GEOMETRICALLY NONLINEAR FINITE ELEMENT ANALYSIS OF A GLULAM TIMBER DOME

by

Julio F. Davalos

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in
Civil Engineering

APPROVED:

S. M. Holzer, Chairman

J. R. Loferski, Co-chairman

D. A. Garst

T. Kuppusamy

S. Thangjitham

July, 1989

Blacksburg, Virginia

GEOMETRICALLY NONLINEAR FINITE ELEMENT ANALYSIS OF A GLULAM TIMBER DOME

by

Julio F. Davalos

S. M. Holzer, Chairman

J. R. Loferski, Co-chairman

Civil Engineering

(ABSTRACT)

A finite element modeling and geometrically nonlinear static analysis of glued-laminated timber domes is presented. The modeling and analysis guidelines include: the generation of the geometry, the selection of finite elements to model the components of a dome (beams, purlins, connections, and tension ring), the specification of boundary conditions, the specification of material properties, the determination of a sufficiently accurate mesh, the determination of design loads and the specification of load combinations, the application of analysis procedures to trace the complete response of the structure, and the evaluation of the response. The modeling assumptions and analysis procedures are applied to a dome model whose geometry is based on an existing glulam timber dome of 133 ft span and 18 ft rise above the tension ring. This dome consists of triangulated networks of curved southern pine glulam members connected by steel hubs. The members lie on great circles of a spherical surface of 133.3 ft radius. The dome is covered with a tongue-and-groove wood decking, which is not considered in this study. Therefore, the surface pressures are converted into member loads and then discretized into nodal concentrated loads.

A geometrically nonlinear, 3-d, 3-node, isoparametric beam element for glulam beams is formulated, and a program is developed for the analysis of rigid-jointed space frames that can trace the response of the structure by the modified Newton-Ralphson and the modified Risk-Wempner methods. The material is assumed to be continuous, homogeneous, and transversely isotropic. The material properties are assumed to be constant through the volume of the element. The transverse isotropy assumption is validated for southern pine by testing small samples in torsion. The accuracy of the modeling assumptions for southern pine glulam beams is experimentally verified by testing full-size, curved and straight, glulam beams under combined loads. The results show that the isobeam element can accurately represent the overall linear response of the beams. However, to analyze glulam domes with the program, connector elements to model the joints and a truss element to model the tension ring must be added. Therefore, the finite element program ABAQUS is used for the analysis of the dome model.

Three dead-load/snow-load combinations are considered in the analysis of the dome model. The space frame joints and the purlin-to-beam connections are modeled with 2-node isobeam elements. A 3-d, 2-node, truss element is used to model the tension ring. Three distinct analyses are considered for rigid and flexible joints: a linear analysis to check the design adequacy of the members. A linearized eigenvalue buckling prediction analysis to estimate the buckling load, which provided accurate estimates of the critical loads when rigid joints were specified. Finally, an incremental, iterative, geometrically nonlinear analysis to trace the complete response of the structure up to failure. It is shown that elastic instability, which is governed by geometric nonlinearities, is the dominant failure mode of the test dome. At the critical load, the induced element stresses remained below the proportional limit of the material. A discussion of the results is presented, and recommendations for future extensions are included.

Acknowledgements

I express my sincere appreciation to my advisor, Professor S. M. Holzer, who has been a wise leader, a great teacher, and a friend. It has been a pleasure working with Dr. J. R. Loferski, whom I thank for his enthusiastic participation in this project and for providing expert advice in wood mechanics. I am grateful to Professor Garst for his friendship and constant encouragement throughout my studies at Virginia Tech. My gratitude is extended to Professors T. Kuppusamy, W. S. Easterling, and S. Thangjitham for participating in the dissertation committee. I am indebted to for his generous cooperation in the installation and maintenance of the computer program ABAQUS. I also thank for the efficient and accurate typing of the manuscript.

I thank for conducting the experimental tests with patience and dedication. I thank for providing guidelines to use the program ABAQUS. The generous help and advice offered by during the program implementation of the 3-d beam element are greatly appreciated. I also thank for her help with the torsion experiments.

This work was partially supported by the U.S.D.A., Grant Administrative Management, Office of Grants and Program Systems, Agreement No. 86-FSTY-9-0178.

I am very grateful to my father and relatives for providing constant moral support. My heart goes to and , to whom I dedicate this dissertation and thank for their patience, understanding, and love.

Table of Contents

List of Illustrations

List of Tables

Chapter 1

INTRODUCTION

1.1 Purpose and Scope

Modern dome construction started with the development of skeletal iron domes, particularly the Schwedler domes, in the nineteenth century. The 207-ft span Schwedler dome, built in Vienna in 1874, is considered the pioneer structure in the development of reticulated domes (Makowski, 1984). Lattice, reticulated, or braced domes are three-dimensional networks of elements with their nodes contained on a surface of revolution, usually a spherical surface. The elements of lattice domes are subjected to three-dimensional actions (Holzer and Loferski, 1987). Lattice domes are very attractive columnless structures that can efficiently cover large areas and are widely used for assembly halls, stadiums, gymnasiums, indoor swimming pools, etc.

Most lattice domes were constructed of steel or aluminum. Quite recently, glued-laminated timber domes, such as the Triax and Varax grid domes, have entered the market and have shown to be competitive. For example, the Tacoma Dome (530 ft span; 110 ft rise above the tension ring), the largest timber dome and the largest clear-span timber roof structure in the world, was the winning design among three bidding contenders. The cost was 20% less than an air-supported fabric roof and 47% less than a cable-supported concrete roof (Pacific Builder and Engineer, 1983). Timber domes offer several advantages over large-span roof structures: lower construction cost, energy conservation, excellent acoustical control (approaching that of a concert hall), fire resistance (resulting in reduced insurance costs), and beauty. Due to these special properties, timber domes can serve as multipurpose facilities accommodating a wide variety of public events.

Although it appears that glulam domes have evolved into well-designed structures, their complete structural response has not been defined and their design is based on simplified methods of analysis. No methods have been established that can predict the ultimate load capacity and ensure the safety of glulam space frames and glulam lattice domes. In response to this need, a project was initiated at VPI&SU to formulate a finite element method of analysis that can predict the complete structural response of glulam space frames and lattice domes up to collapse (USDA proposal, Holzer and Loferski, 1986). The work reported herein is part of this research effort.

This study presents a finite element modeling and geometric nonlinear static analysis of glulam lattice domes. The objectives are:

1. To develop and experimentally verify a 3-d geometrically nonlinear finite element model for curved and straight glulam space beams.

2. To model the space frame joints.

3. To apply a numerical procedure that can trace the nonlinear prebuckling response of glulam lattice domes to collapse and along the path evaluate the element stresses against allowable values (NDS, 1986; Wood Handbook, 1987) and test for local failure modes (Holzer and Loferski, 1987).

4. To study the effects of variations in the joint stiffnesses.

A 3-d, 3-node, isoparametric beam finite element model for curved and straight glulam beams is proposed. The material is assumed homogeneous and transversely isotropic. The material properties are assumed constant throughout the volume of the element. The assumption of transverse isotropy is validated through torsion experiments of small southern pine samples. The modeling assumptions for glulam beams, which are incorporated in the beam element, are experimentally verified in the linear range by testing full-size southern pine glulam beams under combined loads.

The formulation of the nonlinear isoparametric beam finite element, and the program development are presented in detail. A discussion of the optimal computation of the tangent stiffness matrix and the element stresses is included. A well-structured computer program and a user's manual are included in this study. The program can trace geometrically nonlinear equilibrium paths by two methods: the modified Newton/Ralphson method with fixed or variable step incrementation and the modified Riks/Wempner method. Geometrically nonlinear analyses of several rigid-jointed space frames were performed with the computer program, and the results compared very well with available theoretical and experimental solutions. However, in order to use the program for the analysis of glulam lattice domes, connector elements to simulate the flexibility of the joints and a 3-d truss element to model the tension ring must be added.

The modeling assumptions and analysis procedures for glulam domes considered in this study are confined to the response evaluation of a dome model whose geometry is based on an existing Triax dome built in Raleigh, NC. The decking of the dome is not modeled, and therefore, the applied pressures are transformed into discrete applied nodal forces. Two-node connector beam elements are used to model the space frame joints and the purlin-to-beam connections. The model is analyzed for three dead load/snow load combinations using the commercial finite element program ABAQUS. Three distinct analyses are considered: a linear analysis to check the member design criteria specified by the codes, a linearized buckling prediction analysis to estimate the buckling load, and an incremental iterative geometric nonlinear analysis. A discussion of the results for rigid and flexible connections is presented.

1.2 Background

Timber beams are designed on the basis of linear elastic behavior (NDS, 1986; TCM, 1985). Constitutive models to predict the entire stress-strain curve for wood are limited (O'Halloran, 1973; Conners, 1989). However, material laws for shear and torsional responses remain to be defined. The distributions of bending stresses at failure for timber beams has been investigated and represented by linear, bi-linear, and parabolic functions (several references are given by Holzer and Loferski, 1987).

A number of researchers have investigated the structural response of glulam curved beams and glulam non-prirmatic members, including double tapered, pitched and curved, and single and double tapered beams. A comprehensive literature survey is presented by Dewey, 1984. However, there are no analytical models that can predict the complete 3-d response of straight or curved glulam beams up to failure.

Plane-stress continuum finite elements have been used to analyze glulam beams (Dewey, 1984; Kecheter and Gutkowski, 1984). However, modeling space frame members with continuum elements is impractical with today's computing technology. Since a space frame may consist of hundreds and even thousands of members, it is essential that each member can be modeled by just a few finite elements. Therefore, a one-dimensional, structural beam element that can model the dominant response characteristics of the space frame must be selected (Davalos et al., 1988). There exists a library of finite elements broad enough to satisfy the needs of most beam-type modeling

(Belytschko et al., 1977; Bathe and Bolourchi, 1979; Yang, 1984; Jau, 1985; ABAQUS). However, of the many beam elements available in the literature, the isoparametric beam elements (Bathe, 1982; Kani, 1986) are particularly well-suited to model curved and tapered beams. They can represent the nonlinear response of space beams with axial, biaxial bending, shear, and torsional deformations (including warping).

The dominant failure mode of single-layer metallic lattice domes is elastic instability (Soare, 1984; Tsuboi, 1984; Holzer and Loferski, 1987), which is governed by geometric nonlinearity. Details of possible failure modes and ultimate load capacity of glulam lattice domes is limited, and the most informative publication on design, analysis, and construction of Varax domes is a recent paper by Eshelby and Evans (1988). The most valuable background information for the development of a finite element method of analysis for glulam lattice domes is provided by Holzer and Loferski (1987).

Wind and snow loads are the most important design loads for lattice domes (Baker, 1984; Soare, 1984). Several reticulated domes have failed during snow storms (ENR; Holzer et al., 1984). For example, the collapse of the Bucharest dome was attributed to heavy local snow concentrations (Soare, 1984). Most codes of practice do not provide design wind and snow loads, and only a few international codes provide design wind pressures for domes. However, the information provided by these codes is not consistent (Baker, 1984); Baker recommends using the Wind Loading Handbook (1974) and the European Convention and Constructional Steelwork (1978).

There is considerable information on geometrically nonlinear analysis of structures. A comprehensive review and a fairly good treatment of the subject is offered by Bathe (1982). Two useful finite element methods of stability analysis are:

- A linearized buckling prediction analysis, which is useful in predicting the critical load, particularly for "stiff" structures (ABAQUS), and in estimating the length of the nonlinear equilibrium path that must be traced to locate the first critical point in a nonlinear analysis (Holzer and Huang, 1989).

- An incremental iterative nonlinear analysis by the modified Riks/Wempner method, which allows tracing equilibrium paths beyond limit points. Modifications to this method were introduced by Crisfield (1981), and a detailed presentation of the method is given by Holzer, et al. (1981).

Chapter 2

THE CONTINUUM MECHANICS EQUATIONS OF MOTION IN THE FINITE ELEMENT FORMULATION

2.1 Introduction

This chapter presents some fundamental concepts of continuum mechanics that are necessary for the finite element formulation of the nonlinear equilibrium equations. The deformation of a continuum is conveniently defined by a time variable t, and equilibrium is established by formulating the principle of virtual work at discrete time increments along the equilibrium path. The formulation of the virtual work equation makes use of the 2nd Piola-Kirchhoff stress tensor and the Green-Lagrange strain tensor, which are discussed in Section 2.2. The incremental finite element equilibrium equations can be formulated using either the total Lagrangian (T.L.) formulation or the updated Lagrangian (U.L.) formulation, as discussed in Section 2.3. In the U.L. formulation, the development of the finite element matrices requires less computational effort, and therefore, it is the choice of this study.

