NEL,I,2) = DNJ2
ROT(NEL,I,3) = DNJ3

SDN1 = S32*DNJ3 - DNJ2*S33
SDN2 = DNJ1*S33 - S31*DNJ3
SDN3 = S31*DNJ2 - DNJ1*S32

SDN = SQRT(SDN1*SDN1 + SDN2*SDN2 + SDN3*SDN3)
SDNN = S31*DNJ1 + S32*DNJ2 + S33*DNJ3
IF (SDNN.GT.1) SDNN = 1.
IF (SDN.LT.1D-20) THEN
   TET = 0.00
```

```
      ELSE
         TET = ACOS(SDNN)/SDN
      ENDIF

      RX = TET*SDN1
      RY = TET*SDN2
      RZ = TET*SDN3

      RLX = S11*RX + S12*RY + S13*RZ
      RLY = S21*RX + S22*RY + S23*RZ
      RLZ = S31*RX + S32*RY + S33*RZ

      N = 6*(I - 1) + 3

      D(N+1) = RLX
      D(N+2) = RLY
   10 CONTINUE

      DO 50 I = 1,NEDOF
      F(I) = 0.00
      DO 50 J = 1,NEDOF
      F(I) = F(I) + EL(I,J)*D(J)
   50 CONTINUE
      RETURN
      END



      SUBROUTINE SLOAD (NEL)
      IMPLICIT REAL *8(A-H,O-Z)

      COMMON / F/ FT(100,6),FI2(100,6),FIT(100,6)
      COMMON / LF/ F(18)
      COMMON /T/  T(3,3)
      COMMON / IC/ ICON(150,3)

      CALL DIRCOS(NEL,3,S11,S12,S13,S21,S22,S23,S31,S32,S33)
      DO 10 I = 1,3

      N = 6*(I - 1)
      J = ICON(NEL,I)

      FIT(J,1) = FIT(J,1) + S11*F(N+1) + S21*F(N+2) + S31*F(N+3)
      FIT(J,2) = FIT(J,2) + S12*F(N+1) + S22*F(N+2) + S32*F(N+3)
      FIT(J,3) = FIT(J,3) + S13*F(N+1) + S23*F(N+2) + S33*F(N+3)
      FIT(J,4) = FIT(J,4) + S11*F(N+4) + S21*F(N+5) + S31*F(N+6)
      FIT(J,5) = FIT(J,5) + S12*F(N+4) + S22*F(N+5) + S32*F(N+6)
      FIT(J,6) = FIT(J,6) + S13*F(N+4) + S23*F(N+5) + S33*F(N+6)
   10 CONTINUE

      RETURN
      END
```

```
SUBROUTINE  ROTAT(TET,T1,T2,T3,R1,R2,R3,P1,P2,P3)
IMPLICIT REAL *8(A-H,O-Z)

TR = T1*R1 + T2*R2 + T3*R3

TR1 = T2*R3 - R2*T3
TR2 = R1*T3 - T1*R3
TR3 = T1*R2 - R1*T2

CTET = COS(TET)
STET = SIN(TET)
DN = (1 - CTET)*TR

P1 = CTET*R1 + DN*T1 + TR1*STET
P2 = CTET*R2 + DN*T2 + TR2*STET
P3 = CTET*R3 + DN*T3 + TR3*STET

RETURN
END
SUBROUTINE DIS
IMPLICIT REAL *8(A-H,O-Z)

COMMON / ES/ ES(171)
COMMON / S/ S(20000)
COMMON / MAX/ MAXA(450)
COMMON / LM/ LM(18)
COMMON / COOR/ X(100,3),Y(100,3),Z(100,3)
COMMON / MP/ YM,RN,H
COMMON /IDEN/ IDC,IDGS
COMMON / ID/ IDOF(6,100)
COMMON / IC/ ICON(150,3)
COMMON / MHT/ MHT(450)
COMMON / V/ V(450)
COMMON / F/ FT(100,6),FI2(100,6),FIT(100,6)
COMMON / ESL/ ESL(18,18),GS(18,18),EL(18,18)
COMMON / D/ DIT(100,6),DT(100,6),D(18),ROT(150,3,3)
COMMON / ETOT/ ETOT
COMMON / IMFO/ NOEL,NODES,NEDOF,NDOF,NPE
COMMON / LF/ F(18)
COMMON / AB/ AA(450),BB(450),S12(450),PP,S22,DELP
COMMON /T/  T(3,3)
DIMENSION INOD(10)


READ(1,*) IDC
READ(1,*) NINC,TOL,NOD,ID,DTOT
CF = 1.00
CX = 1.00
CY = 1.00
CR = 1.00
READ(1,*) NNOD,(INOD(I),I=1,NNOD)
```

```
      CALL INPOUT
      CALL SFORCE
      CLOSE (1)
      CLOSE (2,STATUS = 'DELETE')
      OPEN(2,FILE = 'GRP')

      CALL ACTIVE(NEQ,NEQP)

      CALL INIT(NEQ)

      DO 5 NEL=1,NOEL
      CALL COLHT(NEQ,NEL)
      CALL DIRCOS(NEL,1,T11,T12,T13,T21,T22,T23,T31,T32,T33)
      DO 3 II = 1,NPE
      ROT(NEL,II,1) = T31
      ROT(NEL,II,2) = T32
      ROT(NEL,II,3) = T33
    3 CONTINUE
    5 CONTINUE

      CALL ADDRES(NEQ,NEQP,NWK)
      IDGS = 1.
      PP   = 0.00
      DPRES = DTOT/NINC

      CALL OUTPUT
      DO 100 INC = 1,NINC
      SUMO = 0.00
      DO 18 I = 1,NODES
      DO 18 J = 1,NDOF
      FIT(I,J) = 0.00
   18 CONTINUE

      WRITE(6,*) '          '
      WRITE(6,*) ' STEP NUMBER = ',INC
      ITER =1
      WRITE(6,*) ' ITERATION   = ',ITER
      CALL ZERO (NWK)
      WRITE(6,*) ' UPDATING '
      IF( IDC.EQ.1 ) CALL SFORCE
      CALL UPDATE (NWK)
      WRITE(6,*) ' PARTITIONING '
      CALL PART (NEQ,NOD,ID)
      WRITE(6,*) ' SOLVING '
      CALL DECOMP (NEQ)
      CALL DISP (DPRES,NOD,ID,NEQ)

