Electronic Theses and Dissertations

Theses, Dissertations, and Major Papers

1990

Analysis of air supported structures.

Sreedhar. Sundaram
University of Windsor

Follow this and additional works at: https://scholar.uwindsor.ca/etd

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Canada

ANALYSIS OF AIR SUPPORTED STRUCTURES

by

SREEDHAR SUNDARAM

'

A Thesis
submitted to the
Faculty of Graduate Studies and Research
through the Department of
Civil and Environmental Engineering
in Partial Fulfillment of the requirements for
the Degree of Master of Applied Science at
the University of Windsor

Windsor, Ontario, Canada
August 1990

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

Canada

HCP 1901

I hereby declare that I am the sole author of this document.

I authorize the University of Windsor to lend this document to other institutions or individuals for the purpose of scholarly research.

SREEDHAR SUNDARAM

I further authorize the University of Windsor to reproduce this document by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

SREEDHAR SUNDARAM

THE UNIVERSITY OF WINDSOR requires the signatures of all persons using or photocopying this document. Please sign below, and give address and date.

ABSTRACT

Air supported structures have been gaining popularity due to a number of reasons. Their inherent architectural advantage and the increased demand for large column-free spaces make them ideal for large arenas and stadiums. The constructional advantages and the availability of high strength steels add to their popularity. But such structures are hindered by the complexity of their behavioural response to all kinds of loads. The high degree of indeterminancy present imposes severe restraints on the available methods of analysis and makes them very difficult to analyze.

The present study encompasses a comparison of the experimental and theoretical analysis of air supported cable roof structures. A 1:120 scale model of the Pontiac Silver Dome in Pontiac, Michigan, U.S.A., was used to substantiate the results obtained by the various solution techniques used in the theoretical study. The energy search method is compared to three standard one-step solution techniques presented by Baron and Venkatesan and Kar and Okazaki.

Computer programs were developed using FORTRAN for all of the above solution methods to simulate various kinds of distributed loads on the structure. For the purpose of experimental study, the model was inflated with different air pressures and the nodal coordinates were measured for each case. Thereafter, keeping the internal pressure constant, the structure was loaded with different types of externally applied patch loads of sand to simulate live loads on the dome.

Load-displacement curves at various nodes on the roof were plotted and compared to the theoretical results obtained. Finally conclusions are drawn and recommendations for further research are presented.

ACKNOWLEDGEMENT

NOTATION

A = *area of cross section of the member*

D = *direction cosine vector*

$\bar{D}$ = *direction cosine vector in the $\bar{x}$, $\bar{y}$, $\bar{z}$ directions*

E = *elastic modulus of the member*

$\{G\}$ = *gradient vector of the total potential energy*

$\{G_E\}$ = *gradient vector*

H = *displacement field at a local minimum*

K_{12} = *stiffness matrix of the structure from State 1 to State 2*

K_{COR} = *correction to the initial estimate of the stiffness matrix*

K_E = *stiffness matrix as per elastic theory*

K_{EST} = *initial estimate of the stiffness matrix*

k_G = *geometric stiffness matrix of an element*

k_L = *linearized stiffness matrix of an element*

k_{NL} = *nonlinear component of the stiffness matrix of an element*

K_{NL} = *nonlinear stiffness matrix of the structure*

K_S = *secant stiffness matrix of the structure*

k_T = *tangent stiffness matrix of an element*

K_T = *tangent stiffness matrix of the structure*

L_y = *length vector in state 1*

$L_2 = $ *length vector in state 2*

$l_o = $ *unstiffened length of the member*

$l_1 = $ *length of the member in State 1*

$l_2 = $ *length of the member in State 2*

$l,m,n = $ *direction cosines in x, y, z directions*

$\bar{l},\bar{m},\bar{n}, = $ *direction cosines in $\bar{x}$, $\bar{y}$, $\bar{z}$, directions*

$P = $ *applied load vector of the structure*

$P_{12} = $ *column vector of loads from State 1 to State 2*

$P_{EQ} = $ *equilibrium loads*

$P_{EST} = $ *initial estimate of the load vector*

$P_{RES} = $ *residual load vector*

$\bar{r}_p = $ *position vector of end p*

$\bar{r}_q = $ *position vector of end q*

$\{S_o\} = $ *initial negative gradient direction*

$T = $ *force in the member*

$T_2 = $ *member force in State 2*

$T_1 = $ *member force in State 1*

$[T_R] = $ *transformation matrix*

$U = $ *displacement vector*

$U_{12} = $ *vector of displacement increments from State 1 to State 2*

$U_{EST} = $ *initial estimate of the displacement vector*

$U_{COR} = $ *correction to the initial estimate of the displacement vector*

$\bar{u}_p$ = displacement vector of end p

$\bar{u}_q$ = displacement vector of end q

$\bar{u}_p, \bar{v}_p, \bar{w}_p$ = displacement from State 1 to State 2 at end p

$\bar{u}_q, \bar{v}_q, \bar{w}_q$ = displacement from State 1 to State 2 at end q

u_x = derivative of u with respect to x

V = volume of the member

W = work done on the member

X, Y, Z = global coordinate system

$\bar{X}, \bar{Y}, \bar{Z}$ = common reference coordinate system

$\bar{X}_p, \bar{Y}_p, \bar{Z}_p$ = coordinates of end p

$\bar{X}_q, \bar{Y}_q, \bar{Z}_q$ = coordinates of end q

δT_{12} = increment of member force from State 1 to State 2

ε = strain in the element

ε_p = strain due to prestress

ε_e = elastic strain in the member

ε_t = specified accuracy

Ξ = potential energy of the element

Π_p = total potential energy

ρ = step length

σ = stress in the element

σ_p = stress at the proportional limit of the member

σ_y = yield stress of the member

CONTENTS

FIGURES

Chapter I

INTRODUCTION

1.1 GENERAL

The presence in nature of such structures such as a spider's web and the tensile action of animal skin must have spurred man to develop tension systems for his use. Primitive roofing systems, such as tents, which used animal skin are even today considered perfect examples of a membrane roof with predominantly tensile forces. The idea of using a suspended cable was probably born in the tropical countries of Asia and Africa where there is an abundance of ropes made of vines and creepers. Available historical evidence suggests that natural ropes were the first materials to be used for constructing small suspension bridges. With the invention of iron and steel as structural materials, the analysis, design and construction of suspension bridges has developed to a considerable extent.

The impetus for the design of modern cable roofs could well be attributed to the success of suspension bridges. The main load bearing element in cable roofs is a steel cable which is occasionally combined with struts or light flexural members. Most of these roofs are usually light in weight and are thus limited in their performance as a roof due to their low gravity stiffness unless the system is pretensioned.

A cable roof can be defined as one in which a cable or a system of cables is used as a load carrying structural element. Cable roofs could be placed under three main categories:

1.   Cable supported roofs

2.   Cable suspended roofs

3.   Cable-cum-air supported roofs.

In cable supported roofs, the cables have only the auxiliary function of providing additional support for elements which are otherwise sufficently strong enough to carry a major portion of the load. In cable suspended roofs, the system of cables carries the roof load directly and has a primary structural function. The third class of roof, which is of interest in this study, is the air supported roof. These are mostly tent or balloon type structures supported by a combination of cables and air inflation.

The idea of using pneumatic principles for buildings was introduced as early as 1917. During World War II "radomes", air supported weather covers for radar installations, were constructed for the U.S. Air Force. Lightweight portable structures were also developed for military and commercial applications after this period. The Pan American Airways Pavilion at the 1958 Brussels World Fair further popularized the use of such structures. The inauguration of the first International Symposium on Pneumatic Structures in Stuttgart, Germany in 1967 was an indication of the growing interest in this field. The greatest exploitation of this type of structures was made at EXPO '70 in Osaka, Japan.

The most recent and notable construction of this type of structure was the building of B.C. Place Stadium in Vancouver, British Columbia and has an estimated capacity of 60000 people. The roof is inflated by means of 16 gigantic computer controlled fans.

The Pontiac Silver Dome is touted as the world's largest air supported structure. Located in Pontiac, Michigan, the stadium has a capacity of 80000 people. It covers an area of 38400 square metres with clear spans of 220 (722 ft.) and 168 metres. (551 ft.) The membrane is teflon coated fibre glass supported by an oblique net of 76 mm (3 in.) diameter cables. The fibre glass membrane spans between the cables forming quadrilateral panels. A model of this structure has been built in the laboratory to test and compare with the results of the various theoretical models generated on the computer.
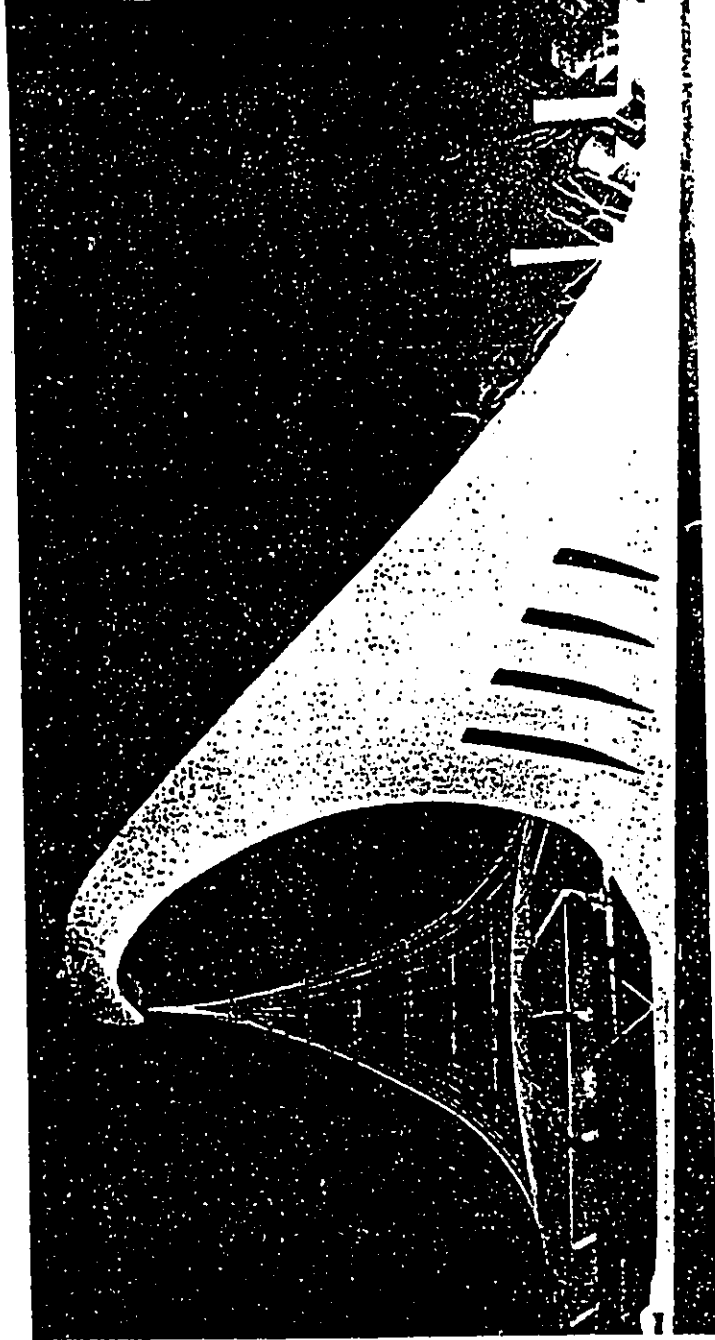
Fig. 1.1

AUSTRALIAN PAVILION AT THE 1970 INTERNATIONAL

EXPOSITION, OSAKA, JAPAN
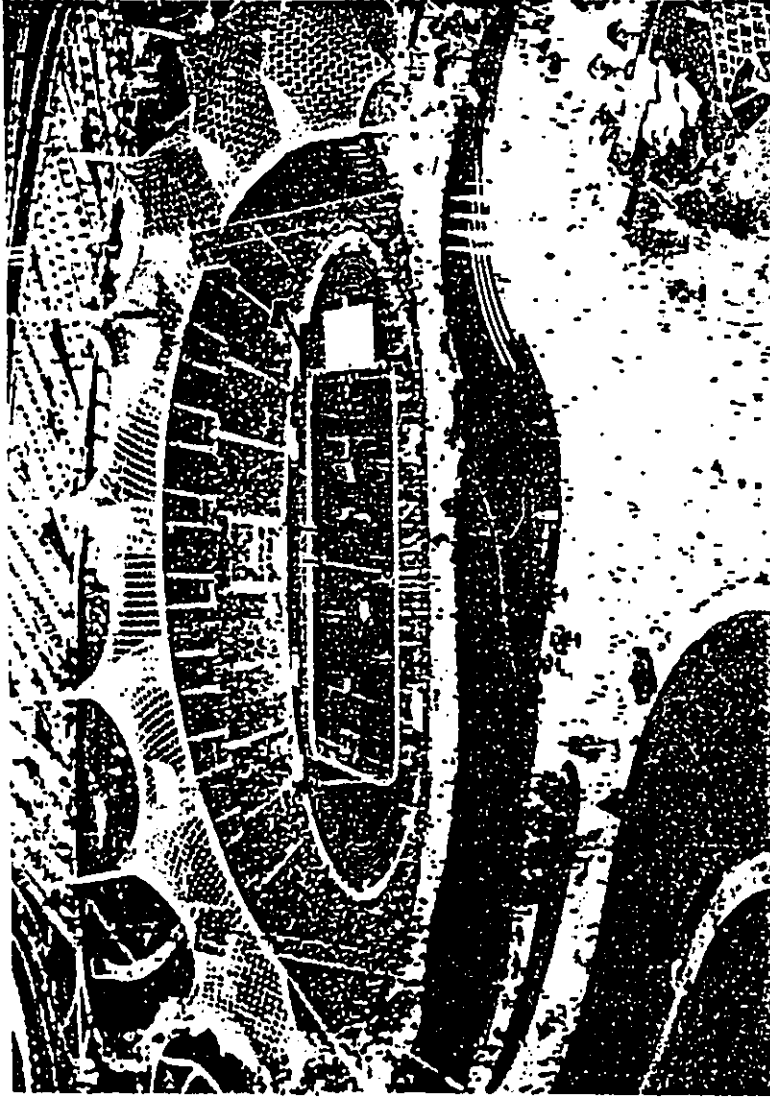
Fig. 1.2

SUSPENDED CABLE ROOF OVER THE 1972
MUNICH OLYMPIC STADIUM

Fig. 1.3

U.S. PAVILION AT THE 1970 INTERNATIONAL EXPOSITION

OSAKA, JAPAN
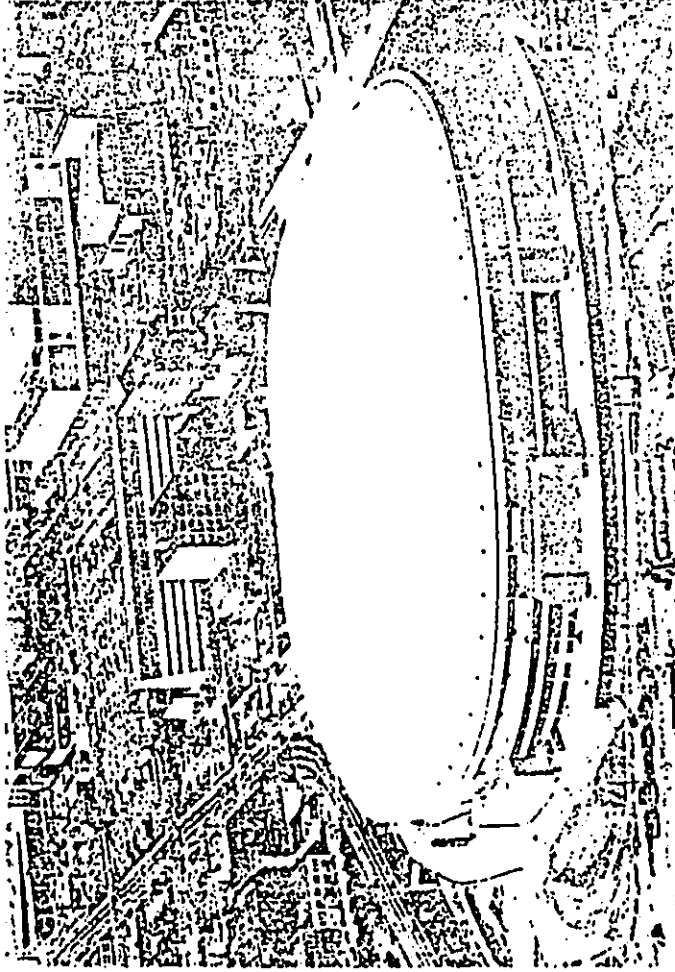
Fig. 1.4

AERIAL VIEW OF THE B.C. PLACE STADIUM,
VANCOUVER, BRITISH COLUMBIA

A cable exhibits a marked nonlinear behaviour (figure 1.5) when it is subjected to loads. This behaviour varies with the type of structure as well as loading. It has to follow the funicular curve of the loads and so undergoes large geometric adjustments especially when the loading is concentrated or unsymmetrical. Thus, geometric nonlinearity will occur in cables irrespective of material nonlinearity. The slope of the tangent to a load displacement curve at any point gives the value of the stiffness at that point.

The high degree of indeterminancy coupled with highly nonlinear behaviour makes the analysis of cable systems very complex. This is somewhat reduced by placing the types of analysis into two categories:

1.   a continous membrane and

2.   a discrete system.

If the structure is treated as a discrete system, as is done in most analytical methods, the solution consists of solving for flexibility and stiffness matrices. Most of these solutions assume that the supporting structure of the roof is rigid i.e., the compression ring around the edges of the roof.

1.2   MERITS

Cable roofs are becoming increasingly popular due to their inherent architectural advantages and the increased demand for large column-free spaces. The economic constructional advantages and the availability of high strength steels add to their popularity. Also, the accessibility to computers with huge amounts of memory has made a numeric analysis of such structures financially feasible.

The contribution of cable roofs to the overall economy of the structure is due to the efficiency with which cables carry their load. A cable is a uniform stress member that has a very high tensile strength. Its own weight is also very small compared to other roof components.
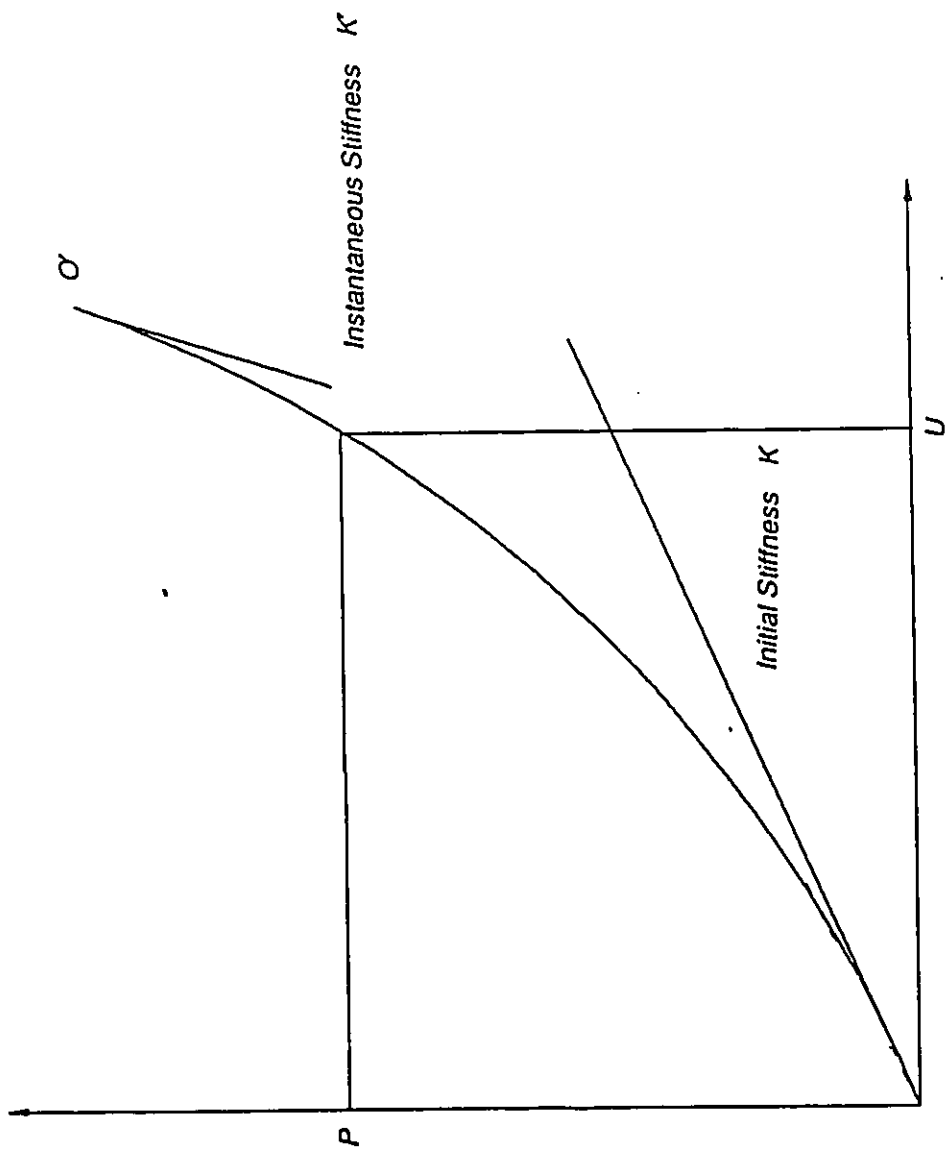
FIG. 1.5

**A TYPICAL NONLINEAR RESPONSE
CURVE OF A CABLE**

Since it is a tensile structure, it fully utilizes the properties of steel with its high tensile strength. There is also no problem of buckling in such structures.

The economy of the structure is further justified by the decrease of erection costs due to the elimination of most of the scaffolding. The total volume enclosed by a cable roof is usually less than that of a shell or truss structure covering the same area which results in savings in terms of heating, ventilating and air conditioning costs. The maintenance requirements with respect to other roof systems are also minimal.

From an aesthetic point of view, the shapes can be made architecturally appealing. This has been proved by the fact that several cable roof buildings with artistic architectural facades have sprung up in the past two decades.

1.3   DEFICIENCIES OF AIR SUPPORTED STRUCTURES

However, such structures do have disadvantages. Primarily, the light weight and small stiffness of the roof make it subject to unacceptable dynamic and aero-elastic responses. Live loads create larger deformations than in conventional systems. This flexibility requires more complicated connections and water proofing details. The boundary supports are large and usually account for more than fifty percent of the total dead weight of the roof structures.

The greatest inherent drawback in such structures could be the difficulty in the exact analysis due to the relationship between tension and geometry. Conventional linear analysis which assumes small elastic deformations and displacements is not applicable unless it is modified by a piecewise incremental analysis which in itself causes residual displacements to be unaccounted.

Special design considerations like anchorages to resist tensile loads, air lock systems to ensure air tightness and mechanical systems for inflation pressure need to be taken care of.

In addition, the roof fabric is susceptible to storm damage caused by a strong wind especially when it is not fully inflated. There is also the danger of unusual loads in a single panel leading to the failure of the entire roof system.

1.4   DEVELOPMENTS IN ANALYSIS

An extensive literature survey was undertaken to study various methods used in the analysis of cable net structures. Basically, most of the analysis methods incorporated the finite element procedure.

Siev[27] presented a method for the determination of the displacements of a general net. The effect of horizontal displacements and changes in geometry were included in the derivation of the equations. It was also suggested that piecewise application of load increments be done in cases of high geometric nonlinearity. The displacements were arrived at using the linear theory and subsequently corrected by iteration. In [28], he presented an analytical and experimental study of prestressed suspended cable roofs using the theory presented in [27].

Thornton and Birnsteil [30] derived nonlinear equations for a general three dimensional unstiffened suspension structure composed of members resisting axial load only. They presented two methods for the solution of nonlinear equations, the method of continuity and the incremental load method.

Baron and Venkatesan[1], using the direct stiffness method, analyzed geometrically nonlinear structures composed of elastic members capable of resisting axial forces. The theory employs the concept of a geometric stiffness matrix in obtaining an estimate, and then successively adjusting the stiffness matrix, forces and the displacements to obtain a set of r⁻ ..s compatible with the final dimensions of the structure.

Kar and Okazaki[15] studied the convergence of solutions in highly nonlinear cable net problems. They recognize that an efficient method of solution should have two important characteristics: (1) the equations and the associated iterative method should ensure convergence; and (2) the convergence should be rapid for the method to be economically efficient. They also present a new iterative technique in [15].

Saafan[25] suggests halving the displacements in each cycle of iteration to improve convergence. However, he states that judgement is needed to determine the number of times the displacement should be halved and proposes a scheme similar to the one used by Haug and Powell[11]. He concludes that only a fraction of the residual loads need to be applied to have good convergence.

The energy search approach has also been applied successfully to the nonlinear analysis of space type structures. It consists of including geometric nonlinearities by using nonlinear strain displacement equations to construct the potential energy for each of the finite elements. A numerical solution is obtained by seeking the minimum of the total potential energy for the assemblage of the finite elements representing the structure. This approach was developed successfully by Bogner, Mallet, Minich and Schmidt[3] and has proven to be suitable for nonlinear problems[22] including instability analysis. Various methods have been used to minimize the potential energy. Bogner used the Fletcher-Powell[6] variable metric search technique. Buchholdt[4] solved the nonlinear equations by the method of steepest descent. Again, Buchholdt[4] used the method of conjugate gradients to minimize the potential energy along with a scaling technique to improve convergence.

Monforton and El-Hakim[24] applied the energy search method to the analysis of general pin-ended truss and cable structures. Geometric and material nonlinearities were directly incorporated within the formulation thereby accounting for large strains and displacements as

well as configuration changes as a structural response. Solutions for the minimum potential energy were generated using the conjugate gradient method. Monforton[23] also adapted the energy search method to microcomputer analysis using a BASIC program.

1.5 SCOPE OF THE PRESENT STUDY

The present study encompasses a comparison of experimental and theoretical analysis of air supported cable roof structures. The energy search approach is compared to five standard one-step solution techniques and further corroborated by an experimental study. For the purpose of comparison, solution techniques presented by Baron and Venkatesan[1], Kar and Okazaki[15] were used.

The conjugate gradient method of function minimization proposed by Hestenes and Stiefel[12] and its extension by Fletcher and Reeves[7] was used to minimize the potential energy function in the energy search method. A scaling technique proposed by Fox and Stanton[8] was used to improve the rate of convergence.

In [1], Baron and Venkatesan present three different solution techniques and a modification for two of them. In each of them, an estimate for the displacement vector is constantly sought. The equilibrium load at each of the nodes is calculated for the geometry obtained with the estimated displacement vectors. These equilibrium loads are then compared with the actual applied nodal loads. Convergence is said to have been obtained when the difference between the equilibrium nodal loads and the actual applied loads are within a specified tolerance limit.

Kar[15], in his iterative technique, represents an effort to scale down the overestimated displacements which are quite common in linearized solution methods for highly nonlinear cable net structures. This technique yields a closer approximation to the actual displacements.

All of these methods are further discussed in Chapter 2.

For the experimental study, a 1:120 scale model of the Pontiac Silver Dome was constructed based on blueprints obtained from the Department of Public Works, Pontiac, Michigan, U.S.A. The model was tested for nodal displacements for various distributed loads on the roof. This is further explained in Chapter 3.

Finally, in Chapter 4, the evaluation and discussion of results are presented. The advantages and drawbacks of various methods vis-a-vis the experimental study are highlighted.

In Chapter 5, conclusions are drawn and recommendations for further points of research are presented.

Chapter II

THEORY

In this chapter, the different methods of analysis are explained. Mathematical models for each of the methods are developed using the finite element procedure. All of these models take into account the deformed geometry of the structure permitting large nodal displacements quite commonly observed in air supported structures.

The principal assumptions in the development of the theoretical models are as follows:

1.   In the theoretical models, the cable network consists of cables spanning between two nodes. In the experimental model, the network is formed by an oblique net formed by the intersection of 9 cables spanning in each direction. The cables are soldered at their intersections to form nodes.

2.   The membrane is assumed to be uniformly loaded in any one panel. The total force is equally distributed to its enclosing nodes.

3.   Members are straight and prismatic between joints.

4.   Stressing the member does not change its cross-sectional area.

5.   The joints of the structure are frictionless.

6.   All loads are conservative, in that their original directions in space are preserved.

7.   A conservative simplification in the analysis of a cable and fabric roof can be made by considering only the cables according to Malcolm and Glockner[21]. Hence the behaviour of the structure can be adequately studied if only the cables are considered to contribute to the structural behaviour.

8.   The boundary supporting structure is assumed to be rigid and does not contribute to the stiffness of the cables.

2.1   ENERGY SEARCH METHOD

The fundamental relations to be incorporated into the various methods have to be first established before outlining the method in detail.

2.1.1   DEFORMATION-DISPLACEMENT RELATION

The undeformed length l_1, (figure 2.1) of a discrete element of a typical tension member is defined in figure 2.1 by the initial position vectors of the element joints with respect to a common reference coordinate system $(\bar{X}, \bar{Y}, \bar{Z})$ as

$$L_1 = \bar{r}_q - \bar{r}_p \tag{2.1}$$

$$= \sqrt{(\bar{X}_q - \bar{X}_p)^2 + (\bar{Y}_q - \bar{Y}_p)^2 + (\bar{Z}_q - \bar{Z}_p)^2} \tag{2.2}$$

The deformed length l_2, after the joints undergo displacements $(\bar{u}, \bar{v}, \bar{w})$ under loading is then expressed as

$$L_2 = (\bar{r}_q + \bar{u}_q) - (\bar{r}_p + \bar{u}_p) \tag{2.3}$$

where $\bar{u}_p$ and $\bar{u}_q$ are the displacement vectors of the element joints. It can be rewritten as:

$$L_2 = \left\{ \left[(\bar{X}_q + \bar{u}_q) - (\bar{X}_p + \bar{u}_p)\right]^2 + \left[(\bar{Y}_q + \bar{v}_q) - (\bar{Y}_p + \bar{v}_p)\right]^2 + \left[(\bar{Z}_q + \bar{w}_q) - (\bar{Z}_p + \bar{w}_p)\right]^2 \right\}^{1/2} \tag{2.4}$$
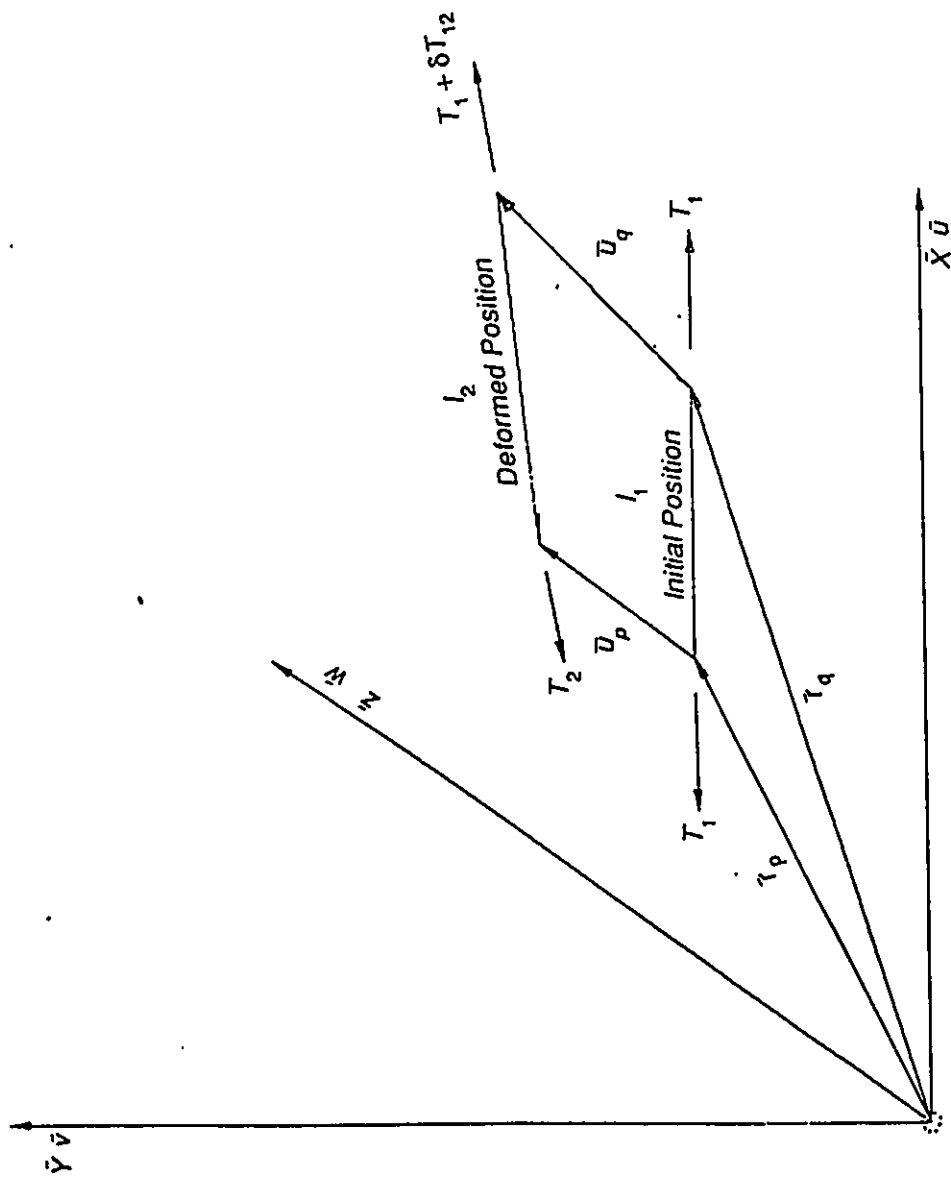
FIG. 2.1

DISCRETE ELEMENT OF A TENSION MEMBER

2.1.2   ANALYSIS OF ELASTIC TENSION MEMBERS

2.1.2.1   STRAIN-DEFORMATION RELATION

There is no deformation in the transverse direction for tension members. The axial deformation can be expressed as

$$\varepsilon = \frac{\partial u}{\partial x} \tag{2.5}$$

2.1.2.2   ELEMENT STRAIN ENERGY

The strain energy density for elastic tension members is defined as:

$$d\Xi = \int_{0}^{\varepsilon + \varepsilon_p} \sigma \, d\varepsilon \tag{2.6}$$

where ε_p is the strain due to prestress.

Integration of (2.6) over the volume V of the elements results in the following expression for the strain energy in terms of the strain ε.

$$\Xi = \frac{1}{2} \int_{V} (\varepsilon + \varepsilon_p) \sigma \, dV \tag{2.7}$$

$$= \frac{E}{2} \int_{V} (\varepsilon + \varepsilon_p)^2 \, dV \tag{2.8}$$

Substituting (2.5) into (2.8) and integrating over the cross-section area yields:

$$\Xi = \frac{AE}{2} \int_{0}^{l} (u_x + \varepsilon_p)^2 \, dx \tag{2.9}$$

The governing differential equation of (2.9) is given by El-Hakim[5] as:

$$\frac{\partial}{\partial x}(u_x + \varepsilon_p) = 0 \tag{2.10}$$

Upon integrating (2.10),

$$u_x + \varepsilon_p = K_1 \tag{2.11}$$

where K_1 is a constant with respect to x and can be determined by integrating (2.11) over l_2.

$$\int_0^{l_2} K_1 \, dx = \int_0^{l_2} (u_x + \varepsilon_p) \, dx \tag{2.12}$$

$$K_1 = 1 - \frac{l_1}{l_2} + \varepsilon_p \tag{2.13}$$

The force in the member is constant and can be written as:

$$T = AEK_1 \tag{2.14}$$

The strain energy in the tension elements is obtained by substituting (2.11) and (2.13) into (2.10) as:

$$\Xi = \frac{AE}{2} l_2 K_1^2 \tag{2.15}$$

2.1.3   ANALYSIS OF INELASTIC TENSION MEMBERS

The inelastic material behaviour is considered in deriving the expressions for the element strain energy. The mathematical model proposed by Kumanan in [19] which uses a compound curve initially linear up to the proportional limit followed by a parabola up to the ultimate stress is adopted to derive the expression.

The parabolic curve of Kumanan (figure 2.2) can be represented by the following equation:

$$\sigma^2 + 2g\varepsilon + 2f\sigma + c = 0 \tag{2.16}$$

and is assumed to have its axis parallel to the ε axis. The constants are evaluated by the following relations:

$$g = -250(\sigma_y - \sigma_p)^2 \tag{2.17}$$

$$f = \frac{g}{E} - \sigma_p \tag{2.18}$$

$$c = \sigma_p^2 \tag{2.19}$$

The yield stress σ_y is determined by the 0.2% offset drawn to the linear portion of the curve. σ_p is determined by loading the cable until it starts to deviate from the linear stress-strain curve. The point of deviation is noted as the stress in the cable at the proportional limit.

The elastic modulus of the seven strand cable was determined to be 145000 MPa and the yield stress and ultimate stress were determined as 1625 MPa and 2160 MPa respectively.

2.1.3.1   STRAIN-DEFORMATION RELATION

In case of inelastic tension members, the axial deformation is expressed up to the second term of Taylor's expansion to account for the larger strains in the inelastic range. Therefore,

$$\varepsilon = \frac{\partial u}{\partial x} + \frac{1}{2}(\frac{\partial u}{\partial x})^2 \tag{2.20}$$

Stress σ

$σ_{ult}$

σ

$σ_y$

$σ_p$

Equation of the curve

$σ^2 + 2gε + 2fσ + c = 0$

dε

$ε_p$

$ε_y$

$ε + ε_p$

$ε_{ult}$
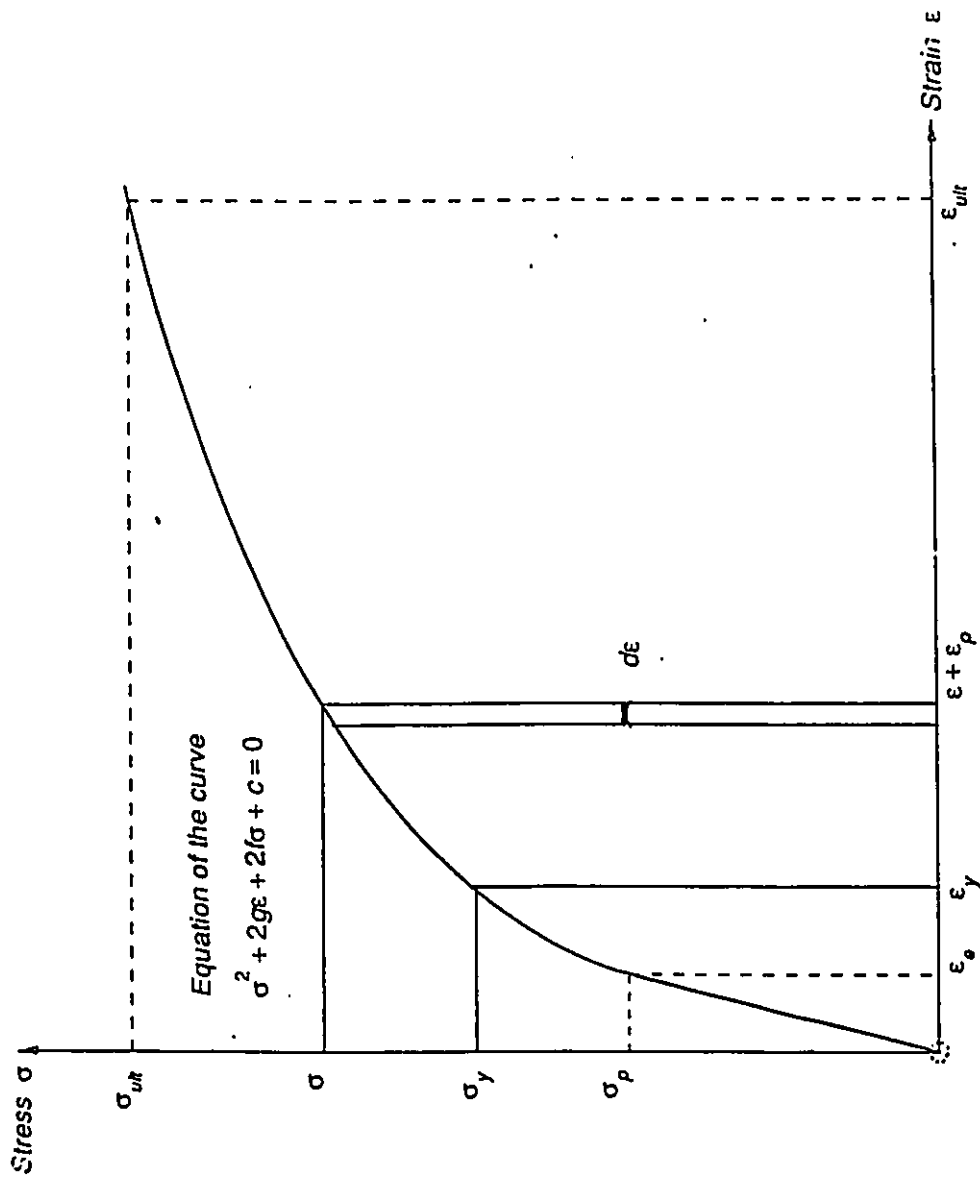
Strain ε

FIG. 2.2

STRESS-STRAIN CURVE FOR A CABLE

2.1.4   ELEMENT STRAIN ENERGY

This is defined by

$$d\Xi = \frac{1}{2}E\varepsilon_e^2 + \int\limits_{\varepsilon_e}^{\varepsilon + \varepsilon_p} \sigma \, d\varepsilon \tag{2.21}$$

where σ is given by (2.16).

The resulting strain energy was derived by El-Hakim in [5] as:

$$\Xi = Al_2\left[\frac{1}{2}E\varepsilon_e^2 - f(K_2 - \varepsilon_e) - \frac{1}{3g}(f^2 - 2gK_2 - c)^{\frac{3}{2}} + \frac{1}{3g}(f^2 - 2g\varepsilon_e - c)^{\frac{3}{2}}\right] \tag{2.22}$$

where

$$K_2 = \varepsilon_{ps} + (1 - \frac{l_1}{l_2}) + \frac{1}{2}(1 - \frac{l_1}{l_2})^2 \tag{2.23}$$

2.1.4.1   ANALYTIC GRADIENT OF THE ELEMENT STRAIN ENERGY

The gradient of the element strain energy is defined as:

$$\{G_E\}^T = \left\{\frac{\partial \Xi}{\partial \bar{u}_p}, \frac{\partial \Xi}{\partial \bar{v}_p}, \frac{\partial \Xi}{\partial \bar{w}_p}, \frac{\partial \Xi}{\partial \bar{u}_q}, \frac{\partial \Xi}{\partial \bar{v}_q}, \frac{\partial \Xi}{\partial \bar{w}_q}\right\} \tag{2.24}$$

The expressions for the gradient are given by Monforton and El-Hakim in [24] and presented in Appendix A.