2.2 *Incremental Deformation of a Continuum*

In nonlinear analysis, we consider a body subjected to arbitrary large motions. The problem of tracing the motion of a body in a stationary Cartesian coordinate system is conveniently described by a time variable t (Bathe, 1982, p. 316). Of particular interest is establishing the equilibrium of the body at discrete times 0, t, and t + Δt. The procedure consists in assuming that equilibrium for all time steps from time 0 to time t, inclusive, has been established. Then, t + Δt is a typical current configuration for which equilibrium is required. In this manner, the complete equilibrium path is traced by referring all quantities to a known configuration as the body moves from one configuration to the next. This description of motion, where each particle ("infinitessimal volumetric element") is defined in terms of a reference position and time (e.g., time t), is called the *Langrangian* or *material description*, and it is commonly used in the study of solid mechanics. In contrast, the *Eulerian description* expresses the motion of a continuous medium in terms of an instantaneous position and time (e.g., time t + Δt), and it is more useful in the study of fluid mechanics (Frederick and Chang, 1972, pp. 67-69). The motion of the body shown in Figure 2.1 illustrates the coordinates of a point P within the body at times 0, t, and t + Δt, where the left superscript refers to the configuration being considered. Using this notation, the incremental displacements between configurations are defined as

$$u_i = {}^{t+\Delta t}u_i - {}^{t}u_i, \qquad i = 1,2,3 \tag{2.1}$$

In addition, based on Fig. 2.1, the following relations can be written:

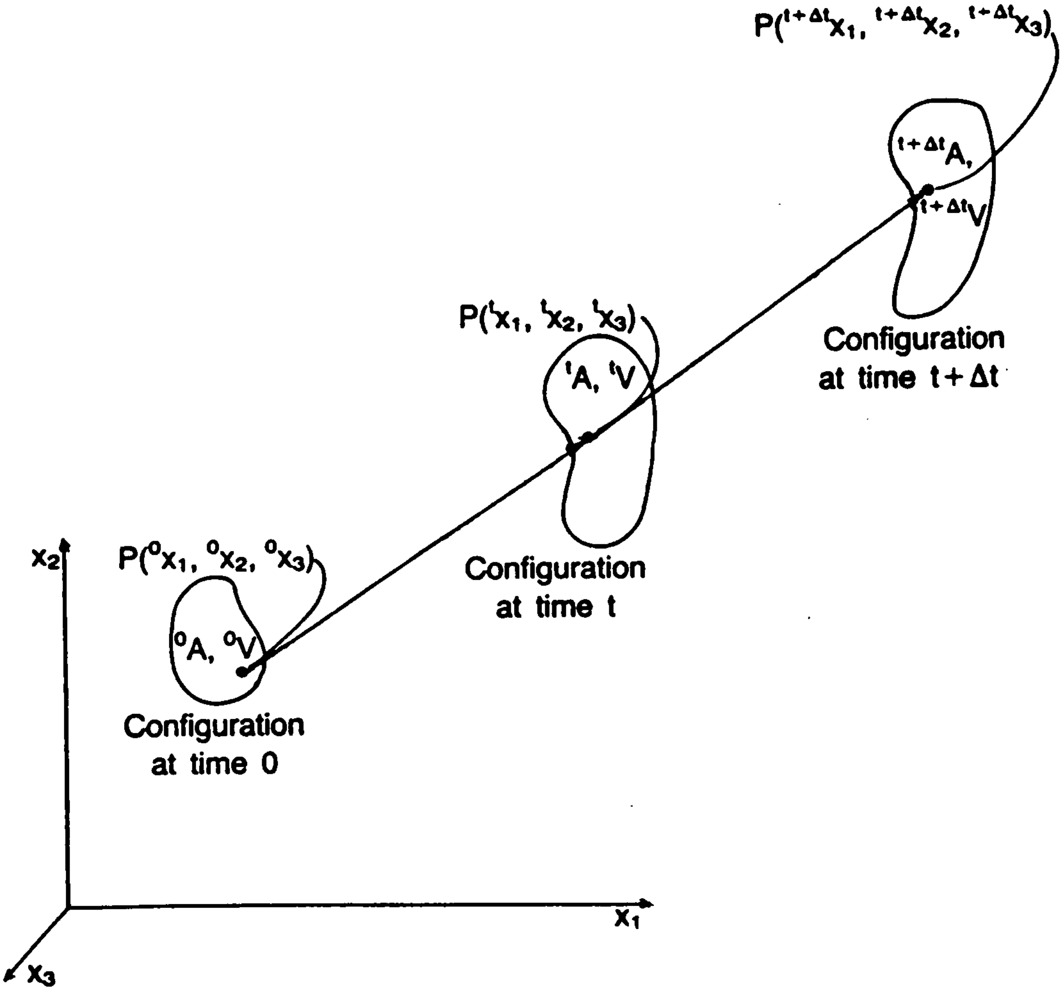


Figure 2.1. Body subjected to arbitrary large motions (Bathe, 1986).

$$\begin{aligned} {}^{t}x_i &= {}^{o}x_i + {}^{t}u_i \\ {}^{t+\Delta t}x_i &= {}^{o}x_i + {}^{t+\Delta t}u_i \\ {}^{t+\Delta t}x_i &= {}^{t}x_i + u_i \end{aligned} \tag{2.2}$$

Principle of virtual work

The virtual work principle applied to the equilibrium of the body at time t + Δt is expressed as

$$\int_{{}^{t+\Delta t}V} {}^{t+\Delta t}\tau_{ij}\ \delta_{t+\Delta t}e_{ij}\ {}^{t+\Delta t}dV = {}^{t+\Delta t}R \tag{2.3}$$

where, ${}^{t+\Delta t}\tau_{ij}$ is the Cauchy stress tensor (internal forces per unit area at time $t + \Delta t$) and ${}_{t+\Delta t}e_{ij}$ is the infinitessimal strain tensor (also called "Cauchy's infinitessimal strain tensor" (Fung, 1969, p. 98) and "Lagrangian lincar strain tensor" (Frederick and Chang, 1972, p. 82)) referred to the configuration at time $t + \Delta t$. The left-hand side of Eq. (2.3) represents the internal virtual work of the body subjected to virtual displacements at time $t + \Delta t$. The term ${}^{t+\Delta t}R$ is the external virtual work of the applied loads. To establish the equilibrium of the body at time $t + \Delta t$, we cannot directly apply Eq. (2.3) for the following reasons (Bathe, 1982, p. 317):

1. We cannot integrate over an unknown volume ${}^{t+\Delta t}V$.

2. The Cauchy stresses at time $t + \Delta t$ cannot be computed by simply adding the incremental stresses to the Cauchy stresses at time t, because the Cauchy stress tensor is not invariant under rigid body motions.

3. The infinitessimal strains cannot be referred to an unknown configuration $t + \Delta t$.

To overcome these difficulties, Eq. (2.3) can be written in terms of the 2nd Piola-Kirchhoff stress tensor and the Green-Lagrange strain tensor.

2nd Piola-Kirchhoff stress tensor

The 2nd Piola-Kirchhoff stress tensor at time t referred to the initial configuration at time o is defined as (Bathe, 1982, p. 319)

$$ {}^t_oS_{ij} = \frac{{}^o\rho}{{}^t\rho}\; {}^o_tx_{i,m}\; {}^t\tau_{mn}\; {}^o_tx_{j,n} \quad \text{– indicial notation} $$

$$ {}^t_o\underline{S} = \frac{{}^o\rho}{{}^t\rho}\; {}^o_t\underline{X}\; {}^t\underline{\tau}\; {}^o_t\underline{X}^T \quad \text{– matrix notation} \qquad (2.4) $$

Solving for the Cauchy stress tensor gives

$$ {}^t\tau_{ij} = \frac{{}^t\rho}{{}^o\rho}\; {}^t_ox_{i,m}\; {}^t_oS_{mn}\, {}^t_ox_{j,n} \quad \text{– indicial notation} $$

$$ {}^t\underline{\tau} = \frac{{}^t\rho}{{}^t\rho}\; {}^t_o\underline{X}\; {}^t_o\underline{S}\; {}^t_o\underline{X}^T \quad \text{– matrix notation} \qquad (2.5) $$

where,

$$ {}^o_tx_{i,m} = \frac{\partial\, {}^ox_i}{\partial\, {}^tx_m} $$

${}^o\rho, {}^t\rho$ = mass densities at times 0 and t

${}^t_ox_{m,i}$ = element (m,i) of the deformation gradient ${}^t_o\underline{X}$.

In a Cartesian coordinate system, the deformation gradient ${}^t_o\underline{X}$ (denoted as $\underline{F}$ by Malvern, 1969, p. 156) is defined as

$$ {}_{o}^{t}X = \begin{bmatrix} \frac{\partial\, {}^{t}x_1}{\partial\, {}^{o}x_1} & \frac{\partial\, {}^{t}x_1}{\partial\, {}^{o}x_2} & \frac{\partial\, {}^{t}x_1}{\partial\, {}^{o}x_3} \\ \frac{\partial\, {}^{t}x_2}{\partial\, {}^{o}x_1} & \frac{\partial\, {}^{t}x_2}{\partial\, {}^{o}x_2} & \frac{\partial\, {}^{t}x_2}{\partial\, {}^{o}x_3} \\ \frac{\partial\, {}^{t}x_3}{\partial\, {}^{o}x_1} & \frac{\partial\, {}^{t}x_3}{\partial\, {}^{0}x_2} & \frac{\partial\, {}^{t}x_3}{\partial^{o}\, x_3} \end{bmatrix} $$

or using indicial notation

$$ {}_{o}^{t}X_{ij} = \frac{\partial\, {}^{t}x_i}{\partial\, {}^{o}x_j} = {}_{o}^{t}x_{i,j} \tag{2.6} $$

Consistent with the range and summation conventions of the indicial notation, expressions (2.4) represent nine equations, each consisting of nine terms. The ratio of the mass densities $\frac{{}^{o}\rho}{{}^{t}\rho}$ can be evaluated from the *Langrangian continuity equation* derived from the *principle of conservation of mass* (Frederick and Chang, 1972, pp. 73-76; Bathe, 1982, p. 319):

$$ {}^{o}\rho = {}^{t}\rho \det {}_{o}^{t}X \tag{2.7} $$

The 2nd Piola-Kirchhoff stresses, also called "pseudostresses" (Malvern, 1969, p. 222), are now completely defined, and their physical significance is illustrated by Bathe (1982, p. 223) using geometric arguments based on the Cauchy stresses.

The characteristics of the 2nd Piola-Kirchhoff stress tensor are:

1. It gives expressions for the stresses in the current configuration referred to a previously defined configuration (e.g., the initial configuration).

2. It is a symmetric tensor by virtue of the symmetry of the Cauchy stress tensor.

3. It is invariant under rigid body motions. That is, we can incrementally decompose ${}^{t+\Delta t}_{o}S_{ij}$ as follows:

$$ {}^{t+\Delta t}_{o}S_{ij} = {}^{t}_{o}S_{ij} + {}_{o}S_{ij} $$

Hence, ${}^{t+\Delta t}_{o}\underline{S}$ changes only when the material is deformed.

4. It has no direct physical interpretation.

In order to use the same reference for all tensors appearing in the stress-strain equation as well as the principle of virtual work, the Green-Lagrange strain tensor is used with the 2nd Piola-Kirchhoff stress tensor.

Green-Lagrange strain tensor

The Green-Lagrange strain tensor was introduced by Green (1841) and Saint-Venant (1844), and its derivation is presented, for example, by Fung (1969, pp. 97-98). This tensor measures the stretching deformations and is usually expressed as

$$ {}^{t}_{o}\varepsilon_{ij} = \frac{1}{2}\left({}^{t}_{o}u_{i,j} + {}^{t}_{o}u_{j,i} + {}^{t}_{o}u_{k,i}\;{}^{t}_{o}u_{k,j}\right) \qquad (2.7) $$

where,

$$ {}^{t}_{o}u_{i,j} = \frac{\partial\,{}^{t}u_{i}}{\partial\,{}^{o}x_{j}} $$

$\frac{1}{2}\left({}^{t}_{o}u_{i,j} + {}^{t}_{o}u_{j,i}\right) = {}^{t}_{o}e_{ij}$ – infinitessimal strain tensor (linear in displacements)

$\frac{1}{2}\left({}^{t}_{o}u_{k,i}\;{}^{t}_{o}u_{k,j}\right) = {}^{t}_{o}\eta_{ij}$ – nonlinear strain tensor (quadratic in displacements)

This strain tensor is exact and holds for any amount of stretching. Its important characteristics are:

1. It can be used with the 2nd Piola-Kirchhoff stress tensor.

2. It is symmetric.

3. It is invariant under rigid body motions, which permits us to incrementally decompose ${}^{t+\Delta t}_{o}\varepsilon_{ij}$:

   $${}^{t+\Delta t}_{\ \ \ o}\varepsilon_{ij} = {}^{t}_{o}\varepsilon_{ij} + {}_{o}\varepsilon_{ij}$$

   Therefore, for a rigid body motion between times t and $t + \Delta t$, ${}^{t+\Delta t}\underline{\varepsilon}_{o} = {}^{t}_{o}\underline{\varepsilon}_{o}$. Also, for a rigid body motion between time 0 and t, ${}^{t}_{o}\underline{\varepsilon} = \underline{0}$.

