

NONLINEAR ANALYSIS OF CABLE STRUCTURES

by

William M. Sage

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of
Master of Science
in
Civil Engineering

APPROVED:

S.M. Holzer, Chairman

R.H. Plaut

T. Kuppusamy

December, 1986
Blacksburg, Virginia

NONLINEAR ANALYSIS OF CABLE STRUCTURES

by

William M. Sage

S.M. Holzer, Chairman

Civil Engineering

(ABSTRACT)

Using the principles of continuum mechanics, the general, incremental, geometrically nonlinear, equilibrium equations for the displacement based finite element method are derived in both the Total Lagrangian and Updated Lagrangian form. The Updated Lagrangian equations are then specialized for the analysis of three dimensional cable structures. Two node and three node isoparametric finite elements are derived and a program is developed which implements both elements. Comparisons between the two elements are made to determine their relative merits.

In addition, four separate iterative solution techniques are investigated in the solution of the equilibrium equations: the Newton-Raphson method, the Modified Riks-Wempner method, an Orthogonal Trajectory Accession method, and a Dynamic Relaxation approach. An extension of this investigation is a comparison between the Riks-Wempner method and Orthogonal Trajectory Accession.

Acknowledgements

The author wishes to extend his sincere appreciation to Dr. S.M. Holzer for his outstanding support and guidance during the course of this research. In addition, the author would like to thank Dr. R.H. Plaut and Dr. T. Kuppusamy for their review of the thesis and participation on the committee.

A special thanks goes to all the faculty of the structures division for their outstanding dedication to the teaching profession.

This thesis is dedicated in loving memory of my father, Raymond J. Sage, and to my family for their support during this research.

Table of Contents

| | |
|--|-----------|
| Chapter 1 | 1 |
| INTRODUCTION | 1 |
| 1.1 Purpose and Scope | 1 |
| 1.2 Literature Review | 2 |
| | |
| Chapter 2 | 5 |
| DEFORMATION OF A CONTINUUM | 5 |
| 2.1 Introduction | 5 |
| 2.2 Notation and Coordinates | 5 |
| 2.3 Deformation | 7 |
| 2.4 Stress at a Point | 8 |
| 2.5 The Cauchy Stress Tensor | 9 |
| 2.6 The Second Piola-Kirchhoff Stress Tensor | 12 |
| 2.7 Strain | 16 |
| 2.8 Constitutive Equations | 19 |
| | |
| Chapter 3 | 21 |
| FINITE ELEMENT FORMULATION | 21 |
| 3.1 Introduction | 21 |
| 3.2 Notation | 21 |
| 3.3 Incremental Formulation | 25 |
| 3.4 Finite Element Discretization | 30 |
| | |
| Chapter 4 | 35 |
| DEVELOPMENT OF CABLE ELEMENTS | 35 |
| 4.1 Introduction | 35 |

| | |
|--|------------|
| 4.2 Incremental Formulation for Cable Elements | 35 |
| 4.3 Linear Element | 41 |
| 4.4 Curvilinear Element | 44 |
| 4.5 Member Actions | 46 |
| Chapter 5 | 51 |
| SOLUTION TECHNIQUES | 51 |
| 5.1 Introduction | 51 |
| 5.2 The Tangent Stiffness Matrix | 51 |
| 5.3 Newton-Raphson Methods | 53 |
| 5.4 Modified Riks-Wempner Method | 58 |
| 5.5 Orthogonal Trajectory Accession | 71 |
| 5.6 A Dynamic Relaxation Approach | 75 |
| 5.7 Convergence Criteria | 81 |
| Chapter 6 | 86 |
| CABLE ANALYSIS | 86 |
| 6.1 Introduction | 86 |
| 6.2 The Freely Hanging Cable | 86 |
| 6.3 The Initial Shape | 89 |
| 6.4 Methods of Analysis | 99 |
| Chapter 7 | 100 |
| PROGRAM DEVELOPMENT | 100 |
| 7.1 Introduction | 100 |
| 7.2 Program Subroutines | 100 |
| Chapter 8 | 134 |

| | |
|---|------------|
| TEST PROBLEMS | 134 |
| 8.1 Introduction | 134 |
| 8.2 Test Set 1 | 134 |
| 8.3 Test Set 2 | 140 |
| 8.4 Test Set 3 | 145 |
| 8.5 Test Set 4 | 150 |
| | |
| Chapter 9 | 159 |
| CONCLUSION | 159 |
| 9.1 Conclusions | 159 |
| 9.2 Recommendations For Further Studies | 160 |
| | |
| Appendix A. | 163 |
| REFERENCES | 163 |
| | |
| Appendix B. | 166 |
| PROGRAM LISTING | 166 |
| | |
| VITA | 199 |

List of Illustrations

| | |
|---|----|
| Figure 2.1. Body Under Loading | 10 |
| Figure 2.2. Stress Across a Surface | 10 |
| Figure 2.3a. Stress on Mutually Perpendicular Surfaces | 11 |
| Figure 2.3b. The Cauchy Stress Components | 11 |
| Figure 2.4. Stress On a Plane Before and After Deformation | 14 |
| Figure 2.5. The Piola-Krichhoff Stress Transformation | 14 |
| Figure 2.6. Strain in a Body | 17 |
| Figure 3.1. Incremental Deformation of a Continuum | 23 |
| Figure 4.1. General Cable Elements | 38 |
| Figure 4.2. Strain in the Cable Element | 40 |
| Figure 4.3. Dead Load (a) Two Node Element and, (b) Three Node Element | 48 |
| Figure 4.4. Uniform Load (a) Two Node Element and, (b) Three Node Element | 50 |
| Figure 5.1. Physical Interpretation of the Newton-Raphson Method | 54 |
| Figure 5.2. The Modified Newton-Raphson Method | 56 |
| Figure 5.3. Failure of the Newton-Raphson Methods | 57 |
| Figure 5.4. Incremental-Iterative Newton-Raphson Method | 59 |
| Figure 5.5. The Modified Riks-Wempner Method | 60 |
| Figure 5.6. Physical Interpretation of the Modified Riks-Wempner Method | 63 |
| Figure 5.7. Scaling the Arc Length in the Modified Riks-Wempner Method | 66 |
| Figure 5.8. Loading and Unloading of the Modified Riks-Wempner Method | 67 |
| Figure 5.9. Determination of Correct Sign | 68 |
| Figure 5.10. Incorrect Sign Change | 69 |
| Figure 5.11. Physical Interpretation of Orthogonal Trajectory Accession | 72 |
| Figure 5.12. Orthogonal Trajectory Accession Finding the Curve | 74 |
| Figure 5.13. Poor Arc Length in Orthogonal Trajectory Accession | 74 |
| Figure 5.14. Problems of Convergence | 85 |

| | |
|--|-----|
| Figure 6.1. Cable Under Point Loads | 88 |
| Figure 6.2. Nonlinear Deflection of a Pretensioned Cable | 90 |
| Figure 6.3. Catenary Segment of Cable | 92 |
| Figure 6.4. Simply Suspended Cable | 95 |
| Figure 6.5. Straight Line Cable Segment | 98 |
| Figure 7.1. Program Structure | 109 |
| Figure 7.2. Main Program | 110 |
| Figure 7.3. Subroutine DATA | 110 |
| Figure 7.4. Subroutine STRUCT | 110 |
| Figure 7.5. Subroutine CODES | 111 |
| Figure 7.6. Subroutine DETMAX | 111 |
| Figure 7.7. Subroutine PROP | 112 |
| Figure 7.8. Subroutine LOAD | 112 |
| Figure 7.9. Subroutine JLOAD | 113 |
| Figure 7.10. Subroutine MACT | 114 |
| Figure 7.11. Subroutine NEWRAP | 115 |
| Figure 7.12. Subroutine RIKWEM | 117 |
| Figure 7.13. Subroutine ORTHOG | 119 |
| Figure 7.14. Subroutine DYNAM | 122 |
| Figure 7.15. Subroutine FORCES | 124 |
| Figure 7.16. Subroutine ELEMFI | 124 |
| Figure 7.17. Subroutine EXTRN1 | 125 |
| Figure 7.18. Subroutine ELEMFI | 126 |
| Figure 7.19. Subroutine EXTRN2 | 127 |
| Figure 7.20. Subroutine STIFF | 128 |
| Figure 7.21. Subroutine CBL1 | 128 |
| Figure 7.22. Subroutine CBL2 | 129 |
| Figure 7.23. Subroutine ASSEMS | 129 |

| | |
|--|-----|
| Figure 7.24. Subroutine TEST | 130 |
| Figure 7.25. Subroutine ENERGY | 130 |
| Figure 7.26. Subroutine UNBALF | 131 |
| Figure 7.27. Subroutine DISPL1 | 131 |
| Figure 7.28. Subroutine RESULT | 132 |
| Figure 7.29. Subroutine JOINTF | 132 |
| Figure 7.30. Subroutine OUTPUT | 132 |
| Figure 7.31. Subroutine UPDATC | 133 |
| Figure 7.32. Subroutine UPDATF | 133 |
| Figure 7.33. Subroutine SHAPE | 133 |
| Figure 7.34. Subroutine DOTPRD | 133 |
| Figure 8.1. Test Set 1: A Freely Hanging Cable | 135 |
| Figure 8.2. Test Set 1: Resulting Displacements of Cable | 137 |
| Figure 8.3. Test Set 1: Comparison of Mesh Refinement | 138 |
| Figure 8.4. Test Set 1: Newton-Raphson Vs. Riks-Wempner | 139 |
| Figure 8.5. Test Set 1: Two Node Element Vs. Three Node Element | 141 |
| Figure 8.6. Test Set 2: Hyperbolic Cable Net | 142 |
| Figure 8.7. Test Set 2: Displacements Along $x = 0$ | 143 |
| Figure 8.8. Test Set 2: Displacements Along $z = 0$ | 144 |
| Figure 8.9. Test Set 3: Displacements of Freely Hanging Cable Using DR | 146 |
| Figure 8.10. Test Set 3: Truss-Spring Model | 147 |
| Figure 8.11a. Test Set 3: Load-Displacement of Truss-Spring Using DR | 148 |
| Figure 8.11b. Test Set 3: Load-Displacement of Truss-Spring Using DR | 149 |
| Figure 8.12. Test Set 4: 21 DOF Lamella Dome | 152 |
| Figure 8.13. Test Set 4: Riks-Wempner, With and Without Limit | 153 |
| Figure 8.14. Test Set 4: Riks-Wempner Vs. OTA | 155 |
| Figure 8.15. Test Set 4: One Degree of Freedom Truss Model | 156 |
| Figure 8.16. Test Set 4: Starting Ability of OTA | 157 |

Chapter 1

INTRODUCTION

1.1 Purpose and Scope

The objective of this study is to investigate the nonlinear behavior exhibited by cable structures under static loading. The intent of the investigation is to apply nonlinear finite element techniques to the analysis of cables and cable grids using a simple two node linear element and a more complex three node curvilinear element. Comparisons between the two elements will be made based on the accuracy achieved by each in an attempt to determine their relative merits.

With the advent of high strength, low relaxation steels, the use of cables as primary structural components has become more attractive to engineers and architects. When combined with the advances in durable membrane fabrics, the use of cable supported and cable reinforced enclosures has become recognized as an efficient and viable alternative to conventional building materials. One major drawback in the design of cable systems, however, is the tendency of the cables to be quite flexible and thus highly nonlinear. The requirement of considering nonlinear effects in the analysis precludes the use of classical, closed form techniques from all but the simplest of structures. Nonlinear finite element analysis of solid structures has become almost routine in the last few decades, yet there are a wide variety of approaches in the analysis of tension structures. This creates difficulty for the engineer in deciding which approach is best for a given structure. A result of this study will be a better understanding of the finite element method as applied to the analysis of cable structures.

In addition to studying the application of the finite element method, it is necessary to focus some attention on the various techniques used in the solution of the equilibrium equations. The tendency of cable systems to be very soft under certain conditions requires that a numerically stable routine be employed in the incremental solution of the problem. For this reason a brief discussion

will be presented on four solution techniques: the Newton-Raphson method, the Modified Riks-Wempner method, an Orthogonal Trajectory Accession method, and an adaptation of Dynamic Relaxation.

In order that the study be, to some degree, self-contained, the development of the nonlinear equations will be given in a general format prior to specialization for cable elements. This requires an overview of the basic principles of continuum mechanics, given in Chapter 2. This is followed by the finite element discretization of a general two-dimensional system for plain stress (strain) analysis in Chapter 3.

Finally, a computer program will be developed which implements the analysis procedures and solution techniques in a well documented, structured format. It is the intention of the author that this program be easily understandable for future researchers who may continue the work started here.

1.2 Literature Review

The use of cables as main structural elements is thought to date back to the construction of foot bridges from natural vines [26]. Not until recent decades, however, has the use of cables for other applications become more acceptable. One author whose extensive work in the physical modeling and practical applications of cables created much of this attention, is Frei Otto [35]. His presentation of the almost limitless forms of cable structures has been the foundation for many further studies in this area.

The current level of practical analysis techniques of cables is probably best illustrated in the texts by Krishna [26], Buchholdt [7], and Sazbo [40]. The emphasis of these texts is on the presentation of classical analysis in a manner suitable for common design problems. In addition, Krishna presents a short computer program which implements the Newton-Raphson method in the solution process.

With increased attention in structural analysis being focused on the use and refinement of the finite element method, numerous articles on the analysis of cable structures have been presented [16,22,29,39,41]. Generally, there are three elements which have been developed for the analysis.

The two node linear element [1,2] and the three node curvilinear element [22,36,41] used in this study are the most common of these. The third type is a curved element which uses the catenary equations as interpolation functions rather than the Lagrangian interpolation used with the three node element [16,39]. This element is particularly attractive in the analysis of structures which consist of single cables, such as mooring lines or cable stayed towers [39]. The application of this element to cable networks, however, is rather limited.

Of primary concern in the analysis of a cable structure is the determination of an initial shape for the system. This "shape-finding" is especially important in the analysis of pneumatic structures. Recent work by Haber [19,20] and others [1,21,22] has focused attention on the development of a generalized approach to solving this problem.

In the analysis of any nonlinear problem, some attention is invariably drawn to the application of an efficient, reliable solution technique. The most common routines are the Newton-Raphson method and the Modified Newton-Raphson method. These approaches are outlined in virtually all textbooks which deal with the solution of nonlinear equations [2,10,48].

The Modified Riks-Wempner method [25,45] is a common alternative to the Newton methods, particularly in the analysis of problems where the equilibrium path contains a turning point. A variation of this approach, developed by Fried [15], uses orthogonal trajectory accession to trace the equilibrium path. Partly as a result of the recent introduction of this method, the published work using this approach is rather limited.

The fourth solution technique which was included in the study is an adaptation of Dynamic Relaxation to the analysis of cables. This technique was developed in the mid 1960's for use in linear analysis. An excellent overview of the method is provided by Underwood [43] in his development of the particular variation used in this study. The application of Dynamic Relaxation is generally unique for each author, however the general procedures remain the same. The use of this technique in the analysis of various structures can be found in [6,8,13,36,37,43]. The paper on a variable step approach to the central difference method by Underwood and Park [38] is particularly interesting in the study of Underwood's approach to Dynamic Relaxation.

Finally, as a guide for others interested in the continuum mechanics based finite element formulation, the works by Malvern [30] and Fung [17] are recommended for a solid introduction to continuum mechanics. In addition, the report of Mondakar and Powell [31] offers a well structured approach to the development of the incremental equilibrium equations.

Chapter 2

DEFORMATION OF A CONTINUUM

2.1 Introduction

The development of the fundamentals of nonlinear finite element analysis requires a basic understanding of the concepts involved in the study of finite deformation of a continuum. This chapter presents these concepts of continuum mechanics, focusing attention on those areas which lead directly to finite element applications. First, a brief discussion on the notations used in the formulations is presented, followed by the development of the stress and strain equations to be used in the following chapters.

2.2 Notation and Coordinates

The need to describe the continuum in such a manner as to be independent of the choice of coordinate systems requires that equations derived in continuum mechanics involve extensive use of tensors and tensor transformations [17,30]. Although it is not the intent of this section to review the theories behind tensor calculus, it is necessary to clarify the notations used in the application of tensors. More in-depth discussions on tensors can be found in references such as 14, 17, and 30.

Indicial Notation

By nature, tensors involve the use of multidimensional space and therein can involve quite complex arithmetic if not expressed in a symbolic form. This symbolic representation used with tensors is referred to as indicial notation [14,17,30] and involves the use of indices to represent the n -dimensional space. In the following discussion, and throughout this work, n is limited to a three dimensional Euclidean space commonly defined by the rectangular cartesian coordinate system (x,y,z) . With the use of indicial notation, however, the coordinate system is represented by (x_1,x_2,x_3) . Further simplification can be made by using just the index i which assumes the values

1,2 or 3. This same coordinate system can then be expressed simply as x_i . As a further demonstration, consider the vector A in an arbitrary orientation with initial point at the origin of the coordinate system. In component form, A may be represented as

$$A = a_1 n_1 + a_2 n_2 + a_3 n_3 \quad (2.2 - 1)$$

where (a_1, a_2, a_3) correspond to the magnitude of the projection of A onto the x_i axes, and (n_1, n_2, n_3) are unit vectors in the directions of the axes. Using indicial notation, Eq. 2.2-1 can be written as

$$A = \sum_{i=1}^3 a_i n_i \quad (2.2 - 2)$$

Summation Convention [14,17,30]

A summation convention used with indicial notation can be employed to further simplify Eq.

2.2-2. The basic rules are as follows:

1. In any expression using index notation, the repetition of an index within one term (whether superscript or subscript) implies that the index is to take on all possible values in its range and the results from that term are to be added. Thus Eq. 2.2-2 can be written as $A = a_i n_i$ where the summation is now implied.
2. If an index is not repeated in a term, it is considered to be a free index and takes on only one value for each separate equation. As an example, the equation $c_i = a_i + b_i$ represents three separate equations. Similarly, the equation $c_{ij} = a_{ij} + b_{ij}$ represents a total of nine equations.

Kronecker Delta and Permutation Symbols

Two additional symbols, which result from index notation and become quite useful in simplifying equations, are the Kronecker delta (δ) and the permutation symbol (e_{ijm}) [14,17,30]. The Kronecker delta is defined as

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \quad (2.2 - 3)$$

and the permutation symbol is defined as

$$e_{ijk} = \begin{cases} 1 & \text{when } i,j,k \text{ are } 1,2,3 \text{ or any even} \\ & \text{permutation of } 1,2,3. \\ 0 & \text{if any two indices are equal} \\ -1 & \text{when } i,j,k \text{ are an odd permutation of } 1,2,3. \end{cases} \quad (2.2 - 4)$$

As an example

$$\begin{aligned} e_{123} &= e_{231} = e_{312} = 1 \\ e_{132} &= e_{213} = e_{321} = -1 \\ e_{113} &= e_{112} = e_{222} = 0 \end{aligned}$$

Coordinate System

In general, the use of tensors allows for the coordinate systems to be either curvilinear or rectangular with the only requirement being that they define a Euclidean space [30]. For this work, however, the focus is on a rectangular cartesian coordinate system and thus all indices are assumed to have values of 1,2, or 3 unless otherwise specified.

2.3 Deformation

The concept of deformation of a medium as employed in continuum mechanics involves the "mapping" of a body from one configuration to another [17]. This mapping process uses the assumption that the body, in its original configuration and final configuration, can be expressed by a continuous mathematical model. This requirement implies that the material within the body is continuous, without voids or gaps, thus eliminating the need to consider molecular structure of the medium [14]. Furthermore, it is assumed that during this deformation, the motion of all particles within the medium is continuous, with the event of tearing or breaking being considered as a special case. Employing these simple guidelines allows for the direct application of the theories of calculus in the formulations to follow [30].

In the mapping process, there are two basic methods used to describe the deformation: the Lagrangian approach and the Eulerian approach. To demonstrate each approach, consider a medium in two configurations: an initial state at time $t = 0$ and a deformed state at time t . A point P within the initial configuration has the coordinates (a_1, a_2, a_3) . In the deformed configuration, the point P becomes P^* with coordinates (x_1, x_2, x_3) .

Lagrangian Description

The Lagrangian description, sometimes referred to as a referential description [18,30], seeks to describe the motion of P solely in terms of the initial configuration. This implies that the coordinates of P^* are functions of the original coordinates a_i and can be expressed as

$$x_i = x_i(a_j) \quad (2.3 - 1)$$

Eulerian Description

The Eulerian description, on the other hand, seeks to describe the motion of P in terms of the current coordinates of the body. Thus the original coordinates, a_i , are expressed in terms of the new coordinates x_i as

$$a_i = a_i(x_j) \quad (2.3 - 2)$$

Though the two approaches are fundamentally equivalent, the Lagrangian approach is more commonly used in solid mechanics formulations whereas the Eulerian description is commonly used in the study of fluid mechanics [14,30].

2.4 Stress at a Point

Stresses within a body come as a direct result of the loading which the body is subjected to. These actions can be broken up into two distinct types, those which act across a surface of the body (internal or external) and those which act on the body's volume or mass [17,30]. The first type, known as surface loadings or surface tractions, are a result of contact between the body and its environment. Such actions include concentrated loads, pressures, and reactions. These loads are considered to be sectionally continuous since they may act independently, over discrete portions of the surface [34]. The second type of loading is referred to as body forces or "action-at-a-distance"

forces [14,30]. These forces are a result of exterior effects on the body such as gravity, magnetic fields, or inertial forces.

The definition of a surface traction leads directly to the concept of stress within a body. To demonstrate, consider the body shown in Figure 2.1 under the influence of various loads P_i . Let a plane Q pass through the body separating it into two distinct portions about the surface S (see Figure 2.1). Across S , the internal force distribution F must act to sustain equilibrium. Let the positive sense of F correspond to the direction of the outward normal to the surface S . Considering a portion of the surface ΔS and its corresponding portion of internal force ΔF , the definition of the average traction vector on ΔS may be stated as $\Delta F/\Delta S$. If the value of ΔS is allowed to shrink down to the point P (see Figure 2.2), it is found that the value of ΔF changes in direction as well as magnitude. However, in the limit, the ratio will approach a finite value at point P . That is,

$$\tau = \lim_{\Delta S \rightarrow 0} \frac{\Delta F}{\Delta S} \quad (2.4 - 1)$$

τ is defined as the stress vector at P and represents the pointwise force per unit area at P acting across S . It should be pointed out that both the magnitude and direction of τ depend on the orientation of n , the location of S , and on time.

2.5 The Cauchy Stress Tensor

With the stress across a surface defined, it is useful to determine the stress on planes perpendicular to the coordinate axes [30] (see Figure 2.3a). With these three vectors defined, it is possible to find the state of stress on any plane oriented arbitrarily to these planes.

Let the traction vector T^i represent the stress vector acting on the surface whose normal corresponds to the coordinate axis x_i . Corresponding to each vector, the components τ_{ij} acting on the surface in the direction of the coordinate axes can be found using a simple vector transformation (see Figure 2.3b)

$$T^i = n_j \tau_{ij} \quad (2.5 - 1)$$

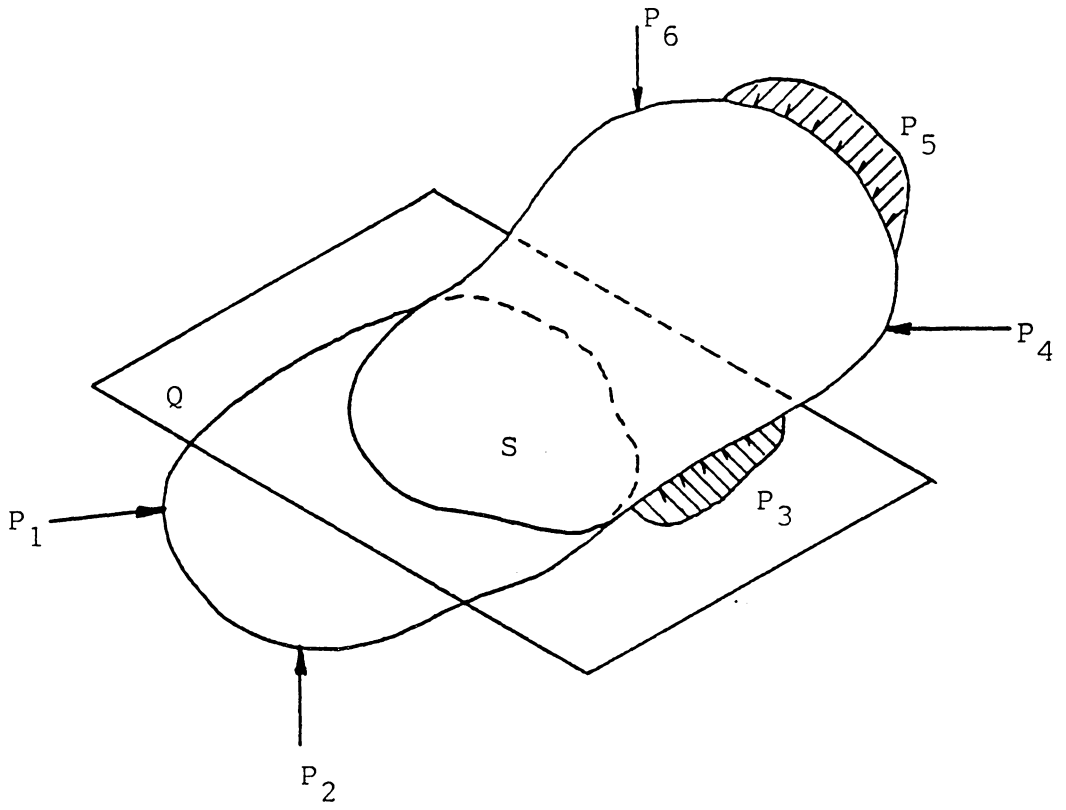


Figure 2.1: Body Under Loading

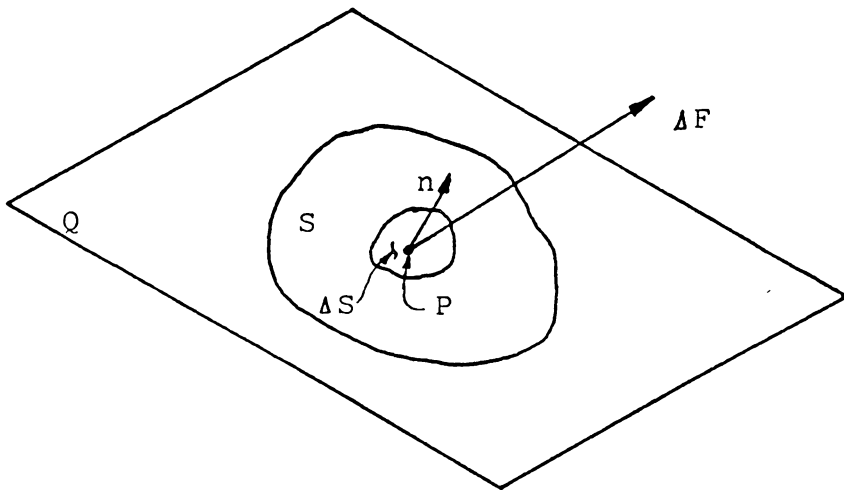


Figure 2.2: Stress Vector Across a Surface

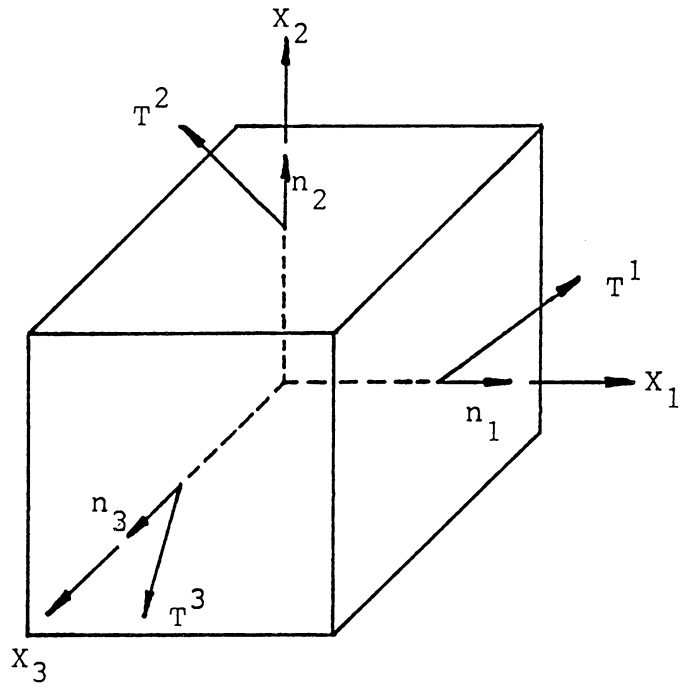


Figure 2.3a : Stress On Mutually Perpendicular Surfaces

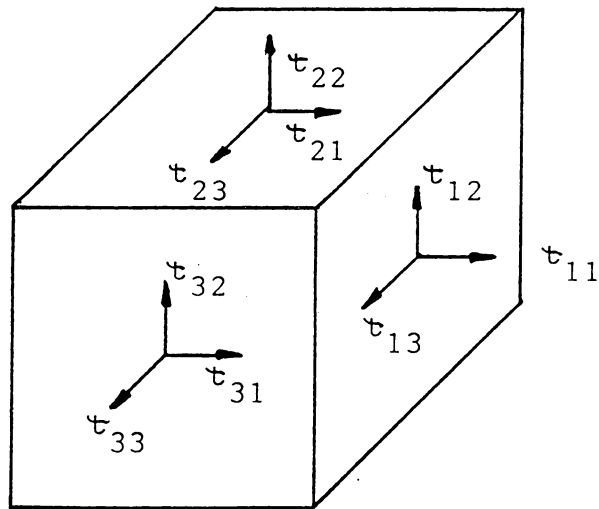


Figure 2.3b : The Cauchy Stress Components

which represents

$$\begin{aligned}
 T^1 &= n_1\tau_{11} + n_2\tau_{12} + n_3\tau_{13} \\
 T^2 &= n_1\tau_{21} + n_2\tau_{22} + n_3\tau_{23} \\
 T^3 &= n_1\tau_{31} + n_2\tau_{32} + n_3\tau_{33}
 \end{aligned}
 \tag{2.5 - 2}$$

The nine stress components τ_{ij} thus establish the three sets of vector components of the second order tensor T , referred to as the Cauchy stress tensor [17,30].

Returning to the three vectors T^i , in general it can be said that each vector may represent the average stress across the given face or simply the magnitude of the stress at the point on the face considered [30]. In either case, the magnitude and direction of the vector vary with location and time. If, however, the block is allowed to shrink down to the origin O , the three vectors will, in the limit, attain finite values and directions. Therein the complete state of stress at the point O is established.

In addition to establishing the state of stress at a point, Cauchy extended this concept to show that the stress across a plane oriented at any arbitrary angle to the three planes described above could be found in terms of the three vectors T^i . This in turn leads to the concepts of principle stresses and principle planes which are of concern in elementary mechanics [17,30].

2.6 The Second Piola-Kirchhoff Stress Tensor

The importance of determining the Cauchy stress tensor for the deformed configuration of the loaded body was established in Section 2.5. In the analysis of nonlinear media, however, the configuration which corresponds to equilibrium may be quite different from the initial shape and is usually not known beforehand. Instead, only a current reference configuration is known upon which a trial solution must be established. The problem of choosing a reference frame will be discussed in the following chapter. At this time, however, the development of a method for expressing the state of stress in one configuration as it acts in another is presented.

To develop the proper transformation, consider an elemental volume from a continuum in an original configuration at some reference time, say $t = 0$, and in a deformed state at time t (see Figure 2.4). Let the force vector at point Q be expressed as dF in the deformed state. Corresponding

to this force is the force dF_o acting at point Q_o in the original configuration. Applying the concepts of Section 2.4, the stress tensor at Q and Q_o can be expressed as equal to dF/dS and dF_o/dS_o , respectively [17].

The relationship between dF and dF_o can follow a variety of forms [2]; however, it is important that they be mathematically consistent in their implementation [17]. The two most common forms used are the First Piola-Kirchhoff stress tensor, sometimes referred to as the Lagrangian Stress [17], and the Second Piola-Kirchhoff stress tensor, referred to often as simply the Kirchhoff stress [17]. As emphasized by Bathe [2], these stresses are pseudo-stresses and have no actual physical meaning. In the development, however, it is helpful to visualize the two approaches as in Figure 2.5 [17]. The First Piola-Kirchhoff rule is simply a one-to-one correspondence between the two forces (see Figure 2.5a), that is

$$dF_o = dF$$

The Second Piola-Kirchhoff rule seeks to relate the two forces in the same manner as the transformation of an elemental length da in the original configuration is transformed to its new length dx in the deformed configuration, that is

$$dF_o = \frac{\partial a}{\partial x} dF \quad \text{just as} \quad da = \frac{\partial a}{\partial x} dx \quad (2.6 - 2)$$

Thus in the Second Piola-Kirchhoff approach, the force dF_o undergoes the same rotations and transformations that da undergoes during deformation (see Figure 2.5b).

To distinguish between the various stresses, let τ_{ij} , T_{ij} , and S_{ij} denote the Cauchy stress, the First Piola-Kirchhoff stress, and the Second Piola-Kirchhoff stress, respectively. In the case of the Cauchy stress, the force dF can be expressed as

$$dF = \tau_{ij} n_j dS \quad (2.6 - 3)$$

Similarly, the expression for the two Piola-Kirchhoff stresses are

$$dF_o = T_{ij} n_o_j dS_o \quad (2.6 - 4)$$

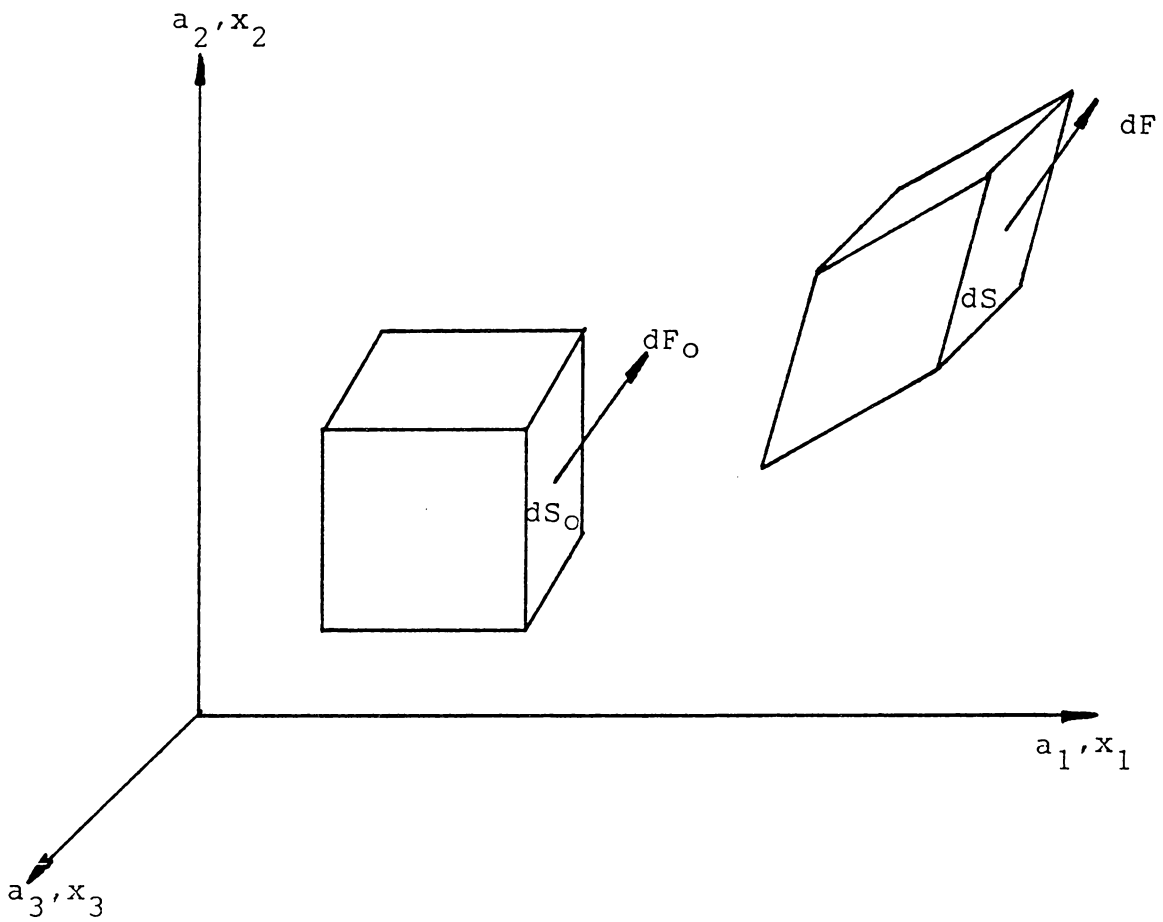


Figure 2.4: Stress On a Plane Before and After Deformation

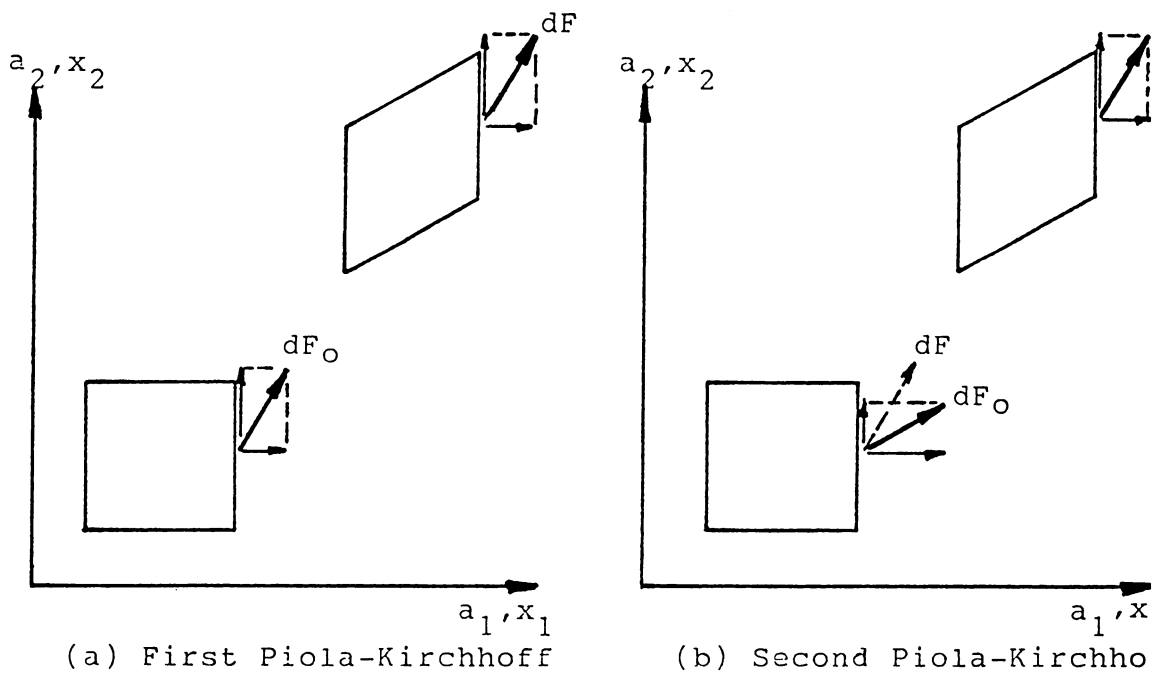


Figure 2.5: The Piola-Kirchhoff Stress Transformations

$$dF_o = S_{ij}n_o dS_o \quad (2.6 - 5)$$

The key to relating τ_{ij} , T_{ij} , and S_{ij} comes as a result of the relationship between ndS and $n_o dS_o$, which takes into account the conservation of mass during deformation and can be expressed as [2,17,30]

$$ndS = \frac{\rho_o}{\rho} X^T n_o dS_o \quad (2.6 - 6)$$

where

ρ_o = the initial mass density of the body.

ρ = the density of the body in the current state.

X = the deformation gradient.

This expression is commonly called Nanson's formula [2,30].

Now, using Eqs. 2.6-4, 2.6-5, and 2.6-6, the expressions for relating the three stress tensors become

$$T_{ij} = \frac{\rho_o}{\rho} X \tau_{ij} \quad (2.6 - 7)$$

$$S_{ij} = \frac{\rho_o}{\rho} X \tau_{ij} X^T \quad (2.6 - 8)$$

In studying the above equations, one observation can be made about the characteristics of the First and Second Piola-Kirchhoff stresses. It is seen from Eq. 2.6-7 that in general the First Piola-Kirchhoff stress tensor is not symmetric whereas Eq. 2.6-8 indicates the Second Piola-Kirchhoff stress tensor is symmetric whenever the Cauchy stress tensor is symmetric. This characteristic of the First Piola-Kirchhoff stress tensor becomes quite troublesome when used with a symmetric strain tensor. Thus, the Second Piola-Kirchhoff stress tensor is used almost exclusively [2,30].

Further discussion on the implementation of the Cauchy and Second Piola-Kirchhoff stresses will be given in the next chapter on incremental formulation.

2.7 Strain

As a body undergoes deformation, each particle within the medium is subjected to a series of rotations and translations which, in general, may or may not be considered as small [17]. During this motion, neighboring particles move relative to one another in a random fashion consistent only with the nature of the loading on the body, the boundary conditions imposed on the system, and the material behavior. Strain in the body as a result of deformation is defined as that part of this relative displacement which is not a direct result of rigid body motion [17,30]. Often it is simply referred to as "the change in length with respect to the original length" which the body experiences.

To study the nature of strain, consider a body in an unstrained configuration at time $t = 0$ with the coordinate system a_i . The body is then allowed to deform to a new configuration at some time t with the coordinate system x_i (see Figure 2.6). In addition, consider two neighboring particles P and Q within the body. In the initial configuration, the position vectors of P and Q are (a_1, a_2, a_3) and $(a_1 + da_1, a_2 + da_2, a_3 + da_3)$, respectively. After deformation, these points become P' and Q' with position vectors (x_1, x_2, x_3) and $(x_1 + dx_1, x_2 + dx_2, x_3 + dx_3)$, respectively. It is assumed that during the motion, the body obeys the previously stated laws of deformation (see Section 2.3) and thus the use of a continuous function to describe the motion is valid.

To determine the strain attained during the motion, it is necessary to determine the change in the length of a line element connecting P and Q . This is easily calculated by finding the difference in the squares of the lengths dS_o and dS , defined in Figure 2.6, the squares being used merely to simplify the mathematical manipulations that follow. The terms for the lengths can be expressed as

$$dS_o^2 = da_i da_i \quad (2.7 - 1)$$

$$dS^2 = dx_i dx_i \quad (2.7 - 2)$$

Thus the difference in the lengths becomes

$$dS^2 - dS_o^2 = dx_i dx_i - da_i da_i \quad (2.7 - 3)$$

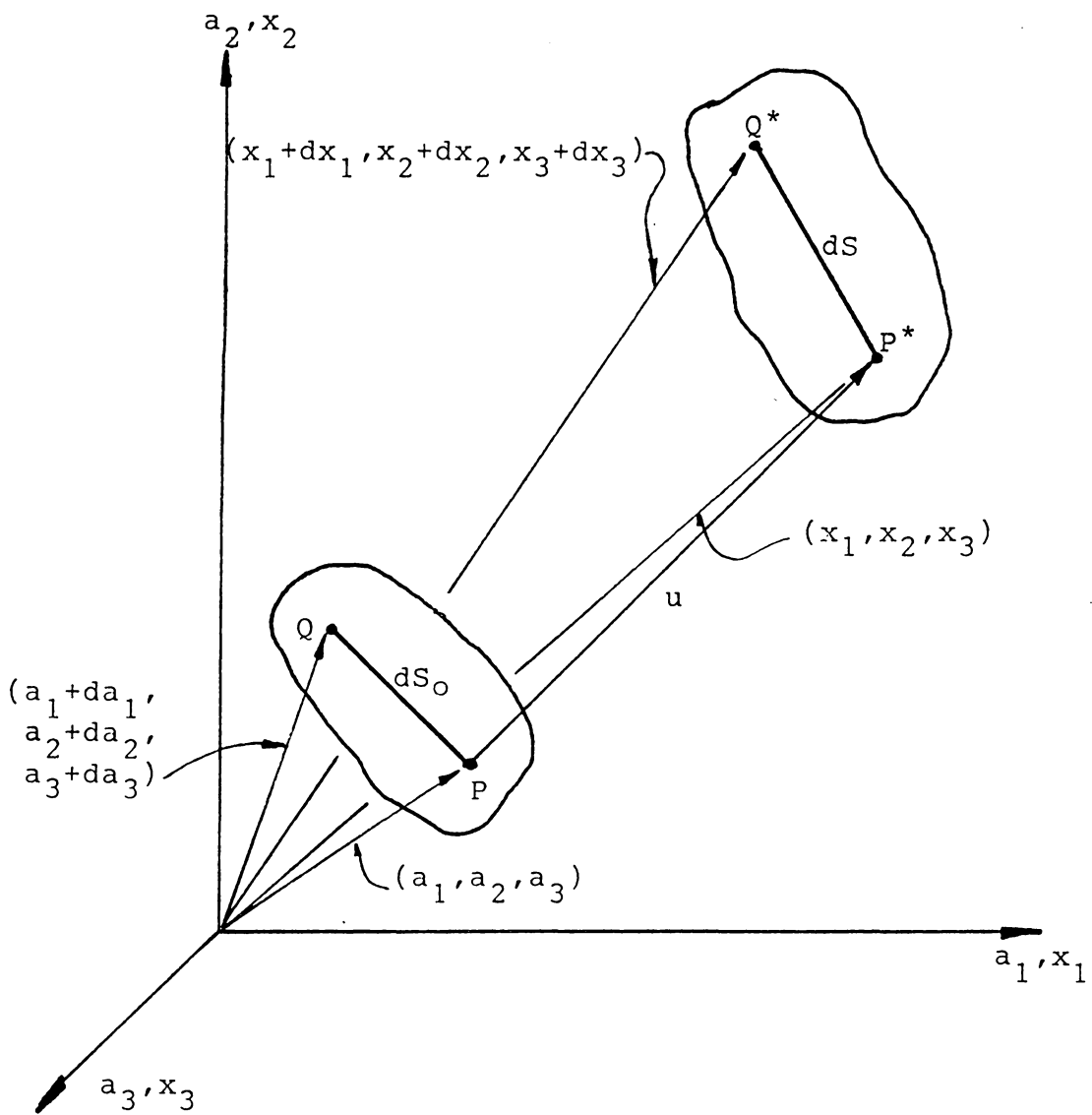


Figure 2.6: Strain In a Body

where, by choice of sign convention, tensile strain is positive. In addition, it is assumed in Eqs 2.7-1 and 2.7-3 that the coordinate systems are rectangular and cartesian which eliminates complex coordinate transformations [14,17,30].

Green-Lagrangian Strain Tensor

To proceed from Eq.2.7-3 it is necessary to consider the choice of reference frame to be used, Lagrangian or Eulerian (see Section 2.3). If the Lagrangian approach is used, it is necessary to express the strain in terms of the initial configuration. This is accomplished by expressing the coordinates in the deformed configuration in terms of the coordinates a_i and the components of the displacement vector u (see Figure 2.6), which is assumed to be a function of a_i , that is

$$x_i = a_i + u_i(a_i) \quad (2.7 - 4)$$

Using Eq. 2.7-4, the differential of x can be found as

$$dx_i = \frac{\partial x_i}{\partial a_j} = \left[\delta_{ri} + \frac{\partial u_r}{\partial a_i} \right] da_i da_j \quad (2.7 - 5)$$

Substituting this expression into Eq.2.7-3, the strain equation becomes

$$dS^2 - dS_o^2 = \left[\left(\delta_{ri} + \frac{\partial u_r}{\partial a_i} \right) \left(\delta_{rj} + \frac{\partial u_r}{\partial a_j} \right) - \delta_{ij} \right] da_i da_j$$

or

$$dS^2 - dS_o^2 = \left[\frac{\partial u_i}{\partial a_j} + \frac{\partial u_j}{\partial a_i} + \frac{\partial u_r}{\partial a_i} \frac{\partial u_r}{\partial a_j} \right] da_i da_j \quad (2.7 - 6)$$

Using Eq. 2.7-5, the Green-Lagrangian strain tensor is defined as

$$\epsilon_{ij} = \frac{1}{2} \left[\frac{\partial u_i}{\partial a_j} + \frac{\partial u_j}{\partial a_i} + \frac{\partial u_r}{\partial a_i} \frac{\partial u_r}{\partial a_j} \right] \quad (2.7 - 7)$$

so that

$$dS^2 - dS_o^2 = 2\epsilon_{ij} da_i da_j \quad (2.7 - 8)$$

Almansi Strain

If the Eulerian approach is chosen, the description of strain is based on the deformed configuration. The motion thus becomes a function of the coordinates x_i and the displacement vector u . In this case, u is assumed to be a function of x_i , and is written as

$$a_i = x_i - u_i(x_j) \quad (2.7 - 9)$$

The differential of a_i then becomes

$$da_i = \frac{\partial_i}{\partial x_j} dx_j = \left[\delta r_i - \frac{\partial u_r}{\partial x_i} \right] dx_i \quad (2.7 - 10)$$

Substitution of Eq. 2.7-10 into Eq. 2.7-3 leads to

$$dS^2 - dS_o^2 = \left[\delta_{ij} - \left(\delta r_i - \frac{\partial u_r}{\partial x_i} \right) \left(\delta r_j - \frac{\partial u_r}{\partial x_j} \right) \right] dx_i dx_j$$

or

$$dS^2 - dS_o^2 = \left[\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j} - \frac{\partial u_r}{\partial x_i} \frac{\partial u_r}{\partial x_j} \right] dx_i dx_j \quad (2.7 - 11)$$

From Eq. 2.7-11, the Almansi Strain Tensor is defined as

$$e_{ij} = \frac{1}{2} \left[\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j} - \frac{\partial u_r}{\partial x_i} \frac{\partial u_r}{\partial x_j} \right] \quad (2.7 - 12)$$

so that

$$dS^2 - dS_o^2 = 2e_{ij} dx_i dx_j \quad (2.7 - 13)$$

2.8 Constitutive Equations

In the previous sections, the mathematical relationships governing stresses and strains as a result of deformation and loading have been established. In order to completely describe the equilibrium conditions of the body, a relationship between stress and strain must be found. This relationship can be developed from kinetic considerations [14,30] and is commonly called a

constitutive equation. Constitutive equations relate the material behavior to the loading and thus provide the desired link between stress and strain. Formulation of this link can become quite complex, however, when the entire range of loading possibilities and material behavior are considered [30]. To overcome this difficulty, the material is normally assumed to behave in an ideal way, consistent with the nature of the problem to be analyzed and the results desired [2,30].

Though there are a variety of formulations which may be followed, the most common assumption is to consider the response of the material to be linear in nature. This response, referred to as ideally elastic or Hookean, takes the form (for a uniaxial stress state)

$$\tau_{ii} = E \varepsilon_{ii} \quad (2.8 - 1)$$

where E is known as Hooke's constant or Young's modulus of elasticity [5,14,30]. Expanding Eq. 2.8-1 to contain the effects of shearing stress leads to the general form

$$\tau_{ij} = C_{ijrs} \varepsilon_{rs} \quad (2.8 - 2)$$

In Eq. 2.8-2, C_{ijrs} represents 81 coefficients of the matrix C , called the constitutive matrix [2,14,17,30]. Thus, in matrix form, Eq. 2.8-2 becomes

$$\tau = C \varepsilon \quad (2.8 - 3)$$

As mentioned in the derivations of τ and ε , both of these matrices are symmetric and therefore the matrix C must also be symmetric, reducing the independent coefficients of C to 36. This number may be reduced further depending on the exact nature of the material considered [2,14,30]. For this study, the material is assumed to be linearly elastic and isotropic which leads to only two independent coefficients for the case of a three dimensional body. Later, the reduction of models to one dimensional, uniaxial loading, further simplifies C , leaving Eq. 2.8-1.

Chapter 3

FINITE ELEMENT FORMULATION

3.1 Introduction

In Chapter 2, the fundamental concepts and equations defining the deformation of a continuum were discussed. This chapter employs these concepts in the displacement based finite element formulation for nonlinear analysis of the continuum.

In general, a nonlinear problem arises if the stiffness and/ or load vector become deformation dependent during loading. Essentially there are three forms of nonlinear effects which result in this dependency. The first form, geometric nonlinearity, is probably the most common of the three. This effect is characterized by such properties as large displacements, large rotations, and stress-strain relations which may or may not be linear. The second type, material nonlinearity, is a result of a material which expresses nonlinear stress-strain behavior during all, or a portion, of loading. The third type of nonlinearity is a result of changing boundary conditions during deformation. The most common form of this effect is a contact problem [2].

As will be discussed in more detail in Chapter 6, the major contribution to the nonlinear behavior of cable structures is from the change in geometry during loading. Although cables do exhibit material nonlinearity, the effect is far overshadowed by the change in cable profile. As a result, the material effects are usually ignored in analysis, assuming a linear stress-strain relationship. This assumption is also adhered to in the work presented here. Thus, attention will be focused only on geometric nonlinearities associated with the analysis.