2.1.5   TOTAL POTENTIAL ENERGY OF THE STRUCTURAL SYSTEM

The total potential energy of an assembly of n tension members is defined as:

$$\Pi_p = \sum_{i=1}^{n} \Xi^{(i)} - W \tag{2.25}$$

where $\Xi^{(i)}$ is the strain energy of the ith element given by (2.15) or (2.22) depending on the elastic or inelastic state of the member. W is the external work done by the loads applied at the joints of the structure.

Since the total potential energy of the structural system is a function of n variables representing the nodal displacements of the structure, the total potential energy of the structural system can be written in terms of an n-component displacement vector U as:

$$\Pi_p(U) = \Xi(U) - W(U) \tag{2.26}$$

where $\Xi(U)$ is the sum of the element strain energies in terms of the independent displacement degrees of freedom U of the system. The external work done W is given by

$$W(U) = U^T P \tag{2.27}$$

where P is the vector of applied loads associated with each degree of freedom in U.

2.1.6   ANALYTIC GRADIENT OF THE TOTAL POTENTIAL ENERGY

The gradient vector $\{G\}$ of the total potential energy is the vector sum of the gradient vector of the element strain energy minus the applied load vector P. Therefore,

$$\{G\} = \sum_{i=1}^{n} \left\{ G_E^{(i)} \right\} - P \tag{2.28}$$

where $\{G_E^{(i)}\}$ is given by the expression in Appendix A.

2.2   STIFFNESS METHOD OF ANALYSIS

This method can be stated in terms of an estimate plus a correction to account for the nonlinear effects of changes in the geometry of a given structure. It uses a tangent stiffness matrix to arrive at an estimate of displacements and then adjusts the stiffness matrix to obtain forces and displacements compatible with the final dimensions of the structure.

In the theory of linear elastic structures, the relations between the external loads P and the joint displacements U can be represented in matrix form as:

$$P = K_E U \tag{2.29}$$

where K_E is the elastic stiffness matrix of the structure. Here the displacements are considered small and the contributions of the member forces to balance the external loads are stated in terms of the initial geometry of the structure.

In the nonlinear theory of structures, when displacements are large, equation (2.29) is no longer valid because the strain-displacement relationship is nonlinear. The equations of joint equilibrium need to be written in terms of the final geometry of the structure which further invalidates the above equation.

If the changes between two deformed states 1 and 2 are considered, an incremental equation of joint equilibrium can be written as:

$$P_{12} = K_{12} U_{12} \qquad (2.30)$$

where P_{12} is a column vector of loads from state 1 to state 2 and U_{12} is a column vector of the increment of displacements.

Solutions to equation (2.30) can be obtained by using various techniques. The final solution is realized by successive estimates and corrections. To facilitate the above, equation (2.30) is rewritten as:

$$P = P_{EST} + P_{COR} \qquad (2.31)$$

$$= K_{EST} U_{EST} + K_{COR} U_{COR} \qquad (2.32)$$

where P_{EST}, K_{EST} and U_{EST} are initial estimates of load vector, stiffness matrix and displacement vector and P_{COR}, K_{COR} and U_{COR} are corrections to the above.

A linearized version of the nonlinear stiffness matrix is produced by rewriting (2.32) as:

$$P = [K_L + K_{NL}] [U_L + U_{NL}] \qquad (2.33)$$

where the subscripts L and NL indicate the linear and nonlinear portion respectively.

The linearized version k_L, of the stiffness of an element is known as the tangent stiffness k_T,

of an element. It is composed of k_E, the stiffness of an element as per linear elastic theory and k_G, the geometric stiffness matrix.

The matrix k_E is associated with the deformations in each element, whereas k_G incorporates the changes in the geometry of an element.

The geometric and tangent stiffness matrices are presented in detail in Appendix B.

2.3   SOLUTION TECHNIQUES

Four techniques are presented to solve the nonlinear equa...on:

$$P = K U \qquad (2.34)$$

Each of the techniques is elaborated with the help of schematic diagrams.

The estimate U_{EST} sought in each of the above techniques is used to calculate the strain ε from the relation:

$$(1 + \varepsilon)^2 = 1 + \frac{2}{l_1^2} L_1^T \Delta U_{EST} + \frac{1}{l_1^2} \Delta U_{EST}^T \Delta U_{EST} \qquad (2.35)$$

where $\Delta U_{EST}^T = [u_q - u_p \quad v_q - v_p \quad w_q - w_p]_{EST}^T$.

The above equation is fully developed in Appendix C.

The magnitude of the member forces are given by:

$$T = T_1 + \delta T_{12} \qquad (2.36)$$

where $\delta T_{12} = \frac{AE}{l_o} \varepsilon l_1$ by Hooke's Law.

Thus the joint loads that are in equlibrium with the member forces are calculated as:

$$P_{EQ} = \sum \frac{T}{l_2} L_2 = 0 \qquad (2.37)$$

where $l_2 = l_1(1 + \varepsilon)$

$$L_2 = L_1 + \Delta U_{EST}$$

The above equations yield a set of equivalent joint loads for which the displacements matrix U_{EST} is an exact solution. This can be further extended to state that if the residual load vector given by:

$$P_{RES} = P - P_{EQ} \tag{2.38}$$

is zero or within a specified tolerance, the exact solution given by U is equal to U_{EST}.

The methods of solution described in the following paragraphs are utilized to achieve the above.

2.3.1   TANGENT STIFFNESS METHOD

This method (figure 2.3) is very similar to the Newton-Raphson technique for the solution of nonlinear equations. The final displacement vector U is a summation of estimates ΔU_{EST} obtained in each cycle of iteration until the condition of convergence given below is met.

$$P_{RES} \leq \varepsilon_1 \tag{2.39}$$

The method can be expressed by the following equation:

$$P_{RES}^{(i-1)} = K_T^{(i-1)} \ \Delta U^{(i)} \qquad i = 1,2,3,\ldots\ldots \tag{2.40}$$

where K_T is the tangent stiffness matrix formed by using the joint coordinates and member forces at the end of each cycle of iteration and is formed by the assemblage of k_T, the stiffness matrix of each element:

$$k_T = \frac{AE}{l_o}[D][D]^T + \frac{T}{l}\left[I - [D][D]^T\right] \tag{2.41}$$

where $[D]^T = [l \ \ m \ \ n]^T$, the direction cosines in the $x, y,$ and z directions respectively.

After each cycle of iteration P_{RES} is calculated using equations (2.35) to (2.38) until it satisfies the condition of equation (2.39). The flow chart for this method is given in figure 2.4.

✱ *Shift deformed geometry of this point and use remaining load vector as the applied vector*

P

U

U_{EST}

ΔU_{EST}
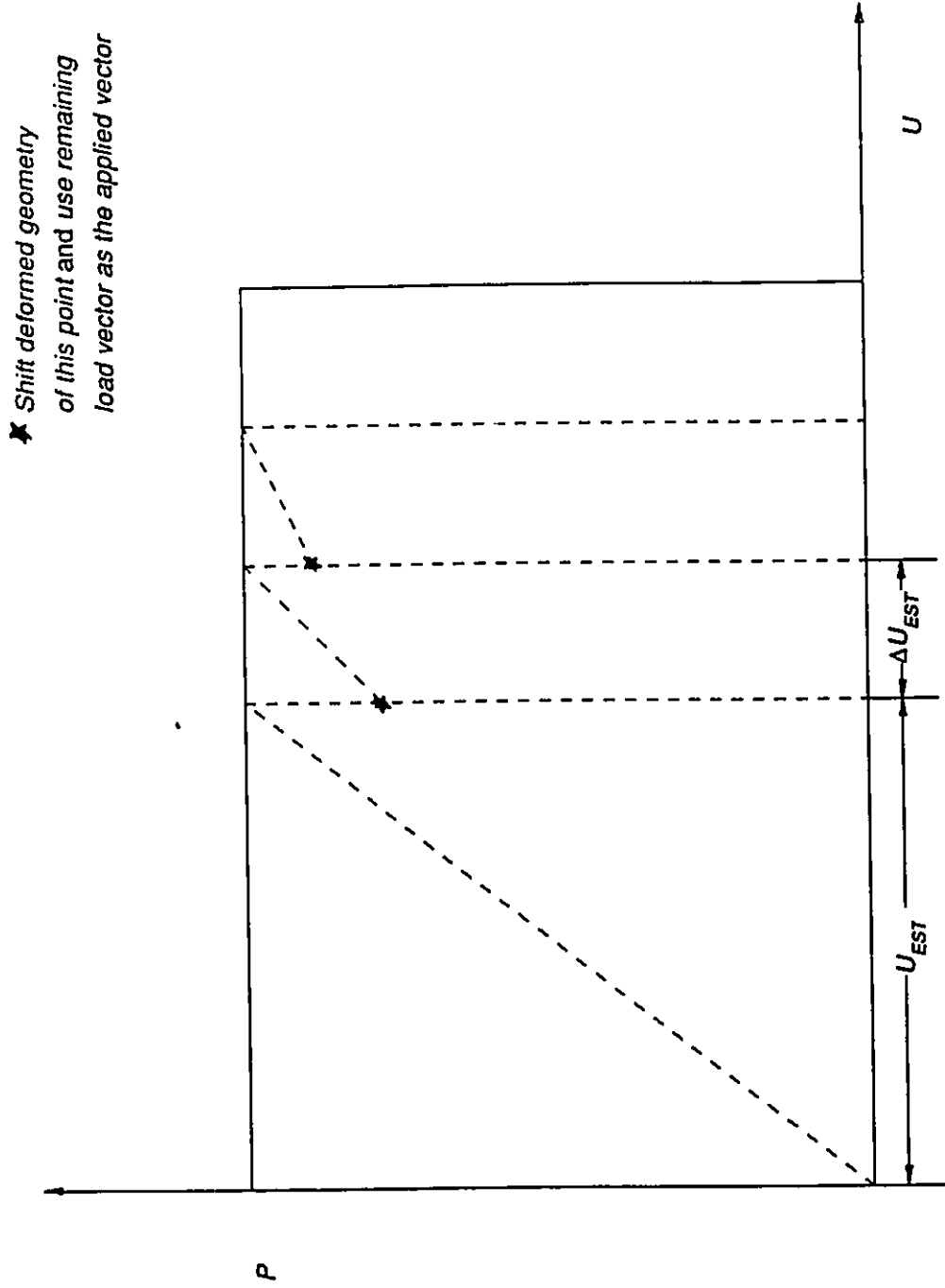
FIG. 2.3

SCHEMATIC DIAGRAM FOR THE TANGENT STIFFNESS METHOD
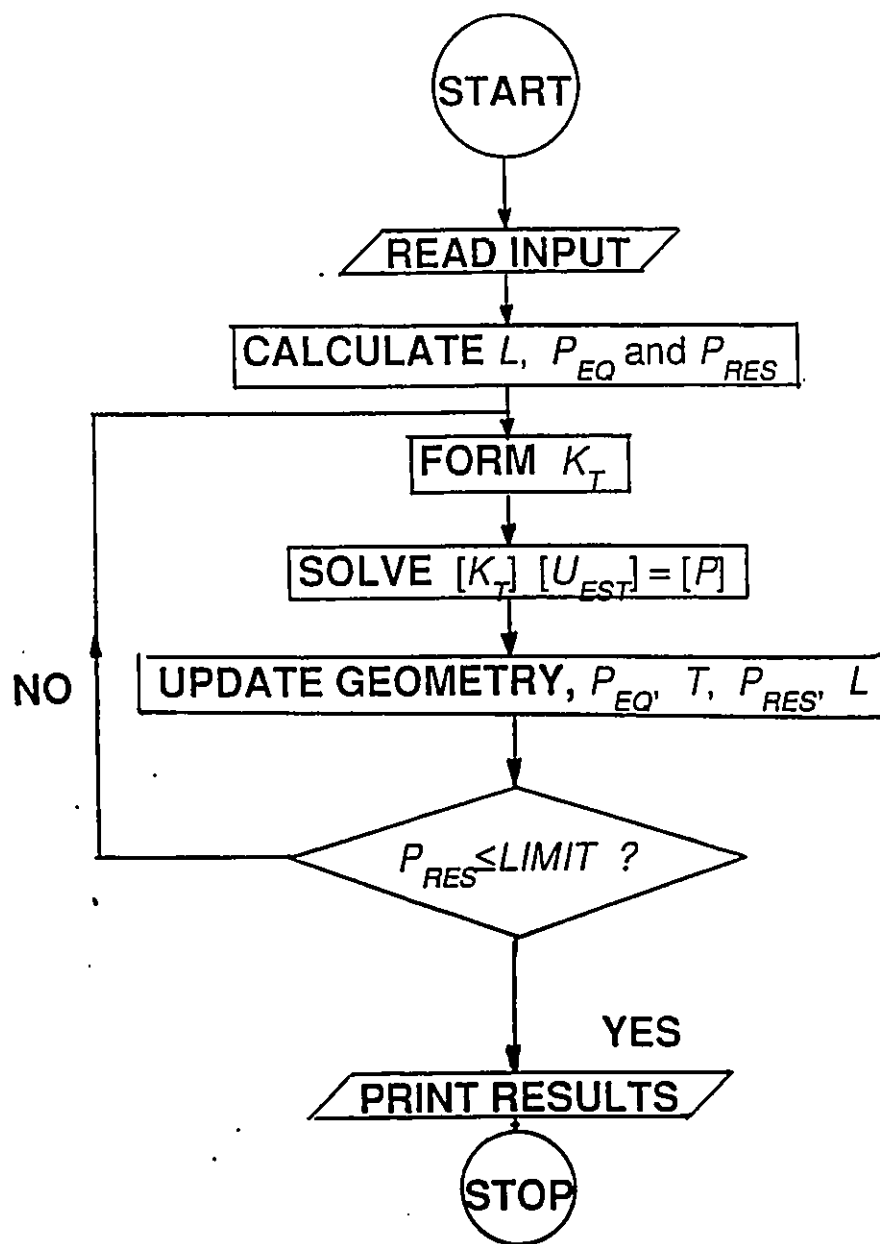
FIG. 2.4

FLOW CHART FOR TANGENT STIFFNESS METHOD

2.3.2 SECANT STIFFNESS METHOD

This method is the extended version of the tangent stiffness method in that that it takes into account the nonlinear behaviour of the structure. It is schematically illustrated in figure 2.5. The method is expressed by the following equation:

$$P_{RES} = K_s \Delta U_{EST} \tag{2.42}$$

$$= K_L + K_{NL}^{(i-1)} U_{EST}^j \qquad i = 1,2,3,\dots\dots\dots \tag{2.43}$$

where K_L is equal to K_T which is explained in Sec 2.2.1 and K_{NL} is the nonlinear component of the stiffness matrix consisting of an assemblage of the nonlinear element stiffness matrix k_{NL} given by:

$$k_{NL} = \left(\frac{AE}{l_o} \varepsilon - \frac{T_1}{l_1} \right) \left[C_1 + C_2 [D][D]^T + C_3 [D] \frac{\Delta U^T}{l} \right] \tag{2.44}$$

where

$$C_1 = \frac{\varepsilon}{1 + \varepsilon} \tag{2.45}$$

$$C_2 = -\frac{\varepsilon(3 + \varepsilon)}{(1 + \varepsilon)(2 + \varepsilon)} \tag{2.46}$$

$$C_3 = \frac{1}{(1 + \varepsilon)(2 + \varepsilon)} \tag{2.47}$$

The above equations are detailed in Appendix C.

In this method, the nonlinear part of the secant stiffness matrix K_{NL} is successively modified during each cycle of iteration as well as P_{RES}, the residual load vector, using equations (2.35) to (2.38) until the condition outlined in equation (2.39) is met. The flow chart for this method is presented in figure 2.6.
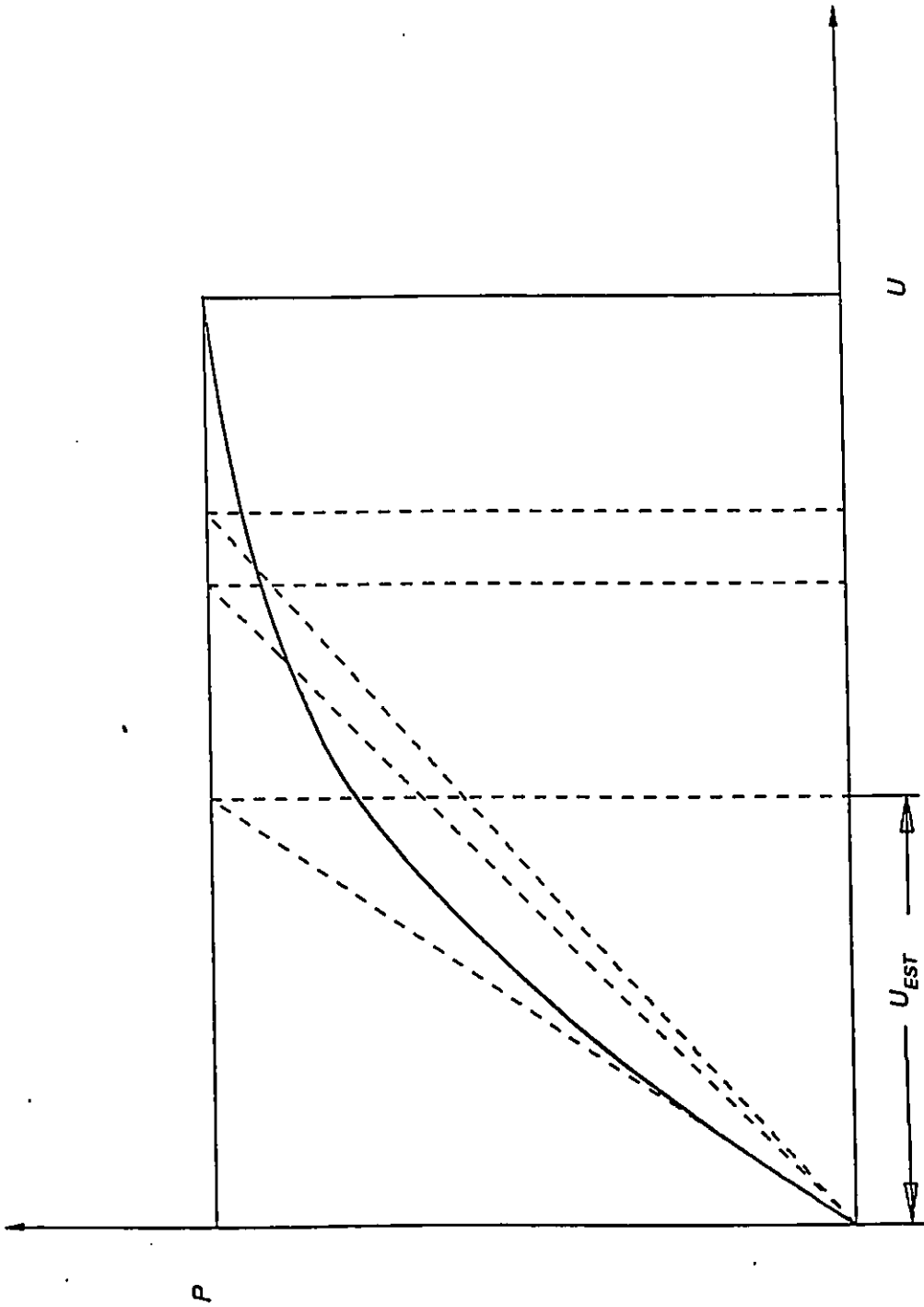
FIG. 2.5

SCHEMATIC DIAGRAM FOR THE
SECANT STIFFNESS METHOD
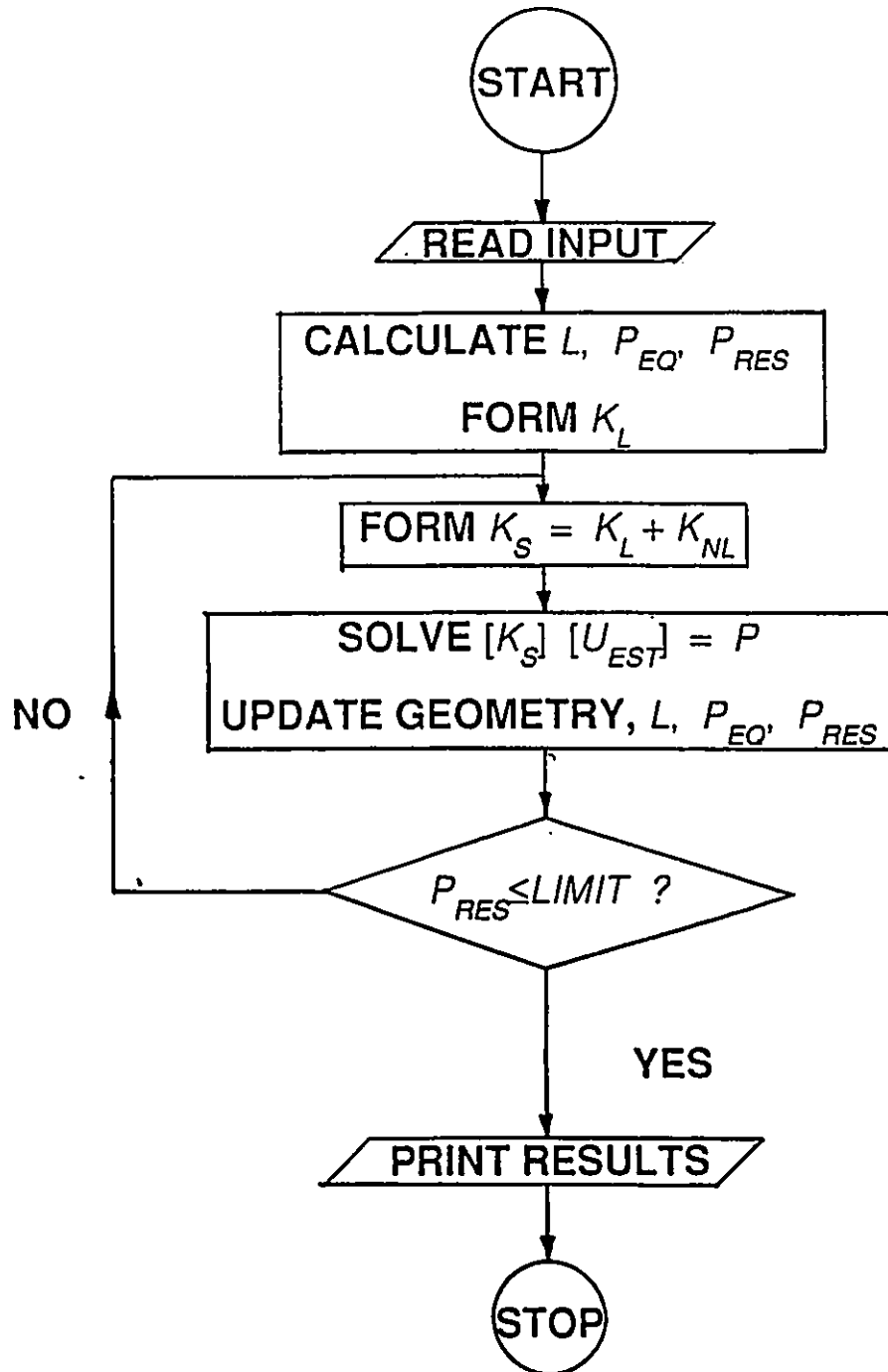
FIG. 2.6

FLOW CHART FOR SECANT STIFFNESS METHOD

2.3.3 KAR'S MODIFICATION METHOD

In highly nonlinear cable net behaviour, the equilibrium load based on a linear solution may be so different that the initial estimates oscillate uncontrollably which results frequently in a nonconvergent situation. Kar's method[15] is an effort to scale down the overestimated displacements to yield a closer approximation to the correct solution and hasten convergence.

Here, the displacement vector ΔU_{EST} obtained as the solution of the linearized equation (2.39) is modified by the ratio of the largest applied load at any cycle of iteration to its corresponding equilibrium load calculated on the basis of equatuon (2.39). The largest applied load is used as a criterion since it is assumed to have the most severe effect on the behaviour of the structure.

The method is outlined in figure 2.7 which represents a load-displacement curve of a joint with the largest applied load. The displacement u_1 is the initial estimate obtained as a solution to the linear equation (2.39). This is modified to u_1' by multiplying it with the ratio of the applied load P to the corresponding equilibrium load P_{EQ}. The modified displacement is used to calculate the new equilibrium load P_{EQ}' using equations (2.35) to (2.38). The flow chart is presented in figure 2.8.

Thus the residual load is obtained as:

$$P_{RES} = P - P_{EQ}' \tag{2.48}$$

and the process repeated in the next iteration until the convergence conditions of equation (2.39) are met.

Shift deformed geometry
of this point and use remaining
load vector as the applied vector

P_{EO}

P'_{EO}

P

P_{EO}

P_{RES}

$P_{EO} - P'_{EO}$

$u'_1 \leftarrow \Delta u$

$u'_2 \leftarrow \Delta u$

Δu

u'_1

u_2

u_1
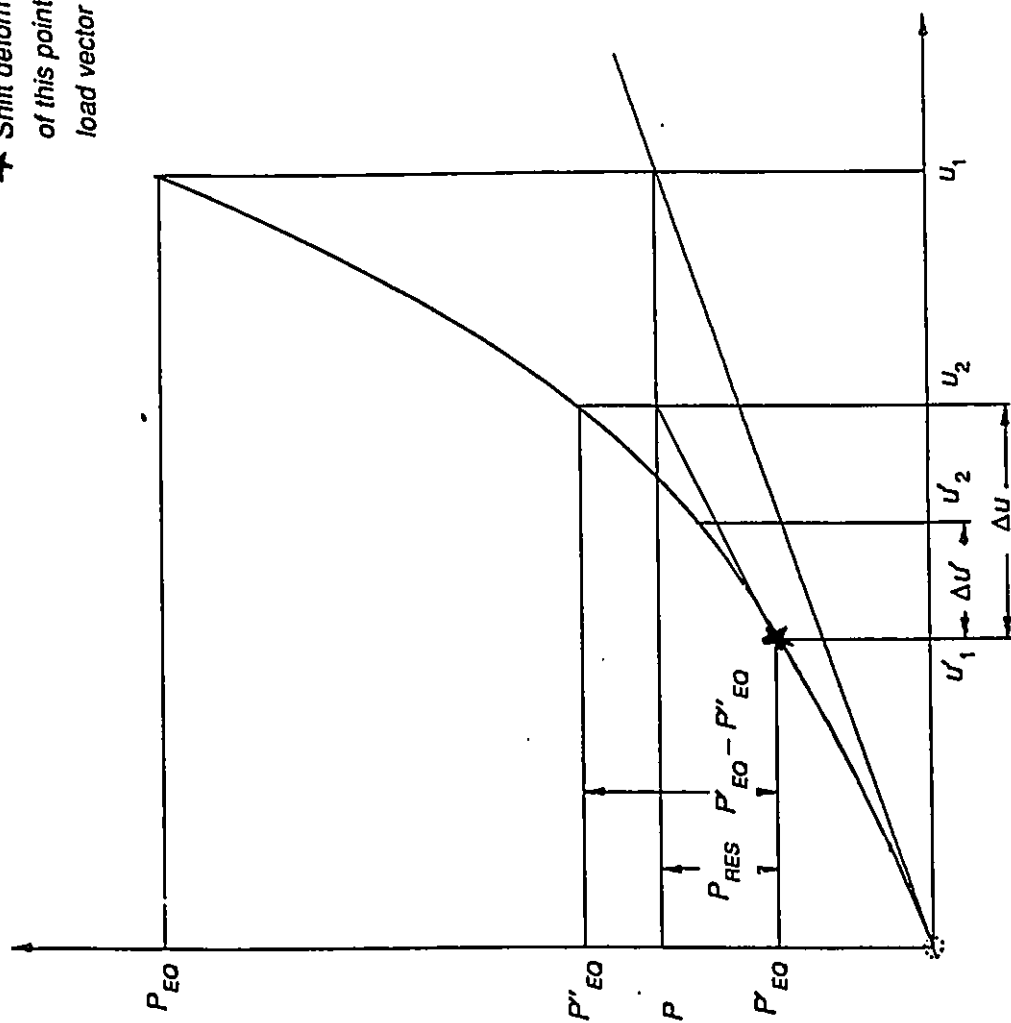
FIG. 2.7

SCHEMATIC DIAGRAM FOR KAR'S

MODIFICATION METHOD
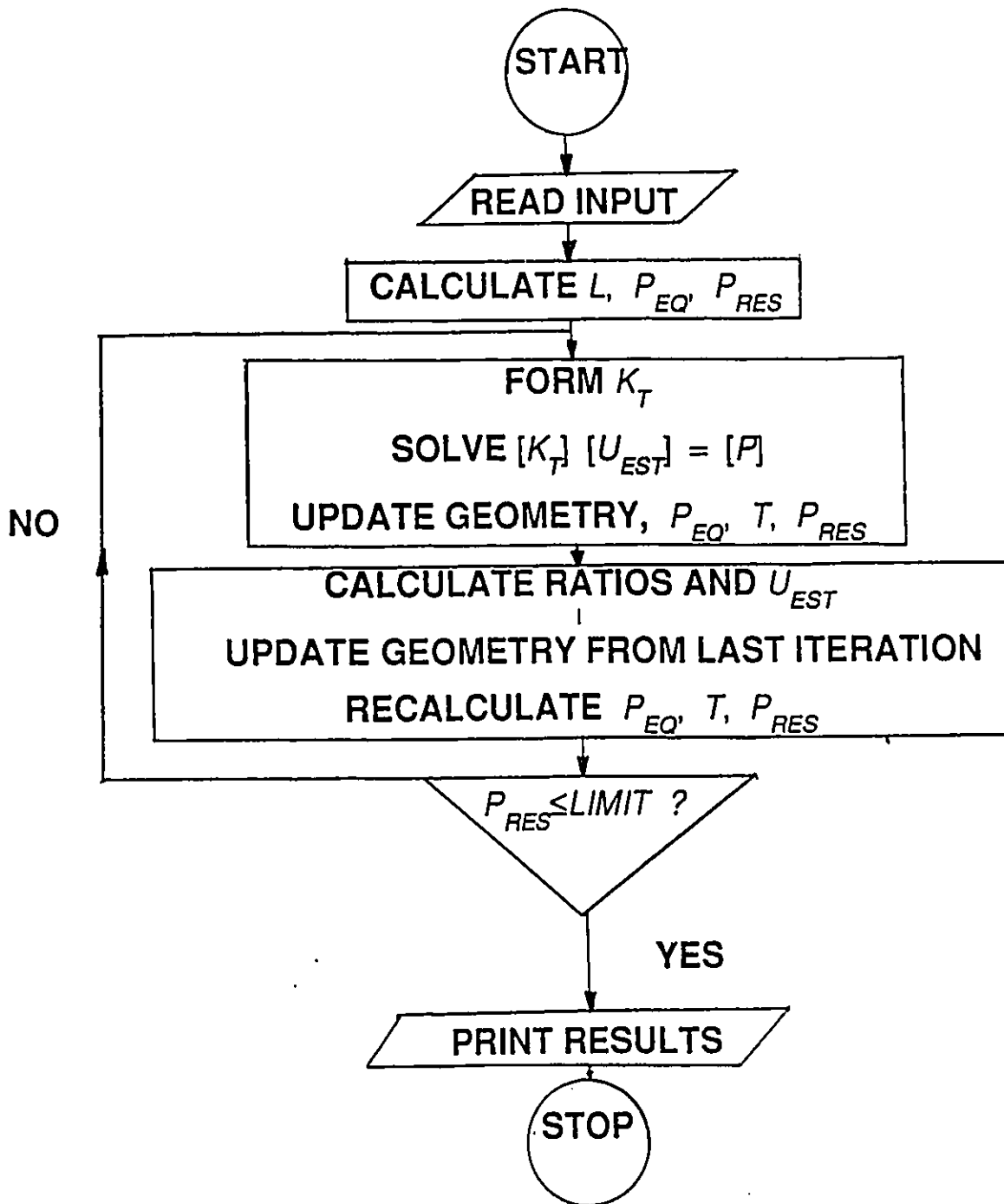
FIG. 2.8

FLOW CHART FOR KAR'S METHOD

Chapter III

METHOD OF SOLUTION

The principle of stationary potential energy can be stated as:

Of all displacement fields U which satisfy geometric compatibility, those which locally minimize the potential energy, $\Pi_p(U)$, also satisfy the equilibrium conditions and are stable equilibrium positions.

Mathematically,

$$\frac{\partial \Pi_p(U)}{\partial U_j}\Big|_{U=H} = 0 \qquad j = 1,2,3,\ldots\ldots N \qquad (3.1)$$

where H is the displacement field at a local minimum. Equilibrium position is said to have been achieved if

$$\Pi_p(H) < \Pi_p(U) \qquad (3.2)$$

for all U in some neighbourhood of H

In matrix formulation of structural problems such as the stiffness method of analysis, the search for the minimum of a total potential energy is based on successive iterations followed by some sort of a refinement. The gradient vector is drawn out into matrix form along with the applied load and displacement vectors. Consequently, these highly organized methods coupled with large scale computing facilities have made the prediction of the performance of complex structural systems routine.

Thus, the use of matrix notation in discrete element idealization simplifies the problem into a set of simultaneous equations suitable for a direct computer solution.

In the solution to the stiffness methods of analysis, the subroutine LSARG supplied by IMSL Inc., Math Library has proved to be adequate. It solves a system of linear algebraic equations having a real general coefficient matrix. It computes the LU factorization of the coefficient matrix using another subroutine LFCRG. The solution of the linear system is then found using the iterative refinement routine LFIRG.

LSARG fails if the upper triangular part of the factorization has a zero diagonal element or if the iterative refinement algorithm fails to converge. These errors occur only if the stiffness matrix, K_T or K_S is singular or very close to a singular matrix.

In the energy search method, the potential energy function constructed is solved by the direct search for the displacement vector U which would minimize the function. A substantial number of high performance structural systems are not adequately treated by linearized analysis techniques. Mathematical programming methods of unconstrained minimization seek to retain the flexibility of application which is characteristic of discrete element idealizations. Such methods also try to improve the representation of structural behaviour by avoiding some of the traditional linearizing assumptions.

There are a wide variety of unconstrained minimization techniques suitable to be applied to different types of optimization problems. They are discussed in detail in [6], [7] and [8]. The conjugate gradient method proposed by Hestenes and Stiefel[12] in 1952 and its extension by Fletcher and Reeves in [7] form the basis of the minimization algorithm used to generate the solutions to the problem presented in this work. (figure 3.1)

The Fletcher-Reeves method of conjugate gradients is an elegant procedure for solving a set of linear simultaneous equations having a symmetric positive definite matrix of coefficients. It begins with an initial approximation vector to the minimum U_o, which is modified after each iteration as the position of minimum of the potential energy function Π_p moves along a line through U_o in some specified direction $\{S\}$.

$z_1 > z_2 > z_3$

$\Pi = z_1$

$\Pi = z_2$

$\Pi = z_3$

X_1

X_2
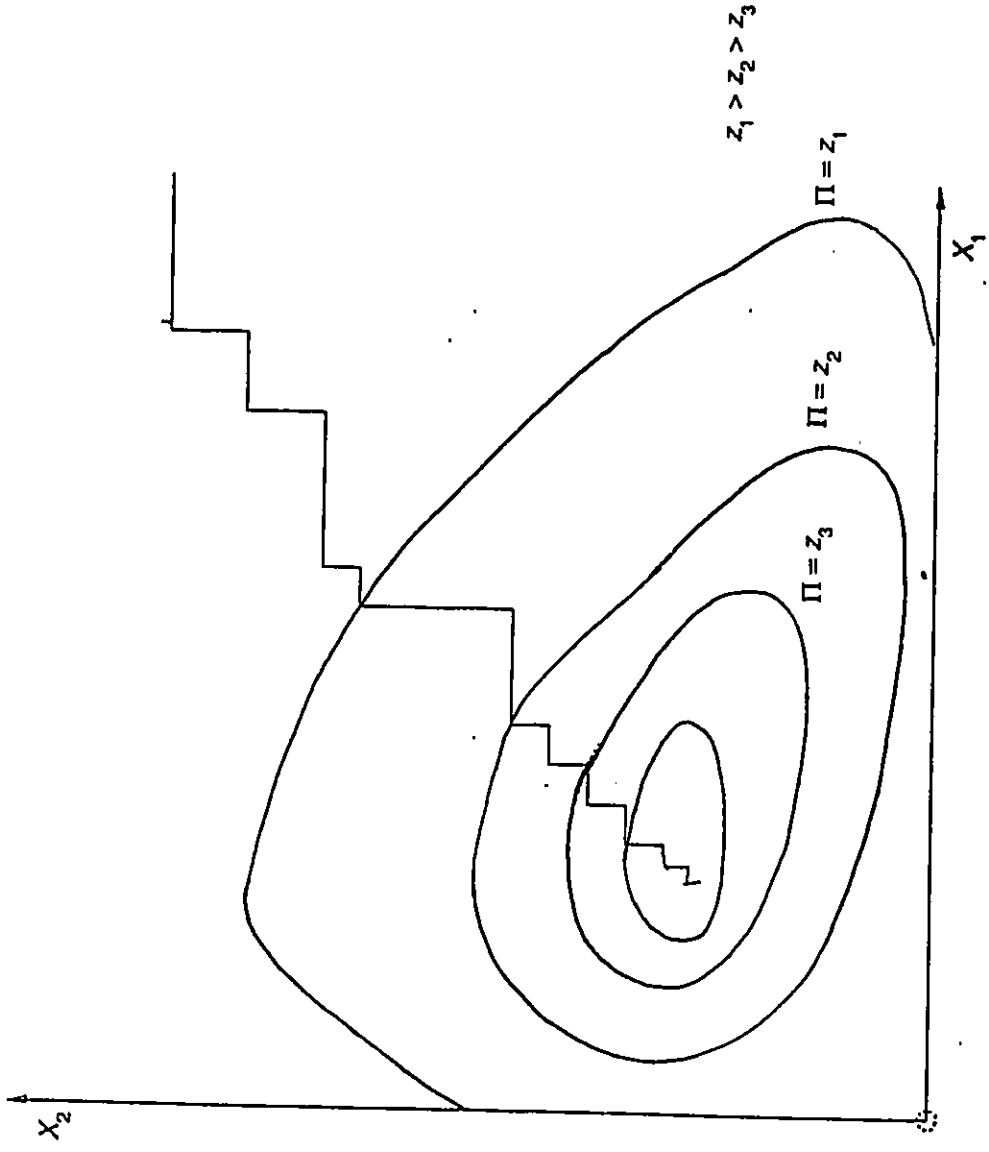
FIG. 3.1

SCHEMATIC DIAGRAM FOR THE CONJUGATE GRADIENT
METHOD FOR A TWO-ARGUMENT QUADRATIC FUNCTION

The initial direction of travel is in the negative gradient direction $\{S_o\}$,

$$\{S_o\} = - \{G_o\} = -\nabla\Pi_p(U_o) \tag{3.3}$$

and the step length ρ is determined by the relation:

$$\frac{df}{d\rho} = 0 \tag{3.4}$$

$$-\{S_i\}^T\nabla\Pi_p(U_o - \rho\{S_o\}) = 0 \tag{3.5}$$

where

$$f(\rho) = \Pi_p(U_o - \rho\nabla\Pi_p(U_o)) \tag{3.6}$$

and

$$U_{i+1} = U_i + \rho\{S_i\} \qquad \text{for } i = 1,2,3,\ldots\ldots\ldots n \tag{3.7}$$

The minimization by successive linear searches is illustrated by the quadratic function.

$$f(U) = f(H) + \frac{1}{2}(X - H)^T\{G\} \tag{3.8}$$

whose minimum is at $U = H$ and the gradient $\{G\}$ of second order partial derivatives is given by:

$$\{G\} = A(U - H) \tag{3.9}$$

and A is a positive definite symmetric matrix of second order partial derivatives.

The condition for the gradient to vanish is seen to be:

$$AU = AH \tag{3.10}$$

In the solution of the above equations, directions $\{S_o\}, \{S_1\}, \{S_2\}, \ldots\ldots\ldots \{S_{i+1}\}$ are generated using A orthogonality:

$$\{S_i\}^T A\{S_j\} = 0 \qquad \text{for } i \neq j \tag{3.11}$$

$$i,j = 1,2,3,\ldots\ldots\ldots n$$

and also the following relation:

$$\{S_{i+1}\} = - \{G_{i+1}\} + \beta_i\{S_i\} \tag{3.12}$$

where

$$\{G_{i+1}\} = \nabla\Pi_p(U_{i+1}) \tag{3.13}$$

and

$$\beta_i = \frac{\{G_{i+1}\}^T \{G_{i+1}\}}{\{G_i\}^T \{G_i\}} \tag{3.14}$$

This process is guaranteed, apart from rounding errors, to locate the minimum of an n-argument quadratic function in at most n iterations.

A cubic fit scheme recommended by Fletcher and Reeves in [6] is used in case the function is a nonquadratic one. If $\{G_i\}$ vanishes, the iterations conclude both to avoid division by zero in the next iteration and also because this is the formal requirement for U_i to be at a minimum. This cannot be achieved in practice because of truncation errors in the computer. Thus, the iterations are stopped when there is no significant reduction in the value of the potential energy function. That is

$$\Pi_p\{U_{i+1}\} - \Pi_p\{U_i\} < \varepsilon_1 \tag{3.15}$$

where ε_1 is the specified accuracy, whose value is recommended to be 1×10^{-16}.

Computer programs for the conjugate gradient method written by Kuester and Mize[18] have been used in this analysis.

Chapter IV

EXPERIMENTAL INVESTIGATION

An experimental study was undertaken to substantiate the results derived by the theoretical models. For this, a model of the Pontiac Silver Dome in Pontiac, Michigan, U.S.A. was constructed and subjected to various different loading conditions of internal pressure and externally applied distributed loads.

4.1   MODEL

A 1:120 model of the Pontiac Silver Dome was built using blue prints from the Department of Public Works and Service, Pontiac, Michigan, U.S.A. The cable net was modelled using a seven strand steel fishing cable of 0.8 mm diameter. The base and the walls were constructed of 9.5 mm plywood and the whole model was made air-tight to contain the pressure when the dome was inflated. To resist the horizontal forces created by the cables, a compression ring was built using 32 x 32 x 6.5 mm ($1\frac{1}{4}$ x $1\frac{1}{4}$ x 4 in.) rolled steel angles and 36 machined metal blocks. These blocks were welded in place at specified locations and were used for cable adjustment with the help of bolts.

The cable net consists of eighteen cables, nine running in each direction, soldered together at 81 points to form a cable net. Free rotation of the cables in the vertical plane was assured by soldering electrical eye connectors to the ends of the cables, which in turn, helped to fasten the cables to the adjustment mechanism.

A 0.05 mm (2 mil) plastic sheet was used as a membrane for the air supported roof. A single continous sheet was used because of the impracticality of modelling the membrane into panels spanning between the cables. However, the flexible nature of the plastic helped to conform to the geometry of the cable net when it was inflated, thus serving the purpose.

The detailed dimensions are given in figures 4.1 to 4.5.