A necessary requirement of the nonlinear incremental finite element formulation is the use of strain tensors that are energetically conjugate to the corresponding stress tensors. Thus, the following two expressions can be used (Bathe, 1982, p. 327):

${}^{t}\tau_{ij}\ \delta_{t}e_{ij}$ = virtual work at time t per unit <u>current volume</u>

${}^{t}_{o}s_{ij}\ \delta^{t}_{o}\varepsilon_{ij}$ = virtual work at time t per unit of <u>original volume</u>

(2.8)

The 2nd Piola-Kirchhoff stress tensor is energetically conjugate to the Green-Lagrange strian tensor (see Bathe, 1982, p. 334), and therefore, the principle of virtual work can be written as

$$\int_{{}^{t}V} {}^{t}\tau_{ij}\ \delta_{t}e_{ij}\ {}^{t}dV = \int_{{}^{o}V} {}^{t}_{o}s_{ij}\ \delta^{t}_{o}\varepsilon_{ij}\ {}^{o}dV \qquad (2.9)$$

This relation holds for all times Δt, 2Δt,..., t , $t + \Delta t$,..., and it is fundamental in the development of the incremental Lagrangrian equations of motion.

2.3 Incremental Formulation

The continuum mechanics equations of motion are expressed incrementally and used in the nonlinear finite element analysis of the continuum. The solution is obtained by linearizing the virtual work equation at each equilibrium point along the path of the load-displacement curve and then applying an incremental iterative procedure that uses as a reference a previously defined configuration. The nonlinear equations of equilibrium can be effectively formulated using the total Lagrangian (T.L.) or the updated Lagrangian (U.L.) formulations.

Total Lagrangian formulation

The T.L. formulation traces the deformation of the continuum by referring all static and kinematic variables to the original configuration at time t = 0. The linearization of the equilibrium equation is more involved than in the U.L. formulation because the strain increments contain an initial displacement effect which results in a more complex strain-displacement matrix than in the U.L. formulation. Also, the final linearized incremental equilibrium equation requires working with the 2nd Piola-Kirchhoff stresses which have little physical significance. In this study, the U.L. formulation is preferred, and it is described in detail next.

Updated Lagrangian formulation

In the solution procedure of the U.L. formulation, the configuration at time t is used as the reference state. Thus, the principle of virtual work (Eq. 2.9) can be written as

$$\int_{{}^tV} {}^{t+\Delta t}_{\ \ \ \ t}S_{ij}\ \delta\, {}^{t+\Delta t}_{\ \ \ \ t}\varepsilon_{ij}\ {}^t dV = {}^{t+\Delta t}R \tag{2.10}$$

The 2nd Piola-Kirchhoff and the Green-Lagrange tensors are decomposed as

$$\begin{aligned} {}^{t+\Delta t}_{\ \ \ \ t}S_{ij} &= \underbrace{{}^t_tS_{ij}}_{\text{known}} + \underbrace{{}_tS_{ij}}_{\text{unknown increments}} = {}^t\tau_{ij} + {}_tS_{ij} \\ {}^{t+\Delta t}_{\ \ \ \ t}\varepsilon_{ij} &= \underbrace{\overbrace{{}^t_t\varepsilon_{ij}}}_{0} + \overbrace{{}_t\varepsilon_{ij}} = {}_t\varepsilon_{ij} \end{aligned} \tag{2.11}$$

According to Eq. (2.7), the incremental Green-Lagrange strain tensor can be expressed as

$${}_t\varepsilon_{ij} = {}_te_{ij} + {}_t\eta_{ij} \tag{2.12}$$

where,

$$\begin{aligned} {}_te_{ij} &= \frac{1}{2}\left({}_tu_{i,j} + {}_tu_{j,i}\right) \quad \text{– linear strain increment} \\ {}_t\eta_{ij} &= \frac{1}{2}\,{}_tu_{k,i}\ {}_tu_{k,j} \quad \text{– nonlinear strain increment} \end{aligned} \tag{2.13}$$

and,

$$\delta\,{}_t\varepsilon_{ij} = \delta\,{}_te_{ij} + \delta\,{}_t\eta_{ij} \tag{2.14}$$

Substituting Eqs. (2.11) into Eq. (2.10), we get

$$\int_{{}^tV} {}_tS_{ij}\,\delta\,{}_t\varepsilon_{ij}\ {}^tdV + \int_{{}^tV} {}^t\tau_{ij}\,\delta\,{}_t\eta_{ij}\ {}^tdV = {}^{t+\Delta t}R - \int_{{}^tV} {}^t\tau_{ij}\,\delta\,{}_te_{ij}\,{}^tdV \tag{2.15}$$

Equation (2.15) is exact. The right-hand side of the equation is known, whereas the left-hand side contains unknown displacement increments.

The next step is to linearize Eq. (2.15):

- $\int_{{}^tV} {}^t\tau_{ij}\,\delta_t\eta_{ij}\,{}^tdV$ is linear in u_i. That is, $\delta_t\eta_{ij}$ is a linear function of the displacement increments u_i.

- $\int_{{}^tV} {}_tS_{ij}\,\delta_t\varepsilon_{ij}\,{}^tdV$ contains higher terms in u_i. In general, ${}_tS_{ij}$ is a nonlinear function of ${}_t\varepsilon_{ij}$, and we need to neglect all higher-order terms in u_i. We begin the linearization process by writing ${}_tS_{ij}$ as a Taylor series in ${}_t\varepsilon_{ij}$ (Bathe, 1986, p. 5-8):

$$ {}_tS_{ij} = \frac{\partial_t S_{ij}}{\partial_t\varepsilon_{rs}}\Big|_t \; ({}_t\varepsilon_{rs}) + \text{higher-order terms} $$

where, $\dfrac{\partial_t S_{ij}}{\partial_t\varepsilon_{rs}}\Big|_t$ is the tangent constitutive relation of the material, ${}_tC_{ijrs}$, and ${}_t\varepsilon_{rs}$ contains linear (${}_te_{rs}$) and nonlinear (${}_t\eta_{rs}$) terms in u_i. Retaining only the linear terms, ${}_tS_{ij}$ is approximated as

$$ {}_tS_{ij} \doteq {}_tC_{ijrs}\,{}_te_{rs} \; . $$

Hence, we obtain

$$ \underbrace{{}_tS_{ij}}\,\delta\,{}_t\varepsilon_{ij} \doteq \underbrace{{}_tC_{ijrs}\,{}_te_{rs}}(\delta_te_{ij} + \delta_t\eta_{ij}) $$

or

does not contain u_i linear in u_i

$$ {}_tS_{ij}\,\delta_t\varepsilon_{ij} \doteq \underbrace{{}_tC_{ijrs}\,{}_te_{rs}\,\overbrace{\delta_te_{ij}}}_{\text{linear in } u_i} + \underbrace{{}_tC_{ijrs}\,{}_te_{rs}\,\overbrace{\delta_t\eta_{ij}}}_{\text{quadratic in } u_i} $$

The linearized result is:

$$ {}_tS_{ij}\,\delta_t\varepsilon_{ij} \doteq {}_tC_{ijrs}\,{}_te_{rs}\,\delta_te_{ij} \qquad (2.16) $$

Substituting Eq. (2.16) into Eq. (2.15), the final incremental linearized equation of equilibrium is

$$\int_{{}^{t}V} {}_{t}C_{ijrs}\,{}_{t}e_{rs}\,\delta_{t}e_{ij}\,{}^{t}dV + \int_{{}^{t}V} {}^{t}\tau_{ij}\,\delta_{t}\eta_{ij}\,{}^{t}dV = {}^{t+\Delta t}R - \int_{{}^{t}V} {}^{t}\tau_{ij}\,\delta_{t}e_{ij}\,{}^{t}dV \tag{2.17}$$

In this study, this governing linearized equilibrium equation is used in the finite element formulation of a nonlinear 3-d beam element presented in Chapter 4.

Chapter 3

MODELING OF GLULAM BEAMS: IMPORTANT CONSIDERATIONS

3.1 Introduction

Timber is a renewable and practically inexhaustible resource. Making structures from wood requires less energy than using any other structural material (steel requires five times the energy needed for wood (Natterer, 1987)). The economical and efficient application of timber to heavy engineered structures is achieved through the use of glued-laminated members (glulam). Glulam structural members are fabricated from smaller pieces of wood glued together, either in straight or curved forms. Some significant advantages of glulam are: higher utilization of wood, production of large members that are not limited by the length of the tree, freedom of form, and better quality control of structural members. Glulam beams are suitable for the construction of large timber space frames and lattice domes. However, to make glulam a structural material of choice and to expand its potential use in structural engineering, mathematical models of glulam beams that can be efficiently used in numerical methods of analysis, such as the finite element method, are needed. Such models must be practical, computationally feasible, and sufficiently accurate.

In this chapter, the general complex cellular structure of wood is examined (Section 3.2), and the accepted orthotropic representation of wood is discussed in some detail (Section 3.3). The transverse shear moduli of timber beams are approximately 1/14 to 1/16 of the longitudinal elastic moduli. Thus, flexural shear deformations must be incorporated in mathematical models of timber beams (Section 3.4). Fabrication standards for glulam beams specify that the longitudinal fibers (grain) of the material be approximately parallel to the axis of the beam. However, the laminae of the cross section of a beam usually exhibit random orientation of the growth rings. Therefore, the orthotropic modeling of glulam beams is too complex, and a practical simplifying assumption is to consider the beam as isotropic in the plane of the cross section (transverse isotropy). This assumption is studied and validated through torsion experiments of small southern pine samples (Section 3.6). Based on Timoshenko's beam theory, the constitutive matrix of a transversely isotropic glulam beam can be written in terms of the longitudinal elastic modulus and the transverse shear modulus of the material. The shear modulus can be computed from torsion tests, as described in Section 3.6, and the longitudinal elastic modulus can be measured at increments along the length of the laminae from tension tests (Section 3.7). To incorporate the variability of elastic parameters within a beam, continuum finite elements can be used. However, in the analysis of large space frames and lattice domes, one-dimensional structural finite elements based on engineering theory must be used (Section 3.7). Finally, a summary of reasonably acceptable simplifying assumptions for the finite element modeling of glulam beams is presented in Section 3.8. These assumptions are used in the formulation of a 3-d isoparametric beam finite element described in Chapter 4.

3.2 Material Characterization

To develop mathematical models for wood, it is necessary to understand its basic cellular structure. A comprehensive description of the mechanical characteristics of wood is given by Schniewind (1982), from which part of the material in this section has been extracted.

Wood is a renewable natural product of biological origin. Wood cells are arranged as bundles of slender, hollow cylindrical tubes (similar to soda straws) aligned parallel to the axis of the tree. The differences in the cellular structure of these bands or groups of cells lead to differences in properties which make wood a very variable, anisotropic, and heterogeneous material. The shell-like tubular cells can resist larger longitudinal tensile forces than compressive forces. Under compression, these slender tubes can buckle and eventually collapse by folding or crumpling of the wood cells, and therefore, the compressive strength parallel to the longitudinal fibers (parallel to grain) is about half of the tensile strength. Also, due to the longitudinal orientation of fibers, the transverse (perpendicular to grain) strength of wood is approximately 1/20 of the tensile strength parallel to grain. Thus, tension perpendicular to grain is an important consideration in the design of curved beams, tapered beams, and connections of wood members.

The wood cells are organized in concentric circular bands which are commonly known as the growth rings of the cross section of a tree. The cylindrical symmetry of rings of cells is the basis for modeling wood as an orthotropic composite layered system, which is discussed next.

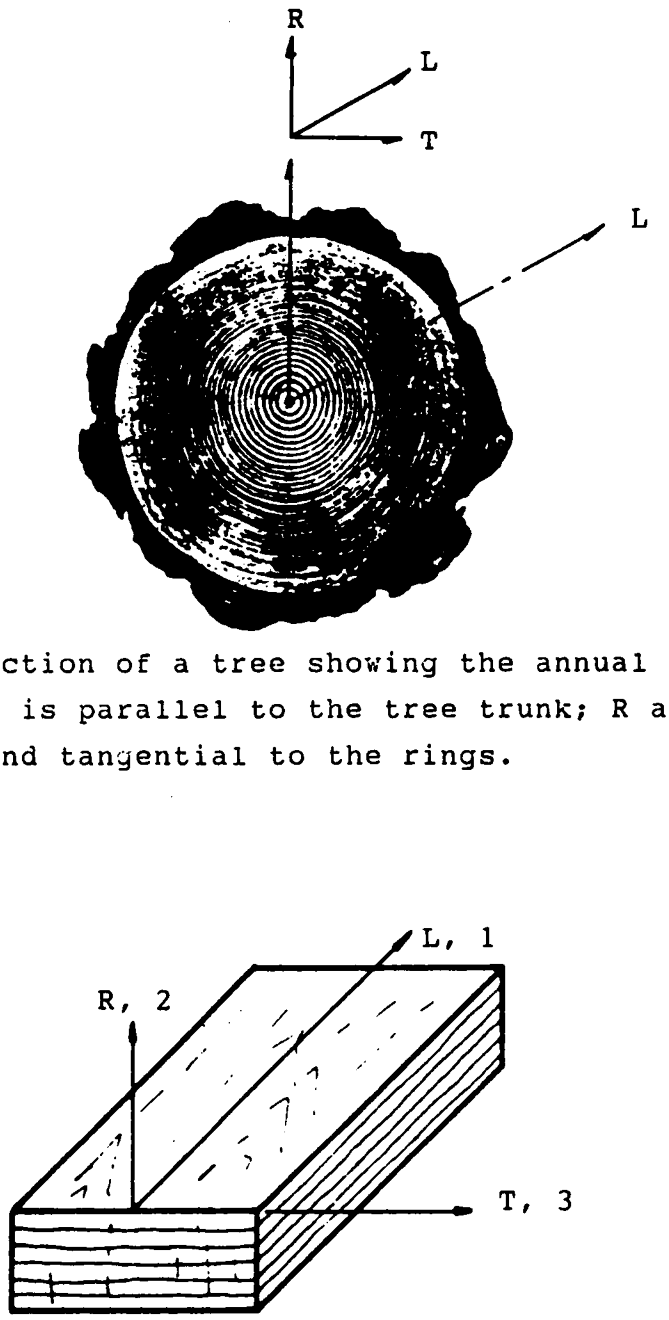
3.3 Structural Model of Wood

Figure 3.1 shows the annual growth rings of the cross section of a tree. Each ring consists of a light earlywood or springwood zone and a dark latewood or summerwood zone. In the earlywood band, the cross section of the cells is larger and the cell walls are thinner than in the latewood band. Therefore, the density, strength, and stiffness of the latewood zone are greater than those of the earlywood zone. This natural arrangement of concentric rings is commonly represented as a composite material of layered earlywood-latewood laminae in cylindrical coordinates L, R, and T (Figure 3.1a). Where, L is the longitudinal axis of the tree, R is the radial axis normal to the growth rings, and T is the tangential axis to the growth rings. In the solution of most problems, rectangular orthotropy (rhombic orthotropy) is assumed rather than cylindrical orthotropy. The justification is that pieces cut from the trunk of a tree are sufficiently small to ignore the curvature of the growth rings and to model the material as a planar layered system (Figure 3.1b).