3.2 Notation

In addition to the tensor notation presented in Chapter 2, the problem of tracing the body as it moves from one configuration to the next requires that each quantity be referenced to (measured with respect to) a known configuration. In general, the motion of the body is broken up into three

separate states (see Figure 3.1), its initial configuration at time $t = 0$, the final equilibrium configuration at time $t + \Delta t$, and an intermediate state at time t . The variable t is used here to distinguish between the different configurations and not necessarily as a strict reference to time.

Consider again the point P within the body. Previously the point P was located simply by the coordinates x_i , where i ranged from 1 to 3, and it was understood in which configuration the coordinates were measured. Now, however, an additional index must be used in order to completely describe not only the location of P but also which configuration is being considered [2]. Thus the coordinates are expressed as

$${}^\tau x_i \quad i = 1, 2, 3 \quad (3.2 - 1)$$

where τ takes on the values 0, t , and $t + \Delta t$. Similarly, displacements of P can be expressed as

$$\begin{aligned} & {}^\tau u_i \quad i = 1, 2, 3 \\ & \tau = 0, t, t + \Delta t \end{aligned} \quad (3.2 - 2)$$

and the volume, surface area, and mass density of the body are expressed as

$$\begin{aligned} & {}^\tau V \\ & {}^\tau S \quad \tau = 0, t, t + \Delta t \\ & {}^\tau \rho \end{aligned} \quad (3.2 - 3)$$

In addition, it is useful to define the incremental displacement between configurations as

$$u_i = {}^{t+\Delta t}u_i - {}^t u_i \quad (3.2 - 4)$$

Using Eqs. 3.2-1, 3.2-2, and 3.2-4, three extensively used relationships can be derived as

$$\begin{aligned} {}^t x_i &= {}^0 x_i + {}^t u_i \\ {}^{t+\Delta t} x_i &= {}^0 x_i + {}^{t+\Delta t} u_i \\ {}^{t+\Delta t} x_i &= {}^t x_i + u_i \end{aligned} \quad (3.2 - 5)$$

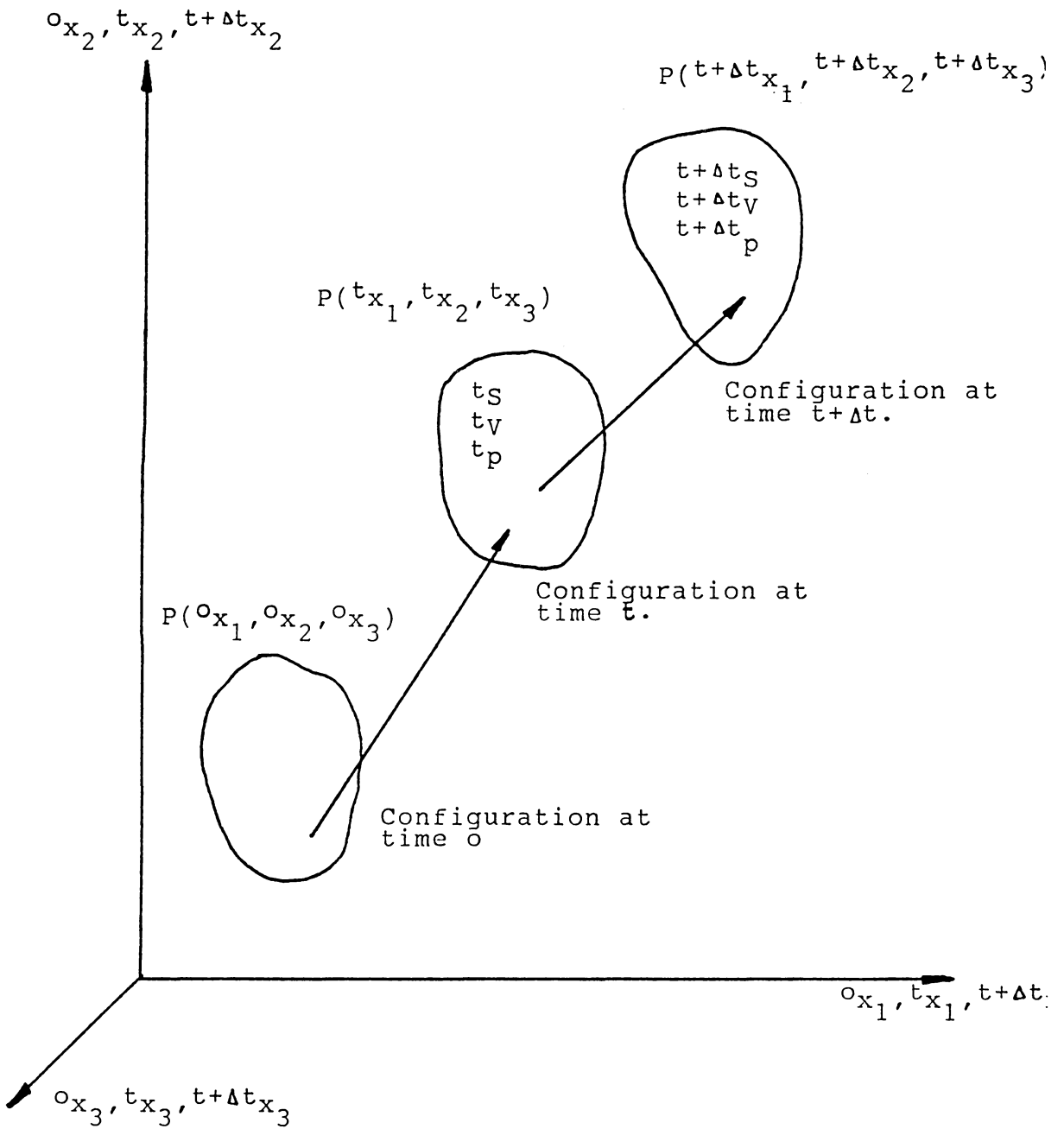


Figure 3.1: Incremental Deformation of a Continuum

As noted in Chapter 2, the expressions for stress and strain in the current state are, in general, developed with reference to a previously defined configuration. For this type of quantity, an additional left subscript is employed to signify the state in which the quantity is measured. As an example, consider the Second Piola-Kirchhoff stress tensor

$${}^t S_{ij}$$

Here, the left superscript denotes the configuration on which the tensor is considered to act, while the left subscript denotes the configuration on which the tensor is measured. Similarly, the Green-Lagrangian strain tensor at time $t + \Delta t$ as measured with respect to the state at time $t = 0$ is given by

$${}^{t+\Delta t} {}_0 \varepsilon_{ij}$$

An exception to this rule occurs when the quantity considered refers to the same configuration on which it acts. For such terms, the left subscript is dropped. Thus, the Cauchy stress, which is always measured in the current configuration, is expressed as

$${}^{t+\Delta t} \tau_{ij} \text{ rather than } {}_{t+\Delta t} {}^{t+\Delta t} \tau_{ij}$$

Throughout the formulation, derivatives of displacements and coordinates are required. To simplify these expressions, a comma separating the right-hand subscripts is used to denote partial differentiation with respect to the coordinate represented by the rightmost of these subscripts. In addition, the left superscript designates the configuration in which the quantity is taken and the left subscript denotes the configuration in which the coordinate is measured. As an example

$${}^{t+\Delta t} {}_0 u_{i,j} = \frac{\partial^{t+\Delta t} u_i}{\partial {}^0 x_j}$$

and

$${}^{t+\Delta t} {}^t x_{i,j} = \frac{\partial^{t+\Delta t} x_i}{\partial {}^t x_j}$$

3.3 Incremental Formulation

The principle of virtual work is a well known method of determining the conditions of equilibrium for a body under a given loading. The method seeks to express the equilibrium of internal forces and external forces solely in terms of displacements of the continuum, consistent with the nature of the loading and the boundary conditions [2,17,48]. Applying virtual work to the deformed configuration at time $t + \Delta t$, the equation of equilibrium of the body becomes [2]

$$\int_{t+\Delta t_V} {}^{t+\Delta t}\tau_{ij} \delta {}^{t+\Delta t}e_{ij} {}^{t+\Delta t}dV = {}^{t+\Delta t}R \quad (3.3 - 1)$$

where ${}^{t+\Delta t}\tau_{ij}$ and ${}^{t+\Delta t}e_{ij}$ are the Cauchy stress tensor and infinitesimal strain tensor discussed in Chapter 2. The left hand side of Eq. 3.3-1 represents the internal work of the system caused by the virtual displacements. It is important to note here that these quantities are expressed in terms of the as yet unknown configuration at $t + \Delta t$.

The term ${}^{t+\Delta t}R$ represents the external work done by the loads acting on the body. In expanded form, this term becomes

$${}^{t+\Delta t}R = \int_{t+\Delta t_V} {}^{t+\Delta t}f_i^B \delta u_i {}^{t+\Delta t}dV + \int_{t+\Delta t_S} {}^{t+\Delta t}f_i^S \delta u_i^S {}^{t+\Delta t}dS \quad (3.3 - 2)$$

where ${}^{t+\Delta t}f_i^B$ and ${}^{t+\Delta t}f_i^S$ represent the body forces and surface tractions, respectively, in the i direction. The quantity u_i represents the i component of the virtual displacement, and the superscript S applied to δu_i in the second term simply denotes that portion of displacement which is experienced over the surface of the body. Here again, the forces and integrals are expressed in terms of the unknown configuration at $t + \Delta t$.

The lack of information defining the equilibrium configuration a priori requires the use of an iterative procedure to attain the desired state through a trial and error approach. The procedure involves linearizing the equation of equilibrium about a known trial configuration and solving this equation for the next trial configuration. The solution is thus obtained by tracing the load-displacement curve in a piecewise linear fashion until equilibrium is reached. This method requires that the equation of equilibrium be expressed in an incremental form, basing the next point on the

curve on some previously defined state. As discussed in Chapter 2, the reference method most commonly used in solid mechanics is the Lagrangian description. This approach can be further divided into two separate formulations, the Total Lagrangian (TL) formulation and the Updated Lagrangian (UL) formulation, to be discussed next.

Total Lagrangian Formulation

The TL formulation seeks to trace the deformation of the continuum by referencing this motion to the original configuration at time $t = 0$. Thus, Eq. 3.3-1 becomes

$$\int_{\circ V} {}^{t+\Delta t} \circ S_{ij} \delta {}^{t+\Delta t} \circ \varepsilon_{ij} \circ dV = {}^{t+\Delta t} R \quad (3.3 - 3)$$

where the use of the Second Piola-Kirchhoff stress tensor, ${}^{t+\Delta t} \circ S_{ij}$, and Green-Lagrangian strain tensor, ${}^{t+\Delta t} \circ \varepsilon_{ij}$, result from the use of a reference configuration other than the current configuration.

The external work ${}^{t+\Delta t} R$ must also be modified to account for the use of a reference configuration. Assuming that the loading does not depend on the deformation of the body, the external virtual work becomes

$${}^{t+\Delta t} R = \int_{\circ V} {}^{t+\Delta t} \circ f_i^B \delta u_i \circ dV + \int_{\circ S} {}^{t+\Delta t} \circ f_i^S \delta u_i^S \circ dS \quad (3.3 - 4)$$

where ${}^{t+\Delta t} \circ f_i^B$ and ${}^{t+\Delta t} \circ f_i^S$ represent the body forces per unit volume and surface tractions per unit area acting on the body at $t + \Delta t$ measured with respect to the state $t = 0$. The use of loads which depend on the configuration of the structure can be handled by assuming the load at $t + \Delta t$ to be approximated as [2]

$${}^{t+\Delta t} R = \int_{t+\Delta t, V^{(k-1)}} {}^{t+\Delta t} \circ f_i^B \delta u_i \circ dV^{(k-1)} + \int_{\circ S^{(k-1)}} {}^{t+\Delta t} \circ f_i^S \delta u_i^S \circ dS^{(k-1)} \quad (3.3 - 5)$$

where $(k - 1)$ represents the volume and surface area corresponding to the $(k - 1)$ iteration. Recalling from Section 3.2 that the displacement of the body may be expressed incrementally, the stress and strain tensors are also expressed as

$${}^{t+\Delta t} \circ S_{ij} = {}^t S_{ij} + \circ S_{ij} \quad (3.3 - 6)$$

$${}^{t+\Delta t}{}_{\sigma}\varepsilon_{ij} = {}^t{}_{\sigma}\varepsilon_{ij} + {}_{\sigma}\varepsilon_{ij} \quad (3.3 - 7)$$

where ${}_{\sigma}S_{ij}$ and ${}_{\sigma}\varepsilon_{ij}$ are the incremental stress and strain tensors, respectively, measured with reference to the initial configuration. From Chapter 2, the expression for strain at time t and $t + \Delta t$ can be written as

$${}^{t+\Delta t}{}_{\sigma}\varepsilon_{ij} = \frac{1}{2}({}^{t+\Delta t}{}_{\sigma}u_{i,j} + {}^{t+\Delta t}{}_{\sigma}u_{j,i} + {}^{t+\Delta t}{}_{\sigma}u_{k,i} {}^{t+\Delta t}{}_{\sigma}u_{k,j}) \quad (3.3 - 8)$$

$${}^t{}_{\sigma}\varepsilon_{ij} = \frac{1}{2}({}^t{}_{\sigma}u_{i,j} + {}^t{}_{\sigma}u_{j,i} + {}^t{}_{\sigma}u_{k,i} {}^t{}_{\sigma}u_{k,j}) \quad (3.3 - 9)$$

and, using Eq. 3.2-4, the incremental strain tensor becomes

$${}_{\sigma}\varepsilon_{ij} = {}_{\sigma}e_{ij} + {}_{\sigma}\eta_{ij} \quad (3.3 - 10)$$

The terms ${}_{\sigma}e_{ij}$ and ${}_{\sigma}\eta_{ij}$ represent the linear and nonlinear portions, respectively, of the incremental strain tensor, defined as

$${}_{\sigma}e_{ij} = \frac{1}{2}({}_{\sigma}u_{i,j} + {}_{\sigma}u_{j,i} + {}^t{}_{\sigma}u_{k,i} {}^t{}_{\sigma}u_{k,j} + {}^t{}_{\sigma}u_{k,i} {}^t{}_{\sigma}u_{k,i}) \quad (3.3 - 11)$$

$${}_{\sigma}\eta_{ij} = \frac{1}{2}{}_{\sigma}u_{k,i} {}_{\sigma}u_{k,j} \quad (3.3 - 12)$$

The first variations of the strain tensors thus become

$$\delta {}^{t+\Delta t}{}_{\sigma}\varepsilon_{ij} = \delta {}_{\sigma}\varepsilon_{ij} \quad (3.3 - 13)$$

$$\delta {}_{\sigma}\varepsilon_{ij} = \delta {}_{\sigma}e_{ij} \quad (3.3 - 14)$$

Substituting Eqs. 3.3-12 and 3.3-13 into Eq. 3.3-3, and subtracting the external work at time t , the incremental equation of equilibrium for the TL description becomes

$$\int_{\sigma_V} {}_{\sigma}S_{ij} \delta {}_{\sigma}\varepsilon_{ij} {}^{\circ}dV + \int_{\sigma_V} {}^t{}_{\sigma}S_{ij} \delta {}_{\sigma}\eta_{ij} {}^{\circ}dV = {}^{t+\Delta t}R - {}^tR \quad (3.3 - 15)$$

In order to obtain an approximate solution to Eq. 3.3-15, the equation is linearized assuming the incremental Second Piola-Kirchhoff stress to be linearly related to the Green-Lagrangian strain

$${}^o\mathcal{S}_{ij} = C_{ijmn} {}^o\epsilon_{mn} \quad (3.3 - 16)$$

where C_{ijmn} represents the coefficients of the constitutive matrix (see Section 2.8).

In addition, since the incremental strains are nonlinear in displacement, it must be assumed that the nonlinear portion can be neglected in order to completely linearize Eq. 3.4-15 [31]. Thus Eq. 3.3-10 can be written as

$${}^o\epsilon_{ij} = {}^oe_{ij} \quad (3.3 - 17)$$

which gives

$${}^o\mathcal{S}_{ij} = C_{ijmn} {}^oe_{mn} \quad (3.3 - 18)$$

Substitution of Eq. 3.3-18 into Eq. 3.3-15 gives

$$\int_{{}^oV} C_{ijmn} {}^oe_{mn} \delta {}^oe_{ij} {}^odV + \int_{{}^oV} {}^t\mathcal{S}_{ij} \delta {}^o\eta_{ij} {}^odV = {}^{t+\Delta t}R - {}^tR \quad (3.3 - 19)$$

As a result of linearization, the external virtual work term at time t does not exactly equal the internal work at time t . To compensate for this discrepancy, a residual work term

$$\delta {}^tW_{res} = {}^tR - \int_{{}^oV} {}^t\mathcal{S}_{ij} \delta {}^oe_{ij} {}^odV \quad (3.3 - 20)$$

is added to Eq. 3.3-19 [31]. The final form of the linearized incremental equilibrium equation becomes

$$\int_{{}^oV} C_{ijmn} {}^oe_{mn} \delta {}^oe_{ij} {}^odV + \int_{{}^oV} {}^t\mathcal{S}_{ij} \delta {}^o\eta_{ij} {}^odV = {}^{t+\Delta t}R - \int_{{}^oV} {}^t\mathcal{S}_{ij} \delta {}^oe_{ij} {}^odV \quad (3.3 - 21)$$

Updated Lagrangian

The UL formulation is essentially the same as the TL formulation in implementation; how-

ever, in the solution procedure, the configuration at time t is used as the reference state. Eq. 3.3-1 thus becomes

$$\int_V {}^{t+\Delta t}{}_t S_{ij} \delta {}^{t+\Delta t}{}_t \varepsilon_{ij} {}^t dV = {}^{t+\Delta t}R \quad (3.3 - 22)$$

Analogous to the TL formulation, the second Piola-Kirchhoff stress tensor in the UL approach is approximated as

$${}^{t+\Delta t}{}_t S_{ij} = {}^t S_{ij} + {}_t S_{ij} \quad (3.3 - 23)$$

However, recalling from the derivation of the Second Piola-Kirchhoff stress tensor in Chapter 2, the stress at time t referred to t is obviously the Cauchy stress at t . That is

$${}^t S_{ij} = {}^t \tau_{ij} \quad (3.3 - 24)$$

and therefore Eq. 3.3-23 becomes

$${}^{t+\Delta t}{}_t S_{ij} = {}^t \tau_{ij} + {}_t S_{ij} \quad (3.3 - 25)$$

Similarly, the Green-Lagrangian strain tensor becomes

$${}^{t+\Delta t}{}_t \varepsilon_{ij} = {}^t \varepsilon_{ij} \quad (3.3 - 26)$$

where

$${}^t \varepsilon_{ij} = {}^t e_{ij} + {}^t \eta_{ij} \quad (3.3 - 27)$$

$${}^t e_{ij} = \frac{1}{2}({}^t u_{i,j} + {}^t u_{j,i}) \quad (3.3 - 28)$$

$${}^t \eta_{ij} = \frac{1}{2}{}^t u_{i,j} {}^t u_{j,i} \quad (3.3 - 29)$$

and the variation of the strain becomes simply

$$\delta^{t+\Delta t} \epsilon_{ij} = \delta_t \epsilon_{ij} \quad (3.3 - 30)$$

To linearize the equilibrium equations, it is assumed again that the stress increment can be expressed as

$${}^t S_{ij} = C_{ijmn} {}^t e_{mn} \quad (3.3 - 31)$$

and the variation of strain can be approximated as

$$\delta_t \epsilon_{ij} = \delta_t e_{ij} \quad (3.3 - 32)$$

Substitution of Eq. 3.3-25, 3.3-27, 3.3-31, and 3.3-32 into Eq. 3.3-22, and making a similar correction to the external virtual work, the linearized incremental equilibrium equation in the UL formulation becomes

$$\int_{tV} C_{ijmn} {}^t e_{mn} \delta_t e_{ij} {}^t dV + \int_{tV} {}^t \tau_{ij} \delta_t \eta_{ij} {}^t dV = {}^{t+\Delta t} R - \int_{tV} {}^t \tau_{ij} \delta_t e_{ij} {}^t dV \quad (3.3 - 33)$$

As seen by comparing Eqs. 3.3-21 and 3.3-33, both TL and UL approaches incorporate the same nonlinear effects with the same basic assumptions. The choice between the two becomes more one of preference, although Bathe [2] suggests that the TL approach may involve more complex manipulations. For this study, the UL approach was used exclusively, although it can be shown that for the assumptions to be made (see Chapter 4), the UL and TL approach become identical [2].

3.4 Finite Element Discretization

This section presents the finite element discretization of the continuum in a general manner, giving the equations for a two dimensional plane strain (plane stress) element. It is the intent here to give an overall indication of the methods involved with the discretization process.

Starting with the linear equilibrium equation derived previously for the UL approach, Eq. 3.3-33, the continuum is now discretized into an assemblage of n discrete finite elements. In so doing, the integrals over the volume and surface area of the continuum are transformed into a sum

of integrals over each element. The complete model of the structure can then be defined by evaluating the characteristics of each element individually and applying the standard techniques of assembly to form the system model [2,10,48].

To begin, let the element be defined by an isoparametric representation. The use of isoparametric discretization allows the geometry and displacements of the element to be described using the same interpolation functions [2,17]. Thus the displacements and coordinates within the element are expressed in terms of the nodal coordinates and displacements as

$$\begin{aligned}
 {}^\tau x_i &= \sum_{k=1}^n N_k {}^\tau x_i^k \\
 {}^\tau u_i &= \sum_{k=1}^n N_k {}^\tau u_i^k \\
 u_i &= \sum_{k=1}^n N_k u_i^k
 \end{aligned}
 \quad \begin{aligned}
 i &= 1, 2 \\
 \tau &= 0, t, t + \Delta t
 \end{aligned}
 \quad (3.4 - 1)$$

where

${}^\tau x_i$ = the coordinate in the i direction
of a point within the element at time τ ,

${}^\tau x_i^k$ = the coordinate of node k in the
 i direction at time τ ,

N_k = the Lagrangian interpolation function
of node k , dependant on the type of
element used.

Similarly

${}^\tau u_i, u_i$ = the total and incremental displacements
in the i direction of a point in the element,

${}^\tau u_i^k, u_i^k$ = the total and incremental displacements
of node k in the direction i .

Recalling the equations for infinitesimal strain and nonlinear strain, Eqs. 3.3-28 and 3.3-29, repeated here

$${}^t e_{ij} = \frac{1}{2}({}^t u_{i,j} + {}^t u_{j,i})$$

$${}^t \eta_{ij} = \frac{1}{2}{}^t u_{i,j} {}^t u_{j,i}$$

and using Eq. 3.4-3, these equations can now be written in matrix form. Starting with Eq. 3.3-28,

$${}^t e = {}^t B_L u \quad (3.4 - 2)$$

where

$${}^t e = [{}^t e_{11} \mid {}^t e_{22} \mid 2 {}^t e_{12}]$$

$${}^t B_L = [{}^t B_L^k]$$

$${}^t B_L^k = \begin{bmatrix} N_{k,1} & 0 \\ 0 & N_{k,2} \\ N_{k,2} & N_{k,1} \end{bmatrix} \quad k = 1, 2 \dots n$$

$$u^T = [u_1^1 \ u_2^1 \ u_1^2 \ u_2^2 \ \dots \ u_1^n \ u_2^n]$$

and the variation of the strain becomes

$$\delta {}^t e = \delta ({}^t B_L u) = {}^t B_L \delta u \quad (3.4 - 3)$$

Substitution of Eq.3.4-3 into Eq. 3.3-29 leads to

$$\delta {}^t \eta_{ij} = \frac{1}{2}(\sum N_{k,i} u_i^k)(\sum N_{k,j} u_j^k) \quad (3.4 - 4)$$

or, in matrix form,

$$\delta {}^t \eta_{ij} = \frac{1}{2} u^T {}^t B_{NL}^T {}^t B_{NL} u \quad (3.4 - 5)$$

where

$${}^tB_{NL} = \begin{bmatrix} N_{k,i} & 0 \\ 0 & N_{k,i} \end{bmatrix} \quad k = 1, 2 \dots n$$

The variation of nonlinear strain thus becomes

$$\delta_t \eta = \delta u^T {}^tB_{NL}^T {}^tB_{NL} u \quad (3.4 - 6)$$

Substitution of Eqs. 3.4-5 and 3.4-8 into 3.3-33 results in the finite element form of the incremental equilibrium equation

$$({}^tK_L + {}^tK_{NL}) u = {}^{t+\Delta t}R - {}^tF \quad (3.4 - 7)$$

or

$${}^tK_T u = {}^{t+\Delta t}R - {}^tF \quad (3.4 - 8)$$

where

$${}^tK_T = {}^tK_L + {}^tK_{NL} \quad (3.4 - 9)$$

$${}^tK_L = \int_V {}^tB_L^T C {}^tB_L dV \quad (3.4 - 10)$$

$${}^tK_{NL} = \int_V {}^tB_{NL}^T {}^t\tau {}^tB_{NL} dV \quad (3.4 - 11)$$

$${}^{t+\Delta t}R = \int_{0_V} N^T {}^{t+\Delta t}f^B dV + \int_{0_S} N^T {}^{t+\Delta t}f^S dS \quad (3.4 - 12)$$

$${}^tF = \int_V {}^tB_N^T \hat{\tau} dV \quad (3.4 - 13)$$

$${}^t\tau = \begin{bmatrix} {}^t\tau_{11} & {}^t\tau_{12} & 0 & 0 \\ {}^t\tau_{21} & {}^t\tau_{22} & 0 & 0 \\ 0 & 0 & {}^t\tau_{11} & {}^t\tau_{21} \\ 0 & 0 & {}^t\tau_{21} & {}^t\tau_{22} \end{bmatrix} \quad (3.4 - 14)$$

$${}^t\tau^{ST} = [{}^t\tau_{11} \mid {}^t\tau_{22} \mid {}^t\tau_{12}] \quad (3.4 - 15)$$

The terms tK_L and ${}^tK_{NL}$ are the linear and nonlinear stiffness matrices for the system which together form the tangent stiffness matrix tK_T at the t configuration. The term tF is a vector of nodal forces which represents the effects of stress within the element under the current state of strain. The two variations in the form of the Cauchy stress matrix are a result of the matrix operations involved within each term.

Chapter 4

DEVELOPMENT OF CABLE ELEMENTS

4.1 Introduction

In the preceding chapters, the general incremental approach to geometrically nonlinear finite element analysis was presented in order to develop a clear understanding of the overall procedures involved in the finite element analysis of a nonlinear continuum. This chapter is devoted to specializing the equilibrium equations for use in the analysis of cable structures. First, the derivation of equations for a general n -node one dimensional element in the (x_1, x_2, x_3) space will be presented, followed by the development of the two elements used in this study: a two node linear element and a three node curvilinear element. In the following derivations, the cable is assumed to be characterized by:

1. Development of stress only in the direction normal to the plane of the cross-section.
2. A small area to span ratio, thus allowing for the stress to be considered as constant over the cross-section.
3. Negligible volumetric changes, allowing for the cross-section to be considered as constant.

The above assumptions are consistent with small strain theory which is used throughout this study. In addition, the updated Lagrangian formulation is used exclusively, although as noted by Bathe [2], either TL or UL formulation may be used in the development of cable elements with equal results.

4.2 Incremental Formulation for Cable Elements

The primary difference between the incremental equations for the cable elements and those of Section 3.4 is a result of the assumptions stated in Section 4.1. If only small strains are considered with no volumetric changes, the cross-section of the element can be considered as constant. This

effectively reduces the stress and strain equations to one dimensional equations in the longitudinal direction of the element. Thus the volume and area integrals associated with Eq. 3.4-9 reduce to integrals with respect to the length of the element [2,36].

To clarify this reduction to one independent variable, consider a general n-node element at time t , where t again takes on the values $0, t, t + \Delta t$ and $n = 2, 3, 4 \dots$ (see Figure 4.1). The expressions for the coordinates, total displacements, and incremental displacements, according to Section 3.4, are

$$\begin{aligned} {}^t x_j &= \sum N_i {}^t x_j^i \\ {}^t u_j &= \sum N_i {}^t u_j^i \\ u_j &= \sum N_i u_j^i \end{aligned} \quad \begin{aligned} i &= 1, 2, \dots, n \\ j &= 1, 2, 3 \end{aligned} \quad (4.2 - 1)$$

or, in matrix form,

$$\begin{aligned} {}^t x &= N {}^t u \\ {}^t u &= N {}^t u \\ u &= N u \end{aligned} \quad (4.2 - 2)$$

where

$$N = [N_1 I_3 \quad N_2 I_3 \quad \dots \quad N_n I_3] \quad (4.2 - 3)$$

$${}^t x^T = [{}^t x_1^1 \quad {}^t x_2^1 \quad {}^t x_3^1 \quad \dots \quad {}^t x_1^n \quad {}^t x_2^n \quad {}^t x_3^n] \quad (4.2 - 4)$$

$$u^T = [u_1^1 \quad u_2^1 \quad u_3^1 \quad \dots \quad u_1^n \quad u_2^n \quad u_3^n] \quad (4.2 - 5)$$

and

$$I_3 = \begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix}$$

where the variables were defined previously.

The incremental strain equations for the one dimensional element are expressed in a slightly different manner than in Section 3.4. In Chapter 2 the equations for the state of strain in a continuous medium were derived using the change in length of a differential element with respect to its original length (see Section 2.7). The strain equations were then expressed in terms of the partials of the displacements with respect to the global coordinates. For the cable element, it is more

convenient to express the strain equations in terms of partials with respect to the arc length of the element.

Following the same approach used in Chapter 2, the strain in the element at $t + \Delta t$ can be expressed as [17] (see Figure 4.2)

$$\varepsilon = \frac{1}{2} \frac{[({}^{t+\Delta t}ds)^2 - ({}^t ds)^2]}{({}^t ds)^2} \quad (4.2 - 5)$$

where the lengths of the differential element can be expressed as

$$({}^t ds)^2 = ({}^t dx_1)^2 + ({}^t dx_2)^2 + ({}^t dx_3)^2 \quad (4.2 - 6)$$

and

$$({}^{t+\Delta t} ds)^2 = ({}^{t+\Delta t} dx_1)^2 + ({}^{t+\Delta t} dx_2)^2 + ({}^{t+\Delta t} dx_3)^2 \quad (4.2 - 7)$$

From Eqs. 3.2-5, the expression

$${}^{t+\Delta t} dx_i = {}^t dx_i + du_i \quad (4.2 - 8)$$

combined with Eqs. 4.2-4, 4.2-5, 4.2-6, and 4.2-7, allows the strain to be expressed as

$$\varepsilon = \frac{1}{2({}^t ds)^2} [({}^t dx_1 + du_1)^2 + ({}^t dx_2 + du_2)^2 + ({}^t dx_3 + du_3)^2 - ({}^t dx_1)^2 - ({}^t dx_2)^2 - ({}^t dx_3)^2]$$

or

$$\varepsilon = \frac{{}^t dx_1}{{}^t ds} \frac{du_1}{{}^t ds} + \frac{{}^t dx_2}{{}^t ds} \frac{du_2}{{}^t ds} + \frac{{}^t dx_3}{{}^t ds} \frac{du_3}{{}^t ds} + \frac{1}{2} \left[\left(\frac{du_1}{{}^t ds} \right)^2 + \left(\frac{du_2}{{}^t ds} \right)^2 + \left(\frac{du_3}{{}^t ds} \right)^2 \right] \quad (4.2 - 9)$$

Separating Eq. 4.2-9 into linear and nonlinear components, the equation becomes

$${}^t \varepsilon_{11} = {}^t e_{11} + {}^t \eta_{11} \quad (4.2 - 10)$$

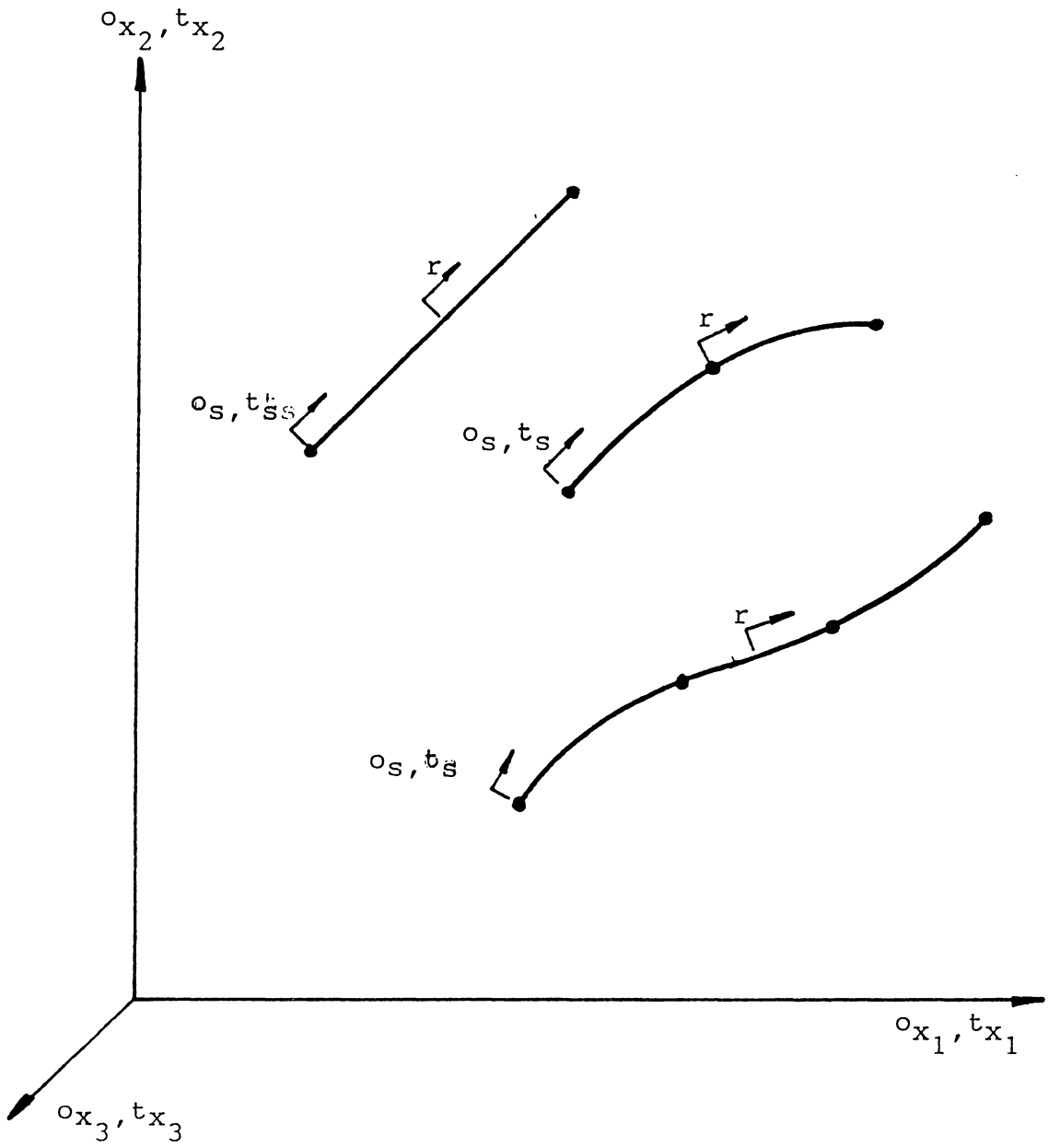


Figure 4.1: General Cable Elements

where

$${}^t e_{11} = \frac{{}^t dx_i}{{}^t ds} \frac{du_i}{{}^t ds} \quad (4.2 - 11)$$

and

$${}^t \eta_{11} = \frac{1}{2} \frac{du_i}{{}^t ds} \frac{du_i}{{}^t ds} \quad i = 1,2,3 \quad (4.2 - 12)$$

Eqs. 4.2-11 and 4.2-12 involve differentiation with respect to the arc length of the element and therefore require an expression for s . For an isoparametric representation, the length of the element is discretized in the same manner as displacements and coordinates [2,36], that is

$${}^t s = \sum N_i {}^t s^i \quad \begin{array}{l} i = 1, \dots, n \\ 0 \leq {}^t s \leq {}^t L \end{array} \quad (4.2 - 13)$$

where ${}^t s^i$ is the arc length associated with node i and ${}^t s$ is measured as shown in Figure 4.1 [2].

As a result of the use of interpolation functions expressed in terms of r (see Figure 4.1), to combine Eqs.4.2-2 into 4.2-11 and 4.2-12, a transformation of differentials is required. This is done using the Jacobian transformation [2,10,48]

$${}^t J = \frac{{}^t ds}{dr} \quad (4.2 - 14)$$

which in general depends on the element type used. Thus, Eq. 4.2-11 becomes, in matrix form,

$${}^t e_{11} = \frac{1}{{}^t J^2} [{}^t x^T N_r^T N_r u] \quad (4.2 - 15)$$

or

$${}^t e_{11} = {}^t B_L u \quad (4.2 - 16)$$

where

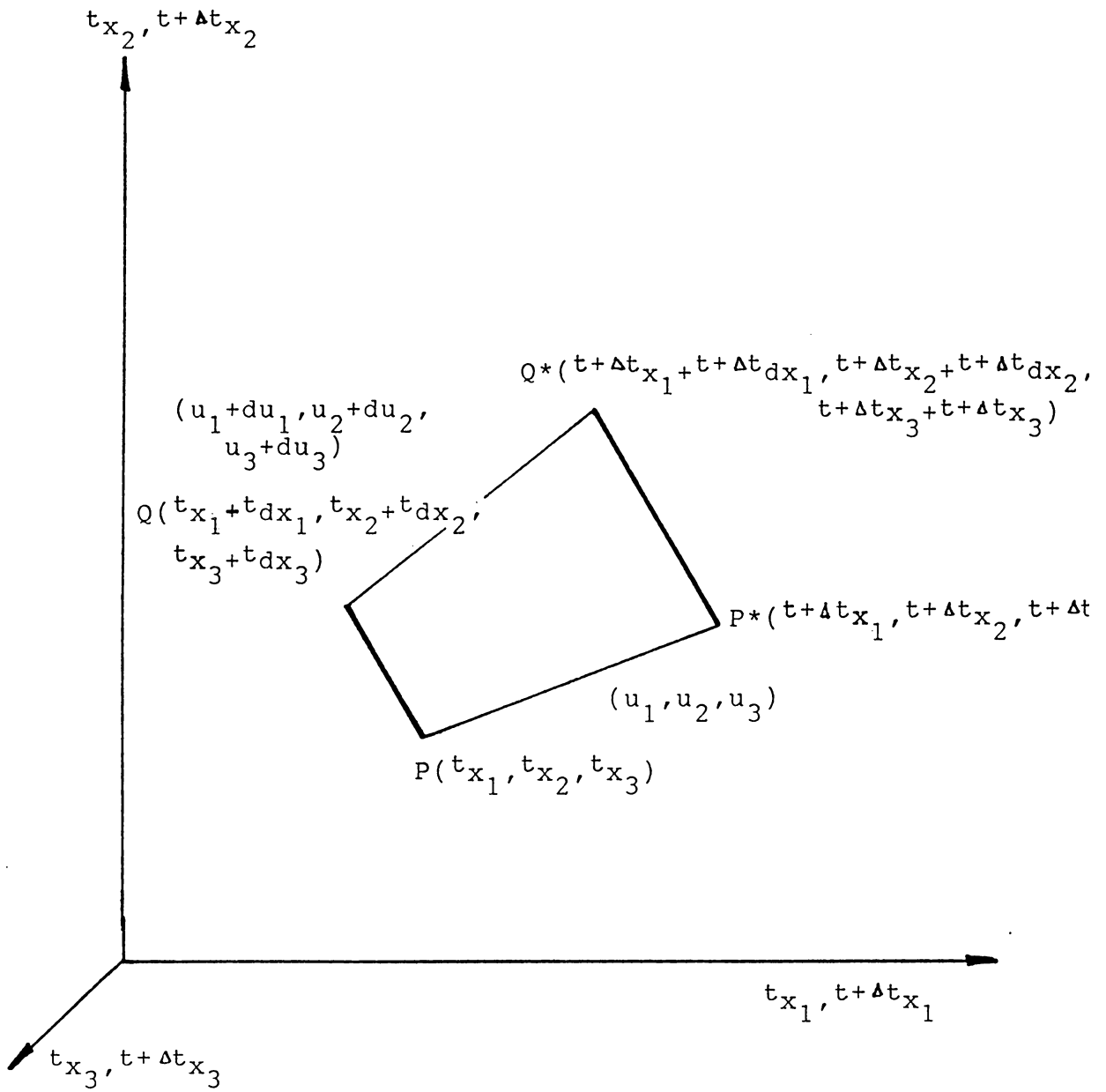


Figure 4.2: Strain In The Cable Element

$${}^tB_L = \frac{1}{{}^tj^2} {}^t x^T N_r^T N_r \quad (4.2 - 17)$$

$$N_r = \begin{bmatrix} N_{i,r} \\ N_{i,r} \\ N_{i,r} \end{bmatrix} \quad i = 1, 2 \dots n \quad (4.2 - 18)$$

Similarly, Eq. 4.2-12 becomes, in matrix form,

$${}^t\eta_{11} = \frac{1}{2} \frac{1}{{}^tj^2} u^T N_r^T N_r u$$

or

$${}^t\eta_{11} = \frac{1}{2} u^T {}^tB_{NL}^T {}^tB_{NL} u \quad (4.2 - 19)$$

where

$${}^tB_{NL} = \frac{1}{{}^tj^2} N_r \quad (4.2 - 20)$$

Finally, combining the equations above into Eq. 3.4-9, the general incremental equilibrium equation for a cable can be given as

$$({}^tK_L + {}^tK_{NL}) u = {}^{t+\Delta t}R - {}^tF \quad (4.2 - 21)$$

where

$${}^tK_L = \int_s {}^tB_L^T E {}^tB_L A {}^t ds \quad (4.2 - 22)$$

$${}^tK_{NL} = \int_s {}^tB_{NL}^T {}^t\tau {}^tB_{NL} A {}^t ds \quad (4.2 - 23)$$

$${}^tF = \int_s {}^tB_L^T {}^t\tau_{11} A {}^t ds \quad (4.2 - 24)$$

and

$${}^t\tau = \begin{bmatrix} {}^t\tau_{11} & 0 & 0 \\ 0 & {}^t\tau_{11} & 0 \\ 0 & 0 & {}^t\tau_{11} \end{bmatrix}$$

4.3 Linear Element

As mentioned previously, the linear element is used quite frequently in the analysis of cable

structures. The formulation for the element is fairly simple and thus provides a good model which is easy to implement. In addition, the element can substitute for a rigid strut in the analysis of structures such as cable beams [7,26].

The shape functions which are used to describe the element are obtained from Lagrangian interpolation [2,10,48] as:

$$\begin{aligned} N_1 &= \frac{1}{2}(1 - r) \\ N_2 &= \frac{1}{2}(1 + r) \quad -1 \leq r \leq +1 \end{aligned} \quad (4.3 - 1)$$

Combining these equations with Eqs. 4.2-3 and 4.2-18 results in

$$N = \frac{1}{2} \begin{bmatrix} 1 - r & 0 & 0 & 1 + r & 0 & 0 \\ 0 & 1 - r & 0 & 0 & 1 + r & 0 \\ 0 & 0 & 1 - r & 0 & 0 & 1 + r \end{bmatrix} \quad (4.3 - 2)$$

$$N_{,r} = \frac{1}{2} \begin{bmatrix} -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \end{bmatrix} \quad (4.3 - 3)$$

Recalling Eq. 4.2-13, the Jacobian transformation (Eq. 4.2-14) becomes

$${}^tJ = \frac{{}^t ds}{dr} = \frac{d}{dr} \left[\frac{1}{2} [(1 - r) | (1 + r)] \left[\frac{0}{{}^tL} \right] \right]$$

or

$${}^tJ = \frac{{}^tL}{2} \cong \frac{L}{2} \quad (4.3 - 4)$$

Combining Eqs. 4.3-4 and 4.3-3, the linear strain-displacement matrix, Eq 4.2-17, becomes

$${}^tB_L = \left(\frac{2}{L}\right)^2 \left(\frac{1}{4}\right) [{}^t x_1^1 \quad {}^t x_2^1 \quad {}^t x_3^1 \quad {}^t x_1^2 \quad {}^t x_2^2 \quad {}^t x_3^2] \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \end{bmatrix}$$

which can be simplified to give

$${}^tB_L = \frac{1}{L^2} [({}^t x_1^1 - {}^t x_1^2) | ({}^t x_2^1 - {}^t x_2^2) | ({}^t x_3^1 - {}^t x_3^2) | ({}^t x_1^2 - {}^t x_1^1) | ({}^t x_2^2 - {}^t x_2^1) | ({}^t x_3^2 - {}^t x_3^1)] \quad (4.3 - 5)$$

The expression for the nonlinear strain-displacement matrix, Eq. 4.2-21, is obtained directly from Eqs. 4.2-20, 4.3-3 and 4.3-4 as

$${}^tB_{NL} = \begin{bmatrix} -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \end{bmatrix} \quad (4.3 - 6)$$

Substitution of Eq. 4.3-5 into Eq. 4.2-23 results in the linear portion of the tangent stiffness matrix

$${}^tK_L = \frac{AE}{L^3} \begin{bmatrix} c_1^2 & c_1 c_2 & c_1 c_3 & -c_1^2 & -c_1 c_2 & -c_1 c_3 \\ & c_2^2 & c_2 c_3 & -c_1 c_2 & -c_2^2 & -c_2 c_3 \\ & & c_3^2 & -c_1 c_3 & -c_2 c_3 & -c_3^2 \\ & & & c_1^2 & c_1 c_2 & c_1 c_3 \\ & & & & c_2^2 & c_2 c_3 \\ \text{sym.} & & & & & c_3^2 \end{bmatrix} \quad (4.3 - 7)$$

where

$$\begin{aligned}
 c_1 &= ({}^t x_1^2 - {}^t x_1^1) \\
 c_2 &= ({}^t x_2^2 - {}^t x_2^1) \\
 c_3 &= ({}^t x_3^2 - {}^t x_3^1)
 \end{aligned}$$

Similarly, Eqs. 4.3-6 and 4.2-24 result in the nonlinear portion of the tangent stiffness matrix

$${}^t K_{NL} = \frac{{}^t P}{L} \begin{bmatrix} 1 & 0 & 0 & -1 & 0 & 0 \\ & 1 & 0 & 0 & -1 & 0 \\ & & 1 & 0 & 0 & -1 \\ & & & 1 & 0 & 0 \\ & & & & 1 & 0 \\ \text{sym.} & & & & & 1 \end{bmatrix} \quad (4.3 - 8)$$

Finally, using Eq. 4.3-5, the internal force vector ${}^t F$ becomes

$${}^t F = \frac{{}^t P}{L} \begin{bmatrix} -c_1 \\ -c_2 \\ -c_3 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} \quad (4.3 - 9)$$

where c_1 , c_2 , and c_3 are defined above.

4.4 Curvilinear Element

The three node curvilinear element is the most common alternative to the two node cable element. Although Section 4.2 gives the general formulation for an n node element, the use of more than three nodes usually results in more effort with little gain in accuracy over the three node element [36,24].

The three node element is again based on the use of Lagrangian interpolation [2,10] which in this case results in the functions

$$\begin{aligned}
N_1 &= \frac{1}{2}(r^2 - r) \\
N_2 &= (1 + r^2) \quad -1 \leq r \leq +1 \\
N_3 &= \frac{1}{2}(r^2 + r)
\end{aligned} \tag{4.4 - 1}$$

Eqs. 4.2-3 and 4.2-18 thus become

$$N = \frac{1}{2} \begin{bmatrix} r^2 - r & 0 & 0 & 2 - 2r^2 & 0 & 0 & r^2 + r & 0 & 0 \\ 0 & r^2 - r & 0 & 0 & 2 - 2r^2 & 0 & 0 & r^2 + r & 0 \\ 0 & 0 & r^2 - r & 0 & 0 & 2 - 2r^2 & 0 & 0 & r^2 + r \end{bmatrix} \tag{4.4 - 2}$$

$$N_{,r} = \frac{1}{2} \begin{bmatrix} 2r - 1 & 0 & 0 & -4r & 0 & 0 & 2r + 1 & 0 & 0 \\ 0 & 2r - 1 & 0 & 0 & -4r & 0 & 0 & 2r + 1 & 0 \\ 0 & 0 & 2r - 1 & 0 & 0 & -4r & 0 & 0 & 2r + 1 \end{bmatrix} \tag{4.4 - 3}$$

The Jacobian transformation can again be shown to equal $L/2$ provided the internal node represents the midpoint of the element [8,36]. Here, the length of the element is computed in the initial configuration as

$$L = \int_{-1}^{+1} [{}^0x^T N_{,r}^T N_{,r} {}^0x]^{1/2} dr \tag{4.4 - 4}$$

This has been assumed in implementation of the element in this study, although there is no strict control over the placement of the center node. The minor errors in specification of the midpoint coordinates are considered to be negligible in order to simplify modeling of initially curved shapes.

Combining Eqs. 4.4-3 and 4.2-17, the linear strain-displacement matrix becomes

$${}^tB_L = \frac{1}{l_j^2} x^T N_{,r}^T N_{,r}$$

or

$${}^tB_L = \frac{1}{L^2} [a(a {}^t x_i^1 + b {}^t x_i^2 + c {}^t x_i^3) | b(a {}^t x_i^1 + a {}^t x_i^2 + c {}^t x_i^3) | c(a {}^t x_i^1 + b {}^t x_i^2 + c {}^t x_i^3)] \tag{4.4 - 5}$$

where $i = 1,2,3$ and

$$\begin{aligned} a &= 2r - 1 \\ b &= -4r \\ c &= 2r + 1 \end{aligned} \quad (4.4 - 6)$$

Similarly, Eq. 4.2-21 becomes

$${}^tB_{NL} = \frac{1}{L} \begin{bmatrix} a & 0 & 0 & b & 0 & 0 & c & 0 & 0 \\ 0 & a & 0 & 0 & b & 0 & 0 & c & 0 \\ 0 & 0 & a & 0 & 0 & b & 0 & 0 & c \end{bmatrix} \quad (4.4 - 7)$$

with a , b , and c defined above.

Since the terms of Eqs. 4.4-5 and 4.4-7 contain the independent variable r , the explicit formulation for K_L and K_{NL} becomes somewhat more complex than for the two node element. For this reason, the tangent stiffness matrix is evaluated in the program using Gaussian quadrature [2,10,48].

4.5 Member Actions

In addition to the application of concentrated nodal forces and initial pretension forces, the program developed for the study (see Chapter 7) considers two general load types: the effect of dead weight per unit length of the cable and the effect of uniformly distributed loads. In both cases, the loads are considered to act only in the vertical (x_2) direction.

Recalling the expression for the external force vector at $t + \Delta t$, Eq. 3.4-12, and applying the assumptions of Section 4.1, the external nodal forces due to the applied loads can be found from

$${}^{t+\Delta t}R = \int_{0S} N^T {}^{t+\Delta t}f^0 dS \quad (4.5 - 1)$$

where ${}^{t+\Delta t}f$ represents the load acting on the element (either a body force or surface traction) at time $t + \Delta t$ measured on the initial configuration.