4.2   SET - UP

The existing pressure system in the structural laboratory was used to provide the inflation pressure for the model. The pressure was regulated by means of a standard pressure regulator. The internal differential pressure inside the model was measured by means of a sensitive digital manometer capable of measuring pressures down to one hundredth of a millibar. Conventional gauges could not be used since differential pressures were as low as 10 pascals. The digital manometer is shown in figure 4.6.

The coordinate measuring system for the nodal points was built in the Central Research Shop and is capable of measuring displacements up to one ten thousandth of a millimetre in the X and Y directions. It consists of three frames, one along the length on each side of the model and the other spanning between them. The cross frame moves across the length of the model on two solid circular bars of the longitudinal frames which act as tracks. The movement is achieved by a ball bushing assembly, placed on either ends of the cross frame.

A second set of solid circular bars is mounted on the cross frame which helps to move the measuring equipment in the lateral direction. The transverse movement is achieved in the same way as before. Figure 4.7 to 4.9 show the entire assembly.

The measurement of the coordinates in the horizontal plane is done by "TURNVISION ", a microprocessor based Vision Readout Unit maufactured by ACU-RITE Inc., New York. This unit

is capable of receiving, storing, manipulating and displaying information. It receives data via the keypad, input by the operator or by positional electronic information decoded from ACU-RITE linear encoders as was done in this experiment. A monochrome monitor with an eighteen mm (seven inch) screen was used to display the information and is shown in figure 4.10.

Two linear encoders were fixed one on one of the longitudinal frames and the other on the cross frame. They had a fiducial trigger output capability i.e., they are capable of incorporating reference axes into the memory of the computer. Thus, any accidental movement would not hamper subsequent measurements. A suitable roller arrangement was constructed to relate the movements of the bearings on to the linear encoders as shown in figures 4.11, 4.12, 4.13 and 4.14.

The measurement in the vertical direction was done by means of a Linear Variable Differential Transformer (LVDT) shown in figure 4.15. This device used in the experiment was manufactured by Hewlett-Packard Inc. and has a capability of measuring displacements up to 14 mm (5.5 in.). A voltmeter and a digital multimeter were connected to the LVDT to convey the positional information in terms of millivolts. The LVDT was calibrated in the lab using a universal testing machine and sensitive dial gauages as shown in figure 4.16. The calibration curve shown in figure 4.17 was used to convert the electronic information into millimetres. The least count of this system is one hundredth of a millimetre.

4.3   LOADING CONDITIONS

The model was tested both for internal pressure as well as for various external distributed loads. Seven external patch loads were applied on the roof and the displacements measured in each case.

4.3.1 INTERNAL PRESSURE

The model was initially inflated with air sufficient to conform to the initial geometry of the structure (figure 4.18). At this point, the roof weight was just supported by the internal air pressure. Thereafter, the internal pressure was increased in four steps and nodal measurements were taken. The prototype had a stable operating pressure of about 240 pascals (5 lb/sq ft) over atmospheric pressure and this would be equivalent to a pressure of 2 pascals over the atmospheric pressure when computed on the basis of a scale factor of 120. Due to the difficulty in maintaining such a low differential pressure and also taking into account the relative imperfections in the construction of the dome, it was decided to use a differential pressure of about 10 pascals. This would not significantly alter the objectives of the study.

Load-deflection curves were plotted for salient nodal points in the structure and the results analyzed are presented in the following chapter.

4.3.2 DISTRIBUTED LOADS

The model was loaded with seven different types of symmetrical and unsymmetrical patch loads. Keeping the internal pressure constant, sand was distributed uniformly on panels to simulate snow and other types of distributed loads. The depth of sand in each of the load cases varied from 6 to 39.5 mm. Since the model is only geometrically similar, true snow loads could not be simulated.

In the first load case, a patch load of sand with a depth of 6 mm was placed on one of the extreme panels in the longitudinal direction as shown in figure 4.19. The patch was enclosed by the nodes 72, 80, 81 and 89 as per the numbering scheme shown in figure 4.18. The initial depth of 6 mm was increased to 39.5 mm in four steps and nodal measurements taken at each step.

In the second load case a patch load was placed simultaneously on the two panels enclosed by nodes 71, 72, 79, 80, 81, 88 and 89 and the nodal measurements were recorded. This is shown in figure 4.20.

Sand was placed on three panels enclosed by 66, 74, 75 and 83 and 71, 72, 79, 80, 81, 88 and 89 to form load case three as shown in figure 4.21. Figure 4.22 shows sand enclosed in the four panels in each extremity of the longitudinal direction. This was the first in the series of symmetrical loads placed on the model. Figure 4.23 shows the fifth load case where sand was placed on four panels at all four extremities. The sixth load case is shown in figure 4.24 and represents snow loads simulating ponding of the roof. The seventh load case shows sand placed over all the panels in figure 4.25. This is an extreme case of loading and the structure ceases to behave as a roof. This is rarely achieved in actual conditions and is incorporated in this case as a hypothetical condition for the purposes of theoretical study.

PAGINATION ERROR.

TEXT COMPLETE.

ERREUR DE PAGINATION.

LE TEXTE EST COMPLET.

NATIONAL LIBRARY OF CANADA.

CANADIAN THESES SERVICE.

BIBLIOTHEQUE NATIONALE DU CANADA.

SERVICE DES THESES CANADIENNES.

O/A = 897 mm

450 mm

310 mm

176 mm

64°

¢
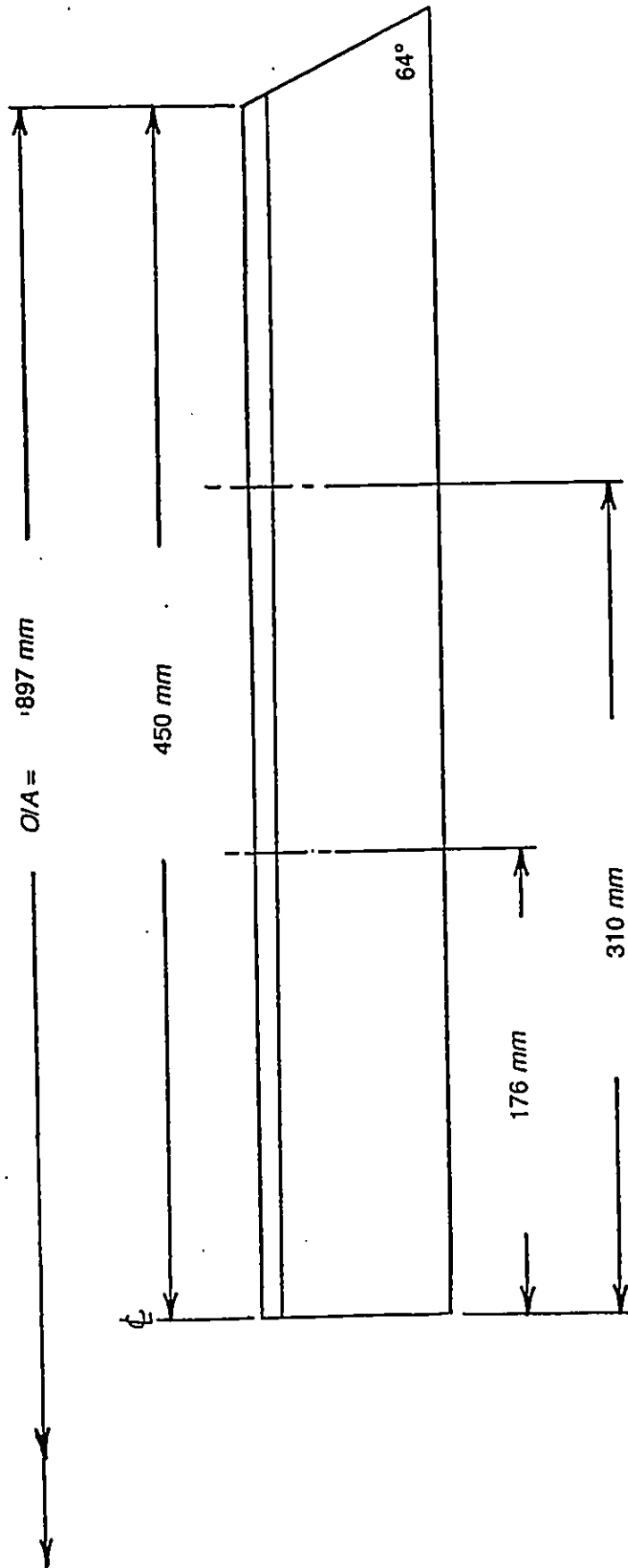
FIG. 4.2

32 x 32 x 6.5 MM ANGLE MEMBER FOR THE
SHORTER SIDE OF THE COMPRESSION RING

O/A = 460 mm

64°

71°
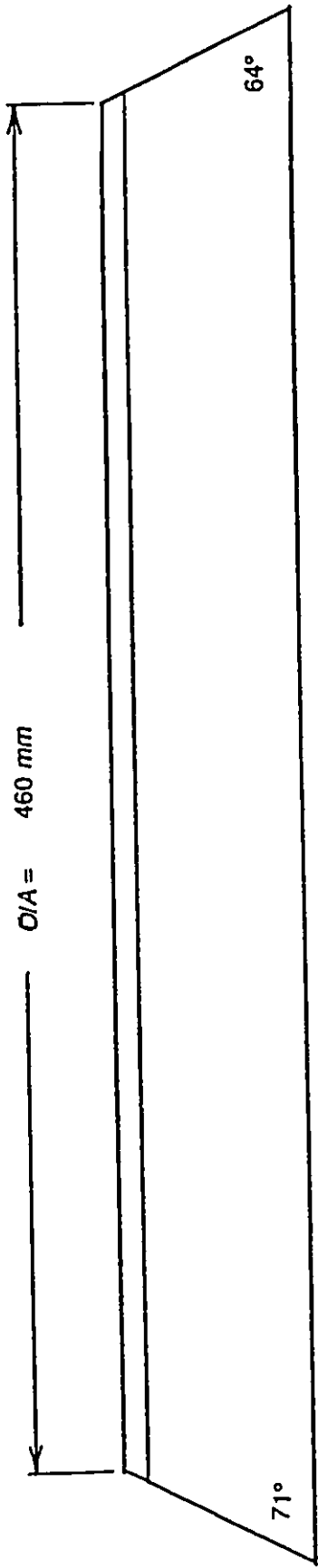
FIG. 4.3

32 x 32 x 6.5 MM ANGLE MEMBER FOR
THE CORNERS OF THE COMPRESSION RING
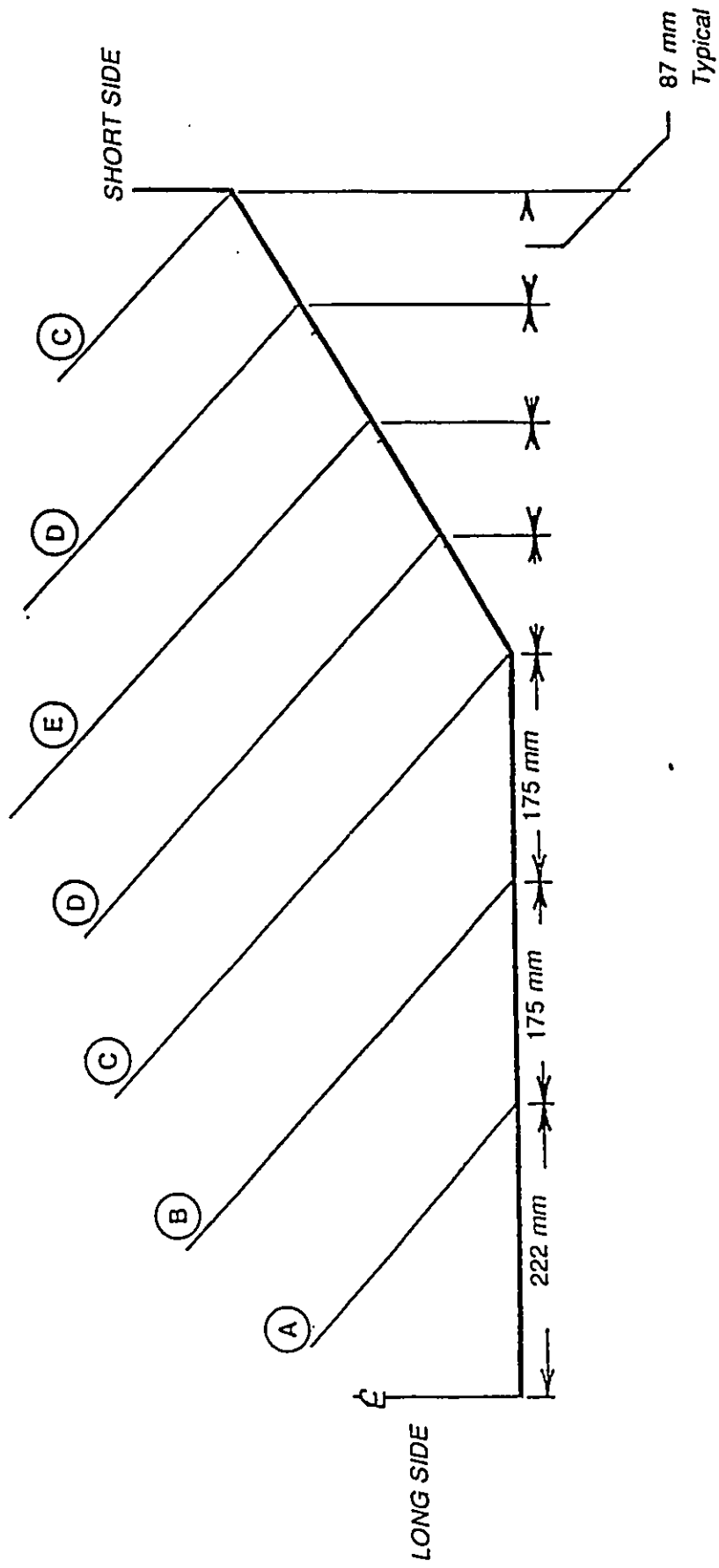
FIG. 4.4
MODEL DIMENSIONS (a)

47
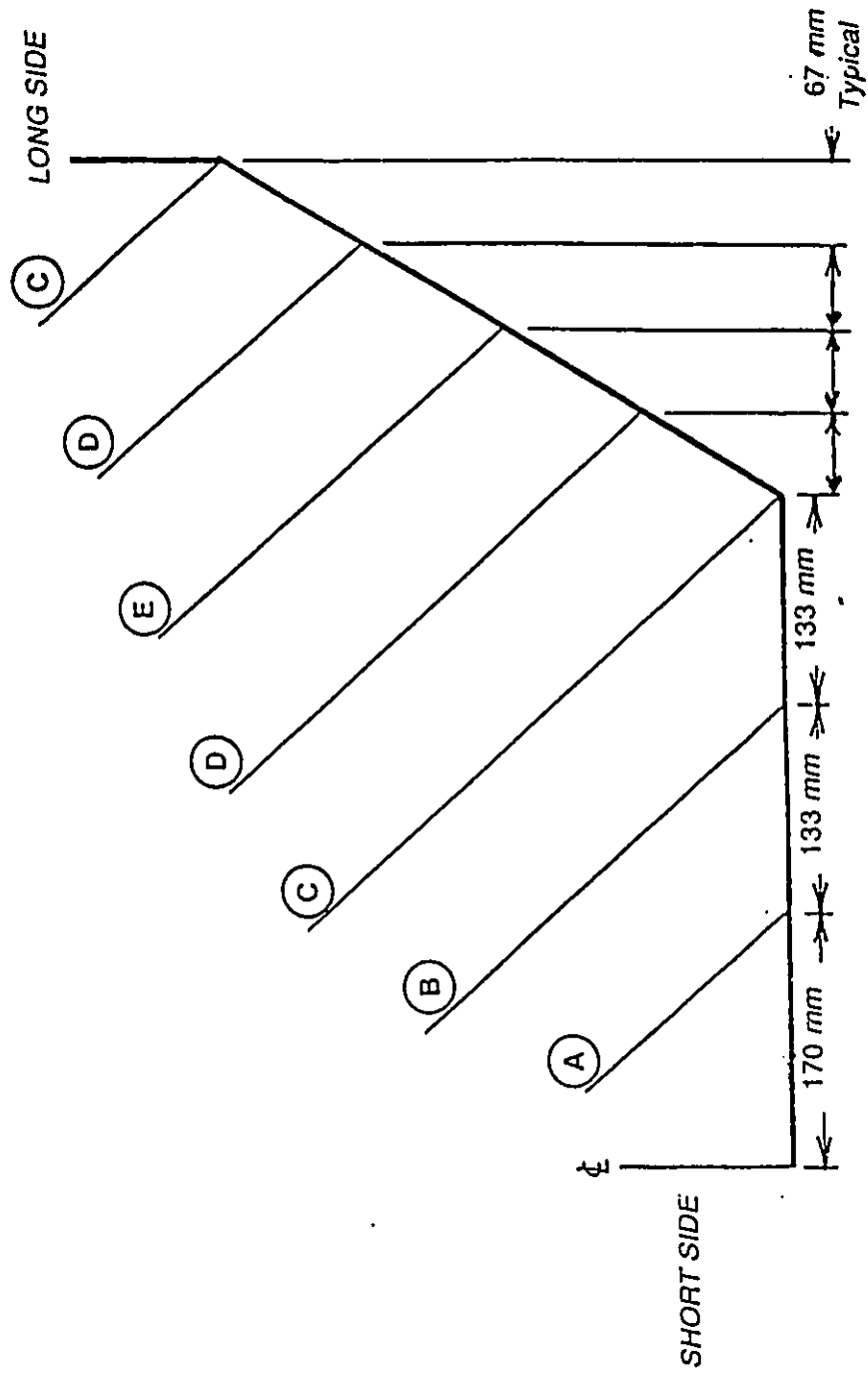
FIG. 4.5

MODEL DIMENSIONS (b)

FIG. 4.6

DIGITAL MANOMETER FOR
INTERNAL PRESSURE MEASUREMENT

FIG. 4.7

THE CROSS FRAME RESTING ON

THE LONGITUDINAL FRAMES
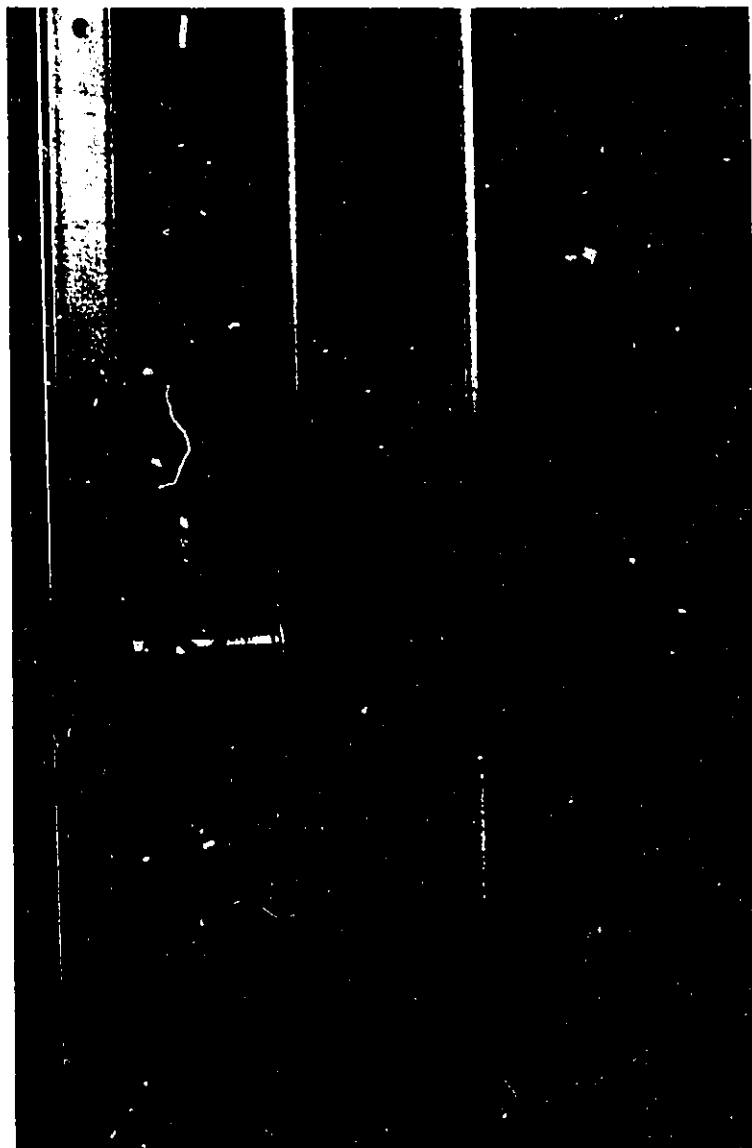
FIG. 4.8

THE CROSS FRAME SUPPORTING
THE LVDT AND ITS ASSEMBLY

FIG. 4.9

THE ENTIRE ASSEMBLY OF THE MEASURING SYSTEM
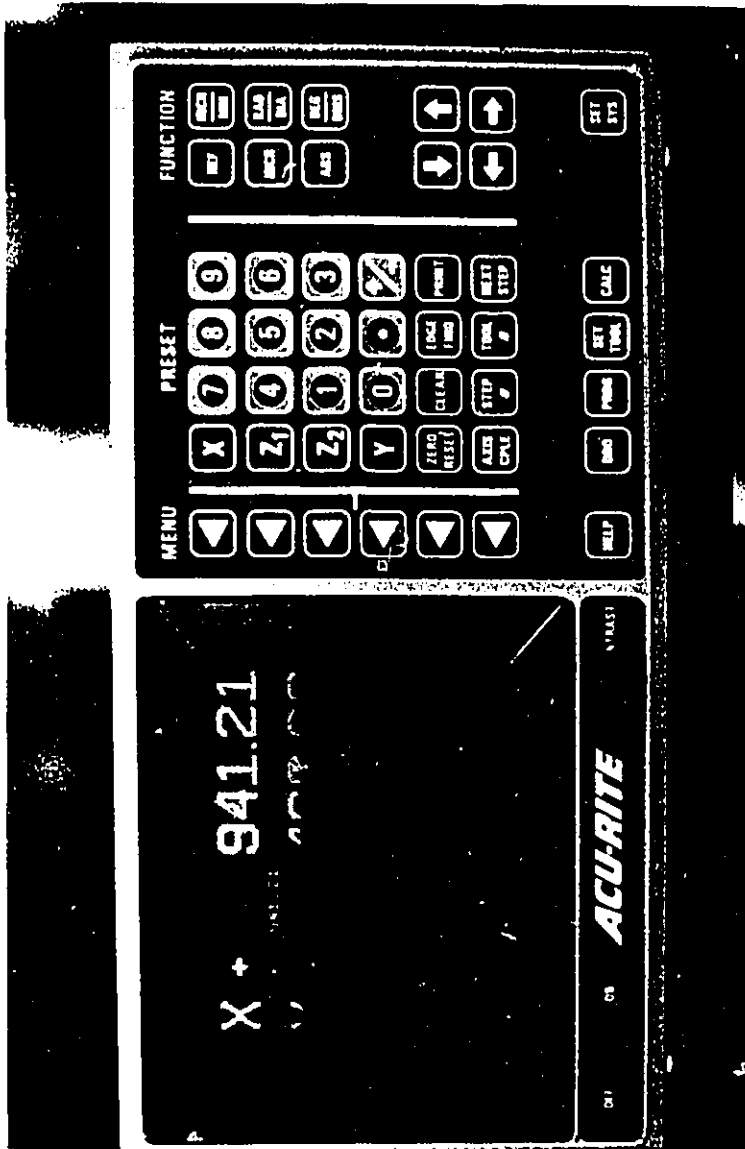
FIG. 4.10

THE MONOCHROME MONITOR WITH A 180 MM SCREEN

FIG. 4.11

THE LONGITUDINAL FRAME SHOWING
ONE OF THE LINEAR ENCODERS
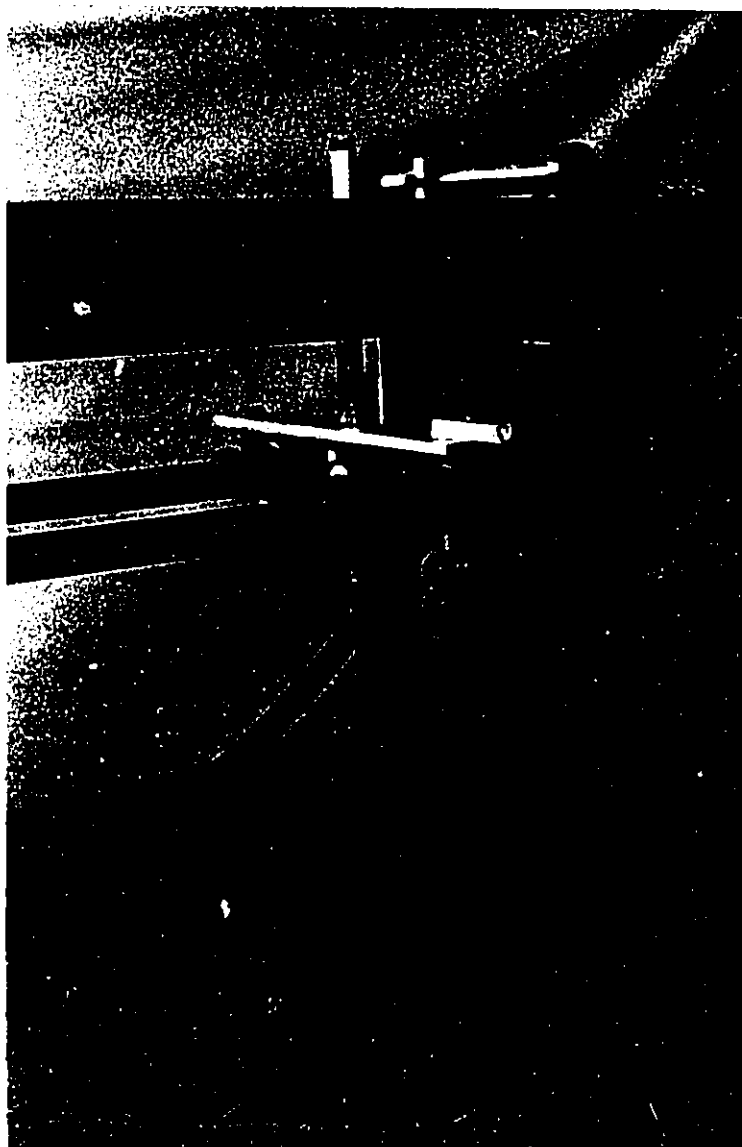
FIG. 4.12

THE TRANSVERSE LINEAR ENCODER FOR
THE MEASUREMENT OF THE Y-COORDINATE

FIG. 4.13

THE ROLLER ARRANGEMENT IN THE Y-DIRECTION

FIG. 4.14

THE BEARING ARRANGEMENT IN THE LONGITUDINAL
DIRECTION

FIG. 4.15
A VIEW OF THE LVDT
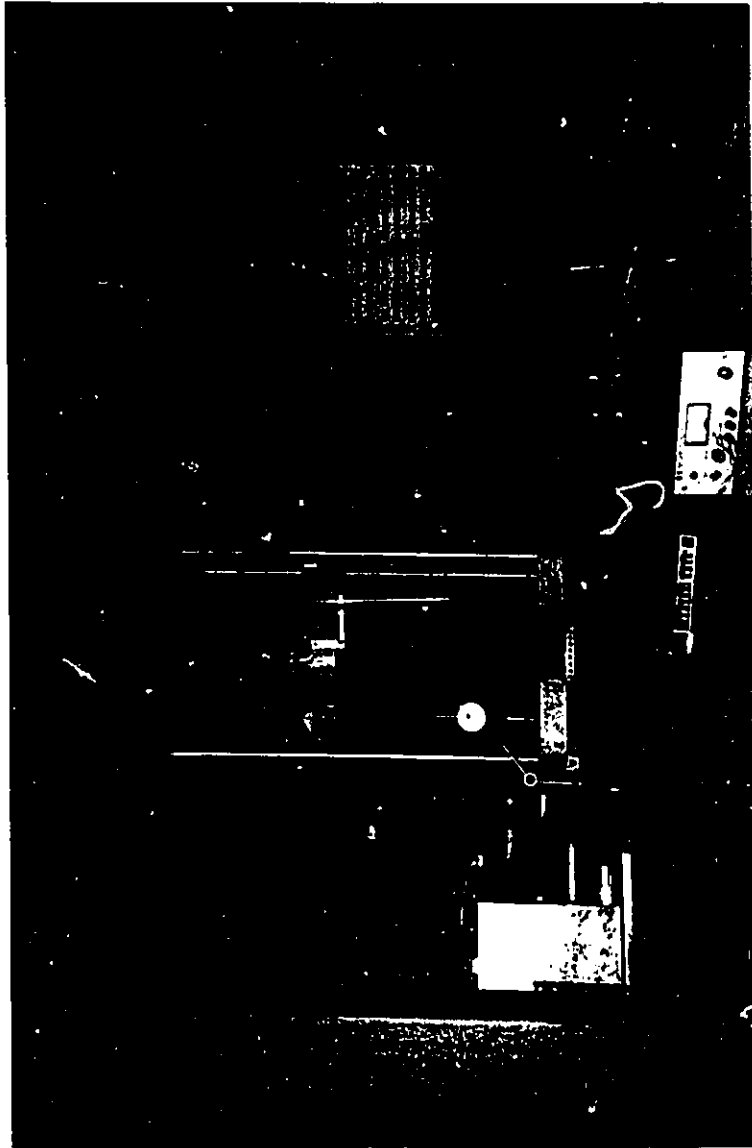
FIG. 4.16

CALIBRATION SET-UP FOR THE LVDT
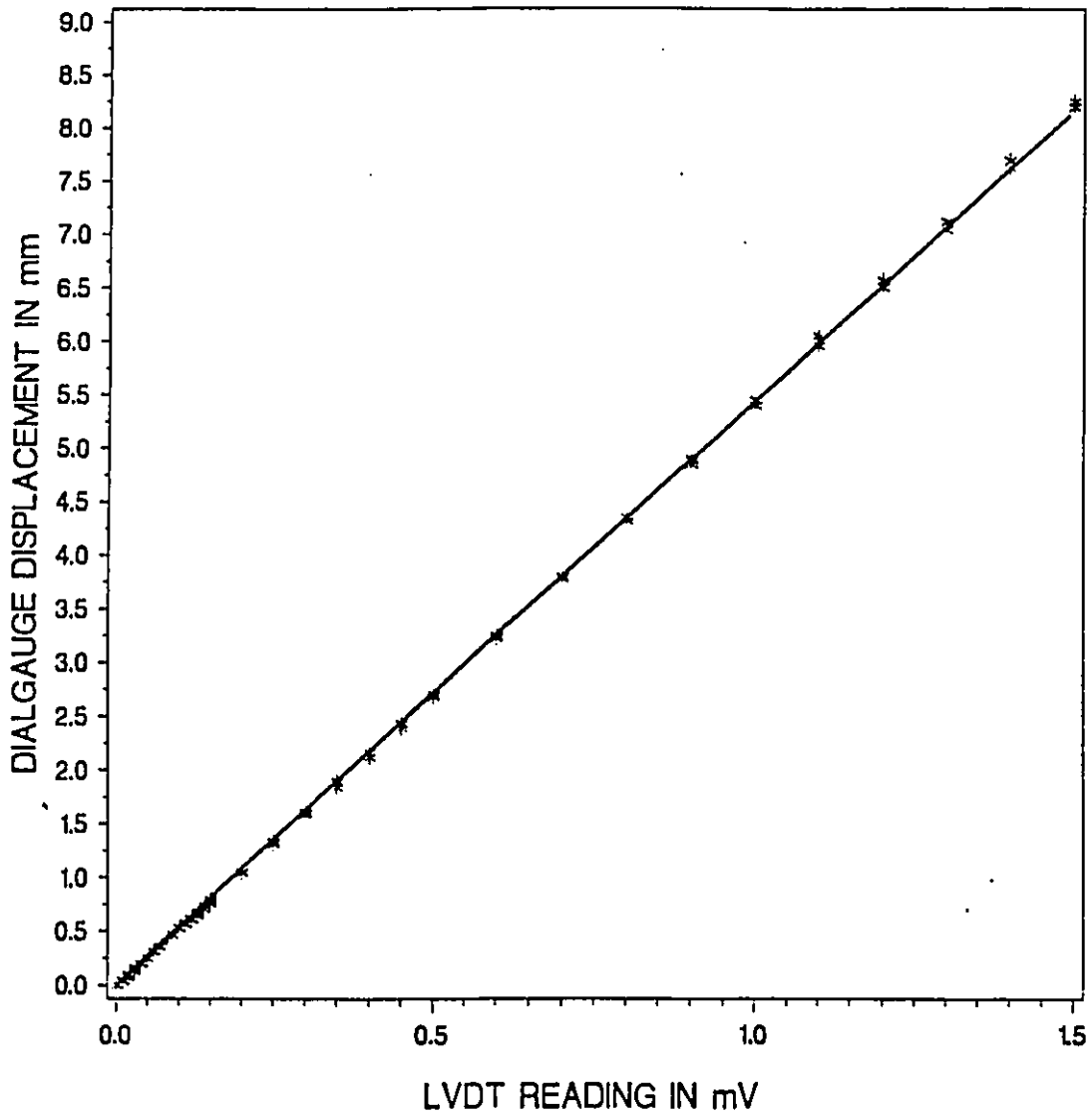
CALIBRATION CURVE FOR LVDT

FIG. 4.17
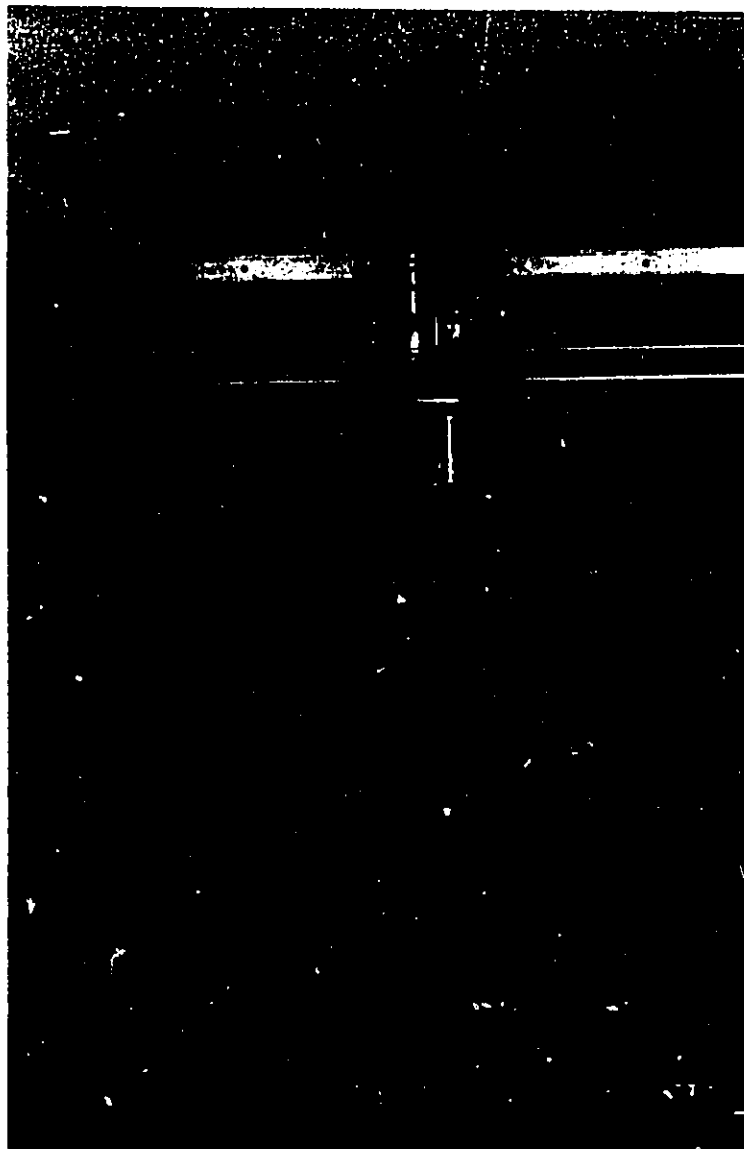
FIG. 4.18

MODEL WITH INTERNAL PRESSURE ONLY
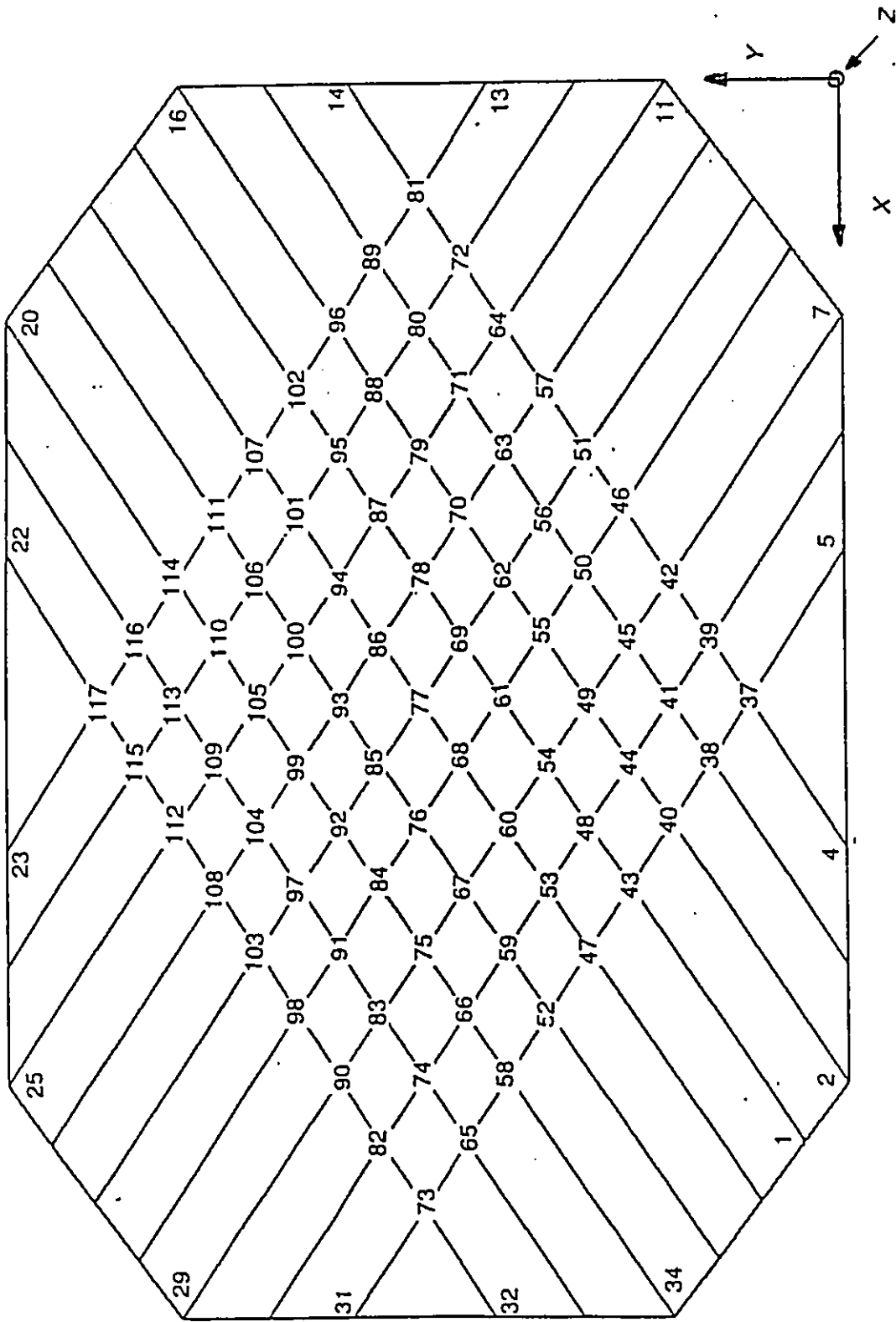
FIG. 4.19