If a wood member is assumed to be orthotropic with its material (L, R, T) and geometric (1, 2, 3) axes coinciding (Fig. 3.1b), the stresses, σ_{ij}, and strains, ε_{ij} , are related by nine independent constants, c_{ij}, and the constitutive equation can be expressed as (Jones 1975):

$$\begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{23} \\ \sigma_{31} \\ \sigma_{12} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} & 0 & 0 & 0 \\ & c_{22} & c_{23} & 0 & 0 & 0 \\ & & c_{33} & 0 & 0 & 0 \\ & & & c_{44} & 0 & 0 \\ & \textit{symmetric} & & & c_{55} & 0 \\ & & & & & c_{66} \end{bmatrix} \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ 2\varepsilon_{23} \\ 2\varepsilon_{31} \\ 2\varepsilon_{12} \end{bmatrix} \qquad (3.1)$$

where the constants c_{ij} can be expressed in terms of the Young's moduli E_L, E_R, E_T; the shear moduli G_{LR}, G_{LT}, and G_{RT}, and the Poisson's ratios ν_{LR}, ν_{LT},, and ν_{RT}.



(a) Cross section of a tree showing the annual growth rings. L is parallel to the tree trunk; R and T are normal and tangential to the rings.

(b) Orthotropic model representing lumber.

Figure 3.1. Orthotropic representation of wood.

The assumption of rhombic symmetry works reasonably well for wood when the material and geometric axes coincide. When these do not coincide, tensor transformations of axes are necessary, and an experimental study conducted by Goodman and Bodig (1970) has shown that for a general orientation of material axes, such transformations do not hold at all. However, when one of the geometric and material axes coincide, transformations in the normal plane to such axis (L-R; L-T, or T-R) are quite satisfactory. However, in most engineering applications such tensor transformations are not practical, particularly in glulam beams where each lamina usually shows an arbitrary orientation of the growth rings. It is then necessary to explore other simplifications in the modeling of solid-sawn and glulam beams; a detailed investigation of the structural modeling of glulam is presented in Sections 3.5 and 3.6.

Since the objective of this chapter is to establish reasonably sound, sufficiently accurate, and practical assumptions for the finite element modeling of timber beams, it is worth noting the ratio of the possible three elastic constants needed in orthotropic beam theory: E_L, G_{LR}, and G_{LT}. Typical ratios of these constants for structural lumber (e.g., southern pine) are (Bodig and Jane 1982):

$$G_{LR} : G_{\mathrm{LT}} \doteq 1.1{:}1 \tag{3.2}$$

$$E_L : G_{LR} \text{ or } G_{\mathrm{LT}} \doteq 14{:}1 \tag{3.3}$$

The difference between G_{LR} and G_{LT} is usually small (Eq. 3.2) and can probably be ignored in the modeling of timber beams (see Section 3.6). Since the shear modulus is approximately 1/14 of the longitudinal modulus (Eq. 3.3), the component of shear deflection in timber beams can be significant, and it is briefly illustrated in the following section.

3.4 Shear Deflection of Timber Beams

Classical beam theory excludes shear deformation and assumes that a normal to the midsurface remains normal during deformations. Timoshenko's beam theory (Mindlin-Reissner plate theory) considers shear deformations and assumes that a plane section originally normal to the midsurface remains plane but because of shear deformations, this section does not necessarily remain normal to the midsurface. A detailed discussion of shear deformation was previously presented by the author (Davalos, 1987).

The consideration of shear deformations results in additional deflection of a beam. The shear deflection of timber beams has received considerable attention by a number of researchers (Newlin and Trayer, 1924; Biblis, 1965), and it is also discussed in some textbooks of wood mechanics (Gurfinkel, 1973; Bodig and Jane, 1982). For a span-to-depth ratio greater than 3, the total deflection of a beam, δ_t, can be expressed as the sum of the contributions due to the bending strains, δ_b, and the shear strains, δ_s. Thus,

$$\delta_t = \delta_b + \delta_s \tag{3.4}$$

where the shear component of deflection, δ_s, can be computed from the shear strain energy (Timoshenko, 1955), and it is a function of the boundary conditions, loading, and the geometry of the beam. A collection of formulas for various boundary and load conditions is available in the literature (Roark and Young, 1982; AITC, 1985). In this section, the effect of shear on the midspan deflection of the two simply supported beams, illustrated in Fig. 3.2, is discussed. The coefficient κ of the second term of the formulas in Fig. 3.2 is the shear correction factor which accounts for the Timoshenko beam theory assumption of constant shear stress distribution (Langhaar, 1962). For a rectangular cross section, $\kappa = \frac{6}{5}$.

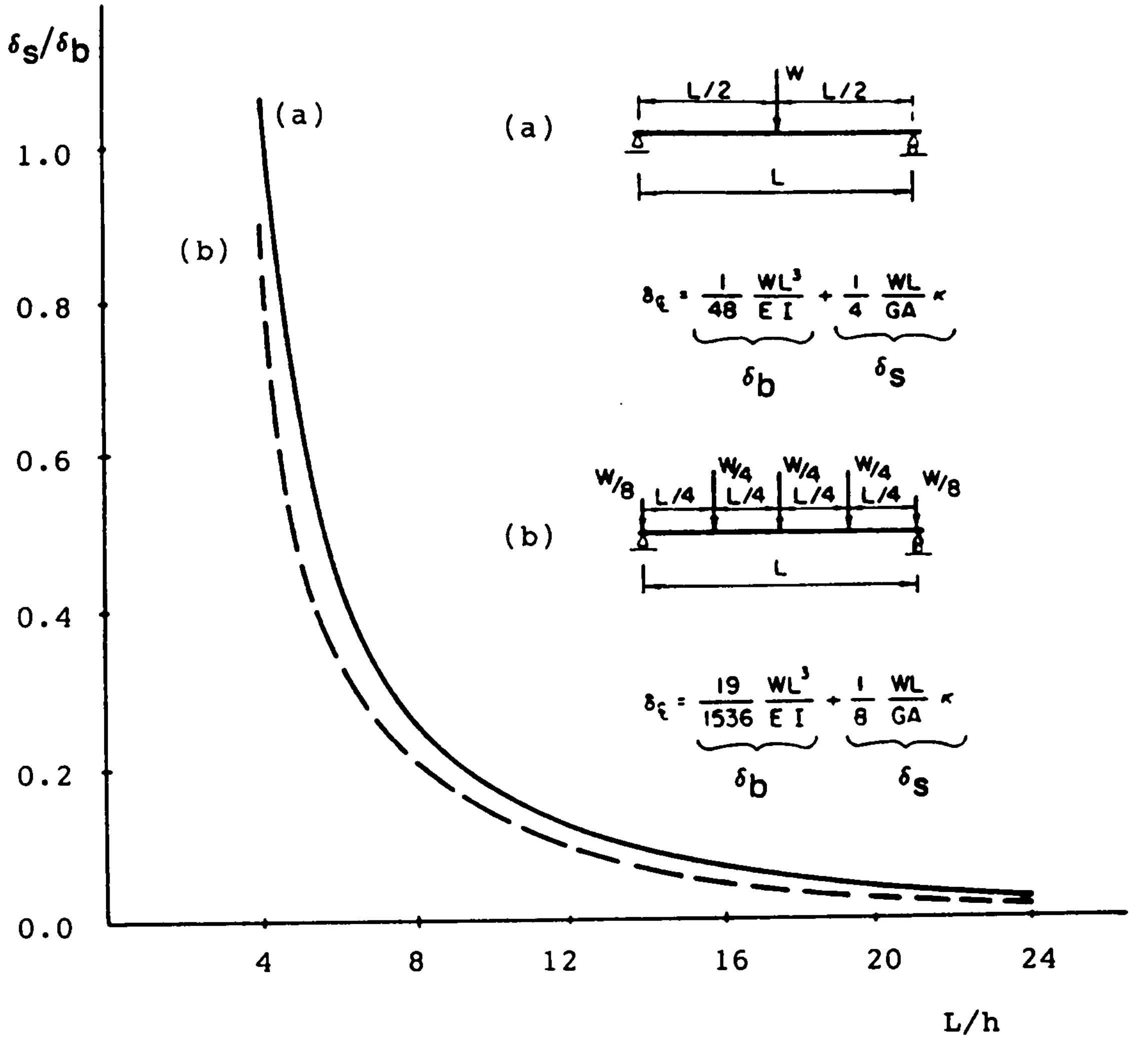


Figure 3.2. Effect of shear deflection as a percentage of bending deflection for two load conditions and different span-to-depth ratios

To emphasize the importance of incorporating shear deformations in mathematical models of timber beams, the ratio of the shear component to the bending component of deflection, $\frac{\delta_s}{\delta_b}$, vs. the span-to-depth ratio of the beam, $\frac{L}{h}$, is plotted in Fig. 3.2 for the two loading conditions shown in the same figure. These plots correspond to a ratio of $\frac{E}{G} = 14$. It can be observed from Fig. 3.2 that at $\frac{L}{h} = 4$ the shear deflection approaches or exceeds the deflection due to bending; at $\frac{L}{h} = 12$, δ_s is approximately 10% of δ_b . On the basis of these examples, it can be concluded that the shear influence on the deflection of wood beams is considerable and must be taken into consideration. The contribution of shear deformation may be particularly important in geometricaly nonlinear problems where coupling of axial and flexural response occurs.

3.5 Structural Model of Glulam

Structural glued-laminated timber (glulam) is an engineered product fabricated from smaller pieces of stress-rated lumber glued together in a timber laminating plant. Glued-laminated timber is produced in accordance with AITC 117-87-MANUFACTURING and ANSI/AITC 190.1-1983. Some important specifications are: the individual laminae cannot exceed 2 inches in thickness but can be produced to any required length by end joining pieces together usually by plain scarf joints or finger joints. The laminae are bonded together under controlled temperature and humidity conditions with synthetic resin adhesives (phenol resorcinol or melamine). The grain (longitudinal fibers) of the laminae are approximately parallel to the length of the member, with a slope of grain as low as 1/16 for outer laminae and 1/12 for inner layers. Allowable stresses for glulam are contained in AITC 117-87-DESIGN, and the design of glulam members and connections are specified in the *Timber Construction Manual* (1985) and the *National Design Specifications for Wood Construction* (1986).

Glued laminated timber offers several advantages over solid-sawn lumber:

1. Fabrication of large structural components from lumber cut from smaller trees.

2. Utilization of lower grade material in the less-stressed midsurface zone of beams, redistribution of defects, and reduction of seasoning checks (cracks due to drying associated with large solid-sawn members).

3. Versatility of producing aesthetically pleasing and structurally efficient shapes, such as curved, doubly curved, tapered, and tapered and curved members.

4. Excellent fire resistance in heavy timber construction. While timber will burn, it forms a char layer on its surface that retards the combustion of the material, maintaining its strength under fire longer than unprotected metals. For example, when the Filene Center for the Performing Arts at Wolf Trap Farm (a multipurpose outdoor theater in Virginia, near Washington, D.C.) burned, all the twisted structural steel was replaced, but the glulam framing retained its design capability and required only in-place sand blasting of its charred surfaces (Civil Engineering, 1986).

The analysis of glued-laminated beams requires a number of sinplifying assumptions:

1. It is assumed that gluelines between laminae are continuous, of infinitesimal thickness, and that no interlayer slip occurs. This assumption is widely accepted by most researchers, provided that the fabrication and quality control of the laminates conform to current standards and specifications (see, for example, Bodig and Jane 1982, ch. 8). This assumption appears to be reasonable for two other reasons: the properties of most adhesives used in the fabrication of glulam are similar to those of wood, and the adhesive is usually less than 1% of the total volume of a laminate. Thus, for practical purposes, glueline contribution to the overall macroscopic member response is ignored.