      CALL NEWCON
      WRITE(6,*) ' TESTING '
      CALL TEST (TOL,TST,NEQ,SUMO)
```

```
      CFF = CF*FIT(NOD,ID)
      CXX = CX*DT(NOD,1)
      CYY = CY*DT(NOD,2)
      CRR = CR*DT(NOD,3)
      WRITE(6,*) '        '
      WRITE(6,4444) INC,CFF,CXX,CYY,CRR,ITER
      WRITE(6,*) '    '
      WRITE(6,*) '*********************************'
      WRITE(6,*) ' '


   22 ITER = ITER + 1
      WRITE(6,*) ' STEP NUMBER = ',INC
      WRITE(6,*) ' ITERATION   = ',ITER
      DO 25 I = 1,NODES
      DO 25 J = 1,NDOF
      FIT(I,J) = 0.00
   25 CONTINUE

      CALL ZERO (NWK)
      WRITE(6,*) ' UPDATING '
      IF( IDC.EQ.1 ) CALL SFORCE
      CALL UPDATE(NWK)
      WRITE(6,*) ' PARTITIONING '
      CALL PART (NEQ,NOD,ID)
      WRITE(6,*) ' SOLVING '
      CALL DECOMP (NEQ)
      CALL DISP(0.00D0,NOD,ID,NEQ)
   28 PP = PP + DELP
      CALL NEWCON

      WRITE(6,*) ' TESTING '
      CALL TEST (TOL,TST,NEQ,SUMO)
      CFF = CF*PP*FT(NOD,ID)
      CXX = CX*DT(NOD,1)
      CYY = CY*DT(NOD,2)
      CRR = CR*DT(NOD,3)

      WRITE(6,*) '    '
      WRITE(6,4444) INC,CFF,CXX,CYY,CRR,ITER
      WRITE(6,*) '    '
      WRITE(6,*) '*****************************************'
      WRITE(6,*) ' '
      IF (TST.EQ.1) GOTO 22
      CALL OUTPUT

      DO 38 I = 1,NNOD
      NN = INOD(I)

      DO 45 J = 1,6
      FI2(NN,J) = PP*FT(NN,J)
   45 CONTINUE
```

```
      WRITE(3,5555) INC,ITER,NN
      WRITE(3,5566) FI2(NN,1),DT(NN,1),FI2(NN,2),DT(NN,2),
     *              FI2(NN,3),DT(NN,3),FI2(NN,4),DT(NN,4),
     *              FI2(NN,5),DT(NN,5),FI2(NN,6),DT(NN,6)
   38 CONTINUE
  100 CONTINUE
      RETURN

 2222 FORMAT(/'      LOAD FACTOR                    = ',F15.10,/,
     *        '      TOTAL LOAD                     = ',E15.8//,
     *        '*************************************************')
 3333 FORMAT('      PRESCRIBED DISPLACEMENT    = ',E15.5)
 4444 FORMAT(I4,2X,E14.7,2X,E14.7,2X,E14.7,2X,E14.7,2X,I2)
 5555 FORMAT(1X,'NUMBER                      : ',I5,/
     *       1X,'NUMBER OF ITERATIONS        : ',I5,/
     *       1X,'NODE NUMBER                 : ',I5,//
     *       1X,'            FORCE             DISPLACEMENT ',/
     *       1X,'       ---------------     ----------------')
 5566 FORMAT(6(7X,E15.8,4X,E15.8,/)//)
      END


C ***************************************************************
C
C ***************************************************************

      SUBROUTINE  FORCE
      IMPLICIT REAL *8(A-H,O-Z)

      COMMON / ES/ ES(171)
      COMMON / S/ S(20000)
      COMMON / MAX/ MAXA(450)
      COMMON / LM/ LM(18)
      COMMON / COOR/ X(100,3),Y(100,3),Z(100,3)
      COMMON / MP/ YM,RN,H
      COMMON /IDEN/ IDC,IDGS
      COMMON / ID/ IDOF(6,100)
      COMMON / IC/ ICON(150,3)
      COMMON / MHT/ MHT(450)
      COMMON / V/ V(450)
      COMMON / F/ FT(100,6),FI2(100,6),FIT(100,6)
      COMMON / ESL/ ESL(18,18),GS(18,18),EL(18,18)
      COMMON / D/ DIT(100,6),DT(100,6),D(18),ROT(150,3,3)
      COMMON / ETOT/ ETOT
      COMMON / IMFO/ NOEL,NODES,NEDOF,NDOF,NPE
      COMMON / LF/ F(18)
      COMMON /T/  T(3,3)
      DIMENSION INOD(10)
```

```
      READ(1,*) IDC
      READ(1,*) NINC,TOL,NOD,ID,DTOT
      CF = 1.00
      CX = 1.00
      CY = 1.00
      CZ = 1.00
      IDGS = 1.
      READ(1,*) NNOD,(INOD(I),I=1,NNOD)
      CALL INPOUT
      CALL SFORCE
      CLOSE(1)
      CLOSE(2,STATUS = 'DELETE')
      OPEN(2,FILE = 'GRP')
      CALL ACTIVE(NEQ,NEQP)
      CALL INIT(NEQ)

      DO 10 NEL=1,NOEL
      CALL COLHT(NEQ,NEL)
      CALL DIRCOS(NEL,1,T11,T12,T13,T21,T22,T23,T31,T32,T33)
      DO 3 II = 1,NPE
      ROT(NEL,II,1) = T31
      ROT(NEL,II,2) = T32
      ROT(NEL,II,3) = T33

    3 CONTINUE
   10 CONTINUE

      CALL ADDRES(NEQ,NEQP,NWK)

      PP = 0.00
      DELP = 1./NINC



      CALL OUTPUT
      DO 200 INC = 1,NINC

      SUMO = 0.00
      ITER = 0
      PP = PP + DELP

   50 ITER = ITER + 1
      WRITE(6,*) ' STEP NUMBER = ',INC
      WRITE(6,*) ' ITERATION   = ',ITER

      DO 55 I = 1,NODES
      DO 55 J = 1,NDOF
      FIT(I,J) = 0.00
   55 CONTINUE
```

```
      CALL ZERO (NWK)
      WRITE(6,*) ' UPDATING '
      IF(IDC.EQ.1) CALL SFORCE
      CALL UDLOAD(PP)
      CALL UPDATE (NWK)