NODE NUMBERING SCHEME

62
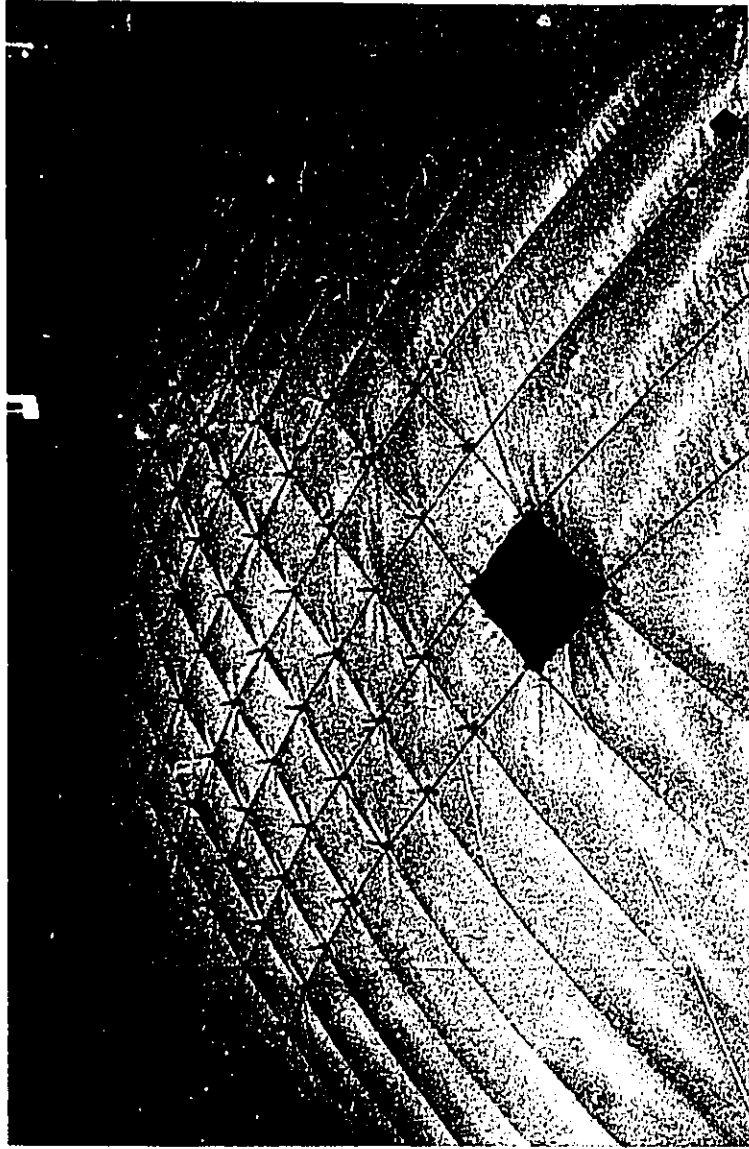
FIG. 4.20
LOAD CASE ONE

LOAD CASE TWO

FIG. 4.21
LOAD CASE TWO

FIG. 4.22

LOAD CASE THREE
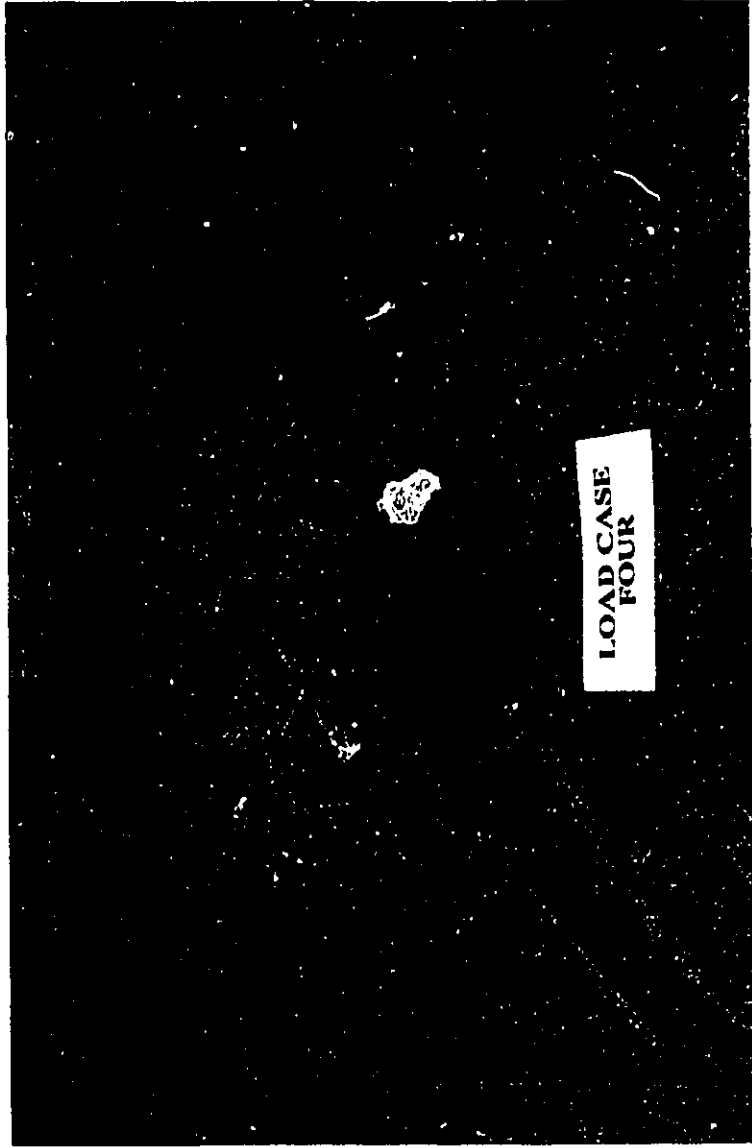
FIG. 4.23
LOAD CASE FOUR
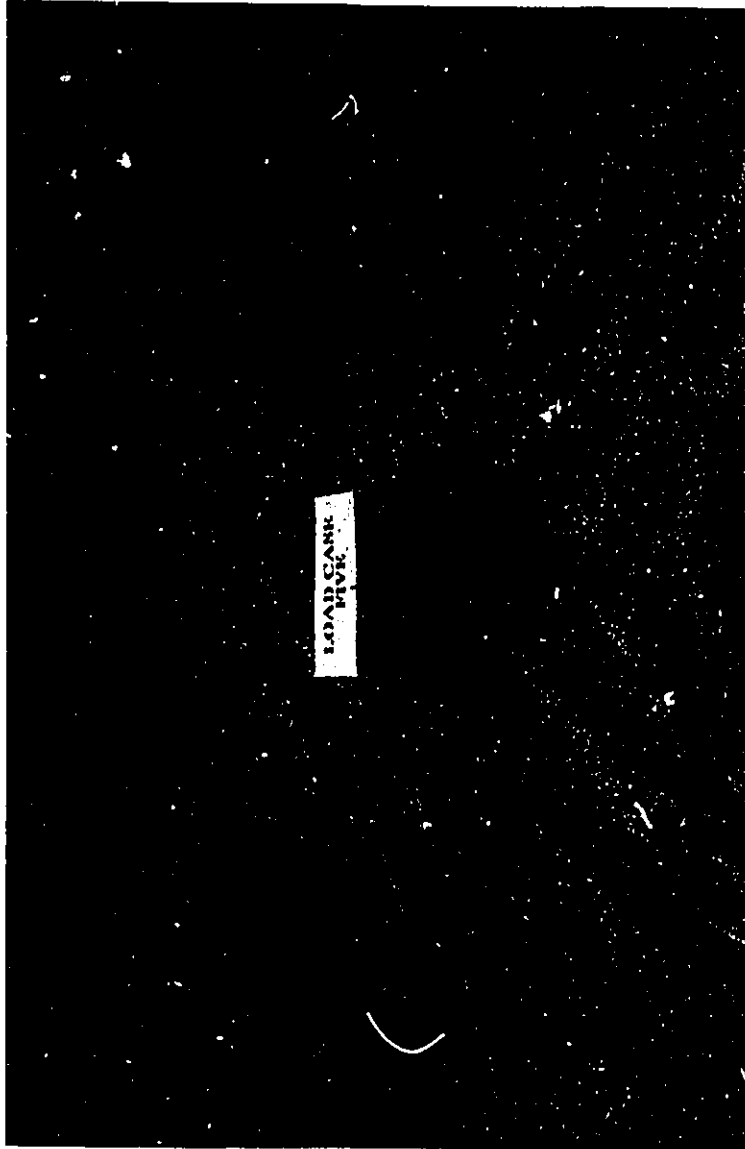
FIG. 4.24
LOAD CASE FIVE

FIG. 4.25
LOAD CASE SIX
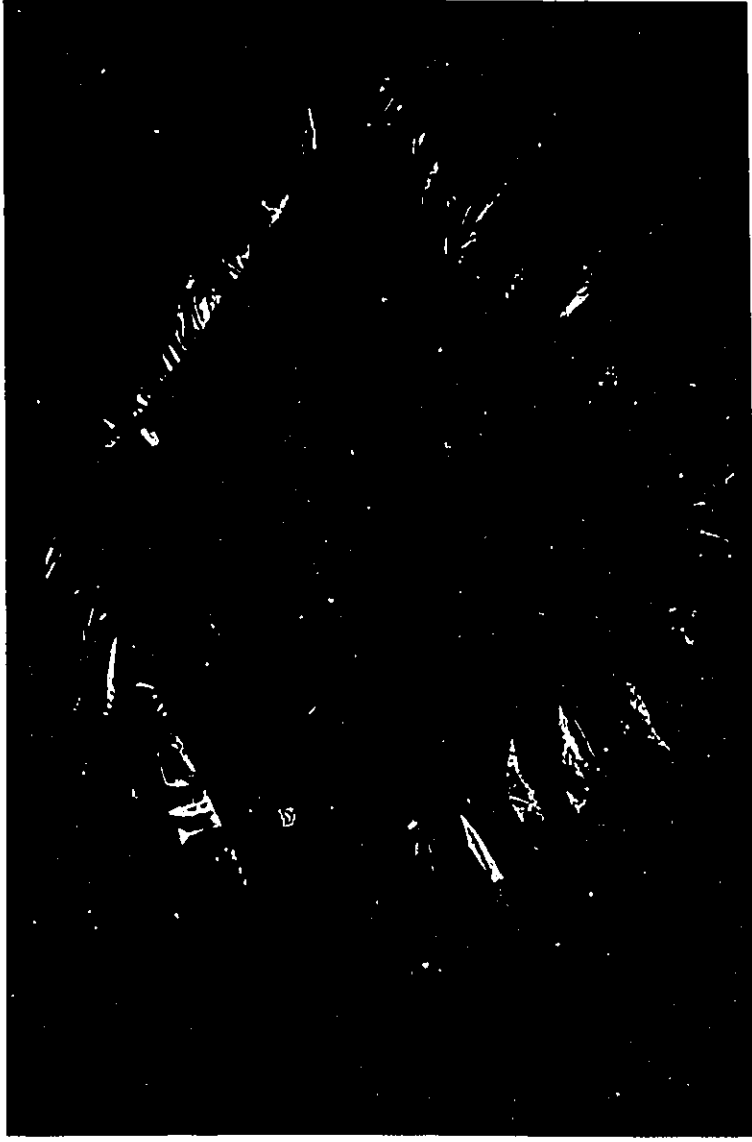
FIG. 4.26
LOAD CASE SEVEN

FIG. 4.27

DEFLECTED PROFILE OF THE ROOF FOR LOAD CASE
SEVEN

Chapter V

RESULTS AND ANALYSIS

In this chapter, each of the methods outlined in chapter III is used to obtain solutions for various cases of distributed loads on the model. Computer programs in Fortran have been developed for all the four solution techniques with the aid of an IBM 4381 computer. Load versus displacement curves are drawn for salient nodal points for each load case. The material properties of the cable are also discussed here.

In the following load cases, load increments were made until the solution method failed to converge to a solution. No difficulties were encountered in obtaining convergence in the unstressed state for the energy search method. On the other hand, the other theoretical models exhibited wild oscillations which resulted in nonconvergent situations. This was because the terms associated with the member force T in equations (C.13) and (C.14) is zero in the first cycle of iteration. To overcome this problem, a small stiffening force was introduced into each member of the structure when analyzing by the matrix methods. This stiffening force had a stabilizing influence on the elements of the structural stiffness matrix although small enough to be neglected in force computations.

A solution based on the elastic deformation matrix was also considered. This neglected the contribution of the cable forces T and the subsequent changes in geometry. However, it was found that solution techniques based on this type of matrix were inadequate to prevent large oscillations during the first few cycles of iteration and thus resulted in divergent solutions.

Among the matrix methods of analysis, the solution technique based on the linearized version of the stiffness of an element, i.e., the tangent stiffness method, tended to underestimate failure loads in that the load-displacement curves for the various nodes flattened at a relatively early stage of loading. It was beset by convergence difficulties especially in highly unstable load configurations.

The secant stiffness method of analysis based on the complete nonlinear version of the stiffness of the elements showed very good agreement with the experimental results. The results obtained by the energy search method and the secant stiffness method almost coincide in the initial stages. This technique also underestimated the ultimate load. The load-displacement curve plotted for this technique also flattened out early but at a later stage of loading as compared to the tangent stiffness method.

The fourth solution technique, which is based on Kar's modification to the solution obtained by the tangent stiffness method, helped to improve the results and bring it on a par with the secant stiffness method, but it still underestimated the ultimate load. As well, the expected hastening of convergence was not achieved.

Due to experimental difficulties, like the slipping of sand into adjacent panels at depths beyond 40 mm, the loading could not be carried out beyond 4 N at each nodal point which translates to a depth of 40 mm of sand. Nonetheless, the trends clearly follow the energy search method very closely and the failure load is clearly shown to be definitely beyond that predicted by any of the matrix methods of analysis.

5.1   LOADING CONDITIONS

5.1.1   INTERNAL PRESSURE

The model was filled with compressed air to achieve a uniform internal pressure. Initially, the model was filled with air just sufficient to hold the cable net in an unstiffened configuration i.e., it was inflated just to the point where a tensile force was starting to develop in the members. The coordinates for this position were worked out previously with the help of blueprints of the Pontiac Silver Dome obtained from the Department of Public Works, Pontiac, Michigan, U.S.A.

The internal pressure was then increased in small increments and the nodal displacements measured. Load versus displacement curves are drawn at nodal points 37, 73, 77, 81, 117 numbered as per the node numbering scheme shown in Figure 4.18. The theoretical models showed good agreement with the experimental results and the curve trends were along expected lines.

The experimental result showed a 20% variation with the results of the theoretical models as shown in figure 5.1. Figures 5.2 and 5.3 show the best correlation between the various models.

5.1.2   DISTRIBUTED LOADS

The internal pressure was kept constant for all the patch load cases. The internal pressure would have to be kept as low as 2 pascals above the atmospheric pressure to simulate a differential pressure of 240 pascals (5 lb/sq ft) above the atmospheric pressure in the prototype. Such a small pressure was impractical to be kept constant and it was decided to increase the internal pressure to 10 pascals since the object of the studies would not be significantly affected.

5.1.2.1 CASE ONE

The results of the computer runs for load case one shown in Fig. 5.4 are plotted along with the experimental observations in the form of load-displacement curves. Figures 5.5 to 5.11 show the load-displacement histories at various salient nodal points on the dome. There is no difference in the results obtained by the various models in the initial stage as shown in figure 5.5. Figure 5.6 indicates a linear relationship between load and displacement for loads up to 4 N for all the methods except the tangent stiffness method. The latter exhibits a linear curve till the superimposed load reaches 1.75 N. Thereafter, it begins to flatten predicting an ultimate load of about 2 N. The secant stiffness method predicts the ultimate load at 4 N and Kar's method at 3.9 N. However, the energy search method predicts the ultimate load to be about 30 N. Good agreement was observed between the theoretical models and the experimental observations for displacements directly under the load as shown in figures 5.5 to 5.8. However in figure 5.9, the curve plotted for node 73 indicates a large variation of as much as 200% over the theoretical models. No displacement was observed at node 77 in any direction and this is reflected in figures 5.10 and 5.11. The energy search method seems to predict the ultimate load more realistically as is indicated by the close agreement with the experimental result.

5.1.2.2 CASE TWO

The second load case covers two extreme panels in the longitudinal direction as shown in figure 5.12. Again, there is little or no variation in the results obtained by the various methods as shown in figure 5.13. The load displacement curves plotted for this case show that nonlinearity sets in when the applied load is more than 2 N for all the models except that of the tangent stiffness method. The curve for this method ceases a linear trend at a superimposed load of just less than 1 N. The ultimate load is predicted to be around 1.25 N. The experimental trend is excellent at node 81 in the Z direction closely following the energy search model. This is shown

in figure 5.13. However, the X observation as shown in figure 5.14 at the same node shows a variation of about 20% with the energy search method. There is no displacement at node 77 as reflected in figures 5.15 and 5.16. Figure 5.17 shows the failure of the matrix methods to converge to a solution. Even the energy search curve has an unsteady and wavering trend before being smoothened at higher loads.

5.1.2.3 CASE THREE

This load case is shown in figure 5.18 and the results plotted have exhibited the same kind of trends as in the previous two cases i.e., the prediction of ultimate load by the tangent stiffness model is approximately half the value predicted by other matrix methods. Also the results generally coincide in the initial loading stages as shown in figure 5.20. The secant stiffness curve indicates an ultimate load of 6 N at node 80. Experimental trends as usual prove that the energy search method is realistic in its predictions.

5.1.2.4 CASE FOUR

This is a symmetrical patch load applied at both ends in the longitudinal direction as shown in figure 5.21. Figure 5.22 to 5.25 highlight the loading symmetry of the structure. Interestingly, the energy search model curves upward in both cases. This would seem to suggest that the sway in the X direction remains constant after an application of 9 N at nodes 73 and 81 respectively. The curves plotted in the direction at the same nodes show that nonlinearity is not pronounced even after a considerable number of load increments. The experimental observations shown in figure 5.22 fail to follow the energy search curve closely and a distinct possibility of severe divergence at higher loads is noticed. As usual, there is an excellent agreement between the results obtained by the various models in the initial loading range. Another interesting feature, shown in figure 5.26 was the behaviour of the energy search curve at node 77. It can be

interpreted as a change of direction of sway as the load increases. The same trend is followed by Kar's model. Displacements at nodes 74 and 75 correlated very well with that of nodes 79 and 80 respectively as is expected in a symmetric load case.

5.1.2.5 CASE FIVE

This load case is symmetric about both the X and Y directions as shown in figure 5.27. Test results showed very good agreement with the theoretical models in the vertical Z direction but the X and Y direction load-displacement curves tended to a slight variation. Figure 5.28 shows the load-displacement curve at node 37 in the Y direction. The experimental curve exhibits a slight kink in the initial stages. A likely reason for this behaviour could be the possible lateral shift in the measuring equipment due to the extreme slope at this point. On the other hand the Z direction curve at the same node shows excellent agreement and correlation. Figures 5.30 and 5.31 also show the same theoretical trends which generally corroborate with experimental observations. Figure 5.32 shows the same trend exhibited by node 77 as in case 3.

The linear trend ends for all the models at a superimposed load of about 3 N for all the matrix methods including the tangent stiffness method. The energy search curve continues to be linear until it reaches 5 N. The symmetrical characteristics of this load case is very clearly shown by the identical trends of the curves at nodes 37, 73, 81 and 117 as shown in figures 5.28, 5.30, 5.33 and 5.34 respectively.

5.1.2.6 CASE SIX

This is a very simple load case with a patch load placed on the four central panels as shown in figure 5.35. The load-displacement plots shown in figures 5.36 to 5.39 indicate the detection of general instability by the energy search method. The displacements oscillate mildly but conform to a general trend. The same trend is predicted by the tangent stiffness method but as usual it

underestimates the behaviour at the node. Figure 5.37 shows an unusual trend of Kar's method. It indicates the failure of Kar's method to successfully hasten convergence in ill-conditioned situations. The same trend is exhibited in figure 5.38 which is the load-displacement curve at node 117 in the Z direction. However, the behaviour of the structure directly under the load is as per predicted lines. This is clearly illustrated in figure 5.39.

5.1.2.7   CASE SEVEN

This case shown in figure 5.40 is very unstable in that the theoretical models with the exception of the energy search method failed to converge beyond a few load increments. Even the energy search method which is normally not beset with convergence problems exhibited a certain degree of divergence at a relatively early stage. In this case, the structure collapsed totally and ceased to behave as a roof even at the first few increments of loading. A considerable sway in both the lateral directions was predicted by the energy search method as shown in figures 5.41 and 5.42. But experimental trends clearly indicated a shift to only one side. Sand spilling into adjacent panels due to ponding could be a possible reason for this type of trend. Overall, there is general disagreement with experimental observations and theroetical values in this case. The load-displacement curves for the vertical direction as shown in figures 5.42 and 5.43 do not corroborate at all with the experimental curves in the initial loading stages but seems to follow in the general direction of the energy search model. However, the energy search method predicts expected trends at node 77 as shown in figure 5.43 and the experimental curve indicates the same type of behaviour although with a lateral shift.

LOAD-DISPLACEMENT CURVES FOR INTERNAL PRESSURE
AT NODE 37 IN THE Z DIRECTION
FIG. 5.1

LOAD-DISPLACEMENT CURVES FOR INTERNAL PRESSURE
AT NODE 77 IN THE Z DIRECTION
FIG. 5.2
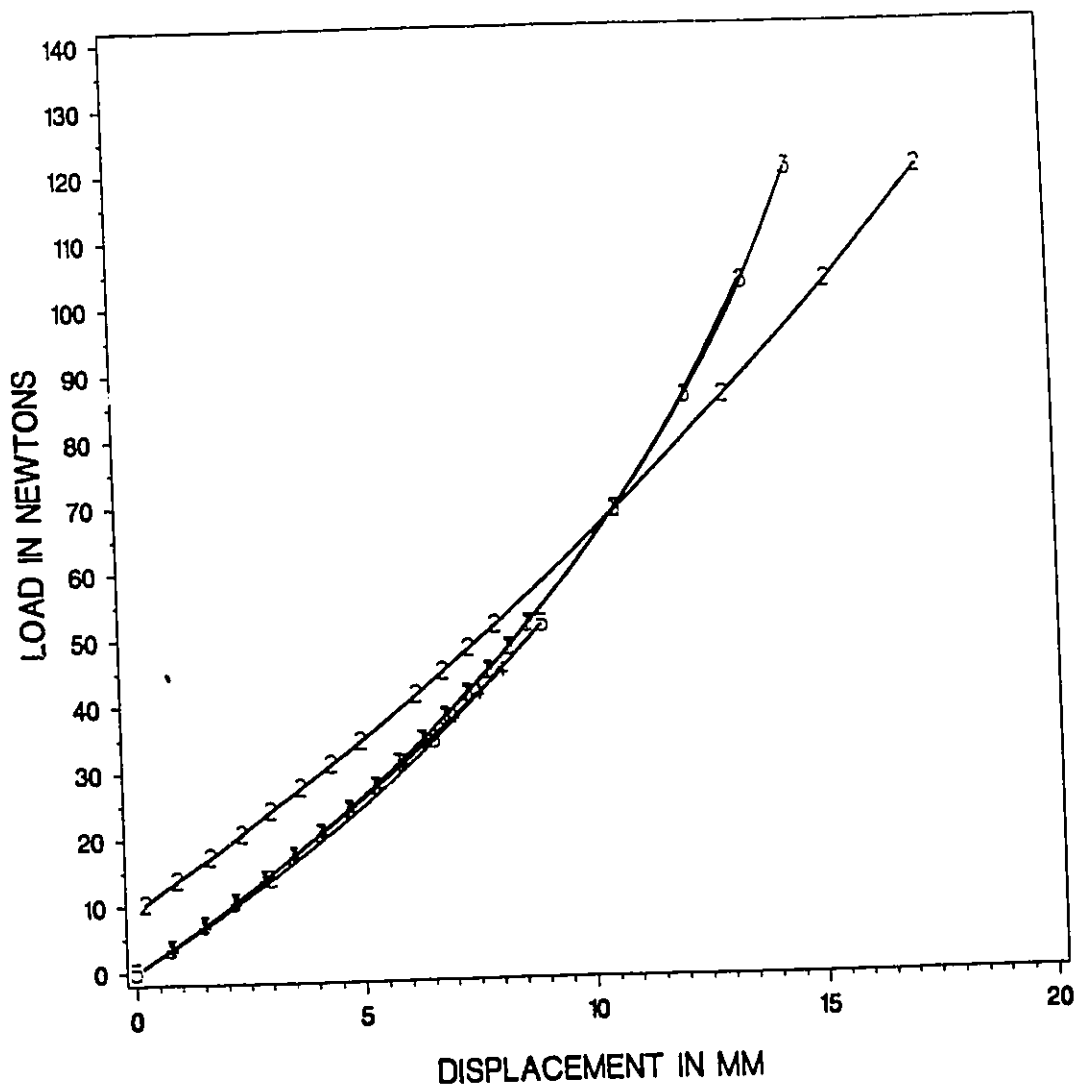
LOAD-DISPLACEMENT CURVES FOR INTERNAL PRESSURE
AT NODE 81 IN THE Z DIRECTION
FIG. 5.3

FIG. 5.4
CASE ONE

LOAD-DISPLACEMENT CURVES FOR CASE ONE
AT NODE 80 IN THE Z DIRECTION
FIG. 5.5

LOAD-DISPLACEMENT CURVES FOR CASE ONE
AT NODE 80 IN THE X DIRECTION
FIG. 5.6

LEGEND

LOAD-DISPLACEMENT CURVES FOR CASE ONE

AT NODE 81 IN THE Z DIRECTION

FIG. 5.7

D-DIVERGENCE & OSCILLATION OF SOLUTION

LOAD-DISPLACEMENT CURVES FOR CASE ONE
AT NODE 81 IN THE X DIRECTION
FIG. 5.8

LOAD IN NEWTONS

DISPLACEMENT IN MM

LEGEND ┼─┼─┼ TAN STIFFNESS   2─2─2 ENERGY SEARCH
        3─3─3 SEC STIFFNESS   4─4─4 KAR'S METHOD
        5─5─5 EXPT. RESULT

LOAD-DISPLACEMENT CURVES FOR CASE ONE
AT NODE 73 IN THE Z DIRECTION
FIG. 5.9

LEGEND +—+—+ TAN STIFFNESS   2—2—2 ENERGY SEARCH
       3—3—3 SEC STIFFNESS   4—4—4 KAR'S METHOD

LOAD-DISPLACEMENT CURVES FOR CASE ONE
AT NODE 77 IN THE Z DIRECTION
FIG. 5.10
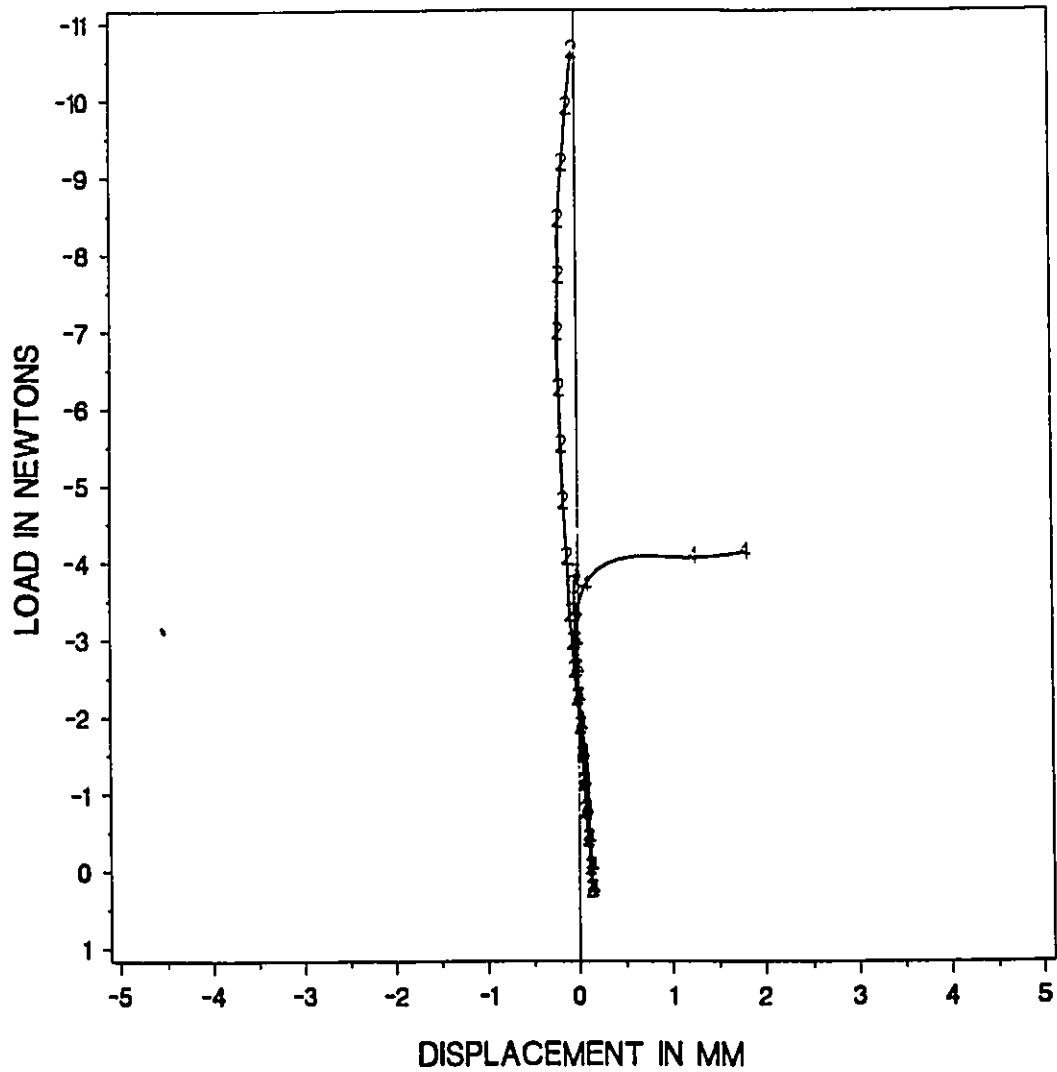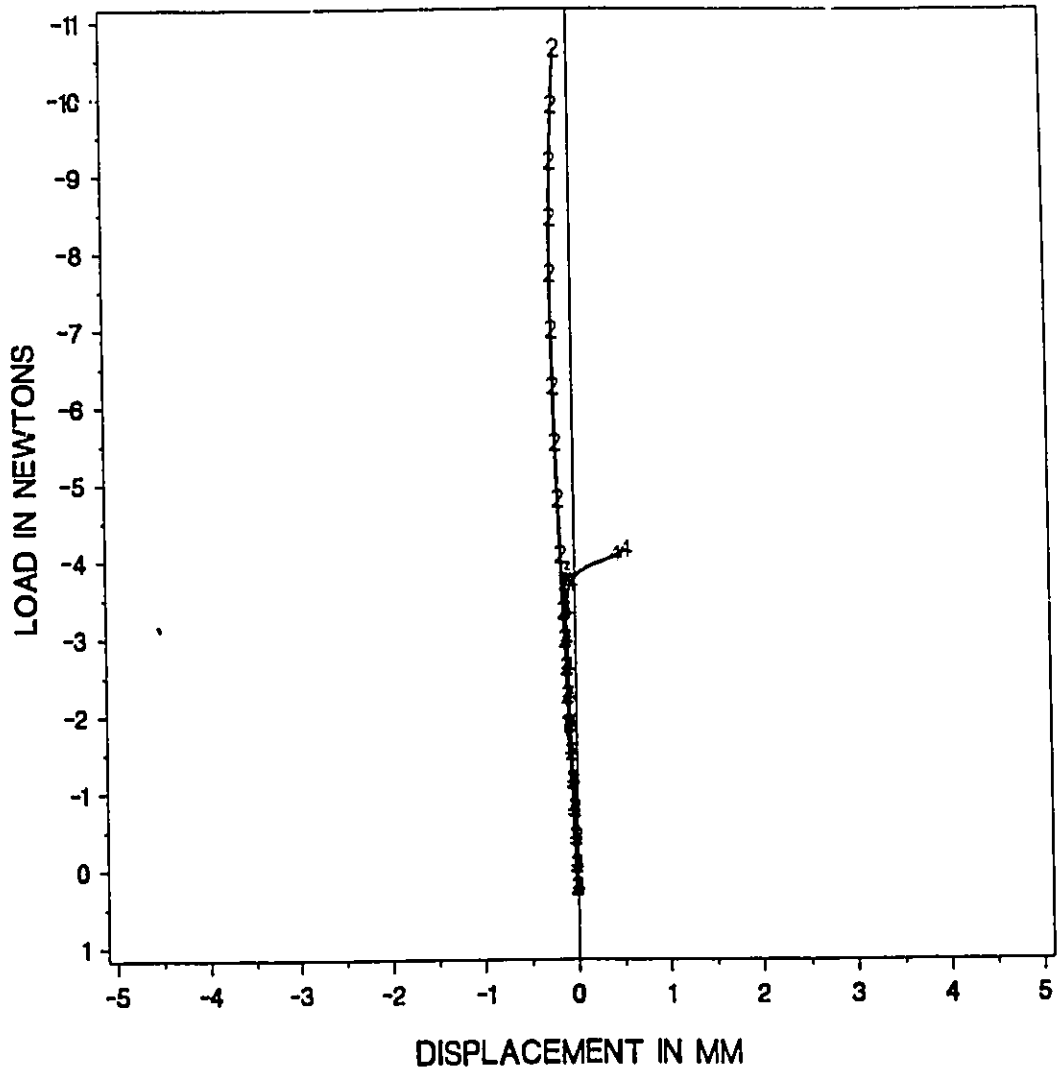
LEGEND   +—+—+ TAN STIFFNESS   2—2—2 ENERGY SEARCH
              3—3—3 SEC STIFFNESS   +—+—+ KAR'S METHOD

LOAD-DISPLACEMENT CURVES FOR CASE ONE
AT NODE 77 IN THE X DIRECTION
FIG. 5.11
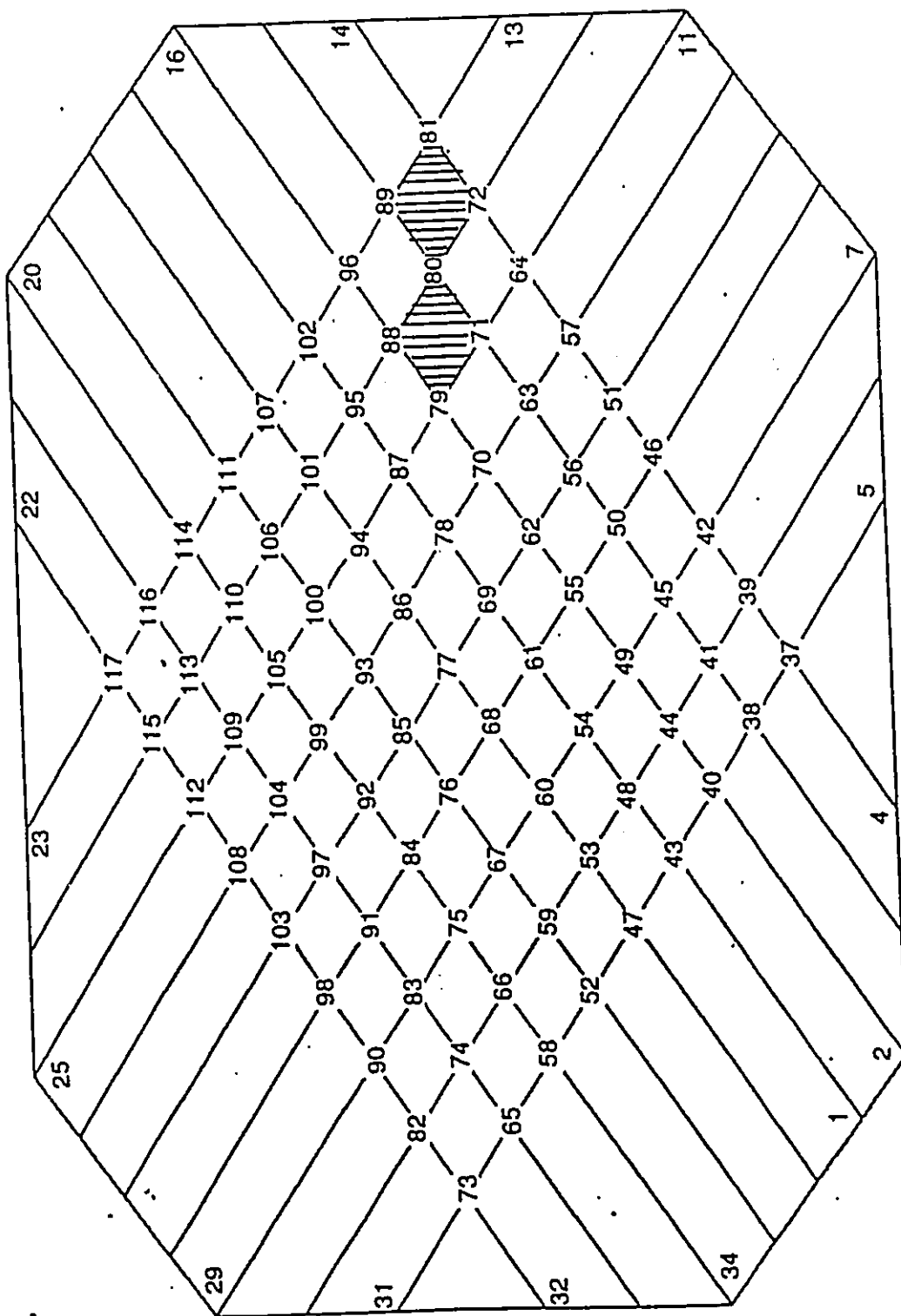
**FIG. 5.12
CASE TWO**

LEGEND +—1—+ TAN STIFFNESS   2—2—2 ENERGY SEARCH
       3—3—3 SEC STIFFNESS   +—4—+ KAR'S METHOD
       5—5—5 EXPT. RESULT

LOAD-DISPLACEMENT CURVES FOR CASE TWO
AT NODE 81 IN THE Z DIRECTION
FIG. 5.13

D-DIVERGENCE & OSCILLATION OF SOLUTION

LOAD-DISPLACEMENT CURVES FOR CASE TWO
AT NODE 81 IN THE X DIRECTION
FIG. 5.14

LEGEND

+—+—+ TAN STIFFNESS		2—2—2 ENERGY SEARCH
3—3—3 SEC STIFFNESS		4—+—+ KAR'S METHOD
5—5—5 EXPT. RESULT		
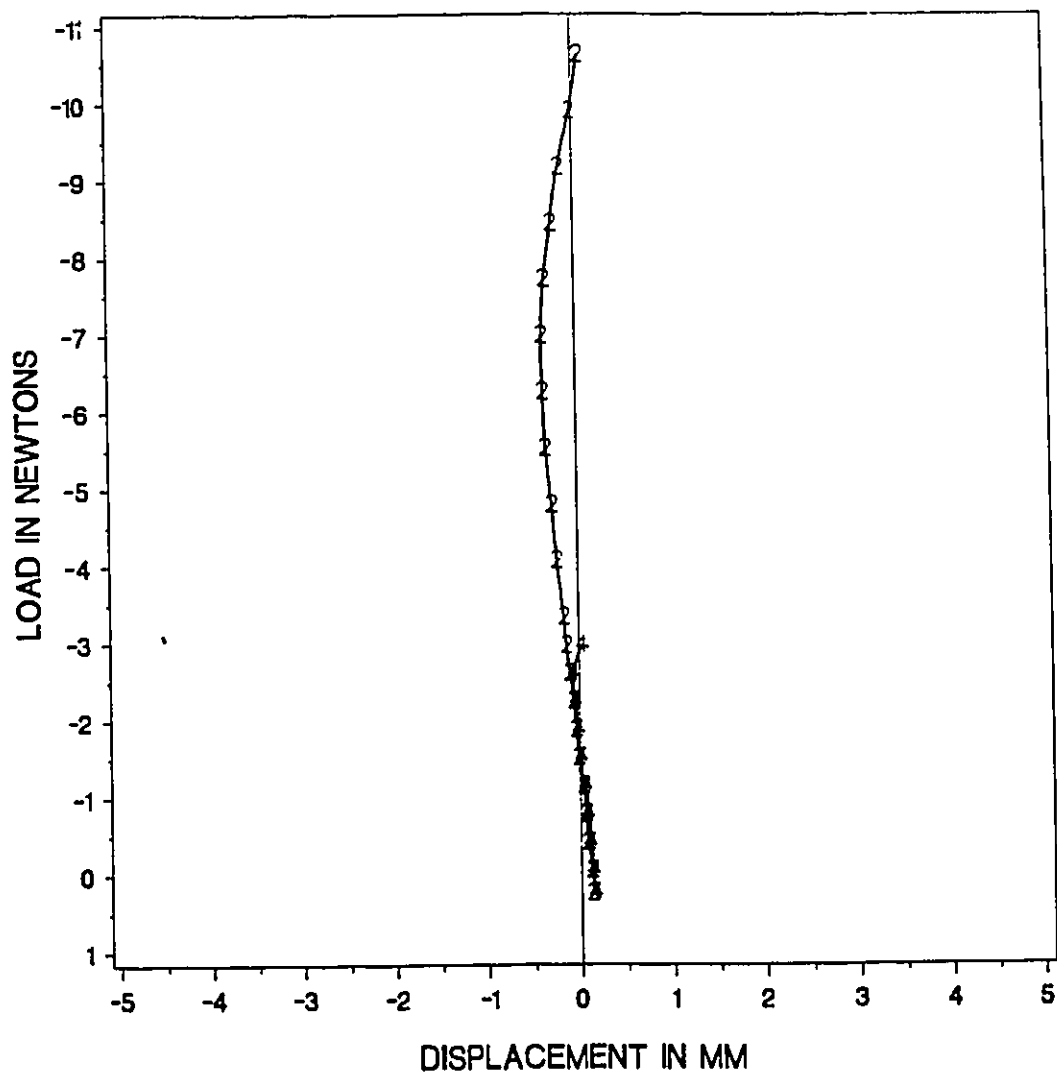
LEGEND   +—+—+ TAN STIFFNESS   2—2—2 ENERGY SEARCH
                 3—3—3 SEC STIFFNESS   4—4—4 KAR'S METHOD

LOAD-DISPLACEMENT CURVES FOR CASE TWO
AT NODE 77 IN THE Z DIRECTION
FIG. 5.15

LEGEND   +—+—+ TAN STIFFNESS   2—2—2 ENERGY SEARCH
            3—3—3 SEC STIFFNESS   4—4—4 KAR'S METHOD

LOAD-DISPLACEMENT CURVES FOR CASE TWO
AT NODE 77 IN THE X DIRECTION
FIG. 5.16

LEGEND   +—+—+ TAN STIFFNESS   2—2—2 ENERGY SEARCH
             3—3—3 SEC STIFFNESS   4—4—4 KAR'S METHOD

LOAD-DISPLACEMENT CURVES FOR CASE TWO
AT NODE 73 IN THE X DIRECTION
FIG. 5.17
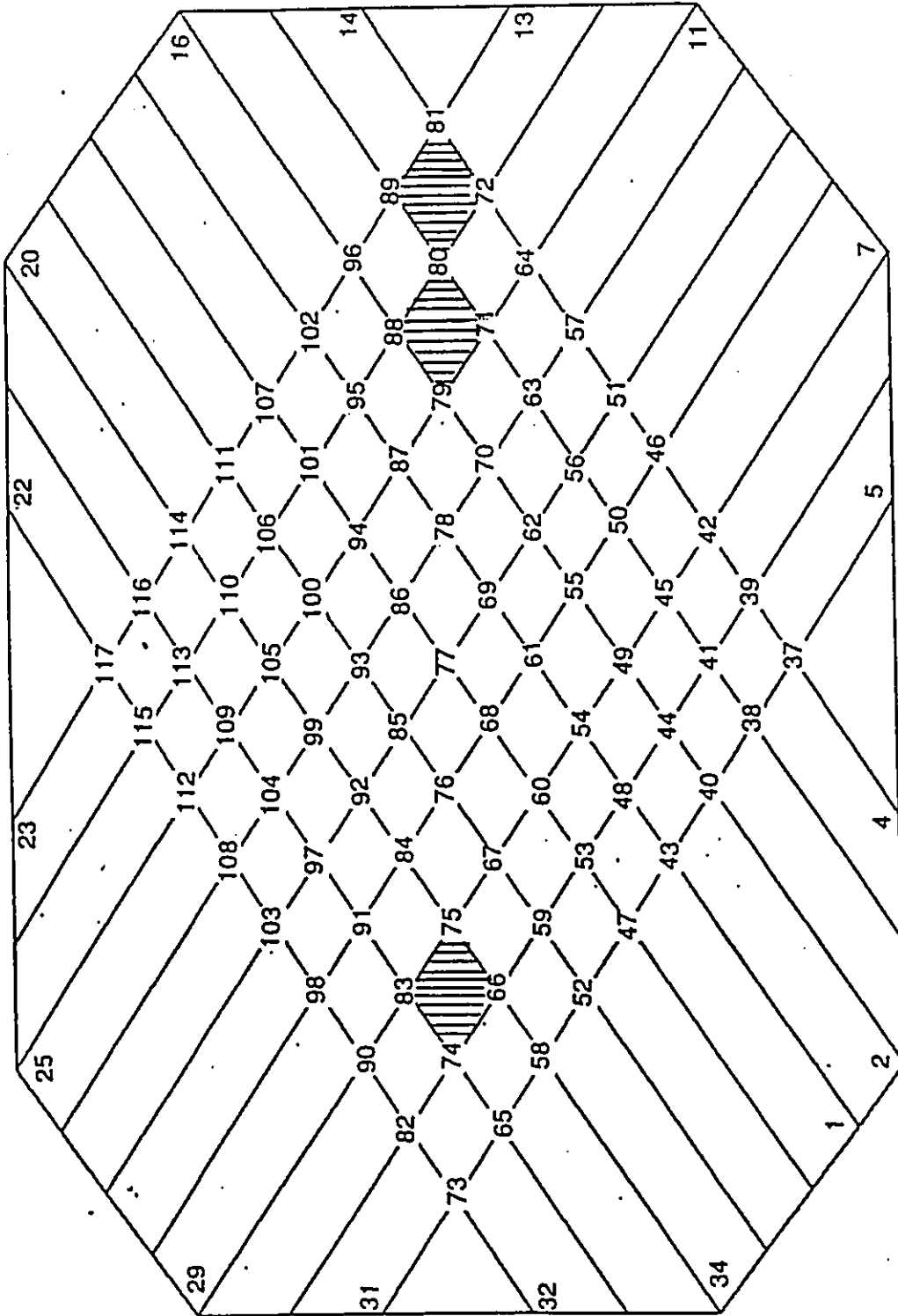
FIG. 5.18
CASE THREE