2. It is assumed, in an average sense, that each lamina is homogeneous. This implies that the mechanical properties of the material do not vary throughout the volume of the lamina. However, it has been shown that the longitudinal elastic modulus, E_L , measured at increments along the length of a lamina is not constant but statistically correlated (Kline and Woeste, 1986). Thus, in research studies, it may be necessary to measure the elastic parameters at increments along the length of the laminae to compute a representative value for each lamina and for the laminate. However, this procedure may not be practical in the analysis of commercially manufactured glulam beams, where stress-rated lumber is assembled in a prescribed lay-up schedule to produce a member of a required strength and stiffness.

3. As stated early in this chapter, fabrication standards limit the slope of grain of the laminae. Therefore, it is assumed that the longitudinal fibers are parallel to the axis of the beam. However, the growth rings of the cross sections of the laminae are randomly oriented, and although each lamina can still be characterized as orthotropic, the material radial (R) and tangential (T) axes do not necessarily coincide with the cross-sectional axes (1, 2), as shown in Fig. 3.3. Thus, to model a glulam beam as orthotropic will require tensor transformation of axes for each layer of the cross section of the beam. Although it is theoretically possible to accurately model the growth rings of the cross section of individual beams, it is not practical in the analysis of structural systems consisting of a large number of members, such as space frames and lattice domes. To efficiently analyze skeletal glulam systems, the beams can be modeled as isotropic in the plane of the cross section. Although this assumption is reflected in current timber design codes, its validity has not been rigorously investigated. Therefore, the feasibility and accuracy of this assumption is established in this study, and the general procedure and details are described in the next section.

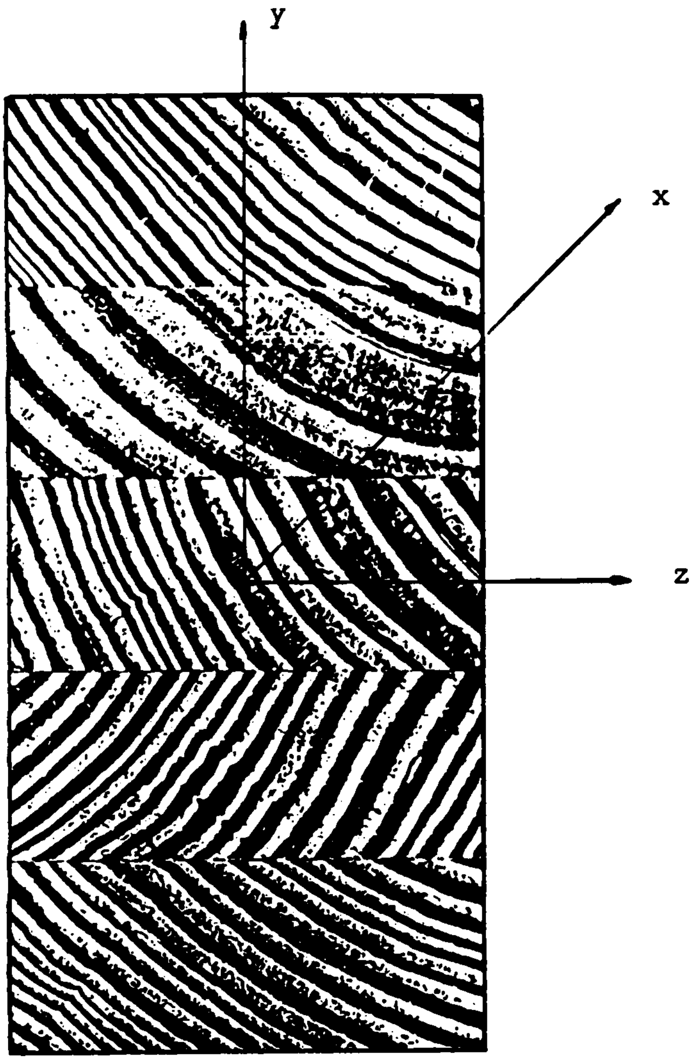


Figure 3.3. Glulam cross section with geometric axes x,y,z. The growth rings of the laminae are randomly oriented.

3.6 Transverse Isotropy and Shear Modulus of Glulam Beams

The stress-strain relations for an orthotropic material in coordinates aligned with principal material directions are given by Eq. (3.1). Based on Timoshenko's beam theory, these equations reduce to:

$$\begin{bmatrix} \sigma_L \\ \sigma_{LR} \\ \sigma_{\mathrm{LT}} \end{bmatrix} = \begin{bmatrix} E_L & 0 & 0 \\ 0 & \frac{1}{\kappa} G_{\mathrm{LR}} & 0 \\ 0 & 0 & \frac{1}{\kappa} G_{\mathrm{LT}} \end{bmatrix} \begin{bmatrix} \varepsilon_L \\ \gamma_{LR} \\ \gamma_{\mathrm{LT}} \end{bmatrix} \tag{3.5}$$

where the indicial notation 1, 2, 3 is replaced by the common L (longitudinal), R (radial), T (tangential) notation of wood; the tensorial strains are replaced by the engineering strains (i.e., $2\varepsilon_{ij} = \gamma_{ij}, i \neq j$); the components of the constitutive matrix c_{ij} are written in terms of the longitudinal elastic modulus E_L and the shear moduli G_{LR} and G_{LT}; the shear moduli are modified by the shear correction factor κ (sec Section 3.4).

Equation (3.5) is restricted to the material symmetry axes of an orthotropic beam. If the beam is considered to be isotropic in the plane of the cross section (the R-T plane), then the principal shear moduli G_{LR} and G_{LT} are equal to each other and can be denoted by a single constant G. In this case, Eq. (3.5) can be written as:

$$\begin{bmatrix} \sigma_x \\ \sigma_{xy} \\ \sigma_{xz} \end{bmatrix} = \begin{bmatrix} E_L & 0 & 0 \\ 0 & \frac{1}{\kappa} G & 0 \\ 0 & 0 & \frac{1}{\kappa} G \end{bmatrix} \begin{bmatrix} \varepsilon_x \\ \gamma_{xy} \\ \gamma_{xz} \end{bmatrix} \tag{3.6}$$

where x, y, z are the geometric axes of the beam, with the longitudinal axis, x, parallel to the fiber direction, L.

Although wood is a complex anisotropic material, the assumption of transverse isotropy greatly simplifies the modeling of glulam beams (Eq. (3.6)). Thus, previous researchers presented arguments for considering glulam beams as isotropic in the plane of the cross section (Mohler and Hemmer, 1977; Van Erp, 1985). Furthermore, beam models that incorporate shear deflections (Eq. (3.6)) require accurate estimates of the shear modulus. Unfortunately, for many wood species, shear modulus data are not readily available. Thus, based on the transverse isotropy assumption, investigators (Trayer and March, 1930; Mohler and Hemmer, 1977; Broker and Schwab, 1988) used Saint-Venant's isotropic torsion theory, including warping effects, to measure the shear modulus of timber beams.

The suitability of the transverse isotropy assumption for timber beams (Eq. (3.6)) can be investigated by measuring the principal material shear moduli G_{LR} and G_{LT}. If these are not significantly different, the material behavior is transversely isotropic. Then, the random orientation of the growth rings can be ignored, and the shear modulus, G, of solid-sawn and glulam beams can be computed from torsion tests using Saint-Venant's isotropic torsion solution. Therefore, the assessment of the validity of transverse isotropy and the determination of the shear moduli of glulam beams can be accomplished through relatively simple torsion tests. The following sections present details of torsion theory and experiments to investigate the validity of transverse isotropy of southern pine beams and to measure the shear moduli of southern pine from torsion tests of small samples.

3.6.1 Review of Torsion Theory

The first analytical expression for torsion was developed for thin wires by Coulomb (1784) while he studied the property of electric charges. The first rigorous solution for torsion of a circular shaft was derived by Navier and published in 1826. Navier extended his formula to rectangular bars by simply replacing the polar moment of inertia of a circular section with that of a rectangular section.

However, the simple extrapolation of Navier's formula to rectangular sections was not supported by experiments. Duleau tested circular and square sections of iron in torsion and found that the shear modulus computed from square sections was 20% less than that from circular sections. Thirty years later, Saint-Venant (1855) introduced the "semi-inverse" method and solved the torsion problem of rectangular sections. His solution showed that when non-circular prismatic members are subjected to torsion, the cross-sections warp and produce out of plane displacements. Lekhnitskii (1963) applied Saint-Venant's solution to anisotropic materials and obtained solutions for circular and rectangular orthotropic bars. This historic sketch has been extracted from a detailed discussion of torsion theories and their historical evolution presented by T. C. Hsu (1984).

The application of anisotropic torsion solution to the experimental determination of the shear moduli of orthotropic materials is described in the literature (Trayer and March, 1930; Semenov, 1966; Tang et al., 1971; Cohen, 1984). A description of torsion of anisotropic bodies and a combination of torsion tests and analyses to compute the shear moduli of sitka spruce in the tangential (G_{LT}) and radial (G_{LR}) planes is given by Trayer and March (1930). A method of obtaining the shear moduli of laminated orthotropic materials by testing specimens in torsion and solving simultaneous equations by a method of successive approximations is described by Semenov (1966). A similar approach is used by Tang et al. (1971) to compute the tangential and radial shear moduli of scarlet oak at various moisture contents.

To determine the radial and tangential shear moduli of wood from torsion experiments requires test specimens with coincident principal material and geometric axes (Fig. 3.1). However, due to its variable nature, solid-sawn and glued-laminated timber rarely conform to orthotropic symmetry (Fig. 3.3). The random orientation of growth rings, particularly in glulam, produces a response that approaches that of a transversely isotropic material (Van Erp,1985). Therefore, a number of researchers have used Saint-Venant's isotropic torsion solution to determine the shear modulus of structural timber. For example, Mohler and Hemmer (1977) stated that the computation of shear stresses and shear moduli based on orthotropic theory is too complex for practical use and not applicable to beams of large cross sections with random growth ring orientations and defects. They also recognized that orthotropic theory applied to glulam beams would generally require complicated tensor transformations for each lamina. Thus, they recommended using isotropic theory for practical computations. Their work was extended by Broker and Schwab (1988), who computed the shear modulus of several softwood and hardwood species from torsion of prismatic bars with square cross sections (20x20x400 mm). For the same specimen size, Scharr (1986) studied the temperature effect on the shear modulus of three species of wood. He presented regression curves of mean values demonstrating that the shear modulus is a time dependent property affected by temperature. An overview of the testing program used in this study to investigate the transverse isotropy assumption for southern pine beams is presented next.

3.6.2 Testing Program: Overview

In the linear range, the torque, T, applied to a prismatic bar is proportional to the measured angle of twist, θ. The proportionality constant, K, is the torsional stiffness, which is a function of the geometry of the cross section and the shear moduli of the test sample. In general, the torsional stiffness can be expressed as

$$K=\frac{T}{\theta} \tag{3.7}$$

After the torsional stiffness is experimentally determined, the shear moduli can be computed from an appropriate torsion theory. To evaluate the validity of transverse isotropy for southern pine and to compute the shear modulus of southern pine glulam beams, a two-part experimental testing program was developed (Fig. 3.4):

First, the principal shear moduli, G_{LR} and G_{LT}, were measured from torsion tests of rectangular southern pine specimens manufactured with coìncident geometric and material symmetry planes

Torsion Testing Program

Orthotropic Samples

Rectangular samples

G_{LR} and G_{LT} are
computed from orthotropic
solution.
18 samples: depth/width =
2, 3, and 4.
(6 samples/ratio)

Circular samples

The values of G_{LR} and G_{LT}
are checked by comparing
the measured and the
computed stiffnesses.
6 samples: diameter = 1″

Glulam Samples

Rectangular and circular samples

The shear modulus G
is computed
from isotropic torsion solution.
12 rectangular samples:
Depth/width = 1, 2, 3, and 4.
4 circular samples: diameter = 1″

Fig. 3.4. Organizational Testing Program for the Torsion Experiment

(rhombic symmetry). For each sample, a torsional stiffness was obtained from the slope of the torque vs. angle of twist curve, and the two principal shear moduli were computed from Lekhnitskii's orthotropic solution (1963). Since the computation of the shear moduli of orthotropic rectangular samples is quite complicated, it was necessary to establish whether or not the calculated values were acceptable. Therefore, samples of circular cross sections, with rhombic symmetry (Fig. 3.5), were tested in torsion, and the measured average torsional stiffness of the circular samples (Eq. 3.7) was compared to an analytical stiffness which was computed from the average G_{LR} and G_{LT} values obtained for the rectangular samples.

Second, small rectangular and circular glulam samples were tested in torsion and their shear moduli were computed from Saint Venant's torsion solution for homogeneous, isotropic bars.

To emulate the commercial glulam manufacturing process, all the test specimens were cut from select structural grade southern pine boards randomly selected from a local lumber yard. The average moisture content (MC) of the samples was 10.2% (COV = 5%), and the average specific gravity (SG) was 0.62 (COV = 7%). The tests were conducted under controlled ambient conditions with a relative humidity of 64%, and a temperature of 70°. The samples contained no knots, splits, or other visible defects.

3.6.3 Test Equipment

The samples were tested in an INSTRON servo-hydraulic torsion testing machine with a load-range capacity of 10,000 in-lb. The load accuracy was ± 1.0 in-lb, and the angle-of-twist accuracy was ± 0.005°. The machine allowed free longitudinal displacement of the sample during testing (i.e., unrestrained warping displacements). The ends of the rectangular samples were clamped in a specially fabricated fixture shown in Fig. 3.6.