Dead Load

Using Eq. 4.5-1 to model the dead load of an element (member action type 1 in the program) results in, for the linear element,

$${}^{t+\Delta t}R = \frac{L}{2} \int_{-1}^{+1} \frac{1}{2} \begin{bmatrix} 1-r & 0 & 0 \\ 0 & 1-r & 0 \\ 0 & 0 & 1-r \\ 1+r & 0 & 0 \\ 0 & 1+r & 0 \\ 0 & 0 & 1+r \end{bmatrix} \begin{bmatrix} 0 \\ {}^{t+\Delta t}f_d \\ 0 \end{bmatrix} dr \quad (4.5 - 2)$$

where ${}^{t+\Delta t}f_d$ is the dead load per unit length of the element. Solving Eq. 4.5-2, the load vector becomes simply

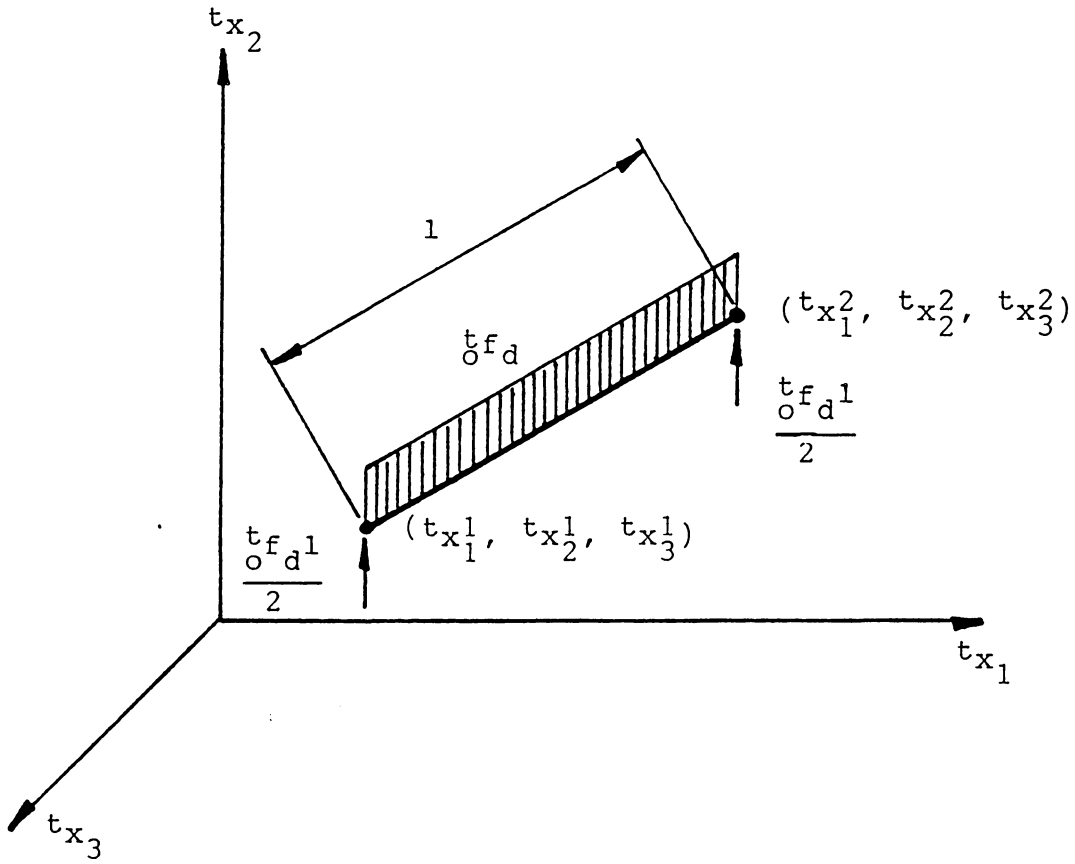
$${}^{t+\Delta t}R = \frac{1}{2} {}^{t+\Delta t}f_d L \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad (4.5 - 3)$$

This result is of course the same as would be generated by multiplying the element length by f_d and assigning equal portions to the two nodes (see Figure 4.3a).

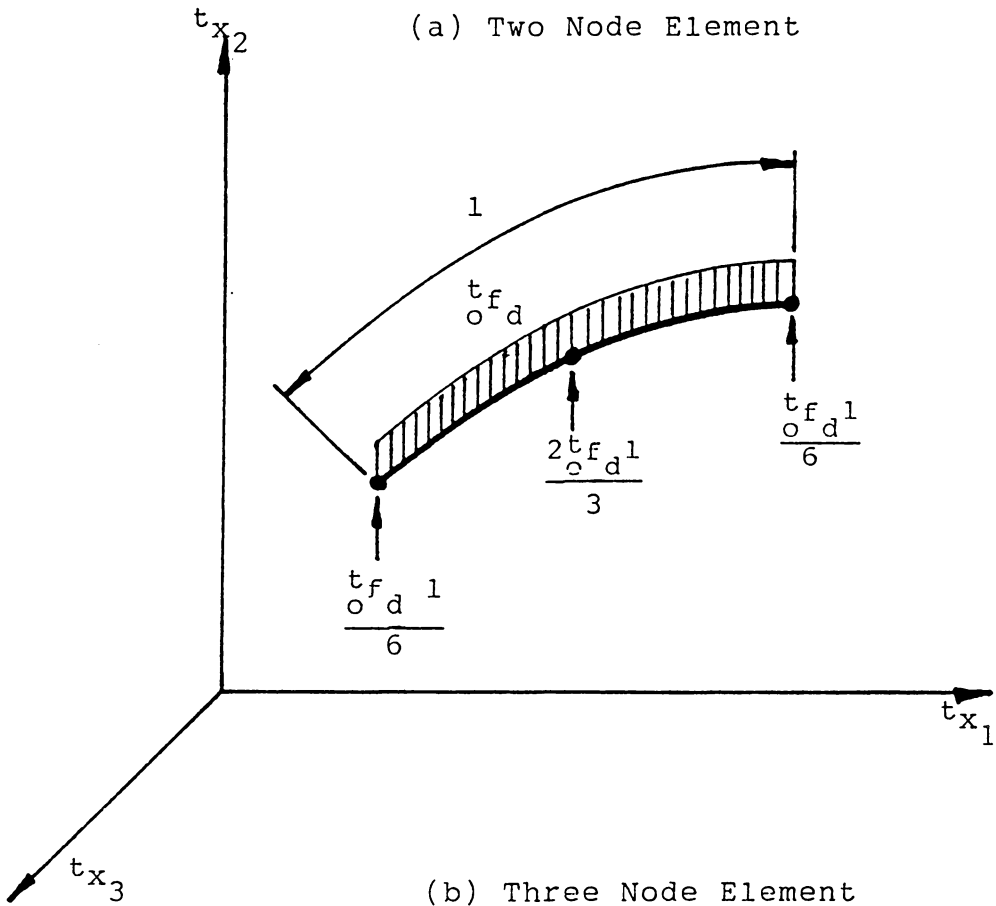
For the curvilinear element, a similar approach can be used as demonstrated by Cook [10] and Zienkiewicz [48] with the results as shown in Figure 4.3b.

Uniformly Distributed Load

For the uniformly distributed load (member action type 2 in the program), the effects on the linear element can be found in the same manner presented above; however, it is simpler to consider the element as shown in Figure 4.4a and observe that the nodal forces are merely the result of



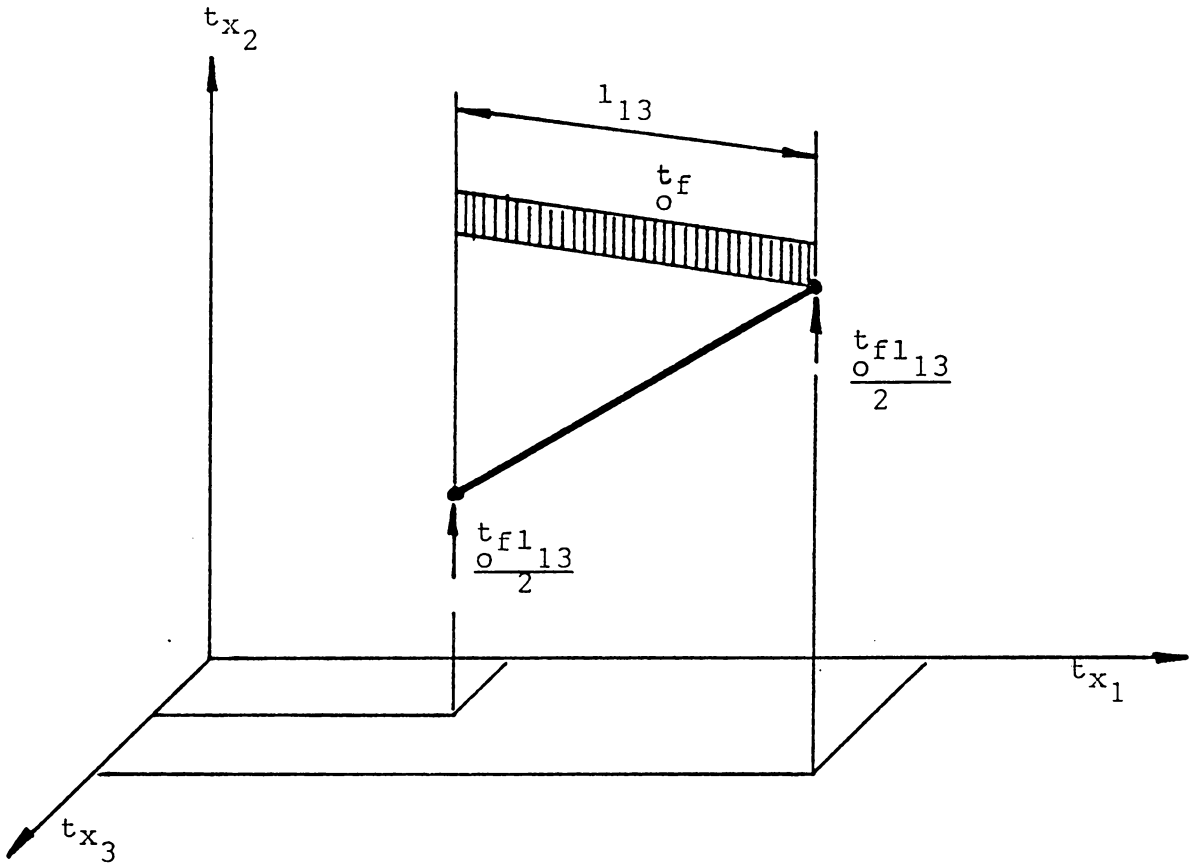
(a) Two Node Element



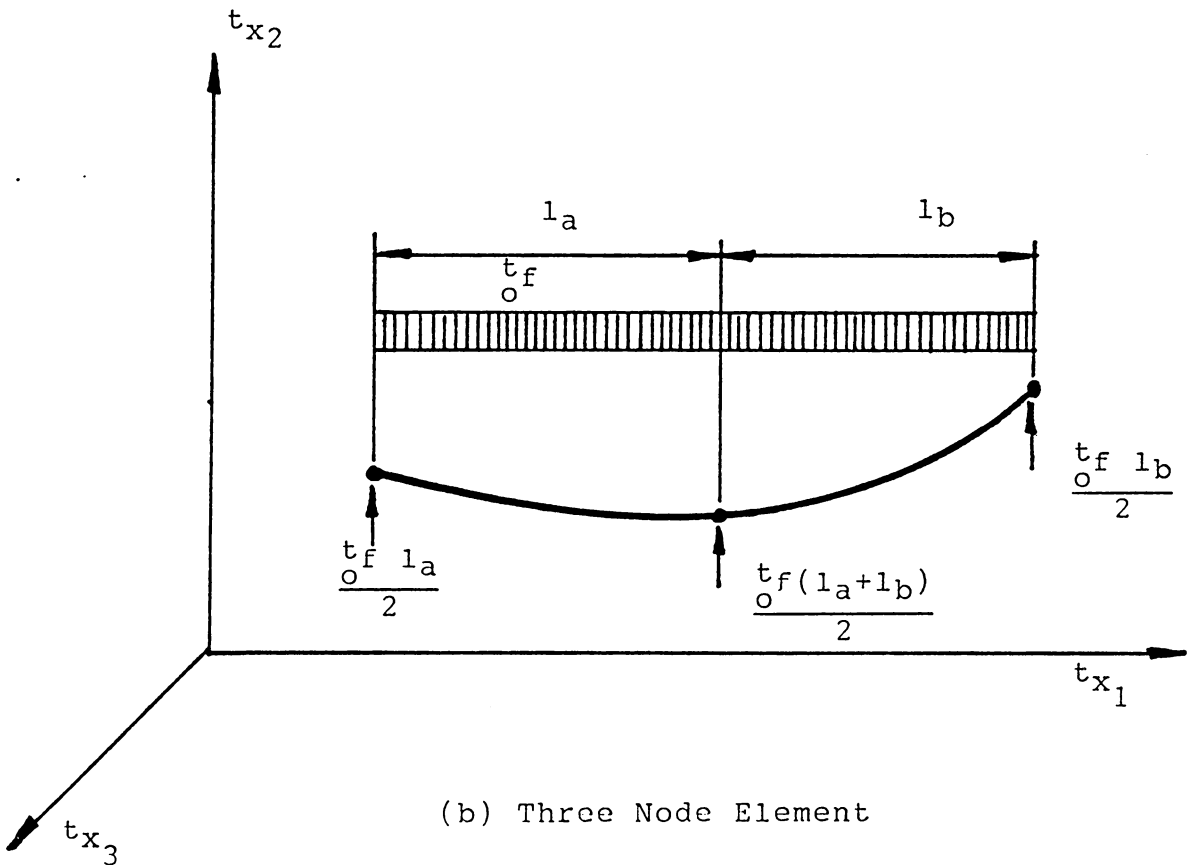
(b) Three Node Element

Figure 4.3: Dead Load (a) Two Node Element and, (b) Three Node Element

loading the projection of the element onto the x_1x_3 plane. A similar approach is used for the three node element as shown in Figure 4.4b. Here the element is divided into two and the contribution of each half is distributed as shown.



(a) Two Node Element



(b) Three Node Element

Figure 4.4: Uniform Load (a) Two Node Element and, (b) Three Node Element

Chapter 5

SOLUTION TECHNIQUES

5.1 Introduction

Analysis of an engineering problem entails three fundamental areas: the selection of an appropriate model for the problem, the use of a solution technique, and interpretation of results [2]. Of these three, perhaps the most important in nonlinear analysis is the selection of an appropriate solution technique. As was mentioned earlier, the solution of a nonlinear problem involves an iterative process, searching the load-displacement path for the desired equilibrium configuration or perhaps the buckling load. When choosing the appropriate method, there are three important characteristics to consider. First, the technique must be stable and predictable in its application. Second, it must converge to the correct solution. Finally, it must be efficient and simple to implement. Although no perfect solution technique exists, one must be aware of the characteristics of the technique being used and understand its limitations.

This chapter is designed to present a brief discussion on the different solution techniques used in this study: the Newton-Raphson approach, the Modified Riks-Wempner method, an Orthogonal Trajectory Accession method, and a Dynamic Relaxation approach. In addition, Section 5.7 gives the convergence criteria used in the program to control the solution procedure. The notations used in this chapter reflect that of reference [25] and thus the displacements of the system are now expressed as ${}^n q_i$ and the incremental displacements become Δq_i . In addition, Q represents the externally applied joint load vector and R is the unbalanced load vector.

5.2 The Tangent Stiffness Matrix

The common goal among solution techniques in nonlinear analysis is to develop an accurate approximation of the actual load-displacement curve (surface) for the given system under consid-

eration. Using this approximation, actual equilibrium configurations for the system are determined, establishing n discrete points along the curve. The simplest approximation is to assume the path to be linear between successive trial points. This is done by linearizing the equilibrium equation about a known equilibrium state and, using the linear form, establishing a trial point for the next equilibrium configuration. The result is a piecewise linear approximation to the actual curve. The details of this process will be discussed further in the sections to follow. For now, it is helpful to develop the general format used by the first three solution techniques to linearize the equations. The Dynamic Relaxation approach uses a slightly different method which will be discussed in Section 5.6.

The simplified form of the equation of equilibrium for a given system can be expressed as

$$f = f(q) \quad (5.2 - 1)$$

Linearization of this equation is performed by a truncated Taylor series expansion about the current equilibrium configuration. This results in

$$f(q + \Delta q) = f(q) + f_{,q}\Delta q \quad (5.2 - 2)$$

or, rearranging,

$$f_{,q}\Delta q = \Delta f \quad (5.2 - 3)$$

where

$$\Delta f = f(q + \Delta q) - f(q)$$

Comparing Eqs. 5.2-3 and 3.4-8, it is seen that

$$f_{,q} = {}^tK_T = \text{Tangent stiffness matrix.}$$

$$\Delta f = {}^{t+\Delta t}R - {}^tF$$

Thus, the tangent stiffness matrix is actually the first partial differential of f with respect to displacements.

As will be shown with each of the solution techniques (including Dynamic Relaxation), the tangent stiffness plays a very important role in the development of an accurate solution technique.

5.3 Newton-Raphson Methods

The iterative method most commonly used in nonlinear analysis is the Newton-Raphson method. Many other iterative techniques are developed directly from this approach, falling into the category of sequential search methods [34]. These methods are designed to search for the point which satisfies the function

$$f(q) = 0 \quad (5.3 - 1)$$

where

$$f(q) = {}^{t+\Delta t}Q - {}^{t+\Delta t}F(q) \quad (5.3 - 2)$$

To demonstrate the procedure, consider a one degree of freedom system with the load-displacement path as shown in Figure 5.1. Let the point O represent a known equilibrium configuration and it is desired to proceed along the curve to point P . If the equation of equilibrium is linearized about point O using the tangent stiffness matrix K_T at O (see Section 5.2), then the first trial configuration for point P can be determined by computing the unbalanced load

$$R^0 = Q^P - F^0 \quad (5.3 - 3)$$

and solving

$$K_T^0 \Delta q^0 = R^0 \quad (5.3 - 6)$$

for Δq^0 (see Figure 5.1). An improved trial configuration corresponding to Δq^0 can be found from

$$q^1 = q^0 + \Delta q^0 \quad (5.3 - 7)$$

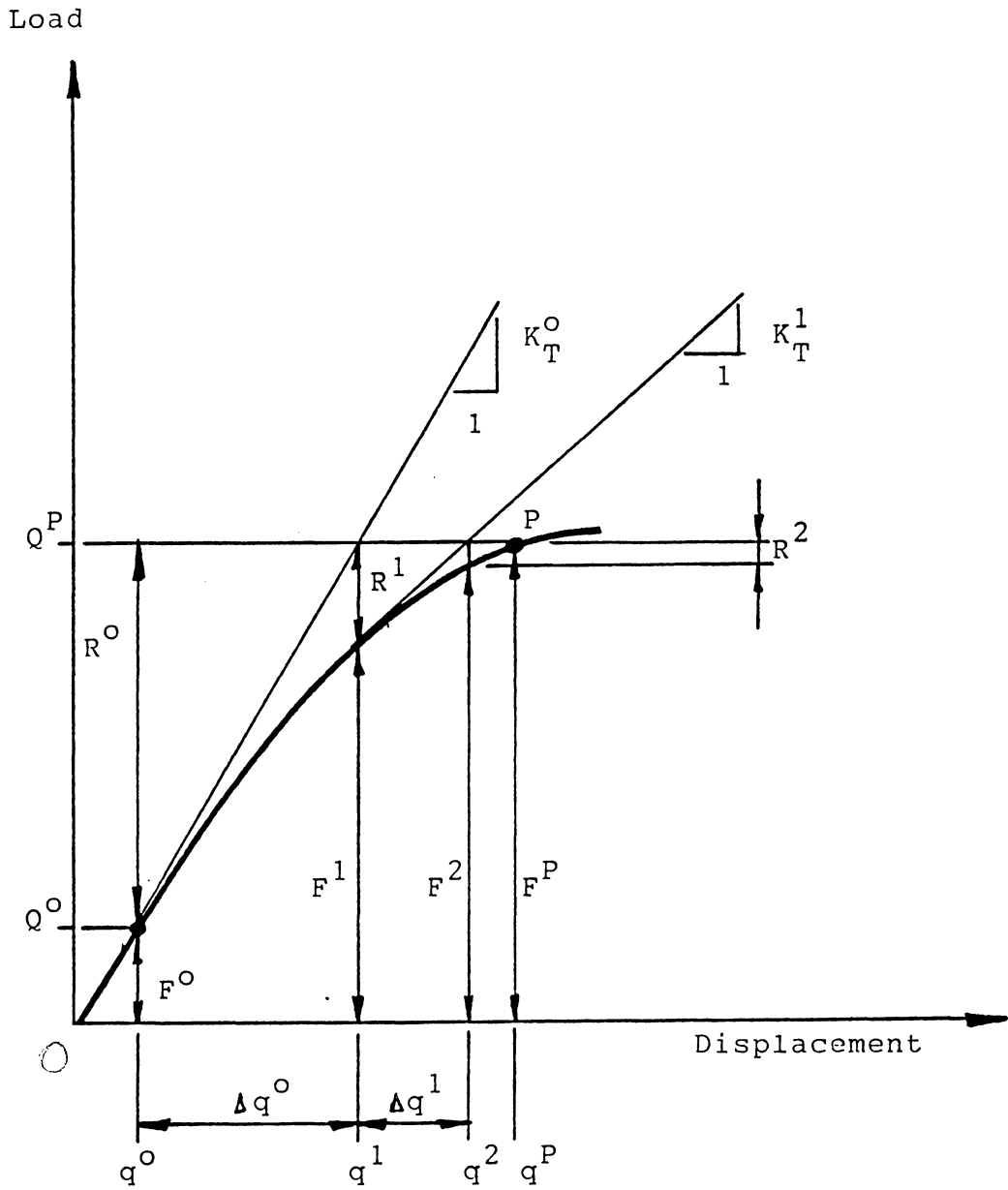


Figure 5.1. Physical Interpretation of the Newton-Raphson Method

Using Eq. 5.3-5, the internal force for the system in configuration 1 can be found. A check for convergence is then used to decide if the state is sufficiently close to equilibrium. If not, the process proceeds to the next trial configuration until convergence is satisfied. The general algorithm can be stated as

$$K_T^k \Delta q^k = R^k \quad (5.3 - 6)$$

and

$$q^{k+1} = q^k + \Delta q^k \quad (5.3 - 7)$$

In general, Eq. 5.3-6 represents a set of n simultaneous equations in the unknown displacements Δq^k . The solution thus involves the generation and factorization of the tangent stiffness matrix K_T^k for each iteration. This process can become quite expensive for large systems. In order to overcome the need for numerous factorizations, a Modified Newton-Raphson approach is often used. In this method, K_T is held constant for a predetermined number of iterations before it is updated as shown graphically in Figure 5.2. That is

$$K_T^0 \Delta q^k = R^k \quad (5.3 - 8)$$

It is apparent from the figure that the Modified Newton-Raphson approach will generally require a greater number of iterations between configurations as compared to the strict Newton-Raphson approach. This of course results in more computer time spent in iteration but less time in generating K_T .

The Newton-Raphson methods are not without their problems, however. The major drawback of either approach is that they cannot trace around turning points on the equilibrium path. The methods therefore cannot be used beyond limit points in stability problems. This and other problems experienced by the methods are given in Figures 5.3a and 5.3b. The Modified Newton-Raphson is particularly unreliable when studying "hardening" load-displacement curves as shown in Figure 5.3c [10].

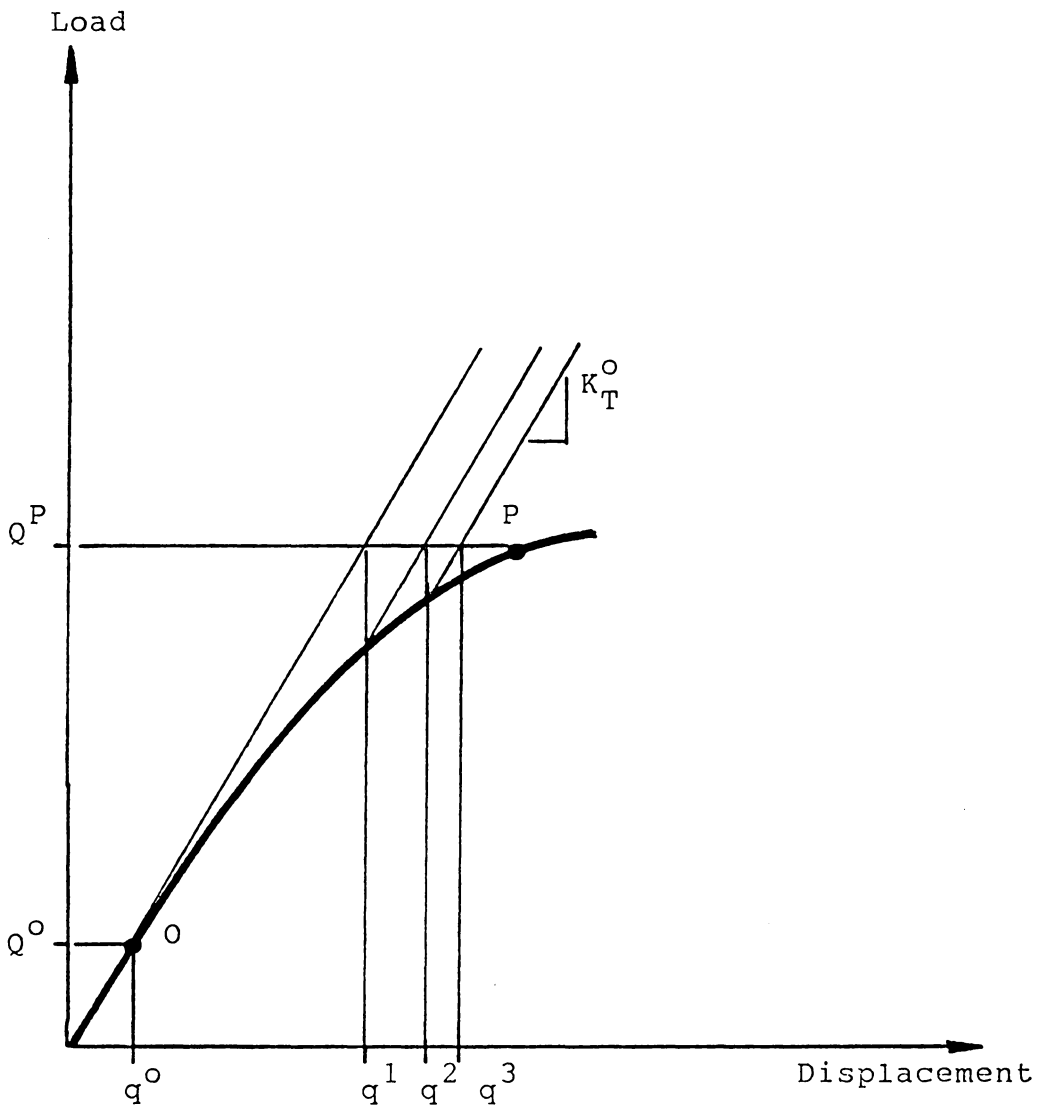
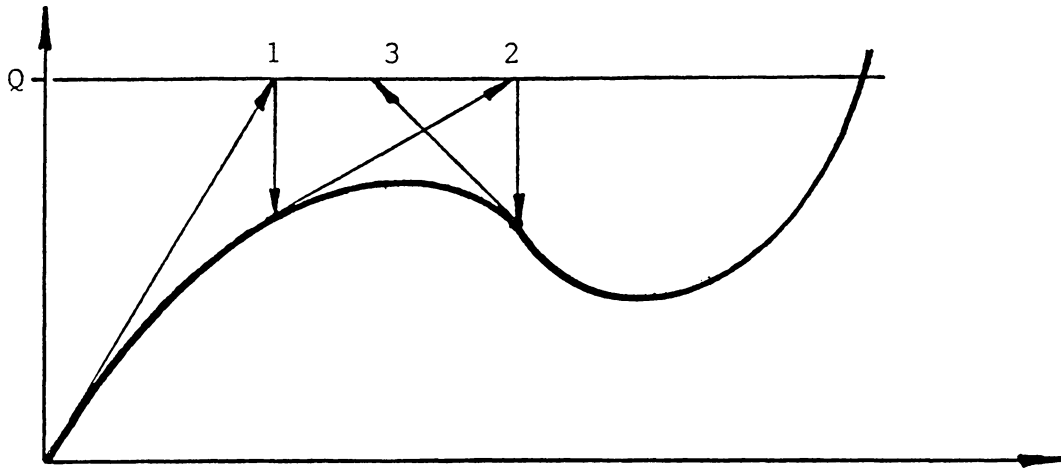
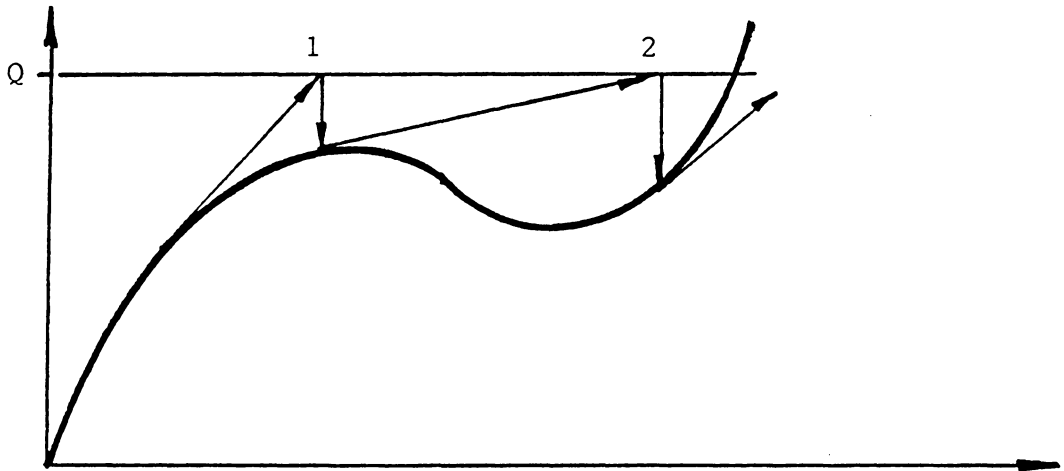


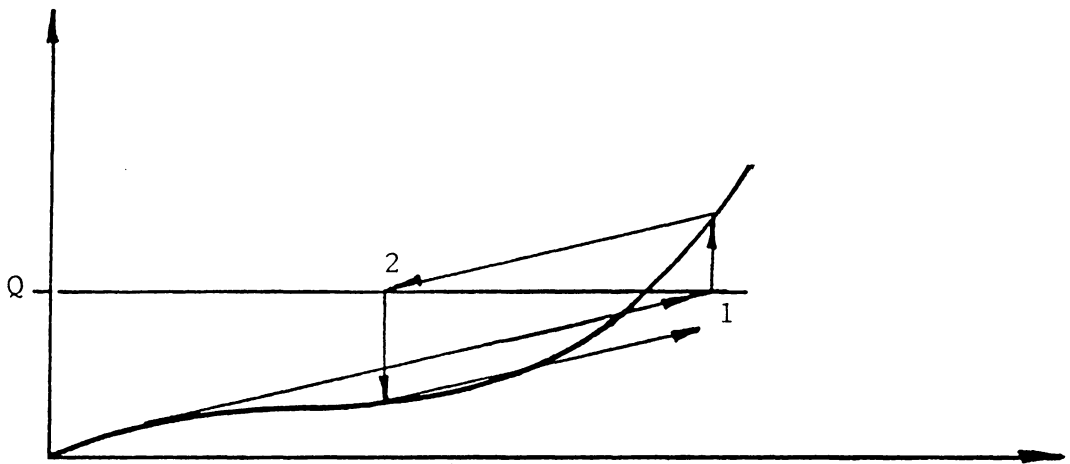
Figure 5.2. The Modified Newton-Raphson Method



(a)



(b)



(c)

Figure 5.3. Failure of the Newton-Raphson Methods

A variation of the above approaches is to apply the load in stages during deformation as shown in Fig 5.4. This allows for the load-displacement path to be traced more closely and thus more accurately. In some cases this approach, referred to as an incremental-iterative approach, can give an indication of where stability problems arise in a structure.

5.4 Modified Riks-Wempner Method

To overcome the difficulties experienced by the Newton-Raphson methods, the Modified Riks-Wempner method is often used [25]. Though based on the Newton-Raphson approach, this method differs in the manner in which it traces the equilibrium path. Where the Newton-Raphson method selects a constant loading increment to control the progress along the path, the Modified Riks-Wempner method uses a generalized arc length Δs along the tangent to the current equilibrium point and then iterates along the normal to Δs until convergence is satisfied. This approach is depicted in Figure 5.5.

In presenting the formulation of the method, it is helpful to redefine the governing equilibrium equation for an n degree of freedom system using the principle of potential energy [25]. The total potential energy of the system is expressed as

$$V(q, \lambda) = \Omega(q, \lambda) + U(q) \quad (5.4 - 1)$$

where

q = Generalized displacement vector.

λ = A loading parameter.

Ω = Potential energy of external loads.

U = Potential internal strain energy.

Assuming the loads on the structure remain independent of deformation, Ω can be written as

$$\Omega = -q^T Q = -q^T (\lambda \bar{Q}) \quad (5.4 - 2)$$

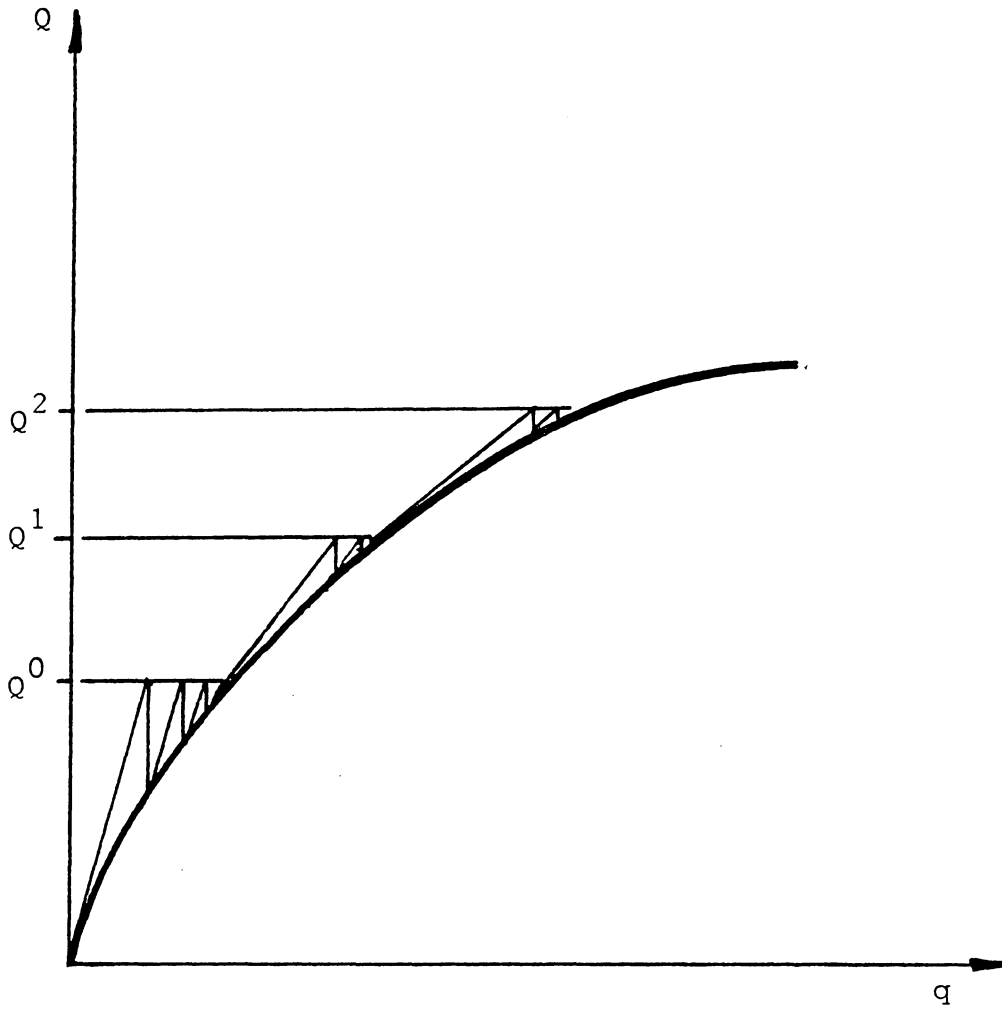


Figure 5.4. Incremental-Iterative Newton-Raphson Method

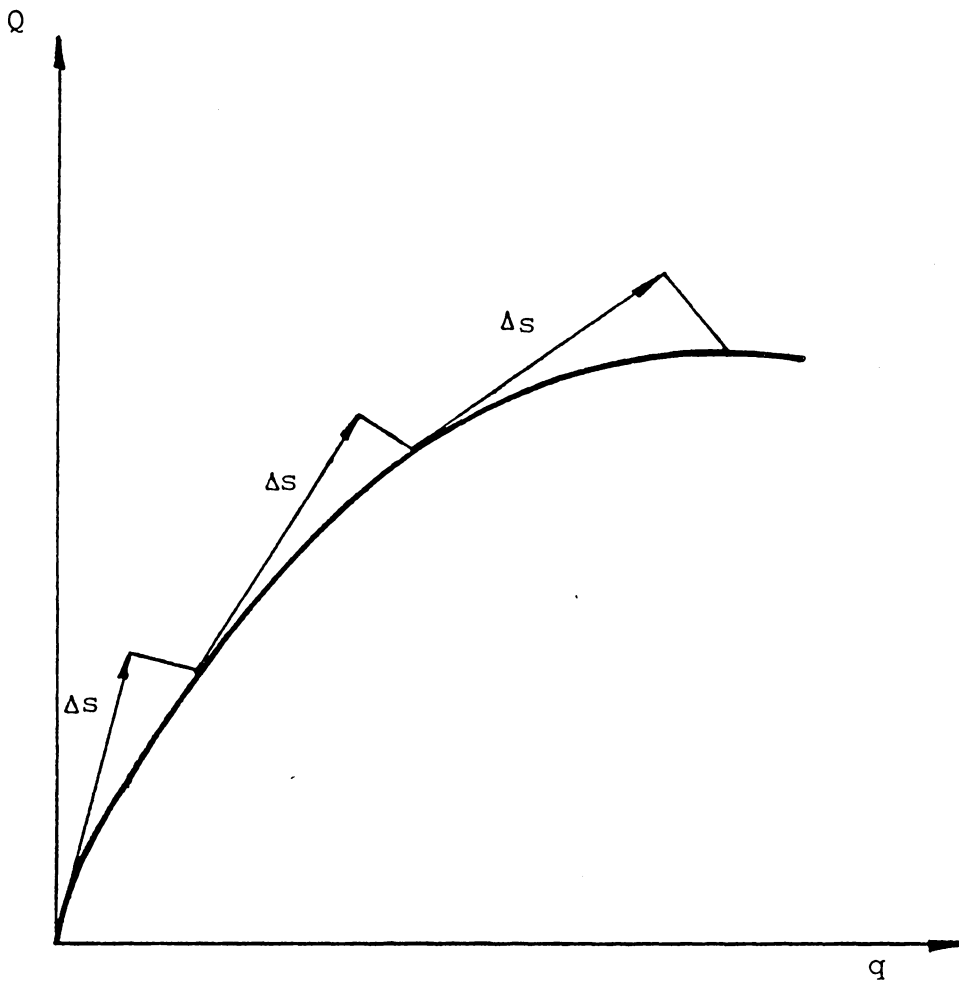


Figure 5.5. The Modified Riks-Wempner Method

where \bar{Q} represents a constant load distribution vector on the structure and Q is the generalized force vector [25].

Applying the principle of stationary potential energy, $\delta V = 0$, the n nonlinear equilibrium equations for the system are obtained as

$$R_i(q, \lambda) = \lambda \bar{Q}_i - \frac{\partial U}{\partial q_i} = 0 \quad i = 1, 2 \dots n \quad (5.4 - 3)$$

Let the point (q^k, λ^k) represent an approximate solution to Eq. 5.4-3. If the equilibrium equation is linearized about this point using a truncated Taylor series expansion, Eq. 5.4-3 becomes

$$R(q^{(k+1)}, \lambda^{(k+1)}) = R(q^k, \lambda^k) + \frac{\partial R(q^k, \lambda^k)}{\partial \lambda} \Delta \lambda^k + \frac{\partial R(q^k, \lambda^k)}{\partial q} \Delta q^k = 0 \quad (5.4 - 4)$$

Recognizing that

$$\frac{\partial R(q^k, \lambda^k)}{\partial \lambda} = \bar{Q} \quad (5.4 - 5)$$

and

$$\frac{\partial R(q^k, \lambda^k)}{\partial q} = K_T^k \quad (5.4 - 6)$$

the linearized incremental equilibrium equation becomes

$$K_T^k \Delta q^k = R^k + \Delta \lambda^k \bar{Q} \quad (5.4 - 7)$$

where

$$R^k = R(q^k, \lambda^k) = \text{Unbalanced force vector at } (q^k, \lambda^k),$$

$$K_T^k = \text{Tangent stiffness matrix at } (q^k, \lambda^k),$$

and

$$q^{k+1} = q^k + \Delta q^k \quad (5.4 - 8)$$

$$\lambda^{k+1} = \lambda^k + \Delta \lambda^k \quad (5.4 - 9)$$

Eq. 5.4-7 shows clearly the difference between the Newton-Raphson method and the Modified Riks-Wempner method. If the load level is held constant during iteration, $\Delta \lambda^k = 0$, and Eq. 5.4-7 becomes Eq. 5.3-6 in the Newton-Raphson method.

In the n-dimensional space represented by Eq. 5.4-7, let the point (q^k, λ^k) denote the k^{th} trial equilibrium configuration and (q^{k+1}, λ^{k+1}) denote the trial configuration $k + 1$. These two points can be expressed by the vectors

$$r^k = \begin{bmatrix} q^k \\ \lambda^k \end{bmatrix} \quad (5.4 - 10)$$

and

$$r^{k+1} = r^k + \Delta r^k \quad (5.4 - 11)$$

where

$$\Delta r^k = \begin{bmatrix} \Delta q^k \\ \Delta \lambda^k \end{bmatrix} \quad (5.4 - 12)$$

Thus

$$r^{k+1} = \begin{bmatrix} q^k \\ \lambda^k \end{bmatrix} + \begin{bmatrix} \Delta q^k \\ \Delta \lambda^k \end{bmatrix} \quad (5.4 - 13)$$

These vectors are shown in Figure 5.6. For the first step of the iteration, Δr^0 represents a vector directed along the tangent to the curve at point 0 (see Figure 5.6) with magnitude

$$\Delta s = [\Delta q^0 \cdot \Delta q^0 + (\Delta \lambda^0)^2]^{1/2} \quad (5.4 - 14)$$

Now, let the incremental displacement be denoted by

$$\Delta q^0 = \Delta \lambda^0 \Delta q^{0I} \quad (5.4 - 15)$$

where Δq^{0I} is obtained by solving

$$K_T^0 \Delta q^{0I} = \bar{Q} \quad (5.4 - 16)$$

Substitution of Eq. 5.4-15 into 5.4-14 results in

$$\Delta s = \Delta \lambda^0 [\Delta q^{0I} \cdot \Delta q^{0I} + 1]^{1/2} \quad (5.4 - 17)$$

which is commonly called the constraint equation of the Riks-Wempner method [25].

Assuming Δs to be prescribed for the load step, the increment in load becomes

$$\Delta \lambda^0 = \pm \frac{\Delta s}{[\Delta q^{0I} \cdot \Delta q^{0I} + 1]^{1/2}} \quad (5.4 - 18)$$

and, combined with Eq. 5.4-15, trial configuration 1 becomes (see Figure 5.6)

$$q^1 = q^0 + \Delta q^0 \quad (5.4 - 19)$$

$$\lambda^1 = \lambda^0 + \Delta \lambda^0 \quad (5.4 - 20)$$

Although Δs represents a constant magnitude during the progress between two equilibrium points, the value of Δs may be altered when proceeding along the path in order to attain a smooth curve (see Figure 5.7). This is done by

$$\Delta \hat{s} = \Delta s \left(\frac{\hat{I}}{I} \right)^{1/2} \quad (5.4 - 21)$$

where

\hat{I} = The number of iterations desired between points (usually ≤ 4).

I = Number of iterations required for previous increment.

$\Delta \hat{s}$ = Length of vector for next step.

To continue to the next trial point, the vector Δr^k is restricted to lie on the hyperplane normal to Δr^0 . In order to maintain the symmetry and bandedness of K_T , Δr^k is usually found by a two step approach in the form(see Figure 5.6) [25]

$$\Delta r^k = \Delta \lambda^k \Delta r^{kI} + \Delta r^{kII} \quad (5.4 - 22)$$

or

$$\Delta r^k = \left[\frac{\Delta q^k}{\Delta \lambda^k} \right] = \Delta \lambda^k \left[\frac{\Delta q^{kI}}{1} \right] + \left[\frac{\Delta q^{kII}}{0} \right] \quad (5.4 - 23)$$

From Eq. 5.4-23, it is seen that the incremental displacement vector thus becomes

$$\Delta q^k = \Delta \lambda^k \Delta q^{kI} + \Delta q^{kII} \quad (5.4 - 24)$$

Substitution of Eq. 5.4-24 into Eq. 5.4-7 gives

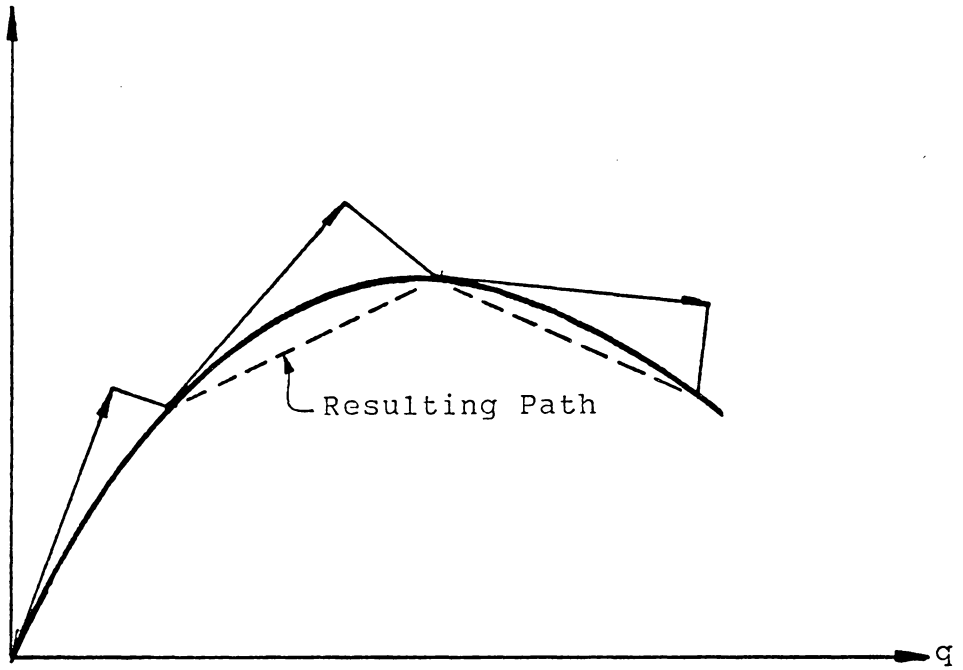
$$K_T^k [\Delta \lambda^k \Delta q^{kI} + \Delta q^{kII}] = R^k + \Delta \lambda^k \bar{Q} \quad (5.4 - 25)$$

This equation can then be separated into the equations

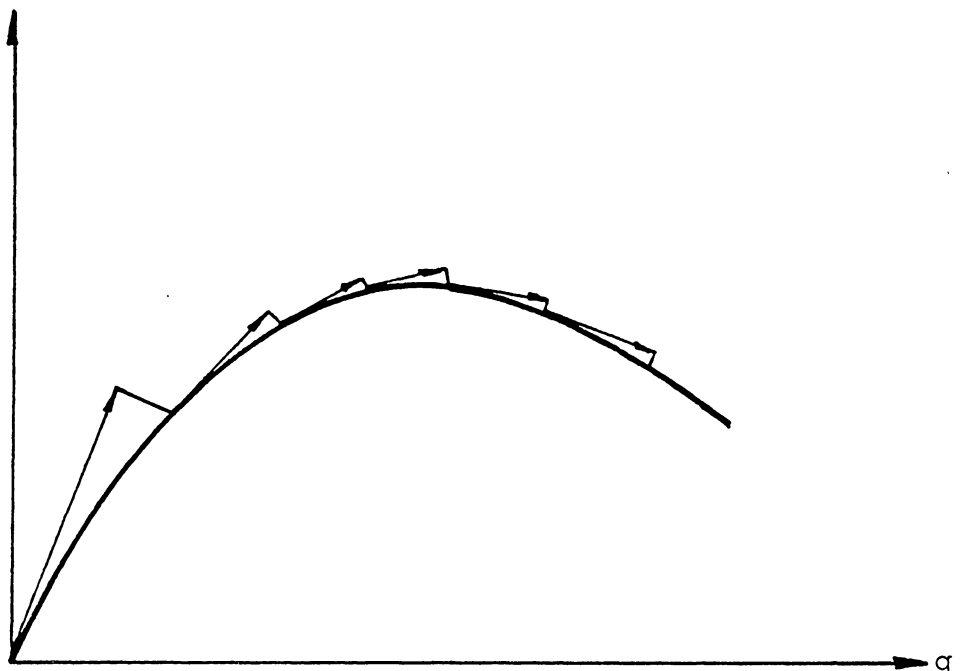
$$K_T^k \Delta q^{kI} = \bar{Q} \quad (5.4 - 26)$$

$$K_T^k \Delta q^{kII} = R^k \quad (5.4 - 27)$$

from which the two vectors Δq^{kI} and Δq^{kII} can be found.



(a) Unscaled Arc Length



(b) Scaled Arc Length

Figure 5.7. Scaling the Arclength in the Modified Riks-Wempner Method

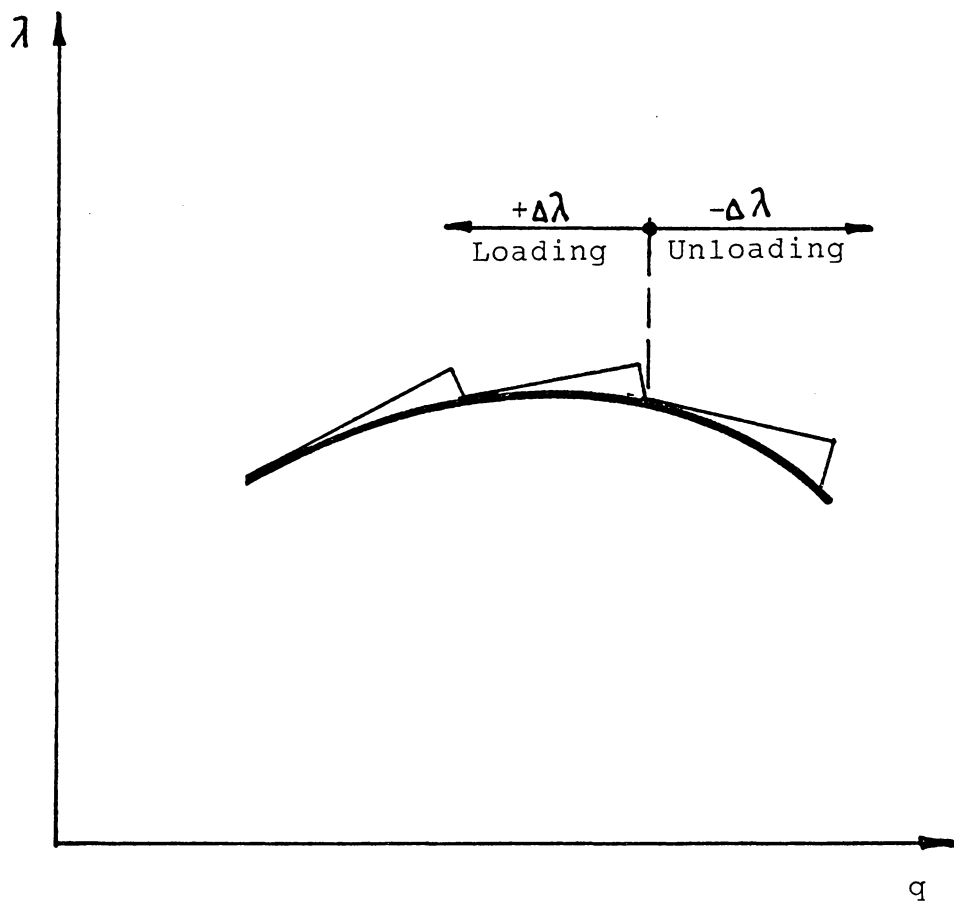


Figure 5.8. Loading and Unloading of the Modified Newton-Raphson Method

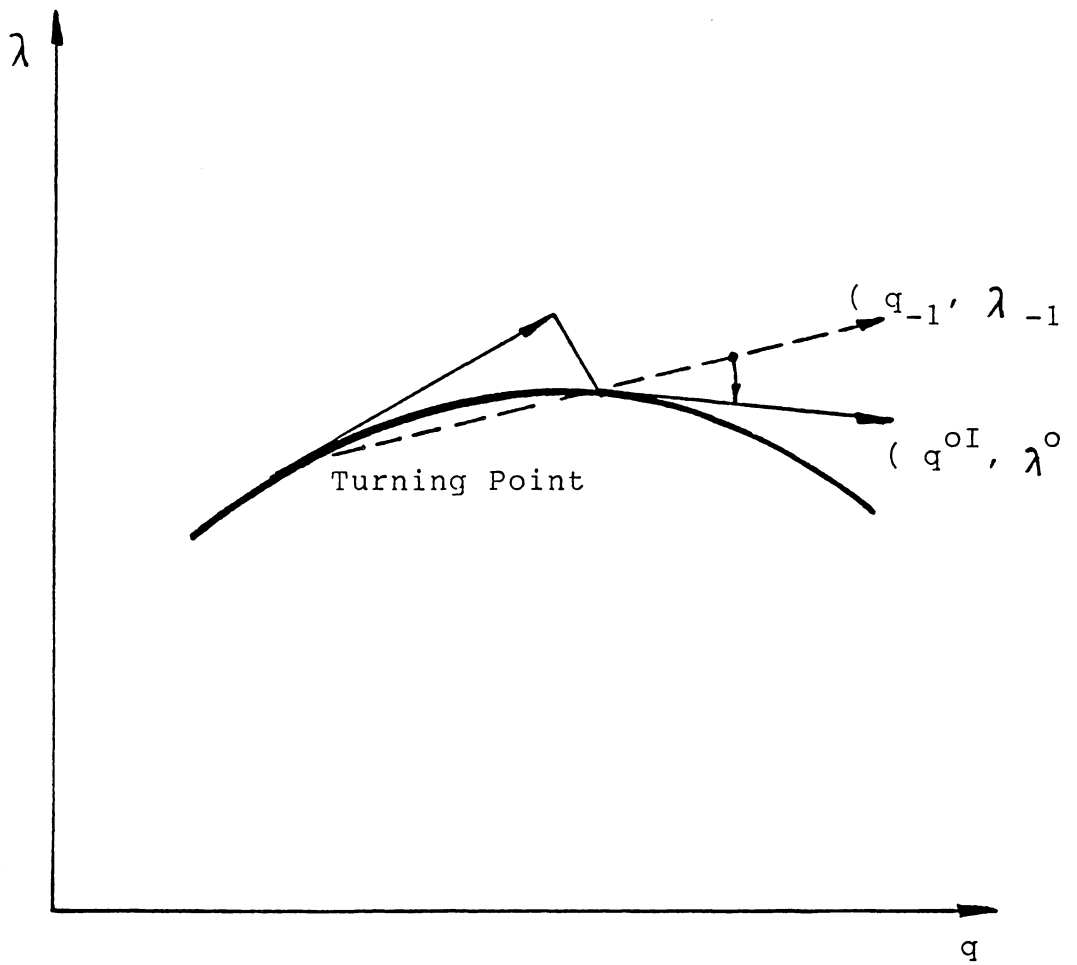


Figure 5.9. Determination of Correct Sign



Figure 5.10. Incorrect Sign Change

The value of $\Delta\lambda^k$ is determined by applying the orthogonality condition

$$\Delta r^0 \cdot \Delta r^k = 0 \quad (5.4 - 28)$$

Substitution of Eqs. 5.4-24 and 5.4-10, with $k = 0$, into Eq. 5.4-28 results in [25]

$$\Delta\lambda^k = - \frac{(\Delta q^0 \cdot \Delta q^{kII})}{(\Delta q^0 \cdot \Delta q^{kI} + \Delta\lambda^0)} \quad (5.4 - 29)$$

The equilibrium configuration at $k + 1$ is then found by combining the results of Eqs. 5.4-24 and 5.4-29 with 5.4-8 and 5.4-9, respectively.