LEGEND
+—+—+ TAN STIFFNESS   2—2—2 ENERGY SEARCH
3—3—3 SEC STIFFNESS   4—4—4 KAR'S METHOD
5—5—5 EXPT. RESULT

LOAD-DISPLACEMENT CURVES FOR CASE THREE
AT NODE 80 IN THE Z DIRECTION
FIG. 5.19

LEGEND
+—+—+ TAN STIFFNESS   2—2—2 ENERGY SEARCH
3—3—3 SEC STIFFNESS   4—4—4 KAR'S METHOD
5—5—5 EXPT. RESULT

LOAD-DISPLACEMENT CURVES FOR CASE THREE
AT NODE 81 IN THE Z DIRECTION
FIG. 5.20

FIG. 5.21
CASE FOUR

LOAD-DISPLACEMENT CURVES FOR CASE FOUR
AT NODE 81 IN THE X DIRECTION
FIG. 5.22

LOAD-DISPLACEMENT CURVES FOR CASE FOUR
AT NODE 73 IN THE X DIRECTION
FIG. 5.23

LEGEND
+——+——+ TAN STIFFNESS   2—2—2 ENERGY SEARCH
3—3—3 SEC STIFFNESS   4—4—4 KAR'S METHOD
5—5—5 EXPT RESULT

LOAD-DISPLACEMENT CURVES FOR CASE FOUR
AT NODE 81 IN THE Z DIRECTION
FIG. 5.24

LOAD-DISPLACEMENT CURVES FOR CASE FOUR
AT NODE 73 IN THE Z DIRECTION
FIG. 5.25

LEGEND   +—+—+ TAN STIFFNESS   2—2—2 ENERGY SEARCH
               3—3—3 SEC STIFFNESS   4—4—4   KAR'S METHOD

LOAD-DISPLACEMENT CURVES FOR CASE FOUR
AT NODE 77 IN THE X DIRECTION
FIG. 5.26
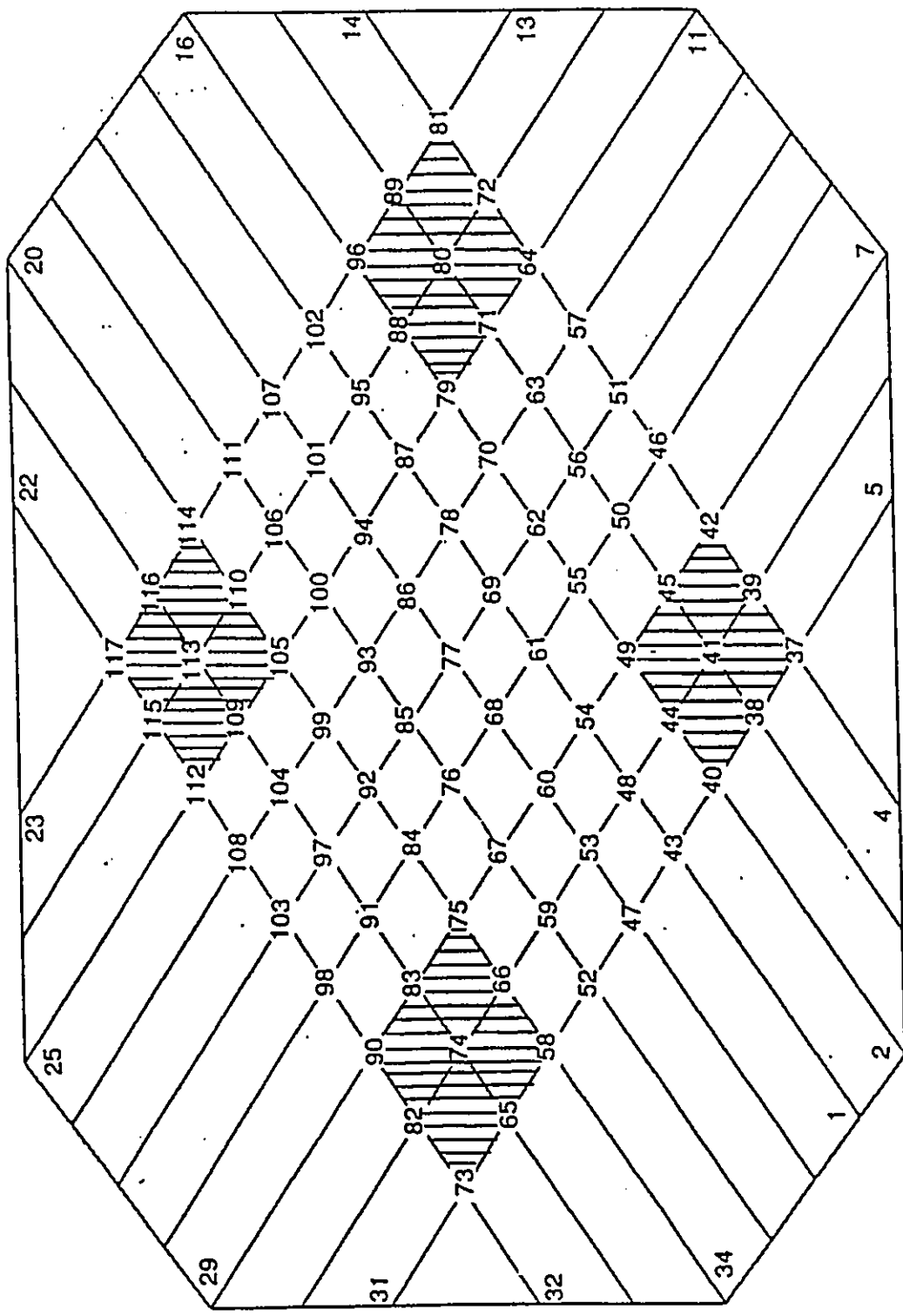
FIG. 5.27
CASE FIVE

LEGEND +—+—+ TAN STIFFNESS   2—2—2 ENERGY SEARCH
        3—3—3 SEC STIFFNESS   4—4—4 KAR'S METHOD
        5—5—5 EXPT RESULT

LOAD-DISPLACEMENT CURVES FOR CASE FIVE
AT NODE 37 IN THE Y DIRECTION
FIG. 5.28
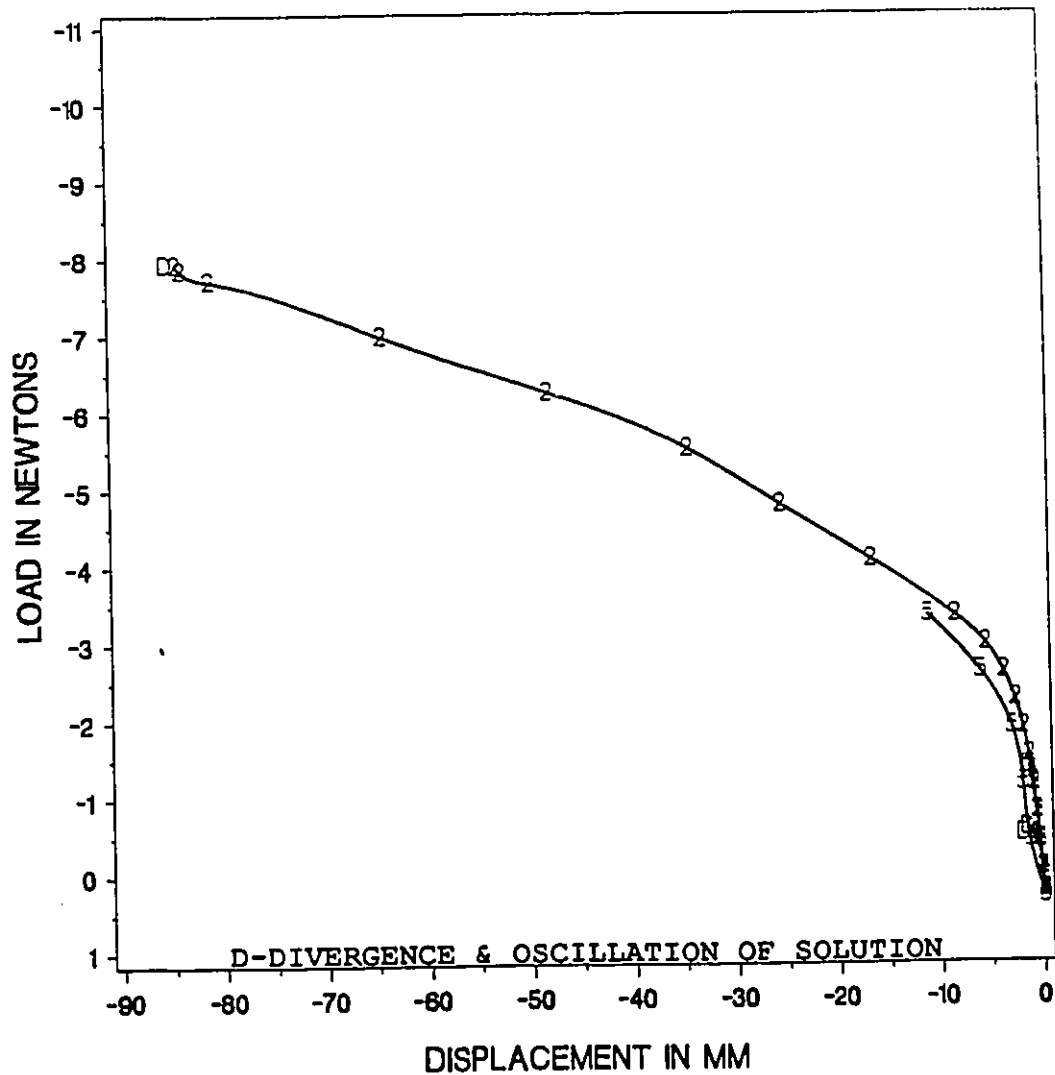
LOAD IN NEWTONS

DISPLACEMENT IN MM

D-DIVERGENCE & OSCILLATION OF SOLUTION

LEGEND   +—+—+ TAN STIFFNESS   2—2—2 ENERGY SEARCH
          3—3—3 SEC STIFFNESS   4—4—4 KAR'S METHOD
                EXPT RESULT
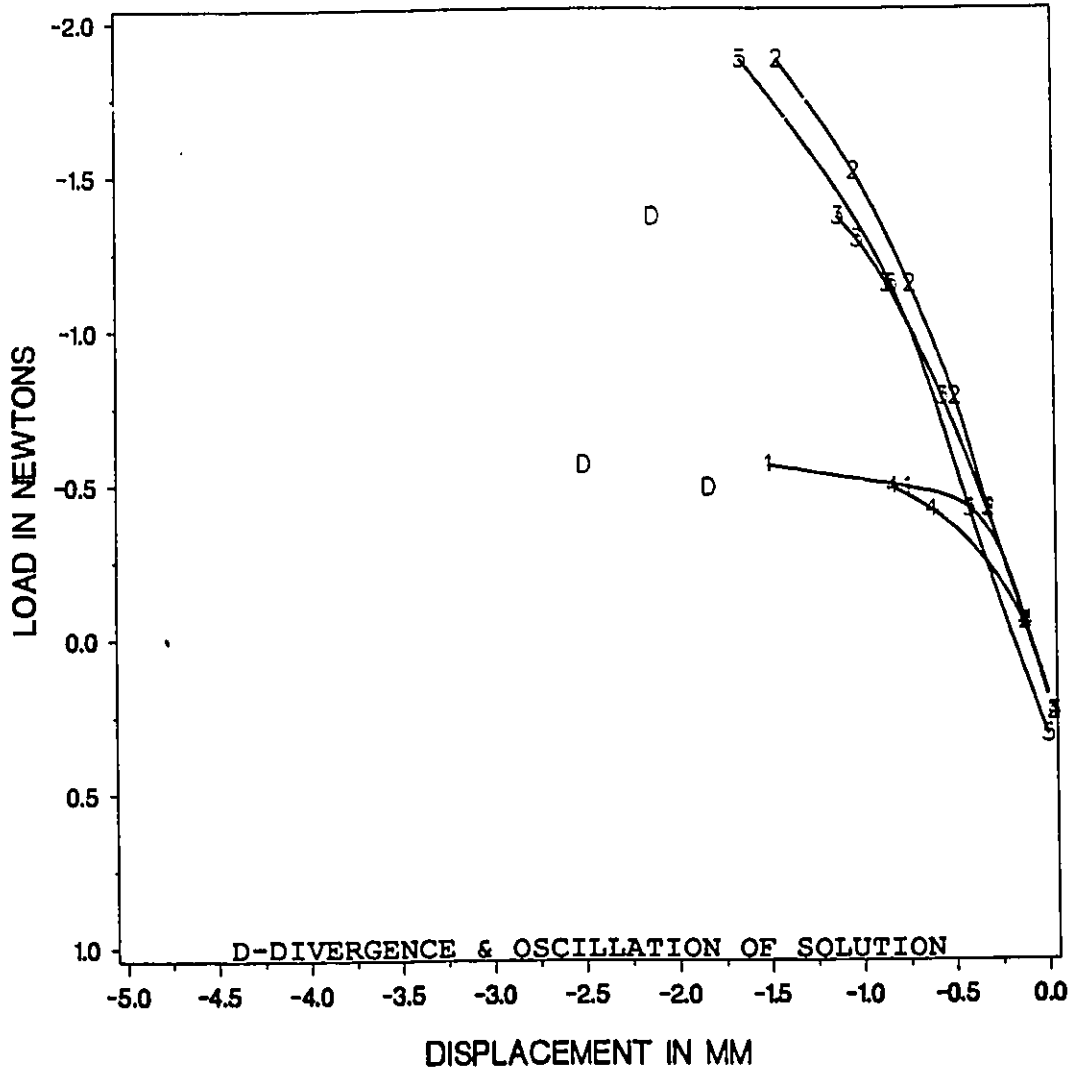
LOAD-DISPLACEMENT CURVES FOR CASE FIVE
AT NODE 37 IN THE Z DIRECTION
FIG. 5.29

LOAD-DISPLACEMENT CURVES FOR CASE FIVE
AT NODE 73 IN THE Z DIRECTION
FIG. 5.30
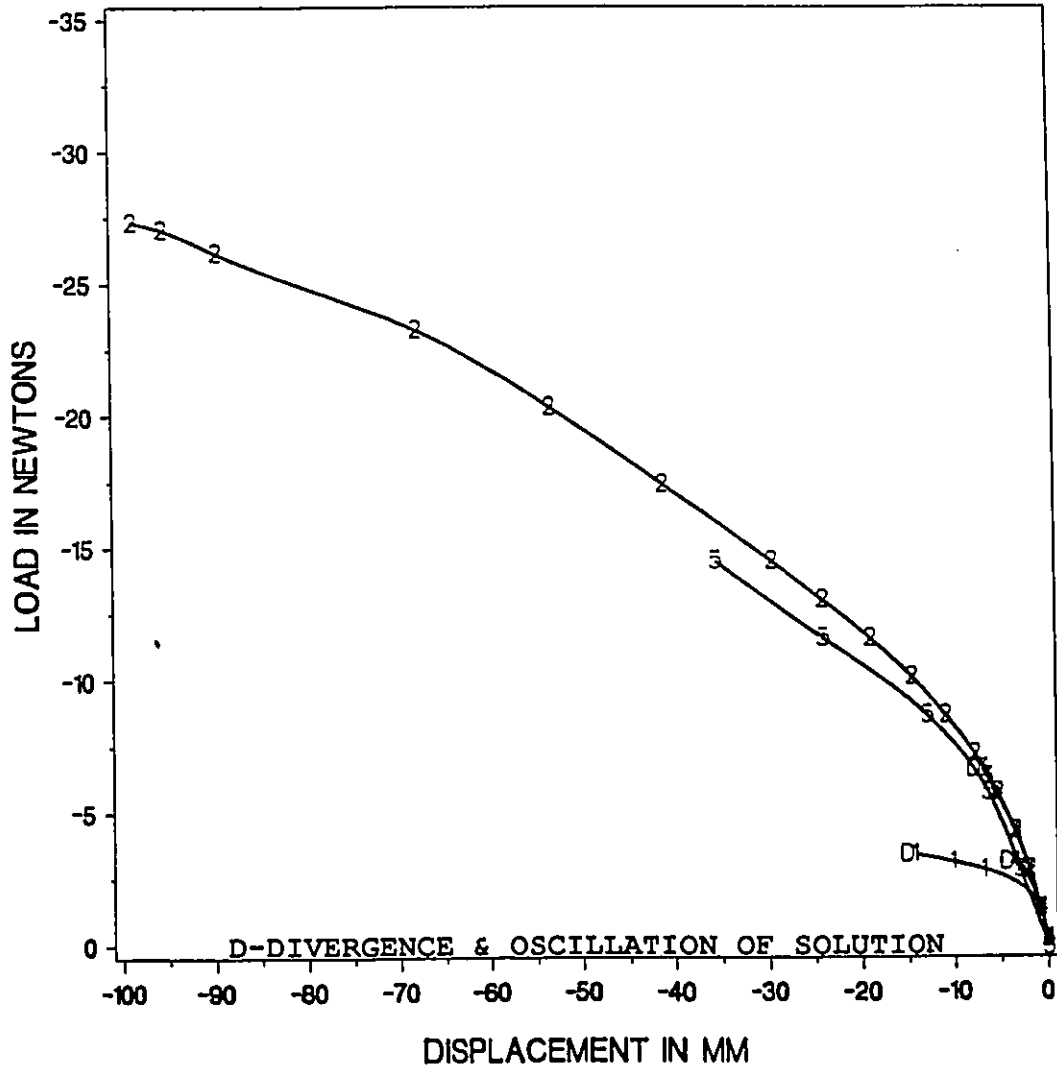
LOAD-DISPLACEMENT CURVES FOR CASE FIVE
AT NODE 74 IN THE Z DIRECTION
FIG. 5.31
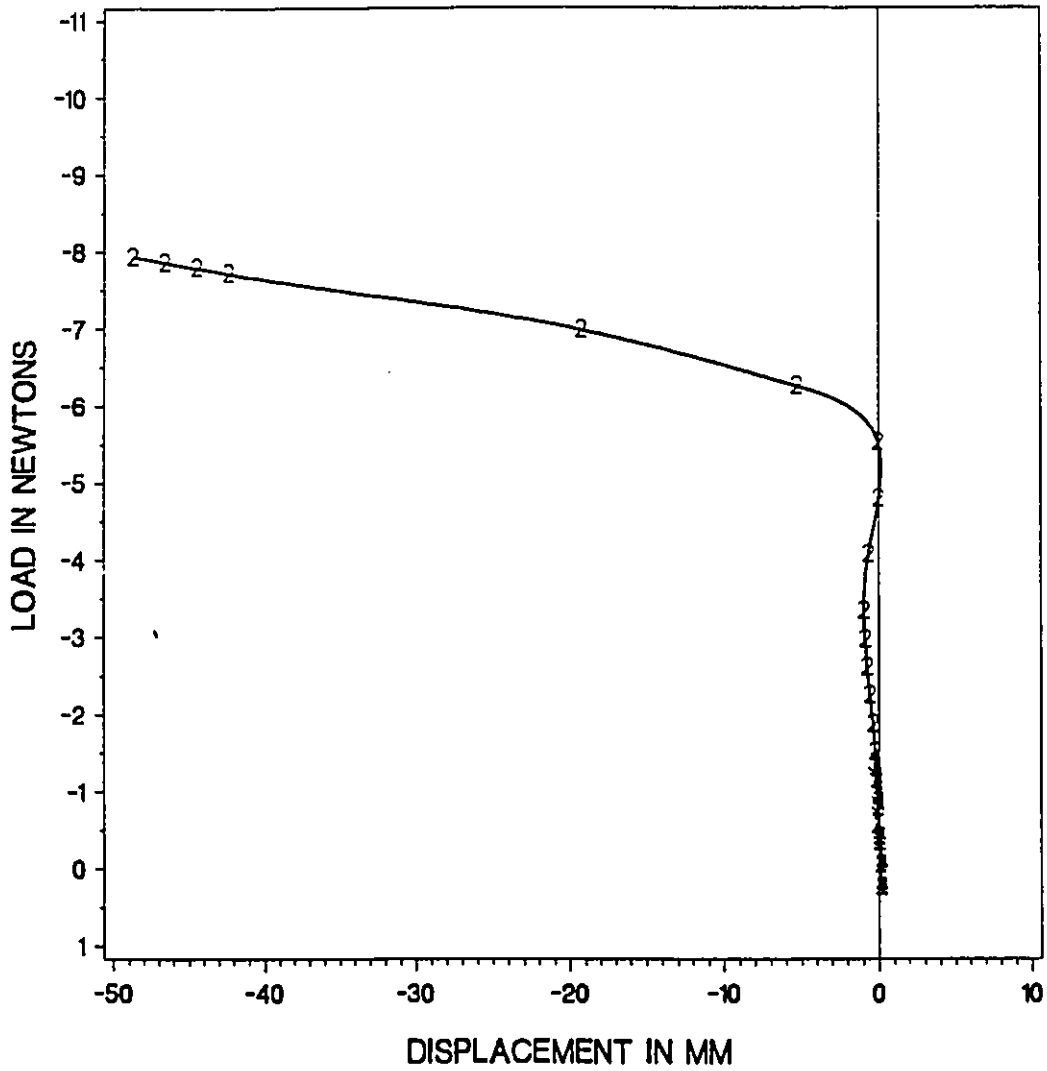
LOAD-DISPLACEMENT CURVES FOR CASE FIVE
AT NODE 77 IN THE Z DIRECTION
FIG. 5.32
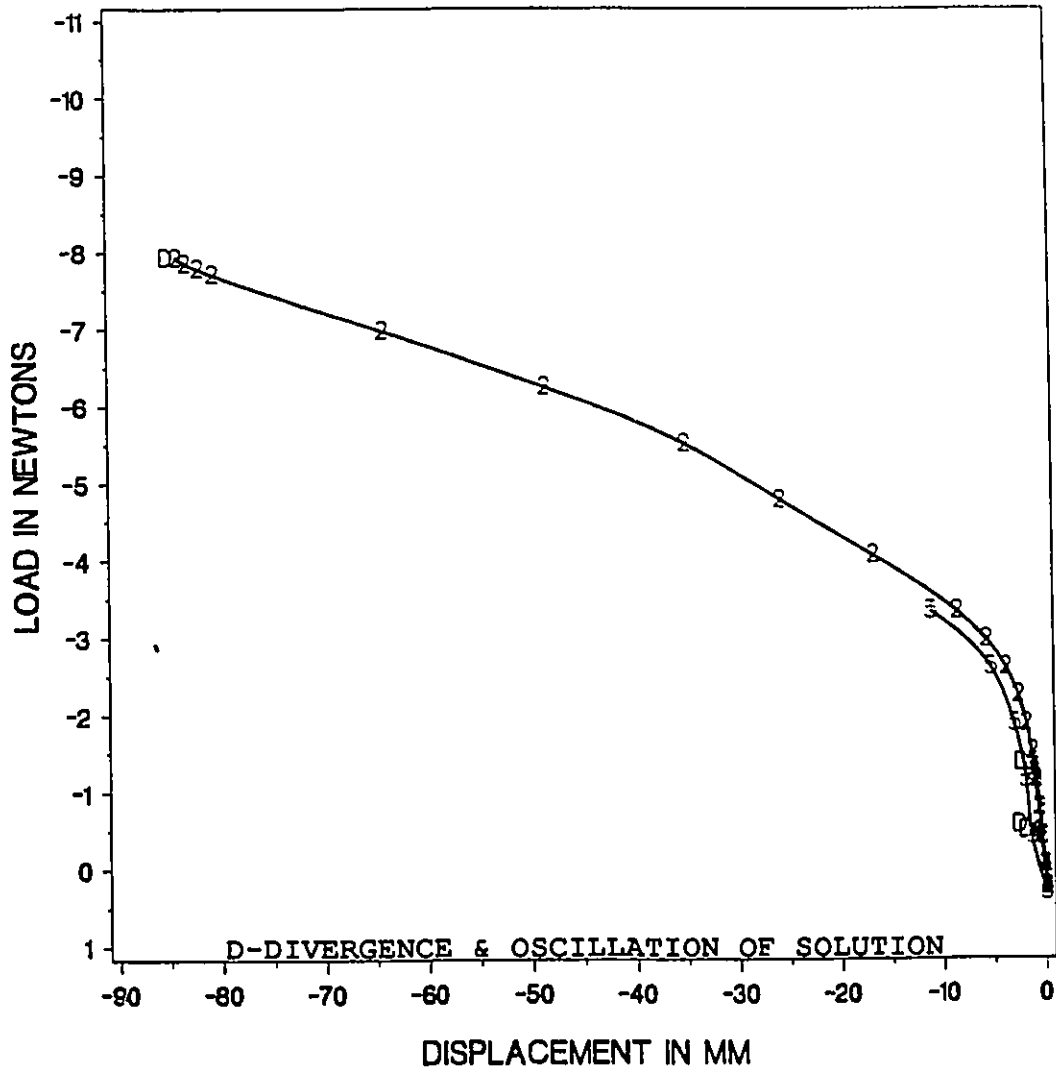
D-DIVERGENCE & OSCILLATION OF SOLUTION

LOAD IN NEWTONS

DISPLACEMENT IN MM

LEGEND
TAN STIFFNESS
SEC STIFFNESS
EXPT RESULT
2—2—2 ENERGY SEARCH
4—4—4 KAR'S METHOD

LOAD-DISPLACEMENT CURVES FOR CASE FIVE
AT NODE 81 IN THE Z DIRECTION
FIG. 5.33
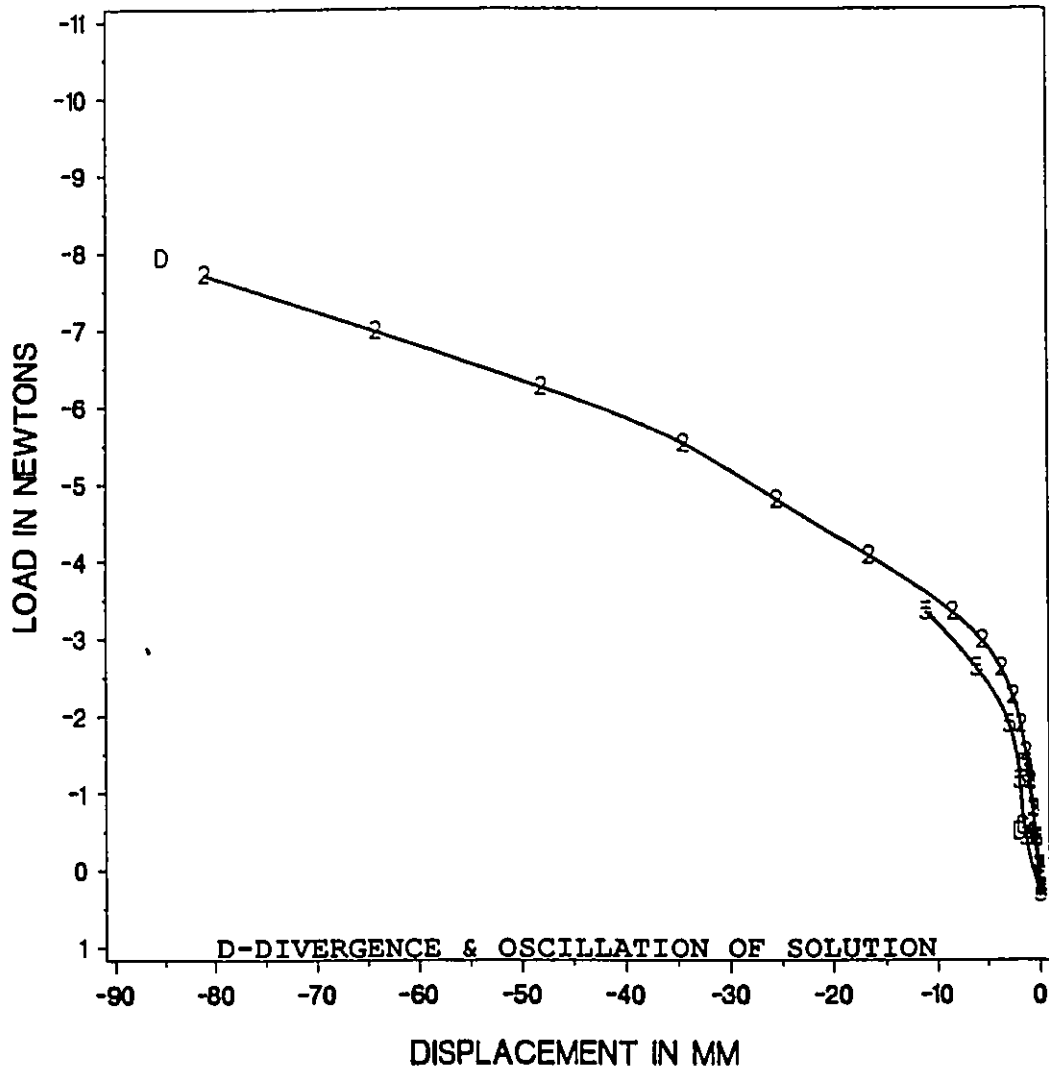
LOAD-DISPLACEMENT CURVES FOR CASE FIVE
AT NODE 117 IN THE Z DIRECTION
FIG. 5.34
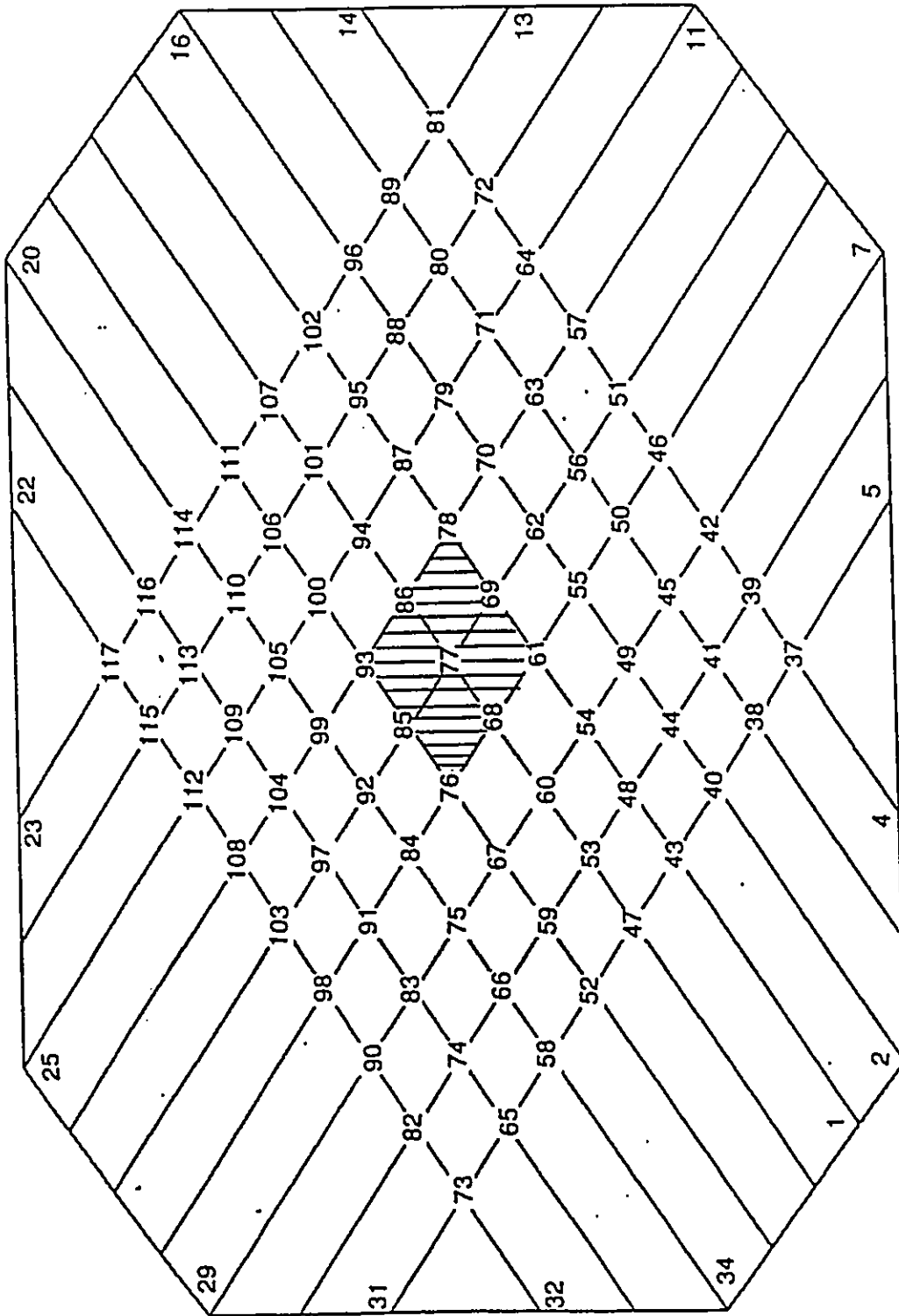
FIG. 5.35
CASE SIX
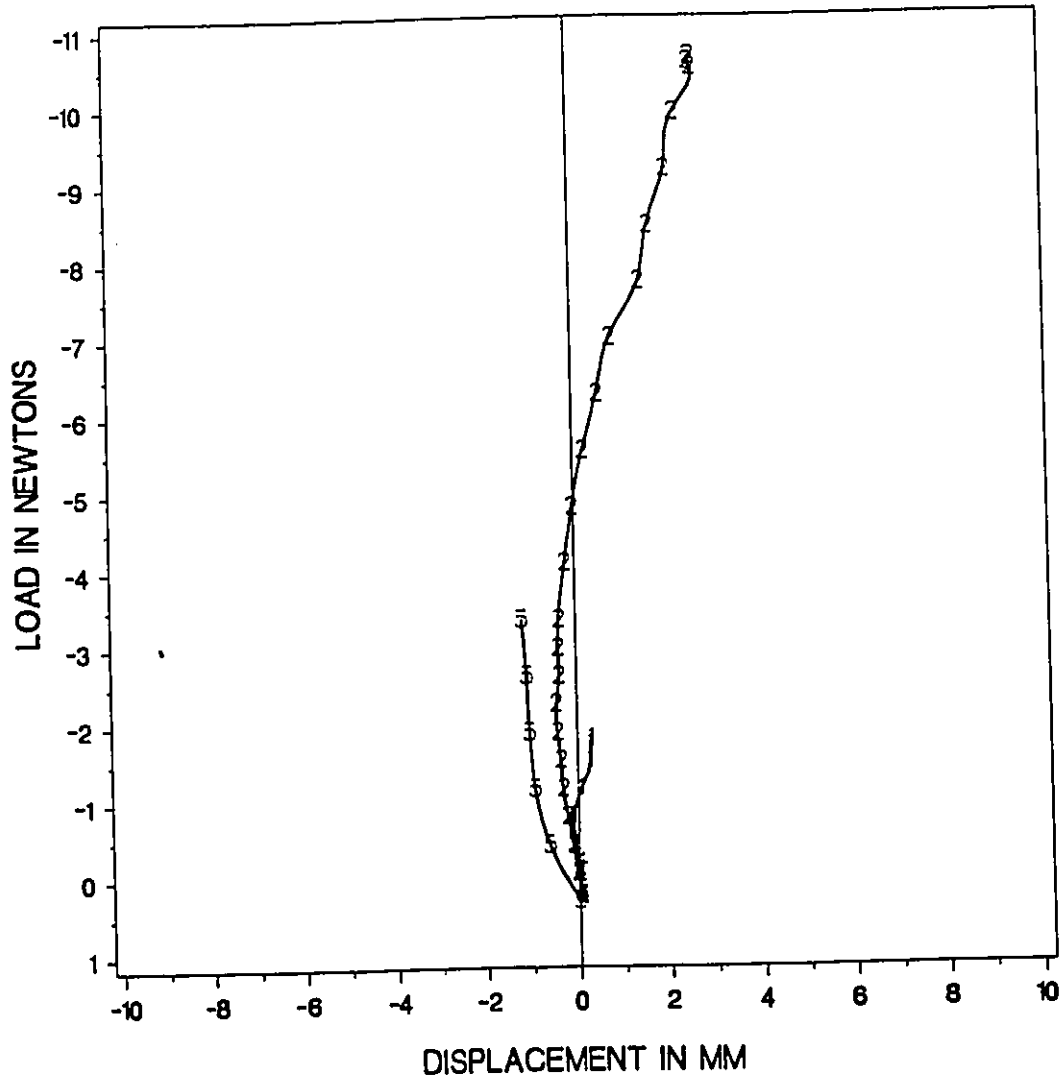
LEGEND
+—+—+ TAN STIFFNESS
⌐—⌐—⌐ SEC STIFFNESS
⌐—⌐—⌐ EXPT RESULT