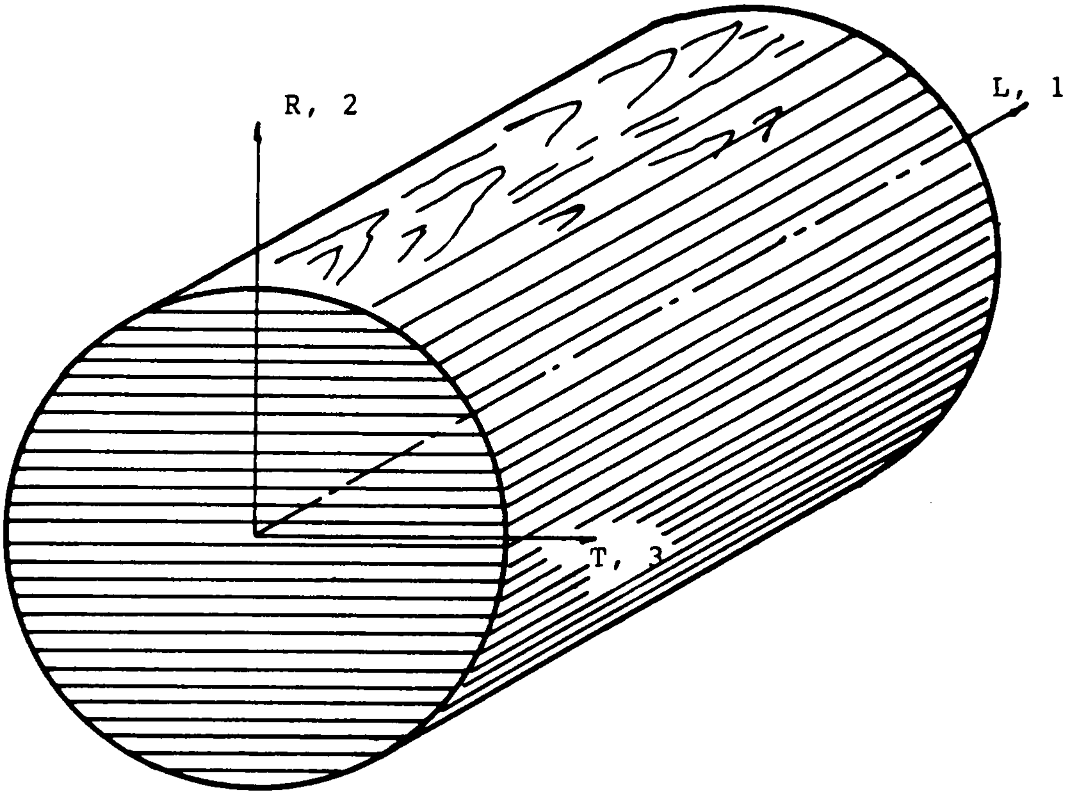


Figure 3.5. Samples of circular cross section with rhombic symmetry used to check G_{LR} and G_{LT} computed by testing rectangular samples.

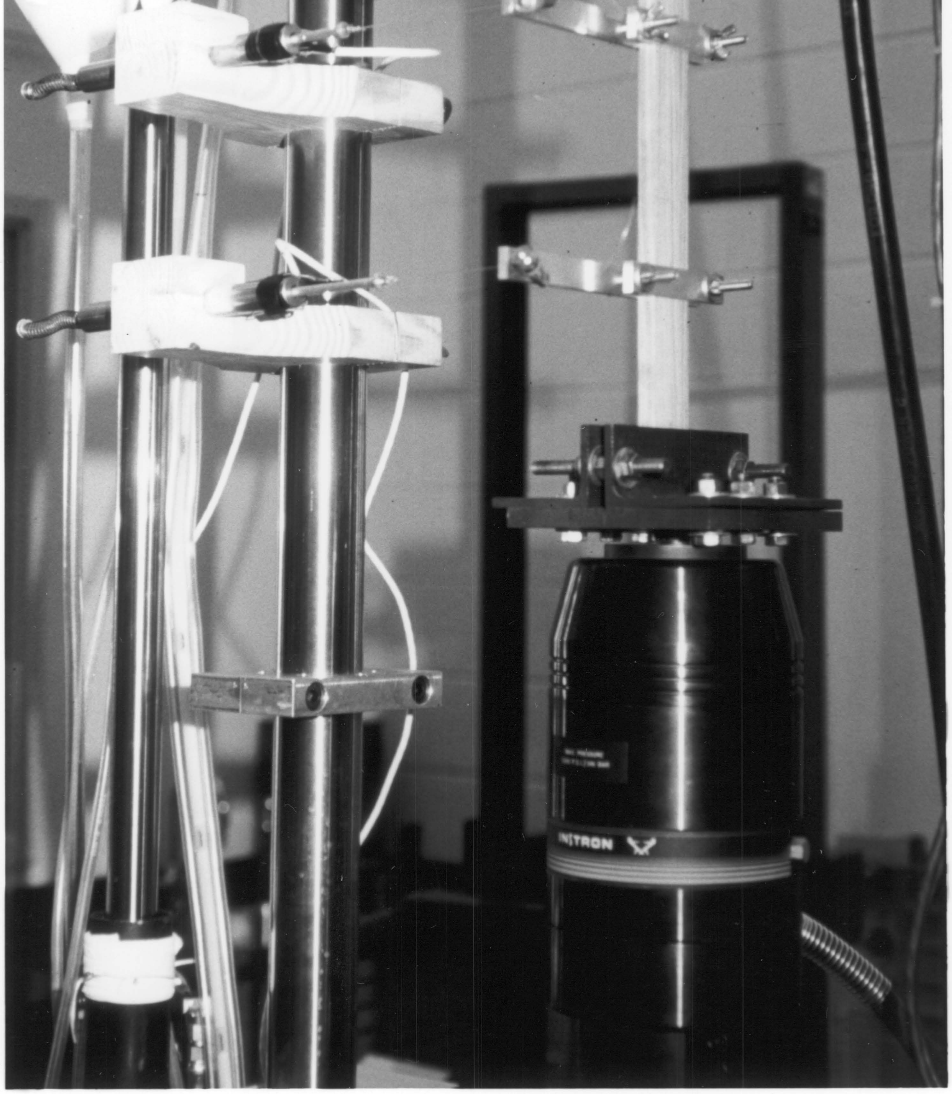


Figure 3.6. Details of the torsion experiment.

Broker and Schwab (1988) studied the boundary disturbance caused by slippage and/or crushing of the specimen at the grips. They showed that the value of the shear modulus based on the total angle of twist, measured relative to the end-grips, was consistently underestimated. To avoid boundary effects, they recommended measuring the relative angle of rotation between two sections of the specimen away from the end-grips. Similarly, for the southern pine samples tested in torsion, the angles of twist were measured over a six-inch gage length of the specimens. The arrangement consisted of two aluminum bars attached to the sample at points located three inches away from the end-grips. As torque was applied, the sample rotation caused the aluminum bars to pull the cores of two linear voltage differential transducers that were secured to the testing machine (Fig. 3.6). The displacements of the transducers were continuously recorded by a data acquisition system, and the angle of twist was computed from the output of the transducers and the geometry of the testing set-up.

Mohler and Hemmer (1977) showed that the shear modulus of timber beams is a time-dependent property. Thus, to minimize the influence of creep, all specimens were tested at a displacement rate of three degrees per minute. Each sample was tested to a twist angle of approximately 10°.

3.6.4 Determination of Shear Modulus

1. **Shear moduli of orthotropic southern pine samples.** Lekhnitskii (1963) presented the solution for torsion of rectangular orthotropic bars in principal material directions. For a torque applied about the longitudinal axis, L (Fig. 3.7), the torsional stiffness is given by

$$K = \frac{T}{\theta} = \frac{G_{\mathrm{LT}} a b^3}{L} \frac{1}{\beta} \tag{3.8}$$

where G_{LT} is the principal shear modulus in the tangential plane, and a, b, and L are the dimensions of the bar (Fig. 3.7). The parameter, β, which is a function of both principal shear moduli, is defined as

$$\beta = \frac{32}{\pi^4}\frac{a^2}{b^2}\frac{G_{LR}}{G_{\mathrm{LT}}}\sum_{i=1,3,5\ldots}^{\infty}\frac{1}{i^4}\left[1-\frac{2}{i\pi}\frac{a}{b}\sqrt{\frac{G_{LR}}{G_{\mathrm{LT}}}}\tanh\left(\frac{i\pi}{2}\frac{b}{a}\sqrt{\frac{G_{\mathrm{LT}}}{G_{LR}}}\right)\right] \qquad (3.9)$$

The torsional stiffness in Eq. (3.8) is a function of the two principal shear moduli, and these cannot be determined from a torsion test of a single specimen. Thus, the most common approach is to compute the shear moduli from two separate torsion tests on two bars with different cross-sectional dimensions (Tang et al., 1971; Cohen, 1984). In this study, G_{LR} and G_{LT} were determined from torsion tests of paired samples manufactured with the same cross-sectional dimensions from flat-sawn and quarter-sawn lumber. This produced specimens with growth ring orientations normal to each other (Fig. 3.8). Three specimen sizes (Table 3.1), with three replications of each size, were used to study the shape effect on the shear moduli. Thus, each size group consisted of three flat-sawn and three quarter-sawn specimens. By pairing a flat-sawn specimen with each quarter-sawn specimen, a total of nine sets of values for G_{LR} and G_{LT} were obtained for each size group. The following equations were used for each pair of samples:

$$G_{\mathrm{LT}} = \frac{K_1 L}{ab^3}\frac{1}{\beta_1} \quad \text{(For fiber orientation of Fig. 3.8(a))} \qquad (3.10)$$

$$G_{LR} = \frac{K_2 L}{ab^3}\frac{1}{\beta_2} \quad \text{(For fiber orientation of Fig. 3.8(b))} \qquad (3.11)$$

where β_1 and β_2 are expressed as

$$\beta_1 = \frac{32}{\pi^4}\frac{a^2}{b^2}\frac{G_{LR}}{G_{\mathrm{LT}}}\sum_{i=1,3,5,\ldots}^{n}\frac{1}{i^4}[1-\frac{2}{i\pi}\frac{a}{b}\sqrt{\frac{G_{LR}}{G_{\mathrm{LT}}}}\tanh(\frac{i\pi}{2}\frac{b}{2}\sqrt{\frac{G_{\mathrm{LT}}}{G_{LR}}})] \qquad (3.12)$$