The sign ambiguity of Eq. 5.4-18 is a result of taking the square root of Δs in the formulation. The physical meaning signifies whether the path represents a loading (+) or unloading (-) of the system (see Figure 5.8). Thus the ability for the load increment to change sign allows the method to trace around corners on the load-displacement path, eliminating some of the problems encountered with the Newton-Raphson approach at limit points. The choice of the sign is determined such that the dot product between the current initial vector $\Delta r^0 = \Delta\lambda^0(\Delta q^0, 1)$ and the previous increment $(\Delta q_{-1}, \Delta\lambda_{-1})$ is always positive (see Figure 5.9) [23]. That is

$$\Delta\lambda^0(\Delta q^{0I}, 1) \cdot (\Delta q_{-1}, \Delta\lambda_{-1}) > 0 \quad (5.4 - 30)$$

or

$$\Delta\lambda^0(\Delta q^{0I} \cdot \Delta q_{-1} + \Delta\lambda_{-1}) > 0$$

Although on occasion the above approach may produce an incorrect sign change, particularly in areas of high curvature (see Figure 5.10), the occurrence is extremely rare and can be avoided by reducing Δs to trace the curve more closely [23]. To reduce storage requirements in the program, the value of $\Delta\lambda_{-1}$ is taken as $\Delta\lambda_{-1}^e$. The above equation only produces a sign change and $\Delta\lambda_{-1}^e$ is sufficiently close to $\Delta\lambda_{-1}$ to be taken as the same.

5.5 Orthogonal Trajectory Accession

Orthogonal Trajectory Accession (OTA) is an iterative approach introduced by Fried [15] in an attempt to overcome the difficulties experienced by both the Newton-Raphson approach and the Modified Riks-Wempner method. The approach used in OTA is essentially the same as in the Modified Riks-Wempner method with one important alteration. Both methods move away from the previous equilibrium point an amount Δs along the tangent to the curve; however, where the Riks-Wempner approach iterates along a path normal to Δs , the OTA approach iterates along paths normal to the current tangent at k (see Figure 5.11). As will be discussed later, this difference allows OTA to start from almost any point and does not require a previous equilibrium configuration.

The derivation of the method proceeds in exactly the same manner as in the Riks-Wempner method until the application of the orthogonality condition. Thus for the OTA approach, the constraint equations for the initial trial configuration are (see Section 5.4 for details)

$$r^0 = \begin{bmatrix} q^0 \\ \lambda^0 \end{bmatrix} \quad (5.5 - 1)$$

$$\Delta\lambda^0 = \pm \frac{\Delta s}{[\Delta q^{0I} \cdot \Delta q^{0I} + 1]^{1/2}} \quad (5.5 - 2)$$

$$\Delta q^0 = \Delta\lambda^0 \Delta q^{0I} \quad (5.5 - 3)$$

$$q^1 = q^0 + \Delta q^0 \quad (5.5 - 4)$$

$$\lambda^1 = \lambda^0 + \Delta\lambda^0 \quad (5.5 - 5)$$

Again, as in the Modified Riks-Wempner approach, the vector Δr^k which seeks to correct the unbalanced force at configuration k is expressed as (see Section 5.4 for derivation)

$$\Delta r^k = \Delta\lambda^k \begin{bmatrix} \Delta q^{kI} \\ 1 \end{bmatrix} + \begin{bmatrix} \Delta q^{kII} \\ 0 \end{bmatrix} \quad (5.5 - 6)$$

The orthogonality condition used in OTA seeks to keep the vector Δr^k normal to the tangent to the curve at the current (k) configuration. This leads to the orthogonality equation

$$\Delta r^k \cdot (\Delta \lambda^k \Delta q^{kI}, \Delta \lambda^k) = 0 \quad (5.5 - 7)$$

where the vector $(\Delta \lambda^k \Delta q^{kI}, \Delta \lambda^k)$ coincides with the tangent at k . Substituting Eq. 5.5-6 into Eq. 5.5-7 results in

$$\Delta \lambda^k = - \frac{(\Delta q^{kI} \cdot \Delta q^{kII})}{(\Delta q^{kI} \cdot \Delta q^{kI} + 1)} \quad (5.5 - 8)$$

and from Eq. 5.5-6

$$\Delta q^k = \Delta \lambda^k \Delta q^{kI} + \Delta q^{kII} \quad (5.5 - 9)$$

Using Eqs. 5.5-8 and 5.5-9, the new configuration can again be found by

$$q^{k+1} = q^k + \Delta q^k \quad (5.5 - 10)$$

$$\lambda^{k+1} = \lambda^k + \Delta \lambda^k \quad (5.5 - 11)$$

Comparing Eqs. 5.5-8 and 5.4-29, the difference in the two approaches becomes clear. In the Modified Riks-Wempner method, the iteration from a current trial configuration to the next trial state requires the tangent to the path at a previous equilibrium configuration in order to satisfy orthogonality requirements. The OTA method, however, requires only the current trial configuration to proceed to the next configuration. Thus in the OTA approach, Δs serves only to control the "location" of the first trial configuration of a given increment. In the Riks-Wempner method, Δs serves to control both the location of the first trial point and the direction of progress to the path for a given increment.

To further clarify this important difference, consider a one degree of freedom system with a load-displacement curve as shown in Figure 5.12. In addition, let A represent an initial trial state. It is obvious this point does not lie on the equilibrium path and could not serve as a starting point

for the Riks-Wempner method. The OTA method, however, does not require an initial equilibrium state to proceed to the next point on the path and thus begins to seek out an equilibrium point by successive iterations normal to the current tangent to the path [15]. The process is demonstrated in Figure 5.12 and the successive trial points are numbered. Once a point on the curve is found, the method proceeds to trace the path in much the same fashion as the Modified Riks-Wempner method.

Although the ability to start from such a poor configuration is a very attractive feature of the method, the tenacity of the approach in remaining on the path is perhaps its greatest ability. As pictured in Figure 5.13, even with a poor value of Δs , the curve can be traced with only a loss of time in excessive iteration (Δs may be scaled in the same manner as in the Riks-Wempner method to overcome this problem).

The attractiveness of this method in the study of stability problems should be apparent; however, the concern of this study was in application to a rather simple load-displacement path with no snap-through characteristics. In addition, there seems to be only a few published applications of OTA. In order to make further evaluations of the differences between OTA and the Modified Riks-Wempner Method, a quick comparison between the two approaches was made using a more complex stability problem. The results are included in Section 8.5 as Test Set 4.

5.6 A Dynamic Relaxation Approach

The last solution approach investigated in the study is based on a Dynamic Relaxation approach. Dynamic Relaxation (DR) employs the fundamentals of structural dynamics to determine an equilibrium configuration for the system. The theory behind the method is that if the system, excited by a constant load, is allowed to undergo damped vibration, it will eventually come to rest in an equilibrium configuration [37]. Thus the equation of equilibrium for the structure becomes the equation of motion

$$M \dot{\ddot{q}} + C \dot{q} + K q = \dot{Q} \quad (5.6 - 1)$$

where the mass matrix M and the damping matrix C are included to account for the effects of inertial forces during motion. The terms ${}^t\ddot{q}$ and ${}^t\dot{q}$ represent the acceleration and velocity of the system at time t where the over dots signify differentiation with respect to time. Attention is now focused on the motion of the system and therefore time becomes a variable in the formulations to follow.

Although DR was first introduced for use in analysis of linear systems, the iterative nature of dynamic processes adapts well to nonlinear analysis [43]. The method is particularly attractive when the structure experiences regions where the the system stiffness matrix becomes very soft or even singular [43]. In addition, if the solution of Eq. 5.6-1 is performed using explicit, direct integration techniques, the equation can be expressed in vector form which simplifies computations and reduces storage requirements [2,37,43].

The approach, however, cannot trace paths which are dynamically unstable or converge at unstable limit points and thus fails in post buckling analysis. In addition, a great number of iterations may be required to converge to a solution for a system which experiences dramatic changes during loading [43].

This section is designed to present a brief overview of DR, focusing attention on one particular variation proposed by Underwood [43] for controlling the mass and damping parameters. Although each author presents his own techniques for implementing DR, the overall procedures in each case are very similar. In the interest of brevity, only the one technique was chosen for use in the study in order to determine the merits of DR in the analysis of cable structures. Furthermore, since the study is concerned only with the effects of static loading, the theories involved with a complete dynamic analysis will not be presented here. The reader is directed to references such as Bathe [2] for more details in dynamic analysis.

Before developing the procedure, an important observation can be made with regards to Eq. 5.6-1. Since the equation is formulated at time t , the term tKq is equivalent to tF which gives

$$M {}^t\ddot{q} + C {}^t\dot{q} + {}^tF = {}^tQ \quad (5.6 - 2)$$

This in effect eliminates the need to develop the stiffness matrix for the system since the internal forces at time t can be found from consideration of the current state of strain in the element. Although this is not an outstanding attribute for the average nonlinear system, it is attractive to the analysis of cables since non-pretensioned cable structures have a tendency to experience small rigid body motions which cause the stiffness matrix to become singular.

As mentioned earlier, the use of an explicit integration scheme in the solution technique is particularly attractive due to simplification of the process and efficient use of storage. The most common approach is the use of central difference equations to approximate the time derivatives. The use of this technique linearizes the equation of motion about the current configuration at time t by expressing the acceleration and velocity at t in terms of those at time $t + \Delta t$ and $t - \Delta t$ as

$${}^t\dot{q} = \frac{1}{2}({}^{t-\Delta t}\dot{q} + {}^{t+\Delta t}\dot{q}) \quad (5.6 - 3)$$

$${}^t\ddot{q} = \frac{1}{\Delta t}(-{}^{t-\Delta t}\dot{q} + {}^{t+\Delta t}\dot{q}) \quad (5.6 - 4)$$

where

$${}^{t-\Delta t}\dot{q} = \frac{1}{\Delta t}(-{}^{t-\Delta t}q + {}^{t+\Delta t}q) \quad (5.6 - 5)$$

Substitution of the above expressions into Eq. 5.6-2 results in the pair of equations which can be solved for the configuration at time $t + \Delta t$, giving

$${}^{t+\Delta t}\dot{q} = \frac{(M/\Delta t - 1/2C)}{(M/\Delta t + 1/2C)} {}^{t-\Delta t}\dot{q} + \frac{{}^tQ - {}^tF}{(M/\Delta t + 1/2C)} \quad (5.6 - 6)$$

$${}^{t+\Delta t}q = {}^tq + \Delta t {}^{t+\Delta t}\dot{q} \quad (5.6 - 7)$$

In order to preserve the advantages of using an explicit form for the solution, the mass and damping matrices in Eqs. 5.6-6 and 5.6-7 are considered to be diagonal in form [37,43]. Although this simplification does introduce a certain amount of error during iteration, as the system converges

to the desired equilibrium point, the inertial effects vanish and thus no longer effect the solution [43].

The primary difference between DR and a more conventional dynamic analysis is that DR is concerned solely with attaining an equilibrium configuration regardless of the motion the system undergoes. As a result, implementation of the method often involves scaling the mass and damping matrices in Eq. 5.6-2 in order to control the speed at which the equilibrium configuration is attained. There are generally three methods used by various authors to perform this scaling process [43]:

1. Assume the mass matrix to be proportional to the main diagonal of the tangent stiffness matrix, that is

$$M = \rho D \quad (5.6 - 8)$$

where D is a diagonal matrix with $D_{ii} = K_{ii}$ and ρ is a constant with units of mass density.

2. Assume that the mass is proportional to the absolute row sum of the tangent stiffness matrix, that is

$$m_{ii} = \frac{1}{4} \Delta t^2 \sum_{j=1}^n |K_{Tij}| \quad (5.6 - 9)$$

This is an adaptation of Gerschgorin's theorem [43].

3. "Choosing m_{ii} such that the transit times for information transfer for degree of freedom i to adjacent and like degrees of freedom is constant "[43].

In all the cases, the damping matrix is normally assumed to be a multiple of M which further reduces the number of unknowns.

The method proposed by Underwood [43] falls into the second category. In this approach, the damping matrix is assumed to be expressed as

$$C = cM \quad (5.6 - 10)$$

where c is a damping coefficient.

Using Gerschgorin's theorem [43], which states that all the eigenvalues of the matrix A lie within one of the circles C_1, C_2, \dots, C_n where C_i has its center at a_{ii} and radius equal to $\sum_{j \neq i}^n |K_{ij}|$, the mass matrix can then be found from Eq. 5.6-9.

A well known requirement for solution stability when using central difference approximations is that the time step for the solution process must remain less than the critical time step expressed as [2,43]

$$\Delta t \leq \Delta t_{cr} = \frac{T_n}{\pi} \quad (5.6 - 11)$$

where T_n is the smallest period of the n degree of freedom system.

The adaptive method proposed by Underwood [43] uses a constant time step $\Delta t = 1.0$ and seeks to adjust M (and thus C as well) to maintain the stability requirements of central difference. As a "margin of safety", Eq. 5.6-9 is computed using $\Delta t = 1.1$ while iterations are performed with $\Delta t = 1.0$. In addition, to account for some of the errors introduced by the central difference approximations, the mass matrix from Eq. 5.6-9 is updated as necessary during the course of the solution process, particularly when the stiffness matrix changes considerably during loading (i.e., stiffening structures). To determine when the mass matrix must be reformed, Underwood [43] uses a "perturbed apparent frequency error measure" [38], ϵ , which is found as

$$\epsilon = \max(\epsilon_1, \dots, \epsilon_{\max \text{ DOF}}) \quad (5.6 - 12)$$

where

$$\begin{aligned} \epsilon_i &= \Delta \frac{t^2}{4} \frac{t a_i}{t b_i} \\ t a_i &= |t \ddot{q}_i - t^{-\Delta t} \ddot{q}_i| \\ t b_i &= |q_i^n - t^{-\Delta t} q_i| \end{aligned}$$

When ε is greater than 1.0, the mass matrix must be reformed or a smaller Δt must be used.

After determining the mass matrix, the value of c is determined using Rayleigh's quotient

$${}^t c = 2\sqrt{{}^t q^T {}^t K^t {}^t q / {}^t q^T M^t {}^t q} \quad (5.6 - 13)$$

or, letting ${}^t \nu = {}^t c \Delta t$,

$${}^t \nu = 2\Delta t \sqrt{{}^t q^T {}^t K^t {}^t q / {}^t q^T M^t {}^t q} \quad (5.6 - 14)$$

The term ${}^t K^t$ in Eqs. 5.6-13 and 5.6-14 is a "local" tangent stiffness matrix defined as

$${}^t K_{ii}^t = \frac{[-F_i({}^{t-\Delta t} q) + F_i({}^t q)]}{\Delta t {}^{t-\Delta t} q_i} \quad (5.6 - 15)$$

which is an estimation of the current tangent stiffness matrix at t . Eq. 5.6-15 is an approximation to the value of critical damping for the current configuration [38,43].

The adaptive procedure can be summarized as follows 43 :

1. Given initial displacement vector ${}^0 q$; assume ${}^0 \dot{q} = 0$.
2. Compute the mass matrix using Eq. 5.6-9 with $\Delta t = 1.1$.
3. Compute the critical damping parameter from Eq. 5.6-15.

4. For $t = 0$;

$${}^{t+\Delta t} \dot{q} = \Delta t M^{-1} r^0 / 2$$

For $t \neq 0$;

$${}^{t+\Delta t} \dot{q} = \frac{2 - {}^t \nu}{2 + {}^t \nu} {}^{t-\Delta t} \dot{q} + 2\Delta t M^{-1} \frac{({}^{t+\Delta t} Q - {}^t F)}{2 + {}^t \nu}$$

5. ${}^{t+\Delta t} q = {}^t q + \Delta t {}^{t+\Delta t} \dot{q}$

6. Compute the error defined in Eq. 5.6-13; if $\epsilon > 1.0$, repeat steps 2-4.
7. Test for convergence.

During iteration, if the argument of the square root term in step 3 is not positive, the value of critical damping is set to zero. In addition, if the value is greater than 4, the value of ν is set equal to a value less than 2 (Underwood suggests 1.9). This is a result of limiting the maximum frequency of the system to 2.0 [43].

As seen by comparing the above process with that of the Newton-Raphson method, the procedures are very similar. The only major difference, other than the addition of inertial effects, is the use of central difference equations to linearize the equilibrium equations about the current configuration.

One slight change made in the implementation of this approach in the study arises in the method used to determine the mass matrix, specifically, the way in which the elements of K_T are determined. The technique suggested by Underwood [43] is to use numerical differentiation of F to obtain K_T rather than actually formulating the matrix. Since the program already included subroutines for formulating the stiffness matrix, the actual coefficients were used in Eq. 5.6-9.

5.7 Convergence Criteria

Since the iteration procedures used in the solution process of nonlinear problems are essentially trial and error methods, obtaining an "exact" answer is impractical if not impossible. Instead, a certain amount of error must be allowed in order to make the analysis feasible. It is important, however, that these errors do not accumulate during the solution process, leading to errors which may be undetectable [2].

In each of the solution techniques described in this chapter, convergence to the correct solution is only guaranteed monotonically with no restriction on the number of iterations required to find this solution. It is therefore necessary to set some guidelines to follow so the iteration process is terminated when a trial solution is sufficiently close to equilibrium to be taken as the true solution.

In general, there are three categories of convergence criteria which can be considered: [2]

1. Those which specify the allowable error in displacements,
2. Those which specify the error level in the residual, or out-of-balance force ${}^{t+\Delta t}R - {}^{t+\Delta t}F^{(i)}$,
3. And those which consider the total energy level of the system.

In each case, the quantity is calculated and compared with some specified tolerance level. If convergence is satisfied, the iteration process proceeds to the next load level.

In the following formulations, vector norms are used to evaluate the error level during the given iteration. The two norms used here are defined as

$$||v||_{\infty} = \max_i |v_i| \quad (5.7 - 1)$$

and

$$||v||_2 = \left(\sum_{i=1}^n |v_i|^2 \right)^{1/2} \quad (5.7 - 2)$$

Eq. 5.7-1 is known as the infinity vector norm of the vector A and Eq. 5.7-2 is the Euclidean norm of A [2]

As a result of the displacement based finite element formulation, the most common choice to control iteration is to specify that the displacements corresponding to the configuration at $t + \Delta t$ be within a certain value of the true displacements. That is, when the trial solutions begin to approach the desired equilibrium point, the incremental displacements will tend to zero. This method can be expressed as [2]

$$\frac{||\Delta q^{(i)}||_2}{||{}^{t+\Delta t}q||_2} \leq \epsilon_D \quad (5.7 - 3)$$

where ε_D is the displacement convergence tolerance specified by the user. One problem is that the displacements ${}^{t+\Delta t}q$ are not known beforehand. Bathe [2] suggests this problem can be overcome by simply using the current approximation ${}^{t+\Delta t}q^{(i)}$ as the true solution. The recommended range for ε_D is 10^{-6} to 10^{-2} [10].

A slight variation of this approach recommended by Cook [10] employs the expression

$$\frac{||\Delta q^{(i)}||_{\infty}}{||{}^{t+\Delta t}q^{(i)}||_{\infty}} \leq \varepsilon_D \quad (5.7 - 4)$$

where here Δq_i is the most recent increment and ${}^{t+\Delta t}q^i$ is the largest total displacement of the same kind. That is, if Δq_i corresponds to a rotational incremental displacement, ${}^{t+\Delta t}q^i$ corresponds to the largest rotational displacement.

Alternately, one may choose to place restrictions on the unbalanced force determined by ${}^{t+\Delta t}Q - {}^{t+\Delta t}F^{(i)}$. Bathe [2] recommends the current out-of-balance load be within a specified amount, ε_F , of the original out-of-balance force determined by

$$\frac{||{}^{t+\Delta t}Q - {}^tF^{(i)}||_2}{||{}^{t+\Delta t}Q - {}^tF||_2} \leq \varepsilon_F \quad (5.7 - 5)$$

The suggested value for ε_F is 10^{-3} .

The third approach is to consider the increment in internal energy of the system between trial configurations, that is, the amount of work done by the out-of-balance forces as compared to the initial increment in internal energy [2]. This can be expressed as

$$\frac{\Delta q^{(i)T} ({}^{t+\Delta t}Q - {}^{t+\Delta t}F^{(i-1)})}{\Delta q^{(i)T} ({}^{t+\Delta t}Q - {}^tF)} \leq \varepsilon_E \quad (5.7 - 6)$$

where the tolerance in energy ε_E is recommended to be $10 \times \varepsilon_D \times \varepsilon_F$ [2]. Additionally, a check for divergence can be made with $\varepsilon_E = 1$ [2].

Although each of the above methods of determining convergence is valid in itself, there are certain instances where one method will detect true convergence while the others cannot. The

simplest example of this is to consider the load-displacement path shown in Figure 5.14a and 5.14b. In Figure 5.14a, using the tolerance in displacements would halt iteration even when the residual force is still rather large. The opposite is true in Figure 5.14b where the out-of-balance tolerance would halt iteration while the displacements are still in error. These problems can be overcome by simply using both approaches in checking for convergence. Bathe [2] suggests that the requirements of the energy check alone will also solve this problem since both the errors in forces and displacements are incorporated in the test.

The program described in Chapter 7 allows for the use of any one of the above approaches or combinations of each. The problems of Chapter 8 primarily use the first two.

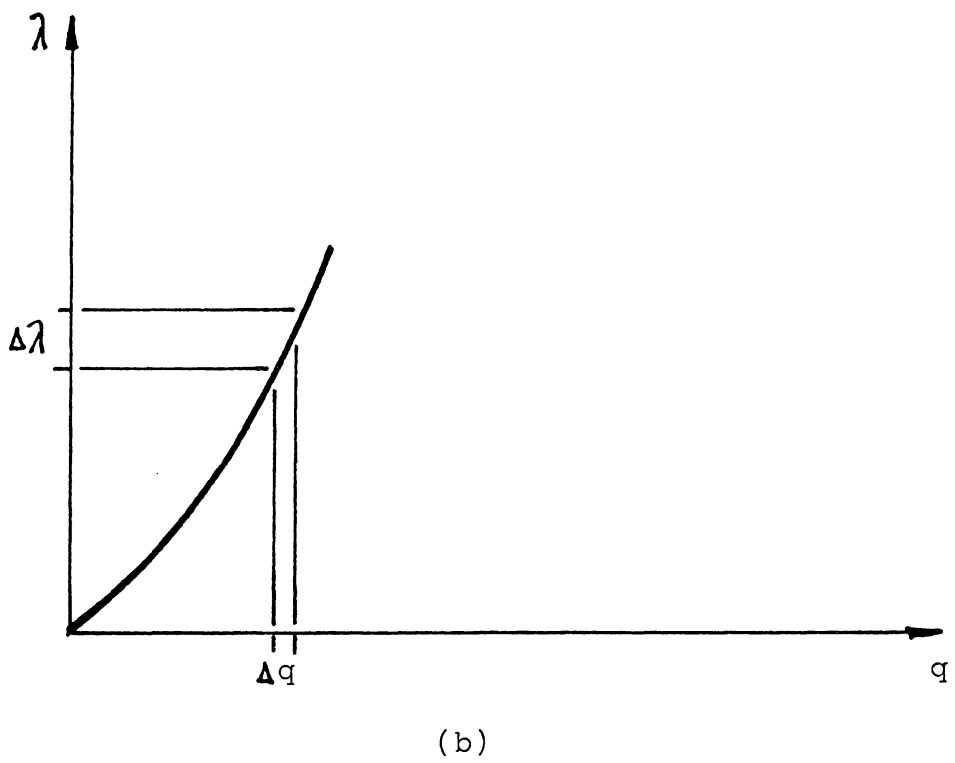
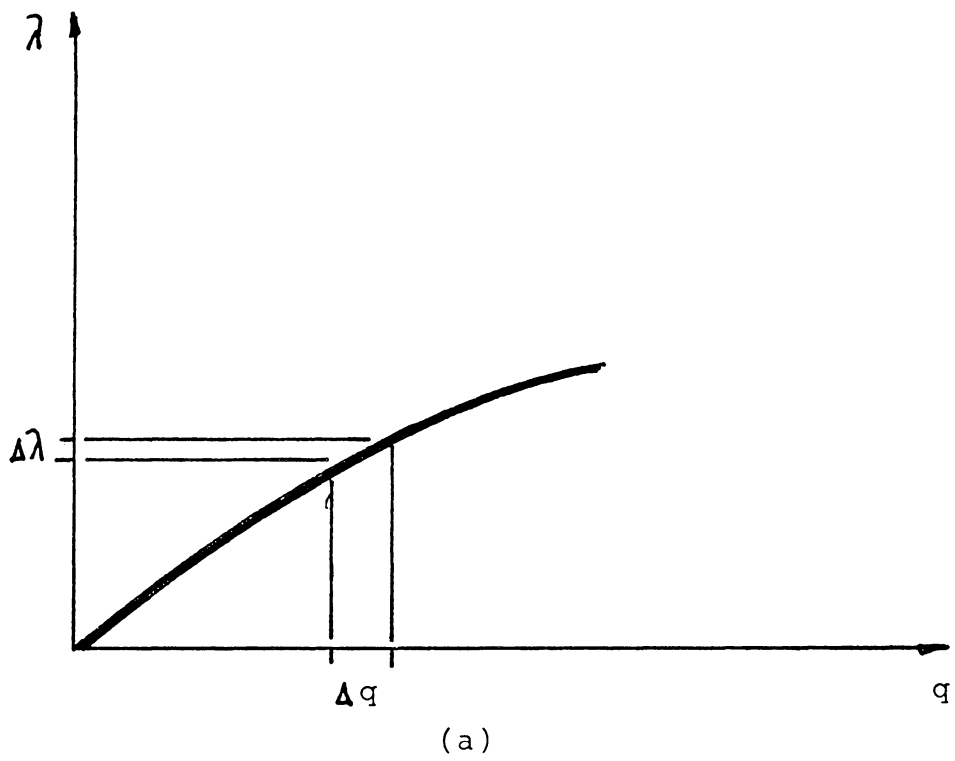


Figure 5.14. Problems of Convergence

Chapter 6

CABLE ANALYSIS

6.1 Introduction

The ability of the finite element method to handle even the most complex of problems in a well established, systematic formulation is one of its major attractions. The fundamental aspects of the problem can not be overlooked, however. This is especially true with tension structures. This chapter is designed to review the characteristics of cable structures and their response to loading. In addition, a brief discussion on the classical methods of analysis will be given. The work to follow is primarily based on Krishna [26] and Buchholdt [7].

6.2 The Freely Hanging Cable

Understanding the behavior of a freely hanging cable is very important when studying cable systems. The concepts discussed here can be directly extended to the analysis of more complex cable layouts.

Cables are characterized by a small diameter to span ratio, resulting in negligible bending or torsional stiffness. This flexibility limits the cable to developing only axial tension in response to loading. Thus the cable must attain a shape which develops only tension in the system. To demonstrate this important concept, consider a freely hanging weightless cable supported at A and B and loaded as shown in Figure 6.1. The equation of equilibrium for the cable is

$$M_x = Me_x - Hz \quad (6.2 - 1)$$

where Me_x is the simple beam bending moment at x produced by the same loading applied to a beam spanning between A and B . The assumption of zero bending stiffness in the cable results in

$$0 = Me_x - Hz \quad (6.2 - 2)$$

or

$$z = \frac{Me_x}{H} \quad (6.2 - 3)$$

Eq.6.2-3 defines the shape of the cable and clearly indicates that the profile is, to some scale, the same as the simple beam bending moment diagram. Thus the cable follows the funicular curve of the loads [26]. It is interesting to note that under the given set of loads, the shape of the cable can be completely described knowing either the value of H or a value of z for a given value of x .

Since the cable shape is a direct function of loading, changes in loading require changes in geometry to maintain equilibrium. Though this is true with any structure, the magnitude of change experienced by the cable in comparison to load change makes the analysis unique. In addition, this change of shape is nonlinear as demonstrated by considering a cable stretched between points A and B as shown in Figure 6.2a. Under a centrally located concentrated load P , the equation of equilibrium of the cable is [26]

$$P = 4(T_1 + \Delta T_1)\frac{w_1}{L} \quad (6.2 - 4)$$

or

$$w_1 = \frac{PL}{4T_2} \quad (6.2 - 5)$$

where T_1 is the initial pretension in the cable, w_1 is the displacement caused by P , and $T_2 = T_1 + \Delta T_1$. Now, apply a second load P at the center of the span and allow the cable to deflect by an additional amount w_2 (see Figure 6.2b). The equation of equilibrium then becomes

$$2P = 4(T_2 + \Delta T_2)(w_1 + w_2)\frac{1}{L} \quad (6.2 - 6)$$

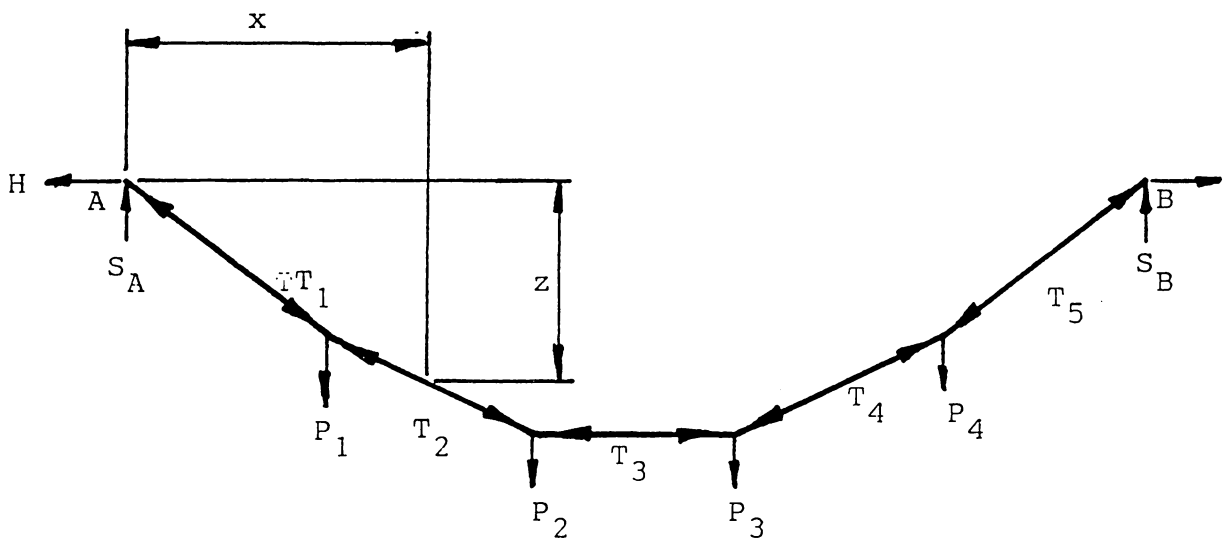


Figure 6.1. Cable Under Point Loads

or, in terms of the incremental displacement

$$w_2 = \frac{PL - 4w_1\Delta T_2}{4(T_2 + \Delta T_2)} \quad (6.2 - 7)$$

Comparing Eqs. 6.2-5 and 6.2-7, it is clear that $w_2 < w_1$ and thus the load-displacement relationship is nonlinear.

As discussed in Chapter 2 and shown above, the primary contribution to the nonlinear effects of cables is from the change in geometry under loading. In this study the cables were considered to experience only geometric nonlinearity with a linear elastic material response. This assumption can be made by observing that changes in geometry are required regardless of material behavior and in general predominate in the analysis.

6.3 The Initial Shape

As noted in Section 6.2, the shape a cable takes depends primarily on the loading imposed on it. In classical analysis of a single, freely hanging cable, the general procedure for analysis is to use a two step approach. The first step is to find the geometry of the cable under its own weight, including the effects of the permanent dead loads. The second step involves analyzing the cable under whatever live loads it will experience during its life. The analysis under dead load involves the assumption of an initial shape for the cable and then determination of the forces which must be developed to maintain this shape. Three basic forms are normally considered for the initial geometry: a catenary, a parabola, and an assembly of straight line elements.

The Catenary

The true shape of a freely hanging cable under a load q per unit length is that of a catenary. To develop the equation of a catenary, consider the segment of cable shown in Figure 6.3a. If an elemental length ds of this cable is removed and analyzed as a free body (see Figure 6.3b), then conditions of equilibrium in the vertical direction require

$$\frac{\partial}{\partial s} \left(T \frac{\partial z}{\partial s} \right) + q = 0 \quad (6.3 - 1)$$

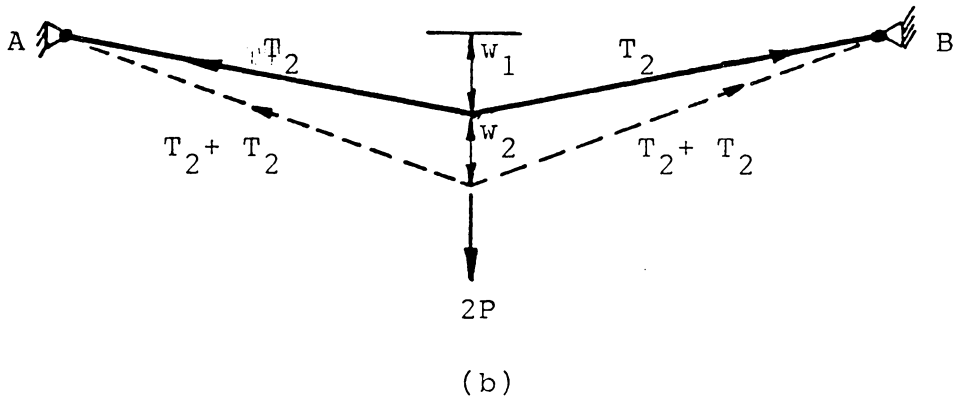
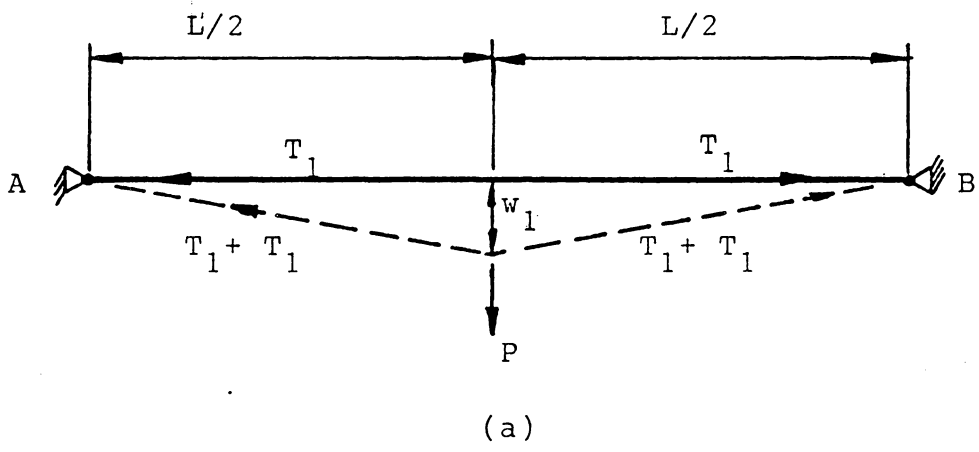


Figure 6.2. Nonlinear Deflection of a Pretensioned Cable

and in the horizontal direction

$$T \frac{\partial x}{\partial s} = \text{constant} = H \quad (6.3 - 2)$$

where

T = Axial tension in the cable.

H = Horizontal component of T .

q = Dead weight per unit length.

The differential equation of equilibrium for the cable can be obtained from Eqs. 6.3-1 and 6.3-2 in the following manner [7]:

$$\frac{\partial}{\partial s} \left(T \frac{\partial z}{\partial s} \frac{\partial x}{\partial x} \right) + q = 0 \quad (6.3 - 3)$$

$$\frac{\partial}{\partial s} \left(T \frac{\partial x}{\partial s} \frac{\partial z}{\partial x} \right) + q = 0 \quad (6.3 - 4)$$

$$\frac{\partial}{\partial s} \left(H \frac{\partial z}{\partial x} \right) + q = 0 \quad (6.3 - 5)$$

$$H \frac{\partial^2 z}{\partial s \partial x} \frac{\partial x}{\partial x} + q = 0 \quad (6.3 - 6)$$

resulting in

$$H \frac{\partial^2 z}{\partial x^2} + q \frac{\partial s}{\partial x} = 0 \quad (6.3 - 7)$$

Eq. 6.3-7 represents the equation of a catenary for the cable.

Integrating Eq. 6.3-7 twice and applying the boundary conditions

$$\begin{aligned} z &= 0 \text{ at } x = 0 \\ z &= Z \text{ at } x = X \end{aligned} \quad (6.3 - 8)$$

results in

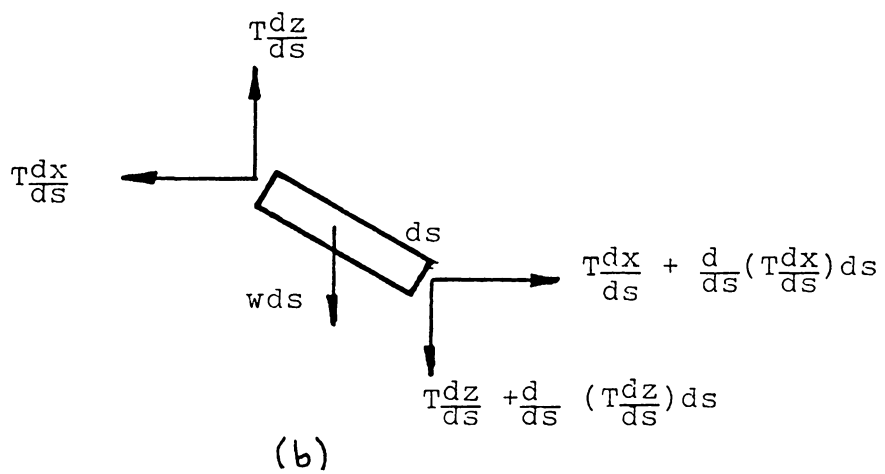
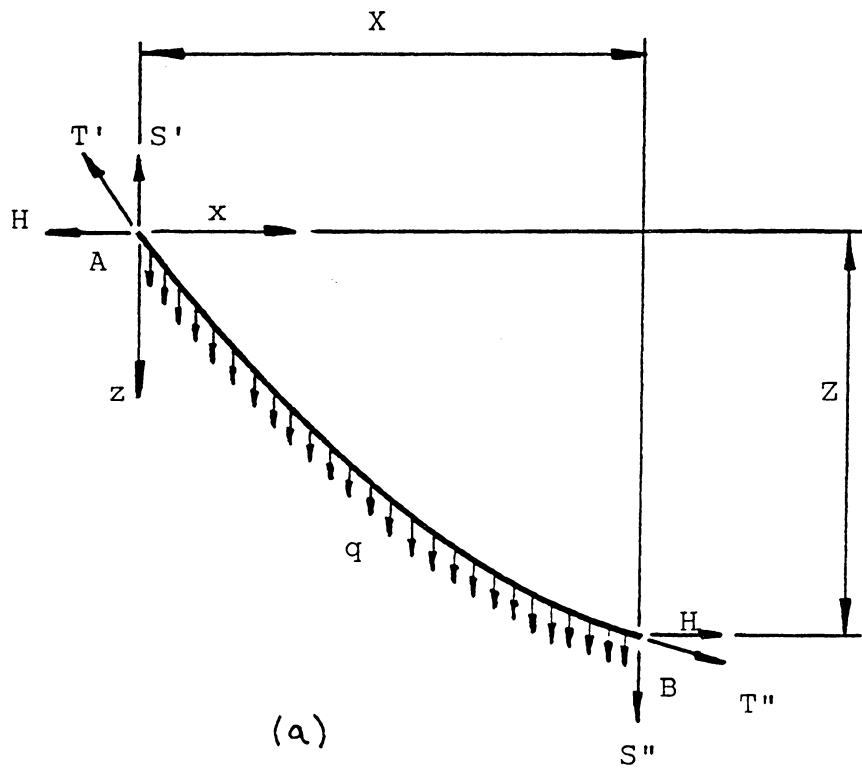


Figure 6.3. Catenary Segment of Cable

$$z = \frac{H}{q} \left[\cosh \alpha - \cosh \left(\frac{2\beta x}{X} - \alpha \right) \right] \quad (6.3 - 9)$$

where

$$\alpha = \sinh^{-1} \left[\frac{\beta(z/x)}{\sinh \beta} \right] + \beta \quad (6.3 - 10)$$

$$\beta = \frac{qX}{2h} \quad (6.3 - 11)$$

Once again, as in Section 6.2, the geometry of the cable can be completely described knowing either the value of H or the coordinates of a point on the curve other than at $x = 0$.

Using Eq. 6.3-9, the values of T' and T'' , the reactions S' and S'' , and the length of the cable can be found from (see Figure 6.3a)

$$T' = H \cosh \alpha \quad (6.3 - 12)$$

$$T'' = \frac{q}{2}(l \coth \alpha - Z) \quad (6.3 - 13)$$

$$S' = H \sinh \alpha \quad (6.3 - 14)$$

$$S'' = \frac{q}{2}(Z \coth \beta - 1) \quad (6.3 - 15)$$

$$l = \frac{2H}{q} \sinh \beta \cosh(\alpha - \beta) \quad (6.3 - 16)$$

If the cable segment represents the configuration shown in Figure 6.4, that is

$$z = 0 \text{ at } x = L \quad (6.3 - 17)$$

$$z = f \text{ at } x = L/2$$

then Eq. 6.3-9 becomes

$$z = \frac{H}{q} \left[\cosh \alpha - \cosh \left(\frac{qx}{H} - \alpha \right) \right] \quad (6.3 - 18)$$

where

$$\alpha = \beta = \frac{qL}{2H} \quad (6.3 - 19)$$

Substitution of Eqs. 6.3-17 into 6.3-18 results in

$$f = \frac{H}{q} [\cosh \alpha - 1] \quad (6.3 - 20)$$

which can be solved for H and this value used in Eq. 6.3-18 to define the cable geometry.

The Parabola

If the value of dead weight is approximated by a uniformly distributed horizontal load acting along the cable, the value of q becomes a constant with respect to x and thus the equilibrium equation becomes [7]

$$H \frac{\partial^2 z}{\partial x^2} + q = 0 \quad (6.3 - 21)$$

This approximation for q can be shown valid if the sag to span ratio is small or if superimposed dead load, acting in a uniform manner, dominates the self weight of the cable.

Eq. 6.3-21 is clearly that of a parabola. If integration of this equation is performed and the boundary conditions of Eqs. 6.3-8 are applied, the resulting expression for the cable geometry is

$$z = \frac{qx}{ZH}(X - x) + \frac{Z}{X}x \quad (6.3 - 22)$$

Here again, the complete geometry of the cable can be found knowing a coordinate on the curve or the value of H .

For a parabola, the expressions for S' , S'' , T' , and T'' become

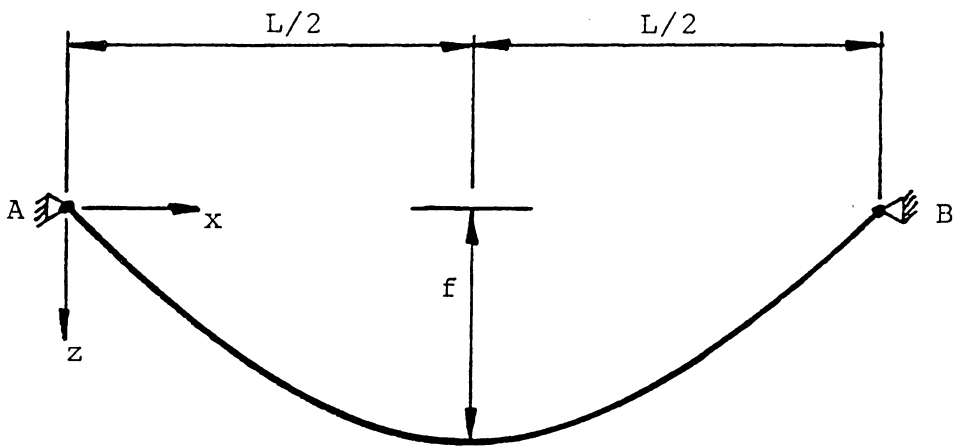


Figure 6.4. Simply Suspended Cable

$$S' = \frac{qx}{2} + H\frac{Z}{X} \quad (6.3 - 23)$$

$$S'' = \frac{qx}{2} - H\frac{Z}{X} \quad (6.3 - 24)$$

$$T' = (H^2 + S'^2)^{1/2} \quad (6.3 - 25)$$

$$T'' = (H^2 + S''^2)^{1/2} \quad (6.3 - 26)$$

The length of the cable is found from

$$l = \int_0^x \left[1 + \left(\frac{dz}{dx} \right)^2 \right]^{1/2} dx \quad (6.3 - 27)$$

which, by using a binomial expansion and neglecting higher order terms [26], gives

$$l = X \left[1 + \frac{8f^2}{3X^2} + \frac{1}{2} \frac{Z^2}{X^2} \right] \quad (6.3 - 28)$$

If the segment represents a cable with the boundary conditions of Eqs. 6.3-17, then the above equations reduce to

$$H = \frac{qL^2}{8f} \quad (6.3 - 29)$$

$$z = \frac{4fx(l-x)}{L^2} \quad (6.3 - 30)$$

$$T_{\max} = H \left[1 + 16 \left(\frac{f}{L} \right)^2 \right]^{1/2} \quad (6.3 - 31)$$

$$l = L \left[1 + \frac{8}{3} \left(\frac{f}{L} \right)^2 \right] \quad (6.3 - 32)$$

Linear Elements

Perhaps the simplest of the approaches is to consider the cable to be composed of a series of linear segments. Though this approximation may at first appear somewhat crude, as the number of segments is increased, the results compare well with the parabola or catenary. The major advantage in this approach is the simplification of the equations. This becomes a primary concern when dealing with large, complex systems.

In presenting the equations for a straight line approximation, consider one segment of the cable as shown in Figure 6.5. The equations defining the cable are clearly

$$z = \frac{Z}{X}x \quad (6.3 - 33)$$

$$l = (X^2 + Z^2)^{1/2} \quad (6.3 - 34)$$

$$T = (H^2 + S^2)^{1/2} \quad (6.3 - 35)$$

where S is determined from the approximation for q . That is

$$S = \frac{ql}{2} \quad (6.3 - 36)$$

or

$$S = \frac{qX}{2} \quad (6.3 - 37)$$

The choice between the three different shapes depends to a large extent on the accuracy desired in the results versus the expense of the analysis. While the straight line elements are by far the simplest to employ, the accuracy of results is questionable unless a large number of elements is used. The differences between the parabolic shape and the catenary are small for a majority of cable systems, thus the use of this approach is more common than the catenary [7,26]. Yet the catenary equations cannot be overlooked for cables with large sag to span ratios or if a high degree of accu-

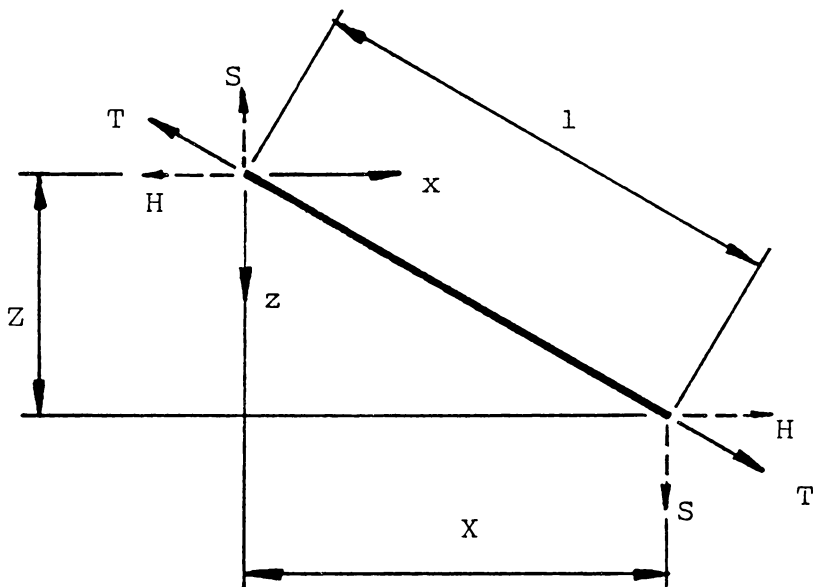


Figure 6.5. Straight Line Cable Segment

racy is desired. In addition, the equations as presented do not take into consideration the effects of cable extension or thermal properties. These can be easily accounted for; see for instance [7,26,40].

6.4 Methods of Analysis

Though the primary emphasis of this study is on the application of nonlinear finite element techniques to the analysis of cables, the study would be incomplete without a brief discussion of other approaches to the problem. The equations presented in Section 6.3 lead to simple hand calculations for single cables even if the effects of elongation or temperature change are considered. However, their extension to more elaborate systems can become quite complex [35,40].

There are generally two approaches used in the solution of cable systems. The first of these involves extending the closed-form solutions while making certain assumptions to reduce complexity. These assumptions include analyzing the cables individually and ignoring the two way action of nets, considering only vertical deflections, or assuming orthogonality in plane at all times [35,40]. The equations of equilibrium are then solved using various analysis techniques such as the central difference method. This general approach can give good results; however, it does not adapt well to computer implementation.

The second approach is to use the matrix-displacement method combined with Newton-Raphson iteration in much the same approach used in the finite element method. This technique is more commonly used in the analysis of cable nets where the focus is on the system as a whole rather than the individual elements. This approach employs straight line segments for the cable, concentrating on the displacements of joints to determine cable forces. With almost all cable nets being pretensioned prior to loading, the use of line elements is quite accurate since the cable assumes a straight line under axial tension only. Furthermore, all loads on the net are assumed to act at the node and not on the cables. This method can be implemented on the computer quite easily and thus is the preferred choice among current authors [7,26,40].

Chapter 7

PROGRAM DEVELOPMENT

7.1 Introduction

The program developed for use in this study is an adaptation of programs written by Butler [9] and Navarro [33] for their work in nonlinear analysis. The structure of the program, shown in Figure 7.1, follows the form presented by Holzer [24] in his work with matrix structural analysis. The intent of the program is to implement the displacement based finite element method in a clear, well structured format which is easy to follow, particularly for those familiar with Holzer's work.

In this chapter, a brief description of the individual subroutines is given along with a Nassi-Schneiderman (NS) Diagram [24] for each. The program listing itself is presented in Appendix B which includes a description for the variables used and a brief description of the format required for input. The coding language used in the program is standard Fortran 77.

The solution routines (SOLVE, FACTOR, REDUCE, and BACSUB) are borrowed directly from Bathe [2] and therefore are not included here.

7.2 Program Subroutines

Main Program

Function: Initialize dimensioning control parameters MX, MXNEQ, and MXNA. Initialize iteration control variables and echo all settings. Read and echo initial force distribution factor QI, maximum distribution factor QIMAX, and load increment DQI. Read and echo degree of freedom to be plotted (if desired), IMPOR. Call DATA and one of the iteration methods: NEWRAP, RIKWEM, ORTHOG, or DYNAM.