      CALL SETLD
      WRITE(6,*) ' SOLVING '
      CALL DECOMP(NEQ)
      CALL REDBAK(NEQ)


      CALL NEWCON
      WRITE(6,*) ' TESTING '
      CALL TEST(TOL,TST,NEQ,SUMO)
      CFF = CF*FI2(NOD,ID)
      CXX = CX*DT(NOD,1)
      CYY = CY*DT(NOD,2)
      CRR = CR*DT(NOD,3)
      WRITE(6,*) '  '
      WRITE(6,666) INC,CFF,CXX,CYY,CRR,ITER
      WRITE(6,*) '  '
      WRITE(6,*) '**********************************
      WRITE(6,*) '  '
      IF (TST.EQ.1) GOTO 50
      DO 70 I = 1,NNOD
      NN = INOD(I)
      WRITE(3,5555) INC,ITER,NN
      WRITE(3,5566) FI2(NN,1),FI2(NN,2),FI2(NN,3)
      WRITE(3,5566) FI2(NN,4),FI2(NN,5),FI2(NN,6)
      WRITE(3,5566) DT(NN,1),DT(NN,2),DT(NN,3)
      WRITE(3,5566) DT(NN,4),DT(NN,5),DT(NN,6)
   70 CONTINUE
      CALL OUTPUT
  200 CONTINUE
      RETURN
  666 FORMAT(3X,I2,2X,E14.7,2X,E14.7,2X,E14.7,2X,E14.7,2X,I2)
 1111 FORMAT(///,3X,'NODAL DISPLACEMENTS : ',//
     *,3X,'NODE              U      ',8X,'      V      ',8X,'     W',/
     *,3X,'-----       -----------',8X,'-------------',8X,
     * '-------------',//)
 5555 FORMAT(1X,3(I5,','))
 5566 FORMAT(1X,3(E15.8,','))
      END


      SUBROUTINE  DISP (DPRES,NOD,ID,NEQ)
      IMPLICIT REAL *8(A-H,O-Z)

      COMMON / S/ S(20000)
```

```
      COMMON / COOR/ X(100,3),Y(100,3),Z(100,3)
      COMMON / ID/ IDOF(6,100)
      COMMON / V/ V(450)
      COMMON / F/ FT(100,6),FI2(100,6),FIT(100,6)
      COMMON / ESL/ ESL(18,18),GS(18,18),EL(18,18)
      COMMON / D/ DIT(100,6),DT(100,6),D(18),ROT(150,3,3)
      COMMON / IMFO/ NOEL,NODES,NEDOF,NDOF,NPE
      COMMON/ AB/ AA(450),BB(450),S12(450),PP,S22,DELP

      NPR = IDOF(ID,NOD)

      DO 20 I = 1,NODES
      DO 20 J = 1,NDOF
      N = IDOF(J,I)
      IF (N.EQ.0) GOTO 20
      V(N) = PP*FT(I,J) - FIT(I,J) - DPRES*S12(N)
   20 CONTINUE

      V(NPR) = DPRES

      CALL REDBAK (NEQ)
      AK = 0.00
      DO 30 I = 1,NEQ
      AA(I) = V(I)
      AK = AK + S12(I)*AA(I)
   30 CONTINUE

      DO 40 I = 1,NODES
      DO 40 J = 1,NDOF
      N = IDOF(J,I)
      IF (N.EQ.0) GOTO 40
      V(N) = FT(I,J)
   40 CONTINUE

      V(NPR) = 0.00

      CALL REDBAK (NEQ)
      BK = 0.00
      DO 50 I = 1,NEQ
      BB(I) = V(I)
      BK = BK + S12(I)*BB(I)
   50 CONTINUE

      FF = PP*FT(NOD,ID) - FIT(NOD,ID)
      DDD = BK - FT(NOD,ID)
      DELP = (FF - AK-S22*DPRES)/DDD

      DO 60 I = 1,NEQ
      V(I) = AA(I) + BB(I)*DELP
   60 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE  PART (NEQ,NOD,ID)
      IMPLICIT REAL *8(A-H,O-Z)
      COMMON / S/ S(20000)
      COMMON / MAX/ MAXA(450)
      COMMON / MHT/ MHT(450)
      COMMON / IMFO/ NOEL,NODES,NEDOF,NDOF,NPE
      COMMON/ AB/ AA(450),BB(450),S12(450),PP,S22,DELP
      COMMON / ID/ IDOF(6,100)

      DO 5 I = 1,NEQ
      S12(I) = 0.00
    5 CONTINUE

      J = 0
      N = IDOF(ID,NOD)
      NN = MAXA(N)
      MH = MHT(N)
      DO 10 I = NN,NN+MH
      JJ = NN + J
      S12(N-J) = S(JJ)
      S(JJ) = 0.00
      J = J+1
   10 CONTINUE

      S22 = S12(N)
      S12(N) = 0.00
      S(NN) = 1.00

      J = 0
      DO 20 I = N+1,NEQ
      J = J + 1
      NN = MAXA(I)
      IF(MHT(I).LT.J) GOTO 20
      JJ = NN + J
      S12(I) = S(JJ)
      S(JJ) = 0.00
   20 CONTINUE


      RETURN
      END


      SUBROUTINE OUTPUT
      IMPLICIT REAL *8(A-H,O-Z)

      COMMON /ID/ IDOF(6,100)
      COMMON /V/ V(450)
      COMMON /MP/ YM,RN,H
      COMMON / COOR/ X(100,3),Y(100,3),Z(100,3)
      COMMON / IMFO/ NOEL,NODES,NEDOF,NDOF,NPE
```

```
      DO 10 I = 1,NODES
      WRITE(2,111) I,X(I,3),Y(I,3),Z(I,3)
   10 CONTINUE
      WRITE(2,222)
  111 FORMAT(I4,',',3(E16.8,','))
  222 FORMAT('                                        ')
      RETURN
      END



      SUBROUTINE  STRESS(NEL)
      IMPLICIT REAL *8(A-H,O-Z)

      COMMON / COOR/ X(100,3),Y(100,3),Z(100,3)
      COMMON/ IC/ ICON(150,3)
      COMMON / D/ DIT(100,6),DT(100,6),D(18),ROT(150,3,3)
      COMMON / ESL/ ESL(18,18),GS(18,18),EL(18,18)
      COMMON / IMFO/ NOEL,NODES,NEDOF,NDOF,NPE
      COMMON / MP/ YM,RN,H
      COMMON / STRS/ SNXX,SNXY,SNYY