2—2—2 ENERGY SEARCH
4—4—4 KAR'S METHOD

LOAD-DISPLACEMENT CURVES FOR CASE SIX
## AT NODE 37 IN THE Z DIRECTION
FIG. 5.36
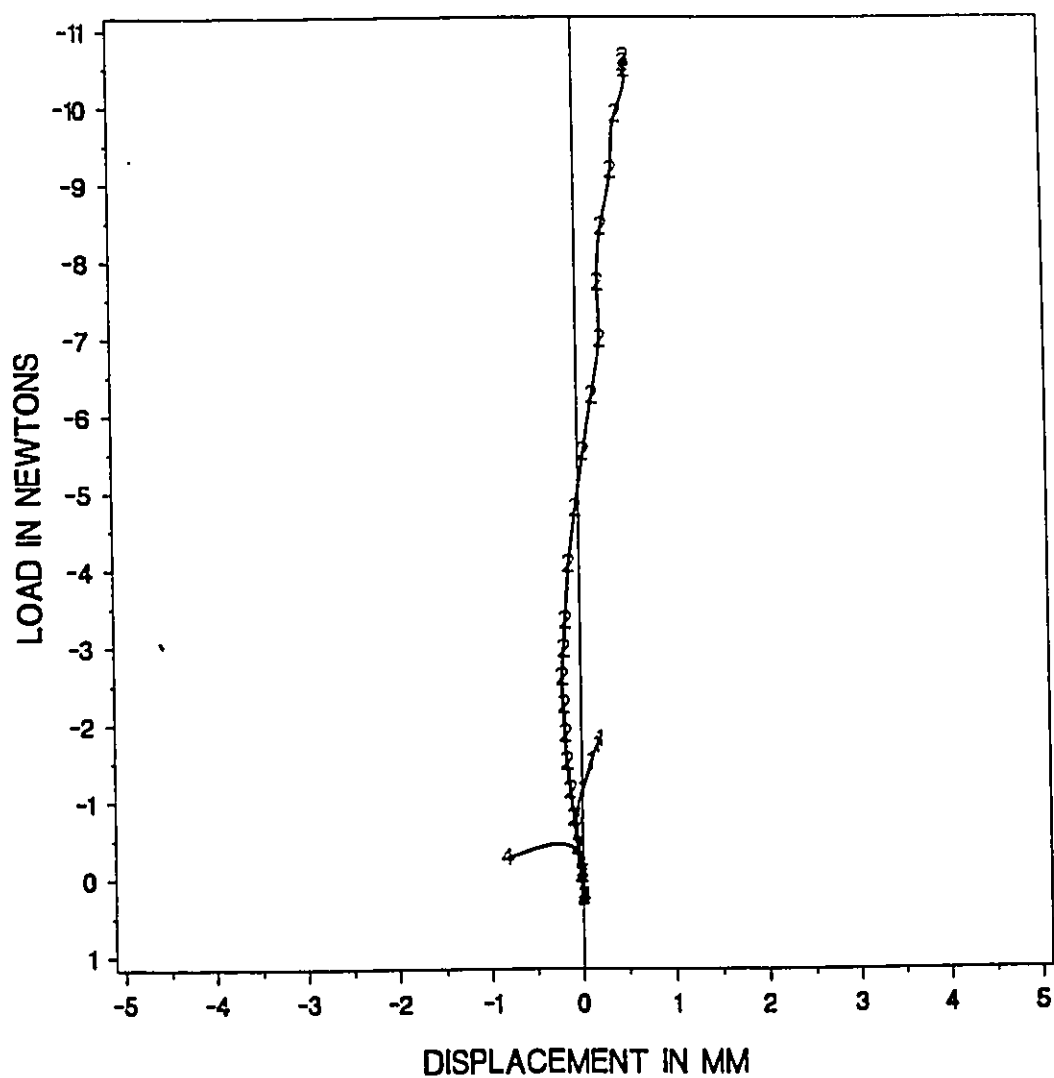
LOAD-DISPLACEMENT CURVES FOR CASE SIX
AT NODE 73 IN THE X DIRECTION
FIG. 5.37
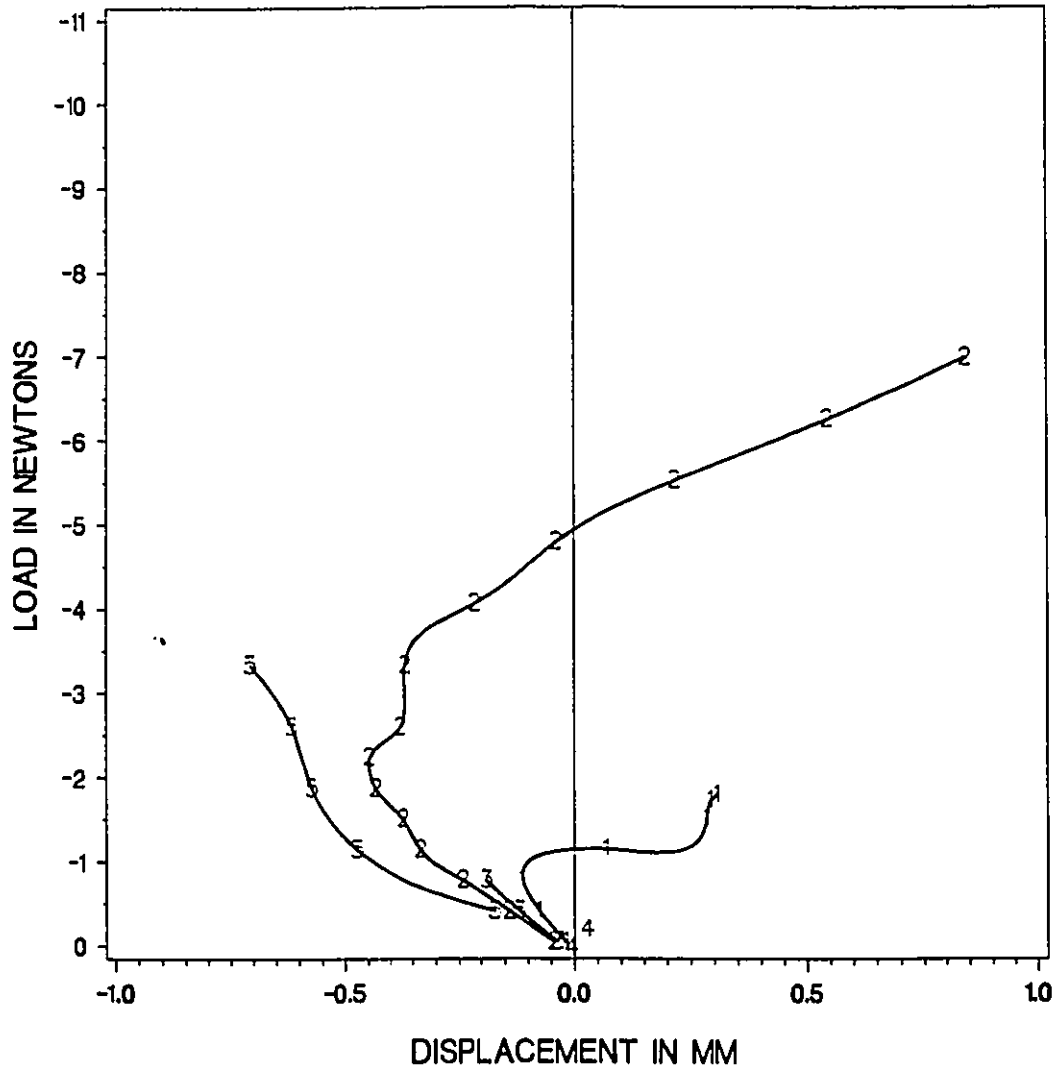
LOAD-DISPLACEMENT CURVES FOR CASE SIX
AT NODE 117 IN THE Z DIRECTION
FIG. 5.38
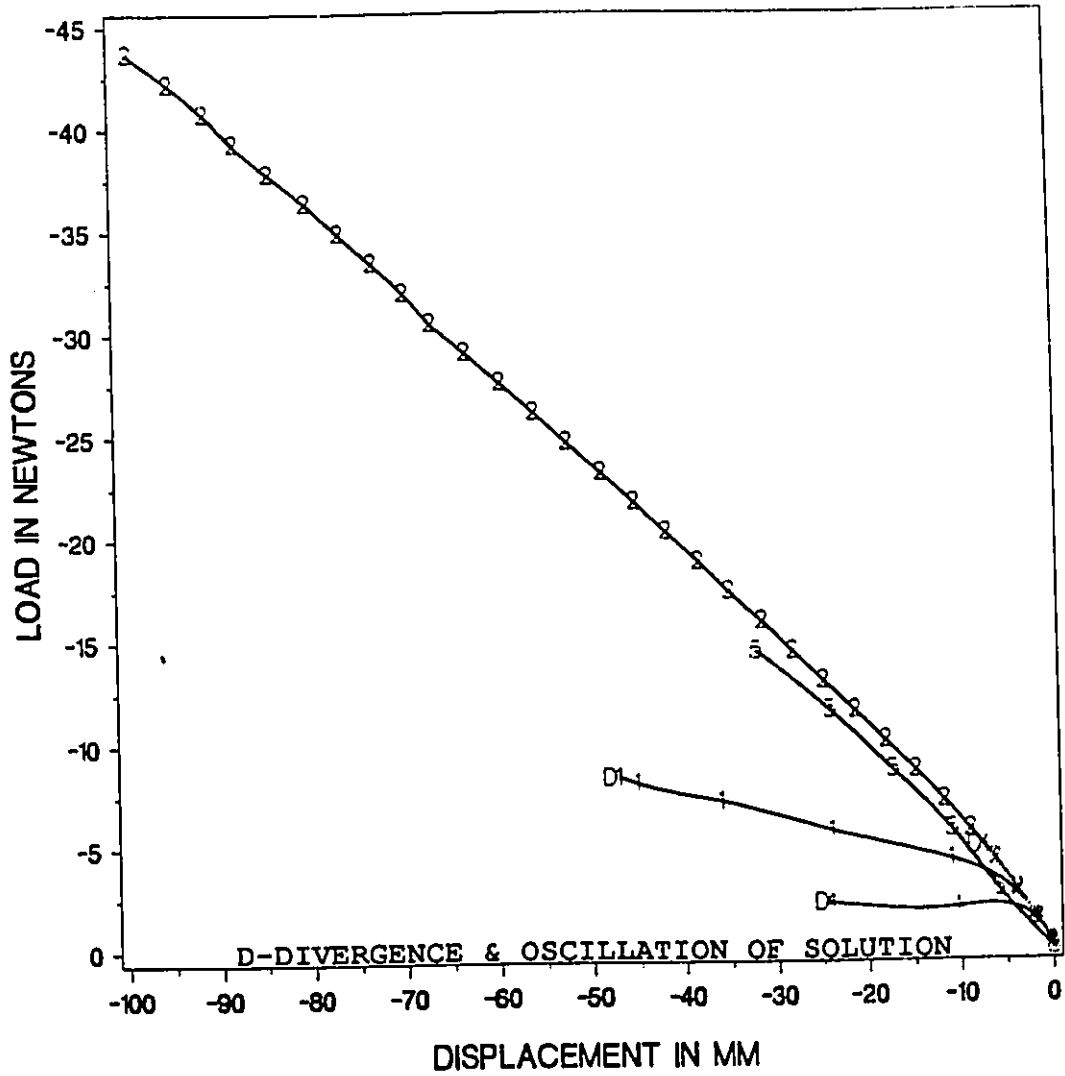
116

LOAD-DISPLACEMENT CURVES FOR CASE SIX
AT NODE 77 IN THE Z DIRECTION
FIG. 5.39
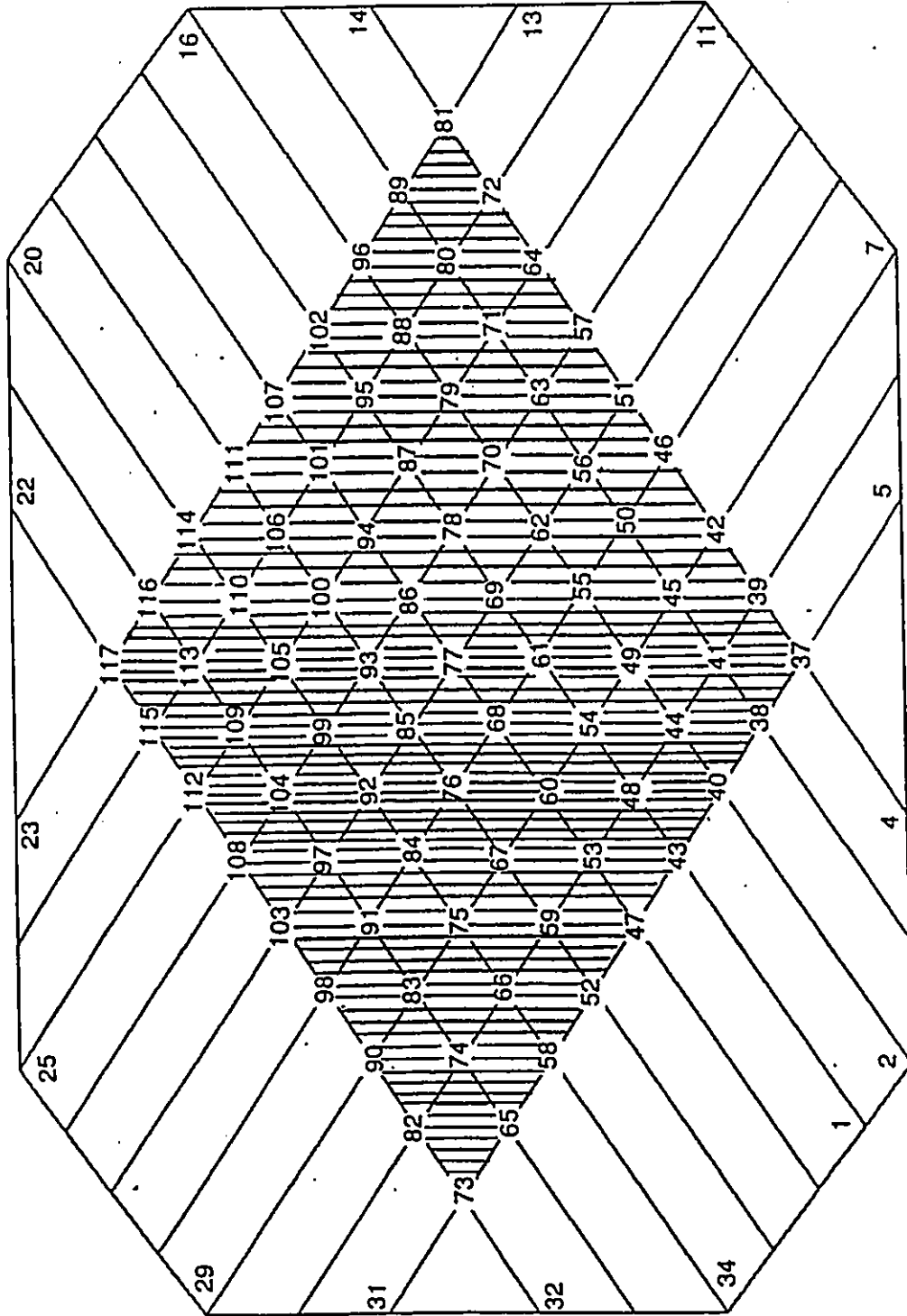
FIG. 5.40
CASE SEVEN
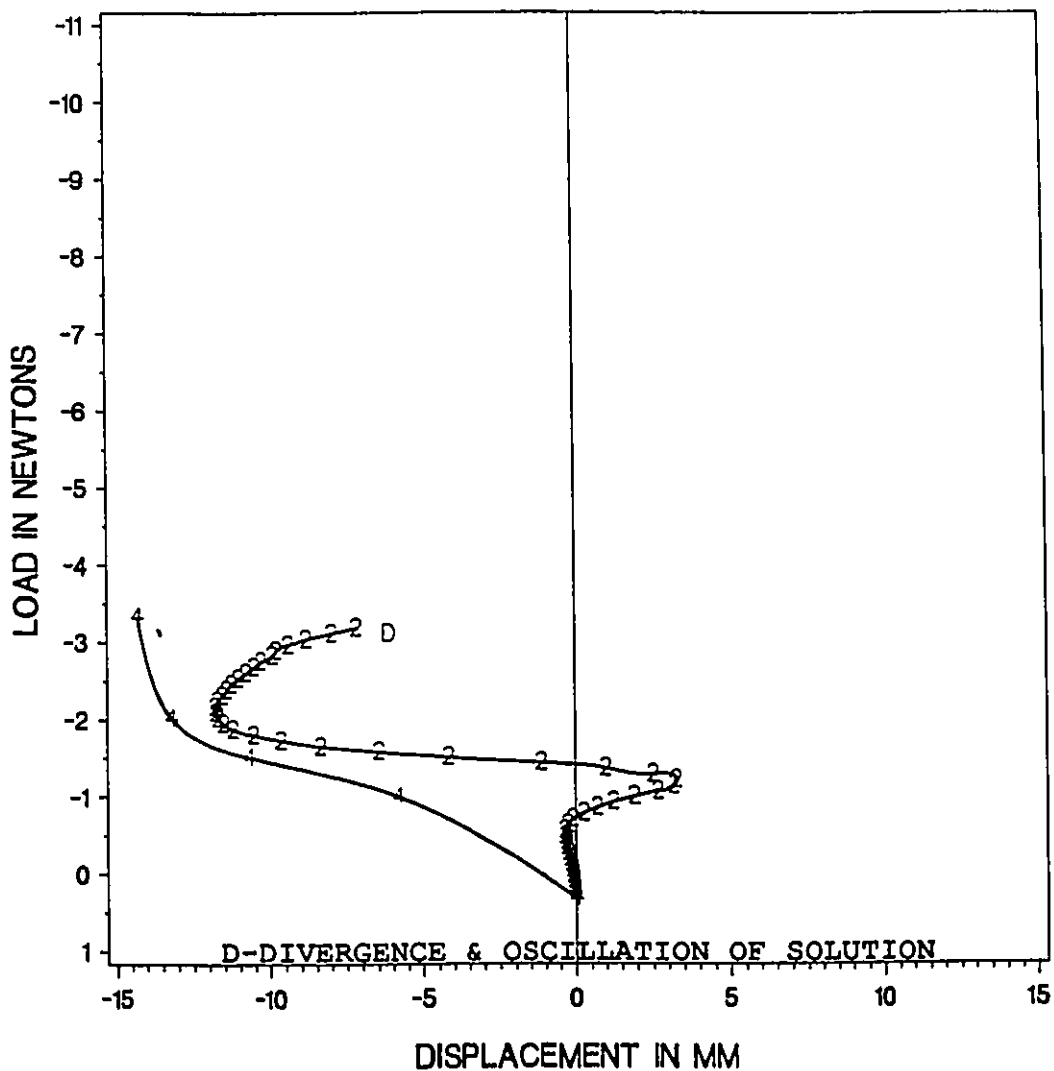
LEGEND +—·—+ TAN STIFFNESS   2—2—2 ENERGY SEARCH
        3—3—3 SEC STIFFNESS   4—4—4 EXPT RESULT

LOAD-DISPLACEMENT CURVES FOR CASE SEVEN
AT NODE 117 IN THE Y DIRECTION
FIG. 5.41
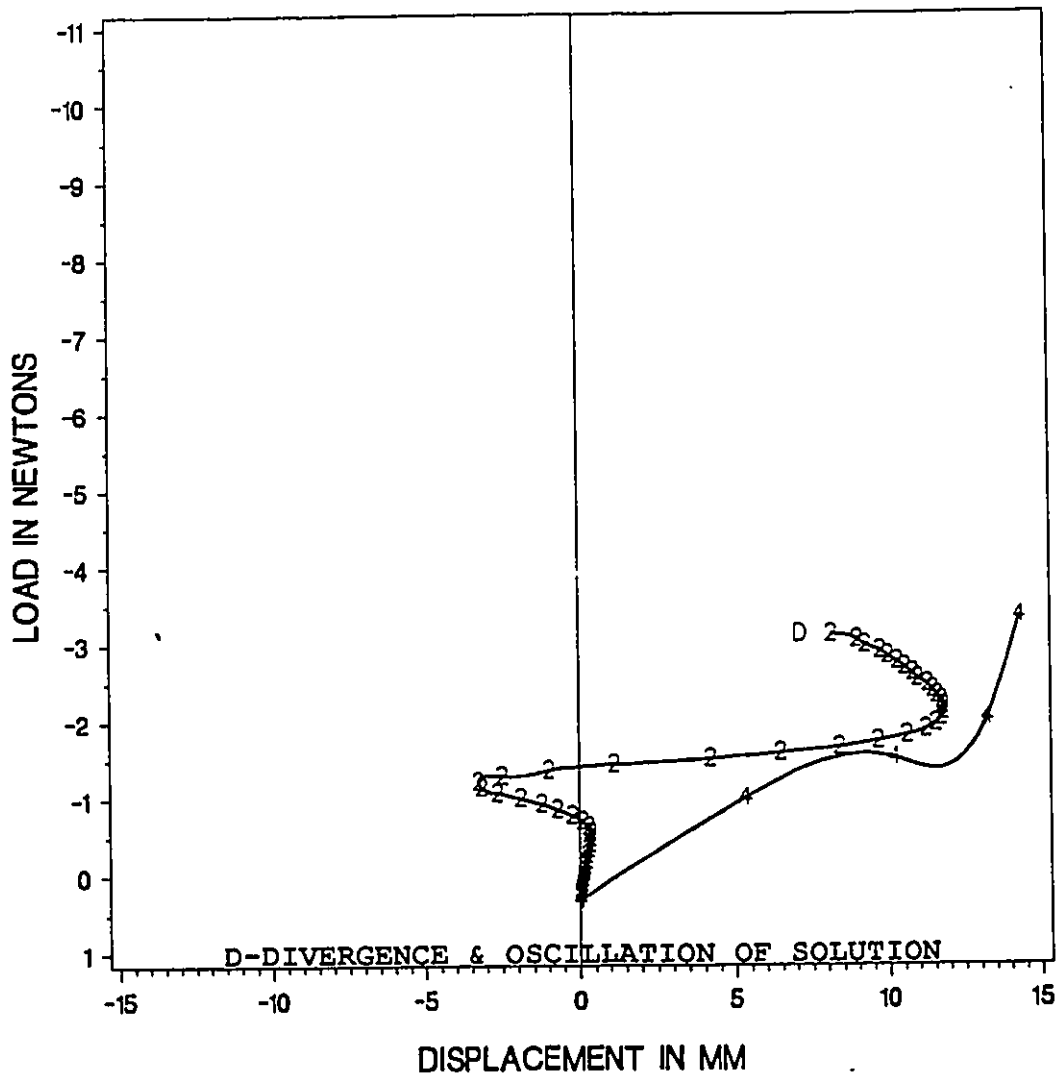
LOAD-DISPLACEMENT CURVES FOR CASE SEVEN
AT NODE 37 IN THE Y DIRECTION
FIG. 5.42
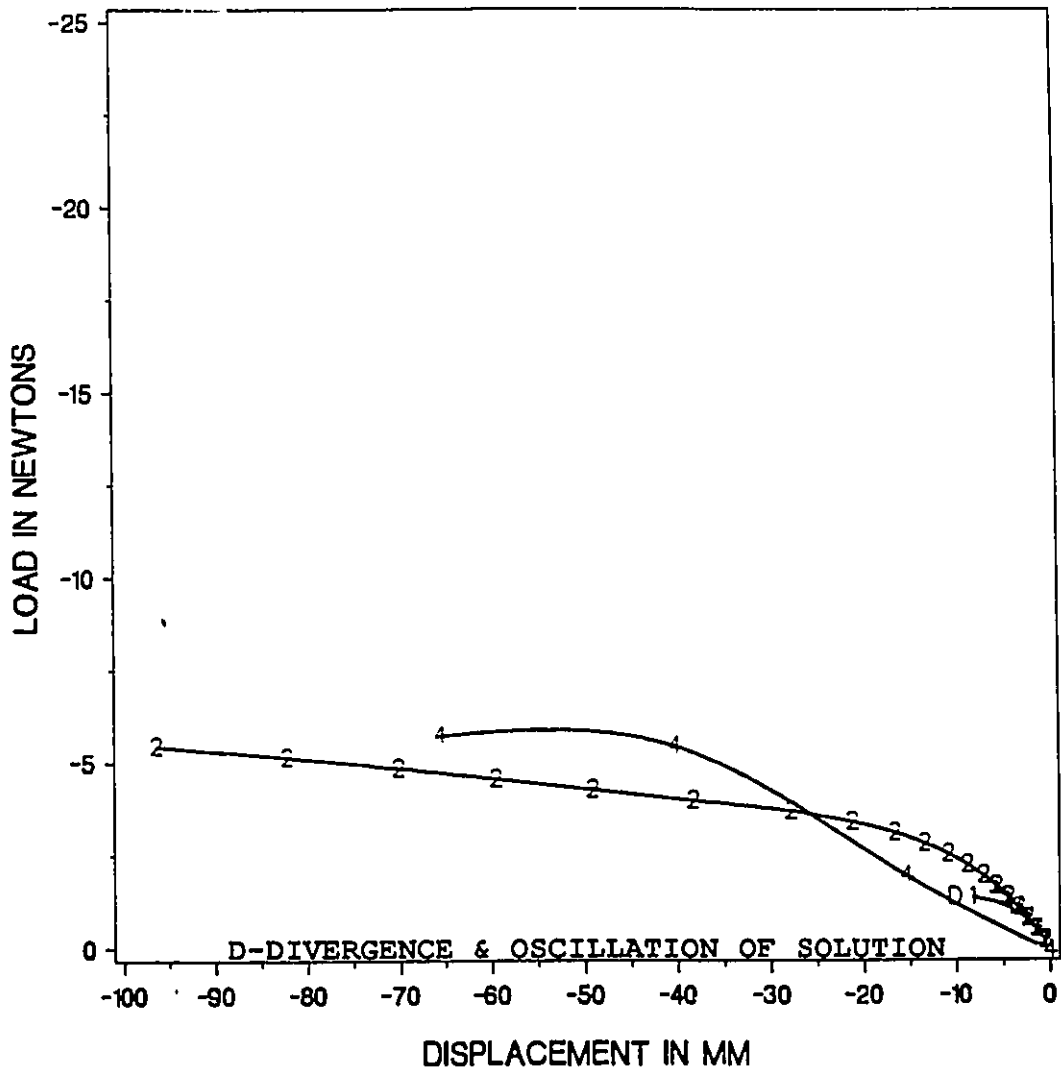
LEGEND   +—+—+ TAN STIFFNESS   2—2—2 ENERGY SEARCH
              3—3—3 SEC STIFFNESS   4—4—4   EXPT RESULT

LOAD-DISPLACEMENT CURVES FOR CASE SEVEN
AT NODE 77 IN THE Z DIRECTION
FIG. 5.43

Chapter VI

CONCLUSIONS AND RECOMMENDATIONS

The finite element method has been incorporated into all of the solution techniques presented here. The matrix methods of analysis are advantageous in that the convergence, if achieved, is done so in a very few iterative cycles. The tangent stiffness method has proved to be the most efficient when the nonlinearity is not pronounced since it converges in fewer iterations than the secant stiffness method; but its inherent drawback lies in its relative inaccuracy in predicting displacements when geometric nonlinearity sets into the structure. The secant stiffness method is slightly better in the prediction of displacements and follows the experimental values more closely; its estimation of failure load is also much lower than predicted by the energy search method although relatively better than that of the tangent stiffness model. Kar's modification applied to the tangent stiffness method did not hasten convergence in any way; but it has resulted in the solutions approximating the energy search method more closely and the estimation of failure load is about 10% higher than that of the secant stiffness method. However, all of the above methods are beset by convergence difficulties when the structure is loaded into the nonlinear range.

On the other hand, the energy search method permits the prediction of large nodal displacements and its formulation also allows the detection of general instabilities.The potential energy formulation is also much simpler to construct than the corresponding matrix methods of analysis. The calculation of the total potential energy is simply the scalar sum of the energy contributions from the individual members which comprise the structure. No convergence

problems were encountered except in cases of highly nonlinear solutions due to large applied loads. The only disadvantage in the use of the energy search method is the amount of time taken to converge to a solution. It is about five times slower than the conventional matrix methods.

The variable metric method[6] of Fletcher and Powell was not used in the search for a minimum because of the need for more storage as compared to similar techniques. Hence it was decided to use the Fletcher-Reeves[7] algorithm of conjugate gradients. The scaling technique proposed by Fox and Stanton[8] helped to hasten convergence.

The conclusions reached in this study were:

1.   There is little or no difference in results obtained by one step matrix solution techniques and the energy search method in the initial stage of loading when geometric or material nonlinearity is not pronounced.

2.   The stiffness method of analysis is more suitable when the strcuture is very stable, i.e., in case of internal pressure loading, as the solution converges very quickly.

3.   The amount of execution time taken by the matrix methods is roughly one-fifth that of the energy search method.

4.   Matrix methods of analysis failed to converge in the first few iterations in an unstiffened state i.e., when the member forces are zero. Such problems did not exist with the energy search method.

5.   The one step solution techniques were beset by convergence difficulties and predicted the failure load to be much lower than was the case. In the energy search method, no convergence problems were encountered until the ultimate load capacity of the structure was reached.

6.   Elaborate matrix formulations were not required in the case of the energy search method. The calculation of the total potential energy was simply the scalar of the potential energies of the individual members comprising the structure.

7.   The proposed modification of Kar to improve the rate of convergence was not successful in nonlinear situations.

Based on the above, it is recommended that for structures whose service load may not exceed 20% of the ultimate load, it may be more economical to use matrix methods of analysis as they show little or no variation with the energy search method. But in cases where the service load is expected to be beyond 20% of the ultimate load capacity of the structure, the energy search procedure proves to be the more accurate method albeit time consuming.

For future research, it is recommended to study the membrane-cable interaction and the effect of marriage between these two elements. It is also recommended that future studies use optical techniques to have relative ease and accuracy in measuring displacements.

Appendix A

STRAIN ENERGY GRADIENT

The analytic gradients of the element strain energy given by equations (2.15) and (2.22) for elastic and inelastic tension members respectively is given by:

$$\frac{\partial \Xi}{\partial \bar{u}_p} = -\frac{\partial \Xi}{\partial \bar{u}_q} = -\zeta \frac{\Delta_x}{S^2} \qquad (A.1)$$

$$\frac{\partial \Xi}{\partial \bar{v}_p} = -\frac{\partial \Xi}{\partial \bar{v}_q} = -\zeta \frac{\Delta_y}{S^2} \qquad (A.2)$$

$$\frac{\partial \Xi}{\partial \bar{w}_p} = -\frac{\partial \Xi}{\partial \bar{w}_q} = -\zeta \frac{\Delta_z}{S^2} \qquad (A.3)$$

where

$$\Delta_x = (\bar{X}_q + \bar{u}_q) - (\bar{X}_p + \bar{u}_p)$$

$$\Delta_y = (\bar{Y}_q + \bar{v}_q) - (\bar{Y}_p + \bar{v}_p)$$

$$\Delta_z = (\bar{Z}_q + \bar{w}_q) - (\bar{Z}_p + \bar{w}_p)$$

$$\zeta = \Xi + Tl_1 \qquad when \, \varepsilon \leq \varepsilon_e$$

$$\zeta = \Xi + Tl_1(2 - \frac{l_1}{l_2}) \qquad when \, \varepsilon > \varepsilon_p$$

and Ξ, T, l_1 and l_2 are as defined in Chapter 2.

Appendix B

GEOMETRIC STIFFNESS MATRIX OF AN AXIALLY LOADED

MEMBER

The member shown in figures B.1 and B.2 has a length l_1 and lies between nodal points p and q. In the initial state 1, the member lies in an $\bar{X}$–$\bar{Y}$ plane and the force at each end of the member is given in vector form by:

$$\bar{T}_1 = \begin{bmatrix} \cos\alpha_1 \\ \sin\alpha_1 \\ \sin\beta_1 \end{bmatrix} T \tag{B.1}$$

where

$$T = scalar\ magnitude\ of\ \bar{T}_1$$

$$\alpha_1 = angle\ of\ the\ local\ x'\ axis\ of\ the\ member\ the\ \bar{X}\ axis$$

$$\beta_1 = angle\ about\ the\ local\ y'\ axis\ measured\ the\ \bar{X}\text{–}\bar{Y}\ plane\ the\ local\ x'\ axis\ and\ is\ selected\ be\ equal\ zero$$

If a small rigid body rotation is given to the member, the resultant rotation in state 2 has components $\delta\alpha_{12}$ about the z' axis and $\delta\beta_{12}$ about the y' axis. Therefore the force at each end of the member is given by:

$$\bar{T}_2 = \begin{bmatrix} \cos\alpha_2 \\ \sin\alpha_2 \\ \sin\beta_2 \end{bmatrix} T \tag{B.2}$$

where the scalar magnitude of $\bar{T}_2$ is the same as $\bar{T}_1$ and

$$\alpha_2 = \alpha_1 + \delta\alpha_{12}$$

$$\beta_2 = \beta_1 + \delta\beta_{12}$$

Since the rigid body rotation is small,

$$\cos\delta\beta_{12} = 1.0 \tag{B.3}$$

$$\cos\delta\alpha_{12} = 1.0 \tag{B.4}$$

The deformations $\Delta\bar{x}, \Delta\bar{y}, \Delta\bar{z}$, shown in figures B.3 and B.4 are further related to α_1 and the angles of rotation $\delta\alpha_{12}$, $\delta\beta_{12}$ as follows:

$$\sin\delta\alpha_{12} = \frac{\Delta\bar{x}\sin\alpha_1 + \Delta\bar{y}\cos\alpha_1}{l} \tag{B.5}$$

$$\sin\delta\beta_{12} = \frac{\Delta\bar{z}}{l} \tag{B.6}$$

The deformations are also related to the displacements of joints p and q by:

$$\Delta U_{12} = \begin{bmatrix} \Delta u \\ \Delta v \\ \Delta w \end{bmatrix} \tag{B.7}$$

$$\begin{bmatrix} u_q - u_p \\ v_q - v_p \\ w_q - w_p \end{bmatrix}_{12} = \begin{bmatrix} -\Delta\bar{x} \\ \Delta\bar{y} \\ \Delta\bar{z} \end{bmatrix}_{12} \tag{B.8}$$

The changes in member force to the applied load vector $\bar{P}$ at joint p can be related by statics to be:

$$\bar{P}_{12} + (\bar{T}_2 - \bar{T}_1) = 0 \tag{B.9}$$

Substituting equations (B.1) to (B.6) into equation (B.9) the following expression results:

$$\bar{P}_{12} = \begin{bmatrix} \bar{P}_{\hat{x}} \\ \bar{P}_{\hat{y}} \\ \bar{P}_{\hat{z}} \end{bmatrix} = -\frac{T}{l} \begin{bmatrix} \sin^2\alpha_1 & -\sin\alpha_1\cos\alpha_1 & 0 \\ -\sin\alpha_1\cos\alpha_1 & \cos^2\alpha_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta v \\ \Delta w \end{bmatrix} \tag{B.10}$$

This can be related to the direction cosines of the element as:

$$\bar{P}_{12} = -\frac{T}{l}\left[I - \bar{D}\bar{D}^T\right]\Delta\bar{u}_{12} \tag{B.11}$$

where

$I = $ *Identity matrix of order* 3

$\bar{D}^T = [l \ \bar{m} \ \bar{n}]$

$= [\cos\alpha \ \sin\alpha \ 0]$ *the direction cosines* for *the* $\bar{X}$-$\bar{Y}$ *plane*

This can be transformed to a global reference plane (X,Y,Z) with the help of an orthogonal transformation matrix given by:

$$[T_R] = \begin{bmatrix} l^2 & lm & ln \\ lm & m^2 & mn \\ ln & mn & n^2 \end{bmatrix} \tag{B.12}$$

where l, m, n are the direction cosines of the local axes to X, Y, and Z respectively.

Therefore

$$\bar{P}_{12} = [T_R]P_{12} \tag{B.13}$$

and

$$\Delta\bar{U}_{12} = [T_R]\Delta U_{12} \tag{B.14}$$

Substitution of equations (B.13) and (B.14) into equation yields:

$$\bar{P}_{12} = -\frac{T}{l}\left[I - [D][D]^T\right]\Delta U_{12} \tag{B.15}$$

The contributions of each element to the structural stiffness matrix is then written as:

$$\begin{bmatrix} P_p \\ P_q \end{bmatrix} = \begin{bmatrix} k & -k \\ -k & k \end{bmatrix}\begin{bmatrix} U_p \\ U_q \end{bmatrix} \tag{B.16}$$

where

$k = k_T + k_G$

$$k_T = \frac{AE}{l_o}[D][D]^T \text{ from classical linear theory}$$

$$k_G = \frac{T}{l}[I - [D][D]^T]$$
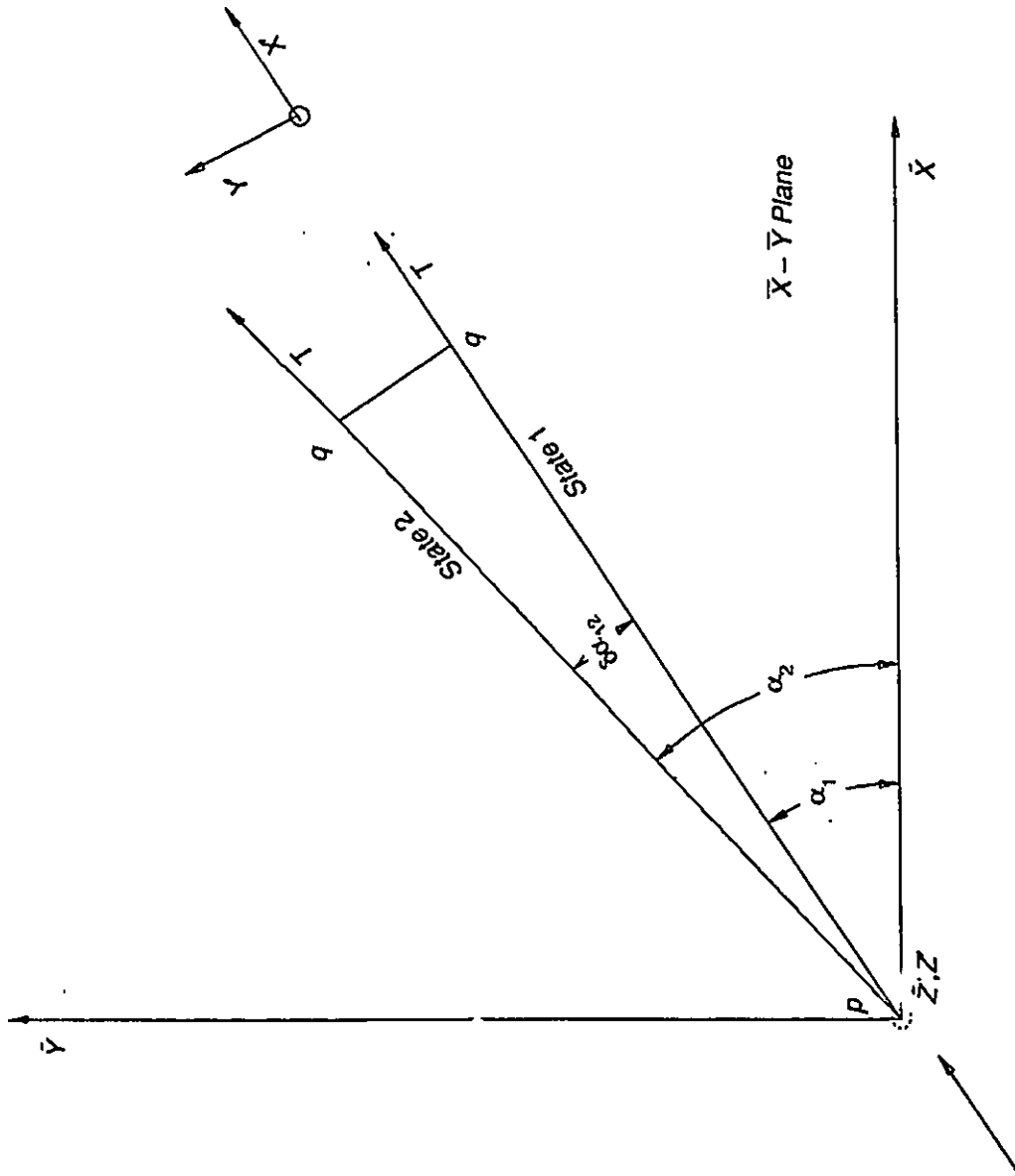
FIG.B.1

STATICAL DETAILS FOR ROTATION IN $\bar{x} - \bar{y}$ PLANE
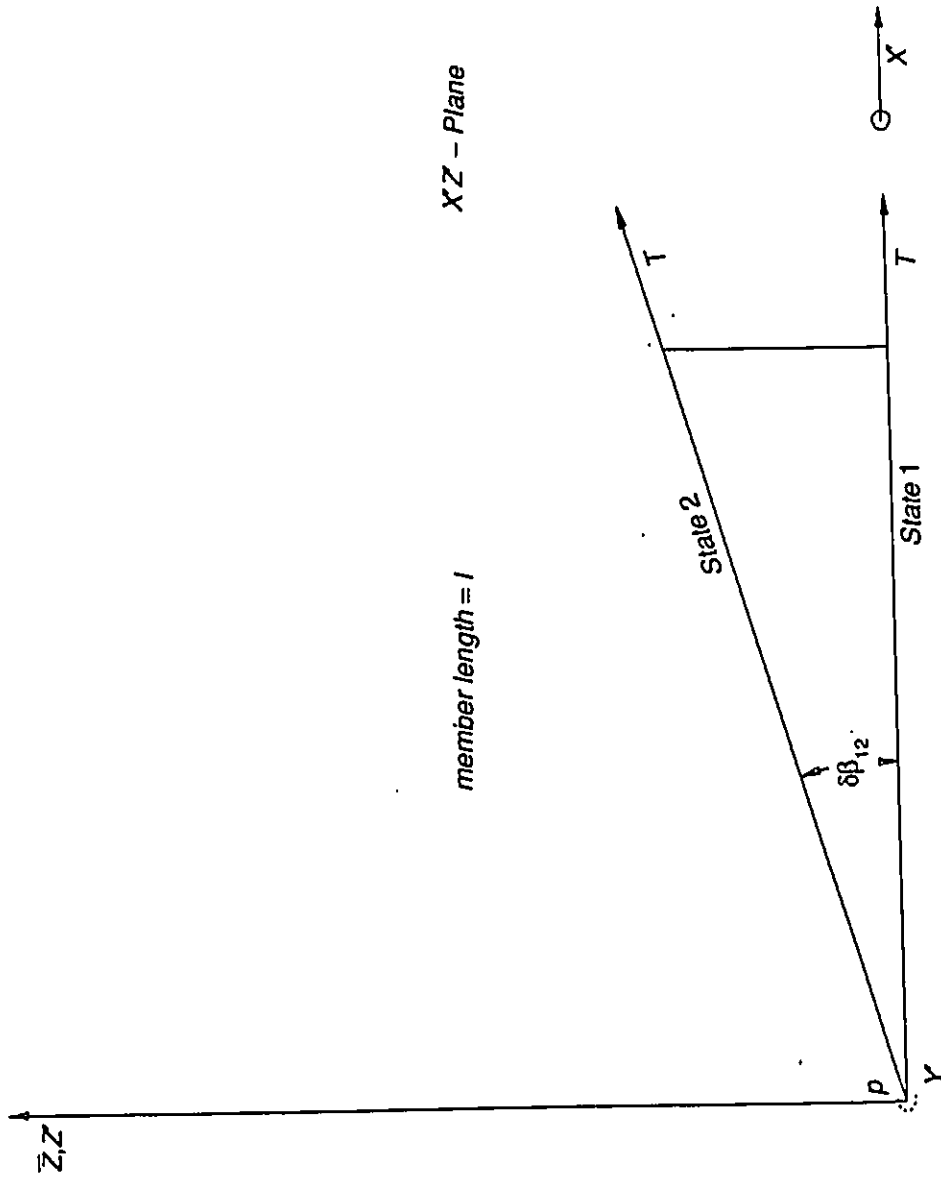
FIG. B.2

STATICAL DETAILS FOR ROTATION IN $x-z$ PLANE

FIG.B.3

GEOMETRIC DETAILS FOR ROTATION IN $\bar{x} - \bar{y}$ PLANE

132

member length = l

XZ – Plane

State 2

State 1

δβ₁₂

Δz̄

p

q

x

Y

Z̄, z
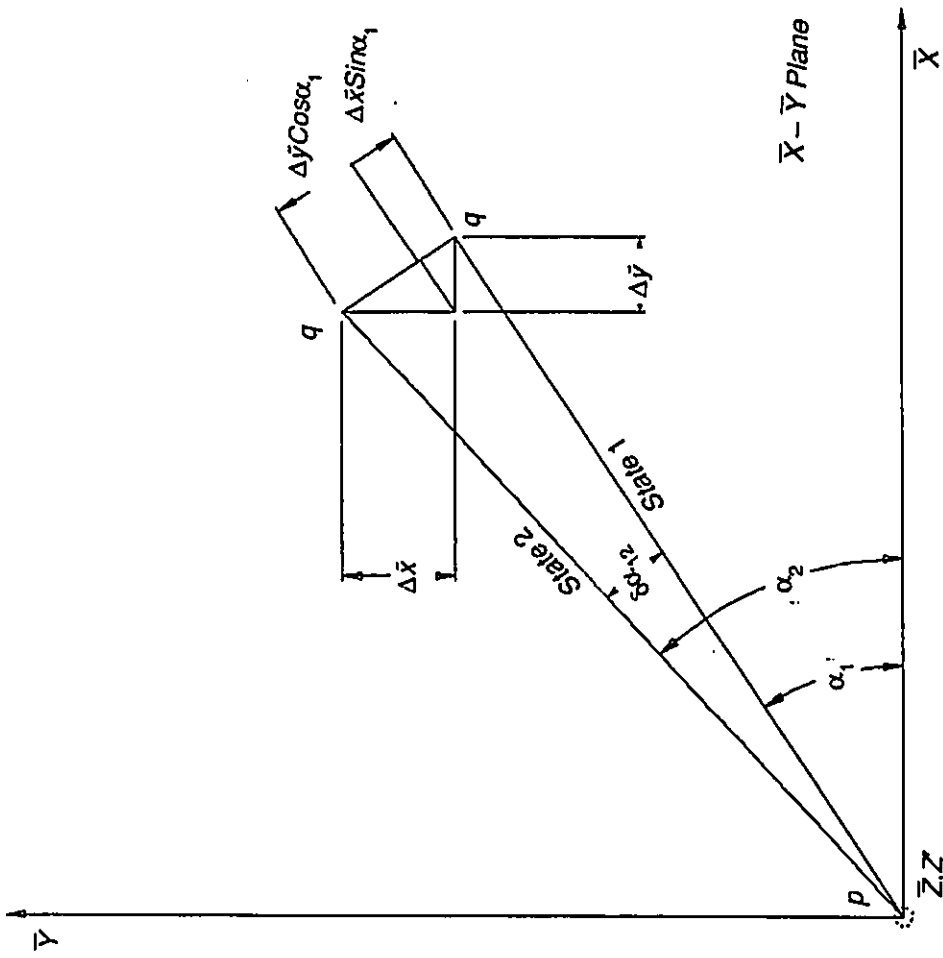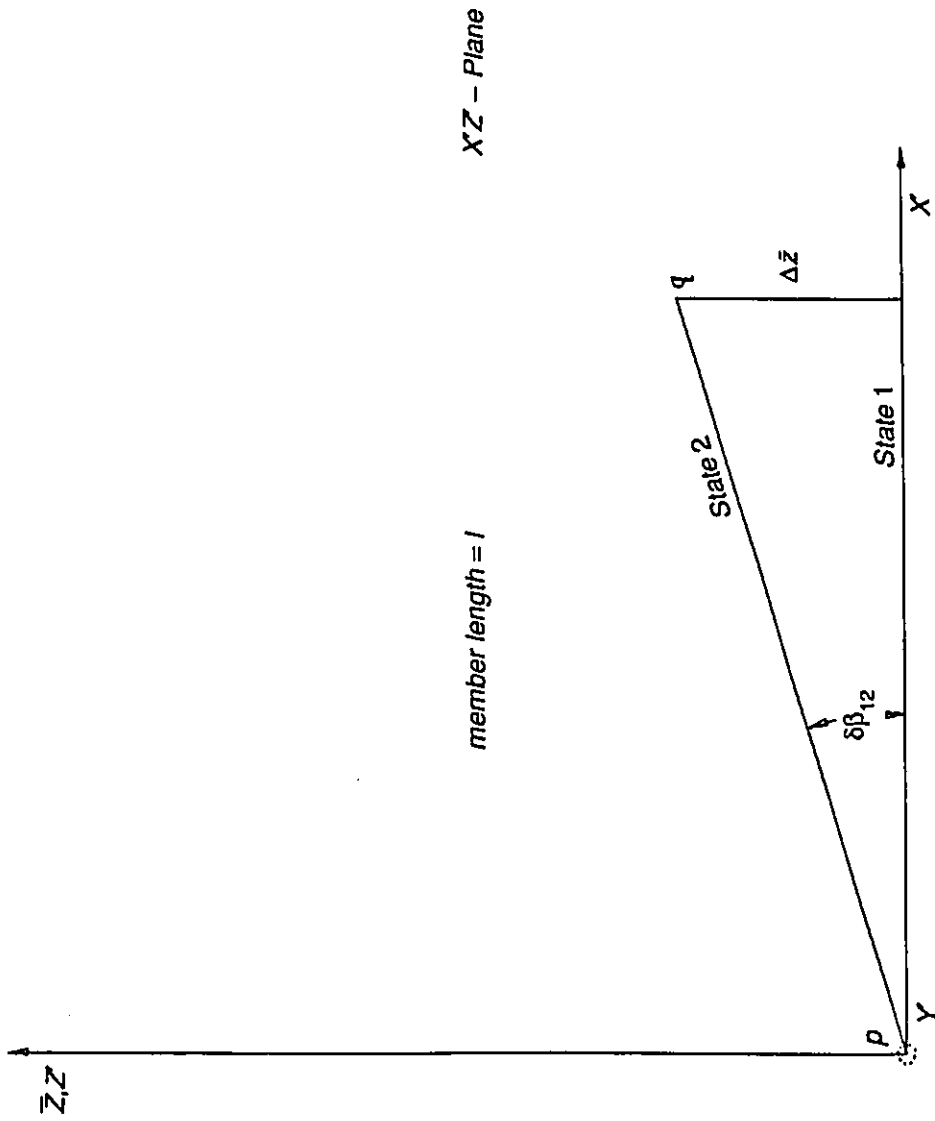
FIG. B.4

GEOMETRIC DETAILS FOR ROTATION IN $x - z$ PLANE

Appendix C

COMPLETE NONLINEAR STIFFNESS MATRIX

The axial element of a structure displaced from state 1 to state 2 is shown in figure 2.1. The displacements u_p and u_q of ends p and q respectively are considered to be large and the strain ε of the element is considered small for Hooke's Law to apply. Therefore

$$\varepsilon_{12} = \frac{l_2 - l_1}{l_1} = \frac{\delta l_{12}}{l_1} \tag{C.1}$$

$$\Delta U_{12} = S - L$$

$$\begin{bmatrix} u_q - u_p \\ v_q - v_p \\ w_q - w_p \end{bmatrix}_{12} = \begin{bmatrix} \Delta u \\ \Delta v \\ \Delta w \end{bmatrix}_{12} \tag{C.2}$$

Then,

$$(1 + \varepsilon_{12})^2 = 1 + \frac{2}{l_1}[D]^T \Delta U_{12} + [D]^T[D] \tag{C.3}$$

$$= (1 + \Psi) \tag{C.4}$$

where $\Psi = \frac{2}{l_1}[D]^T[\Delta U]_{12} + [D]^T[D]$.