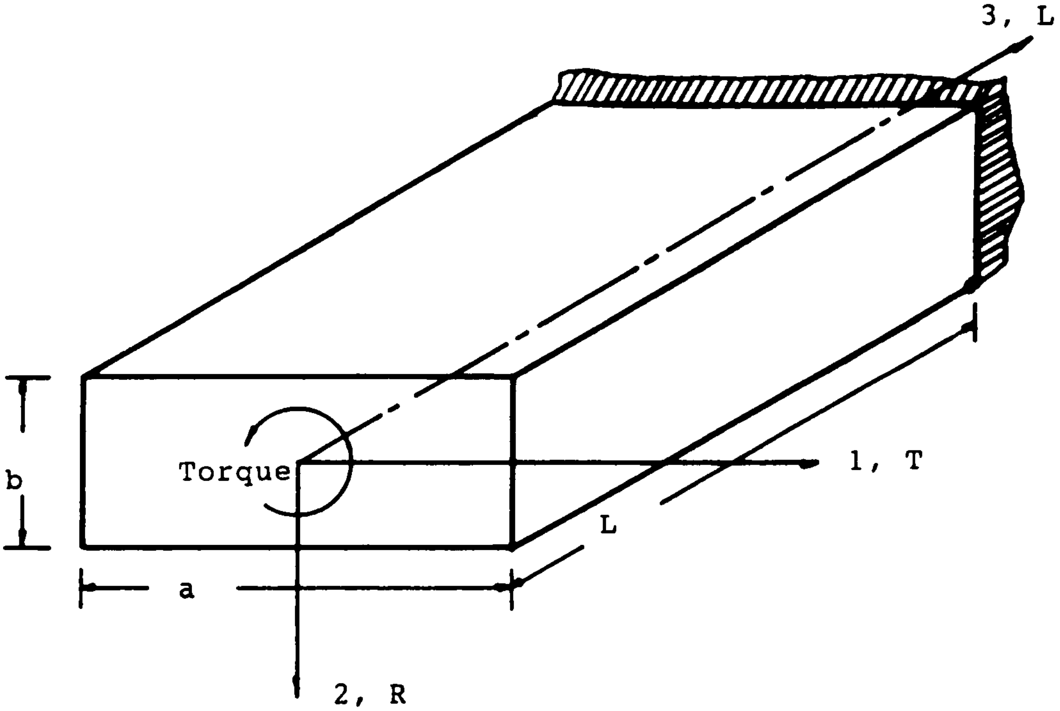
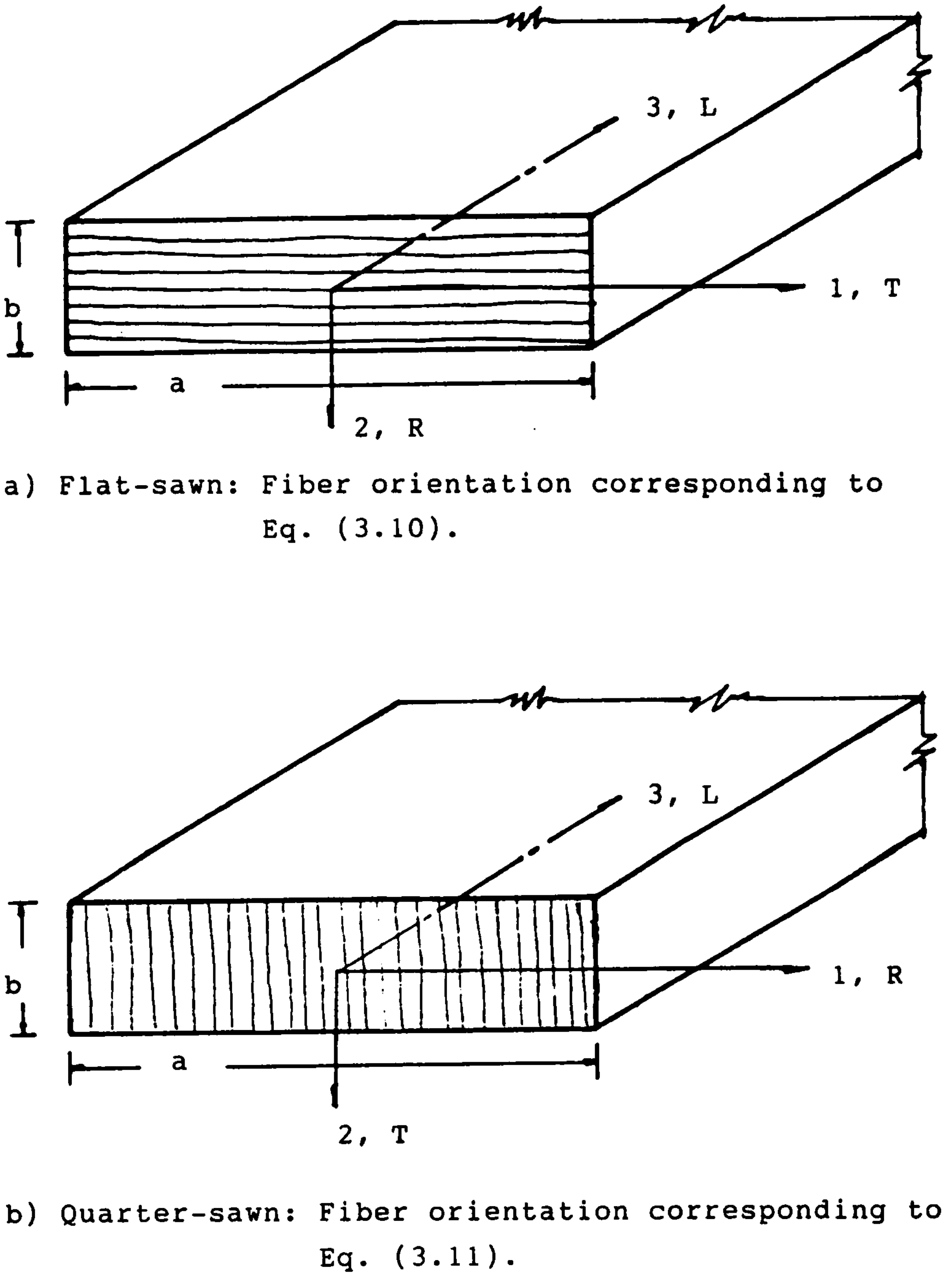


Figure 3.7. Orientation of the torsion sample corresponding to Lekhnitskii's solution (Eq. 3.8).



(a) Flat-sawn: Fiber orientation corresponding to Eq. (3.10).

(b) Quarter-sawn: Fiber orientation corresponding to Eq. (3.11).

Figure 3.8. Fiber orientation of a pair of samples, with the same cross-sectional dimensions, to determine G_{LR} and G_{LT}.

Table 3.1. Test-samples with three planes of material symmetry.

Size category	Number	Size (axbxL, inches)	Fiber orientation
SP1	3	1.0x0.5x12	P
SN1	3	1.0x0.5x12	N
SP2	3	1.5x0.5x12	P
SN2	3	1.5x0.5x12	N
SP3	3	2.0x0.5x12	P
SN3	3	2.0x0.5x12	N

NOTE: P = Parallel to longer cross-section dimension (Fig. 3.8(a)).

N = Normal to longer cross-section dimension (Fig. 3.8(b)).

Average moisture content = 11.12% (COV=8.2%)

Average sp. gravity = 0.63 (COV= 12%)

$$\beta_2 = \frac{32}{\pi^4}\frac{a^2}{b^2}\frac{G_{\mathrm{LT}}}{G_{LR}}\sum_{i=1,3,5,\ldots}^{n}\frac{1}{i^4}[1-\frac{2}{i\pi}\frac{a}{b}\sqrt{\frac{G_{\mathrm{LT}}}{G_{LR}}}\tanh(\frac{i\pi}{2}\frac{b}{2}\sqrt{\frac{G_{LR}}{G_{\mathrm{LT}}}})] \quad (3.13)$$

Equations (3.10) and (3.12) were solved by a method of successive substitutions (Semenov 1966). As a first approximation, the ratio of the shear moduli was set equal to 1.0 (i.e., $G_{LR} = G_{\mathrm{LT}}$), and $\beta_1^{(1)}$ and $\beta_2^{(1)}$ were calculated from Eqs. (3.12) and (3.13) by computing the summations for a finite number of terms (i = 1,3,5, ..., n) until they converged to a tolerance of 10^{-4}. Then, from Eqs. (3.10) and (3.11), the moduli $G_{LR}^{(1)}$ and $G_{LT}^{(1)}$ were computed and used to calculate $\beta_1^{(2)}$, $\beta_2^{(2)}$, $G_{LR}^{(2)}$, and $G_{LT}^{(2)}$. This iterative procedure was continued until the shear moduli values converged to a tolerance of 5.0 psi. Alternatively, a commercial computer program for the solution of a system of nonlinear equations can be used.

For each specimen, the torsional stiffness was calculated from the linear portion of the applied torque vs. angle of twist curve. Figure 3.9 shows plots of the mean stiffness and ranges for the flat-sawn samples of each size group. The mean G_{LR} and G_{LT} values of the three size groups are presented in Table 3.2. The grand means of all the samples are: $G_{LR} = 160{,}870$ psi (COV = 1.0%) and $G_{\mathrm{LT}} = 150{,}156$ psi (COV = 2.4%). The difference between G_{LR} and G_{LT} is not significant (approximately 7%) and may be neglected in practice. Thus, the southern pine samples tested in this study behave as transversely isotropic beams. The small difference between mechanical properties in the radial and tangential directions is characteristic of many species (Grossman 1973). For example, Trayer and March (1930) concluded that the difference between G_{LR} and G_{LT} of sitka spruce was about 10%, and no great error was introduced by computing the shear modulus from torsion tests of circular samples and isotropic theory.

The values of the shear moduli computed from orthotropic torsion theory (Eqs. (3.10) and (3.11)) are quite sensitive to small variations of the cross-sectional dimensions and the meas-

Table 3.2. Mean values of G_{LR} and G_{LT} for the three sample size-groups.

Sample combination*	depth/width	Average G_{LR} (psi)	Average G_{LT} (psi)
SP1 - SN1	2	161,046 (COV=0.87%)	148,669 (COV=2.7%)
SP2 - SN2	3	161,782 (COV=0.96%)	150,444 (COV=2.9%)
SP3 - SN3	4	159,783 (COV=0.74%)	151,355 (COV=1.0%)
Grand Mean Values		160,870 (COV=0.98%)	150,156 (COV=2.4%)

* Nine values for each combination (see Table 3.1).

NOTE: COV = coefficient of variation

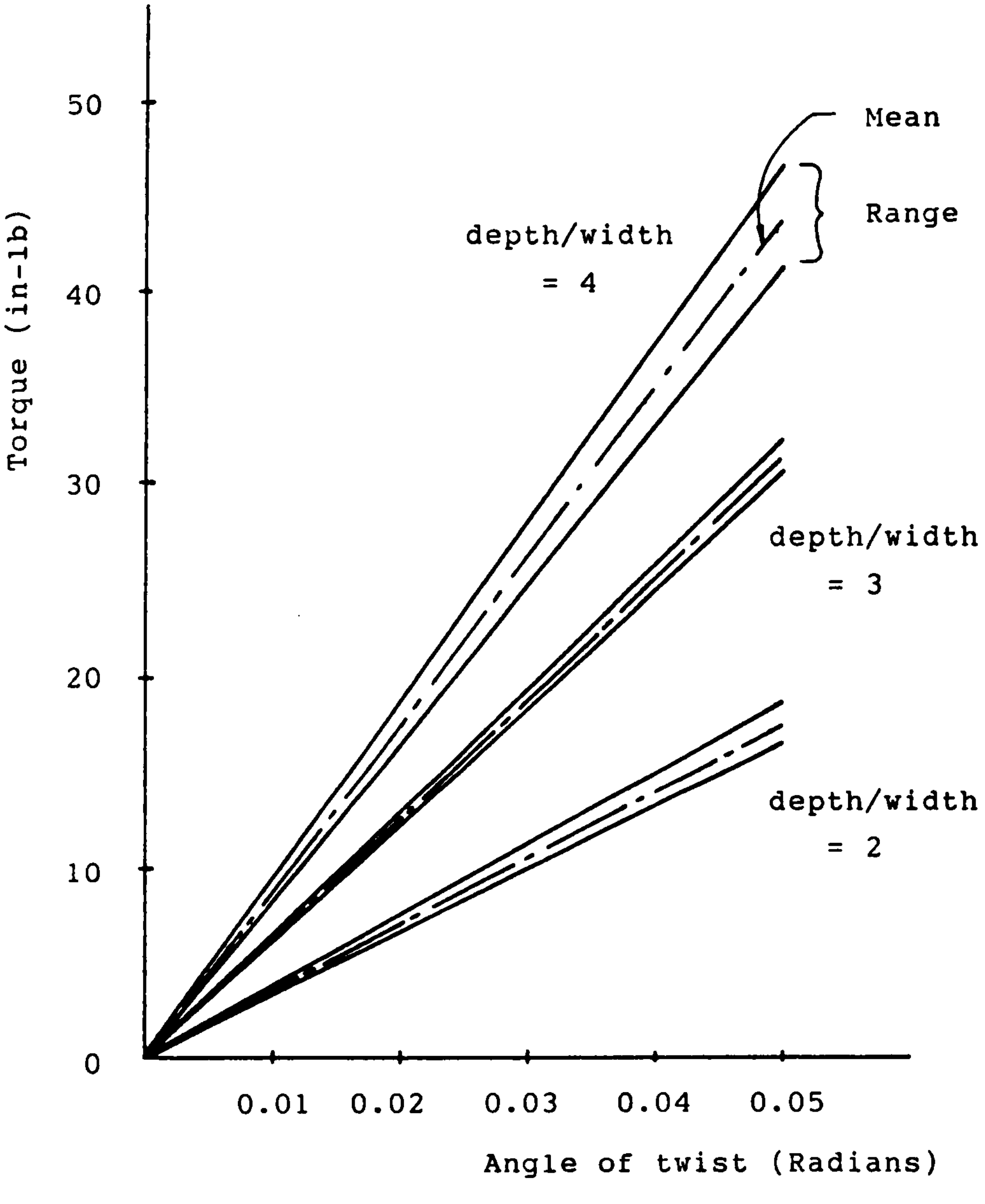


Figure 3.9. Mean and range values of the torsional stiffnesses (K = Torque/angle of twist) of the flat-sawn samples (see Fig. 3.8(a)).

ured torsional stiffness. Thus, to verify the shear moduli values obtained for the rectangular samples, six specimens of circular cross section (diameter = 1"), exhibiting rhombic symmetric (Fig. 3.5), were tested in torsion. From the orthotropic torsion solution for round bars presented by Lekhnitskii (1963), the torsional stiffness can be written as

$$K=\frac{T}{\theta}=\frac{\pi r^4 G_{LR} G_{LT}}{L(G_{LR}+G_{LT})} \tag{3.14}$$

where r is the radius of the cross section, and L is the length. From the G_{LR} and G_{LT} grand mean values of the rectangular samples, the stiffness of each circular sample was computed from Eq. (3.14). The computed average stiffness was compared to the experimental average stiffness of the circular samples (Table 3.3). Since the difference between the experimental and analystical means was less than 3%, we can conclude that the measured values of G_{LR} and G_{LT} are reasonably accurate.

2. **Shear modulus of southern pine glulam samples.** The shear moduli of southern pine glulam samples were computed from isotropic torsion theory. Saint-Venant's solution for homogeneous, elastic, isotropic, rectangular sections is obtained from Eq. (3.8) by replacing G_{LR} and G_{LT} by G:

$$K=\frac{T}{\theta}=\frac{Gab^3}{L}k \tag{3.15}$$

where k depends on the width-to-depth ratio (a/b) of the cross section (Timoshenko, 1970). Similarly, Eq. (3.14) reduces to Navier's torsion solution for isotropic circular cylinders:

$$K=\frac{T}{\theta}=\frac{\pi r^4 G}{2L} \tag{3.16}$$

Twelve rectangular and four circular southern pine glulam samples were tested in torsion (Table 3.4). The samples were fabricated with three to six laminae (lamina thickness = 0.35"). To study the shape-effect on the shear modulus, rectangular samples of four width-to-depth

Table 3.3 Measured and computed torsional stiffnesses of the orthotropic circular samples (diameter = 1").