NS Diagram: Figure 7.2

Subroutine DATA

Function: Input control routine for structural data. Read and echo number of elements NE, number of joints NJ, and element type, MTYP.

NS Diagram: Figure 7.3

Subroutine STRUCT

Function: Input and process data which defines the configuration of the structure. Read and echo member incidences, MINC, for each element. Initialize the joint code matrix JCODE to unity. Read and echo joint constraints and initialize the corresponding locations of JCODE to zero. Call CODES, DETMAX, and PROP.

NS Diagram: Figure7.4

Subroutine CODES

Function: Form the joint code matrix JCODE and the member code matrix MCODE.

NS Diagram: Figure7.5

Subroutine DETMAX

Function: Determine the length of the tangent stiffness vector (in skyline storage [2]) NKT, the vector which stores the addresses of the main diagonal terms MAXA, and the number of elements below the skyline of each column.

NS Diagram: Figure 7.6

Subroutine PROP

Function: Read and echo joint coordinates X. For each element, compute the element length ELEN(I), read and echo AREA(I) and modulus of elasticity EMOD(I).

NS Diagram: Figure 7.7

Subroutine LOAD

Function: Controlling module for input of loading conditions. Initialize joint load vectors QJ and QP to zero. Call JLOAD and MACT.

NS Diagram: Figure 7.8

Subroutine JLOAD

Function: Read and echo joint number, joint direction, and joint force for incremental joint loads QJ and non-incremental joint loads QP.

NS Diagram: Figure 7.9

Subroutine MACT

Function: For each member action (see Section 4.5), read and echo member number MN(I), action type MAT(I), and the value of the action ACT(I). Initialize internal stress matrix TAU to zero. Read and echo initial pretension force (if any) for each element.

NS Diagram: Figure 7.10

Subroutine NEWRAP

Function: Perform Newton-Raphson or Modified Newton-Raphson iterations until the desired load level is reached or the maximum number of iterations is exceeded. Call FORCES, UPDATF, STIFF, SOLVE, UPDATC, TEST, and RESULT. In addition, DYNAM can be accessed for regions in which the stiffness matrix becomes singular.

NS Diagram: Figure 7.11

Subroutine RIKWEM

Function: Perform the Modified Riks-Wempner iterative process to determine the desired solution. Call FORCES, UPDATF, STIFF, SOLVE, DOTPRD, UPDATC, TEST, and RESULT.

NS Diagram: Figure 7.12

Subroutine ORTHOG

Function: Use Orthogonal Trajectory Accession to iterate along the equilibrium path. Call FORCES, UPDATF, STIFF, SOLVE, DOTPRD, UPDATC, TEST, and RESULT.

NS Diagram: Figure 7.13

Subroutine DYNAM

Function: Use a Dynamic Relaxation approach to iterate along the equilibrium curve. Call STIFF, FORCES, UPDATF, UPDATC, TEST, and RESULT.

NS Diagram: Figure 7.14

Subroutine FORCES

Function: Control module for determining internal forces of each member and updating the external joint loads. Call ELEMFI and EXTRN1, or ELEMFI2 and EXTRN2.

NS Diagram: Figure 7.15

Subroutine ELEMFI

Function: Determine the current level of internal stress for the two node element. Compute the global element forces FG and sum them into the system internal force vector F.

NS Diagram: Figure 7.16

Subroutine EXTRN1

Function: Update the external loads on the member for the two node element.

NS Diagram: Figure 7.17

Subroutine ELEMFI2

Function: Compute the current internal stress of the member for a three node element using Gaussian quadrature [2]. Compute the global element forces FG and sum into the system internal force vector F. Call SHAPE.

NS Diagram: Figure 7.18

Subroutine EXTRN2

Function: Update the external load vector for the three node element.

NS Diagram: Figure 7.19

Subroutine STIFF

Function: Control module for determining the tangent stiffness vector SKT in the current configuration. Call CBL1 or CBL2 and ASSEMS.

NS Diagram: Figure 7.20

Subroutine CBL1

Function: Compute the element tangent stiffness matrix SE for the two node element using the process described in Section 4.3.

NS Diagram: Figure 7.21

Subroutine CBL2

Function: Compute the element tangent stiffness matrix SE for the three node element using Gaussian quadrature [2]. Call SHAPE.

NS Diagram: Figure 7.22

Subroutine ASSEMS

Function: Map the element stiffness matrix ~~SE~~ into the system tangent stiffness vector SKT using MAXA and MCODE.

NS Diagram: Figure 7.23

Subroutine TEST

Function: Control module for convergence tests. Call ENERGY, UNBALF, DISPL1, and DISPL2 when their corresponding tolerances are less than 1.0. Test for divergence if TOLENE = 1.0.

NS Diagram: Figure 7.24

Subroutine ENERGY

Function: Compute the error in incremental internal energy and test for convergence. If DIVER > 0 then test for divergence.

NS Diagram: Figure 7.25

Subroutine UNBALF

Function: Test for convergence in unbalanced forces by computing the norm of out-of-balance forces.

NS Diagram: Figure 7.26

Subroutine DISPLI

Function: Compute the Euclidean vector norm of displacements and test for convergence in displacements.

NS Diagram: Figure 7.27

Subroutine RESULT

Function: Control module for displaying results of the current iteration. Initialize the joint reaction matrix P to zero and call JOINTF and OUTPUT.

NS Diagram: Figure 7.28

Subroutine JOINTF

Function: Determine the joint forces from the global element forces FG and sum into P.

NS Diagram: Figure 7.29

Subroutine OUTPUT

Function: Compute the joint displacements DJ from D using JCODE. Output desired results determined by the value of NPRINT.

NS Diagram: Figure 7.30

Subroutine UPDATC

Function: Update the joint coordinates X by adding the current incremental displacements to the previous joint coordinates.

NS Diagram: Figure 7.31

Subroutine UPDATF

Function: Update the current load level QT by multiplying the incremental joint load load vector QJ by the load increment QI and adding the constant joint load vector QP to the result.

NS Diagram: Figure 7.32

Subroutine SHAPE

Function: Compute the linear and nonlinear strain-displacement matrices of the three node element at the Gauss point PR .

NS Diagram: Figure 7.33

Subroutine DOTPRD

Function: Compute the dot product of the two vectors $DOT1$ and $DOT2$.

NS Diagram: Figure 7.34

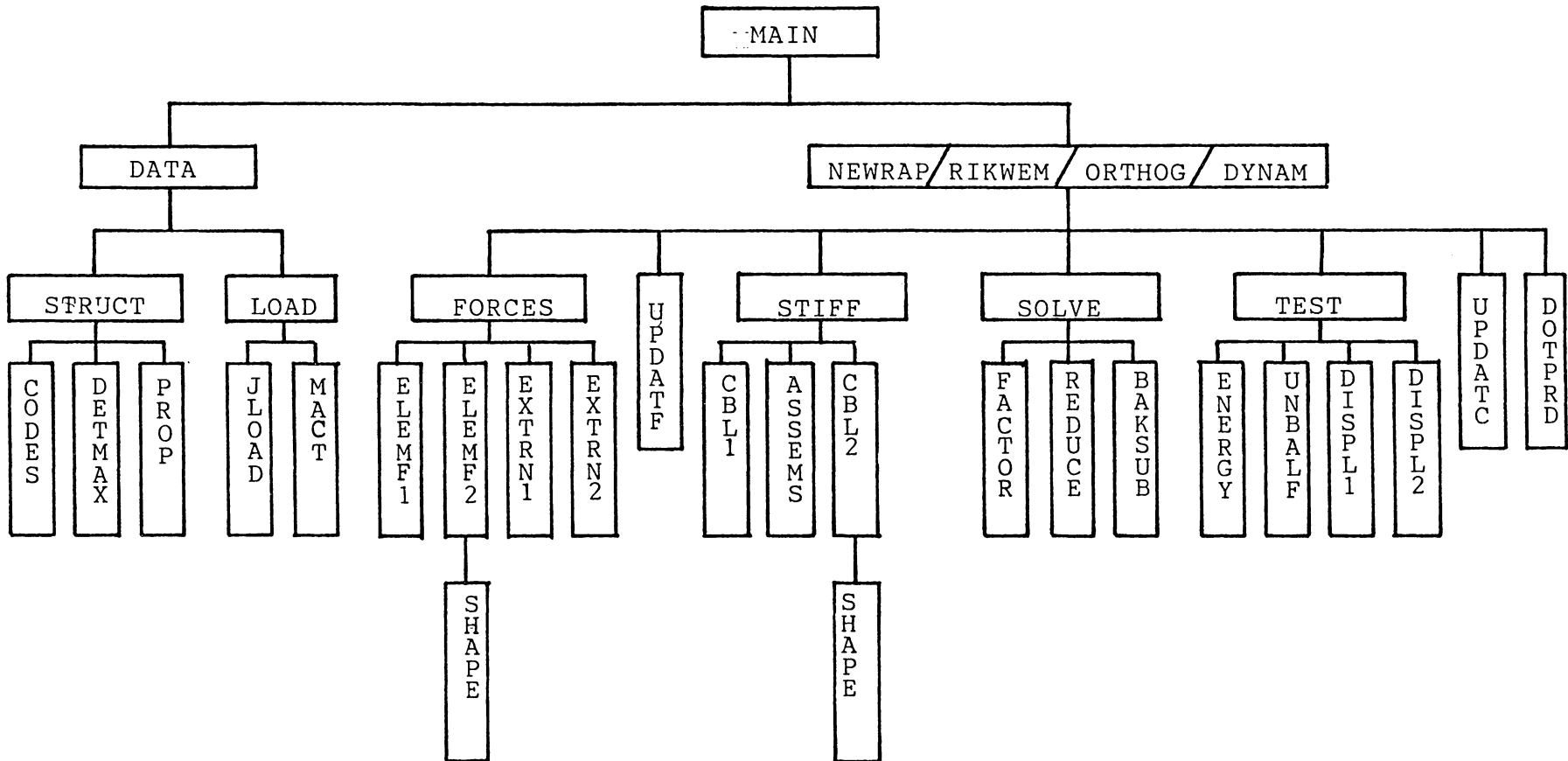


Figure 7.1 : Program Structure

| | | | |
|---|-------------|-------------|------------|
| Initialize dimension parameters: MX, MXNEQ, MXNA | | | |
| Initialize iteration parameters; iteration method, iteration limits,convergence criteria | | | |
| Read and echo: QI, QIMAX, DQI | | | |
| Call DATA | | | |
| Do case, ITEMTH | | | |
| ITEMTH = 1 | ITEMTH = 2 | ITEMTH = 3 | ITEMTH = 4 |
| Call NEWRAP | Call RIKWEM | Call ORTHOG | Call DYNAM |

Figure 7.2: Subroutine MAIN

| | |
|-----------------------------|---------------------|
| Read and echo: NE, NJ, MTyp | |
| If NE < MX and NJ < MX | |
| Then | Else |
| Call STRUCT | Print error message |
| Call LOAD | STOP |

Figure 7.3: Subroutine DATA

| | |
|---|---|
| If MTyp = 1 | |
| Then | Else |
| Read and echo MINC(J,I) J = 1,2 I = 1,NE | Read and echo MINC(J,I) J = 1,3 I = 1,NE |
| Initialize JCODE(L,J) = 1 | L = 1,3 J = 1,NJ |
| Read joint constraint JNUM, JDIR | |
| While JNUM ≠ 0 | |
| Print JNUM JDIR | |
| JCODE(JNUM, JDIR) = 0 | |
| Read JNUM, JDIR | |
| Call CODES | |
| Call DETMAX | |
| Call PROP | |

Figure 7.4: Subroutine STRUCT

| |
|---|
| NEQ = 0 |
| Do for J = 1, NJ |
| Do for L = 1, 3 |
| If JCODE(L, J) ≠ 0 |
| Then |
| NEQ = NEQ + 1 |
| JCODE(L, J) = NEQ |
| Else |
| Initialize NN: NN = 2 If MTYP = 1 Else NN = 3 |
| Do for I = 1, NE |
| Do for L = 1, NN |
| J = MINC(L, I) |
| N = 3*J, M = N-1, K = M-1 |
| MCODE(K, I) = JCODE(1, J) |
| MCODE(M, I) = JCODE(2, J) |
| MCODE(N, I) = JCODE(3, J) |

Figure 7.5: Subroutine CODES

| | |
|---|------------|
| Initialize KHT = 0 | I = 1, NEQ |
| Initialize NN: NN = 2 If MTYP = 1 Else NN = 3 | |
| Do for I = 1, NE | |
| J = 1 | |
| While MCODE(J, I) = 0 | |
| J = J + 1 | |
| MIN = MCODE(J, I) | |
| J = J + 1 | |
| Do for L = J, NN*3 | |
| K = MCODE(L, I) | |
| If K ≠ 0 | |
| Then | Else |
| KHT(K) = max of (KHT(K), (K-MIN)) | |
| MAXA(1) = 1 | |
| Do for I = 1, NEQ | |
| MAXA(I+1) = MAXA(I) + KHT(I) + 1 | |
| NKT = MAXA(NEQ + 1) - 1 | |

Figure 7.6: Subroutine DETMAX

| | |
|--|----------------------|
| Read and echo joint coordinates: X(1,J), X(2,J), X(3,J) J = 1,NJ | |
| If MTYP = 1 | |
| Then | Else |
| Do for I = 1,NE | Do I = 1,NE |
| J = MINC(1,I) | Compute ELENG(I) |
| K = MINC(2,I) | using Eq. 4.4-4 |
| EL1 = X(1,K)-X(1,J) | Read AREA(I),EMOD(I) |
| EL2 = X(2,K)-X(2,J) | Print AREA(I), |
| EL3 = X(3,K)-X(3,J) | EMOD(I), ELENG(I) |
| ELENG(I) = SQRT(EL1 ² + EL2 ² + EL3 ²) | |
| Read AREA(I), EMOD(I) | |
| Print AREA(I), EMOD(I), ELENG(I) | |

Figure 7.7: Subroutine PROP

| | |
|------------------------------------|------------|
| Initialize QJ(I) = 0 and QP(I) = 0 | I = 1, NEQ |
| Call JLOAD | |
| Call MACT | |

Figure 7.8: Subroutine LOAD

| | | |
|-------------------------|------------------|------|
| Read JNUM, JDIR, FORCE | | |
| Then | If JNUM \neq 0 | Else |
| Print headings | | |
| While JNUM \neq 0 | | |
| Print JNUM, JDIR, FORCE | | |
| K = JCODE(JDIR, JNUM) | | |
| QJ(K) = FORCE | | |
| Read JNUM, JDIR, FORCE | | |
| Read JNUM, JDIR, FORCE | | |
| Then | If JNUM \neq 0 | Else |
| Print headings | | |
| While JNUM \neq 0 | | |
| Print JNUM, JDIR, FORCE | | |
| K = JCODE(JDIR, JNUM) | | |
| QP(K) = FORCE | | |
| Read JNUM, JDIR, FORCE | | |

Figure 7.9: Subroutine JLOAD

| | |
|------------------------------------|-----------|
| I = 1 | |
| Read MN(I), MAT(I), ACT(I) | |
| If MN(I) ≠ 0 | |
| Then | Else |
| Print headings | |
| While MN(I) ≠ 0 | |
| Print MN(I), MAT(I), ACT(I) | |
| I = I + 1 | |
| Read MN(I), MAT(I), ACT(I) | |
| NAT = NAT - 1 | |
| Initialize TAU(1,I) = 0. | I = 1, NE |
| Read MEM, TAU(1, MEM) | |
| If MEM < NE | |
| Then | Else |
| Print headings | |
| While MEM < NE | |
| Print MEM, TAU(1, MEM) | |
| Read MEM, TAU(1, MEM) | |
| If MTYP = 2 | |
| Then | Else |
| TAU(2,I) = TAU(1,I) | |
| TAU(3,I) = TAU(1,I) I = 1, NE | |

Figure 7.10: Subroutine MACT

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|-------------------------------------|-------------------------|------|------------|--|--|------------|--|--|------------|--|--|------|-------------------------------------|------|------------|--|--|-----------|--|--|-----------------|--|--|------|----------------------------------|------|---------------------|--|--|------|--|--|------|-----------------|------|----------------|-----------|--|---------------------|-----------|--|---------------|-----------|--|
| Call FORCES | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Call UPDATF | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| While $QI \leq QIMAX$ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ITENUM = ITEUPD | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ITECNT = 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| INCONV = 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| NTRY = 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| KSTBL = 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| While $INCONV \neq 0$ and $ITECNT \leq ITEMAX$ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <table border="1"> <tr><td>Then</td><td>If $ITENUM \geq ITEUPD$</td><td>Else</td></tr> <tr><td>Call STIFF</td><td></td><td></td></tr> <tr><td>ITENUM = 0</td><td></td><td></td></tr> <tr><td>Call SOLVE</td><td></td><td></td></tr> <tr><td>Then</td><td>If $KSTBL \neq 0$ and $NTRY \leq 2$</td><td>Else</td></tr> <tr><td>Call DYNAM</td><td></td><td></td></tr> <tr><td>KSTBL = 0</td><td></td><td></td></tr> <tr><td>ITENUM = ITEUPD</td><td></td><td></td></tr> <tr><td>Then</td><td>If $KSTBL \neq 0$ and $NTRY > 2$</td><td>Else</td></tr> <tr><td>Print error message</td><td></td><td></td></tr> <tr><td>STOP</td><td></td><td></td></tr> <tr><td>Then</td><td>If $ITECNT = 0$</td><td>Else</td></tr> <tr><td>DDI(I) = DD(I)</td><td>I = 1,NEQ</td><td></td></tr> <tr><td>D(I) = D(I) + DD(I)</td><td>I = 1,NEQ</td><td></td></tr> <tr><td>FPI(I) = F(I)</td><td>I = 1,NEQ</td><td></td></tr> </table> | Then | If $ITENUM \geq ITEUPD$ | Else | Call STIFF | | | ITENUM = 0 | | | Call SOLVE | | | Then | If $KSTBL \neq 0$ and $NTRY \leq 2$ | Else | Call DYNAM | | | KSTBL = 0 | | | ITENUM = ITEUPD | | | Then | If $KSTBL \neq 0$ and $NTRY > 2$ | Else | Print error message | | | STOP | | | Then | If $ITECNT = 0$ | Else | DDI(I) = DD(I) | I = 1,NEQ | | D(I) = D(I) + DD(I) | I = 1,NEQ | | FPI(I) = F(I) | I = 1,NEQ | |
| Then | If $ITENUM \geq ITEUPD$ | Else | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Call STIFF | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ITENUM = 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Call SOLVE | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Then | If $KSTBL \neq 0$ and $NTRY \leq 2$ | Else | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Call DYNAM | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| KSTBL = 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ITENUM = ITEUPD | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Then | If $KSTBL \neq 0$ and $NTRY > 2$ | Else | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Print error message | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| STOP | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Then | If $ITECNT = 0$ | Else | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DDI(I) = DD(I) | I = 1,NEQ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| D(I) = D(I) + DD(I) | I = 1,NEQ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| FPI(I) = F(I) | I = 1,NEQ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Figure 7.11: Subroutine NEWRAP

| |
|--|
| Call UPDATC |
| Call FORCES |
| Call TEST |
| ITENUM = ITENUM + 1 |
| ITECNT = ITECNT + 1 |
| QI = QI + DQI |
| FP(I) = F(I) I = 1, NEQ |
| Call RESULT |
| If INCONV \neq 0 |
| Then Else |
| Print error message |
| STOP |
| Call UPDATF |

Figure 7.11: Subroutine NEWRAP (con't)

| |
|--|
| ITECNT = 0 |
| KSTBL = 0 |
| NTRY = 0 |
| ITP = 4 |
| Call FORCES |
| While $QI \leq QIMAX$ and $ITECNT \leq ITEMAX$ |
| Call STIFF |
| Call SOLVE: Find DD01 |
| If $KSTBL \neq 0$ |
| Then |
| Else |
| STOP |
| If ITECNT = 0 |
| Then |
| Else |
| DS = DQI* (DOTPRD(DD01,DD01,NEQ) + 1) |
| DQI = DS/ (DOTPRD(DD01,DD01,NEQ) + 1) |
| TEMP = DQI* (DOTPRD(DD01,DD01,NEQ) + DQI1) |
| If TEMP > 0 |
| Then |
| Else |
| SGN = 1. |
| SGN = -1. |
| DQI = SGN*DQI |
| DQI1 = DQI |
| DD0(I) = DQI*DD01(I) |
| D(I) = D(I) + DD0(I) |
| DDP(I) = DD0(I) |
| I = 1,NEQ |
| QI = QI + DQI |
| Call UPDATC |
| Call FORCES |

Figure 7.12: Subroutine RIKWEM

| |
|---|
| INCONV = 1 |
| ITENUM = ITEUPD |
| IT = 0 |
| While INCONV \neq 0 and IT \leq ITEMAX |
| Call UPDATF |
| If ITENUM \geq ITEUPD |
| Then |
| Else |
| Call STIFF |
| ITENUM = 0 |
| Call SOLVE: Find DD1 |
| Call SOLVE: Find DD2 |
| DQI = $-(\text{DOTPRD}(\text{DD0}, \text{DD2}, \text{NEQ}) /$ $(\text{DOTPRD}(\text{DD0}, \text{DD1}, \text{NEQ}) + \text{DQI1}))$ |
| DD(I) = DQI*DD1(I) + DD2 |
| D(I) = D(I) + DD(I) I = 1, NEQ |
| DDP(I) = DDP(I) + DD(I) |
| Call UPDATC |
| Call FORCES |
| QI = QI + DQI |
| Call TEST |
| ITENUM = ITENUM + 1, IT = IT + 1 |
| ITECNT = ITECNT + 1 |
| Call RESULT |
| If INCONV \neq 0 |
| Then |
| Else |
| Print error message |
| STOP |
| DS = DS*SQRT(ITEDES/IT) |

Figure 7.12: Subroutine RIKWEM (con't)

| |
|---|
| ITECNT = 0 |
| KSTBL = 0 |
| NTRY = 0 |
| LCNT = 0 |
| DS = DQI |
| INCONV = 1 |
| Call FORCES |
| While INCONV \neq 0 and LCNT \leq 10 |
| Call UPDATF |
| Call STIFF |
| Call SOLVE: Find DD1 |
| Call SOLVE: Find DD2 |
| If KSTBL \neq 0 |
| Then |
| STOP |
| Else |
| DQI = $-(\text{DOTPRD}(\text{DD1}, \text{DD2}, \text{NEQ}) /$ $(\text{DOTPRD}(\text{DD1}, \text{DD1}, \text{NEQ}) + \text{DQI1}))$ |
| DD(I) = DQI*DD1(I) + DD2(I) |
| D(I) = D(I) + DD(I) I = 1, NEQ |
| Call UPDATC |
| Call FORCES |
| QI = QI + DQI |
| Call TEST |
| LCNT = LCNT + 1 |
| Initialize DDP(I) = 0 I = 1, NEQ |
| DQI1 = 1 |
| While QI \leq QIMAX and ITECNT \leq ITEMAX |
| Call STIFF |
| Call SOLVE: Find DD01 |

Figure 7.13: Subroutine ORTHOG

| | |
|---|-----------|
| If KSTBL \neq 0 | |
| Then | Else |
| STOP | |
| DQI = DS/SQRT(DOTPRD(DD01,DD01,NEQ) + 1) | |
| TEMP = DQI*(DOTPRD(DD01,DD01,NEQ) + DQI1) | |
| If TEMP > 0 | |
| Then | Else |
| SGN = 1. | SGN = -1. |
| DQI = SGN*DQI | |
| DQI1 = DQI | |
| DD0(I) = DQI*DD01(I) | I = 1,NEQ |
| D(I) = D(I) + DD0(I) | |
| DDP(I) = DD0(I) | |
| QI = QI + DQI | |
| Call UPDATC | |
| Call FORCES | |
| INCONV = 1 | |
| ITENUM = ITEUPD | |
| IT = 0 | |
| While INCONV \neq 0 and IT \leq ITEMAX | |
| Call UPDATF | |
| If ITENUM \geq ITEUPD | |
| Then | Else |
| Call STIFF | |
| ITENUM = 0 | |
| Call SOLVE: Find DD1 | |
| Call SOLVE: Find DD2 | |
| DQI = -(DOTPRD(DD0,DD2,NEQ)/ (DOTPRD(DD0,DD1,NEQ) + DQI1)) | |
| DD(I) = DQI*DD1(I) + DD2 | I = 1,NEQ |
| D(I) = D(I) + DD(I) | |
| DDP(I) = DDP(I) + DD(I) | |

Figure 7.13: Subroutine ORTHOG (con't)

| |
|--|
| Call UPDATC |
| Call FORCES |
| QI = QI + DQI |
| Call TEST |
| ITENUM = ITENUM + 1 |
| IT = IT + 1 |
| ITECNT = ITECNT + 1 |
| Call RESULT |
| If INCONV \neq 0 |
| Then Else |
| Print error message |
| STOP |

Figure 7.13: Subroutine ORTHOG (con't)

| | | |
|--|--|--|
| $DP(I) = D(I)$ $D(I) = D(I) + DT * Z(I)$ $DD(I) = D(I) - DP(I)$ $FPI(I) = F(I)$ | | $I = 1, NEQ$ |
| If $LNT \neq 0$ | | Else |
| Then | | |
| Compute error from Eq.5.6-13 | | |
| If $E < 1$ | | |
| Then | | |
| Call UPDATC Call FORCES Call TEST $LNT = LNT + 1$ | | $MFLG = 1$ $NREF = NREF + 1$ Stop if $NREF > 10$ |
| $FP(I) = F(I)$ Call RESULT | | $I = 1, NEQ$ |
| If $INCONV \neq 0$ | | Else |
| Then | | |
| STOP | | |
| $QI = QI + DQI$ Call UPDATF | | |

Figure 7.14: Subroutine DYNAM

| | | | |
|--------------------------------|--|-----------------|--|
| Initialize F(I) = 0. I = 1,NEQ | | | |
| If MTYP = 1 | | | |
| Then | | Else | |
| Do for I = 1,NE | | Do for I = 1,NE | |
| Call ELEMFI | | Call ELEMFI | |
| If NAT ≠ 0 | | If NAT ≠ 0 | |
| Then | | Then | |
| Else | | Else | |
| Call EXTRN1 | | Call EXTRN2 | |

Figure 7.15: Subroutine FORCES

| | |
|---|-------------|
| Do for J = 1,6 | |
| L = MCODE(J,I) | |
| If L ≠ 0 | |
| Then | Else |
| DDE(J) = DD(L) | DDE(J) = 0. |
| Do for L = 1,2 | |
| XL(L) = X(1,J), YL(L) = X(2,J), ZL(L) = X(3,J) | |
| EL2 = ELENG(I) ² | |
| A = (XL(2)-XL(1))/EL2, B = (YL(2)-YL(1))/EL2, C = (ZL(2)-ZL(1))/EL2 | |
| EPS = A*(DDE(4)-DDE(1)) + B*(DDE(5)-DDE(2)) + C*(DDE(6)-DDE(3)) | |
| TAU(1,I) = TAU(1,I) + EMOD(I)*EPS | |
| PT = TAU(1,I)*AREA(I) | |
| FG(4,I) = PT*A*ELENG(I) FG(5,I) = PT*B*ELENG(I) FG(6,I) = PT*C*ELENG(I) FG(1,I) = -FG(4,I) FG(2,I) = -FG(5,I) FG(3,I) = -FG(6,I) | |
| Do for J = 1,6 | |
| L = MCODE(J,I) | |
| If L ≠ 0 | |
| Then | Else |
| F(L) = F(L) + FG(J,I) | |

Figure 7.16: Subroutine ELEMFI

| | |
|---|------------|
| Initialize Q(I) = QJ(I) | I = 1, NEQ |
| Initialize FF(I) = 0. | I = 1, 6 |
| Do for I = 1, NAT | |
| EL = MN(I) | |
| If MAT(I) = 1 | |
| Then | Else |
| DUM1 = ACT(I)*ELENG(EL)/2 | |
| FF(2) = DUM1 FF(5) = DUM1 | |
| If MAT(I) = 2 | |
| Then | Else |
| Do for J = 1, 2 | |
| K = MINC(J, EL) | |
| XL(J) = X(1, K), ZL(J) = X(3, K) | |
| EL1 = XL(2) - XL(1) EL3 = ZL(2) - ZL(1) EL13 = SQRT(EL1 ² + EL3 ²) | |
| FF(2) = EL13*ACT(I)/2 FF(5) = FF(2) | |
| Do for J = 1, 6 | |
| L = MCODE(J, I) | |
| If L ≠ 0 | |
| Then | Else |
| Q(L) = Q(L) + FF(J) | |

Figure 7.17: Subroutine EXTRN1

| | |
|--|-------------|
| Do for J= 1,9. | |
| L = MCODE(J,I) | |
| If L ≠ 0 | |
| Then | Else |
| DDE(J) = DD(L) | DDE(J) = 0. |
| Do for NA = 1,NGAUSS | |
| EPS = 0. | |
| Call SHAPE | |
| DV = WGT(NA,NGAUSS)*AREA(I)*ELENG(I)/2 | |
| Do for J = 1,9 | |
| EPS = EPS + BL(J)*DDE(J) | |
| TAU(NA,I) = EMOD(*I)*EPS | |
| Do for J = 1,9 | |
| FG(J,I) = FG(J,I) + BL(J)*TAU(NA,I)*DV | |
| Do for J = 1,9 | |
| L = MCODE(J,I) | |
| If L ≠ 0 | |
| Then | Else |
| F(L) = F(L) + FG(J,I) | |

Figure 7.18: Subroutine ELEM2

| | | | |
|--|--|---------------|--|
| Initialize Q(I) = QJ(I) | | I = 1, NEQ | |
| Do for I = 1, NAT | | | |
| EL = MN(I) | | | |
| Initialize FF(J) = 0. | | J = 1, 9 | |
| | | If MAT(I) = 1 | |
| Then | | Else | |
| DUM1 = ACT(I)*ELENG(EL) | | | |
| FF(2) = DUM1/6 | | | |
| FF(5) = DUM1*2/3 | | | |
| FF(8) = FF(2) | | | |
| | | If MAT(I) = 2 | |
| Then | | Else | |
| Do for K = 1, 3 | | | |
| J = MINC(K, EL) | | | |
| XL(K) = X(1, J), ZL(K) = X(2, J) | | | |
| EL1A = XL(2)-XL(1), EL1B = XL(3)-XL(2) | | | |
| EL3A = ZL(2)-ZL(1), EL3B = ZL(3)-ZL(2) | | | |
| EL13A = SQRT(EL1A ² + EL3A ²) | | | |
| EL13B = SQRT(EL1B ² + EL3B ²) | | | |
| FF(2) = EL13A*ACT(I)/2 | | | |
| FF(8) = EL13B*ACT(I)/2 | | | |
| FF(5) = FF(2) + FF(8) | | | |
| Do for J = 1, 9 | | | |
| L = MCODE(J, EL) | | | |
| If L ≠ 0 | | | |
| Then | | Else | |
| Q(L) = Q(L) + FF(J) | | | |

Figure 7.19: Subroutine EXTRN2

| | |
|------------------------|------------------|
| Initialize SKT(I) = 0. | I = 1.NEQ |
| Then | Else |
| Do for N = 1, NE | Do for N = 1, NE |
| Call CBL1 | Call CBL2 |
| Call ASSEMS | Call ASSEMS |

Figure 7.20: Subroutine STIFF

| | |
|----------------------------------|-----------------|
| Initialize SE(I,J) = 0. | I = 1,9 J = 1,9 |
| Compute BL(I) using Eq. 4.3-5 | I = 1,6 |
| Do for I = 1,6 | |
| Do for J = 1,6 | |
| SE(I,J) = SE(I,J) + BL(I)*BL(J) | |
| DUM1 = TAU(1,N)*AREA(N)/ELENG(N) | |
| Do for J = 1,6 | |
| SE(J,J) = SE(J,J) + DUM1 | |
| Do for J = 1,3 | |
| SE(J,J + 3) = SE(J,J + 3) + DUM1 | |

Figure 7.21: Subroutine CBL1

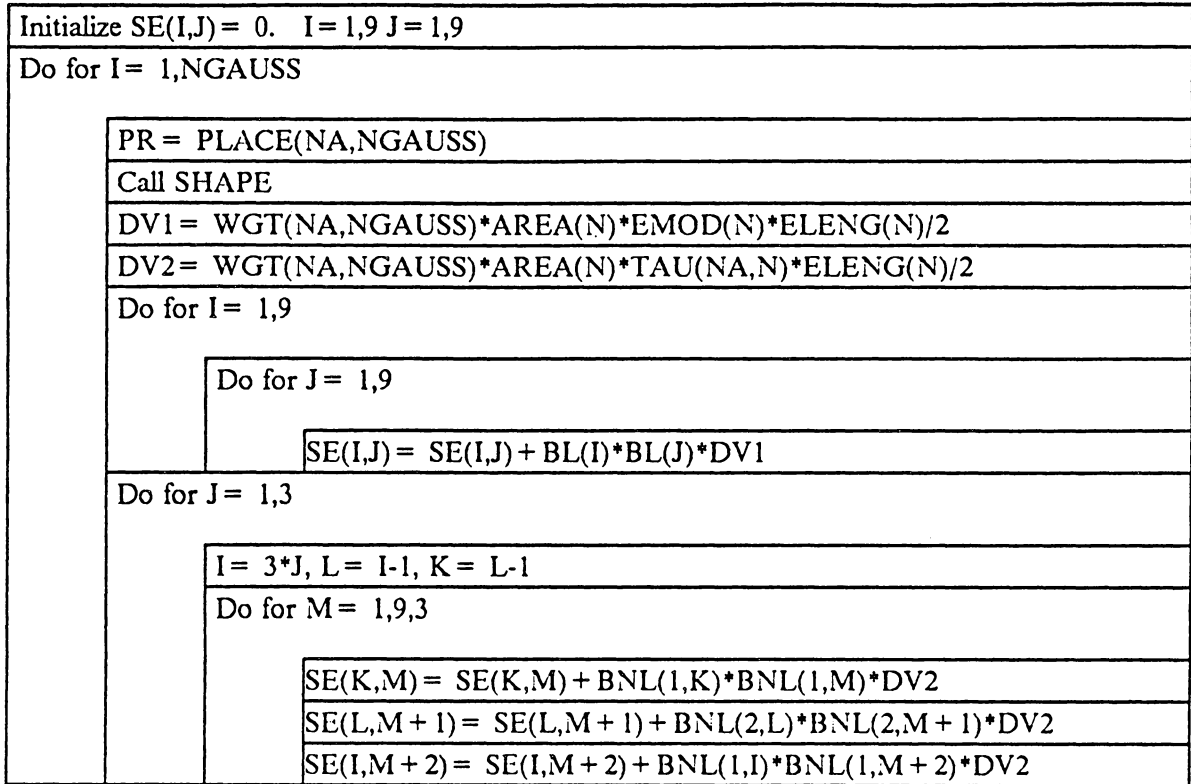


Figure 7.22: Subroutine CBL2

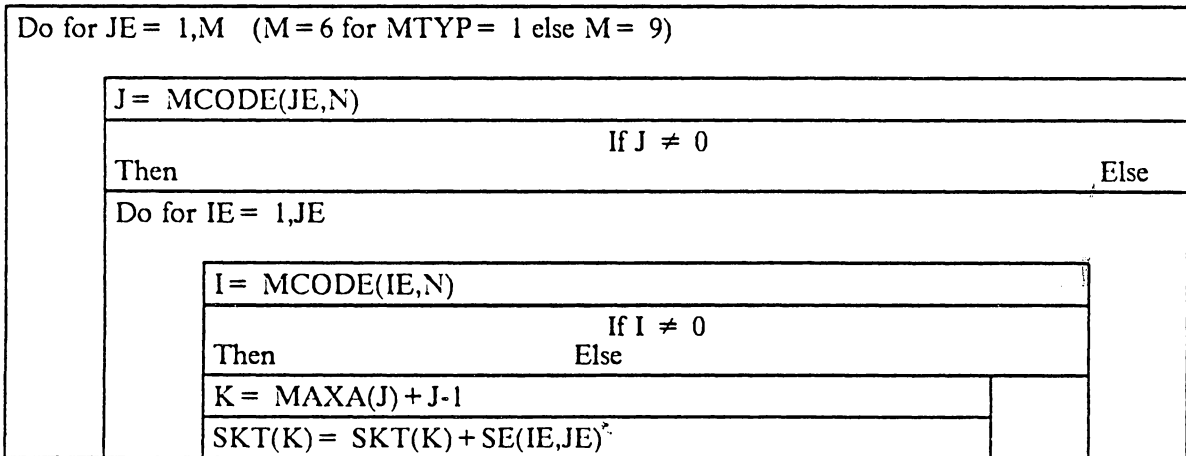


Figure 7.23: Subroutine ASSEMS

| | |
|--------------------|------|
| INCONV = 0 | |
| If TOLENE \leq 1 | |
| Then | Else |
| Call ENERGY | |
| If TOLFOR < 1 | |
| Then | Else |
| Call UNBALF | |
| If TOLDI1 < 1 | |
| Then | Else |
| Call DISPL1 | |
| If TOLDI2 < 1 | |
| Then | Else |
| Call DISPL2 | |

Figure 7.24: Subroutine TEST

| | | | | | | | |
|---|------|--|--------------------------|--|------|--|--|
| WORKP = 0 ; WORKI = 0 | | | | | | | |
| Do for I = 1 to NE | | | | | | | |
| <table border="1" style="margin-left: 40px;"> <tr> <td> WORKI = WORKI + DD(I)*(QT(I)-FPI(I)) WORKP = WORKP + DDI(I)*(QT(I)-FP(I)) </td> </tr> </table> | | WORKI = WORKI + DD(I)*(QT(I)-FPI(I)) WORKP = WORKP + DDI(I)*(QT(I)-FP(I)) | | | | | |
| WORKI = WORKI + DD(I)*(QT(I)-FPI(I)) WORKP = WORKP + DDI(I)*(QT(I)-FP(I)) | | | | | | | |
| If DIVER > 0 | | | | | | | |
| Then | Else | | | | | | |
| <table border="1" style="margin-left: 40px;"> <tr> <td>Then</td> <td>Else</td> </tr> <tr> <td>Print Divergence message</td> <td></td> </tr> <tr> <td>STOP</td> <td></td> </tr> </table> | Then | Else | Print Divergence message | | STOP | | |
| Then | Else | | | | | | |
| Print Divergence message | | | | | | | |
| STOP | | | | | | | |
| If TOLENE > 0 | | | | | | | |
| Then | Else | | | | | | |
| <table border="1" style="margin-left: 40px;"> <tr> <td>Then</td> <td>Else</td> </tr> <tr> <td>INCONV = INCONV + 1000</td> <td></td> </tr> </table> | Then | Else | INCONV = INCONV + 1000 | | | | |
| Then | Else | | | | | | |
| INCONV = INCONV + 1000 | | | | | | | |

Figure 7.25: Subroutine ENERGY

| |
|---|
| UNBFI = 0 |
| UNBFP = 0 |
| Do for I = 1 to NEQ |
| UNBFI = UNBFI + ((QT(I)-F(I))**2 UNBFP = UNBFP + ((QT(I)-FP(I))**2 |
| If UNBFP ≠ 0 |
| Then C = (DSQRT(UNBFI))/(DSQRT(UNBFP)) |
| Else INCONV = INCONV + 100 |
| C > TOLFOR |
| Then INCONV = INCONV + 100 |
| Else |

Figure 7.26: Subroutine UNBALF

| |
|---|
| DELTAD = 0 |
| TOTALD = 0 |
| Do for I = 1 to NEQ |
| DELTAD = DELTAD + (DD(I))**2 TOTALD = TOTALD + (D(I))**2 |
| If TOTALD ≠ 0 |
| Then C = (DSQRT(DELTAD))/(DSQRT(TOTALD)) |
| Else Print error message |
| C > TOLDI1 |
| Then INCONV = INCONV + 10 |
| Else |

Figure 7.27: Subroutine DISPL1

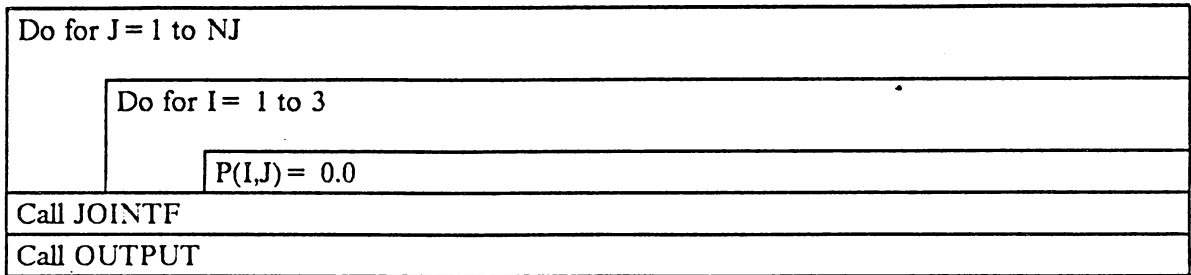


Figure 7.28: Subroutine RESULT

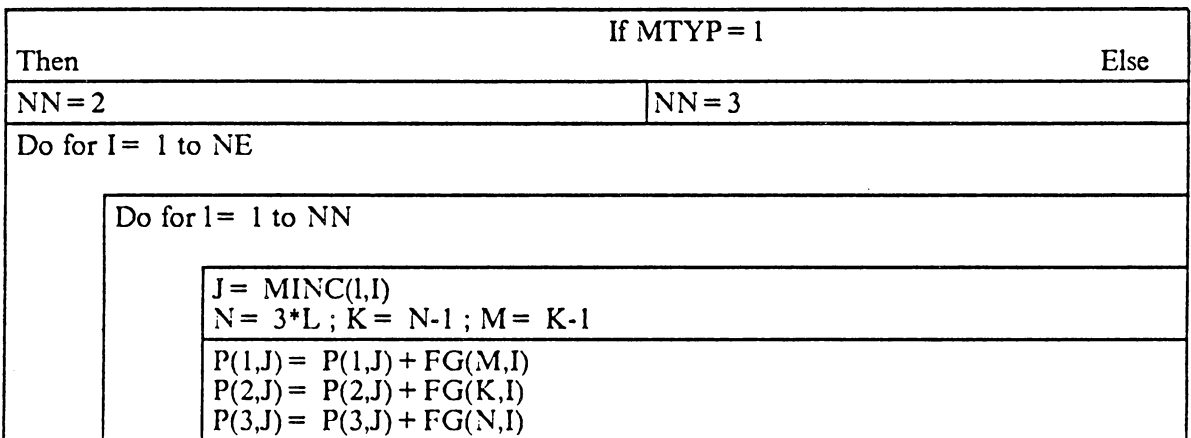


Figure 7.29: Subroutine JOINTF

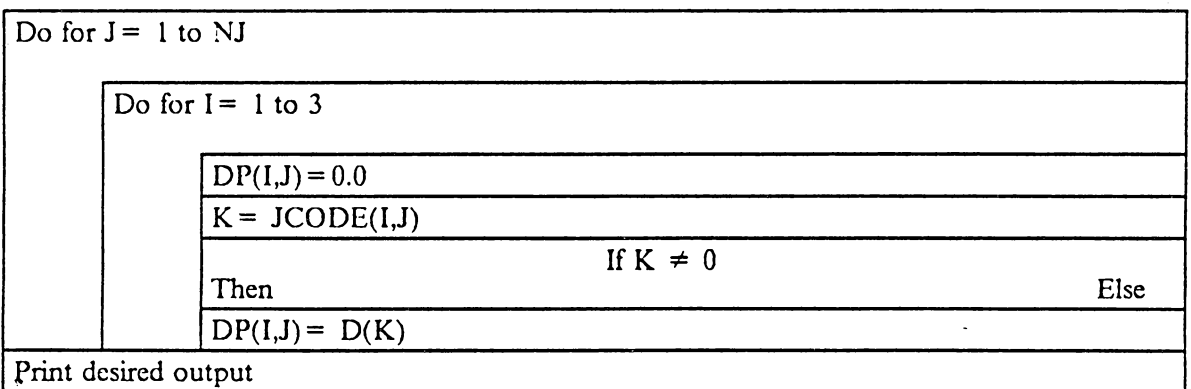


Figure 7.30: Subroutine OUTPUT

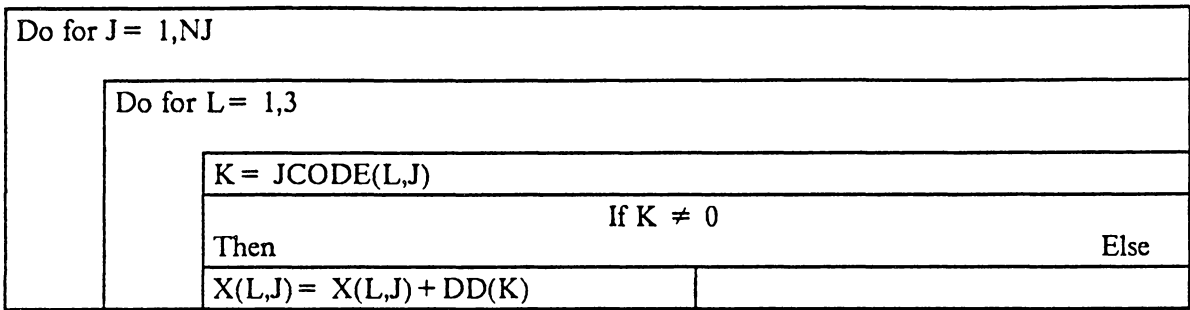


Figure 7.31: Subroutine UPDATC

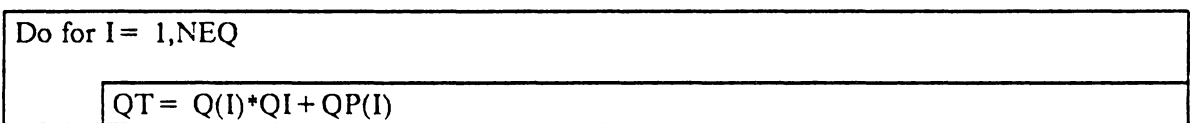


Figure 7.32: Subroutine UPDATF

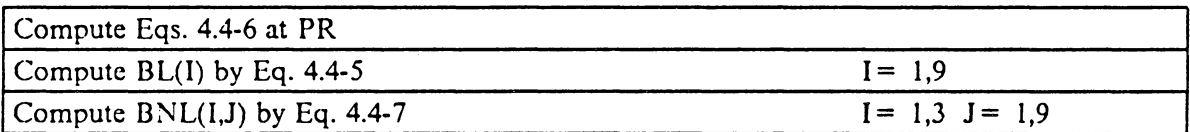


Figure 7.33: Subroutine SHAPE

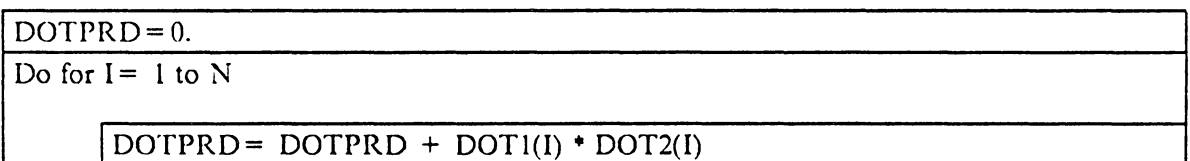


Figure 7.34: Subroutine DOTPRD

Chapter 8

TEST PROBLEMS

8.1 Introduction

The test problems used in this study can be divided into two distinct categories. Test sets 1 through 3 are used to validate the program, demonstrate the characteristics of cable structures, and comment on the use of the two node and three node elements. Test set 4 is included to compare the Modified Riks-Wempner method with Orthogonal Trajectory Accession using two models which exhibit a more complex load-displacement path.

In each of the test problems, the following convergence criteria are used (see Section 5.7):

Tolerance in displacements:

$$\text{TOLD1} = 0.001 \text{ (infinity vector norm)}$$

$$\text{TOLD2} = 0.01 \text{ (Euclidean vector norm)}$$

Tolerance in unbalanced forces:

$$\text{TOLFOR} = 0.1$$

The test for convergence in energy was not used as the above three checks are sufficient.

8.2 Test Set 1

The problem for the first test set was taken from the work of Krishna [26] and involves a freely suspended cable subjected to three loading conditions (see Figure 8.1). The three load conditions were chosen to study the behavior of the cable under the influence of the same total loading distributed in different manners.

As suggested by Krishna, the horizontal projection of the cable was divided at tenth points along the span. The nodes corresponding to these points were then determined by assuming the cable to be a parabola with a maximum sag of 10 ft. (see Chapter 6).

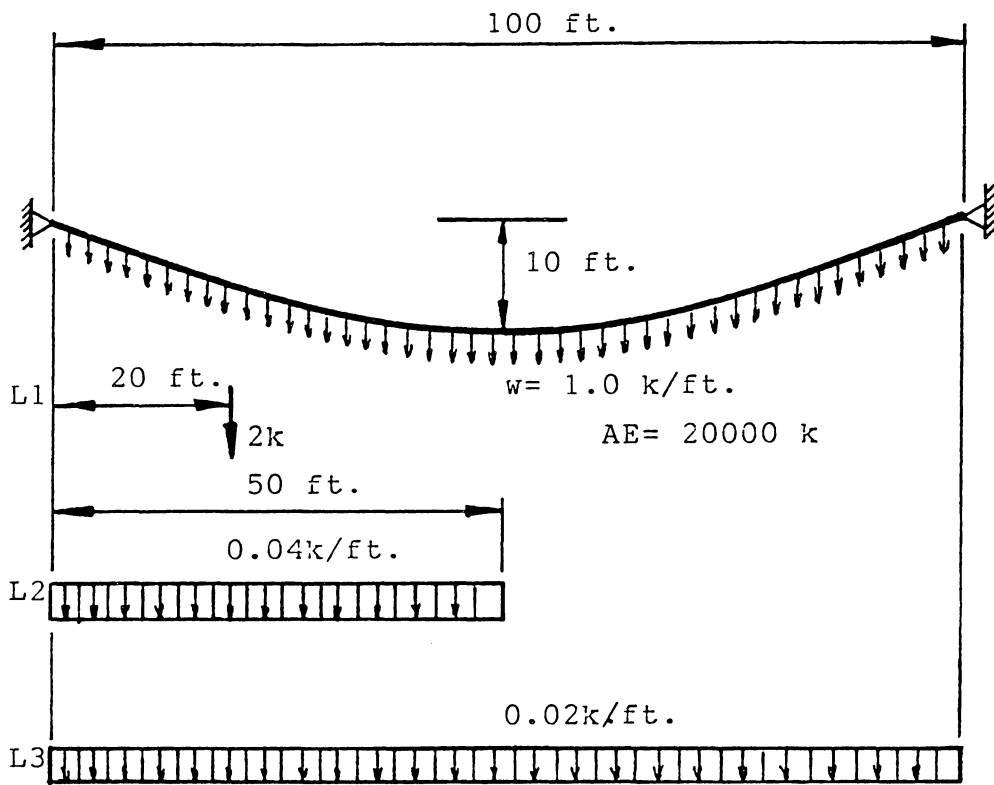


Figure 8.1: Test Set 1: A Freely Hanging Cable

The vertical displacements along the cable from the initial configuration are shown in Figure 8.2 for the three load cases. The results were obtained using Newton-Raphson with a load increment of 0.25. Although Krishna presents his results in a similar graphical form, the results of this study compare quite well with his.

As emphasized by Krishna, it is interesting to note the change of the cable profile when subjected to unsymmetric loads, particularly concentrated loads. One problem experienced in the analysis of the unstressed cable under the concentrated and partial uniform loads was the tendency for the stiffness matrix to become singular. This problem was a result of the initial motion of the elements being essentially rigid body prior to the development of axial tension. To overcome this problem, the cable was analyzed in a two part manner as described in Chapter 6. First, the cable was analyzed under its own weight. A second run was then performed using the final shape and stresses from the first run as starting values. Thus, the cable for these loads was pretensioned and the singularity problem was corrected.

Krishna suggests that most simple cable problems can be analyzed fairly accurately with no fewer than 10 linear elements [26]. To check the accuracy attained in the present study, the cable under the full uniform loading was reanalyzed using 20 straight line elements. For this particular analysis, in order to show more clearly the load-displacement behavior of the cable, EA of the cable was reduced to 20^k with an initial pretension of 5 kips and only the vertical displacement of the center of the cable was considered. Figure 8.3 shows the resulting path for both the 10 element mesh and 20 element mesh. As shown, the accuracy gained with the use of 20 elements is negligible.

As a quick comparison between the Newton-Raphson method and the Riks-Wempner method, this same load-displacement path was traced using Riks-Wempner. For this test, EA = 20^k with the initial pretension force of 5^k . The results in Figure 8.4 show only a slight discrepancy.