The incremental equation of statics at joint p is:

$$\Delta P_{12} + \frac{(T_1 + \delta T_{12})}{l_1(1 + \varepsilon_{12})}[L_1 + \Delta U_{12} - \frac{T_1}{l_1}L_1] = 0 \tag{C.5}$$

where

.

$$\Delta P_{12} = P_2 - P_1$$

$$\delta T_{12} = T_2 - T_1$$

$$L_1^T = [(X_q - X_p) \ (Y_q - Y_p) \ (Z_q - Z_p)]^T$$

Using Hooke's Law:

$$\delta T_{12} = \frac{AE}{l_o} \delta l_{12} \tag{C.6}$$

$$= \frac{AE}{l_o} \varepsilon_{12} l_1 \tag{C.7}$$

where l_o is the unrestrained length of the member.

The manipulations of equation (C.1) and (C.2) yield the following relations:

$$\frac{1}{1 + \varepsilon_{12}} = (1 + \Psi)^{-\frac{1}{2}} \tag{C.8}$$

$$\frac{\varepsilon_{12}}{1 + \varepsilon_{12}} = 1 - (1 + \Psi)^{-\frac{1}{2}} \tag{C.9}$$

$$\frac{1}{2} \Psi \{ L_1 + \Delta U_{12} \} = \frac{1}{l_1^2} L_1 L_1^T \Delta U_{12} + \frac{1}{2l_1^2} L_1 \Delta U_{12}^T \Delta U_{12} + \frac{1}{2} \Psi[I] \Delta U_{12} \tag{C.10}$$

Substituting equation (C.7) into equation (C.5) and using equations (C.8), (C.9) and (C.10):

$$\Delta P_{12} = k \Delta U_{12} \tag{C.11}$$

$$= \{ k_L + k_{NL} \} \Delta U_{12} \tag{C.12}$$

where

$$k_L = \frac{AE}{l_o} [D][D]^T + \frac{T_1}{l_1} [I - [D][D]^T] \tag{C.13}$$

$$k_{NL} = \left[\frac{AE}{l_o} - \frac{T_1}{l_1} \right] \left[C_1[I] + C_2[D][D]^T + C_3 L_1 \Delta U_{12}^T \right] \tag{C.14}$$

$$C_1 = \frac{\varepsilon_{12}}{1 + \varepsilon_{12}} \tag{C.15}$$

$$C_2 = -\frac{\varepsilon_{12}(3 + \varepsilon_{12})}{(1 + \varepsilon_{12})(2 + \varepsilon_{12})} \tag{C.16}$$

$$C_3 = \frac{1}{(1 + \varepsilon_{12})(2 + \varepsilon_{12})} \tag{C.17}$$

Appendix D

COMPUTER PROGRAM FOR THE TANGENT STIFFNESS METHOD

```
C... This is a program for the Tangent Stiffness Method
C... for the nonlinear analysis of cable net
C... structures. The following are the variables used
C... in this program.
C...
C... A(NEL,2)        = Nodal connectivity matrix
C... AREA(NEL)       = Area of the element
C... B(NEL)          = Tension in the member during an
C...                   iteration
C... COOR(NN,3)      = Coordinates at each node
C... CORA(NN,3)      = Updated coordinates at the node
C... COUNT           = Number of iterations
C... DCOS(NEL,3)     = Direction cosine of the element
C... DELU(NDF)       = Sum of displacements after every
C...                   iteration
C... DOF(NN,3)       = Degree of freedom at each node
C... E(NEL)          = Elastic modulus of the element
C... GAMMA(NEL)      = Strain in the element
C... ID(3,3)         = Identity matrix of order 3
C... KT(NDF,NDF)     = Tangent stiffness matrix
C... LEN(NEL)        = Length of the element
C... LENO(NEL)       = Unstiffened length of the
C...                   element
C... LIMIT           = Specified accuracy for
C...                   convergence
C... NDF             = Number of degrees of freedom
C... NEL             = Number of elements.
C... NN              = Number of nodes
C... PEQ(NDF)        = Equilibrium nodal load
C... PNODE(NDF)      = Applied nodal load
C... PREM(NDF)       = Residual load at the nodes
C... T(NEL)          = Pretension in the element
C... TNODE(NDF)      = Total equivalent nodal load
C... U(NDF)          = Displacements at the degrees of
C...                   freedom
C... VCT(NEL,6)      = Variable correlation matrix
C...
      PARAMETER (NEL = 180,NN = 117,NDF = 243)
      COMMON/ONE/CORA
      COMMON/TWO/VCT
      COMMON/THREE/A
      COMMON/FOUR/DOF
      COMMON/WORKSP/RWKSP
C...
      REAL ALEN,AREA(NEL),B(NEL),COOR(NN,3),CORA(NN,3),
     +DCOS(NEL,3),DELU(NDF),DIFJ(3,3),E(NEL),ELK(3,3),
     +ELM(6,6),GAMMA(NEL),ID(3,3),INTER(3,3),KT(NDF,NDF),
     +LEN(NEL),LENO(NEL),LIMIT,PEQ(NDF),PNODE(NDF),PREM(NDF),
     +RNODE(NDF),RWKSP(59557),SIFJ(3,3),T(NEL),TNODE(NDF),
     +U(NDF)
```

```
C...
        INTEGER A(NEL,2),COUNT,DOF(NN,3),END1,END2,
       +VCT(NEL,6)
C...
C...    Read input data
C...
        READ *,((A(I,J),J = 1,2),I=1,NEL)
        READ *,((DOF(I,J),J = 1,3),I = 1,NN)
        READ *,((COOR(I,J),J = 1,3),I = 1,NN)
        READ *,(T(I),  I = 1,NEL)
        READ *,(PNODE(I),I = 1,NDF)
        READ *,(AREA(I),I = 1,NEL)
        READ *,(E(I),  I = 1,NEL)
C...
C...    Increase work storage for the LSARG routine
C...
        CALL IWKIN(59557)
C...
C...    Print structural information
C...
        PRINT 30,'NODAL CONNECTIVITY TABLE',
       +'MEMBER','END P','END Q'
        DO 1 I = 1,NEL
        PRINT 31,I,A(I,1),A(I,2)
1       CONTINUE
        PRINT 32,'DEGREE OF FREEDOM','NODE','X','Y','Z'
        DO 2 I = 1,NN
        PRINT 33,I,DOF(I,1),DOF(I,2),DOF(I,3)
2       CONTINUE
C...
C...    Form and print variable correlation table
C...
        PRINT 34,'VARIABLE CORRELATION TABLE','ELEMENT',
       +'END P','END Q','I
        DO 4 I = 1,NEL
365     CONTINUE
        DO 3  J = 1,6
        M = 2
        Q = J - 3
        IF (J.LE.3) THEN
        M = 1
        Q = J
        END IF
        PAN = A(I,M)
        VCT(I,J)  = DOF(PAN,Q)
3       CONTINUE
        PRINT 35,I,(VCT(I,J),J = 1,6)
4       CONTINUE
        PRINT 36,'NODAL COORDINATES','NODE','X','Y','Z'
        DO 5  I = 1,NN
```

```
            PRINT 37,I,COOR(I,1),COOR(I,2),COOR(I,3)
5           CONTINUE
            PRINT 38,'SECTIONAL PROPERTIES OF MEMBERS',
           +'MEMBER','AREA','MODULUS','TENSION'
            DO 6  I = 1,NEL
            PRINT 39,I,AREA(I),E(I),T(I)
6           CONTINUE
            DO 7 I = 1,NN
            DO 7 J = 1,3
            CORA(I,J) = COOR(I,J)
7           CONTINUE
            PRINT 40,'LOAD CONDITION','DOF','LOAD'
            DO 8 I = 1,NDF
            PRINT 41,I,PNODE(I)
C...
C...  Initialize load and displacement vectors
C...
            PREM(I) = PNODE(I)
            TNODE(I) = PNODE(I)
            U(I) = 0
            DELU(I) = 0
8           CONTINUE
C...
C...  To calculate the direction cosines of the members
C...  and the total equivalent nodal loads
C...
            DO 10 I = 1,NEL
            END1 = A(I,1)
            END2 = A(I,2)
            CALL DIRCOS(I,U,LEN,DCOS)
            DO 9   J = 1,3
            OR = DOF(END2,J)
            OS = DOF(END1,J)
            IF(OR.NE.0) THEN
            TNODE(OR) = TNODE(OR) + T(I)*DCOS(I,J)
            END IF
            IF (OS.NE.0) THEN
            TNODE(OS) = TNODE(OS) - T(I)*DCOS(I,J)
            END IF
9           CONTINUE
C...
C...  Calculate unstiffened length of the member
C...
            LENO(I) = LEN(I)/(T(I)/(AREA(I)*E(I)) + 1.)
C...
C...  Initialize member strains
C...
            GAMMA(I) = 0
10          CONTINUE
C...
```

```
C...    Form identity matrix
C...
        DO 11 L = 1,3
        DO 11 M = 1,3
        ID(L,M) = 0
        IF(L.EQ.M) THEN
        ID(L,M) = 1.
        END IF
11      CONTINUE
        COUNT = 0
12      COUNT = COUNT + 1
C...
C...    Initialize stiffness matrix
C...
        DO 13 I = 1,NDF
        RNODE(I) = PREM(I)
        DO 13 J = 1,NDF
        KT(I,J) = 0
13      CONTINUE
C...
C...    Form a new stiffness matrix
C...
        DO 19 I = 1,NEL
        DO 14 L = 1,3
   .    DO 14 M = 1,3
        DIFJ(L,M) = DCOS(I,L)*DCOS(I,M)
14      CONTINUE
        CALL MATADD (ID,DIFJ,2,INTER,3)
        DO 15 L = 1,3
        DO 15 M = 1,3
        INTER(L,M) = INTER(L,M)*T(I)/LEN(I)
        SIFJ(L,M) = DIFJ(L,M)*AREA(I)*E(I)/LENO(I)
15      CONTINUE
        CALL MATADD(SIFJ,INTER,1,ELK,3)
        DO 17 L = 1,3
        DO 16 M = 1,3
        ELM(L,M) = ELK(L,M)
        IR = M + 3
        ELM(L,IR) = - ELK(L,M)
16      CONTINUE
        LOR = L + 3
        DO 17 J = 1,6
        ELM(LOR,J) = - ELM(L,J)
17      CONTINUE
        DO 18 J = 1,6
        KORA = VCT(I,J)
        DO 18 M = 1,6
        LORA = VCT(I,M)
        IF((KORA.EQ.0).OR.(LORA.EQ.0)) GOTO 19
        KT(KORA,LORA) = KT(KORA,LORA) + ELM(J,M)
```

```
18      CONTINUE
19      CONTINUE
        DO 20 I = 1,NDF
        RNODE(I) = RNODE(I)*1000.
        DO 20 J = 1,NDF
        KT(I,J) = KT(I,J)*1000.
20      CONTINUE
C...
C...    Solve for the displacements
C...
        CALL LSARG(NDF,KT,NDF,RNODE,1,U)
        CALL ORDY(U,CORA)
        DO 21 J = 1,NDF
        DELU(J) = DELU(J) + U(J)
        RNODE(J) = RNODE(J)/1000.
        PEQ(J) = 0
21      CONTINUE
C...
C...    Update structure geometry
C...
        DO 24 I = 1,NEL
        ALEN = LEN(I)
        CALL DIRCOS(I,U,LEN,DCOS)
C...
C...    Calculate strain in the member
C...
        GAMMA(I) = (LEN(I) - ALEN)/ALEN
        B(I) = 0
        END1 = A(I,1)
        END2 = A(I,2)
        DO 22 J = 1,3
        CALL CALCUL(I,J,U,U2,U1)
        DIF = U2 - U1
        B(I) = B(I) + DCOS(I,J)*DIF*E(I)*AREA(I)/LEN(I)
22      CONTINUE
C...
C...    Calculate member tension and nodal equilibrium
C...    load
C...
        T(I) = T(I) + B(I)
        DO 23  J = 1,3
        OR = DOF(END2,J)
        OS = DOF(END1,J)
        IF(OR.NE.0) THEN
        PEQ(OR) = PEQ(OR) - T(I)*DCOS(I,J)
        END IF
        IF(OS.NE.0) THEN
        PEQ(OS) = PEQ(OS) + T(I)*DCOS(I,J)
        END IF
23      CONTINUE
```

```
24      CONTINUE
C...
C...    Calculate residual nodal loads
C...
        DO 25 L = 1,NDF
        PREM(L) = PEQ(L) + TNODE(L)
25      CONTINUE
C...
C...    Check for convergence
C...
        DO 26 L = 1,NDF
        LIMIT = 1.E-4
        IF(ABS(PREM(L)).GT. ABS(LIMIT)) GOTO 12
26      CONTINUE
27      PRINT 42,'FINAL NODAL DISPLACEMENTS','NODE','X',
       +'Y','Z'
        PRINT *, 'NO OF ITERATIONS = ',COUNT
        DO 28 J = 1,NN
        COLE = DOF(J,1)
        UX = 0
        DOLE = DOF(J,2)
        UY = 0
        MOLE = DOF(J,3)
        UZ = 0
        IF (COLE.NE.0) THEN
        UX = DELU(COLE)
        END IF
        IF (DOLE.NE.0) THEN
        UY = DELU(DOLE)
        END IF
        IF (MOLE.NE.0) THEN
        UZ = DELU(MOLE)
        END IF
        PRINT 43,J,UX,UY,UZ
28      CONTINUE
        PRINT 44,'FINAL MEMBER FORCES','MEMBER','FORCE'
        DO 29 I = 1,NEL
        PRINT 45,I,T(I)
29      CONTINUE
30      FORMAT (15X,A24//15X,A6,2X,A5,2X,A5//)
31      FORMAT (17X,I2,6X,I2,5X,I2)
32      FORMAT (15X,A17//15X,A4,3X,A1,3X,A1,3X,A1//)
33      FORMAT (16X,I3,2X,I3,1X,I3,1X,I3)
34      FORMAT (///10X,A26///10X,A7,5X,A5,8X,A5///14X,
       +A1,5X,2('U',3X,'V',3X,'W',3X)/20X,3('P',3X),
       +3('Q',3X)//)
35      FORMAT (13X,I2,4X,I2,5(2X,I2))
36      FORMAT (//15X,A17///7X,A4,8X,A1,9X,A1,9X,A1//)
37      FORMAT (7X,I3,4X,F8.1,2X,F8.1,2X,F8.1)
38      FORMAT (//15X,A24//13X,A6,3X,A4,4X,A7,3X,A7//)
```

```fortran
39      FORMAT (14X,I3,4X,F8.2,2X,F8.3,2X,F9.3)
40      FORMAT (///15X,A14//15X,A3,5X,A4//)
41      FORMAT (15X,I3,3X,F8.3)
42      FORMAT (15X,A24//18X,A4,6X,A1,10X,A1,10X,A1)
43      FORMAT (16X,I3,2X,F14.7,2X,F14.7,2X,F14.7)
44      FORMAT (15X,A19///15X,A6,6X,A6//)
45      FORMAT (17X,I3,5X,F14.7)
        STOP
        END
C...
C...    Subroutine DIRCOS calculates the lengths and
C...    direction cosines of each element
C...
        SUBROUTINE DIRCOS(I,U,LEN,DCOS)
        PARAMETER (NEL = 180,NDF = 243,NN = 117)
        COMMON/ONE/CORA
        COMMON/THREE/A
        REAL U(NDF),LEN(NEL),DCOS(NEL,3),CORA(NN,3)
        INTEGER A(NEL,2)
        END1 = A(I,1)
        END2 = A(I,2)
        LEN(I) = 0
        DO 101 J = 1,3
        CALL CALCUL(I,J,U,U2,U1)
   .    DIF = U2 - U1
        X2 = CORA(END2,J)
        X1 = CORA(END1,J)
        LEN(I) = LEN(I) + (X2 - X1)**2
        DCOS(I,J) = X2 - X1
101     CONTINUE
        LEN(I) = SQRT(LEN(I))
        DO 102 J = 1,3
        DCOS(I,J) = DCOS(I,J)/LEN(I)
102     CONTINUE
        RETURN
        END
C...
C...    Subroutine MATADD is used to add or subtract two
C...    matrices
C...    Mode = 1 indicates addition
C...    Mode = 2 indicates subtraction
C...
        SUBROUTINE MATADD(MAT1,MAT2,MODE,MAT3,IRW)
        REAL MAT1(IRW,IRW),MAT2(IRW,IRW),MAT3(IRW,IRW)
        DO 201 L = 1,IRW
        DO 201 M = 1,IRW
        MAT3(L,M) = MAT1(L,M) - MAT2(L,M)
        IF(MODE.EQ.1) THEN
        MAT3(L,M) = MAT1(L,M) + MAT2(L,M)
        END IF
```

```
201    CONTINUE
       RETURN
       END
C...
C...   Subroutine CALCUL is used to relate nodal
C...   deflections to corresponding degree of freedom
C...
       SUBROUTINE CALCUL(I,J,U,U2,U1)
       PARAMETER(NDF = 243,NEL = 180)
       COMMON/TWO/VCT
       REAL U2,U1,U(NDF)
       INTEGER VCT(NEL,6)
       IR = J + 3
       MO = VCT(I,J)
       NO = VCT(I,IR)
       U1 = 0
       U2 = 0
       IF(MO.NE.0) THEN
       U1 = U(MO)
       END IF
       IF(NO.NE.0) THEN
       U2 = U(NO)
       END IF
       RETURN
       END
C...
C...   Subroutine ORDY updates the coordinates of the
C...   nodes after every iteration
C...
       SUBROUTINE ORDY (U,CORA)
       PARAMETER (NEL = 180,NDF = 243, NN = 117)
       COMMON/FOUR/DOF
       REAL CORA(NN,3),U(NDF)
       INTEGER DOF(NN,3)
       DO 401 N = 1,NN
       DO 401 J = 1,3
       LO = DOF(N,J)
       IF(LO.NE.0) THEN
       CORA(N,J) = CORA(N,J) + U(LO)
       END IF
401    CONTINUE
       RETURN
       END
```

Appendix E

COMPUTER PROGRAM FOR THE SECANT STIFFNESS METHOD

```
C...    This is a program for the Secant Stiffness Method
C...    for the nonlinear analysis of cable net
C...    structures. The following are the variables used
C...    in this program.
C...    A(NEL,2)            = Nodal connectivity matrix
C...    AREA(NEL)           = Area of the element
C...    B(NEL)              = Tension in the member at each
C...                          iteration
C...    COOR(NN,3)          = Coordinates of the nodes
C...    CORA(NN,3)          = Updated coordinates of the
C...                          nodes
C...    COUNT               = Number of iterations
C...    DCOS(NEL,3)         = Direction cosines of the
C...                          element
C...    DELU(NDF)           = Sum of displacements due to
C...                          each iteration
C...    E(NEL)              = Elastic modulus of the
element
C...    GAMMA(NEL)          = Strain in the element
C...    ID(3,3)             = Identity matrix of order 3
C...    KL(NDF,NDF)         = Tangent stiffness matrix
C...    KNL(NDF,NDF)        = Secant Stiffness matrix
C...    K(NDF,NDF)          = Stiffness matrix of the
C...                          structure
C...    LEN(NEL)            = Length of the element
C...    LENO(NEL)           = Unstiffened length of the
C...                          element
C...    LIMIT               = Specified accuracy for
C...                          convergence
C...    NDF                 = Number of degrees of freedom
C...    NEL                 = Number of elements.
C...    NN                  = Number of nodes
C...    PEQ(NDF)            = Equlibrium nodal load
C...    PNODE(NDF)          = Applied nodal load
C...    PRES(NDF)           = Residual load at the node
C...    TNODE(NDF)          = Total equivalent nodal load
C...    U(NDF)              = Displacement at the node for
C...                          each iteration
C...    VCT(NEL,6)          = Variable correlation matrix
C...
C...

        PARAMETER (NEL = 180,NN = 117, NDF = 243)
        COMMON/ONE/CORA
        COMMON/TWO/VCT
        COMMON/THREE/A
        COMMON/FOUR/DOF
        COMMON/EXPT/LAC
        COMMON/WORKSP/RWKSP
C...
        REAL AID(3,3),ALEN,AREA(NEL),B(NEL),COOR(NN,3),
```

```
      +CORA(NN,3),DCOS(NEL,3),DELU(NDF),DIFJ(3,3),E(NEL),
      +ELK(3,3),ELM(6,6),ENLK(3,3),ENLM(6,6),ENTAR(3,3),
      +GAMMA(NEL),ID(3,3),INTER(3,3),K(NDF,NDF),KL(NDF,NDF),
      +KNL(NDF,NDF),LEN(NEL),LENO(NEL),LIMIT,PEQ(NDF),
      +PNODE(NDF),PREM(NDF),RNODE(NDF),RWKSP(59557),
      +SIFJ(3,3),T(NEL),TEMP(3,3),TIFJ(3,3),TNODE(NDF),
      +U(NDF)
C...
      INTEGER A(NEL,2),COUNT,DOF(NN,3),END1,END2,
      +VCT(NEL,6)
C...
C...  Read input data
C...
      READ *,((A(I,J),J = 1,2),I=1,NEL)
      READ *,((DOF(I,J),J = 1,3),I = 1,NN)
      READ *,((COOR(I,J),J = 1,3),I = 1,NN)
      READ *,(T(I),  I = 1,NEL)
      READ *,(PNODE(I),I = 1,NDF)
      READ *,(AREA(I),I = 1,NEL)
      READ *,(E(I),  I = 1,NEL)
C...
C...  Increase work storage for LSARG routine
C...
      CALL IWKIN(59557)
C...
C...  Print structural information
C...
      PRINT 33,'NODAL CONNECTIVITY TABLE','MEMBER',
      +'END P','END Q'
      DO 1 I = 1,NEL
      PRINT 34,I,A(I,1),A(I,2)
1     CONTINUE
      PRINT 35,'DEGREE OF FREEDOM','NODE','X','Y','Z'
      DO 2 I = 1,NN
      PRINT 36,I,DOF(I,1),DOF(I,2),DOF(I,3)
2     CONTINUE
C...
C...  Form variable correlation table and print
C...
      PRINT 37,'VARIABLE CORRELATION TABLE','ELEMENT',
      +'END P','END Q','I
      DO 4 I = 1,NEL
      DO 3 J = 1,6
      M = 2
      Q = J - 3
      IF (J.LE.3) THEN
      M = 1
      Q = J
      END IF
      PAN = A(I,M)
```

```
         VCT(I,J) = DOF(PAN,Q)
3        CONTINUE
         PRINT 38,I,(VCT(I,J),J = 1,6)
4        CONTINUE
         PRINT 39,'SECTIONAL PROPERTIES OF MEMBERS',
        +'MEMBER','AREA','MODULUS','TENSION'
         DO 5 I = 1,NEL
         PRINT 40,I,AREA(I),E(I),T(I)
5        CONTINUE
         PRINT 41,'NODAL COORDINATES','NODE','X','Y','Z'
         DO 6 I = 1,NN
         PRINT 42,I,COOR(I,1),COOR(I,2),COOR(I,3)
6        CONTINUE
         DO 7 I = 1,NN
         DO 7 J = 1,3
         CORA(I,J) = COOR(I,J)
7        CONTINUE
         PRINT 43,'LOAD CONDITION','DOF','LOAD'
         DO 8 I = 1,NDF
         PRINT 44,I,PNODE(I)
C...
C...    Initialize load and displacement vectors
C...
         TNODE(I) = PNODE(I)
    .    PREM(I) = PNODE(I)
         DELU(I) = 0
         U(I) = 0
8        CONTINUE
C...
C...    To calculate the direction cosines of the members
C...
         DO 10 I = 1,NEL
         T(I) = TR(I)
         END1 = A(I,1)
         END2 = A(I,2)
         CALL DIRCOS(I,U,LEN,DCOS)
         DO 9 J = 1,3
         OR = DOF(END2,J)
         OS = DOF(END1,J)
C...
C...    Calculate total equivalent nodal loads
C...
         IF(OR.NE.0) THEN
         TNODE(OR) = TNODE(OR) + T(I)*DCOS(I,J)
         END IF
         IF(OS.NE.0) THEN
         TNODE(CS) = TNODE(OS) - T(I)*DCOS(I,J)
         END IF
9        CONTINUE
C...
```

```
C...    Calculate the unstiffened length of the element
C...
        LENO(I) = LEN(I)/(T(I)/(AREA(I)*E(I)) + 1.)
C...
C...    Initialize the member strains
C...
        GAMMA(I) = 0
10      CONTINUE
C...
C...    Form the identity matrix
C...
        DO 11 L = 1,3
        DO 11 M = 1,3
        ID(L,M) = 0
        IF(L.EQ.M) THEN
        ID(L,M) = 1.
        END IF
11      CONTINUE
        COUNT = 0
12      COUNT = COUNT + 1
C...
C...    Initialize the stiffness matrix
C...
        DO 13 L = 1,NDF
  .     RNODE(L) = PREM(L)
        DO 13 M = 1,NDF
        KL(L,M) = 0
        KNL(L,M) = 0
13      CONTINUE
C...
C...    Form a new stiffness matrix
C...
        DO 22 I = 1,NEL
        DO 14 L = 1,3
        DO 14 M = 1,3
        TEMP(L,M) = DCOS(I,L)
14      CONTINUE
        DO 15 J = 1,3
        CALL CALCUL(I,J,U,U2,U1)
        DO 15 L = 1,3
        TEMP(L,J) = TEMP(L,J)*(U2 - U1)
15      CONTINUE
        DO 16 L = 1,3
        DO 16 M =1,3
        DIFJ(L,M) = DCOS(I,L)*DCOS(I,M)
16      CONTINUE
        CALL MATADD(ID,DIFJ,2,INTER,3)
        DO 17 L = 1,3
        DO 17 M = 1,3
        INTER(L,M) = INTER(L,M)*T(I)/LEN(I)
```

```
          SIFJ(L,M)  = DIFJ(L,M)*AREA(I)*E(I)/LENO(I)
          TEMP(L,M)  = TEMP(L,M)/(LEN(I)*(1. + GAMMA(I))
        +*(2. + GAMMA(I)))
          TIFJ(L,M)  = DIFJ(L,M)*GAMMA(I)*(3.+GAMMA(I))
          TIFJ(L,M)  = TIFJ(L,M)/((1. + GAMMA(I))
        +*(2. + GAMMA(I)))
          AID(L,M)  = ID(L,M)*GAMMA(I)/(1. + GAMMA(I))
17        CONTINUE
          CALL MATADD(SIFJ,INTER,1,ELK,3)
          CALL MATADD(AID,TIFJ,2,ENTAR,3)
          CALL MATADD(ENTAR,TEMP,1,ENLK,3)
          DO 18 L = 1,3
          DO 18 M = 1,3
          ENLK(L,M)  = ENLK(L,M)*(AREA(I)*E(I)/LENO(I) -
        +T(I)/LEN(I))
18        CONTINUE
          DO 20 L = 1,3
          DO 19 M = 1,3
          ELM(L,M)  = ELK(L,M)
          ENLM(L,M)  = ENLK(L,M)
          IR = M + 3
          ELM(L,IR)  = - ELK(L,M)
          ENLM(L,IR)  = - ENLK(L,M)
19        CONTINUE
  `       LOR = L + 3
          DO 20 J = 1,6
          ELM(LOR,J)  = - ELM(L,J)
          ENLM(LOR,J)  = - ENLM(L,J)
20        CONTINUE
          DO 21  J = 1,6
          KORA = VCT(I,J)
          DO 21 M = 1,6
          LORA  = VCT(I,M)
          IF((KORA.EQ.0).OR.(LORA.EQ.0)) GOTO 40
          KL(KORA,LORA)  = KL(KORA,LORA) + ELM(J,M)
          KNL(KORA,LORA)  = KNL(KORA,LORA) + ENLM(J,M)
21        CONTINUE
22        CONTINUE
          CALL MATADD(KL,KNL,1,K,NDF)
          DO 23 I = 1,NDF
          RNODE(I)  = RNODE(I)*1000.
          DO 23 J = 1,NDF
          K(I,J)  = K(I,J)*1000.
23        CONTINUE
C...
C...  Solve for the displacements
C...
          CALL LSARG(NDF,K,NDF,RNODE,1,U)
C...
C...  Update the structure geometry
```

```
C...
      CALL ORDY(U,CORA)
      DO 24 J = 1,NDF
      RNODE(J) = RNODE(J)/1000.
      PEQ(J) = 0
24    CONTINUE
      DO 27 I = 1,NEL
      ALEN = LEN(I)
      CALL DIRCOS(I,U,LEN,DCOS)
C...
C...  Calculate new member strains
C...
      GAMMA(I) = (LEN(I) - ALEN)/ALEN
      B(I) = 0
      END1 = A(I,1)
      END2 = A(I,2)
      DO 25 J = 1,3
      CALL CALCUL(I,J,U,U2,U1)
      DIF = U2 - U1
      B(I) = B(I) + DCOS(I,J)*DIF*E(I)*AREA(I)/LEN(I)
25    CONTINUE
C...
C...  Calculate new member tensions
C...
   .  T(I) = T(I) + B(I)
      DO 26 J = 1,3
      OR = DOF(END2,J)
      OS = DOF(END1,J)
C...
C...  Calculate nodal equilibrium loads
C...
      IF(OR.NE.0) THEN
      PEQ(OR) = PEQ(OR) - T(-)*DCOS(I,J)
      END IF
      IF(OS.NE.0) THEN
      PEQ(OS) = PEQ(OS) + T(I)*DCOS(I,J)
      END IF
26    CONTINUE
27    CONTINUE
C...
C...  Calculate residual loads at the nodes
C...
      DO 28 L = 1,NDF
      DELU(L) = DELU(L) + U(L)
      PREM(L) =    (PEQ(L) + TNODE(L))
28    CONTINUE
C...
C...  Check for convergence
C...
      DO 29 L = 1,NDF
```

```
        LIMIT = 1.E-4
        IF(ABS(PREM(L)).GT. ABS(LIMIT)) GOTO 33
29      CONTINUE
C...
C...    Print final diaplacements and member forces
C...
30      PRINT 45,'FINAL NODAL DISPLACEMENTS','NODE','X',
       +'Y','Z'
        PRINT *, 'NO OF ITERATIONS = ',COUNT
        DO 31 J = 1,NN
        COLE = DOF(J,1)
        UX = 0
        DOLE = DOF(J,2)
        UY = 0
        MOLE = DOF(J,3)
        UZ = 0
        IF (COLE.NE.0) THEN
        UX = DELU(COLE)
        END IF
        IF (DOLE.NE.0) THEN
        UY = DELU(DOLE)
        END IF
        IF (MOLE.NE.0) THEN
        UZ = DELU(MOLE)
   `    END IF
        PRINT 46,J,UX,UY,UZ
31      CONTINUE
        PRINT 47,'FINAL MEMBER FORCES','MEMBER','FORCE'
        DO 32 I = 1,NEL
        PRINT 48,I,T(I)
32      CONTINUE
33      FORMAT (15X,A24//15X,A6,2X,A5,2X,A5//)
34      FORMAT (17X,I2,6X,I2,5X,I2)
35      FORMAT (15X,A17//15X,A4,3X,A1,3X,A1,3X,A1//)
36      FORMAT (16X,I3,2X,I3,1X,I3,1X,I3)
37      FORMAT (///10X,A26///10X,A7,5X,A5,8X,A5///14X,
       +A1,5X,2('U',3X,'V',3X,'W',3X)/20X,3('P',3X),
       +3('Q',3X)//)
38      FORMAT (13X,I2,4X,I2,5(2X,I2))
39      FORMAT (//15X,A24//13X,A6,3X,A4,4X,A7,3X,A7//)
40      FORMAT (14X,I3,4X,F8.2,2X,F8.3,2X,F9.3)
41      FORMAT (//15X,A17///7X,A4,8X,A1,9X,A1,9X,A1//)
42      FORMAT (7X,I3,4X,F8.1,2X,F8.1,2X,F8.1)
43      FORMAT (///15X,A14//15X,A3,5X,A4//)
44      FORMAT (15X,I3,3X,F8.3)
45      FORMAT (15X,A24//18X,A4,6X,A1,10X,A1,10X,A1)
46      FORMAT (16X,I3,2X,F14.7,2X,F14.7,2X,F14.7)
47      FORMAT (15X,A19///15X,A6,6X,A6//)
48      FORMAT (17X,I3,5X,F14.7)
        STOP
```

```
          END
C...
C...    Subroutine DIRCOS calculates direction cosines
C...    and lengths of the elements
C...
          SUBROUTINE DIRCOS(I,U,LEN,DCOS)
          PARAMETER (NEL = 180,NDF = 243,NN = 117)
          COMMON/ONE/CORA
          COMMON/THREE/A
          COMMON/EXPT/LAC
          REAL U(NDF),LEN(NEL),DCOS(NEL,3),CORA(NN,3)
          INTEGER A(NEL,2)
          END1 = A(I,1)
          END2 = A(I,2)
          LEN(I) = 0
          DO 101 J = 1,3
          CALL CALCUL(I,J,U,U2,U1)
          DIF = U2 - U1
          X2 = CORA(END2,J)
          X1 = CORA(END1,J)
          LEN(I) = LEN(I) + (X2 - X1)**2
          DCOS(I,J) = X2 - X1
101       CONTINUE
          LEN(I) = SQRT(LEN(I))
   .      DO 102 J = 1,3
          DCOS(I,J) = DCOS(I,J)/LEN(I)
102       CONTINUE
          RETURN
          END
C...
C...    Subroutine MATADD adds or subtracts two
C...    matrices
C...    Mode = 1 indicates addition
C...    Mode = 2 indicates subtraction
C...
          SUBROUTINE MATADD(MAT1,MAT2,MODE,MAT3,IRW)
          REAL MAT1(IRW,IRW),MAT2(IRW,IRW),MAT3(IRW,IRW)
          DO 201 L = 1,IRW
          DO 201 M = 1,IRW
          MAT3(L,M) = MAT1(L,M) - MAT2(L,M)
          IF(MODE.EQ.1) THEN
          MAT3(L,M) = MAT1(L,M) + MAT2(L,M)
          END IF
201       CONTINUE
          RETURN
          END
C...
C...    Subroutine CALCUL relates nodal deflections to
C...    its corresponding degrees of freedom
C...
```

```
      SUBROUTINE CALCUL(I,J,U,U2,U1)
      PARAMETER(NDF = 243,NEL = 180)
      COMMON/TWO/VCT
      COMMON/EXPT/LAC
      REAL U2,U1,U(NDF)
      INTEGER VCT(NEL,6)
      IR = J + 3
      MO = VCT(I,J)
      NO = VCT(I,IR)
      U1 = 0
      U2 = 0
      IF(MO.NE.0) THEN
      U1 = U(MO)
      END IF
      IF(NO.NE.0) THEN
      U2 = U(NO)
      END IF
      RETURN
      END
C...
C...  Subroutine ORDY updates the nodal coordinates
C...  after every iteration
C...
      SUBROUTINE ORDY (U,CORA)
      PARAMETER (NEL = 180,NDF = 243, NN = 117)
      COMMON/FOUR/DOF
      REAL CORA(NN,3),U(NDF)
      INTEGER DOF(NN,3)
      DO 101 N = 1,NN
      DO 101 J = 1,3
      LO = DOF(N,J)
      IF(LO.NE.0) THEN
      CORA(N,J) = CORA(N,J) + U(LO)
      END IF
101   CONTINUE
      RETURN
      END
```