      CALL DIRCOS (NEL,3,T11,T12,T13,T21,T22,T23,T31,T32,T33)

      I = ICON(NEL,1)
      J = ICON(NEL,2)
      K = ICON(NEL,3)

      X21 = X(J,3) - X(I,3)
      Y21 = Y(J,3) - Y(I,3)
      Z21 = Z(J,3) - Z(I,3)

      X31 = X(K,3) - X(I,3)
      Y31 = Y(K,3) - Y(I,3)
      Z31 = Z(K,3) - Z(I,3)

      V1 = Y21*Z31 - Y31*Z21
      V2 = X31*Z21 - X21*Z31
      V3 = X21*Y31 - X31*Y21

      AREA = 0.5*SQRT(V1*V1 + V2*V2 + V3*V3)

      DX1 = D(1)
      DY1 = D(2)

      DX2 = D(7)
      DY2 = D(8)

      DX3 = D(13)
      DY3 = D(14)
```

```
X1 = 0.00
Y1 = 0.00

X2 = T11*X21 + T12*Y21 + T13*Z21
Y2 = T21*X21 + T22*Y21 + T23*Z21

X3 = T11*X31 + T12*Y31 + T13*Z31
Y3 = T21*X31 + T22*Y31 + T23*Z31

B1 = Y2 - Y3
B2 = Y3 - Y1
B3 = Y1 - Y2

A1 = X3 - X2
A2 = X1 - X3
A3 = X2 - X1

CON = 1/(2.*AREA)

EXX = (B1*DX1 + B2*DX2 + B3*DX3)*CON
EYY = (A1*DY1 + A2*DY2 + A3*DY3)*CON
EXY =  A1*DX1 + A2*DX2 + A3*DX3 + B1*DY1 + B2*DY2 + B3*DY3
EXY = EXY*CON

CK11 = YM*H/(1 - RN*RN)
CK12 = (1- RN)/2.

SNXX = CK11*(EXX + RN*EYY)
SNYY = CK11*(EYY + RN*EXX)
SNXY = CK11*CK12*EXY

RETURN
END



SUBROUTINE INPOUT
IMPLICIT REAL *8(A-H,O-Z)

COMMON /COOR/ X(100,3),Y(100,3),Z(100,3)
COMMON /IC/ ICON(150,3)
COMMON /ID/ IDOF(6,100)
COMMON /MP/ YM,RN,H
COMMON /IDEN/ IDC,IDGS
COMMON /F/ FT(100,6),FI2(100,6),FIT(100,6)
COMMON / D/ DIT(100,6),DT(100,6),D(18),ROT(150,3,3)
COMMON / IMFO/ NOEL,NODES,NEDOF,NDOF,NPE
COMMON /FEL/ FEL(100,18)
```

```
      WRITE(2,1111)
      READ(1,*) NOEL,NODES,NEDOF,NDOF,NPE,NELOD
      WRITE(2,1122) NOEL,NODES,NEDOF,NDOF,NELOD

      DO 10 I = 1,NOEL
      DO 10 J = 1,3
      ROT(I,J,1) = 0.00
      ROT(I,J,2) = 0.00
      ROT(I,J,3) = 0.00
      ICON(I,J) = 0
   10 CONTINUE

      DO 30 I = 1,NODES
      DO 20 J = 1,3
      X(I,J) = 0.00
      Y(I,J) = 0.00
      Z(I,J) = 0.00
   20 CONTINUE
      DO 30 J = 1,NDOF
      IDOF(J,I) = 0
      FT(I,J) = 0.00
      FI2(I,J) = 0.00
      FIT(I,J) = 0.00
      DT(I,J) = 0.00
      DIT(I,J) = 0.00
   30 CONTINUE



      WRITE(2,1133)
      DO 40 I=1,NODES
      READ(1,*) NOD,X(NOD,1),Y(NOD,1),Z(NOD,1)
      WRITE(2,121) NOD,X(NOD,1),Y(NOD,1),Z(NOD,1)
      X(NOD,2) = X(NOD,1)
      Y(NOD,2) = Y(NOD,1)
      Z(NOD,2) = Z(NOD,1)
      X(NOD,3) = X(NOD,2)
      Y(NOD,3) = Y(NOD,2)
      Z(NOD,3) = Z(NOD,2)
   40 CONTINUE


      WRITE(2,1211)
      DO 50 I=1,NOEL
      READ(1,*) NEL,NOD1,NOD2,NOD3
      ICON(NEL,1)=NOD1
      ICON(NEL,2)=NOD2
      ICON(NEL,3)=NOD3
      WRITE(2,222) NEL,ICON(NEL,1),ICON(NEL,2),ICON(NEL,3)
   50 CONTINUE
```

```
      WRITE(2,1155)
      DO 60 I=1,NODES
      READ(1,*) NOD,IDOF(1,NOD),IDOF(2,NOD),IDOF(3,NOD),
     *                  IDOF(4,NOD),IDOF(5,NOD),IDOF(6,NOD)
      WRITE(2,333) NOD,IDOF(1,NOD),IDOF(2,NOD),IDOF(3,NOD),
     *                  IDOF(4,NOD),IDOF(5,NOD),IDOF(6,NOD)

   60 CONTINUE

      WRITE(2,1166)
      READ(1,*) YM,RN,H
      WRITE(2,444) YM,RN,H

      WRITE(2,1177)
      DO 65 I=1,NOEL
      DO 65 J = 1,18
      FEL(I,J) = 0.00
   65 CONTINUE
      DO 70 I=1,NELOD
      READ(1,*) NEL,NOD,LG,FX,FY,FZ,BX,BY,BZ
      IF( NOD.EQ.ICON(NEL,1)) NOD = 1
      IF( NOD.EQ.ICON(NEL,2)) NOD = 2
      IF( NOD.EQ.ICON(NEL,3)) NOD = 3
      IF( LG.EQ.1 ) CALL TRLD(NEL,FX,FY,FZ,BX,BY,BZ)
      NN = 6*(NOD - 1)
      FEL(NEL,NN+1) = FX
      FEL(NEL,NN+2) = FY
      FEL(NEL,NN+3) = FZ
      FEL(NEL,NN+4) = BX
      FEL(NEL,NN+5) = BY
      FEL(NEL,NN+6) = BZ
   70 CONTINUE