To determine the behavior of the three node element, the cable was reanalyzed using 5 curvilinear elements. The node points along the cable were chosen to correspond to those used for the linear element to determine if the same accuracy could be achieved using half the number of elements of the previous analysis. The behavior of the element, however, was quite erratic for all but the full uniform load condition. The solution process was much more susceptible to singularity

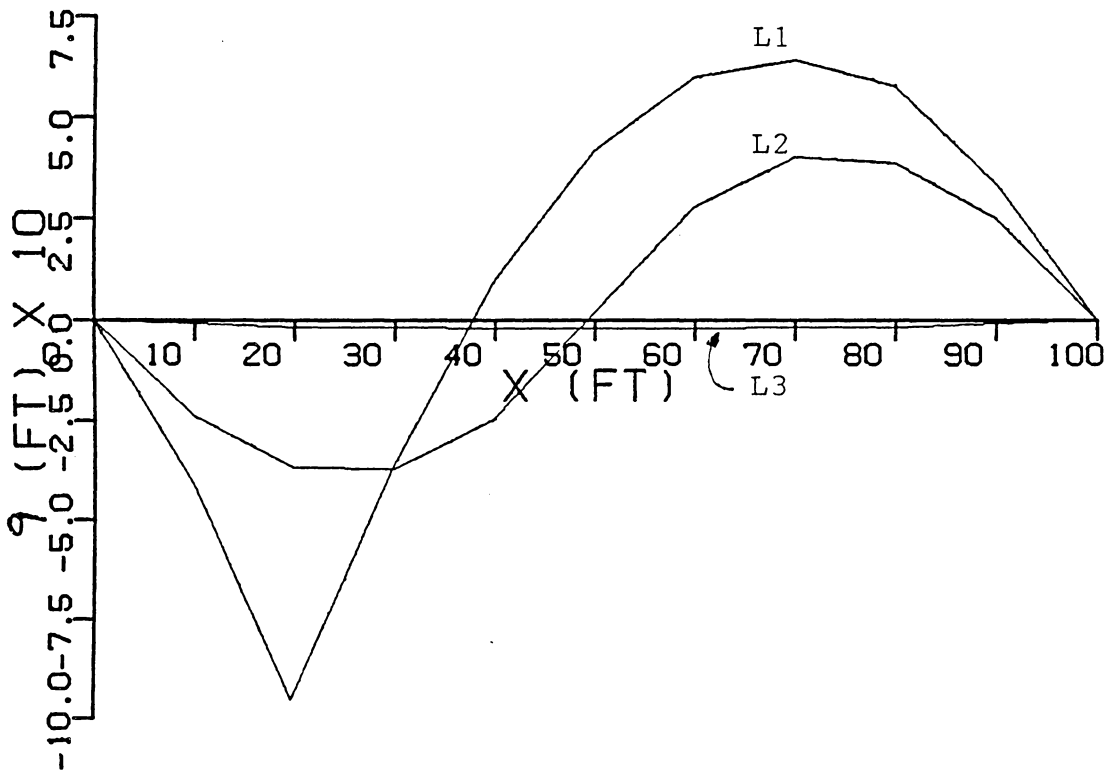


Figure 8.2: Test Set 1: Resulting Displacements of Cable

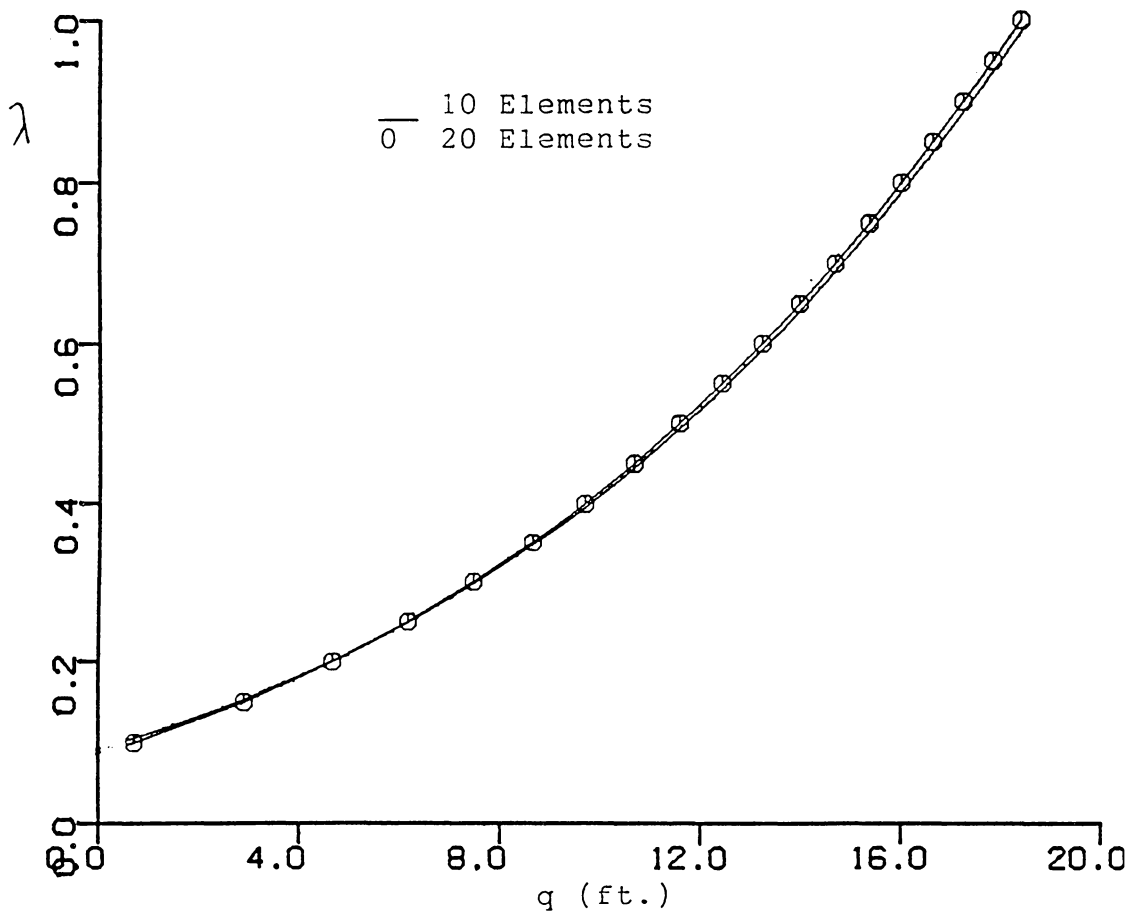


Figure 8.3: Test Set 1: Comparison of Mesh Refinement

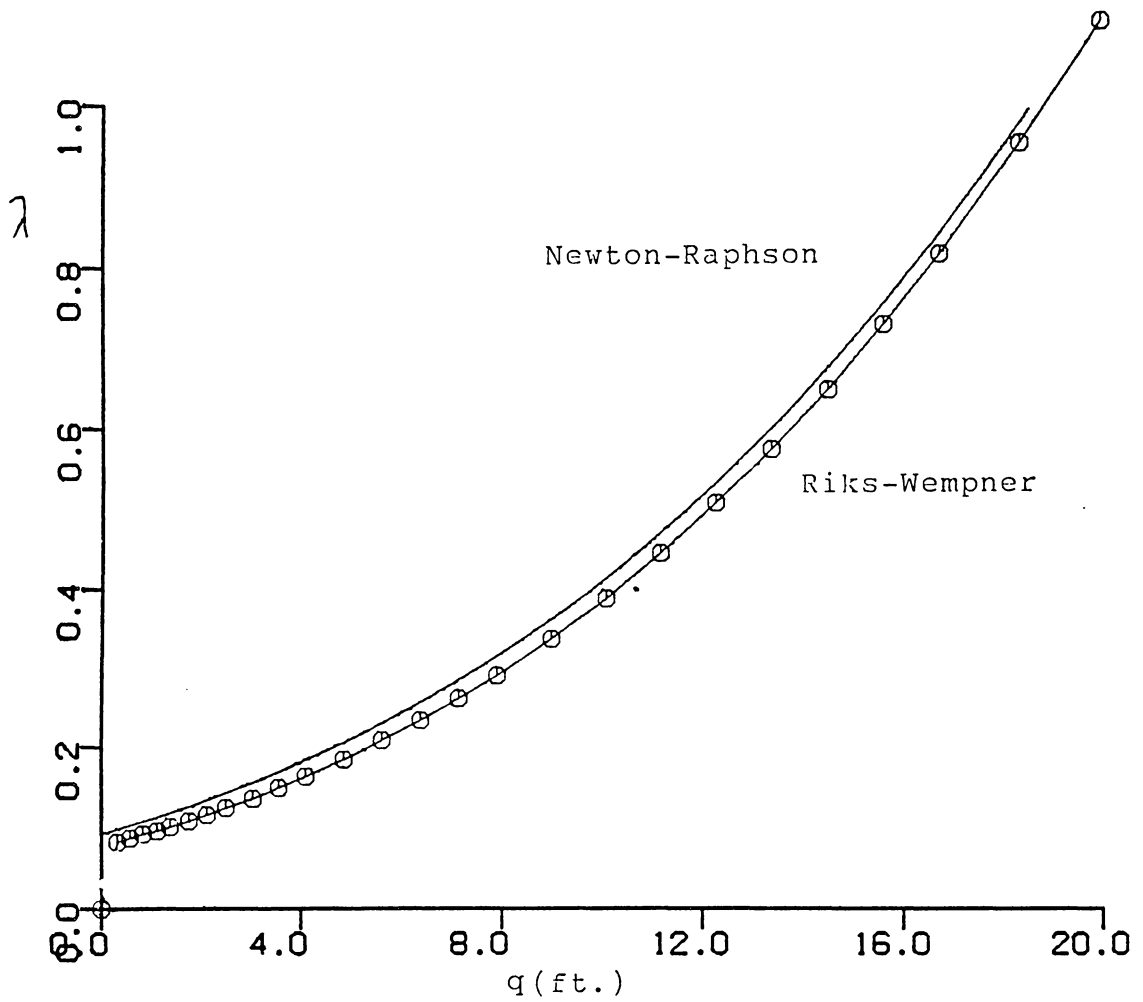


Figure 8.4: Test Set 1: Newton-Raphson Vs. Riks-Wempner

problems, even with pretensioning. The resulting center node load-displacement path under the uniformly distributed load is shown in Figure 8.5 along with that found using linear elements. The two plots compare well with the three node element being slightly stiffer than expected. In both cases $EA = 200^k$ with no initial pretension force. When 10 curvilinear elements were used, the problem of singularity was even greater and thus no sound results could be obtained to verify whether the accuracy gained using more elements would be significant.

One unique behavior of the element was noticed during trial solutions for the concentrated load condition. When the cable was pretensioned, it had a tendency to flatten out rather than maintaining a curved shape. This of course is a direct response to the initial unbalanced internal forces at the nodes; however, this may be a problem when studying pretensioned cables with a large initial force.

8.3 Test Set 2

The second test problem is also adapted from Krishna [26] and consists of the hyperbolic pretensioned cable net shown in Figure 8.6. This problem was chosen to study the three dimensional characteristics of a cable net. In addition, the configuration is representative of a common shape used in many roofing applications of cable networks. The loadings used in the analysis consist of:

1. Dead load only = 9 k/node
2. Dead load (9 k) + live load (13.5 k) = 22.5 k/node
3. Dead load + live load on section ABD
4. Dead load + wind pressure (9 k/node) on ABD + wind suction (9 k/node) on BCD.

The horizontal component of pretensioning in the cable is 180^k and EA of the hogging and sagging cables is 44000^k and 66000^k respectively. The Newton-Raphson method was used for the iterative process.

For this problem, only the two node element was used in the analysis since the net is pretensioned and thus the cable assumes a straight line between node points. This is the same assumption

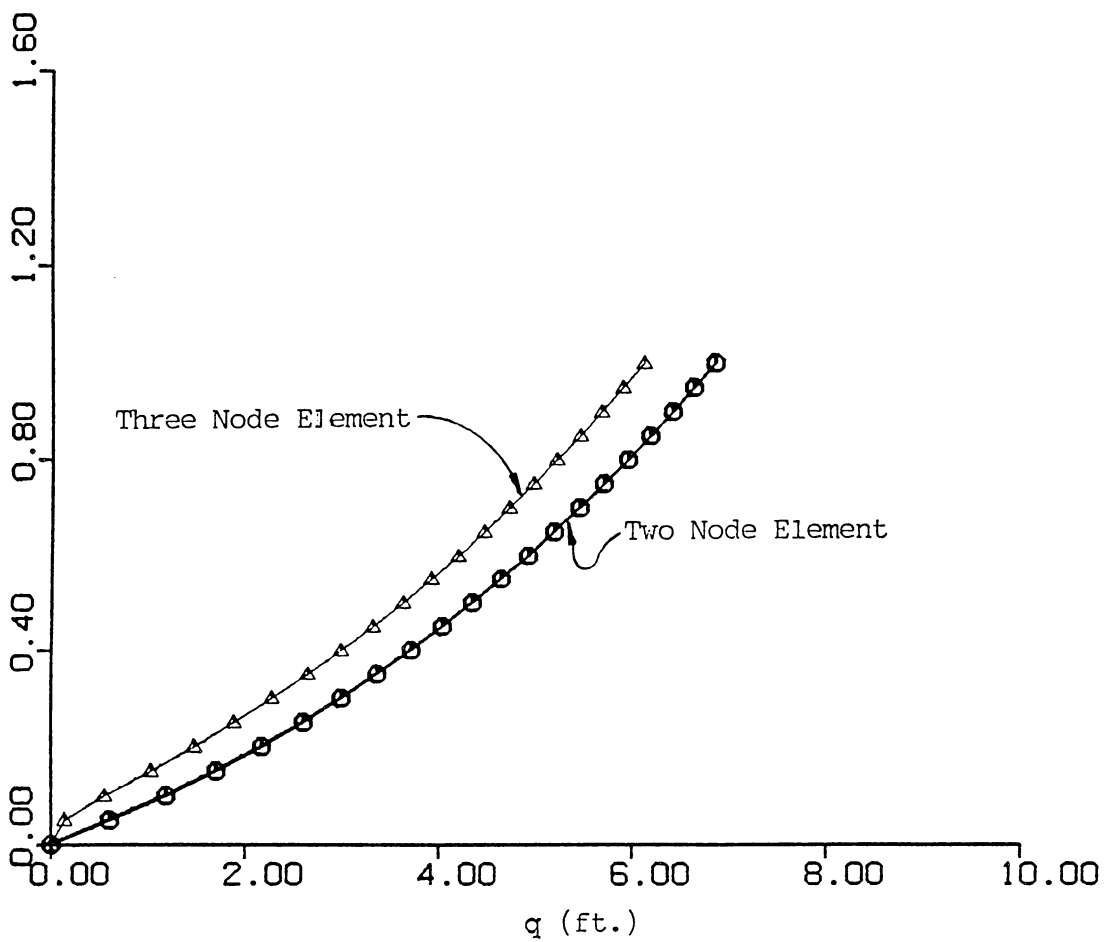


Figure 8.5: Test Set 1: Two Node Element Vs. Three Node Element

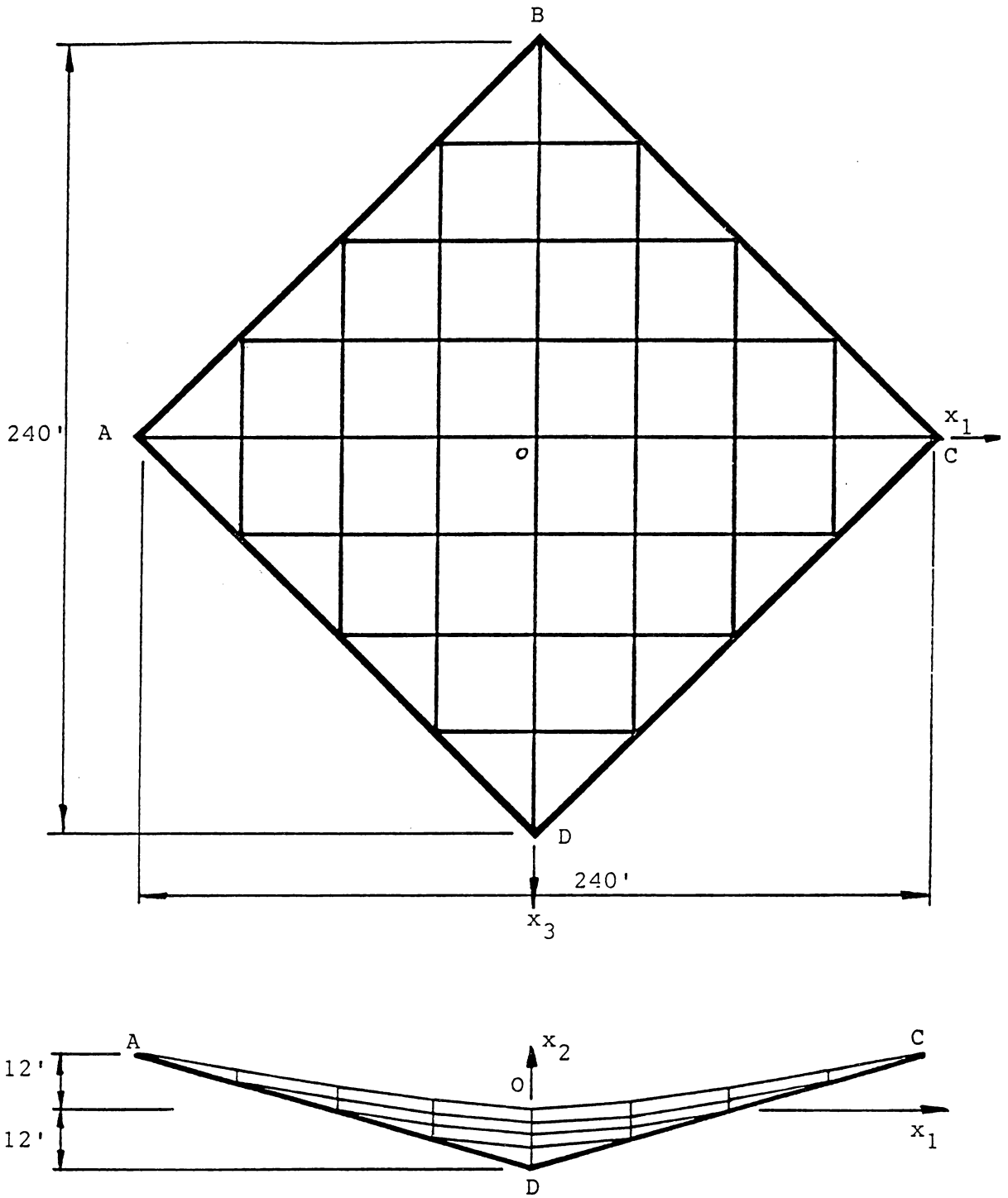


Figure 8.6: Test Set 2: Hyperbolic Cable Net

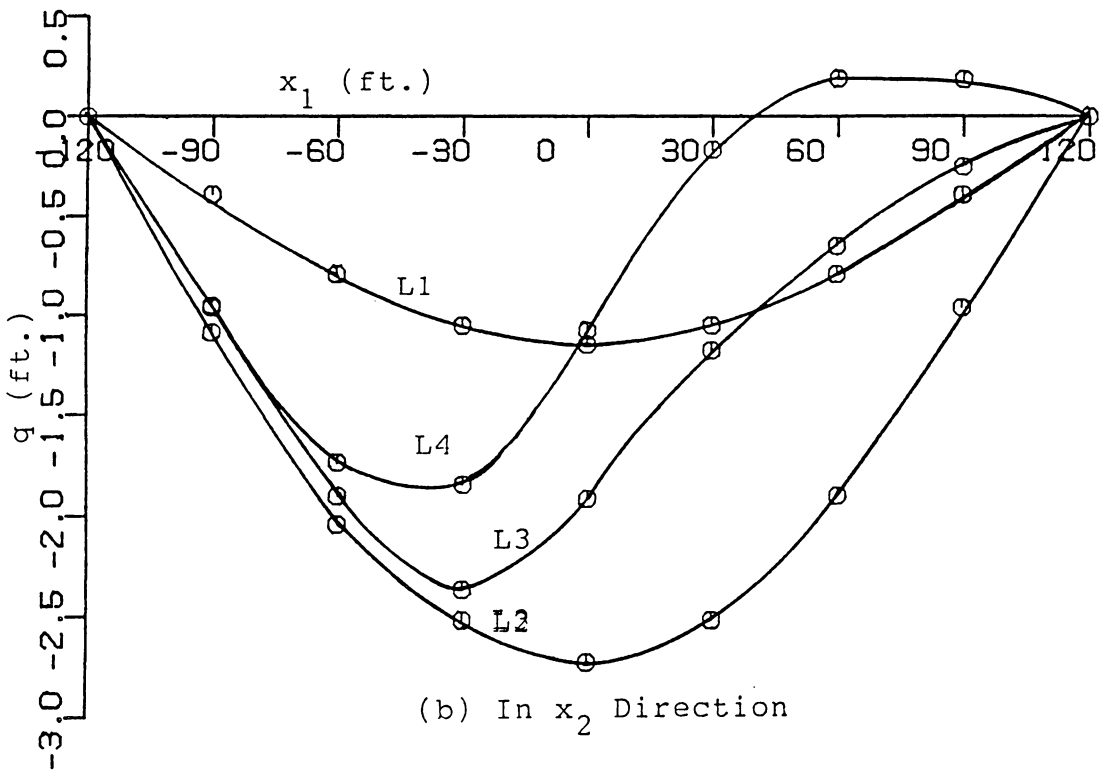
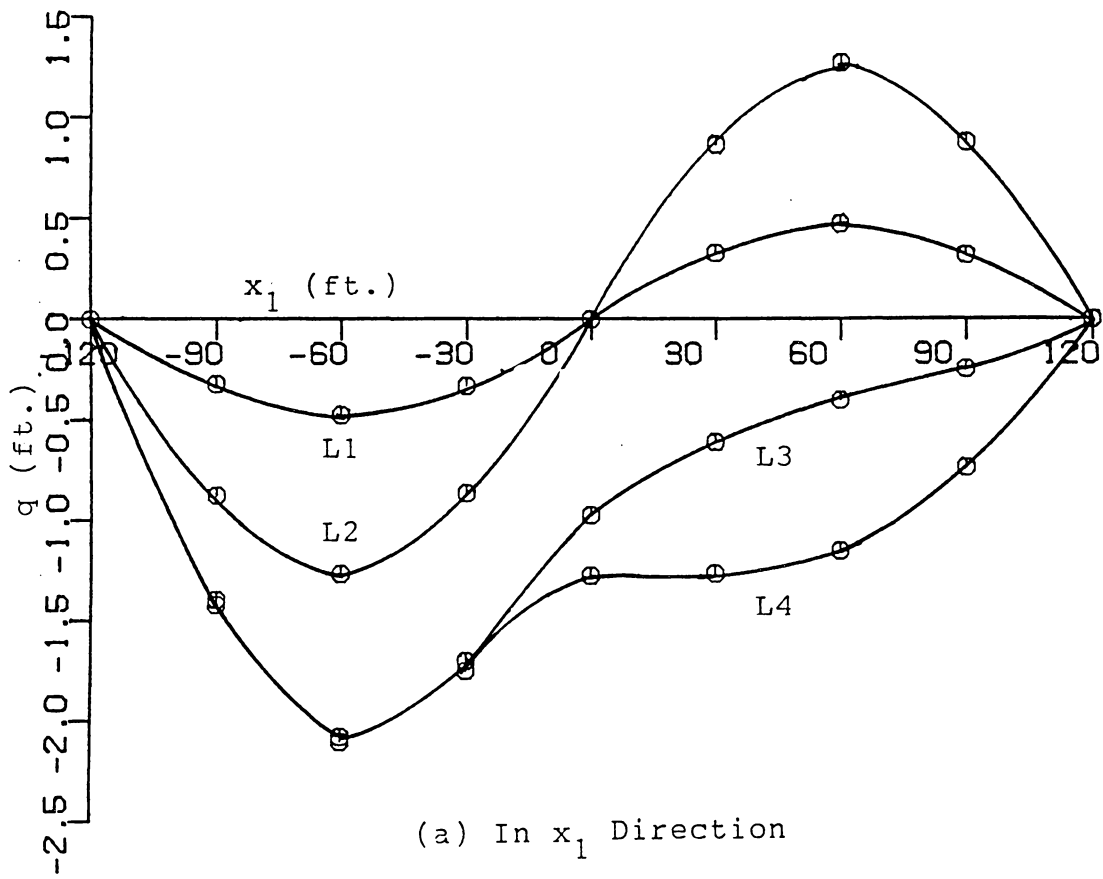
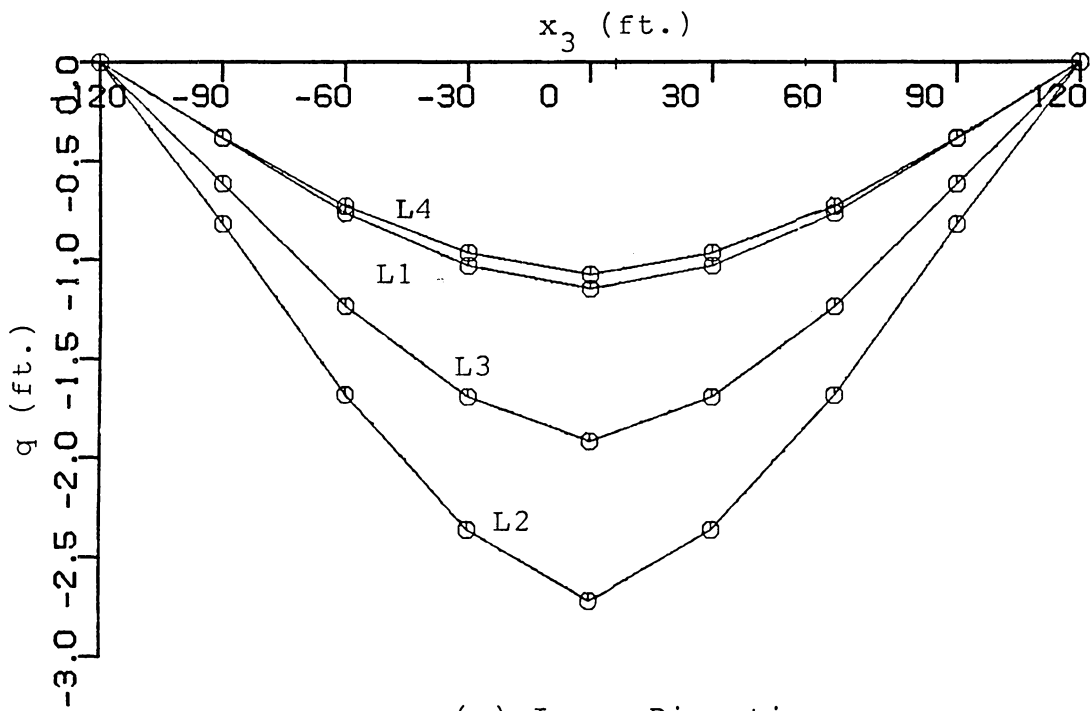
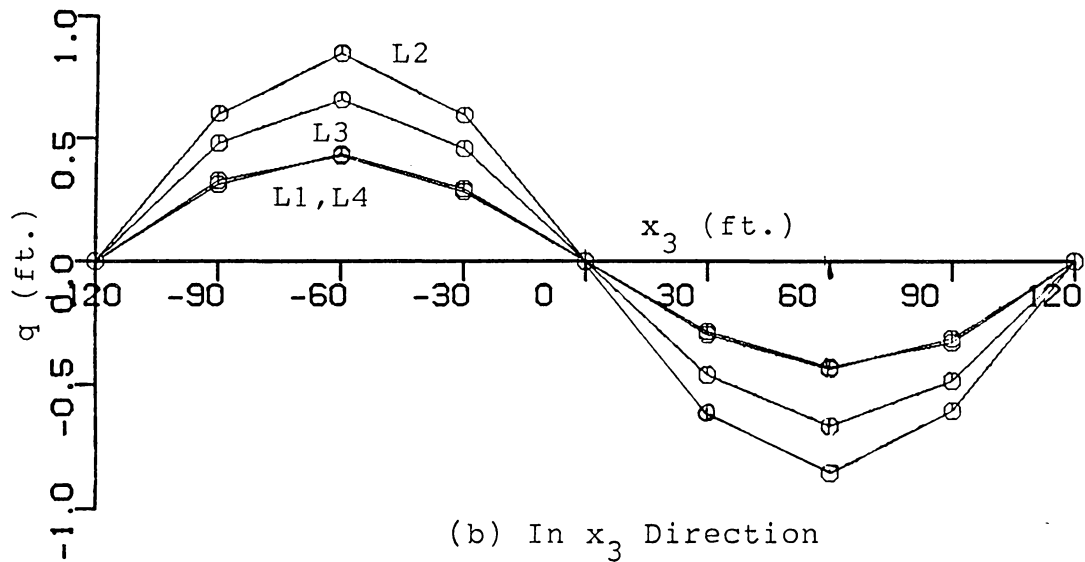


Figure 8.7: Test Set 2: Displacements Along $x=0$



(a) In x_2 Direction



(b) In x_3 Direction

Figure 8.8: Test Set 2: Displacements Along $z=0$

used by Krishna and results in negligible error, particularly when the network is covered with rigid cladding [26].

As in test set 1, the resulting displacements of the cable compare quite well with those obtained by Krishna. The plots in Figures 8.7a and b show the displacements of the net from the initial configuration in the x_1 and x_2 direction for each of the load cases along the x_1 axis. Figures 8.8a and b show the displacements in the x_2 and x_3 directions along the x_2 axis. One comment made by Krishna, which is reflected in the results, is the tendency for cables to act as a continuous membrane in such systems.

8.4 Test Set 3

Test set 3 is an investigation of Dynamic Relaxation. The first problem of test set 3 is an application of the Dynamic Relaxation approach described in Section 5.6 to the analysis of cable structures. The problem consists of reanalyzing the structure of test set 1 under the effect of the full uniform loading. The original intent of the test was to determine if DR could be used successfully in finding an equilibrium configuration for the unstressed cable under the concentrated loading. The technique, however, did not behave well during the analysis of the uniform loading and thus this load case was not investigated.

Using the value of $\Delta t = 1.0$ as suggested by Underwood [43], the solution process for the uniform loading quickly diverged, resulting in failure of the solution. After an extensive trial and error approach, it was determined that a final solution could not be attained unless the value of Δt was within the range of 0.20 to 0.35. Above this range the solution began to oscillate and diverge. Below 0.2 the solution process moved much too slowly for convergence. In addition, the resulting displacements within this range were roughly twice what they should be.

One reason for this problem was thought to be a result of the mass matrix obtained using the approach given in Section 5.6. Since the mass matrix is directly formulated from the stiffness matrix of the system, the inertial forces were considerably higher than the actual loads applied to each node. The motion of the structure was thus dominated by the mass and not the loading. To test this hypothesis, the value of EA was reduced to 20^6 and the problem was reanalyzed using both

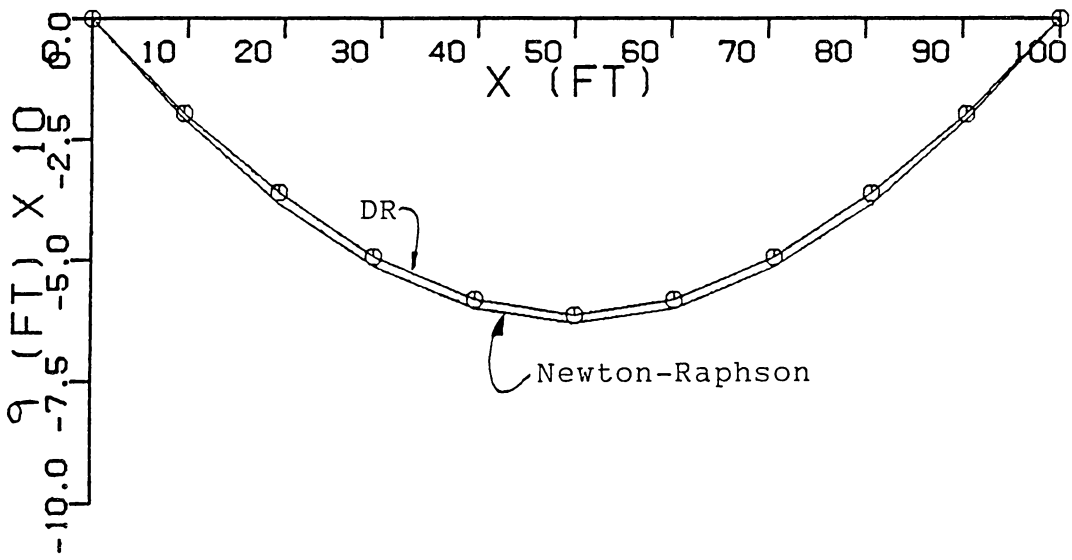


Figure 8.9: Test Set 3: Displacements of Freely Hanging Cable Using DR.

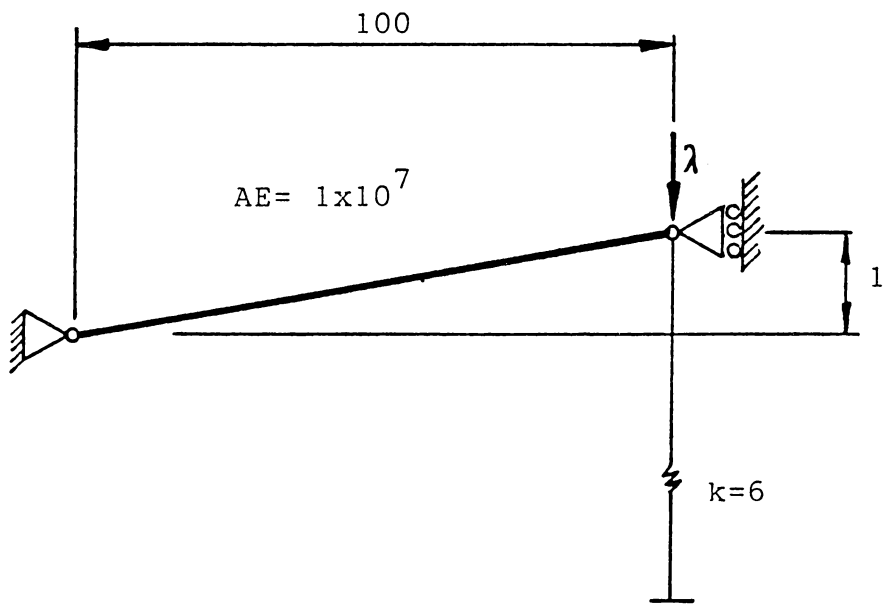


Figure 8.10: Test Set 3: Truss-Spring Model

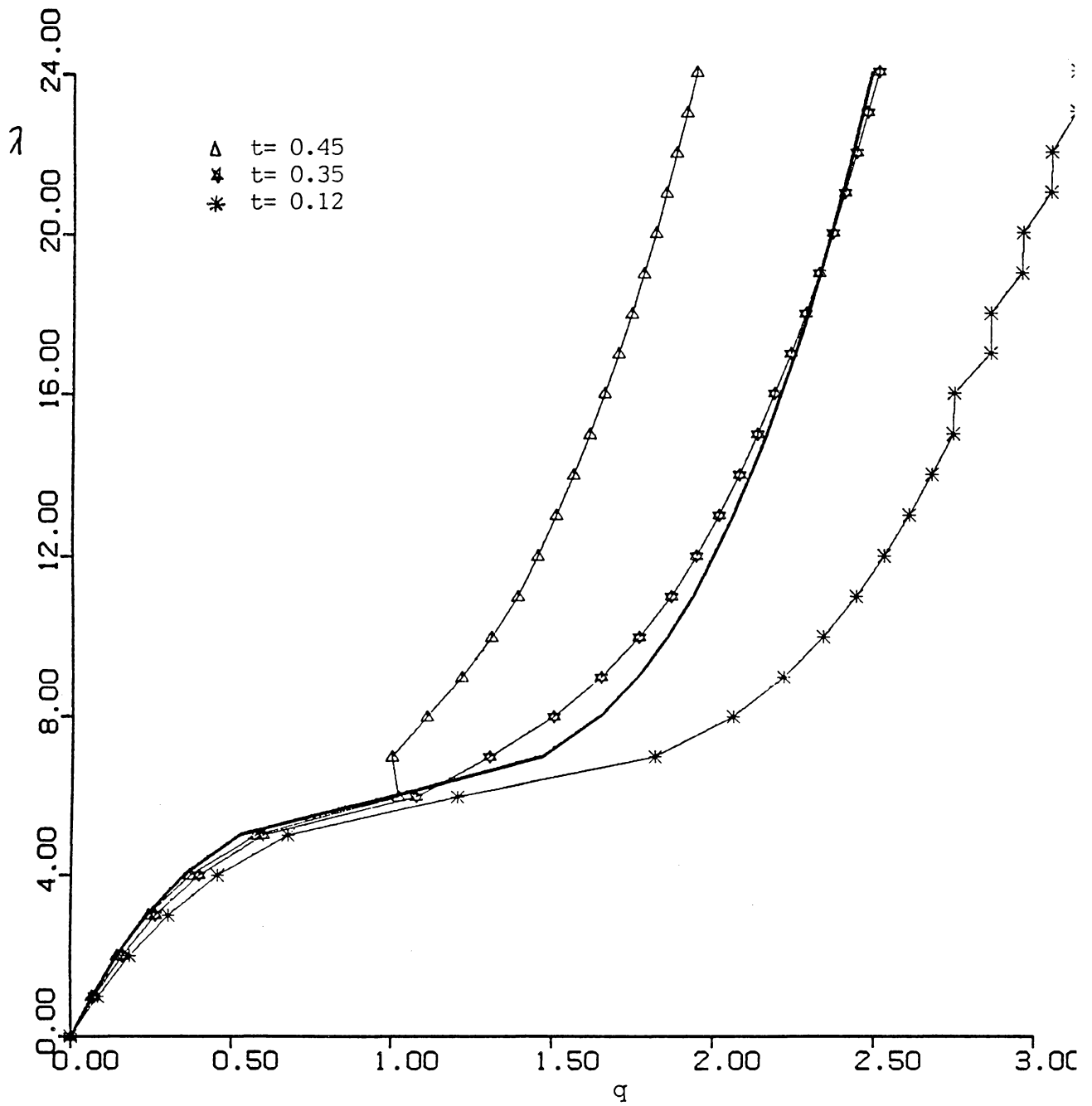


Figure 8.11a: Test Set 3: Load-Displacement of Truss-Spring

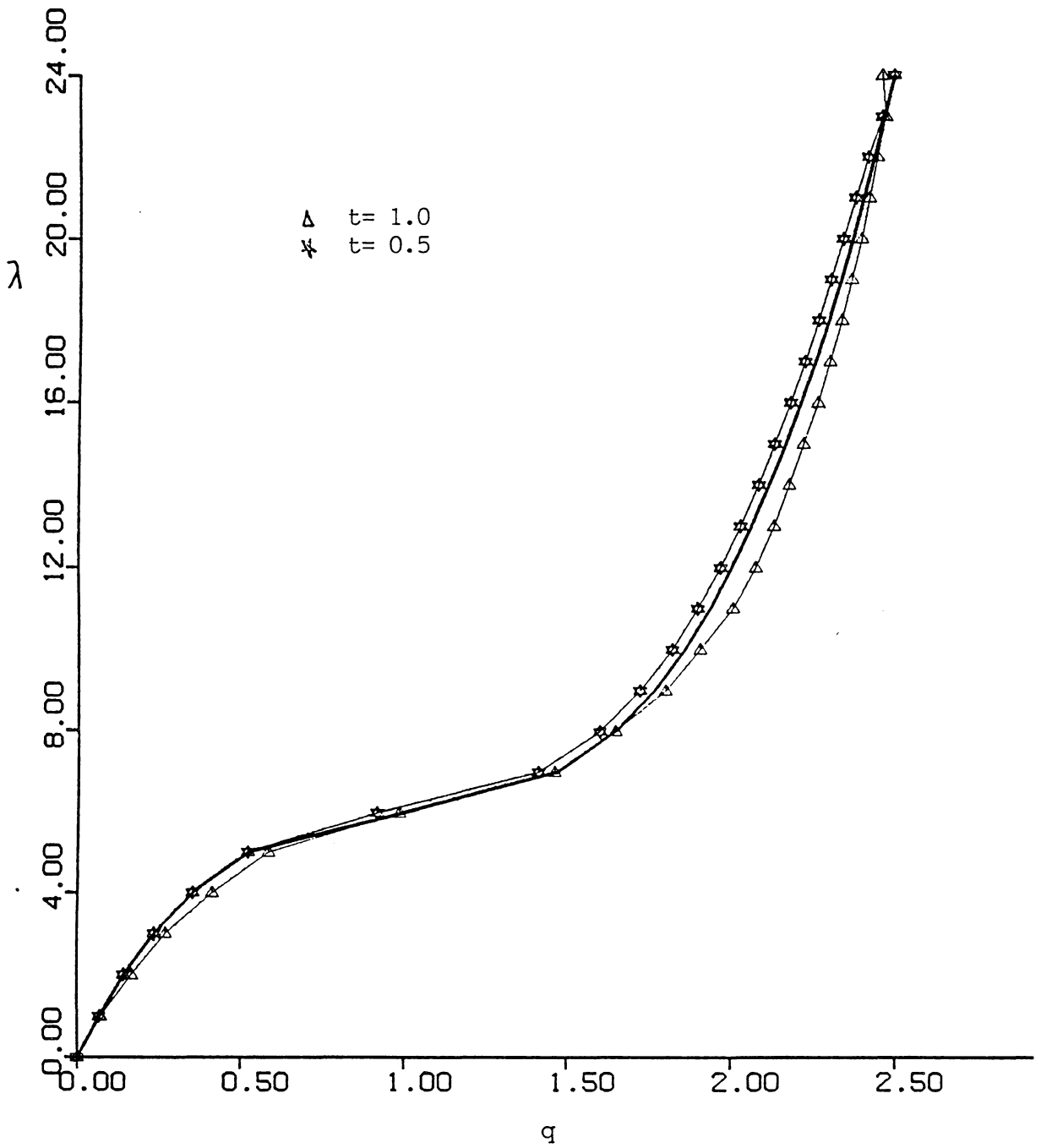


Figure 8.11b: Test Set 3: Load-Displacement of Truss-Spring

Newton-Raphson and DR. The vertical displacements along the cable are shown in Figure 8.9. As shown, the values obtained compare well with those using Newton-Raphson; however, the value of the time step used to obtain the results was 0.23.

To test the validity of DR further, the one degree of freedom problem used by Underwood [43] was analyzed. The model is the simple truss and spring configuration shown in Figure 8.10. The results are included here to illustrate the problems experienced with DR.

The initial approach to this problem was to use the same parameters chosen by Underwood for his solution. These consisted of setting the value of the lumped mass to a constant value of 30 lbs. and $\Delta t = 1.0$. The resulting load-displacement curves for $\Delta t = 1.0$ and $\Delta t = 0.50$ are shown in Figure 8.11a where the dark line is the path determined by Underwood. As shown, the results using $\Delta t = 1.0$ are reasonably close with a better correlation obtained with $\Delta t = 0.50$.

A second approach for the analysis was to form the lumped mass using the method given in Section 5.6. This results in a value of roughly one half of the value above. With $\Delta t = 1.0$, the method was not able to obtain a solution, with the resulting displacements diverging considerably. As in the analysis of the cable system, the value of Δt was reduced until a solution was obtained. The range of Δt which converged to a solution was from 0.12 to 0.5. Figure 8.11b shows the results determined using $\Delta t = 0.45$, 0.35, and 0.12. The results of $\Delta t = 0.45$ and 0.12 show the approximate limits beyond which the solution fails.

8.5 Test Set 4

The problems presented thus far have shown virtually no differences in the results obtained using either Newton-Raphson, Riks-Wempner, or Orthogonal Trajectory Accession. The primary reason for this is the simple load-displacement path exhibited by these structures. As an extension of the development of OTA presented in Chapter 5, it was decided that a brief comparison between Riks-Wempner and OTA should be performed using a more complex load-displacement path in order to determine a few characteristics of OTA. The intent here is to create a foundation for future work in applications of OTA to other systems.

To investigate the differences between the two methods, two structures exhibiting snap-through characteristics were analyzed. The first structure consists of a 21 degree of freedom lamella dome subjected to a vertical concentrated load at the crown (see Figure 8.12). This model was previously analyzed by Holzer and Vu [25,45] in their work with the Modified Riks-Wempner method and offers a good model on which to test the tracing abilities of OTA. The particular curve to be traced for this study corresponds to the projection of the equilibrium path onto the $\bar{\lambda} - q_1$ plane, where q_1 is the vertical displacement of the crown and $\bar{\lambda}$ is the nondimensionalized loading parameter (see Section 5.4) expressed as $\bar{\lambda} = 10^4\lambda/EA$.

To begin the investigations, a benchmark test was run using the Riks-Wempner method with the initial load increment $\Delta\lambda^0 = 3.0 \times 10^{-4}$ as used by Vu [45]. For the first run the simple method of scaling the arc length during iterations described in Section 5.4 was used. It was soon discovered, however, that as the solution progressed, the arc length began to grow too large to follow the path closely. To control this problem, Δs was arbitrarily limited to a maximum of twice the original arc length. The two plots in Figure 8.13 show the difference obtained when this limit is in effect and when it is not.

Comparing the results given by Vu [45] with those obtained here, a slight discrepancy is noticed, particularly in the later portions of the plot. Two reasons which can account for this are the use of a different formulation in the strain-displacement model and the use of different tolerance levels. In Vu's work, the tolerance levels for both displacements and unbalanced forces was set at 1×10^{-8} , which are very restrictive. When the program was run with these tolerance levels, a great deal more iteration was required with little variation in the resulting path. The more likely cause for the difference is the models used. It was found during the progress of implementing the linear element that the model is very susceptible to large increments in displacements as a result of the incremental formulation. The model used by Vu is not restricted in the size of displacements or rotations.

Using the scaled results of Figure 8.12 as the benchmark, the Orthogonal Trajectory Accession approach was used to analyze the structure. For this method, the initial value of $\Delta s = 0.27$ was used, which is roughly the same as Δs^0 obtained in the Riks-Wempner solution. Here again, in

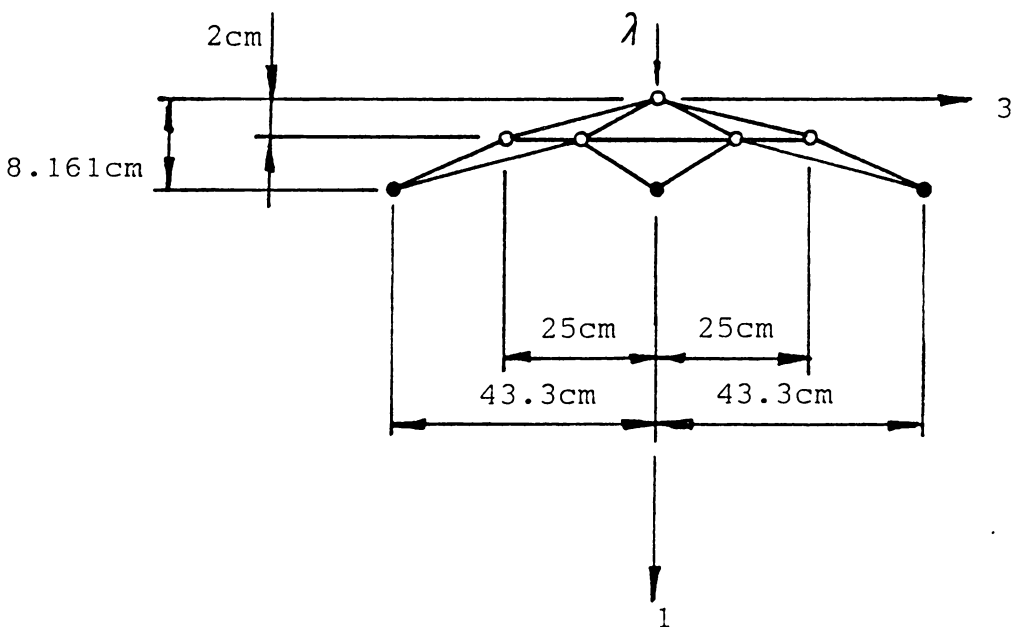
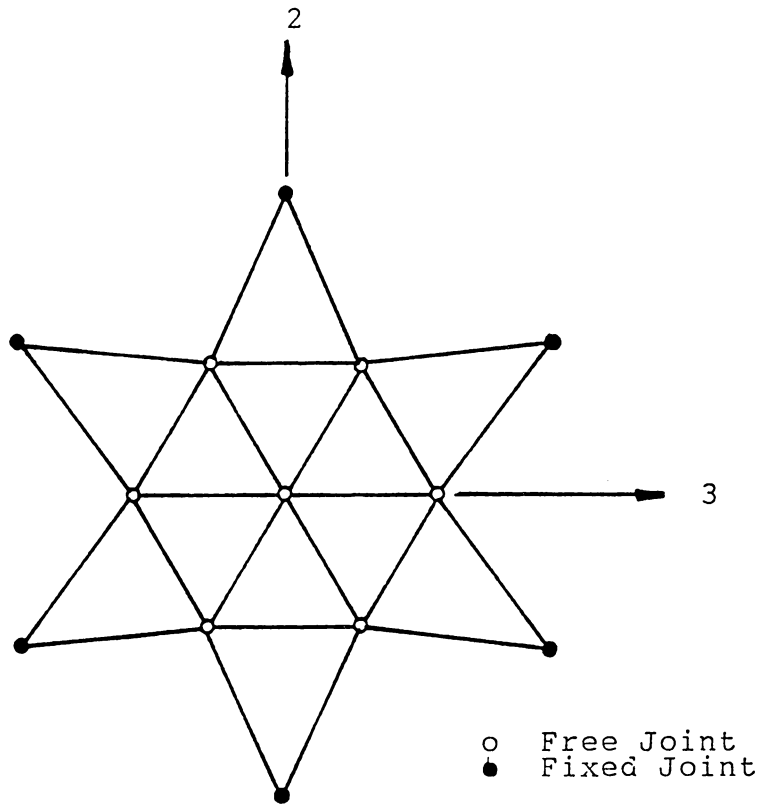


Figure 8.12: Test Set 4: 21 DOF Lamella Dome

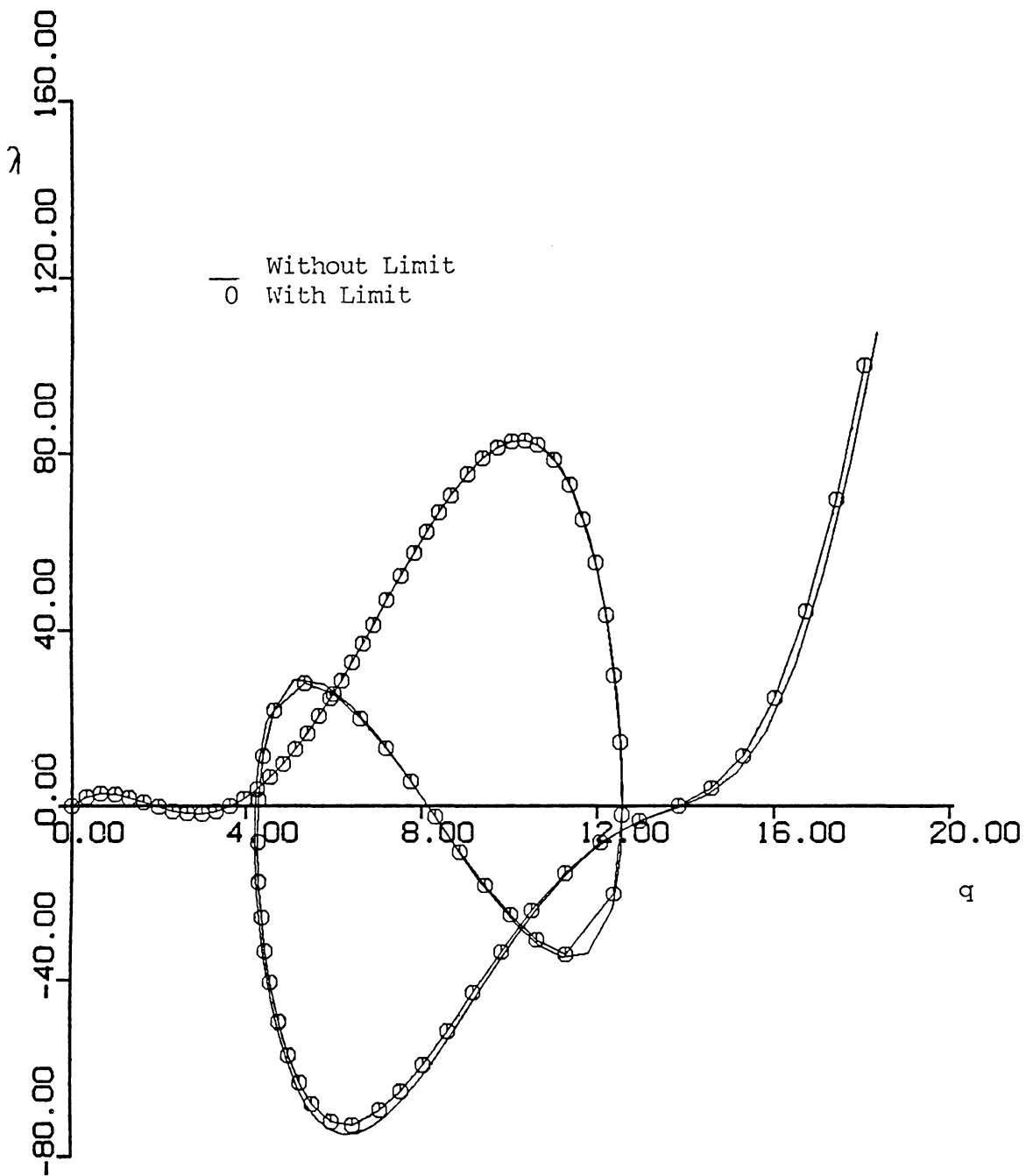


Figure 8.13: Test Set 4: Riks-Wempner, With and Without Limit

order to keep the value of the arc length from becoming too large during iteration, the maximum value was limited to twice the original arc length. Figure 8.14 shows the resulting path traced by OTA as compared with that determined from the Riks-Wempner method. It is interesting to note the difference in the two solutions as the paths progress. The two solutions are virtually identical until just after the maximum positive limit point is reached. During the descending portion of the path, the arc length of the Riks-Wempner method begins to grow at a much more rapid rate than OTA (see Figure 8.13). Thus, a greater amount of error was introduced in the remaining portions of the plot. Although this problem can be overcome by simply specifying a smaller limit on Δs for the Riks-Wempner method, it is interesting to note the much smoother path obtained using OTA with essentially the same restrictions applied. In addition, the plot traced by OTA corresponds much better to that given by Vu [45].