Appendix F

COMPUTER PROGRAM FOR KAR'S MODIFICATION METHOD

```
C...    This is a program for Kar's modification method
C...    for the nonlinear analysis of cables networks.
C...    The following are the variables used in this
C...    program
C...    AREA(NEL)          = Area of the element
C...    A(NEL,2)           = Nodal connectivity matrix
C...    B(NEL)             = Tension in the element due to
C...                         displacements
C...    COOR(NN,3)         = Coordinates at each node
C...    CORA(NN,3)         = Updated coordinates at each
node
C...    COUNT              = Number of iterations
C...    DCOS(NEL,3)        = Direction cosines of each
C...                         element
C...    DELU(NDF)          = Summation of displacements in
C...                         every iteration
C...    DOF(NN,3)          = Degree of freedom at each node
C...    E(NEL)             = Elastic modulus in each element
C...    GAMMA(NEL)         = Strain in each element
C...    ID(3,3)            = Identity matrix of order 3
C...    KT(NDF,NDF)        = Tangent stiffness matrix
C...    LEN(NEL)           = Length of the element
C...    LENO(NEL)          = Unstiffened length of each
C...                         element
C...    LIMIT              = Specified accuracy
C...    NDF                = Number of degrees of freedom
C...    NEL                = Number of elements.
C...    NN                 = Number of nodes
C...    PEQ(NDF)           = Equilibrium nodal load
C...    PNODE(NEL)         = Applied nodal load
C...    PREM(NDF)          = Residual load at each node
after
C...                         every iteration
C...    T(NEL)             = Pretension in the element
C...    TNODE(NEL)         = Total equivalent nodal load
C...    U(NDF)             = Displacements at the degrees of
C...                         freedom in each iteration
C...    VCT(NEL,6)         = Variable correlation matrix
C...
        PARAMETER (NEL = 180,NN = 117, NDF = 243)
        COMMON/TWO/VCT
        COMMON/THREE/A
        COMMON/FOUR/DOF
        COMMON/WORKSP/RWKSP
C...
        REAL ALEN,AREA(NEL),B(NEL),COOR(NN,3),CORA(NN,3),
       +DCOS(NEL,3),DELU(NDF),DIFJ(3,3),E(NEL),ELK(3,3),
       +ELM(3,3),GAMMA(NEL),GLEN,ID(3,3),INTER(3,3),
       +K(NDF,NDF),KT(NDF,NDF),LEN(NEL),LENO(NEL),LIMIT,
       +LIN(NEL),MORA(NN,3),PEQ(NDF),PNODE(NDF),PREM(NDF),
```

```
      +U(NDF),RNODE(NDF),RWKSP(59557),SIFJ(3,3),T(NEL),
      +TNODE(NDF),TR(NEL)
C...
       INTEGER A(NEL,2),COUNT,DOF(NN,3),END1,END2,
      +VCT(NEL,6)
C...
C...  Read input data
C...
       READ *,((A(I,J),J = 1,2),I=1,NEL)
       READ *,((DOF(I,J),J = 1,3),I = 1,NN)
       READ *,((COOR(I,J),J = 1,3),I = 1,NN)
       READ *,(T(I),  I = 1,NEL)
       READ *,(PNODE(I),I = 1,NDF)
       READ *,(AREA(I),I = 1,NEL)
       READ *,(E(I),  I = 1,NEL)
C...
C...  Increase temporary work storage for LSARG routine
C...
       CALL IWKIN(59557)
C...
C...  Print out the structural data
C...
       PRINT 36,'NODAL CONNECTIVITY TABLE','MEMBER',
      +'END P','END Q'
       DO 1 I = 1,NEL
       PRINT 37,I,A(I,1),A(I,2)
1      CONTINUE
       PRINT 38,'DEGREE OF FREEDOM','NODE','X','Y','Z'
       DO 2 I = 1,NN
       PRINT 39,I,DOF(I,1),DOF(I,2),DOF(I,3)
2      CONTINUE
C...
C...  Form the variable correlation table
C...
       PRINT 40,'VARIABLE CORRELATION TABLE','ELEMENT',
      +'END P','END Q','I
       DO 4 I = 1,NEL
       DO 3  J = 1,6
       M = 2
       Q = J - 3
       IF (J.LE.3) THEN
       M = 1
       Q = J
       END IF
       PAN = A(I,M)
       VCT(I,J)  = DOF(PAN,Q)
3      CONTINUE
       PRINT 41,I,(VCT(I,J),J = 1,6)
4      CONTINUE
       PRINT 42,'NODAL COORDINATES','NODE','X','Y','Z'
```

```
          DO 5  I = 1,NN
          PRINT 43,I,COOR(I,1),COOR(I,2),COOR(I,3)
5         CONTINUE
          PRINT 44,'SECTIONAL PROPERTIES OF MEMBERS',
         +'MEMBER','AREA','MODULUS','TENSION'
          DO 6  I = 1,NEL
          PRINT 45,I,AREA(I),E(I),T(I)
6         CONTINUE
          DO 7  I = 1,NN
          DO 7  J = 1,3
          CORA(I,J) = COOR(I,J)
7         CONTINUE
          PRINT 46,'LOAD CONDITION','DOF','LOAD'
          DO 8 I = 1,NDF
          PRINT 47,I,PNODE(I)
C...
C...      Initialize the load and displacement vectors
C...
          TNODE(I) = PNODE(I)
          PREM(I) = PNODE(I)
          U(I) = 0
          DELU(I) = 0
8         CONTINUE
C...
C...      To calculate the direction cosines of the members
C...      and the total equivalent nodal loads
C...
          DO 10 I = 1,NEL
          END1 = A(I,1)
          END2 = A(I,2)
          CALL DIRCOS(I,U,CORA,LEN,DCOS)
          DO 9 J = 1,3
          OR = DOF(END2,J)
          OS = DOF(END1,J)
          IF(OR.NE.0) THEN
          TNODE(OR) = TNODE(OR) + T(I)*DCOS(I,J)
          END IF
          IF(OS.NE.0) THEN
          TNODE(OS) = TNODE(OS) - T(I)*DCOS(I,J)
          END IF
9         CONTINUE
C...
C...      Calculate the unstressed length of the member
C...
          LENO(I) = LEN(I)/(T(I)/(AREA(I)*E(I)) + 1.)
C...
C...      Initialize strain to zero
C...
          GAMMA(I) = 0
10        CONTINUE
```

```
C...
C...    Form the identity matrix
C...
        DO 11 L = 1,3
        DO 11 M = 1,3
        ID(L,M) = 0
        IF(L.EQ.M) THEN
        ID(L,M) = 1.
        END IF
11      CONTINUE
        COUNT = 0
12      COUNT = COUNT + 1
C...
C...    Initialize the stiffness matrix
C...
        DO 13 I = 1,NDF
        RNODE(I) = PREM(I)
        DO 13 J = 1,NDF
        KT(I,J) = 0
13      CONTINUE
C...
C...    Form a new stiffness matrix
C...
        DO 19 I = 1,NEL
        DO 14 L = 1,3
        DO 14 M = 1,3
        DIFJ(L,M) = DCOS(I,L)*DCOS(I,M)
14      CONTINUE
        CALL MATADD (ID,DIFJ,2,INTER,3)
        DO 15 L = 1,3
        DO 15 M = 1,3
        INTER(L,M) = INTER(L,M)*T(I)/LEN(I)
        SIFJ(L,M) = DIFJ(L,M)*AREA(I)*E(I)/LENO(I)
15      CONTINUE
        CALL MATADD(SIFJ,INTER,1,ELK,3)
        DO 17 L = 1,3
        DO 16 M = 1,3
        ELM(L,M) = ELK(L,M)
        IR = M + 3
        ELM(L,IR) = - ELK(L,M)
16      CONTINUE
        LOR = L + 3
        DO 17 J = 1,6
        ELM(LOR,J) = - ELM(L,J)
17      CONTINUE
        DO 18 J = 1,6
        KORA = VCT(I,J)
        DO 18 M = 1,6
        LORA = VCT(I,M)
        IF((KORA.EQ.0).OR.(LORA.EQ.0)) GOTO 19
```

```
            KT(KORA,LORA)  =  KT(KORA,LORA)  +  ELM(J,M)
18          CONTINUE
19          CONTINUE
            DO 20 I = 1,NDF
            RNODE(I)  =  RNODE(I)*1000.
            DO 20 J = 1,NDF
            KT(I,J)  =  KT(I,J)*1000.
20          CONTINUE
C...
C...        Solve for the displacements
C...
            CALL  LSARG(NDF,KT,NDF,RNODE,1,U)
C...
C...        Update the geometry
C...
            DO 21 I = 1,NN
            DO 21 J = 1,3
            MORA(I,J)  =  CORA(I,J)
21          CONTINUE
            CALL  ORDY(U,MORA)
            DO 22 J = 1,NDF
            RNODE(J)  =  RNODE(J)/1000.
            PEQ(J)  =  0
22          CONTINUE
            DO 25 I = 1,NEL
            TR(I)  =  T(I)
            GLEN  =  LEN(I)
            CALL  DIRCOS(I,U,MORA,LIN,DCOS)
C...
C...        Calculate new strains
C...
            GAMMA(I)  =  (LIN(I) - GLEN)/GLEN
            B(I)  =  0
            END1  =  A(I,1)
            END2  =  A(I,2)
            DO 23 J = 1,3
            CALL  CALCUL(I,J,U,U2,U1)
            DIF  =  U2 - U1
            B(I)  =  B(I)  +  DCOS(I,J)*DIF*E(I)*AREA(I)/LIN(I)
23          CONTINUE
C...
C...        Calculate new tension and equilibrium loads
C...
            TR(I)  =  TR(I)  +  B(I)
            DO 24 J = 1,3
            OR  =  DOF(END2,J)
            OS  =  DOF(END1,J)
            IF(OR.NE.0)  THEN
            PEQ(OR)  =  PEQ(OR)  -  B(I)*DCOS(I,J)
            END IF
```

```
        IF(OS.NE.0) THEN
        PEQ(OS) = PEQ(OS) + B(I)*DCOS(I,J)
        END IF
24      CONTINUE
25      CONTINUE
C...
C...    Calculate the ratio of the greatest applied load
C...    to its corresponding equilibrium load
C...
        GREAT = RNODE(1)
        RATIO = GREAT/(-PEQ(1))
        DO 26 I = 2,NDF
        IF(RNODE(I) .GT. GREAT) THEN
        GREAT = RNODE(I)
        RATIO = GREAT/(-PEQ(I))
        END IF
26      CONTINUE
C...
C...    Modify the displacements
C...
        DO 27 I = 1,NDF
        U(I) = U(I)*RATIO
27      CONTINUE
        CALL ORDY(U,CORA)
        DO 28 J = 1,NDF
        RNODE(J) · RNODE(J)/1000.
        PEQ(J) = 0
28      CONTINUE
C...
C...    Reupdate the geometry
C...
        DO 31 I = 1,NEL
        ALEN = LEN(I)
        CALL DIRCOS(I,U,CORA,LEN,DCOS)
C...
C...    Recalculate the strains
C...
        GAMMA(I) = (LEN(I) - ALEN)/ALEN
        B(I) = 0
        END1 = A(I,1)
        END2 = A(I,2)
        DO 29 J = 1,3
        CALL CALCUL(I,J,U,U2,U1)
        DIF = U2 - U1
        B(I) = B(I) + DCOS(I,J)*DIF*E(I)*AREA(I)/LEN(I)
29      CONTINUE
C...
C...    Recalculate tension and equilibrium loads
C...
        T(I) = T(I) + B(I)
```

```
         DO 30 J = 1,3
         OR = DOF(END2,J)
         OS = DOF(END1,J)
         IF(OR.NE.0) THEN
         PEQ(OR) = PEQ(OR) - T(I)*DCOS(I,J)
         END IF
         IF(OS.NE.0) THEN
         PEQ(OS) = PEQ(OS) + T(I)*DCOS(I,J)
         END IF
30       CONTINUE
31       CONTINUE
C...
C...     Calculate total displacement and the residual
C...     load
C...
         DO 32 L = 1,NDF
         DELU(L) = DELU(L) + U(L)
         PREM(L) = PEQ(L) + TNODE(L)
32       CONTINUE
         DO 33 L = 1,NDF
         LIMIT = 1.E-3
         IF(ABS(PREM(L)).GT. ABS(LIMIT)) GOTO 12
33       CONTINUE
C...
C...     Print out the final displacements and member
C...     forces
C...
         PRINT 48,'FINAL NODAL DISPLACEMENTS','NODE','X',
        +'Y','Z'
         PRINT *, 'NO OF ITERATIONS = ',COUNT
         DO 34 J = 1,NN
         COLE = DOF(J,1)
         UX = 0
         DOLE = DOF(J,2)
         UY = 0
         MOLE = DOF(J,3)
         UZ = 0
         IF (COLE.NE.0) THEN
         UX = DELU(COLE)
         END IF
         IF (DOLE.NE.0) THEN
         UY = DELU(DOLE)
         END IF
         IF (MOLE.NE.0) THEN
         UZ = DELU(MOLE)
         END IF
         PRINT 49,J,UX,UY,UZ
34       CONTINUE
         PRINT 50,'FINAL MEMBER FORCES','MEMBER','FORCE'
         DO 35 I = 1,NEL
```

```
          PRINT 51,I,T(I)
35        CONTINUE
36        FORMAT (15X,A24//15X,A6,2X,A5,2X,A5//)
37        FORMAT (17X,I2,6X,I2,5X,I2)
38        FORMAT (15X,A17//15X,A4,3X,A1,3X,A1,3X,A1//)
39        FORMAT (16X,I3,2X,I3,1X,I3,1X,I3)
40        FORMAT (///10X,A26///10X,A7,5X,A5,8X,A5///14X,
         +A1,5X,2('U',3X,'V',3X,'W',3X)/20X,3('P',3X),
         +3('Q',3X)//)
41        FORMAT (13X,I2,4X,I2,5(2X,I2))
42        FORMAT (//15X,A17///7X,A4,8X,A1,9X,A1,9X,A1//)
43        FORMAT (7X,I3,4X,F8.1,2X,F8.1,2X,F8.1)
44        FORMAT (//15X,A24//13X,A6,3X,A4,4X,A7,3X,A7//)
45        FORMAT (14X,I3,4X,F8.2,2X,F8.3,2X,F9.3)
46        FORMAT (///15X,A14//15X,A3,5X,A4//)
47        FORMAT (15X,I3,3X,F8.3)
48        FORMAT (15X,A24//18X,A4,6X,A1,10X,A1,10X,A1)
49        FORMAT (16X,I3,2X,F14.7,2X,F14.7,2X,F14.7)
50        FORMAT (15X,A19///15X,A6,6X,A6//)
51        FORMAT (17X,I3,5X,F14.7)
          STOP
          END
C...
C...
C...      Subroutine DIRCOS is used to calculate the
C...      lengths and direction cosines of each element
C...
          SUBROUTINE DIRCOS(I,U,CORA,LEN,DCOS)
          PARAMETER (NEL = 180,NDF = 243,NN = 117)
          COMMON/THREE/A
          REAL U(NDF),LEN(NEL),DCOS(NEL,3),CORA(NN,3)
          INTEGER A(NEL,2)
          END1 = A(I,1)
          END2 = A(I,2)
          LEN(I) = 0
          DO 101 J = 1,3
          CALL CALCUL(I,J,U,U2,U1)
          DIF = U2 - U1
          X2 = CORA(END2,J)
          X1 = CORA(END1,J)
          LEN(I) = LEN(I) + (X2 - X1)**2
          DCOS(I,J) = X2 - X1
101       CONTINUE
          LEN(I) = SQRT(LEN(I))
          DO 102 J = 1,3
          DCOS(I,J) = DCOS(I,J)/LEN(I)
102       CONTINUE
          RETURN
          END
C...
```

```
C...
C...   Subroutine MATADD is used in the addition or
C...   subtraction of two matrices. Mode = 1 indicates
C...   addition. Mode = 2 indicates subtraction
C...
       SUBROUTINE MATADD(MAT1,MAT2,MODE,MAT3,IRW)
       REAL MAT1(IRW,IRW),MAT2(IRW,IRW),MAT3(IRW,IRW)
       DO 201 L = 1,IRW
       DO 201 M = 1,IRW
       MAT3(L,M) = MAT1(L,M) - MAT2(L,M)
       IF(MODE.EQ.1) THEN
       MAT3(L,M) = MAT1(L,M) + MAT2(L,M)
       END IF
201    CONTINUE
       RETURN
       END
C...
C...
C...   Subroutine CALCUL is used to relate the nodal
C...   degree of freedom to its deflection
C...
       SUBROUTINE CALCUL(I,J,U,U2,U1)
       PARAMETER(NDF = 243,NEL = 180)
       COMMON/TWO/VCT
       REAL U2,U1,U(NDF)
       INTEGER VCT(NEL,6)
       IR = J + 3
       MO = VCT(I,J)
       NO = VCT(I,IR)
       U1 = 0
       U2 = 0
       IF(MO.NE.0) THEN
       U1 = U(MO)
       END IF
       IF(NO.NE.0) THEN
       U2 = U(NO)
       END IF
       RETURN
       END
C...
C...   Subroutine ORDY updates the nodal coordinates
C...   of the structure after every iteration
C...
       SUBROUTINE ORDY (U,CORA)
       PARAMETER (NEL = 180,NDF = 243, NN = 117)
       COMMON/FOUR/DOF
       REAL CORA(NN,3),U(NDF)
       INTEGER DOF(NN,3)
       DO 401 N = 1,NN
       DO 401 J = 1,3
```

```
      LO = DOF(N,J)
      IF(LO.NE.0) THEN
      CORA(N,J) = CORA(N,J) + U(LO)
      END IF
401   CONTINUE
      RETURN
      END
```

Appendix G

COMPUTER PROGRAM FOR THE ENERGY SEARCH METHOD

:

```
C...   This is a program for the Energy Search Method
C...   to obtain a minimum for the energy function using
C...   the Conjugate Gradient Method along with a
scaling
C...   a scaling transformation. The following are the
C...   variables used in this program.
C...   A(NEL,2)              = Nodal connectivity matrix
C...   AREA(NEL)             = Area of each element
C...   COOR(NN,3)            = Coordinates of the nodes
C...   CORA(NN,3)            = Updated coordinates of the
C...                           nodes
C...   DCOS(NEL,3)           = Direction cosines of each
C...                           element
C...   DOF(NN,3)             = Degree of freedom matrix
C...   E(NEL)                = Elastic modulus of element
C...   FORCE(NEL)            = Final tension in the member
C...   K(NDF,NDF)            = Stiffness matrix of the
C...                           structure
C...   KOUNT                 = Number of iterations
C...   LEN(NEL)              = Length of the member
C...   NDF                   = Number of degrees of freedom
C...   NEL                   = Number of elements
C...   NN                    = Number of nodes
C...   P(NDF)                = Applied nodal load
C...   R(NDF)                = Scaling factor
C...   S(NEL)                = Length of the member after
C...                           deformataion
C...   T(NEL)                = Initial tension in the member
C...   X(NDF)                = Displacements at the nodes
C...   VCT(NEL,6)            = Variable correlation matrix
C...
C...

       PARAMETER (NEL = 180,NDF = 243,NN = 117,
      +LIMIT = 1000)
C...

       DOUBLE PRECISION AREA(NEL),C,COOR(NN,3),
      +CORA(NN,3),DCOS(NEL,3),E(NEL),ELK(NEL,6),
      +EEP,F,FO,FORCE(NEL),G(NDF),GE,H(486),K(NDF,NDF),
      +LEN(NEL),P(NDF),R(NDF),RIG,S(NEL),T(NEL),UX,UY,
      +UZ,X(NDF),X1,X2,Y1,Y2,Z1,Z2
C...

       INTEGER A(NEL,2),COLE,DOF(NN,3),DOLE,ROLE,
      +VCT(NEL,6),
C...

       COMMON/ONE/LEN,E,AREA,T,R,P,FORCE
       COMMON/TWO/A,COOR,VCT,CORA
       COMMON/THREE/EEP,GE,FO,C
       COMMON/FOUR/KOUNT
C...

       EXTERNAL FUNC
```

```
C...
      EPS = 1.D-8
      EST = -100.D00
      EEP = .0105103
      C = 2321951.D0
      FO = 1506.D0
      GE =  2568285.D0
C...
C... Read input data
C...
      READ *,((A(I,J),J = 1,2),I=1,NEL)
      READ *,((DOF(I,J),J = 1,3),I = 1,NN)
      READ *,((COOR(I,J),J = 1,3),I = 1,NN)
      READ *,(T(I),  I = 1,NEL)
      READ *,(P(I),I = 1,NDF)
      READ *,(AREA(I),I = 1,NEL)
      READ *,(E(I),  I = 1,NEL)
C...
C... Print structural information
C...
      PRINT 19,'NODAL CONNECTIVITY TABLE','MEMBER',
     +'END P','END Q'
      DO 1 I = 1,NEL
      PRINT 20,I,A(I,1),A(I,2)
1     CONTINUE
      PRINT 21,'DEGREE OF FREEDOM','NODE','X','Y','Z'
      DO 2 I = 1,NN
      PRINT 22,I,DOF(I,1),DOF(I,2),DOF(I,3)
2     CONTINUE
C...
C... Form and print variable correlation table
C...
      PRINT 23,'VARIABLE CORRELATION TABLE','ELEMENT',
     +'END P','END Q','I
      DO 4 I = 1,NEL
      DO 3  J = 1,6
      M = 2
      L = J - 3
      IF (J.LE.3) THEN
      M = 1
      L = J
      END IF
      MAN = A(I,M)
      VCT(I,J) = DOF(MAN,L)
3     CONTINUE
      PRINT 24,I,(VCT(I,J),J = 1,6)
4     CONTINUE
      PRINT 25,'NODAL COORDINATES','NODE','X','Y','Z'
      DO 6 I = 1,NN
      DO 5 J = 1,3
```

```fortran
          CORA(I,J) = COOR(I,J)
5         CONTINUE
          PRINT 26,I,COOR(I,1),COOR(I,2),COOR(I,3)
6         CONTINUE
          PRINT 27,'SECTIONAL PROPERTIES OF MEMBERS',
         +'MEMBER','AREA','MODULUS','TENSION'
          DO 7 I = 1,NEL
          PRINT 28,I,AREA(I),E(I),T(I)
7         CONTINUE
C...
C...   To calculate the lengths of the members and to
C...   use the scaling transformations
C...
          DO 8  I = 1,NDF
          X(I) = 0
8         CONTINUE
          DO 12 I = 1,NEL
          END1 = A(I,1)
          END2 = A(I,2)
          X1 = COOR(END1,1)
          X2 = COOR(END2,1)
          Y1 = COOR(END1,2)
          Y2 = COOR(END2,2)
          Z1 = CCOR(END1,3)
          Z2 = COOR(END2,3)
          LEN(I) = DSQRT((X2 - X1)**2 + (Y2 - Y1)**2 +
         +(Z2 - Z1)**2)
          DCOS(I,1) = (X2 - X1)/LEN(I)
          DCOS(I,2) = (Y2 - Y1)/LEN(I)
          DCOS(I,3) = (Z2 - Z1)/LEN(I)
          RIG = AREA(I)*E(I)/LEN(I)
          DO 10 J = 1,3
          DO 9 M = 1,3
          ELK(J,M) = DCOS(I,J)*DCOS(I,M)*RIG
          IM = M + 3
          ELK(J,IM) = - ELK(J,M)
9         CONTINUE
          IJ = J + 3
          DO 10 L = 1,6
          ELK(IJ,L) = - ELK(J,L)
10        CONTINUE
          DO 11 J = 1,6
          KORA = VCT(I,J)
          DO 11 M = 1,6
          LORA = VCT(I,M)
          IF((KORA.NE.0) .AND.(LORA.NE.0)) THEN
          K(KORA,LORA) = K(KORA,LORA) + ELK(J,M)
          END IF
11        CONTINUE
12        CONTINUE
```

```
C...
C...   Form the scaling factor matrix
C...
       DO 13 M = 1,NDF
       R(M) = 1.D0
       IF(K(M,M).NE.0) THEN
       R(M) = 1.D0/DSQRT(K(M,M))
       END IF
13     CONTINUE
       N = NDF
       PRINT 29,'LOAD CONDITION','DOF','LOAD'
       DO 14 I = 1,NDF
       PRINT 30,I,P(I)
14     CONTINUE
C...
C...   To call the subroutine of Conjugate Gradients
C...
       CALL DFMCG(FUNC,KOUNT,LIMIT,EPS,EST,IER,F,
      +G,H,N,X)
C...
C...   To print out the final values
C...
       PRINT 31,F,KOUNT,IER
       PRINT 32,'MEMBER','DIMENSIONAL VARIABLES VECTOR',
      +'DIMENSION FUNCTION GRADIENT VECTOR','X(I)',
      +'G(I)'
       DO 15 I = 1,NDF
       X(I) = X(I)*R(I)
       PRINT 33,I,X(I),G(I)
15     CONTINUE
C...
C...   To print the nodal displacements
C...
       PRINT 34,'FINAL NODAL DISPLACEMENTS','NODE','X',
      +'Y','Z'
       DO 16 J = 1,NN
       COLE = DOF(J,1)
       UX = 0
       DOLE = DOF(J,2)
       UY = 0
       ROLE = DOF(J,3)
       UZ = 0
       IF (COLE.NE.0) THEN
       UX = X(COLE)
       END IF
       IF (DOLE.NE.0) THEN
       UY = X(DOLE)
       END IF
       IF (ROLE.NE.0) THEN
       UZ = X(ROLE)
```

```
          END IF
          PRINT 35,J,UX,UY,UZ
16        CONTINUE
          PRINT 34,'FINAL  NODAL  COORDINATES','NODE','X',
      +'Y','Z'
          DO 17 I = 1,NN
          PRINT 35,I,CORA(I,1),CORA(I,2),CORA(I,3)
17        CONTINUE
          PRINT 36,'FINAL MEMBER FORCES','MEMBER','FORCE'
          DO 18 I = 1,NEL
          PRINT 37,I,FORCE(I)
18        CONTINUE
19        FORMAT (15X,A24//15X,A6,2X,A5,2X,A5//)
20        FORMAT (17X,I3,6X,I2,5X,I3)
21        FORMAT (15X,A17//15X,A4,3X,A1,3X,A1,3X,A1//)
22        FORMAT (16X,I3,2X,I3,1X,I3,1X,I3)
23        FORMAT (///10X,A26///10X,A7,5X,A5,8X,A5///14X,
      +A1,5X,2('U',3X,'V',3X,'W',3X)/20X,3('P',3X),
      +3('Q',3X)//)
24        FORMAT (13X,I3,4X,I3,5(2X,I3))
25        FORMAT (//15X,A17///7X,A4,8X,A1,9X,A1,9X,A1//)
26        FORMAT (7X,I3,4X,F8.1,2X,F8.1,2X,F8.1)
27        FORMAT (//15X,A24//13X,A6,3X,A4,4X,A7,3X,A7//)
28        FORMAT (14X,I3,4X,F9.6,2X,F8.1,2X,F9.3)
29        FORMAT (///15X,A14//15X,A3,5X,A4//)
30        FORMAT (15X,I3,3X,F8.3)
31        FORMAT (15X,'POTENTIAL ENERGY = ',E13.5///15X,
      +'NO. OF ITERATIONS = ',I4//15X,'IER = ',I8)
32        FORMAT (10X,A6,6X,A28,30X,A29///40X,A4,53X,A4///)
33        FORMAT (12X,I4,19X,E13.5,44X,E13.5)
34        FORMAT (15X,A24///18X,A4,6X,A1,10X,A1,10X,A1)
35        FORMAT (16X,I3,2X,F9.3,2X,F9.3,2X,F9.3)
36        FORMAT (15X,A19///15X,A6,6X,A6//)
37        FORMAT (17X,I3,5X,F20.11)
          STOP
          END
C...
C...
C...   Subroutine FUNC calculates the value of the
C...   potential energy and its gradient needed in the
C...   subroutine DFMCG
          SUBROUTINE FUNC(N,X,F,G)
          PARAMETER (NEL = 180, NDF = 243)
          DOUBLE PRECISION AREA(NEL),C,D,E(NEL),EEP,F,F1,
      +FO,FORCE(NEL),G(NDF),GE,KAY,LEN(NEL),P(NDF),R(NDF),
      +S(NEL),T(NEL),TERM1,TERM2,TERM3,X(NDF)
          COMMON/ONE/LEN,E,AREA,T,R,P,FORCE
          COMMON/THREE/EEP,GE,FO,C
          COMMON/FOUR/KOUNT
C...
```

```fortran
C...    Initialize the value of potential energy and
C...    modify the displacements
C...
        F = 0.D0
        DO 101 J = 1,NDF
        X(J) = R(J)* X(J)
        G(J) = 0.D0
101     CONTINUE
C...
C...    Calculate the deformed lengths
C...
        DO 105 I = 1,NEL
        S(I) = 0.D0
        DO 102 J = 1,3
        CALL ORDY (I,J,MO,NO,D,X)
        S(I) = S(I) + D**2
102     CONTINUE
        S(I) = DSQRT(S(I))
C...
C...    Calculate the strain in the members
C...
        KAY = 1.D0 - LEN(I)/S(I) + T(I)/(AREA(I)*E(I))
        IF (KAY.LT.0) THEN
        U = 0.D0
        F1 = 0.D0
        FORCE(I) = 0.D0
        GOTO 211
        END IF
C...
C...    Check if the strain is less than the proportional
C...    limit
C...
        IF (KAY.LE.EEP) THEN
C...
C...    Use elastic strain energy equation
C...
        U = S(I)*AREA(I)*E(I)*KAY**2/2.D0
        FORCE(I) = AREA(I)*E(  )*KAY
        F1 = (U + FORCE(I)*I   ,I))/S(I)**2
        GOTO 103
        END IF
C...
C...    Check if the strain is greater than the
C...    proportional limit
C...
        IF (KAY.GT.EEP) THEN
        KAY = KAY + (1.D0 - LEN(I)/S(I))**2/2.D0
        TERM1 = E(I)*EEP**2/2.D0 + FO*(KAY - EEP)
        TERM2 = DSQRT( 0**2 + 2.D0*GE*KAY - C)
        FORCE(I) = AREA(I)*(FO + TERM2)
```

```
      TERM3 = DSQRT(FO**2 + 2.D0*GE*EEP - C)
C...
C...  Use inelastic strain energy equation
C...
      U = AREA(I)*S(I)*(TERM1 + TERM2**3/(3.D0*GE)-
     +TERM3**3/(3.D0*GE))
      F1 = (U + FORCE(I)*LEN(I))/S(I)**2
      END IF
103   F = F + U
      DO 104 J = 1,3
      CALL ORDY (I,J,MO,NO,D,X)
      IF (MO.NE.0) THEN
      G(MO) = G(MO) - F1*D
      END IF
      IF (NO.NE.0) THEN
      G(NO) = G(NO) + F1*D
      END IF
104   CONTINUE
105   CONTINUE
      WORK = 0.D0
      DO 106 I = 1,NDF
      WORK = WORK + P(I)*X(I)
C...
C...  Calculate strain energy gradient
C...
      G(I) = ((G(I) - P(I)))* R(I)
      X(I) = X(I)/R(I)
106   CONTINUE
C...
C...  Calculate the total potential energy of the
C...  structure
C...
      F = F - WORK
      RETURN
      END
C...
C...
C...  Subroutine ORDY is used to relate nodal
C...  deflections to its corresponding degree of
C...  freedom and it also updates nodal coordinates
C...
      SUBROUTINE ORDY(I,J,MO,NO,D,X)
      PARAMETER(NEL = 180,NDF = 243,NN = 117)
      DOUBLE PRECISION COOR(NN,3),CORA(NN,3),D,DIFU,
     +R(NDF),U1,U2,X(NDF),X1,X2
      COMMON/TWO/A,COOR,VCT,CORA
      INTEGER A(NEL,2),END2,END1,VCT(NEL,6)
      IR = J + 3
      END1 = A(I,1)
      END2 = A(I,2)
```

```
            X2 = COOR(END2,J)
            X1 = COOR(END1,J)
            MO = VCT(I,J)
            NO = VCT(I,IR)
            U1 = 0
            U2 = 0
            IF (MO.NE.0) THEN
            U1 = X(MO)
            END IF
            IF (NO.NE.0) THEN
            U2 = X(NO)
            END IF
            CORA(END2,J) = X2 + U2
            CORA(END1,J) = X1 + U1
            DIFU = U2 - U1
            D = X2 - X1 + DIFU
            RETURN
            END
C...
C...    Subroutine DFMCG is typed here
C...    Description of parameters:-
C...    FUNCT - User written subroutine concerning the
C...    function to be minimized. It must be of the form
C...    SUBROUTINE FUNC(N,ARG,VAL,GRAD) and must serve
C...    the following purpose: For each N dimensional
C...    argument vector ARG, function value and gradient
C...    must be computed and on return, stored in VAL and
C...    GRAD respectively. ARG, VAL and GRAD must be of
C...    double precision
C...    N          = Number of variables
C...    X          = Vector of dimension N containing the
C...                 argument vector where the iteration
C...                 starts. On return, X holds the
C...                 argument corresponding to the
C...                 computed minimum function value.
C...                 Double precision vector.
C...    F          = Single variable containing the
C...                 minimum function value on return i.e.
C...                 F = F(X). Double precision variable.
C...    G          = Vector of dimension N containing the
C...                 gradient vector corresponding to the
C...                 minimum on return i.e., G = G(X).
C...                 Double precision vector.
C...    EST        = Is an estimate of the minimum
C...                 function value. Single Precision.
C...    EPS        = Test value representing the expected
C...                 absolute error. A reasonable choice
C...                 is 10 **(-16) i.e., somewhat greater
C...                 than 10 **(-D), where D is the number
C...                 of significant digits in floating
```

```
C...                    point representation. Single
precision
C...                    variable.
C...    LIMIT    = Maximum number of iterations.
C...    IER      = Error parameter
C...    IER      = 0 means convergence was obtained
C...    IER      = 1 means no convergence in the limit
of
C...                    iterations.
C...    IER      = -1 means error in gradient
C...                    calculations
C...    IER      = 2 means linear search indicates the
C...                    probability of no minimum.
C...    H        = Working storage of dimension 2*N.
C...                    Double precision array
C...
        SUBROUTINE DFMCG(FUNC,KOUNT,LIMIT,EPS,EST,IER,
       +F,G,H,N,X)
C...
C...  Dimensioned dummy variable
C...
        DIMENSION X(243),G(243),H(486)
        DOUBLE PRECISION X,G,GNRM,H,HNRM,F,FX,FY,OLDF,
       +OLDG,SNRM,AMBDA,DALFA,T,Z,W,DX,DY,ALFA,GRAM
C...
C...  Compute function value and gradient vector for
C...  initial argument
C...
        CALL FUNC(N,X,F,G)
C...
C...  Reset iteration counter
C...
        KOUNT = 0
        IER = 0
        N1 = N + 1
C...
C...  Start iteration cycle for every N + 1 iterations
C...
1       DO 43 II = 1,N1
C...
C...  Step iteration counter and save function value
C...
        KOUNT = KOUNT + 1
        OLDF = F
C...
C...  Compute square of gradient and terminate if zero
C...
        GNRM = 0.D0
        DO 2 J = 1,N
2       GNRM = GNRM + G(J)*G(J)
```

```
        IF (GNRM) 46,46,3
C...
C...    Each time the iteration loop is executed, the
C...    first step will be in the direction of the
C...    steepest descent.
C...
3       IF (II -1) 4,4,6
4       DO 5 J = 1,N
5       H(J) = - G(J)
        GO TO 8
C...
C...    Further direction vectors will be chosen
C...    corresponding to the Conjugate Gradient Method
C...
6       AMBDA = GNRM/OLDG
        DO 7 J = 1,N
7       H(J) = AMBDA*H(J) - G(J)
C...
C...    Compute test value for directional vector and
C...    directional derivatives
C...
8       DY = 0.D0
        HNRM = 0.D0
        DO 9 J = 1,N
        K = J + N
C...
C...    Save argument vector
C...
        H(K) = X(J)
        HNRM = HNRM + DABS(H(J))
9       DY = DY + H(J)*G(J)
C...
C...    Check whether function will decrease stepping
C...    along H and skip linear search if not
C...
        IF (DY) 10,42,42
C...
C...    Compute scale factor used in linear search
C...    subroutine
C...
10      SNRM = 1.D0/HNRM
C...
C...    Search minimum along direction H
C...    Search along H for positive directional
derivative
C...
        FY = F
        ALFA = 2.D0 * (EST - F)/DY
        AMBDA = SNRM
C...
```

```
C...    Use estimate for step size only if it is positive
C...    and less than SNRM. Otherwise use SNRM as
stepsize
C...
        IF (ALFA) 13,13,11
11      IF (ALFA - AMBDA) 12,13,13
12      AMBDA = ALFA
13      ALFA = 0.D0
C...
C...    Save function and derivative values for old
C...    argument
C...
14      FX = FY
        DX = DY
C...
C...    Step argument along H
C...
        DO 15 I = 1,N
15      X(I) = X(I) + AMBDA*H(I)
C...
C...    Compute function value and gradient for  new
C...    argument
C...
        CALL FUNC(N,X,F,G)
        FY = F
C...
C...    Compute directional derivative DY for new
C...    argument. Terminate search if DY is positive. If
C...    DY is zero, the minimum is found.
C...
        DY = 0.D0
        DO 16 I = 1,N
16      DY = DY + G(I)*H(I)
        IF (DY) 17,38,20
C...
C...    Terminate search also if the function value
C...    indicates that a minimum has been passed.
C...
17      IF (FY - FX) 18,20,20
C...
C...    Repeat search and double step size for further
C...    searches
C...
18      AMBDA = AMBDA + ALFA
        ALFA = AMBDA
C...
C...    Terminate if the change in argument gets very
C...    large
C...
        GRAM = HNRM*AMBDA - 1.D10
```

```
        IF (HNRM*AMBDA - 1.D10) 14,14,19
C...
C...    Linear search indicates that no minimum exists
C...
19      IER = 2
C...
C...    Restore old values of function and arguments
C...
        F = OLDF
        DO 100 J = 1,N
        G(J) = H(J)
        K = N + J
100     X(J) = H(K)
        RETURN
C...
C...    End of search loop
C...    Interpolate cubically in the interval defined by
C...    the search above and compute the argument X for
C...    which the interpolation polynomial is minimized.
C...
20      T = 0.
21      IF (AMBDA) 22,38,22
22      Z = 3.D0*(FX - FY)/AMBDA + DX + DY
        ALFA = DMAX1(DABS(Z),DABS(DX),DABS(DY))
        DALFA = Z/ALFA
        DALFA = DALFA*DALFA - DX/ALFA*DY/ALFA
        IF (DALFA) 23,27,27
C...
C...    Restore old values of function and arguments
C...
23      DO 24 J = 1,N
        K = N + J
24      X(J) = H(K)
        CALL FUNC(N,X,F,G)
C...
C...    Test for repeated failure of iteration
C...
25      IF (IER) 47,26,47
26      IER = -1
        GOTO 1
27      W = ALFA*DSQRT(DALFA)
        ALFA = DY - DX + W + W
        IF (ALFA) 270,271,270
270     ALFA = (DY - Z + W)/ALFA
        GOTO 272
271     ALFA = (Z + DY - W)/(Z + DX + Z + DY)
272     ALFA = ALFA*AMBDA
        DO 28 I = 1,N
28      X(I) = X(I) + (T - ALFA)*H(I)
C...
```

```
C...   Terminate, if the value of the actual function at
C...   X is less than the function values at the
interval
C...   ends. Otherwise reduce the interval by choosing
C...   one end point equal to X and repeat the
C...   interpolation. Which end point is chosen depends
C...   on the value of the function and its gradient at
X
C...

       CALL FUNC(N,X,F,G)
       IF (F - FX) 29,29,30
29     IF (F - FY) 38,38,30
C...
C...   Compute the directional derivative
C...
30     DALFA = 0.D0
       DO 31 I = 1,N
31     DALFA = DALFA + G(I)*H(I)
       IF (DALFA) 32,35,35
32     IF (F - FX) 34,33,35
33     IF (DX - DALFA) 34,38,34
34     FX = F
       DX = DALFA
       T = ALFA
       AMBDA = ALFA
       GOTO 21
35     IF (FY - F) 37,36,37
36     IF (DY - DALFA) 37,38,37
37     FY = F
       DY = DALFA
       AMBDA = AMBDA - ALFA
       GOTO 20
C...
C...   Terminate if function has not decreased during
C...   last iteration. Otherwise save gradient normal.
C...
38     IF (OLDF - F + EPS) 19,25,39
39     OLDG = GNRM
C...
C...   Compute difference of old and new argument vector
C...
       T = 0.
       DO 40 J = 1,N
       K = J + N
       H(K) = X(J) - H(K)
40     T = T + DABS(H(K))
C...
C...   Test length of difference vector if at least N +
1
C...   iterations have been executed. Terminate, if
```

```
C...    length is less than EPS.
C...
        IF (KOUNT - N1) 42,41,41
41      IF (T - EPS) 45,45,42
C...
C...    Terminate if number of iterations exceed limit
C...
42      IF (KOUNT - LIMIT) 43,44,44
43      IER = 0
C...
C...    End of iteration cycle and start next iteration
C...    cycle
C...
        GOTO 1
C...
C...    No convergence after limit iterations
C...
44      IER = 1
        IF (GNRM - EPS) 46,46,47
C...
C...    Test for sufficiently small gradient
C...
45      IF (GNRM - EPS) 46,46,25
46      IER = 0
47      RETURN
        END
```

REFERENCES

1. Baron, F. and Venkatesan, M.S. *Nonlinear Analysis of Cable and Truss Structures*, Journal of the Structural Division, ASCE No. ST2, pp.679-710, Feb.1971.

2. Bathish, G. N. *Membrane Analysis of Cable Roofs*, Ph.D. Dissertation, University of Pennsylvania, 1966.

3. Bogner, F. K., Mallet, R. H., Minich, M. D., and Schmit, L. A. *Development and Evaluation of Energy Search Methods of Nonlinear Structural Analysis*, AFFDL-TR-65-113, Air Force Flight Dynamics Laboratory, Wright-Patterson Air Force Base, Ohio, 1965.

4. Buchholdt, H.A., Das, N.K. and Al-Hilli, A.J., *A Gradient Method for the Analysis of Cable Structures with Flexible Boundaries* , International Conference on Tension Roof Structures, London, April 1974.

5. El-Hakim, N. E. M. *Nonlinear Energy Search Analysis of Truss-Type Structural Systems* M.A.Sc. Thesis, Faculty of Graduate Studies and Research, University of Windsor, 1975.

6. Fletcher, R. and Powell, M. J. D. *A Rapidly Convergent Descent Method for Minimization*, Computer Journal, Vol.6, pp.163-168, 1963.

7. Fletcher, R. and Reeves, L. M. *Function Minimization by Conjugate Gradients*, Computer Journal, Vol.7, pp.149-154, 1964.

8. Fox, R. L. and Stanton, E. L. *Developments in Structural Analysis by Direct Energy Minimization*, AIAA Journal, Vol.6, pp.1036-1042, June 1968.

9. Greenberg, D. P. *Suspension Roof Structures: Their Elastic and Inelastic Behaviour and their Ultimate Load Capacities*, Ph.D. Dissertation, Cornell University, 1968.

10. Greenberg, D. P. *Inelastic Analysis of Suspension Roof Structures*, Journal of the Structural Division, ASCE, No.ST5, pp.905-931, May 1970.

11. Haug, E. and Powell, G. H. *Discussion of "Inelastic Analysis of Suspension Roof Structures", by Donald P. Greenberg*, Journal of the Structural Division, ASCE, No.ST4, pp.1360-1363, April 1971.

12. Hestenes, M.R., *Conjugate Direction Methods in Optimization*, Springer-Verlag, New York, 1980.

13. Jonatowski, J. J. and Birnstiel, C. *Inelastic Stiffened Suspension Space Structures*, Journal of the Structural Division, ASCE, No.ST6, pp.1143-1166, June 1970.

14. Journal of the Precast Concrete Institute, *B.C. Place Stadium*, pp.132 - 143, March - April 1984.

15. Kar, A. K. and Okazaki, C. Y. *Convergence in Highly Nonlinear Cable Net Problems*, Journal of the Structural Division, ASCE, No.ST3, pp.321-334, March 1973.

16. Krishna, P. *Cable Suspended Roofs*, McGraw-Hill Book Company, New York, 1978.

17. Krishna, P. and Agarwal, T. P. *Study of Suspended Roof Model*, Journal of the Structural Division, ASCE, No.ST6, pp.1671-1684, June 1971.

18. Kuester, J.L. and Mize, J.Y. *Optimization Techniques with Fortran*, McGraw-Hill Book Company, New York 1973.

19. Kumanan, T. *Elastic and Inelastic Analysis of Prestressed Cable-Roof Structures*, Ph.D. Dissertation, Faculty of Graduate Studies and Research, University of Windsor, 1971.

20. Leonard, J. W. *Tension Structures: Behavior and Analysis*, McGraw-Hill Book Company, New York, 1988.

21. Malcolm, D.J. and Glockner, P.G. *Optimum Cable Configurations for Air Supported Structures*, Journal of the Structural Division, ASCE, No. ST2, Feb. 1979.

22. Mallet, R. H. *A Mathematical Programming Approach to Nonlinear Structural Analysis*, Research Report No. EDL 2-65-10, AFFDL Research Contract AF 33(615) - 1022.

23. Monforton, G. R. *Energy Search: Nonlinear Microcomputer Analysis of Suspension Structures*, Proceedings of the First National Conference on Microcomputers in Civil Engineering, Orlando, Florida, NN 1-3, pp. 271-276, 1983.

24. Monforton, G. R. and El-Hakim N. M. *Analysis of Truss Cable Structures*, Computers and Structures, Vol.11, pp.327-335, 1986.

25. Saafan, S. A. *Theoretical Analysis of Suspension Roofs*, Journal of the Structural Division, ASCE, No.ST2, pp.393-405, Feb. 1970.

26. Schmit, L. A. Jr., *A Research Proposal to Develop and Evaluate Energy Search Methods of Structural Analysis*, Research Proposal, Case Institute of Technology, Cleveland, Ohio.

27. Siev, A., *A General Analysis of Prestressed Nets*, International Association for Bridge and Structural Engineering, Vol.23, pp.283-293, 1963.

28. Siev, A., *Prestressed Suspended Roofs Bounded by Main Cables*, International Association for Bridge and Structural Engineering, Vol.27, pp.171-185, 1967.

29. Szabo, J., Kollar, L. and Pavlovic, M. N. *Structural Design of Cable Roofs*, Ellis Horwood Limited, England, 1984.

30. Thornton, C. H. and Birnstiel, C., *Three Dimensional Suspension Structures*, Journal of the Structural Division, ASCE, No.ST2, pp.247-270, April 1967.

VITA AUCTORIS

1966	Born on January 12th in Hyderabad, India
1981	Graduated from St.George's Grammar School, Hyderabad, India
1983	Graduated from St.Joseph's Junior College, Hyderabad, India
	Accepted into the Engineering Program at Kakatiya University, Warangal, India
1987	Graduated with honours from Kakatiya University
	Started employment as an engineer in KK Constructions, Hyderabad, India in June
1988	Accepted into the M.A.Sc Program in Civil Engineering at the University of Windsor