  111 FORMAT(5X,I2,5X,I2,5X,I2,5X,I2,5X,5X,I2,5X,F10.4)
  121 FORMAT(4X,I3,',',F10.5,',',5X,',',F10.5,',',5X,F10.5)
  222 FORMAT(5X,I2,5X,I2,5X,I2,5X,I2/)
  333 FORMAT(4X,I3,5X,I2,5X,I2,5X,I2,5X,I2,5X,I2,5X,I2/)

  555 FORMAT(4X,I3,6(2X,F5.1,2X),/)
 1111 FORMAT(///,'                    ANALYSIS  STARTED        ')
 1122 FORMAT(///,5X,'NUMBER OF ELEMENTS : ',I2,/
     *          ,5X,'NUMBER OF NODES    : ',I2,/
     *          ,5X,'NEDOF              : ',I2,/
     *          ,5X,'NDOF               : ',I2,/
     *          ,5X,'NPE                : ',I2,//)
 1133 FORMAT(/,3X,'NODAL COORDINATES :',//
     *         ,3X,' NODE       X            Y              Z',/
     *         ,3X,' ----   ---------   ----------    ----------',//)
 1155 FORMAT(///,5X,'JOINT FIXITIES',//
     *       ,5X,'NODE    DX    DY    DZ    TET-X    TET-Y    TET-Z',/
     *       ,5X,'-----  ----   ---   ---   -----    ------   -----')
```

```
66 FORMAT(///,3X,'ELASTIC PROPERTIES : ',// )
44 FORMAT(    10X,'ELASTIC MODULUS        = ',E15.6,/
  *           10X,'POISSONS RATIO         = ',E15.6,/
  *           10X,'THICKNESS               = ',E15.6)
77 FORMAT(///,3X,'NODAL FORCES : ',//
  *    ,5X,'NODE      FX      FY       FZ         MX       MY         MZ',/
  *    ,5X,'----    ------  -----    -----     -----    -----     ------')

11 FORMAT(///)
   RETURN
   END



   SUBROUTINE  NEWCON
   IMPLICIT REAL *8(A-H,O-Z)

   COMMON / IMFO/ NOEL,NODES,NEDOF,NDOF,NPE
   COMMON / D/ DIT(100,6),DT(100,6),D(18),ROT(150,3,3)
   COMMON / COOR/ X(100,3),Y(100,3),Z(100,3)
   COMMON / ID/ IDOF(6,100)
   COMMON / IC/ ICON(150,3)
   COMMON / V/ V(450)


   DO 10 I=1,NODES
   DO 10 J=1,NDOF
   N=IDOF(J,I)
   IF ( N.EQ.0 ) GOTO 5
   DIT(I,J) = V(N)
   GOTO 10
 5 DIT(I,J) = 0.00
10 CONTINUE

   DO 15 NOD =1,NODES
   DT(NOD,1) = DT(NOD,1) + DIT(NOD,1)
   DT(NOD,2) = DT(NOD,2) + DIT(NOD,2)
   DT(NOD,3) = DT(NOD,3) + DIT(NOD,3)
   DT(NOD,4) = DT(NOD,4) + DIT(NOD,4)
   DT(NOD,5) = DT(NOD,5) + DIT(NOD,5)
   DT(NOD,6) = DT(NOD,6) + DIT(NOD,6)
15 CONTINUE

   DO 20 NOD=1,NODES
   X(NOD,2) = X(NOD,3)
   X(NOD,3) = X(NOD,2) + DIT(NOD,1)
   Y(NOD,2) = Y(NOD,3)
   Y(NOD,3) = Y(NOD,2) + DIT(NOD,2)
   Z(NOD,2) = Z(NOD,3)
   Z(NOD,3) = Z(NOD,3) + DIT(NOD,3)
20 CONTINUE
```

```
      RETURN
      END



      SUBROUTINE TEST (TOL,TST,NEQ,SUMO)
      IMPLICIT REAL *8(A-H,O-Z)

      COMMON / IMFO/ NOEL,NODES,NEDOF,NDOF,NPE
      COMMON / ETOT/ ETOT
      COMMON / D/ DIT(100,6),DT(100,6),D(18),ROT(150,3,3)
      COMMON / F/ FT(100,6),FI2(100,6),FIT(100,6)
      COMMON / ID/ IDOF(6,100)

      EC = 0.00
      SUM1 = 0.00
      SUM2 = 0.00

      DO 10 I = 1,NODES
      DO 10 J = 1,6
      N = IDOF(J,I)
      IF (N.EQ.0) GOTO 10
      SUMO = SUMO + ABS(DIT(I,J))
      SUM1 = SUM1 + ABS(DT(I,J))
      SUM2 = SUM2 + ABS(DIT(I,J))
   10 CONTINUE
      ERR = SUM2/SUMO

      WRITE(6,100) ERR,TOL
  100 FORMAT(/,5X,'ERROR = ',F10.5,/,
     *                    5X,'TOL   = ',F10.5)

      IF ( ERR.LE.TOL ) TST = 0
      IF ( ERR.GT.TOL ) TST = 1

      RETURN
      END




      SUBROUTINE ZERO(NWK)
      IMPLICIT REAL *8(A-H,O-Z)

      COMMON / S/ S(20000)
      DO 10 I = 1,NWK
      S(I) = 0.00
   10 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE  INIT(NEQ)
      IMPLICIT REAL *8(A-H,O-Z)

      COMMON / MAX/ MAXA(450)
      COMMON / MHT/ MHT(450)
      COMMON / V/ V(450)
      COMMON / IMFO/ NOEL,NODES,NEDOF,NDOF,NPE

      DO 10 I = 1,NEQ
      V(I) = 0.00
      MHT(I) = 0.00
      MAXA(I) = 0.00
   10 CONTINUE
      RETURN
      END



      SUBROUTINE  UPDATE(NWK)
      COMMON / IMFO/ NOEL,NODES,NEDOF,NDOF,NPE
      DO 100 NEL = 1,NOEL

      CALL LSTIFF (NEL)
      CALL LLOAD (NEL)
      CALL STRESS(NEL)
      CALL GSTIFF (NEL)
      CALL ESLOC  (NEL)
      CALL ESGLOB (NEL)
      CALL ASMBLY (NEL)
      CALL SLOAD (NEL)
  100 CONTINUE
      RETURN
      END



      SUBROUTINE  DIRCOS (NEL,L,C11,C12,C13,C21,C22,C23,C31,C32,C33)
      IMPLICIT REAL *8(A-H,O-Z)