The second model used for this investigation is the simple truss configuration shown in Figure 8.15. This particular problem was taken from the theory manual of ABAQUS [23] and was chosen to demonstrate the starting characteristics of OTA. As mentioned in Section 5.5, the formulation of OTA allows for the use of a starting configuration which need not be an equilibrium state of the system. To verify this ability, an initial configuration corresponding to $\lambda = 0.5$ and $q = 0$ was chosen. As shown in Figure 8.16, this value of loading exceeds the first limit point of the equilibrium path. The solution process, however, successfully found and traced the path with only a marginal error resulting from the large initial increment in displacement. The complete path shown in the figure was generated using OTA starting from a zero stress state with virtually no difference from that given in ABAQUS [23].

A conclusion that OTA is a substantially better solution technique than the Modified Riks-Wempner method cannot be made after these two brief test problems. One conclusion can be drawn which may help others more fully investigate the method. The most important point to consider with both OTA and Riks-Wempner is an efficient and effective means of controlling the arc length during iteration. As demonstrated by the two models used here, proper control of the arc length is very important in order to reduce the propagation of errors during the solution. Although the scaling method used in this study works well for the structures under consideration, a

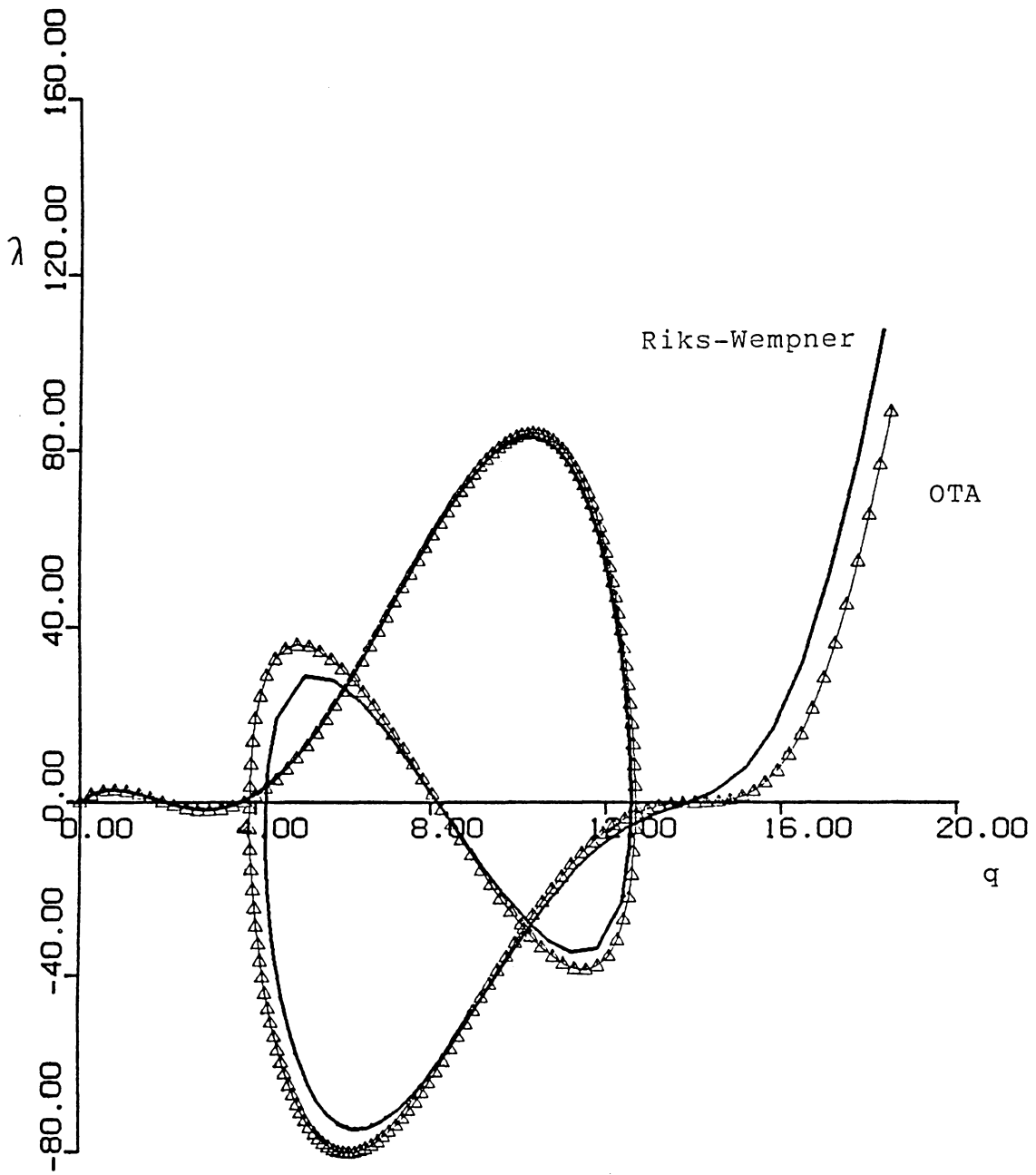


Figure 8.14: Test Set 4: Riks-Wempner Vs. OTA

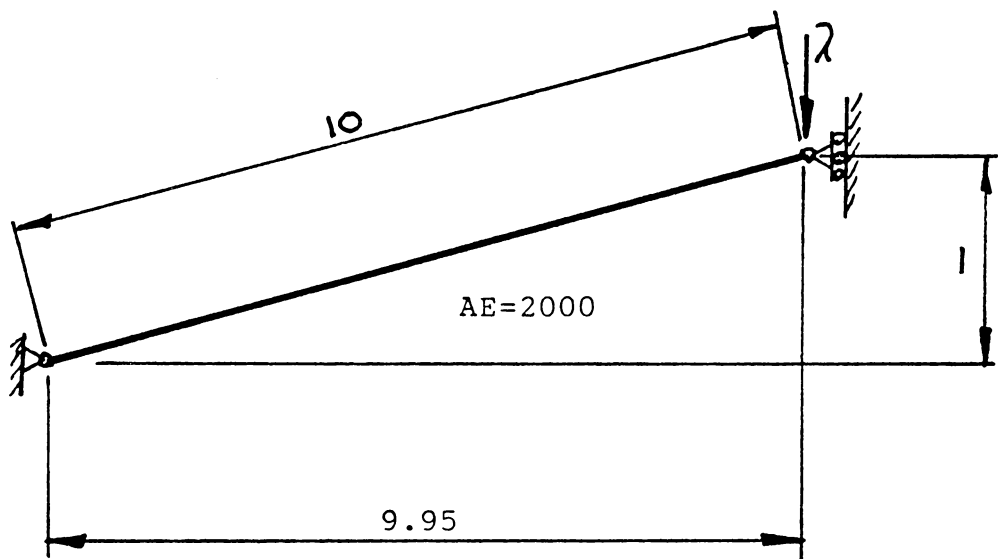


Figure 8.15 Set 4: One Degree of Freedom Truss Model

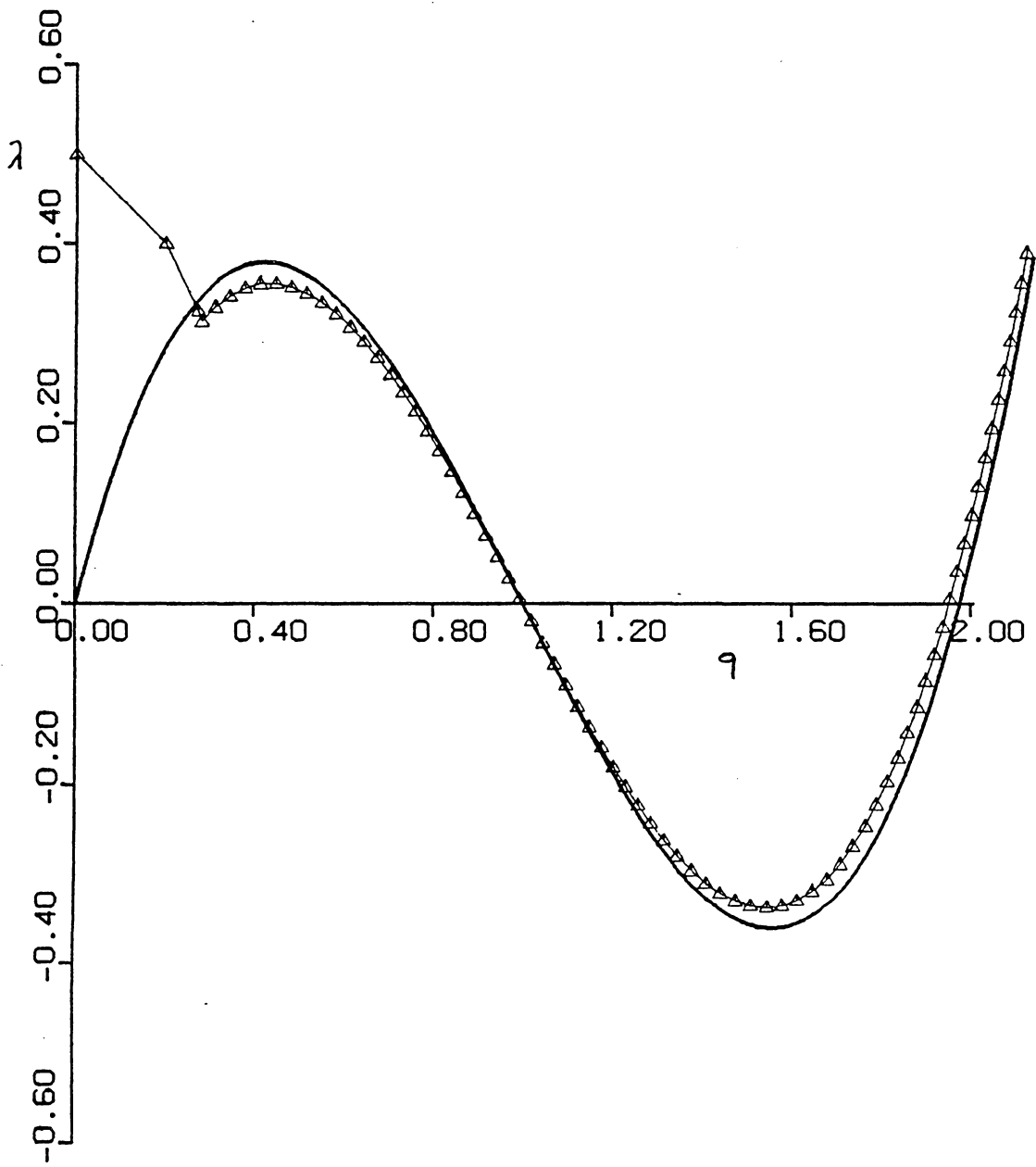


Figure 8.16: Test Set 4: Starting Ability of OTA

substantial amount of trial and error testing was performed in order to gain a reasonably accurate plot without excessive iteration.

Chapter 9

CONCLUSION

9.1 Conclusions

Using the principles of continuum mechanics, the nonlinear displacement-based finite element equations of equilibrium were developed using both the Total Lagrangian and Updated Lagrangian formulations. The Updated Lagrangian equations were then specialized for two one-dimensional elements: a linear two node element and a curvilinear three node element. The two elements were then implemented in a computer program developed for the analysis of three-dimensional cable structures under static loading.

In addition to the implementation of the two elements, four solution techniques were investigated in the solution of the equilibrium equations: the Newton-Raphson method, The Modified Riks-Wempner method, a modification of the Riks-Wempner method referred to by Fried [15] as Orthogonal Trajectory Accession, and a Dynamic Relaxation approach suggested by Underwood [43].

To study the characteristics of cable structures and the application of the solution techniques in the analysis of cables, three test sets were performed. In the first test, a freely hanging cable was analyzed using both the two node and three node elements. In this problem the two node element was found to perform quite well as compared to the published results. The three node element, however, exhibited erratic behavior for all but the simplest of the load cases. The primary reason for this is thought to be the sensitivity of the element to singularity problems. In addition, the three node element did not behave well when subjected to an initial pretensioning. Although many authors warn of the sensitivity of the curvilinear element to incorrect strain-displacement modeling, the exact problem in the formulation used here was never determined.

The second test set was concerned with the behavior of a hyperbolic pretensioned cable network. The intent was to show the characteristics of cables in three dimensional networks. The results of this analysis, obtained with the two node element, compare quite favorably with those published by Krishna [26].

The third test set was an application of a Dynamic Relaxation approach to the analysis of nonlinear problems. DR is an adaptation of the principles of structural dynamics to the solution of the equilibrium equations. In this study, one particular approach proposed by Underwood [43] was chosen for implementation in the program. This method, however, was extremely sensitive in the analysis of cable structures and thus no clear conclusions could be drawn as to the effectiveness of this approach. It was originally thought that this method would be able to solve those problems where singularity caused the other solution techniques to fail. Considerable effort was spent with little results to prove this, however.

The fourth test set was included to provide some comparisons between the Modified Riks-Wempner method and Orthogonal Trajectory Accession. As demonstrated on two separate models, OTA is very similar to the Riks-Wempner method in application. Both were able to follow the equilibrium paths to completion; however, OTA was able to follow the true path more closely under the same restrictions as applied to the Riks-Wempner method. In addition, a demonstration of the starting abilities of OTA from a poor trial configuration was included.

9.2 Recommendations For Further Studies

1. As an extension of the work started here, the extension of the program to include beam and membrane elements would provide for a much more versatile tool in conducting a thorough investigation of tension structures in general. A large portion of the program was borrowed directly from the work of Navarro [33] and thus implementation of this beam element should prove rather simple.
2. The problems experienced with the three node element require a thorough investigation of the behavior of this particular element. Many of the authors referenced in this work express cau-

tion when considering implementation of this element. It is particularly important to model the strain-displacement relations correctly.

3. Although the problems experienced with the implementation of the Dynamic Relaxation approach in this study may suggest that DR is not a reliable method in the analysis of cables, the approach used here was only one adaptation among numerous other variations. It is felt that a more thorough investigation of DR would prove the technique to have some attractive features in the solution of highly nonlinear problems.
4. As an extension of the above suggestion, many authors express the need for more research into the dynamic characteristics of cable structures, particularly cable networks. A more extensive investigation of the dynamics of cables should be performed prior to further work with DR in order to predict the behavior of the method more accurately.
5. A very broad adaptation of this work involves the extension of the program capabilities to include the analysis of pneumatic structures. This would involve the development of a membrane element for implementation in the program and the consideration of follower forces in the formulation of the equilibrium equations.
6. A test problem was intentionally included as a basis for future work with OTA. Although only two brief tests were performed, the results should indicate the need for further work in this area. Particular attention needs to be given to an effective way to control the arc length during iteration.
7. The program was written to follow the structure used by Holzer [24] as closely as possible since the modular format makes the program easy to modify. An interesting extension of the program would be an adaptation to the PC. This would involve the development of pre and post processing programs in addition to considering the limitations of memory and speed. As dis-

cussed by Haber [19,20], the use of an interactive environment is particularly valuable when analyzing cable structures.

Appendix A.

REFERENCES

1. Argyris, J.H., Angelopoulos, T., and Bichat, B., "A General Method for the Shape Finding of Lightweight Tension Structures", Computer Methods in Applied Mechanics and Engineering , Vol. 3, 1974, pp. 135-149.
2. Bathe, K.J., Finite Element Procedures In Engineering Analysis , Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1982.
3. Bathe, K.J. and Cimiento, A.P., "Some Practical Procedures for the Solution of Nonlinear Finite Element Equations", Computer Methods in Applied Mechanics and Engineering , Vol. 27, 1980, pp. 59-85.
4. Belytschko, T., "Transient Analysis", in Structural Mechanics Computer Programs , edited by Pilkey, W., Saczaaski, K., and Schaeffer, H. , University Press of Virginia, 1974.
5. Boresi, A.P., Sidebottom, O.M., Seely, F.B., and Smith, J.O., Advanced Mechanics of Materials , 3rd. ed., John Wiley and Sons, New York, 1978.
6. Brew, J.S. and Brotton, D.M., "Non-linear Structural Analysis By Dynamic Relaxation", International Journal for Numerical Methods in Engineering , Vol. 3, 1971, pp. 463-483.
7. Buchholdt, H.A., Introduction to Cable Roof Structures , Cambridge University Press, New York, 1985.
8. Bunce, J. W., "A Note on the Estimation of Critical Damping In Dynamic Relaxation", International Journal for Numerical Methods in Engineering , Vol. 4, 1972, pp. 301-304.
9. Butler, M.J., "A Comparison of Two Models for Geometrically Nonlinear Finite Element Analysis of Plane Frames", M.S. Thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, December, 1983.
10. Cook, R.D., Concepts and Applications of Finite Element Analysis , 2nd ed., John Wiley and Sons, New York, 1981.
11. Day, A.S. and Bunce, J.H., "Analysis of Cable Networks by Dynamic Relaxation", Civil Engineering and Public Works Review , Vol. 4, 1970, pp. 383-386.
12. Felippa, C.A., "Dynamic Relaxation Under General Incremental Control", in Proceedings of the International Conference on Innovative Methods for Nonlinear Problems , edited by Lin, W., Belytschko, T., and Park, K., Pineridge Press Int., 1984.
13. Felippa, C.A., "Finite Element Analysis of Three-Dimensional Cable Structures", in Computational Methods in Nonlinear Mechanics , edited by Oden, J.T., 1974.
14. Frederick, D, and Chang, T.S., Continuum Mechanics , Scientific Publishers, Inc., Boston, Massachusetts, 1972.
15. Fried, I., "Orthogonal Trajectory Accession to the Nonlinear Equilibrium Curve", Computer Methods in Applied Mechanics and Engineering , Vol. 47, 1984, pp. 283-297.

16. Fried, I., "Large Deformation Static and Dynamic Finite Element Analysis of Extensible Cables", Computers and Structures , Vol. 15, 1982, pp. 315-319.
17. Fung, Y.C., Foundations of Solid Mechanics , Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1965.
18. Gadala, M.S., Dokainish, M.A., and Oravas, G., "Geometric and Material Nonlinearity Problems", International Journal for Numerical Methods in Engineering , Vol. 20, 1984, pg. 887.
19. Haber, R.B. and Abel, J.F., "Initial Equilibrium Solution Methods for Cable Reinforced Membranes, Part I - Formulations", Computer Methods in Applied Mechanics and Engineering , Vol. 30, 1982, pp. 263-284.
20. Haber, R.B. and Abel, J.F., "Initial Equilibrium Solution Methods for Cable Reinforced Membranes, Part II - Implementation", Computer Methods in Applied Mechanics and Engineering , Vol. 30, 1982, pp. 285-306.
21. Haug, E. and Powell, G.H., "Analytical Shape Finding for Cable Nets", IASS Pacific Symposium - Part II on Tension Structures and Space Frames , Session 1, October 17-23, 1971, Tokyo and Kyoto.
22. Henghold, W.M. and Russel, J.J., "Equilibrium and Natural Frequencies of Cable Structures (A Nonlinear Finite Element Approach)", Computers and Structures , Vol. 6, 1976, pp 267-271.
23. Hibbitt, Karlsson and Sorenson, Inc., ABAQUS Ver. 4.5: Theory Manual, September, 1984.
24. Holzer, S.M., Computer Analysis of Structures, Matrix Structural Analysis, Structured Programming , Elsevier, 1985.
25. Holzer, S.M., Watson, L.T., and Vu, P., "Stability Analysis of Lamella Domes", Proceedings of the ASCE Symposium on Long Span Roof Structures , St. Louis, Missouri, October, 1981, pp. 179-209.
26. Krishna, P., Cable-Supported Roofs , McGraw-Hill, New York, 1978.
27. Knudson, W.C. and Scordelis, A.C., "Cable Forces for Desired Shapes in Cable-Net Structures", IASS Pacific Symposium - Part II on Tension Structures and Space Frames , Session 1, October 17-23, 1971, Tokyo and Kyoto.
28. Lewis, W.J. and Jones, M.S., "Dynamic Relaxation of the Non-linear Static Response of Pretensioned Cable Roofs", Computers and Structures , Vol. 18, 1984, pp. 989-997.
29. Li, C.T. and Srivastava, N.K., "Analysis of Pneumatic Shells With or Without Cables; General Finite Element Formulation", Computers and Structures , Vol. 4, 1974, pp.813-828.
30. Malvern, L.E., Introduction to the Mechanics of a Continuous Medium , Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1969.
31. Mondkar, D.P. and Powell, G.H., "Static and Dynamic Analysis of Nonlinear Structures", Report No. EERC75-10 , Earthquake Engineering Research Center, University of California, Berkeley, 1975.
32. Mondkar, D.P., and Powell, G.H., "Finite Element Analysis of Non-Linear Static Response", International Journal for Numerical Methods in Engineering , Vol. 11, 1977, pp. 499-520.
33. Navarro, J.F., "Continuum Mechanics Formulation for Incremental Solution of One-Dimensional Finite Element Meshes", M.S. Thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, December, 1984.

34. Oden, J.T., Finite Elements of Nonlinear Continua , McGraw-Hill, Inc., New York, 1972.
35. Otto, F., Tensile Structures , MIT Press, Cambridge, Massachusetts, 1973
36. Ozdemir, H., "A Finite Element Approach for Cable Problems", International Journal of Solids and Structures , Vol. 15, 1979, pp.427-437.
37. Papadrakakis, M., "A Method for the Automatic Evaluation of the Dynamic Relaxation Parameters", Computer Methods in Applied Mechanics and Engineering , Vol. 25, 1981, pp. 35-48.
38. Park, K.C. and Underwood, P.G., "A Variable-Step Central Difference Method For the Dynamics Analysis - Part 1. Theoretical Aspects", Computer Methods in Applied Mechanics and Engineering , Vol 22, 1980, pp. 241-258.
39. Peyrot, A.H. and Goulois, A.M., "Analysis of Cable Structures", Computers and Structures , Vol. 10, 1979, pp. 805-813.
40. Szabo, J. and Kollar, L., Structural Design of Cable-Suspended Roofs , John Wiley and Sons, New York, 1984.
41. Schrefler, B.A., Odorzi, S., and Wood, R.D., "A Total Lagrangian Geometrically Non-Linear Analysis of Combined Beam and Cable Structures", Computers and Structures , Vol. 17, 1983, pp. 115-127.
42. Tuchman, J.L., "Olympic Domes First of Their Kind", Engineering News-Record , March, 6, 1986, pp. 24-27.
43. Underwood, P., "Dynamic Relaxation", in Computational Methods for Transient Analysis , edited by Belytschko, T. and Hughes, T., Elsevier Science, 1983, pp. 245-265.
44. Verma, V.K. and Leonard, J.W., "Nonlinear Behavior of Cable-Reinforced Membranes", Journal of the Engineering Mechanics Division , ASCE,
45. Vu, P.D., "Tracing Nonlinear Equilibrium Paths by the Modified Riks-Wempner Method", M.S. Thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, October, 1981.
46. Webster, R.L., "On the Static Analysis of Structures With Strong Geometric Nonlinearity", Computers and Structures , Vol. 11, 1980, pp. 137-145.
47. Weiner, R.I. and Magladry, R., "Air Supported Roof Systems, Occupant Safety in Large Fabric Roof Structures", Professional Safety , November, 1980, pp. 30-34.
48. Zienkiewicz, O.C., The Finite Element Method , 3rd. ed., McGraw-Hill, New York, 1977

Appendix B.
PROGRAM LISTING

-----*-----

 * NONLINEAR F.E. ANALYSIS OF CABLES *

 -----*-----

WRITTEN BY WILLIAM M. SAGE
 1986

THIS PROGRAM IS WRITTEN TO PERFORM GEOMETRICALLY NONLINEAR FINITE ELEMENT ANALYSIS OF CABLE STRUCTURES USING THE UPDATED LAGRANGIAN APPROACH. FOUR SEPERATE SOLUTION TECHNIQUES ARE INCLUDED TO SOLVE THE EQUILIBRIUM EQUATIONS: THE NEWTON RAPHSON METHOD, THE MODIFIED RIKS-WEMPNER METHOD, AN ORTHOGONAL TRAJECTORY ACCESSION METHOD, AND A DYNAMIC RELAXATION METHOD.

TWO ELEMENT TYPES CAN BE USED: A TWO NODE LINEAR ELEMENT AND A THREE NODE CURVILINEAR ELEMENT. IN EACH CASE, THE ELEMENTS HAVE CONSTANT CROSS-SECTION AND MATERIAL PROPERTIES.

IN ADDITION TO SPECIFYING NODAL LOADS WHICH ARE INCREMENTALLY APPLIED, LOADS WHICH REMAIN CONSTANT MAY BE APPLIED AND AN INITIAL PRETENSION FORCE CAN BE SPECIFIED FOR RESTART PURPOSES. TWO LOAD TYPES CAN BE USED: DEAD WEIGHT PER UNIT LENGTH AND UNIFORM LOAD PER UNIT HORIZONTAL PROJECTION.

-----*-----

 * VARIABLES *

 -----*-----

- AC(NEQ) = ACCELERATION AT CURRENT TIME STEP.
- ACP(NEQ) = ACCELERATION OF PREVIOUS TIME STEP.
- ACT(NAT) = MAGNITUDE OF ELEMENT ACTION NUMBER NAT.
- AREA(NE) = CROSS-SECTIONAL AREA OF ELEMENT NE.
- BL(9) = LINEAR STRAIN DISPLACEMENT MATRIX.
- BNL(3,9) = NONLINEAR STRAIN DISPLACEMENT MATRIX.
- D(NEq) = TOTAL DISPLACEMENT CORRESPONDING TO DOF NEQ.
- DD(NEQ) = INCREMENTAL DISPLACEMENT CORRESPONDING TO DOF NEQ.
- DD1(NEQ) = DD FOR THE FIRST ITERATION.
- DE(9) = LOCAL ELEMENT INCREMENTAL DISPLACEMENTS.
- DIVER = FLAG TO TEST FOR DIVERGENCE.
- DJ(3,NJ) = TOTAL JOINT DISPLACEMENT.
- DMASS(NEQ) = LUMPED MASS MATRIX.
- DOT1, DOT2 = VECTORS BETWEEN WHICH DOT PRODUCT IS FOUND.
- DP(NEQ) = DISPLACEMENT FROM PREVIOUS TIME STEP.
- DQI = INCREMENT IN LOAD LEVEL.
- DT = TIME STEP SIZE.
- E = PERTURBED APPARENT ERROR LEVEL IN DYNAM.
- ELENG(NE) = ELEMENT LENGTH.
- EMOD(NE) = MODULOUS OF ELASTICITY OF ELEMENT NE.
- F(NEQ) = INTERNAL FORCE VECTOR FOR CURRENT STRESS LEVEL.
- FF(9) = LOCAL ELEMENT END FORCES.
- FG(9,NE) = INTERNAL ELEMENT FORCES IN GLOBAL COORDINATES.
- FP(NEQ) = INTERNAL FORCES OF THE PRESIDING EQUILIBRIUM POINT.
- FPI(NEQ) = INTERNAL FORCES OF THE PRESIDING ITERATION.
- IMPOR = DEGREE OF FREEDOM CHOSEN TO BE PLOTTED.
- INCONV = INDICATOR OF CONVERGENCE.
- ITDES = DESIRED NUMBER OF ITERATIONS BETWEEN EQ. POINTS IN RIKS-WEMPNER AND OTA.

C ITEMAX = MAXIMUM NUMBER OF ITERATIONS IN N/R, R/K, AND OTA.
C ITEUPD = NUMBER OF ITERATIONS BETWEEN UPDATING THE TANGENT
C STIFFNESS MATRIX.
C JCODE(3,NJ) = JOINT CODE MATRIX CONTAINING DOF INFORMATION.
C KFLG = LINEAR/NONLINEAR OPTION (TWO NODE ELEMENT ONLY).
C KHT(NEQ+1) = COLUMN HEIGHT OF FULL TANGENT STIFFNESS MATRIX.
C LCNT = MAX NUMBER OF ITERATIONS IN DYNAMIC RELAXATION.
C MAT(NAT) = MEMBER ACTION TYPE FOR ACTION NAT.
C 1 - DEAD WEIGHT PER UNIT LENGTH.
C 2 - UNIFORMLY DISTRIBUTED LOAD (IN HORZ.).
C MAXA(NEQ+1) = ADDRESSES OF THE MAIN DIAGONAL TERMS OF FULL
C TANGENT STIFFNESS MATRIX.
C MCODE(9,NE) = MEMBER CODE MATRIX CONTAINING DOF INFO. OF ELEMNT.
C MFLG = FLAG FOR UPDATING MASS MATRIX.
C MINC(3,NE) = MEMBER INCIDENCE MATRIX CONTAINING NUMBER OF JOINT
C AT EACH NODE FOR THE ELEMENT.
C MN(NAT) = MEMBER NUMBER SUBJECTED TO LOAD NAT.
C MTYP = FLAG FOR MEMBER TYPE USED
C 1 - TWO NODE ELEMENT
C 2 - THREE NODE ELEMENT
C MX = MAXIMUM NUMBER OF ELEMENTS AND JOINTS ALLOWED.
C MXNA = MAXIMUM NUMBER OF ACTIONS ALLOWED.
C MXNEQ = MAXIMUM NUMBER OF DOF.
C NE = NUMBER OF ELEMENTS.
C NEQ = NUMBER OF EQUATIONS (DOF).
C NGAUSS = ORDER OF GAUSSIAN QUADRATURE.
C NJ = NUMBER OF JOINTS.
C NPRINT = DESIRED OUTPUT INDICATOR.
C 1 - EQUILIBRIUM PATH PLOT DATA.
C 2 - EQUILIBRIUM PATH PLOT AND ELEMENT STRESSES.
C 3 - FULL DEBUGGING OUTPUT.
C 4 - GLOBAL JOINT DISPLACEMENTS AND FORCES.
C NREF = ITERATION COUNTER FOR NUMBER TRYS TO REFORM DMASS.
C P(3,NJ) = JOINT FORCES.
C PLACE(3,3) = LOCATION OF GAUSS POINTS.
C Q(NEQ) = APPLIED LOAD DISTRIBUTION VECTOR.
C QI = LOAD MULTIPLYING PARAMETER AT CURRENT LOAD LEVEL.
C QIMAX = MAXIMUM ALLOWABLE LOAD LEVEL.
C QJ(NEQ) = INCREMENTAL EXTERNALLY APPLIED JOINT LOADS.
C QP(NEQ) = CONSTANT EXTERNALLY APPLIED JOINT LOADS.
C QT(NEQ) = APPLIED LOAD LEVEL AT CURRENT CONFIGURATION.
C SE9,9 = LOCAL ELEMENT STIFFNESS MATRIX.
C SKT(I) = SYSTEM STIFFNESS STORED IN SKYLINE FASHION.
C SL = LOCAL ELEMENT STIFFNESS CONTRIBUTION IN DYNAM.
C TAU(3,NE) = STRESS AT GAUSS POINT FOR EACH ELEMENT
C (MAX OF 3 GAUSS POINTS IN THREE NODE ELEMENT).
C TOLDI1 = TOLERANCE IN DISPLACEMENT (EUCLIDEAN VECTOR NORM).
C TOLDI2 = TOLERANCE IN DISPLACEMENT (INFINITY VECTOR NORM).
C TOLENE = TOLERANCE IN INTERNAL ENERGY.
C TOLFOR = TOLERANCE IN UNBALANCED FORCES.

C VN = CURRENT CRITICAL DAMPING PARAMETER.
 C WGT(3,3) = WEIGHTING VALUES FOR GAUSSIAN QUADRATURE.
 C X(3,NJ) = JOINT COORDINATES.
 C XL, YL, ZL = LOCAL JOINT COORDINATES FOR A GIVEN ELEMENT.
 C Z(NEQ) = DUMMY VECTOR IN RIKS-WEMPNER NAD OTA.
 C = CURRENT VELOCITY VECTOR IN DYNAM.
 C ZP(NEQ) = PREVIOUS VELOCITY VECTOR IN DYNAM.

 * INPUT FORMAT *

NOTE: ALL READ STATEMENTS ARE FREE FORMAT.

- 1 - (IN MAIN) QI,QIMAX,DQI
- 2 - (IN MAIN) IMPOR
- 3 - (IN DATA) NE,NJ,MTYP
- 4 - (IN STRUCT) MINC(J,I) J= 1 TO NUMBER OF NODES / ELEMENT
 I= 1 TO NE
- 5 - (IN STRUCT) JOINT CONSTRAINTS
 JUM,JDIR
 AFTER LAST CONSTRAINT
 0,0
- 6 - (IN PROP) JOINT COORDINATES
 X(1,J),X(2,J),X(3,J) J= 1 TO NUMBER OF JOINTS
- 7 - (IN PROP) MEMBER PROPERTIES
 AREA(I),EMOD(I) I= 1 TO NE
- 8 - (IN JLOAD) INCREMENTAL JOINT LOADS
 JNUM,JDIR,FORCE
 AFTER LAST INCREMENTAL JOINT LOAD
 0,0,0
- 9 - (IN JLOAD) CONSTANT JOINT LOADS
 JNUM,JDIR,FORCE
 AFTER LAST CONSTANT JOINT LOAD
 0,0,0
- 10 - (IN MACT) MEMBER ACTIONS
 MN(I),MAT(I),ACT(I)
 AFTER LAST MEMBER ACTION
 0,0,0
- 11 - (IN MACT) INITIAL STRESSES
 MEM, TAU(1,MEM)
 AFTER LAST INITIAL STRESS
 30,0 (MEM GREATER THEN NE)

 * MAIN *

INITIALIZE PARAMETERS, CALL DATA, INITIALIZE VARIABLES,
 AND CALL NEWRAP, RIKWEM, ORTHOG, OR DYNAM.

PARAMETER (MX=30,MXNEQ=3*(MX-1),MXNA=25)
 IMPLICIT REAL*8 (A-H,O-Z)
 DIMENSION ACT(MX),AREA(MX),D(MXNEQ),DD(MXNEQ),DDO(MXNEQ),
 \$ DD1(MXNEQ),DD2(MXNEQ),DDO1(MXNEQ),DE(9),DDE(9),
 \$ DOT1(MXNEQ),DOT2(MXNEQ),ELENG(MX),EMOD(MX),F(MXNEQ),
 \$ FG(9,MXNEQ),FP(MXNEQ),FPI(MXNEQ),JCODE(3,MX),KHT(100),
 \$ MAT(MX),MAXA(100),MCODE(9,MX),MINC(3,MX),MN(MX),
 \$ PLACE(3,3),Q(MXNEQ),QJ(MXNEQ),QP(MXNEQ),QT(MXNEQ),
 \$ SKT(450),TAU(3,MX),WGT(3,3),X(3,MX),Z(MXNEQ)

INITIALIZE GAUSS QUADRATURE PARAMETERS
 - NUMBER OF GAUSS POINTS
 NGAUSS=3
 - GAUSS POINT LOCATIONS
 DATA PLACE /0.00,0.00,0.00, -.577350269189600,+.577350269189600
 \$,0.00,-.774596669241500,+.774596669241500,0.00/
 - WEIGHT
 DATA WGT /2.00,0.00,0.00, 1.00,1.00,0.00, .555555555555600,

```

$          .8888888888889D0,.5555555555556D0/
C
C  COVERGENCE PARAMETERS .....
C  - TOLDI1 : INCONV + 10
C    TOLDI1=1.D-03
C  - TOLDI2 : INCONV + 1
C    TOLDI2=1.D-02
C  - TOLFOR : INCONV + 100
C    TOLFOR = 1.00-01
C  - TOLENE : INCONV + 1000
C    TOLENE = 1.D-03+1.D0
C  - DIVERGENCE
C    DIVER = 1.D0
C
C    WRITE(6,50) TOLDI1,TOLDI2,TOLFOR,TOLENE,DIVER
C
C  TYPE OF OUTPUT .....
C  - NPRINT = 1 FOR EQUILIBRIUM PATH PLOT
C            2 FOR EQUILIBRIUM PATH RESULTS
C            3 FOR FULL DEBUGGING OUTPUT
C            4 FOR FULL FINAL OUTPUT
C
C    READ*,NPRINT
C    WRITE(6,70) NPRINT
C
C  LINEAR OR NONLINEAR ANALYSIS (FOR DEBUGGING) .....
C  - KFLG = 0 FOR NONLINEAR ANALYSIS
C          = 1 FOR LINEAR ANALYSIS
C    KFLG=0
C
C  ITERATION PROCESS FOR ANALYSIS .....
C  - ITEMTH = 1 NEWTON RAPHSON METHOD
C            2 MODIFIED RIKS WEMPNER METHOD
C            3 ORTHOGONAL TRAJECTORY ACCESSION
C            4 A DYNAMIC RELAXATION METHOD
C
C    ITEMTH=1
C    WRITE(6,80) ITEMTH
C
C  NUMBER OF ITERATIONS .....
C  - NUMBER OF ITERATIONS BETWEEN UPDATING SKT.
C    ITEUPD = 1
C  - NUMBER OF MAXIMUN ITERATIONS BETWEEN TWO EQUILIBRIUM POINTS.
C    ITEMAX = 10
C  - NUMBER OF ITERATIONS DESIRED FOR CONVERGENCE IN MODIFIED
C    RIKS WEMPNER METHOD.
C    ITDES = 1
C  - NUMBER OF MAXIMUM ITERATIONS IN THE DYNAMIC RELAXATION ROUTINE.
C    READ*,LCNT
C
C    WRITE(6,90) ITEUPD,ITEMAX,ITDES,LCNT
C
C  READ AND ECHO INITIAL DATA .....
C  READ*,QI,QIMAX,DQI
C  WRITE(6,100) QI,QIMAX,DQI
C  READ*,IMPOR
C  WRITE(6,200) IMPOR
C
C  CALL DATA (ACT,AREA,ELENG,EMOD,JCODE,KHT,MAT,MATFLG,MAXA,MCODE,
$    MINC,MN,MTYP,MX,MXNA,MXNEQ,NAT,NE,NEQ,NGAUSS,NJ,NKT,NPRINT,
$    PLACE,QJ,QP,TAU,WGT,X)
C
C  INITIALIZE VARIABLES .....
C  DO 30 I=1,NEQ
C    D(I)=0.D0
C    DD(I)=0.D0
C    Q(I)=QJ(I)
C    Z(I)=0.D0
C    FP(I)=0.D0
C    FPI(I)=0.D0
C 30 CONTINUE
C
C  IF (ITEMTH.EQ.1) THEN
C    CALL NEWRAP (ACT,AREA,D,DD,DD1,DE,DDE,DIVER,DQI,ELENG,EMOD,
$    F,FG,FP,FPI,IMPOR,ITEMAX,ITEUPD,JCODE,KFLG,MAT,MATFLG,
$    MAXA,MCODE,MINC,MN,MTYP,NAT,NE,NEQ,NGAUSS,NJ,NKT,NPRINT,
$    PLACE,Q,QI,QIMAX,QJ,QP,QT,SKT,TAU,TOLDI1,TOLDI2,TOLENE,
$    TOLFOR,WGT,X,Z,LCNT)
C
C  ELSE IF(ITEMTH.EQ.2) THEN
C    QI=0.D0
C    CALL RIKWEM(ACT,AREA,D,DD,DE,DDE,DDO,DD1,DD2,DD01,DIVER,DOT1,
$    DOT2,DQI,ELENG,EMOD,F,FG,FP,FPI,IMPOR,ITDES,ITEMAX,ITEUPD,
$    JCODE,KFLG,MAT,MATFLG,MAXA,MCODE,MINC,MN,MTYP,NAT,NE,NEQ,
$    NGAUSS,NJ,NKT,NPRINT,PLACE,Q,QI,QIMAX,QJ,QP,QT,SKT,TAU,

```

```

C      $      TOLDI1,TOLDI2,TOLENE,TOLFOR,WGT,X,Z)
C      ELSE IF(ITEMTH.EQ.3) THEN
C          CALL ORTHOG(ACT,AREA,D,DD,DE,DDE,DDO,DD1,DD2,DDO1,DIVER,DOT1,
C          $      DOT2,DQI,ELENG,EMOD,F,FG,FP,FPI,IMPOR,ITDES,ITEMAX,ITEUPD,
C          $      JCODE,KFLG,MAT,MATFLG,MAXA,MCODE,MINC,MN,MTYP,NAT,NE,NEQ,
C          $      NGAUSS,NJ,NKT,NPRINT,PLACE,Q,QI,QIMAX,QJ,QP,QT,SKT,TAU,
C          $      TOLDI1,TOLDI2,TOLENE,TOLFOR,WGT,X,Z)
C      ELSE IF(ITEMTH.EQ.4) THEN
C          NTRY=0
C          CALL DYNAM(ACT,AREA,D,DD,DD1,DE,DDE,DIVER,DQI,ELENG,EMOD,
C          $      F,FG,FP,FPI,IMPOR,ITEMAX,ITEUPD,JCODE,KFLG,MAT,MATFLG,
C          $      MAXA,MCODE,MINC,MN,MTYP,NAT,NE,NEQ,NGAUSS,NJ,NKT,NPRINT,
C          $      PLACE,Q,QI,QIMAX,QJ,QP,QT,SKT,TAU,TOLDI1,TOLDI2,TOLENE,
C          $      TOLFOR,WGT,X,Z,NTRY,LCNT)
C      END IF
C      WRITE(6,300)
C      50 FORMAT(//T10,'CONVERGENCE PARAMETERS'/T10,
C      $      'TOLDI1 ..... ',F5.3/T10,
C      $      'TOLDI2 ..... ',F5.3/T10,
C      $      'TOLFOR ..... ',F5.3/T10,
C      $      'TOLENE ..... ',F5.3/T10,
C      $      'DIVER ..... ',F5.3)
C      70 FORMAT(//T10,'NPRINT ..... ',I2)
C      80 FORMAT(//T10,'ITERATION METHOD .... ',I2)
C      90 FORMAT(//T10,'ITERATION PARAMETERS'/T10,
C      $      'ITEUPD ..... ',I3/T10,
C      $      'ITEMAX ..... ',I3/T10,
C      $      'ITEDES ..... ',I3/T10,
C      $      'LCNT ..... ',I3)
C      100 FORMAT(//T10,'LOADING PARAMETERS'/T10,
C      $      'QI ..... ',F9.6/T10,
C      $      'QIMAX ..... ',F9.6/T10,
C      $      'DQI ..... ',F9.6)
C      200 FORMAT(//T10,'DEGREE OF FREEDOM TO BE PLOTTED .... IMPOR ..',I4)
C      300 FORMAT(/T25,'***** NORMAL EXIT *****')
C      STOP
C      END
C      *****
C      -----*          DATA          *-----
C      *****
C      READS AND ECHO THE NUMBER OF ELEMENTS, NE, NUMBER OF JOINTS, NJ,
C      IF AT LEAST ONE EXCEEDS MX, PRINT ERROR MESSAGE AND STOP; ELSE
C      CALLS STRUCT AND LOADS.
C      SUBROUTINE DATA(ACT,AREA,ELENG,EMOD,JCODE,KHT,MAT,MATFLG,MAXA,
C      $      MCODE,MINC,MN,MTYP,MX,MXNA,MXNEQ,NAT,NE,NEQ,NGAUSS,NJ,
C      $      NKT,NPRINT,PLACE,QJ,QP,TAU,WGT,X)
C      IMPLICIT REAL*8 (A-H,O-Z)
C      DIMENSION ACT(*),AREA(*),ELENG(*),EMOD(*),JCODE(3,*),KHT(*),
C      $      MAT(*),MAXA(*),MCODE(9,*),MINC(3,*),MN(*),PLACE(3,*),
C      $      QJ(*),QP(*),TAU(3,*),X(3,*),WGT(3,*)
C      READ AND ECHO NE, NJ, MTYP .....
C      READ*,NE,NJ,MTYP
C      WRITE(6,100) NE,NJ,MTYP
C      CHECK LIMITS .....
C      IF (NE.LE.MX.AND.NJ.LE.MX) THEN
C          CALL STRUCT(AREA,ELENG,EMOD,JCODE,KHT,MAXA,MCODE,MINC,MTYP,
C          $      MXNEQ,NE,NEQ,NGAUSS,NJ,NKT,NPRINT,PLACE,WGT,X)
C          CALL LOAD(ACT,JCODE,MAT,MATFLG,MN,MXNA,MTYP,NAT,NE,NEQ,NPRINT,
C          $      QJ,QP,TAU)
C      ELSE
C          WRITE(6,200)
C          STOP
C      END IF
C      100 FORMAT(//T10,'NUMBER OF ELEMENTS .....NE...',I3/
C      $      T10,'NUMBER OF JOINTS .....NJ...',I3/
C      $      T10,'TYPE OF ELEMENTS .....MTYP.',I3)
C      200 FORMAT('***** ERROR : NE OR NJ EXCEEDS MX. *****')
C      RETURN
C      END
C      *****
C      -----*          STRUCT          *-----
C      *****
C      READ AND ECHO MEMBER INCIDENCES, MINC. INITIALIZE JOINT CODE,

```

```

C      JCODE, TO UNITY; READ AND ECHO FOR EACH CONSTRAINT THE JOINT
C      NUMBER, JNUM, AND THE JOINT DIRECTION, JDIR; STORE ZERO IN THE
C      CORRESPONDING LOCATION OF JCODE. CALL CODES, DETMAX, AND PROP.
C
C      SUBROUTINE STRUCT(AREA,ELENG,EMOD,JCODE,KHT,MAXA,MCODE,MINC,MTYP,
$      MXNEQ,NE,NEQ,NGAUSS,NJ,NKT,NPRINT,PLACE,WGT,X)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION AREA(*),ELENG(*),EMOD(*),JCODE(3,*),KHT(*),MAXA(*),
$      MCODE(9,*),MINC(3,*),PLACE(3,*),WGT(3,*),X(3,*)
C
C      READ AND ECHO MINC .....
      IF (MTYP.EQ.1) THEN
        WRITE(6,100)
        DO 5 I=1,NE
          READ*,MINC(1,I),MINC(2,I)
          WRITE(6,200) I,MINC(1,I),MINC(2,I)
5       CONTINUE
      ELSE
        WRITE(6,110)
        DO 7 I=1,NE
          READ*,MINC(1,I),MINC(2,I),MINC(3,I)
          WRITE(6,210) I,MINC(1,I),MINC(2,I),MINC(3,I)
7       CONTINUE
      END IF
C
C      INITIALIZE JCODE .....
      DO 20 L=1,3
        DO 10 J=1,NJ
          JCODE(L,J)=1
10      CONTINUE
20     CONTINUE
C
C      READ CONSTRAINTS AND STORE ZERO IN JCODE .....
      WRITE(6,300)
      READ*,JNUM,JDIR
      IF ( JNUM.NE.0 ) THEN
30      IF ( JNUM.NE.0 ) THEN
          WRITE(6,400) JNUM,JDIR
          JCODE(JDIR,JNUM)= 0
          READ*, JNUM,JDIR
          GO TO 30
        END IF
      END IF
C
      CALL CODES(JCODE,MCODE,MINC,MTYP,MXNEQ,NE,NEQ,NJ,NPRINT)
      CALL DETMAX(KHT,MAXA,MCODE,MTYP,NE,NEQ,NKT)
      CALL PROP(AREA,ELENG,EMOD,MINC,MTYP,NE,NGAUSS,NJ,NPRINT,PLACE,
$      WGT,X)
      ELSE
        WRITE(6,500)
        STOP
      END IF
C
100  FORMAT(/T10,'MEMBER INCIDENCE (MINC) :'/T10,'ELEMENT NODE:',
$      T26,'1 2')
110  FORMAT(/T10,'MEMBER INCIDENCE (MINC) :'/T10,'ELEMENT NODE:',
$      T26,'1 2 3')
200  FORMAT(T11,I3,T24,2(I3,4X))
210  FORMAT(T11,I3,T24,3(I3,4X))
300  FORMAT(/T10,'JOINT CONSTRAINTS :'/T10,'JOINT DIRECTION'/)
400  FORMAT(T11,I3,T22,I1)
500  FORMAT('***** ERROR: NO JOINT CONSTRAINTS IN THE STRUCTURE *****')
C
      RETURN
      END
C
C      *****
C-----*                               LOAD                               *-----
C      *****
C
C      INITIALIZE TO ZERO THE JOINT LOAD DISTRIBUTION VECTOR, QJ, AND
C      THE NON-INCREMENTAL JOINT LOAD VECTOR, QP; CALL JLOAD AND MACT.
C
C      SUBROUTINE LOAD(ACT,JCODE,MAT,MATFLG,MN,MXNA,MTYP,NAT,NE,NEQ,
$      NPRINT,QJ,QP,TAU)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION ACT(*),JCODE(3,*),MAT(*),MN(*),QJ(*),QP(*),TAU(3,*)
C
C      INITIALIZE QJ AND QP .....
      DO 10 I=1,NEQ
        QJ(I)=0.000
        QP(I)=0.000
10     CONTINUE
C
      CALL JLOAD (JCODE,QJ,QP)
      CALL MACT (ACT,MAT,MATFLG,MN,MXNA,MTYP,NAT,NE,NPRINT,TAU)
C
      RETURN

```

```

C      END
C
C      *****
C-----*          CODES          *-----
C      *****
C
C      GENERATE THE JOINT CODE, JCODE, BY ASSIGNING INTEGERS IN SEQUENCE
C      BY COLUMNS, TO ALL NONZERO ELEMENTS OF JCODE FROM 1 TO NEQ. TEST
C      THE NUMBER OF EQUATIONS, NEQ, AGAINST MXNEQ. GENERATE THE MEMBER
C      CODE, MCODE, BY TRANSFERING VIA MINC COLUMNS OF JCODE INTO COLUMNS
C      OF MCODE. PRINT JCODE AND MCODE.
C
C      SUBROUTINE CODES (JCODE,MCODE,MINC,MTYP,MXNEQ,NE,NEQ,NJ,NPRINT)
C      IMPLICIT REAL*8 (A-H,O-Z)
C      DIMENSION JCODE(3,*),MCODE(9,*),MINC(3,*)
C
C      GENERATE JCODE .....
C      NEQ=0
C      DO 20 J=1,NJ
C        DO 10 L=1,3
C          IF (JCODE(L,J).NE.0) THEN
C            NEQ=NEQ+1
C            JCODE(L,J)=NEQ
C          END IF
C        10 CONTINUE
C      20 CONTINUE
C
C      CHECK LIMITS .....
C      IF (NEQ.GT.MXNEQ) THEN
C        WRITE(6,100)
C        STOP
C      END IF
C
C      GENERATE MCODE .....
C      IF (MTYP.EQ.1) THEN
C        NN=2
C      ELSE
C        NN=3
C      END IF
C      DO 40 I=1,NE
C        DO 30 L=1,NN
C          J=MINC(L,I)
C          N=3*L
C          M=N-1
C          K=M-1
C          MCODE(K,I)=JCODE(1,J)
C          MCODE(M,I)=JCODE(2,J)
C          MCODE(N,I)=JCODE(3,J)
C        30 CONTINUE
C      40 CONTINUE
C
C      PRINT JCODE AND MCODE .....
C      WRITE(6,200)
C      DO 50 J=1,NJ
C        WRITE(6,300) J,(JCODE(I,J),I=1,3)
C      50 CONTINUE
C      WRITE(6,400)
C      DO 60 J=1,NE
C        WRITE(6,500) J,(MCODE(I,J),I=1,NN*3)
C      60 CONTINUE
C
C      100 FORMAT('***** ERROR: NEQ EXCEEDS MXNEQ *****')
C      200 FORMAT('//T10,'JOINT CODE (JCODE) :'/T10,'JOINT')
C      300 FORMAT(T11,I3,T20,3(I3,4X))
C      400 FORMAT('//T10,'MEMBER CODE (MCODE) :'/T10,'ELEMENT')
C      500 FORMAT(T11,I3,T20,9(I3,4X))
C
C      RETURN
C      END
C
C      *****
C-----*          DETMAX          *-----
C      *****
C
C      DETERMINE THE COLUMN HEIGHT OF FULL TANGENT STIFFNESS MATRIX,
C      KHT, THE ADDRESSES OF THE DIAGONAL TERMS, MAXA; AND THE NUMBER
C      OF ELEMENTS BELOW THE SKYLINE, NKT.
C
C      SUBROUTINE DETMAX ( KHT,MAXA,MCODE,MTYP,NE,NEQ,NKT )
C      IMPLICIT REAL*8 (A-H,O-Z)
C      DIMENSION KHT(*),MAXA(*),MCODE(9,*)
C
C      INITIALIZE KHT .....
C      DO 10 I=1,NEQ
C        KHT(I)=0