      COMMON /IC/ ICON(150,3)
      COMMON /COOR/  X(100,3),Y(100,3),Z(100,3)
      COMMON /T/  T(3,3)

      I = ICON(NEL,1)
      J = ICON(NEL,2)
      K = ICON(NEL,3)

      X12 = X(J,L) - X(I,L)
      Y12 = Y(J,L) - Y(I,L)
      Z12 = Z(J,L) - Z(I,L)
```

```
      C13 = Z12/V12

      VZ1 = Y12*Z13 - Y13*Z12
      VZ2 = X13*Z12 - X12*Z13
      VZ3 = X12*Y13 - X13*Y12

      VZ = SQRT(VZ1*VZ1 + VZ2*VZ2 + VZ3*VZ3)

      C31 = VZ1/VZ
      C32 = VZ2/VZ
      C33 = VZ3/VZ

      C21 = C32*C13 - C12*C33
      C22 = C11*C33 - C31*C13
      C23 = C31*C12 - C11*C32

      T(1,1) = C11
      T(1,2) = C12
      T(1,3) = C13
      T(2,1) = C21
      T(2,2) = C22
      T(2,3) = C23
      T(3,1) = C31
      T(3,2) = C32
      T(3,3) = C33

      RETURN
      END



      SUBROUTINE ESLOC(NEL)
      IMPLICIT REAL *8(A-H,O-Z)

      COMMON / ESL/ ESL(18,18),GS(18,18),EL(18,18)
      COMMON / IMFO/ NOEL,NODES,NEDOF,NDOF,NPE

      DO 10 I = 1,NEDOF
      DO 10 J = 1,NEDOF
      ESL(I,J) = EL(I,J) + GS(I,J)
   10 CONTINUE
      RETURN
      END



      SUBROUTINE  ESGLOB(NEL)
      IMPLICIT REAL *8(A-H,O-Z)
```

```
      COMMON / ESL/ ESL(18,18),GS(18,18),EL(18,18)
      COMMON / IMFO/ NOEL,NODES,NEDOF,NDOF,NPE
      COMMON /T/  T(3,3)
      COMMON /ES/ ES(171)
      DIMENSION ESG(18,18)

      CALL DIRCOS(NEL,3,T11,T12,T13,T21,T22,T23,T31,T32,T33)

      DO 10 II = 1,6
      DO 10 JJ = 1,6
      I = 3*(II-1)
      J = 3*(JJ-1)

      E11 = ESL(I+1,J+1)
      E12 = ESL(I+1,J+2)
      E13 = ESL(I+1,J+3)
      E21 = ESL(I+2,J+1)
      E22 = ESL(I+2,J+2)
      E23 = ESL(I+2,J+3)
      E31 = ESL(I+3,J+1)
      E32 = ESL(I+3,J+2)
      E33 = ESL(I+3,J+3)

      A11 = E11*T11 + E12*T21 + E13*T31
      A12 = E11*T12 + E12*T22 + E13*T32
      A13 = E11*T13 + E12*T23 + E13*T33
      A21 = E21*T11 + E22*T21 + E23*T31
      A22 = E21*T12 + E22*T22 + E23*T32
      A23 = E21*T13 + E22*T23 + E23*T33
      A31 = E31*T11 + E32*T21 + E33*T31
      A32 = E31*T12 + E32*T22 + E33*T32
      A33 = E31*T13 + E32*T23 + E33*T33

      ESG(I+1,J+1) = T11*A11 + T21*A21 + T31*A31
      ESG(I+1,J+2) = T11*A12 + T21*A22 + T31*A32
      ESG(I+1,J+3) = T11*A13 + T21*A23 + T31*A33
      ESG(I+2,J+1) = T12*A11 + T22*A21 + T32*A31
      ESG(I+2,J+2) = T12*A12 + T22*A22 + T32*A32
      ESG(I+2,J+3) = T12*A13 + T22*A23 + T32*A33
      ESG(I+3,J+1) = T13*A11 + T23*A21 + T33*A31
      ESG(I+3,J+2) = T13*A12 + T23*A22 + T33*A32
      ESG(I+3,J+3) = T13*A13 + T23*A23 + T33*A33

10    CONTINUE

      N = 0
      DO 20 I=1,NEDOF
      DO 20 J=I,NEDOF
      N = N+1
      ES(N) = ESG(I,J)
       WRITE(3,*) N,ES(N)
20    CONTINUE

      RETURN
      END


      SUBROUTINE LSTIFF(NEL)
      IMPLICIT REAL *8(A-H,O-Z)

     COMMON /PREV/ XI(3,3),XIX(3,3),XIY(3,3),RX(3,3),RY(3,3),
     *              SX(3,3),SY(3,3),RLX(3,3),RLY(3,3),RMX(3,3),
     *              RMY(3,3),A1,A2,A3,B1,B2,B3,AREA,DETJ,W

      COMMON / MP/ YM,RN,H
      COMMON / IC/ ICON(150,3)
      COMMON / ESL/ ESL(18,18),GS(18,18),EL(18,18)
      COMMON / IMFO/ NOEL,NODES,NEDOF,NDOF,NPE
```

```
      DIMENSION B11(3),B16(3),B22(3),B26(3),B31(3),B32(3),B36(3),
     *          B45(3),B54(3),B64(3),B65(3),B73(3),B74(3),B75(3),
     *          B83(3),B84(3),B85(3)

      DO 5 I = 1,NEDOF
      DO 5 J = 1,NEDOF
      EL(I,J) = 0.00
    5 CONTINUE

      E = YM
      SMEM = E*H/(1 - RN*RN)
      SBEN = E*H*H*H/(12*(1 - RN*RN))
      SSHR = 0.82246*E*H/(2.*(1+RN))

      AS = AREA/(H*H)
      PHI = 1/(1 + 0.5*AS)
      SSHR = PHI*SSHR

      D11 = SMEM
      D12 = RN*SMEM
      D22 = SMEM
      D33 = 0.5*(1 - RN)*SMEM
      D44 = SBEN
      D45 = RN*SBEN
      D55 = SBEN
      D66 = 0.5*(1-RN)*SBEN
      D77 = SSHR
      D88 = SSHR