```

```

10 CONTINUE
C
C GENERATE KHT .....
IF (MTYP.EQ.1)THEN
  NN=2
ELSE
  NN=3
END IF
DO 30 I=1,NE
  J= 1
15 IF (MCODE(J,I).EQ.0) THEN
  J=J+1
  GO TO 15
END IF
MIN=MCODE(J,I)
J=J+1
DO 20 L=J,NN*3
  K=MCODE(L,I)
  IF ( K.NE.0 ) THEN
    KHT(K)= MAX0( KHT(K),(K-MIN) )
  END IF
20 CONTINUE
30 CONTINUE
C
C GENERATE MAXA AND FIND NKT .....
MAXA(1)=1
DO 40 I=1,NEQ
  MAXA(I+1)= MAXA(I) + KHT(I) + 1
40 CONTINUE
NKT= MAXA(NEQ+1) - 1
C
C PRINT KHT,MAXA,NKT .....
WRITE(6,100)
DO 50 I=1,NEQ
  WRITE(6,200) I,KHT(I),MAXA(I)
50 CONTINUE
WRITE(6,300) NKT
C
100 FORMAT(/T10,'COLUMN',4X,'COLUMN HEIGHT',4X,'ADDRESS OF DIAG.'/
$ T13,'I',10X,'KHT(I)',10X,'MAXA(I)')
200 FORMAT(T11,I3,10X,I3,10X,I3)
300 FORMAT(/T10,'NUMBER OF ELEMENTS IN THE LAST COLUMN, NKT :',I7)
C
RETURN
END
C
C *****
C-----* PROP *-----
C *****
C
C READ AND ECHO THE JOINT COORDINATES, X; COMPUTE FOR EACH ELEMENT
C THE LENGTH, ELENG. READ THE CROSS SECTIONAL AREA AND THE MODULUS
C OF ELASTICITY. PRINT ELEMENT PROPERTIES.
C
SUBROUTINE PROP(AREA,ELENG,EMOD,MINC,MTYP,NE,NGAUSS,NJ,NPRINT,
$ PLACE,WGT,X)
$ IMPLICIT REAL*8(A-H,O-Z)
$ DIMENSION AREA(*),ELENG(*),EMOD(*),MINC(3,*),PLACE(3,*),WGT(3,*),
$ X(3,*),XL(9),YL(9),ZL(9)
C
IF (NPRINT.EQ.3) THEN
  WRITE(6,200)
END IF
C
C READ AND ECHO JOINT COORDINATES .....
WRITE(6,210)
DO 30 J=1,NJ
  READ*,X(1,J),X(2,J),X(3,J)
  WRITE(6,220) J,X(1,J),X(2,J),X(3,J)
30 CONTINUE
C
C COMPUTE ELEMENT LENGTH AND READ ELEMENT PROPERTIES .....
WRITE(6,250)
IF (MTYP .EQ. 1) THEN
  LINEAR ELEMENT .....
  DO 60 I=1,NE
    J=MINC(1,I)
    K=MINC(2,I)
    EL1=X(1,K)-X(1,J)
    EL2=X(2,K)-X(2,J)
    EL3=X(3,K)-X(3,J)
    ELENG(I)=DSQRT(EL1*EL1+EL2*EL2+EL3*EL3)
C
    READ*,AREA(I),EMOD(I)
    WRITE(6,230) I,AREA(I),EMOD(I),ELENG(I)
60 CONTINUE
C

```



```

C      END
C-----*-----*****
C      *-----NEWRAP-----*-----
C      *-----*****
C      PERFORM THE NEWTON-RAPHSON OR THE MODIFIED NEWTON-RAPHSON METHOD.
C      CALL STIFF, SOLVE, FORCES, TEST, RESULT, UPDATC AND UPDATF.
C
C      SUBROUTINE NEWRAP( ACT, AREA, D, DD, DD1, DE, DDE, DIVER, DQI, ELENG, EMOD,
C      $ F, FG, FP, FPI, IMPOR, ITEMAX, ITEUPD, JCODE, KFLG, MAT, MATFLG,
C      $ MAXA, MCODE, MINC, MN, MTYPE, NAT, NE, NEQ, NGAUSS, NJ, NKT, NPRINT,
C      $ PLACE, Q, QI, QIMAX, QJ, QP, QT, SKT, TAU, TOLDI1, TOLDI2, TOLENE,
C      $ TOLFOR, WGT, X, Z, LCNT)
C      IMPLICIT REAL*8 ( A-H, O-Z)
C      DIMENSION ACT(*), AREA(*), D(*), DD(*), DD1(*), DDE(*), DE(*), ELENG(*),
C      $ EMOD(*), F(*), FG(9,*), FP(*), FPI(*), JCODE(3,*), MAT(*),
C      $ MAXA(*), MCODE(9,*), MINC(3,*), MN(*), PLACE(3,*), Q(*),
C      $ QJ(*), QP(*), QT(*), SKT(*), TAU(3,*), WGT(3,*), X(3,*), Z(*)
C
C      CALL FORCES( ACT, AREA, DD, DDE, ELENG, EMOD, F, FG, MAT, MCODE, MINC, MN,
C      $ MTYPE, NAT, NE, NEQ, NGAUSS, NPRINT, PLACE, Q, QJ, TAU, WGT, X)
C      CALL UPDATF( NEQ, Q, QI, QP, QT)
C
C      ITERATION FROM THE FIRST LOAD INCREMENT TO QIMAX .....
C 5 IF (QI.LE.QIMAX) THEN
C      ITENUM = ITEUPD
C      ITECNT = 0
C      INCONV = 1
C      NTRY = 0
C      KSTBL=0
C
C      ITERATION BETWEEN TWO EQUILIBRIUM POINTS .....
C 6 IF (INCONV.NE.0.AND.ITECNT.LE.ITEMAX) THEN
C      IF (ITENUM.GE.ITEUPD) THEN
C      $ CALL STIFF( AREA, ELENG, EMOD, KFLG, MAXA, MCODE, MINC,
C      $ MTYPE, NE, NGAUSS, NKT, NPRINT, PLACE, SKT, TAU, WGT, X)
C      ITENUM = 0
C      END IF
C
C      CALL SOLVE ( DD, F, ITENUM, KSTBL, MAXA, NEGPIV, NEQ, NKT, NPRINT, QT,
C      $ SKT)
C
C      IF (KSTBL.NE.0 .AND. NTRY.LE.2) THEN
C      $ CALL DYNAM( ACT, AREA, D, DD, DD1, DE, DDE, DIVER, DQI, ELENG, EMOD,
C      $ F, FG, FP, FPI, IMPOR, ITEMAX, ITEUPD, JCODE, KFLG, MAT, MATFLG,
C      $ MAXA, MCODE, MINC, MN, MTYPE, NAT, NE, NEQ, NGAUSS, NJ, NKT, NPRINT,
C      $ PLACE, Q, QI, QI, QJ, QP, QT, SKT, TAU, TOLDI1, TOLDI2, TOLENE,
C      $ TOLFOR, WGT, X, Z, NTRY, LCNT)
C      KSTBL=0
C      ITENUM=ITEUPD
C      GO TO 6
C      ELSE IF (KSTBL.NE.0 .AND. NTRY .GT. 2) THEN
C      $ WRITE(6,175)
C      $ STOP
C      END IF
C
C      IF (ITECNT.EQ.0) THEN
C      $ DO 10 I=1, NEQ
C      $ DD1(I) = DD(I)
C      $ CONTINUE
C      $ END IF
C      $ DO 20 I=1, NEQ
C      $ D(I)=D(I)+DD(I)
C      $ FPI(I)=F(I)
C      $ CONTINUE
C      $ CALL UPDATC ( DD, JCODE, NJ, NPRINT, X)
C
C      CALL FORCES( ACT, AREA, DD, DDE, ELENG, EMOD, F, FG, MAT, MCODE, MINC,
C      $ MN, MTYPE, NAT, NE, NEQ, NGAUSS, NPRINT, PLACE, Q, QJ, TAU, WGT, X)
C
C      IF (MATFLG .NE. 0) THEN
C      $ CALL UPDATF( NEQ, Q, QI, QP, QT)
C      $ END IF
C      $ CALL TEST ( D, DD, DD1, DIVER, F, FP, FPI, INCONV, JCODE, NEQ, NJ,
C      $ NPRINT, QT, TOLDI1, TOLDI2, TOLENE, TOLFOR)
C
C      IF (NPRINT.EQ.3) THEN
C      $ WRITE(6,150) ITECNT
C      $ DO 30 I=1, NEQ
C      $ WRITE(6,200) QT(I), F(I), D(I)
C      $ CONTINUE
C      $ END IF
C      $ ITENUM = ITENUM + 1
C      $ ITECNT = ITECNT + 1
C      $ GO TO 6

```

```

C      END IF
C      IF (NPRINT.EQ.3) THEN
C        WRITE(6,300) INCONV,ITECNT
C      END IF
C      DO 40 I=1,NEQ
C        FP(I)=F(I)
40     CONTINUE
C      CALL RESULT (D,FG,IMPOR,JCODE,MINC,MTYP,NE,NEQ,NJ,NPRINT,
C        $      QI,QT,TAU,X)
C      IF (INCONV.NE.0) THEN
C        WRITE(6,400)
C        STOP
C      END IF
C      QI = QI + DQI
C      CALL UPDATF (NEQ,Q,QI,QP,QT)
C      GO TO 5
C      END IF
C      150 FORMAT(/T10,'NEWTON-RAPHSON ITERATION ',I5/T15,'QT',20X,'F',
C        $      20X,'D')
C      175 FORMAT(/' ***** POOR TRIAL CONFIGURATION *****')
C      200 FORMAT(T10,3(5X,D15.7))
C      300 FORMAT('0INCONV=',I4,5X,'ITECNT=',I4)
C      350 FORMAT(' ',20X,F13.7/' ',20X,F14.8)
C      400 FORMAT(' ***** LAST SOLUTION IS NOT CONVERGED *****')
C      RETURN
C      END
C      *****
C-----*          RIKWEM          *-----
C      *****
C      RIKWEM PERFORMS MODIFIED RIKS/WEMPNER ITERATION ON A NORMAL
C      STIFF, SOLVE, FORCES, TEST, RESULT, UPDATE, AND DOTPRD ARE
C      CALLED
C
C      SUBROUTINE RIKWEM( ACT,AREA,D,DD,DE,DDE,DDO,DD1,DD2,DDO1,DIVER,
C        $      DOT1,DOT2,DQI,ELENG,EMOD,F,FG,FP,FPI,IMPOR,ITDES,
C        $      ITEMAX,ITEUPD,JCODE,KFLG,MAT,MATFLG,MAXA,MCODE,MINC,
C        $      MN,MTYP,NAT,NE,NEQ,NGAUSS,NJ,NKT,NPRINT,PLACE,Q,QI,
C        $      QIMAX,QJ,QP,QT,SKT,TAU,TOLDI1,TOLDI2,TOLENE,TOLFOR,WGT,
C        $      X,Z)
C      IMPLICIT REAL*8 (A-H,O-Z)
C      DIMENSION ACT(*),AREA(*),D(*),DD(*),DD1(*),DDE(*),DE(*),DDO1(*),
C        $      DD2(*),DDO(*),DOT1(*),DOT2(*),ELENG(*),EMOD(*),F(*),
C        $      FGI(9,*),FP(*),FPI(*),JCODE(3,*),MAT(*),MAXA(*),
C        $      MCODE(9,*),MINC(3,*),MN(*),PLACE(3,*),Q(*),QJ(*),QP(*),
C        $      QT(*),SKT(*),TAU(3,*),WGT(3,*),X(3,*),Z(*),DDP(100)
C
C      ITECNT=0
C      NEGPIV=0
C      KSTBL=0
C      NTRY=0
C      ITP=4
C
C      CALL FORCES( ACT, AREA, DD, DDE, ELENG, EMOD, F, FG, MAT, MCODE, MINC, MN,
C        $      MTYP, NAT, NE, NEQ, NGAUSS, NPRINT, PLACE, Q, QJ, TAU, WGT, X)
C
C      SET UP TANGENT VECTOR, DS .....
5     IF(QI.LE.QIMAX.AND.ITECNT.LE.ITEMAX) THEN
C      CALL STIFF(AREA,ELENG,EMOD,KFLG,MAXA,MCODE,MINC,MTYP,NE,
C        $      NGAUSS,NKT,NPRINT,PLACE,SKT,TAU,WGT,X)
C      CALL SOLVE(DDO1,Z,0,KSTBL,MAXA,NEGPIV,NEQ,NKT,NPRINT,Q,SKT)
C      IF (KSTBL .NE. 0) THEN
C        STOP
C      END IF
C
C      COMPUTE DS FOR THE FIRST STEP OF THE FIRST ITERATION,
C      COMPUTE DQI FOR ALL REMAINING ITERATIONS .....
C      IF(ITECNT.EQ.0) THEN
C        DS=DQI*DSQRT( DOTPRD(DDO1,DDO1,NEQ)+1.DO)
C        DSMAX=DS*2.
C      ELSE
C        DQI=DS/DSQRT( DOTPRD(DDO1,DDO1,NEQ)+1.DO)
C        TEMP=DQI*( DOTPRD(DDO1,DDP,NEQ)+DQI1)
C        IF (TEMP .GT. 0.000) THEN
C          SGN=1.000
C        ELSE
C          SGN=-1.000
C        END IF
C        DQI=SGN*DQI

```

```

C      END IF
C
C      SAVE THE VALUES FOR THE FIRST TRIAL CONFIGURATION;
C      UPDATE THE CONFIGURATION .....
      DQI1=DQI
      DO 10 I=1,NEQ
          DDP(I)=0.000
          DDO(I)=DQI*DDO1(I)
          D(I)=D(I)+DDO(I)
          DDP(I)=DDP(I)+DDO(I)
          DD(I)=DDO(I)
10     CONTINUE
      QI=QI+DQI
C
C      CALL UPDATC(DD,JCODE,NJ,NPRINT,X)
C
C      IF(NPRINT.EQ.3) THEN
C          WRITE(6,500) QI,D(IMPOR)
C      END IF
C
C      CALL FORCES(ACT,AREA,DD,DDE,ELENG,EMOD,F,FG,MAT,MCODE,MINC,
$      MN,MTYP,NAT,NE,NEQ,NGAUSS,NPRINT,PLACE,Q,QJ,TAU,WGT,X)
      INCONV=1
      ITEUM=ITEUPD
      IT=0
C
C      ITERATE ALONG THE NORMAL .....
7     IF(INCONV.NE.0.AND.IT.LE.ITEMAX) THEN
C         CALL UPDATF(NEQ,Q,QI,QP,QT)
C
C         IF(ITEUM.GE.ITEUPD) THEN
C             CALL STIFF(AREA,ELENG,EMOD,KFLG,MAXA,MCODE,MINC,MTYP,
$             NE,NGAUSS,NKT,NPRINT,PLACE,SKT,TAU,WGT,X)
C             ITEUM=0
C         END IF
C
C         CALL SOLVE(DD1,Z,ITEUM,KSTBL,MAXA,NEGPIV,NEQ,NKT,NPRINT,Q,
$         SKT)
C         CALL SOLVE(DD2,F,1,KSTBL,MAXA,NEGPIV,NEQ,NKT,NPRINT,QT,SKT)
C
C         APPLY THE CONSTRAINT EQUATION .....
C         DQI=- (DOTPRD(DDO,DD2,NEQ))/(DOTPRD(DDO,DD1,NEQ)+DQI1)
C
C         UPDATE THE CONFIGURATION .....
      DO 20 I=1,NEQ
          DD(I)=DQI*DD1(I)+DD2(I)
          D(I)=D(I)+DD(I)
          DDP(I)=DDP(I)+DD(I)
20     CONTINUE
C
C         CALL UPDATC(DD,JCODE,NJ,NPRINT,X)
C         CALL FORCES(ACT,AREA,DD,DDE,ELENG,EMOD,F,FG,MAT,MCODE,
$         MINC,MN,MTYP,NAT,NE,NEQ,NGAUSS,NPRINT,PLACE,Q,QJ,TAU,
$         WGT,X)
C
C         QI=QI+DQI
C
C         CALL TEST(D,DD,DD1,DIVER,F,FP,FPI,INCONV,JCODE,NEQ,NJ,
$         NPRINT,QT,TOLDI1,TOLDI2,TOLENE,TOLFOR)
C         ITEUM=ITEUM+1
C         IT=IT+1
C
C         IF(NPRINT.EQ.3) THEN
C             WRITE(6,600) IT,QI,D(IMPOR),INCONV
C         END IF
C         GO TO 7
C         END IF
C         ITECNT=ITECNT+1
C         IF(NPRINT.EQ.3) THEN
C             WRITE(6,700) IT
C         END IF
C
C         CALL RESULT(D,FG,IMPOR,JCODE,MINC,MTYP,NE,NEQ,NJ,NPRINT,
$         QI,QT,TAU,X)
C
C         IF(INCONV.NE.0) THEN
C             WRITE(6,800)
C             STOP
C         END IF
C
C      SCALE DS TO CONTROL NUMBER OF ITERATIONS BETWEEN
C      EQUILIBRIUM POINTS .....
C      IF (IT.LE.2 .AND. ITP.LE.2) THEN
C          DS=DS*1.500
C          DS=DS*DSQRT(DFLOAT(ITDES)/DFLOAT(IT))
C          IF (DABS(DS).GT.DSMAX) THEN

```



```

      DQI=- (DOTPRD( DD1,DD2,NEQ ))/(DOTPRD( DD1,DD1,NEQ )+1.00)
C
C   UPDATE THE CONFIGURATION .....
      DO 20 I=1,NEQ
          DD(I)=DQI*DD1(I)+DD2(I)
          D(I)=D(I)+DD(I)
20   CONTINUE
      QI=QI+DQI
C
C   CALL UPDATC(DD,JCODE,NJ,NPRINT,X)
C
      IF (NPRINT.EQ.3) THEN
          WRITE(6,200) LCNT,QI,D(IMPOR)
      END IF
      IF (NPRINT.EQ.1) THEN
          WRITE(6,400) QI,D(IMPOR)
      END IF
C
      CALL FORCES(ACT,AREA,DD,DDE,ELENG,EMOD,F,FG,MAT,MCODE,MINC,
$           MN,MTYP,NAT,NE,NEQ,NGAUSS,NPRINT,PLACE,Q,QJ,TAU,WGT,X)
C
      CALL TEST(D,DD,DD1,DIVER,F,FP,FPI,INCONV,JCODE,NEQ,NJ,
$           NPRINT,QT,TOLDI1,TOLDI2,TOLENE,TOLFOR)
C
      LCNT=LCNT+1
C
      GO TO 10
      END IF
C
C   24 CONTINUE
C
      DO 21 I=1,NEQ
          DDP(I)=0.000
21   CONTINUE
      DQI1=1.00
C
      SET UP TANGENT VECTOR, DS .....
30 IF(QI.LE.QIMAX.AND.ITECNT.LE.ITEMAX) THEN
      CALL STIFF(AREA,ELENG,EMOD,KFLG,MAXA,MCODE,MINC,MTYP,NE,
$           NGAUSS,NKT,NPRINT,PLACE,SKT,TAU,WGT,X)
      CALL SOLVE(DD01,Z,0,KSTBL,MAXA,NEGPIV,NEQ,NKT,NPRINT,Q,SKT)
      IF (KSTBL .NE. 0) THEN
          STOP
      END IF
      PRINT*,'DD01',DD01(IMPOR)
C
C   COMPUTE DQI FOR TRIAL EQUILIBRIUM POINT .....
      DQI=DS/DSQRT(DOTPRD(DD01,DD01,NEQ)+1.00)
      TEMP=DQI*(DOTPRD(DD01,DDP,NEQ)+DQI1)
      IF (TEMP .GT. 0.000) THEN
          SGN=1.000
      ELSE
          SGN=-1.000
      END IF
      DQI=SGN*DQI
C
C   SAVE THE VALUES FOR THE FIRST TRIAL CONFIGURATION;
C   UPDATE THE CONFIGURATION .....
      DQI1=DQI
      DO 40 I=1,NEQ
          DD(I)=DQI*DD01(I)
          D(I)=D(I)+DD(I)
          DDP(I)=DD(I)
40   CONTINUE
      QI=QI+DQI
C
C
      CALL UPDATC(DD,JCODE,NJ,NPRINT,X)
C
      IF(NPRINT.EQ.3) THEN
          WRITE(6,500) QI,D(IMPOR)
      END IF
C
      CALL FORCES(ACT,AREA,DD,DDE,ELENG,EMOD,F,FG,MAT,MCODE,MINC,
$           MN,MTYP,NAT,NE,NEQ,NGAUSS,NPRINT,PLACE,Q,QJ,TAU,WGT,X)
      INCONV=1
      ITENUM=ITEUPD
      IT=0
C
C   ITERATE ALONG THE NORMAL TO THE CURRENT TANGENT .....
50 IF (INCONV.NE.0.AND.IT.LE.ITEMAX) THEN
      CALL UPDATF(NEQ,Q,QI,QP,QT)
C
      IF (ITENUM.GE.ITEUPD) THEN
          CALL STIFF(AREA,ELENG,EMOD,KFLG,MAXA,MCODE,MINC,MTYP,
$           NE,NGAUSS,NKT,NPRINT,PLACE,SKT,TAU,WGT,X)

```



```

$      TOLFOR,WGT,X,Z,NTRY,LCNT)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION ACT(*),AREA(*),D(*),DD(*),DD1(*),DDE(*),DE(*),ELENG(*),
$      EMOD(*),F(*),FG(9,*),FP(*),FPI(*),JCODE(3,*),MAT(*),
$      MAXA(*),MCODE(9,*),MINC(3,*),MN(*),PLACE(3,*),Q(*),
$      QJ(*),QP(*),QT(*),SKT(*),TAU(3,*),WGT(3,*),X(3,*),Z(*),
$      DMASS(60),QTEMP(60),AC(60),ACP(60),DP(60),ZP(60),
$      FPTEMP(60)

C      WRITE(6,100) NTRY
C
C      INITIALIZE VARIABLES .....
C      DT=1.000
C      READ*,DT
C      DUM1=1.100*1.100/4.DO
C      CALL FORCES(ACT,AREA,DD,DDE,ELENG,EMOD,F,FG,MAT,MCODE,
$      MINC,MN,MTYP,NAT,NE,NEQ,NGAUSS,NPRINT,PLACE,Q,QJ,
$      TAU,WGT,X)
C      CALL UPDATF(NEQ,Q,QI,QP,QT)
C
C      ITERATION FROM THE FIRST LOAD INCREMENT TO QIMAX .....
14 IF (QI.LE.QIMAX) THEN
    INCONV=1
    LNT=0
    MFLG=1
    NREF=1
    DO 10 I=1,NEQ
      QTEMP(I)=QT(I)
      ACP(I)=0.DO
      ZP(I)=0.DO
      FP(I)=0.DO
10 CONTINUE
15 IF (INCONV.NE.0 .AND. LNT.LE.LCNT) THEN
    C      FORM (REFORM) MASS MATRIX .....
    C      IF (MFLG.NE.0) THEN
    C
    C      DO 18 I=1,NEQ
    C      DMASS(I)=0.DO
18 CONTINUE
    C
    C      CALL STIFF(AREA,ELENG,EMOD,KFLG,MAXA,MCODE,MINC,MTYP,
    C      NE,NGAUSS,NKT,NPRINT,PLACE,SKT,TAU,WGT,X)
    C
    C      CALL EIGEN(MAXA,NEQ,NPRINT,SKT)
    C
    C      DO 30 I=1,NEQ
    C      DO 20 J=I,NEQ
    C      KHTT=MAXA(J+1)-MAXA(J)-1
    C      NA=J-KHTT-1
    C      NB=J-I
    C      IF (NA.GE.I) THEN
    C      SS=0.DO
    C      ELSE
    C      SS=SKT(MAXA(J)+NB)
    C      END IF
    C      DMASS(I)=DMASS(I)+DABS(SS)*DUM1
    C      IF (I.NE.J) THEN
    C      DMASS(J)=DMASS(J)+DABS(SS)*DUM1
    C      END IF
20 CONTINUE
30 CONTINUE
    MFLG=0
    C
    C      IF (NPRINT.EQ.3) THEN
    C      WRITE(6,245)
    C      WRITE(6,250) (DMASS(I),I=1,NEQ)
    C      END IF
    C      END IF
    C
    C      COMPUTE CURRENT VELOCITY VECTOR .....
    C      IF (LNT.EQ.0) THEN
    C      DO 40 I=1,NEQ
    C      Z(I)=DT*(QTEMP(I)-F(I))/(2.DO*DMASS(I))
    C      Z(I)=DT*(QTEMP(I)-F(I))/(2.DO*30)
    C      AC(I)=2*Z(I)/DT
40 CONTINUE
    ELSE
    TEMP1=0.DO
    TEMP2=0.DO
    DO 50 I=1,NEQ
    IF (ZP(I).EQ.0.DO) THEN
    SL=0.DO
    ELSE
    SL=(F(I)-FPI(I))/(DT*ZP(I))
    END IF
    TEMP1=TEMP1+D(I)*D(I)*SL

```

```

        TEMP2=TEMP2+D(I)*D(I)*DMASS(I)
        TEMP2=TEMP2+D(I)*D(I)*30
50      CONTINUE
        TEMP3=TEMP1/TEMP2
C
C      COMPUTE NEW DAMPING COEFFICIENT .....
        IF (TEMP3.LE.0) THEN
            VN=0.00
        ELSE IF (TEMP3.GE.4) THEN
            VN=1.900
        ELSE
            VN=2*DT*DSQRT(TEMP3)
        END IF
C
        IF (NPRINT.EQ.3) THEN
            WRITE(6,400) VN
        END IF
C
        T1=2-VN
        T2=2+VN
C
        DO 60 I=1,NEQ
            ZP(I)=Z(I)
            Z(I)=T1*Z(I)/T2+2*DT*(QTEMP(I)-F(I))/(DMASS(I)*T2)
            Z(I)=T1*Z(I)/T2+2*DT*(QTEMP(I)-F(I))/(30*T2)
            ACP(I)=AC(I)
            AC(I)=(Z(I)-ZP(I))/DT
60      CONTINUE
        END IF
C
C      COMPUTE NEW DISPLACEMENT VECTOR .....
        DO 70 I=1,NEQ
            DP(I)=D(I)
            D(I)=D(I)+DT*Z(I)
            DD(I)=D(I)-DP(I)
            FPI(I)=F(I)
70      CONTINUE
C
        IF (NPRINT.EQ.3) THEN
            WRITE(6,345)
            WRITE(6,350) (D(I),I=1,NEQ)
        END IF
C
C      IF (LNT.NE.0) THEN
        COMPUTE ERROR MEASURE .....
        E=0.00
        DO 80 I=1,NEQ
            DUM2=4.00*DABS(D(I)-DP(I))
            IF (DUM2.EQ.0.00) THEN
                DUM3=0.00
            ELSE
                DUM3=DT*DT*DABS(AC(I)-ACP(I))/DUM2
            END IF
            E=DMAX1(E,DUM3)
80      CONTINUE
C
        IF (E.LT.1.00) THEN
C
C          UPDATE THE CONFIGURATION AND TEST FOR CONVERGENCE ...
            CALL UPDATC(DD,JCODE,NJ,NPRINT,X)
C
            CALL FORCES(ACT,AREA,DD,DDE,ELENG,EMOD,F,FG,MAT,MCODE,
$             MINC,MN,MTYP,NAT,NE,NEQ,NGAUSS,NPRINT,PLACE,Q,QJ,
$             TAU,WGT,X)
C
            CALL TEST(D,DD,DD1,DIVER,F,FP,FPI,INCONV,JCODE,
$             NEQ,NJ,NPRINT,QT,TOLDI1,TOLDI2,TOLENE,TOLFOR)
C
            LNT=LNT+1
            NREF=1
        ELSE
            MFLG=1
            NREF=NREF+1
            IF (NREF.GT.10) THEN
                WRITE(6,600)
                STOP
            END IF
        END IF
C
        END IF
        IF (NPRINT.EQ.3) THEN
            WRITE(6,200) LNT
        END IF
        GO TO 15
C
    END IF

```



```

C          $          XL(9),YL(9),ZL(9),MINC(3,*)
C
C      INITIALIZE LOCAL TANGENT STIFFNESS MATRIX, SE .....
C      DO 8 I=1,9
C          DO 5 J=1,9
C              SE(I,J)=0.000
C      5 CONTINUE
C      8 CONTINUE
C
C      COMPUTE LINEAR CONTRIBUTION TO LOCAL TANGENT STIFFNESS MATRIX ...
C      DO 10 L=1,2
C          J=MINC(L,N)
C          XL(L)=X(1,J)
C          YL(L)=X(2,J)
C          ZL(L)=X(3,J)
C          IF (N.GT.7) THEN
C              END IF
C      10 CONTINUE
C
C      TEMP1=XL(2)-XL(1)
C      TEMP2=YL(2)-YL(1)
C      TEMP3=ZL(2)-ZL(1)
C      TEMP4=AREA(N)*EMOD(N)/ELENG(N)**3
C
C      BL(1)=-TEMP1
C      BL(2)=-TEMP2
C      BL(3)=-TEMP3
C      BL(4)= TEMP1
C      BL(5)= TEMP2
C      BL(6)= TEMP3
C
C
C      DO 30 I=1,6
C          DO 20 J=1,6
C              SE(I,J)=SE(I,J)+BL(I)*BL(J)*TEMP4
C      20 CONTINUE
C      30 CONTINUE
C
C      COMPUTE THE NONLINEAR CONTRIBUTION (UPPER TRI. ONLY) .....
C      IF (KFLG.EQ. 0) THEN
C          DUM1=TAU(1,N)*AREA(N)/ELENG(N)
C          DO 40 J=1,6
C              SE(J,J)=SE(J,J)+DUM1
C      40 CONTINUE
C          DO 50 J=1,3
C              SE(J,J+3)=SE(J,J+3)-DUM1
C      50 CONTINUE
C          END IF
C
C      IF (NPRINT .EQ. 5) THEN
C          WRITE(6,100) N
C          DO 60 I=1,6
C              WRITE (6,200) (SE(I,J),J=1,6)
C      60 CONTINUE
C          END IF
C
C      100 FORMAT(' ','SE MATRIX FOR ELEMENT',I3)
C      200 FORMAT(' ','6(E13.7,3X))
C
C      RETURN
C      END
C
C-----* CBL2 (CURVALINEAR ELEMENT) *-----
C*****
C
C      CBL2 USES GAUSS QUADRATURE TO COMPUTE THE ELEMENT STIFFNESS
C      MATRIX, SE, FOR A CURVALINEAR ELEMENT. FIRST CALLING SHAPE
C      TO COMPUTE THE MATRICIES BL AND BNL, THEN SUMMING CONTRIBUTIONS
C      OF THE LINEAR AND NONLINEAR PARTS INTO SE.
C
C      SUBROUTINE CBL2( AREA,ELENG,EMOD,MINC,N,NGAUSS,NPRINT,PLACE,SE,
C      $          TAU,WGT,X)
C      $ IMPLICIT REAL*8(A-H,O-Z)
C      $ DIMENSION AREA(*),BL(9),BNL(3,9),ELENG(*),EMOD(*),MINC(3,*),
C      $          PLACE(3,*),SE(9,*),TAU(3,*),WGT(3,*),X(3,*)
C
C      IF (NPRINT.EQ.3) THEN
C          WRITE(6,100)
C      END IF
C
C      INITIALIZE ELEMENT STIFFNESS MATRIX, SE. ....
C      DO 8 K=1,9
C          DO 5 L=1,9
C              SE(K,L)=0.00

```

```

5 CONTINUE
8 CONTINUE
C
C START GAUSS QUADRATURE LOOP USING NGAUSS BY NGAUSS RULE. ....
DO 90 NA= 1,NGAUSS
PR= PLACE(NA,NGAUSS)
C
CALL SHAPE(BL,BNL,ELENG,MINC,N,PR,X)
C
DV1= WGT(NA,NGAUSS)*AREA(N)*EMOD(N)*ELENG(N)/2.000
DV2= WGT(NA,NGAUSS)*AREA(N)*TAU(NA,N)*ELENG(N)/2.000
C
COMPUTE THE LINEAR CONTRIBUTION TO SE. ....
DO 50 I=1,9
DO 40 J=1,9
SE(I,J)=SE(I,J)+BL(I)*BL(J)*DV1
40 CONTINUE
50 CONTINUE
C
COMPUTE THE NONLINEAR CONTRIBUTION TO SE. ....
DO 70 J=1,3
I=3*J
L=I-1
K=L-1
DO 60 M=1,9,3
SE(K,M)=SE(K,M)+BNL(1,K)*BNL(1,M)*DV2
SE(L,M+1)=SE(L,M+1)+BNL(2,L)*BNL(2,M+1)*DV2
SE(I,M+2)=SE(I,M+2)+BNL(3,I)*BNL(3,M+2)*DV2
60 CONTINUE
70 CONTINUE
90 CONTINUE
C
IF(NPRINT.EQ.5) THEN
WRITE(6,125) N
DO 102 I=1,9
WRITE(6,150) ( SE(I,J),J=1,9)
C 102 CONTINUE
END IF
C
100 FORMAT(/'CBL2 CALLED')
125 FORMAT(' ','SKT FOR ELEMENT',I3)
150 FORMAT(' ',9(D14.8,3X))
C
RETURN
END
C
C
C *****
C-----* ASSEMS *-----
C *****
C
ASSEMS ASSIGNS STIFFNESS COEFFICIENTS FROM THE ELEMENT STIFFNESS
MATRIX, SE, OF ELEMENT N TO THE SYSTEM STIFFNESS VECTOR, SKT,
USING MCODE AND MAXA.
C
SUBROUTINE ASSEMS(MAXA,MCODE,N,SE,SKT,M)
IMPLICIT REAL*8(A-H,O-Z)
DIMENSION SKT(*),MCODE(9,*),SE(9,*),MAXA(*)
C
IF (NPRINT.EQ.3) THEN
WRITE(6,100)
END IF
C
DO 20 JE=1,M
J=MCODE(JE,N)
IF (J.NE.0) THEN
DO 10 IE=1,JE
I=MCODE(IE,N)
IF (I.NE.0) THEN
K=MAXA(J)+J-I
SKT(K)=SKT(K)+SE(IE,JE)
END IF
10 CONTINUE
END IF
20 CONTINUE
C
100 FORMAT(/'ASSEMS CALLED')
C
RETURN
END
C
C *****
C-----* FORCES *-----
C *****

```

```

C      FORCES COMPUTES THE INTERNAL FORCE VECTOR, F, FOR THE CURRENT
C      DISPLACEMENT BY CALLING ELEM1 IF MTYP= 1 OR ELEM2 IF MTYP= 2.
C      IN ADDITION, THE EXTERNAL FORCE VECTOR, Q, IS UPDATED AT THE BY
C      CURRENT CONFIGURATION BY CALLING EITHER EXTRN1 OR EXTRN2.
C
C      SUBROUTINE FORCES(ACT,AREA,DD,DDE,ELENG,EMOD,F,FG,MAT,MCODE,MINC,
$      MN,MTYP,NAT,NE,NEQ,NGAUSS,NPRINT,PLACE,Q,QJ,TAU,WGT,X)
C      IMPLICIT REAL*8(A-H,O-Z)
C      DIMENSION ACT(*),AREA(*),DD(*),DDE(*),ELENG(*),EMOD(*),F(*),
$      FG(9,*),MAT(*),MCODE(9,*),MINC(3,*),MN(*),PLACE(3,*),
$      Q(*),QJ(*),TAU(3,*),WGT(3,*),X(3,*)
C
C      INITIALIZE INTERNAL FORCE VECTOR .....
C      DO 10 I=1,NEQ
C          F(I)=0.000
10 CONTINUE
C
C      IF (MTYP .EQ. 1) THEN
C          LINEAR ELEMENT .....
C          DO 20 I=1,NE
C              CALL ELEM1(AREA,DD,DDE,ELENG,EMOD,F,FG,I,MCODE,MINC,NPRINT,
$              TAU,X)
20 CONTINUE
C          IF (NAT .NE. 0) THEN
C              CALL EXTRN1(ACT,ELENG,MAT,MCODE,MINC,MN,NAT,NE,NEQ,NPRINT,
$              Q,QJ,X)
C          END IF
C
C      ELSE
C          CURVILINEAR ELEMENT .....
C          DO 30 I=1,NE
C              CALL ELEM2(AREA,DD,DDE,ELENG,EMOD,F,FG,I,MCODE,MINC,
$              NGAUSS,NPRINT,PLACE,TAU,WGT,X)
30 CONTINUE
C          IF (NAT .NE. 0) THEN
C              CALL EXTRN2(ACT,ELENG,MAT,MCODE,MINC,MN,NAT,NE,NEQ,NGAUSS,
$              NPRINT,PLACE,Q,QJ,WGT,X)
C          END IF
C
C      END IF
C
C      RETURN
C      END
C
C      *****
C      *-----* ELEM1 (LINEAR ELEMENT) *-----*
C      *****
C
C      ELEM1 COMPUTES THE CURRENT STRAIN, EPS, AND CAUCHY STRESS, TAU,
C      OF ELEMENT I. THEN THE INTERNAL ELEMENT FORCE ,PT, DUE TO THE
C      STRESS IS COMPUTED AND TRANSFORMED INTO THE GLOBAL INTERNAL
C      FORCE VECTOR, F.
C
C      SUBROUTINE ELEM1(AREA,DD,DDE,ELENG,EMOD,F,FG,I,MCODE,MINC,NPRINT,
$      TAU,X)
C      IMPLICIT REAL*8(A-H,O-Z)
C      DIMENSION AREA(*),DD(*),DDE(*),ELENG(*),EMOD(*),F(*),
$      FG(9,*),MCODE(9,*),MINC(3,*),TAU(3,*),X(3,*),XL(9),
$      YL(9),ZL(9)
C
C      FIND LOCAL INCREMENTAL DISPLACEMENTS .....
C      DO 10 J=1,6
C          L=MCODE(J,I)
C          IF (L .NE. 0) THEN
C              DDE(J)=DD(L)
C          ELSE
C              DDE(J)=0.000
C          END IF
10 CONTINUE
C
C      FIND LOCAL ELEMENT COORDINATES .....
C      DO 18 L=1,2
C          J=MINC(L,I)
C          XL(L)=X(1,J)
C          YL(L)=X(2,J)
C          ZL(L)=X(3,J)
18 CONTINUE
C
C      EL2=ELENG(I)*ELENG(I)
C      A=(XL(2)-XL(1))/EL2
C      B=(YL(2)-YL(1))/EL2

```

```

C      C=(ZL(2)-ZL(1))/EL2
C      COMPUTE INCREMENTAL STRAIN AND STRESS IN ELEMENT .....
EPS=A*(DDE(4)-DDE(1))+B*(DDE(5)-DDE(2))+C*(DDE(6)-DDE(3))
TAU(1,I)=TAU(1,I)+EMOD(I)*EPS
C      DETERMINE GLOBAL ELEMENT FORCES .....
PT=TAU(1,I)*AREA(I)
FG(4,I)=PT*A*ELENG(I)
FG(5,I)=PT*B*ELENG(I)
FG(6,I)=PT*C*ELENG(I)
FG(1,I)=-FG(4,I)
FG(2,I)=-FG(5,I)
FG(3,I)=-FG(6,I)
C      SUM ELEMENT FORCES INTO SYSTEM INTERNAL FORCE VECTOR .....
DO 20 J=1,6
  L=MCODE(J,I)
  IF (L.NE. 0) THEN
    F(L)=F(L)+FG(J,I)
  END IF
20 CONTINUE
C      RETURN
END
C      *****
C-----*      ELEM2 (CURVALINEAR ELEMENT)      *-----
C      *****
C      ELEM2 USES GAUSSIAN QUADRATURE TO COMPUTE THE CURRENT STRAIN,
C      EPS, AND CAUCHY STRESS, TAU, AT THE GAUSS POINTS. FROM THESE
C      VALUES, THE CURRENT INTERNAL FORCE, FG, IS COMPUTED AND MAPPED
C      INTO THE GLOBAL INTERNAL FORCE VECTOR, F.
C      SUBROUTINE ELEM2(AREA,DD,DDE,ELENG,EMOD,F,FG,I,MCODE,MINC,NGAUSS,
C      $      NPRINT,PLACE,TAU,WGT,X)
C      $      IMPLICIT REAL*8(A-H,O-Z)
C      $      DIMENSION AREA(*),BL(9),BNL(3,9),DD(*),DDE(*),ELENG(*),EMOD(*),
C      $      F(*),FG(9,*),MCODE(9,*),PLACE(3,*),TAU(3,*),WGT(3,*),
C      $      X(3,*),MINC(3,*)
C      IF (NPRINT.EQ.3) THEN
C        WRITE(6,100)
C      END IF
C      DO 5 L=1,9
C        FG(L,I)=0.000
C      5 CONTINUE
C      FIND LOCAL INCREMENTAL DISPLACEMENTS .....
DO 10 J=1,9
  L=MCODE(J,I)
  IF (L.NE. 0) THEN
    DDE(J)=DD(L)
  ELSE
    DDE(J)=0.000
  END IF
10 CONTINUE
C      START GAUSS QUADRATURE LOOP .....
DO 50 NA=1,NGAUSS
  EPS=0.000
  PR=PLACE(NA,NGAUSS)
  CALL SHAPE(BL,BNL,ELENG,MINC,I,PR,X)
  DV=WGT(NA,NGAUSS)*AREA(I)*ELENG(I)/2.000
  COMPUTE STRAIN AT GAUSS POINT .....
DO 20 J=1,9
  EPS=EPS+BL(J)*DDE(J)
20 CONTINUE
  COMPUTE STRESS AT GAUSS POINT .....
TAU(NA,I)=TAU(NA,I)+EMOD(I)*EPS
  SUM CONTRIBUTIONS INTO LOCAL FORCE VECTOR .....
DO 30 J=1,9
  FG(J,I)=FG(J,I)+BL(J)*TAU(NA,I)*DV
30 CONTINUE
50 CONTINUE
C      SUM ELEMENT FORCES INTO SYSTEM INTERNAL FORCE VECTOR .....
DO 60 J=1,9
  L=MCODE(J,I)
  IF (L.NE. 0) THEN

```



```

C      $          XL(9),YL(9),ZL(9)
C      IF (NPRINT.EQ.3) THEN
C          WRITE(6,200)
C          END IF
C      INITIALIZE EXTERNAL JOINT LOAD VECTOR .....
C      DO 10 I=1,NEQ
C          Q(I)=QJ(I)
C      10 CONTINUE
C      DO 100 I=1,NAT
C          EL=MN(I)
C          DO 20 J=1,9
C              FF(J)=0.000
C          20 CONTINUE
C      IF (MAT(I) .EQ. 1) THEN
C          DUM1=ACT(I)*ELENG(EL)
C          FF(2)=DUM1/6.000
C          FF(5)=DUM1*2.000/3.000
C          FF(8)=FF(2)
C      ELSE IF (MAT(I) .EQ. 2) THEN
C          DO 30 K=1,3
C              J=MINC(K,EL)
C              XL(K)=X(1,J)
C              ZL(K)=X(3,J)
C          30 CONTINUE
C          EL1A=XL(2)-XL(1)
C          EL1B=XL(3)-XL(2)
C          EL3A=ZL(2)-ZL(1)
C          EL3B=ZL(3)-ZL(2)
C          EL13A=DSQRT(EL1A*EL1A+EL3A*EL3A)
C          EL13B=DSQRT(EL1B*EL1B+EL3B*EL3B)
C          FF(2)=EL13A*ACT(I)/2.000
C          FF(8)=EL13B*ACT(I)/2.000
C          FF(5)=FF(2)+FF(8)
C      ELSE IF (MAT(I) .EQ. 3) THEN
C          DO 35 K=1,3
C              J=MINC(K,EL)
C              XL(K)=X(1,J)
C              YL(K)=X(2,J)
C              ZL(K)=X(3,J)
C          35 CONTINUE
C          DO 80 NA=1,NGAUSS
C              PR=PLACE(NA,NGAUSS)
C              A=PR-0.500
C              B=-2.000*PR
C              C=PR+0.500
C              ETA(1)=(PR*PR-PR)/2.000
C              ETA(2)=1.000-PR*PR
C              ETA(3)=(PR*PR+PR)/2.000
C              DJ11=A*XL(1)+B*XL(2)+C*XL(3)
C              DJ12=A*YL(1)+B*YL(2)+C*YL(3)
C              DO 40 J=1,3
C                  M=3*J
C                  L=M-1
C                  K=L-1
C                  FF(K)=FF(K)-ETA(J)*J12*ACT(I)*WGT(NA,NGAUSS)
C                  FF(L)=FF(L)+ETA(J)*J11*ACT(I)*WGT(NA,NGAUSS)
C                  FF(M)=FF(M)+1
C              40 CONTINUE
C          80 CONTINUE
C          END IF
C      DO 90 J=1,9
C          L=MCODE(J,EL)
C          Q(L)=Q(L)+FF(J)
C      90 CONTINUE
C      100 CONTINUE
C      200 FORMAT(/'EXTRN2 CALLED')
C      RETURN
C      END
C      *****
C      -----*          SOLVE          *-----
C      *****
C      IF THE STIFFNESS MATRIX HAS BEEN UPDATED (ITENUM=0) CALL FACTOR,
C      OBTAIN THE UNBALANCE FORCE, CALL REDUCE AND BACSUB TO SOLVE THE
C      SYSTEM OF LINEAR EQUATIONS BY GAUSS ELIMINATION.
C      SUBROUTINE SOLVE(DD,F,ITENUM,KSTBL,MAXA,NEGPIV,NEQ,NKT,NPRINT,QT,
C      $          SKT)
C      IMPLICIT REAL*8 (A-H,O-Z)

```



```

10 CONTINUE
C
C CHECK WITH TOLERANCES .....
IF ( TOTALD.NE.0 ) THEN
  C = ( DSQRT(DELTA D) ) / ( DSQRT(TOTALD) )
  IF ( C.GT.TOLDI1 ) THEN
    INCONV = INCONV + 10
  END IF
ELSE
  WRITE(6,100)
  STOP
END IF
C
100 FORMAT(' ERROR: DISPLACEMENTS ARE 0 -DISPL1-')
C
RETURN
END

```

```

C
C -----* ***** DISPL2 ******-----
C

```

```

C PERFORM THE DISPLACEMENT CONVERGENCE TEST USING THE INFINITY
C VECTOR NORM OF DISPLACEMENTS

```

```

C SUBROUTINE DISPL2 ( D,DD,INCONV,JCODE,NJ,TOLDI2 )
C IMPLICIT REAL*8 (A-H,O-Z)
C DIMENSION D(*),DD(*),JCODE(3,*)

```

```

C ROTMAX = 0.00
C ROTDEL = 0.00
C TRAMAX = 0.00
C TRADEL = 0.00

```

```

C INFINITE NORM OF DISPLACEMENTS .....

```

```

C DO 20 I=1,3
C   DO 10 J=1,NJ
C     K=JCODE(I,J)
C     IF ( K.NE.0 ) THEN
C       IF ( I.NE.3 ) THEN
C         TRAMAX = DMAX1( DABS( D(K) ),TRAMAX )
C         TRADEL = DMAX1( DABS( DD(K) ),TRADEL )
C       ELSE
C         ROTMAX = DMAX1( DABS( D(K) ),ROTMAX )
C         ROTDEL = DMAX1( DABS( DD(K) ),ROTDEL )
C       END IF
C     END IF

```

```

C   10 CONTINUE
C   20 CONTINUE

```

```

C CHECK WITH TOLERANCES .....
C IF ( ROTMAX.EQ.0.000.AND.TRAMAX.EQ.0.000 ) THEN
C   WRITE(6,100)
C   STOP
C END IF

```

```

C IF ( ROTMAX.NE.0.000 ) THEN
C   ROTAT = ROTDEL/ROTMAX
C   IF ( ROTAT.GT.TOLDI2 ) THEN
C     INCONV = INCONV + 1
C   END IF
C END IF

```

```

C IF ( TRAMAX.NE.0.00 ) THEN
C   TRASL = TRADEL/TRAMAX
C   IF ( TRASL.GT.TOLDI2 ) THEN
C     INCONV = INCONV + 1
C   END IF
C END IF

```

```

C 100 FORMAT(' ERROR: DISPLACEMENTS ARE 0 -DISPL2-')
C RETURN
C END

```

```

C
C -----* ***** UNBALF ******-----
C

```

```

C PERFORM A CONVERGENCE TEST FOR THE UNBALANCE FORCE.

```

```

C SUBROUTINE UNBALF ( F,FP,INCONV,NEQ,QT,TOLFOR )
C IMPLICIT REAL*8 (A-H,O-Z)
C DIMENSION F(*),FP(*),QT(*)

```

```

C UNBFI =0.000

```



```

        WRITE(6,600) J,(P(I,J),I=1,3)
60    CONTINUE
    END IF
C
100  FORMAT(' ',F13.7/' ',F14.8)
200  FORMAT (T10,'QI EQUILIB =',F17.9,' D EQUILIB =',F17.9)
300  FORMAT ('-',80(' '))/T10,'QI=',F15.7)
500  FORMAT (//T10,'GLOBAL JOINT DISPLACEMENTS'/T10,'JOINT',6X,
$      '1-DIRECTION',8X,'2-DIRECTION',8X,'3-DIRECTION')
600  FORMAT (T11,I3,3(2X,F17.9))
700  FORMAT(//T10,'STRESSES IN ELEMENTS: '/T10,'ELEMENT',T20,'STRESS')
800  FORMAT (T10,I3,T20,F13.4)
900  FORMAT (//T10,'JOINT FORCES: '/T10,'JOINT',7X,'1-DIRECTION',8X,
$      '2-DIRECTION',8X,'3-DIRECTION')
C
    RETURN
    END
C
C-----*          SHAPE          *-----
C-----*          *-----
C
C   SHAPE FORMULATES THE LINEAR AND NONLINEAR STRAIN DISPLACEMENT
C   MATRICIES AT THE GIVEN GAUSS POINTS, PR.
C
C   SUBROUTINE SHAPE(BL,BNL,ELENG,MINC,N,PR,X)
C   IMPLICIT REAL*8(A-H,O-Z)
C   DIMENSION BL(9),BNL(3,9),ELENG(*),MINC(3,*),X(3,*),XL(9),YL(9),
C   $          ZL(9)
C
C   IF (NPRINT.EQ.3) THEN
C     WRITE(6,100)
C   END IF
C
C   FIND NODAL COORDINATES OF ELEMENT N. ....
C   DO 10 L=1,3
C     J=MINC(L,N)
C     XL(L)=X(1,J)
C     YL(L)=X(2,J)
C     ZL(L)=X(3,J)
C 10 CONTINUE
C
C   DETJAC=ELENG(N)/2.000
C   A=(2.000*PR-1.000)/(2.000*DETJAC)
C   B=-2.000*PR/DETJAC
C   C=(2.000*PR+1.000)/(2.000*DETJAC)
C   TEMP1=A*XL(1)+B*XL(2)+C*XL(3)
C   TEMP2=A*YL(1)+B*YL(2)+C*YL(3)
C   TEMP3=A*ZL(1)+B*ZL(2)+C*ZL(3)
C
C   COMPUTE BL OF ELEMENT N. ....
C   BL(1)=A*TEMP1
C   BL(2)=A*TEMP2
C   BL(3)=A*TEMP3
C   BL(4)=B*TEMP1
C   BL(5)=B*TEMP2
C   BL(6)=B*TEMP3
C   BL(7)=C*TEMP1
C   BL(8)=C*TEMP2
C   BL(9)=C*TEMP3
C
C   COMPUTE BNL OF ELEMENT N. ....
C   BNL(1,1)=A
C   BNL(1,4)=B
C   BNL(1,7)=C
C   BNL(2,2)=A
C   BNL(2,5)=B
C   BNL(2,8)=C
C   BNL(3,3)=A
C   BNL(3,6)=B
C   BNL(3,9)=C
C
C 100 FORMAT('/SHAPE CALLED')
C
    RETURN
    END

```

**The vita has been removed from
the scanned document**