      DO 10 K = 1,3
      DO 20 L = 1,3
      B11(L) = XIX(L,K)
      B16(L) = RX(L,K)
      B22(L) = XIY(L,K)
      B26(L) = SY(L,K)
      B31(L) = XIY(L,K)
      B32(L) = XIX(L,K)
      B36(L) = SX(L,K) + RY(L,K)
      B45(L) = XIX(L,K)
      B54(L) = XIY(L,K)
      B64(L) = XIX(L,K)
      B65(L) = XIY(L,K)
      B73(L) = XIX(L,K)
      B74(L) = RLX(L,K)
      B75(L) = RMX(L,K) + XI(L,K)
      B83(L) = XIY(L,K)
      B84(L) = RLY(L,K) + XI(L,K)
      B85(L) = RMY(L,K)
   20 CONTINUE

      DO 30 I = 1,3
      DO 30 J = I,3
      E11 = B11(I)*D11*B11(J) + B31(I)*D33*B31(J)
      E12 = B11(I)*D12*B22(J) + B31(I)*D33*B32(J)
      E16 = B11(I)*(D11*B16(J) + D12*B26(J)) + B31(I)*D33*B36(J)
      E21 = B22(I)*D12*B11(J) + B32(I)*D33*B31(J)
      E22 = B22(I)*D22*B22(J) + B32(I)*D33*B32(J)
      E26 = B22(I)*(D12*B16(J) + D22*B26(J)) + B32(I)*D33*B36(J)
      E33 = B73(I)*D77*B73(J) + B83(I)*D88*B83(J)
      E34 = B73(I)*D77*B74(J) + B83(I)*D88*B84(J)
      E35 = B73(I)*D77*B75(J) + B83(I)*D88*B85(J)
      E43 = B74(I)*D77*B73(J) + B84(I)*D88*B83(J)
      E44 = B54(I)*D55*B54(J) + B64(I)*D66*B64(J) +
     *      B74(I)*D77*B74(J) + B84(I)*D88*B84(J)
      E45 = B54(I)*D45*B45(J) + B64(I)*D66*B65(J) +
     *      B74(I)*D77*B75(J) + B84(I)*D88*B85(J)
      E53 = B75(I)*D77*B73(J) + B85(I)*D88*B83(J)
```

```
      E54 = B45(I)*D45*B54(J) + B65(I)*D66*B64(J) +
     *      B75(I)*D77*B74(J) + B85(I)*D88*B84(J)
      E55 = B45(I)*D44*B45(J) + B65(I)*D66*B65(J) +
     *      B75(I)*D77*B75(J) + B85(I)*D88*B85(J)
      E61 = B16(I)*D11*B11(J) + B26(I)*D12*B11(J) +B36(I)*D33*B31(J)
      E62 = B16(I)*D12*B22(J) + B26(I)*D22*B22(J) +B36(I)*D33*B32(J)
      E66 = B16(I)*(D11*B16(J) + D12*B26(J)) + B36(I)*D33*B36(J) +
     *      B26(I)*(D12*B16(J) + D22*B26(J))

      II = 6*(I - 1)
      JJ = 6*(J - 1)

      EL(II+1,JJ+1) = EL(II+1,JJ+1) + W*DETJ*E11
      EL(II+1,JJ+2) = EL(II+1,JJ+2) + W*DETJ*E12
      EL(II+1,JJ+6) = EL(II+1,JJ+6) + W*DETJ*E16
      EL(II+2,JJ+1) = EL(II+2,JJ+1) + W*DETJ*E21
      EL(II+2,JJ+2) = EL(II+2,JJ+2) + W*DETJ*E22
      EL(II+2,JJ+6) = EL(II+2,JJ+6) + W*DETJ*E26
      EL(II+3,JJ+3) = EL(II+3,JJ+3) + W*DETJ*E33
      EL(II+3,JJ+4) = EL(II+3,JJ+4) + W*DETJ*E34
      EL(II+3,JJ+5) = EL(II+3,JJ+5) + W*DETJ*E35
      EL(II+4,JJ+3) = EL(II+4,JJ+3) + W*DETJ*E43
      EL(II+4,JJ+4) = EL(II+4,JJ+4) + W*DETJ*E44
      EL(II+4,JJ+5) = EL(II+4,JJ+5) + W*DETJ*E45
      EL(II+5,JJ+3) = EL(II+5,JJ+3) + W*DETJ*E53
      EL(II+5,JJ+4) = EL(II+5,JJ+4) + W*DETJ*E54
      EL(II+5,JJ+5) = EL(II+5,JJ+5) + W*DETJ*E55
      EL(II+6,JJ+1) = EL(II+6,JJ+1) + W*DETJ*E61
      EL(II+6,JJ+2) = EL(II+6,JJ+2) + W*DETJ*E62
      EL(II+6,JJ+6) = EL(II+6,JJ+6) + W*DETJ*E66

 30   CONTINUE
 10   CONTINUE

      DO 40 I = 1,18
      DO 40 J = 1,I
      EL(I,J) = EL(J,I)
 40   CONTINUE

      DO 50 I = 1,18
      EL(4,I)  = -EL(4,I)
      EL(I,4)  = -EL(I,4)
      EL(10,I) = -EL(10,I)

      EL(I,10) = -EL(I,10)
      EL(16,I) = -EL(16,I)
      EL(I,16) = -EL(I,16)
 50   CONTINUE
111   FORMAT(35X'EL(',I4,',',I4,') = ',E16.8)
      RETURN
      END

      SUBROUTINE GSTIFF(NEL)
      IMPLICIT REAL *8(A-H,O-Z)

      COMMON / COOR/ X(100,3),Y(100,3),Z(100,3)
      COMMON/ IC/ ICON(150,3)
      COMMON / D/ DIT(100,6),DT(100,6),D(18),ROT(150,3,3)
      COMMON / ESL/ ESL(18,18),GS(18,18),EL(18,18)
      COMMON / IMFO/ NOEL,NODES,NEDOF,NDOF,NPE
      COMMON / MP/ YM,RN,H
      COMMON /IDEN/ IDC,IDGS
      COMMON / STRS/ SXX,SXY,SYY
      DIMENSION PS(3,3)
```

```
      DO 5 I = 1,NEDOF
      DO 5 J = 1,NEDOF
      GS(I,J) = 0.00
    5 CONTINUE

      IF(IDGS.EQ.0) RETURN
      CALL DIRCOS (NEL,3,T11,T12,T13,T21,T22,T23,T31,T32,T33)

      SXX = STRS(NEL,1)
      SYY = STRS(NEL,2)
      SXY = STRS(NEL,3)

      I = ICON(NEL,1)
      J = ICON(NEL,2)
      K = ICON(NEL,3)

      X21 = X(J,3) - X(I,3)
      Y21 = Y(J,3) - Y(I,3)
      Z21 = Z(J,3) - Z(I,3)

      X31 = X(K,3) - X(I,3)
      Y31 = Y(K,3) - Y(I,3)
      Z31 = Z(K,3) - Z(I,3)

      V1 = Y21*Z31 - Y31*Z21
      V2 = X31*Z21 - X21*Z31
      V3 = X21*Y31 - X31*Y21

      AREA = 0.5*SQRT(V1*V1 + V2*V2 + V3*V3)
      CON  = 1/(4*AREA)

      X1 = 0.00
      Y1 = 0.00

      X2 = T11*X21 + T12*Y21 + T13*Z21
      Y2 = T21*X21 + T22*Y21 + T23*Z21

      X3 = T11*X31 + T12*Y31 + T13*Z31
      Y3 = T21*X31 + T22*Y31 + T23*Z31

      B1 = Y2 - Y3
      B2 = Y3 - Y1
      B3 = Y1 - Y2

      A1 = X3 - X2
      A2 = X1 - X3

      A3 = X2 - X1

      P1X = 2*CON*B1
      P2X = 2*CON*B2
      P3X = 2*CON*B3

      P1Y = 2*CON*A1
      P2Y = 2*CON*A2
      P3Y = 2*CON*A3

      GS11 = AREA*( P1X*(SXX*P1X+SXY*P1Y) + P1Y*(SXY*P1X+SYY*P1Y) )
      GS12 = AREA*( P1X*(SXX*P2X+SXY*P2Y) + P1Y*(SXY*P2X+SYY*P2Y) )
      GS13 = AREA*( P1X*(SXX*P3X+SXY*P3Y) + P1Y*(SXY*P3X+SYY*P3Y) )
      GS22 = AREA*( P2X*(SXX*P2X+SXY*P2Y) + P2Y*(SXY*P2X+SYY*P2Y) )
      GS23 = AREA*( P2X*(SXX*P3X+SXY*P3Y) + P2Y*(SXY*P3X+SYY*P3Y) )
      GS33 = AREA*( P3X*(SXX*P3X+SXY*P3Y) + P3Y*(SXY*P3X+SYY*P3Y) )

      GS( 1, 1) = GS11
      GS( 2, 2) = GS11
      GS( 3, 3) = GS11
```

```
     GS(15,16) = GS(15,16) + W*( P3X*(SXX*R3X+SXY*R3Y)
    *                          + P3Y*(SXY*R3X+SYY*R3Y) )
     GS(15,17) = GS(15,17) + W*( P3X*(SXX*S3X+SXY*S3Y)
    *                          + P3Y*(SXY*S3X+SYY*S3Y) )
     GS(16,16) = GS(16,16) + W*( R3X*(SXX*R3X+SXY*R3Y)
    *                          + R3Y*(SXY*R3X+SYY*R3Y) )
     GS(16,17) = GS(16,17) + W*( R3X*(SXX*S3X+SXY*S3Y)
    *                          + R3Y*(SXY*S3X+SYY*S3Y) )
     GS(17,17) = GS(17,17) + W*( S3X*(SXX*S3X+SXY*S3Y)
    *                          + S3Y*(SXY*S3X+SYY*S3Y) )

10   CONTINUE

15   DO 20 I = 1,18
     DO 20 J = I,18
     GS(J,I) = GS(I,J)
20   CONTINUE

     DO 30 I = 1,18
     GS(4,I) = -GS(4,I)
     GS(I,4) = -GS(I,4)
     GS(10,I)= -GS(10,I)
     GS(I,10)= -GS(I,10)
     GS(16,I)= -GS(16,I)
     GS(I,16)= -GS(I,16)

30   CONTINUE
     RETURN
     END
```

```
      SUBROUTINE  TRLD(NEL,FX,FY,FZ,BX,BY,BZ)

      IMPLICIT REAL *8(A-H,O-Z)

      CALL DIRCOS(NEL,3,T11,T12,T13,T21,T22,T23,T31,T32,T33)

      F1 = T11*FX + T12*FY + T13*FZ
      F2 = T21*FX + T22*FY + T23*FZ
      F3 = T31*FX + T32*FY + T33*FZ
      F4 = T11*BX + T12*BY + T13*BZ
      F5 = T21*BX + T22*BY + T23*BZ
      F6 = T31*BX + T32*BY + T33*BZ
      FX = F1
      FY = F2
      FZ = F3
      BX = F4
      BY = F5
      BZ = F6
      RETURN
      END


      SUBROUTINE SFORCE

      IMPLICIT REAL *8(A-H,O-Z)

      COMMON / F/ FT(100,6),FI2(100,6),FIT(100,6)
      COMMON / LF/ F(18)
      COMMON /T/  T(3,3)
      COMMON / IC/ ICON(150,3)
      COMMON /FEL/ FEL(100,18)
      COMMON / IMFO/ NOEL,NODES,NEDOF,NDOF,NPE

      DO 5 I=1,NODES
      DO 5 J=1,6
      FT(I,J) = 0.00
    5 CONTINUE


      DO 10 K = 1,NOEL

      CALL DIRCOS(K,3,S11,S12,S13,S21,S22,S23,S31,S32,S33)

      DO 10 I = 1,3

      J = ICON(K,I)
      N = 6*(I-1)

      F1 = FEL(K,N+1)
      F2 = FEL(K,N+2)
      F3 = FEL(K,N+3)
      F4 = FEL(K,N+4)
      F5 = FEL(K,N+5)
      F6 = FEL(K,N+6)
```

```
      FT(J,1) = FT(J,1) + S11*F1 + S21*F2 + S31*F3
      FT(J,2) = FT(J,2) + S12*F1 + S22*F2 + S32*F3
      FT(J,3) = FT(J,3) + S13*F1 + S23*F2 + S33*F3
      FT(J,4) = FT(J,4) + S11*F4 + S21*F5 + S31*F6
      FT(J,5) = FT(J,5) + S12*F4 + S22*F5 + S32*F6
      FT(J,6) = FT(J,6) + S13*F4 + S23*F5 + S33*F6
   10 CONTINUE

      RETURN
      END
```