Sample	Measured K (in-lb/rad)	Computed K (in-lb/rad)	S.G.	M.C. (%)
C1	1302	1296	0.56	9.4
C2	1335	1288	0.62	8.4
C3	1254	1288	0.54	9.3
C4	1293	1276	0.56	9.4
C5	1322	1278	0.55	9.5
C6	1328	1251	0.63	9.4
Mean Values	1306 (COV=2.3%)	1280 (COV= 1.2%)	0.58 (COV=6.6%)	9.2 (COV=4.5%)

ratios, with three replications per ratio, were tested. The average shear modulus of each cross-sectional shape is presented in Table 3.4. The grand means of the rectangular and circular glulam samples are respectively 162,051 psi (COV = 0.61%) and 162,017 psi (COV = 0.22%).

3.6.5 Conclusions

From the results of the testing experiments, the following conclusions can be made:

1. The average values of the principal shear moduli G_{LR} and G_{LT} of the orthotropic samples are: G_{LR} = 160,870 psi and G_{LT} = 150,156 psi. The difference between these values is not significant, an the tested specimens can be characterized as transversely isotropic. This conclusion can be extended to full-size beams, and the shear modulus can be computed from torsion tests and isotropic theory.

2. The average shear moduli of the rectangular and circular glulam samples are 162,051 psi and 162,017 psi, respectively. The test results showed that the depth-to-width ratios of the rectangular samples (depth/width = 1, 2, 3, and 4) had no effect on the values of the computed shear moduli. Therefore, in practice, the shear moduli of southern pine glulam beams may be estimated from torsion tests of small glulam samples (e.g., 1″ x 1″ in cross section). However, torsion tests of full-size glulam beams and statistical analyses are needed for the precise determination of design shearing modulus data.

Table 3.4. Mean values of G for the rectangular and circular glulam samples.

Cross section	Average G (psi)	Average (S.G.)	Average M.C. (%)	Number of samples
Rectangular Samples				
1"x1"	161,526 (COV=0.8%)	0.61	10.69	3
1"x0.5"	161,480 (COV=0.32%)	0.63	10.41	3
1.5"x0.5"	162,430 (COV=0.60%)	0.61	10.73	3
2.0"x0.5"	162,766 (COV=0.46%)	0.56	10.56	3
Grand Mean Values	162,051 (COV=0.61%)	0.60	10.60	
Circular Samples				
Diameter = 1"	162,017 (COV=0.22%)	0.67	9.74	4

3.7 Determination of the Longitudinal Elastic Modulus, E_L, for Glulam Beams

The constitutive matrix for a 3-d glulam beam, including shear deformations, is given by Eq. (3.5). This matrix requires the elastic parameters G and E_L. The shear modulus, G, of the material can be computed from torsion tests as discussed in Section 3.6. The longitudinal elastic modulus, E_L, for glued-laminated beams can be estimated from the moduli of the constituent laminae. The modulus of elasticity is variable and statistically correlated along the length of a lamina. For example, Kline et al. (1986) measured the elastic moduli at 30-inch increments along the length of full-size boards and proposed a second order Markov process to model the lengthwise variability of E_L for two grades of southern pine.

In this study, a 3-d beam finite element model for glulam is developed (see Ch. 4) and experimentally verified (see Ch. 5) by testing straight and curved southern pine glulam beams under combined loading. The glulam beams were manufactured in the laboratory from selected southern pine boards obtained from a local lumber supplier. Before the beams were assembled, the laminae were tested in tension to measure the longitudinal elastic moduli at 30-inch increments along their lengths. Each lamina was subjected to constant tensile stress, and the strains were measured at the centers of the 30-inch increments with electrical transducers of 2-inch gage lengths. The details of the laboratory-built "clip-on" electrical transducer (CET) are published elsewhere (Loferski et al., 1989).

The laminae of seven glulam beams (4 straight and 3 curved) were tested in tension (a total of 49 laminae), and the complete experimental details and results are discussed by Yadama (1989). As an illustration, the test results of the laminae for two beams, one curved and one straight, are given in Tables 3.5 and 3.6, and it can be observed that the differences in E_L for two adjacent increments of a lamina can be as much as 40%.

In the finite element analysis of a laboratory-built glulam beam, the variability of elastic parameters can be simulated by modeling the beam with a fine mesh of either continuum elements (e.g., a 3-d version of Foschi and Barret's element, (1980)) or structural beam elements. For example, each lamina of the curved beam of Table 3.5 can be modeled by a mesh of 3 elements to account for the three measurements of E_L along the lamina. A 12-node brick element, with three degrees of freedom per node, can be used to model the beam. This results in 27 elements and 420 degrees of freedom (Fig. 3.10(a)). The alternative is to use a one-dimensional, 3-node structural beam element with six degrees of freedom per node. Each third of the beam is modeled by one element, which results in 3 elements and 42 degrees of freedom (Fig. 3.10(b)). Since the formulation of structural beam elements allows complete generality in material behavior at an integration point of the beam cross section (Fig. 3.10(b)), the variability of E_L through the depth of the beam can still be simulated by each element of the mesh.

In the analysis of timber space frames and lattice domes, consisting of hundreds of beams, the use of continuum elements is impractical, if not impossible, with today's computing technology (Davalos, et al., 1988). Thus, structural beam elements based on engineering theory (e.g., Timoshenko's beam theory and Saint-Venant's torsion theory) must be used. These elements can efficiently represent the dominant structural response of space glulam beams and are computationally feasible. Moreover, information about the variability of elastic parameters of commercially manufactured glulam beams is not available, and either constants or statistical mean values for the whole volume of each beam must be used. In Tables 3.5 and 3.6, the average values of E_L for each section of the beams are given as well as the averages computed from the transformed section formula recommended by ASTM 3737-87. In the finite element analysis of the beams (see Ch. 5), the grand average values of E_L are used.

Table 3.5. Longitudinal elastic modulus at increments along the lengths of the laminae of the curved beam C1. All laminae are 1"x4"x8'.

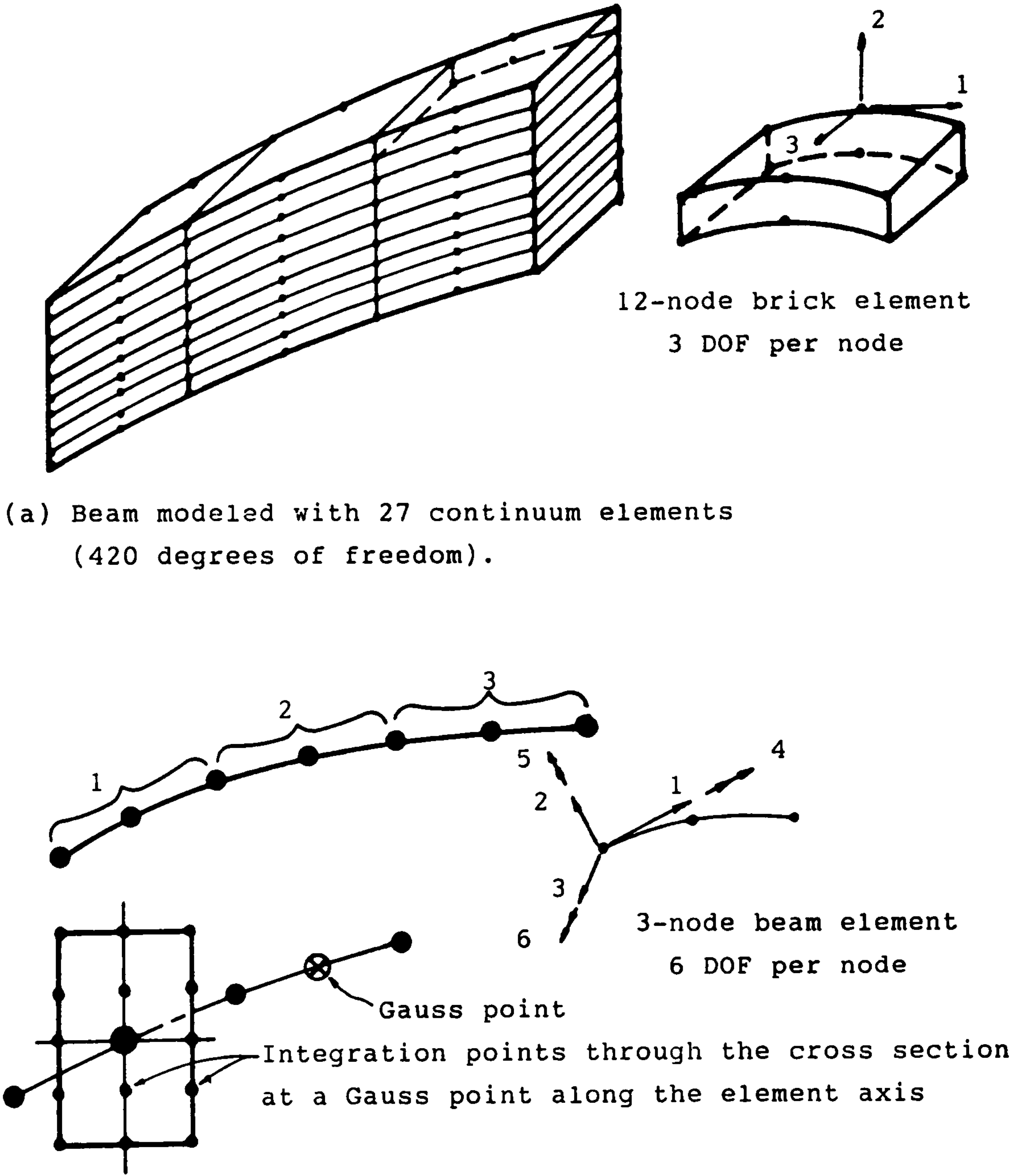
Lamina		E_L (10^6 psi) at increments along the length	
number	Section 1	Section 2	Section 3
1	2.38204	1.73614	2.65586
2	1.92218	1.72618	2.56968
3	1.61312	2.01260	2.46156
4	1.35788	1.77898	1.94808
5	1.92310	1.49130	1.50222
6	1.72594	1.43432	1.05132
7	1.72620	1.48600	1.84988
8	1.41304	2.50532	2.09086
9	2.29114	1.98206	1.85568
Average*	1.81718	1.79477	1.998349
Average by ASTM-3737-87	1.87535	1.82099	2.12003

* Grand average = 1.87 x 10^6 psi

Table 3.6. Longitudinal elastic modulus at increments along the lengths of the laminae of the straight beam S3. All laminae are 2"x6"x 12'.

Lamina number	E_L (10^6 psi) at increments along the length Section 1	Section 2	Section 3
1	2.66874	2.36966	2.03044
2	1.25404	1.79872	1.87676
3	1.56338	1.91560	1.62074
4	1.44502	1.93492	2.83496
5	1.80782	2.09936	2.42618
Average*	1.74779	2.02365	2.15782
Average by ASTM-3737-87	1.89995	2.02874	2.14217

* Grand average = 1.98 x 10^6 psi



(a) Beam modeled with 27 continuum elements
(420 degrees of freedom).

(b) Beam modeled with 3 structural elements
(42 degrees of freedom).

Figure 3.10. Curved beam of Table 3.5 modeled with continuum and structural elements.

3.8 *Summary of Important Considerations for the Finite Element Modeling of Space Glulam Beams*

The practical and efficient modeling of glulam beams used in space frames and lattice domes can be accomplished by considering the following recommendations:

1. The material can be considered continuous, homogeneous and transversely isotropic (Sections 3.5 and 3.6).

2. For laboratory specimens, the longitudinal elastic modulus, E_L , can be computed from tension tests of the laminae (Section 3.7), and the shear modulus can be estimated from torsion tests and Saint-Venant's isotropic torsion solution (Section 3.6). However, for commercially manufactured glulam beams, average values of elastic constants for the volume of the beam must be used (Section 3.7).

3. The formulation of the beam finite element must include shear deformations (Section 3.4).

4. For curved glulam beams used in lattice domes (radius of curvature > 100 ft.), the radial stresses can usually be neglected (Davalos, 1987).

5. The ultimate load capacity of single-layer timber space frames is usually governed by elastic instability (see Ch. 6). Thus, the finite element analysis of the space structure must include geometric nonlinearities. Material nonlinearity can be included in the finite element model following the procedure described by Conners (1989). The incorporation of creep laws in the finite element analysis of timber space frames is discussed by Holzer et al. (1989).

Chapter 4

FORMULATION OF A NONLINEAR 3-D BEAM FINITE ELEMENT

4.1 Introduction

The development of the continuum mechanics equilibrium equation in the Updated Lagrangian formulation is presented in Chapter 2. Based on this equation and the modeling considerations for glulam beams given in Chapter 3, the development of a nonlinear isoparametric beam finite element (isobeam) is presented in this chapter. Of the many beam elements that have been proposed for general nonlinear analysis, the isoparametric element introduced by Bathe (1982) was selected in this study for the modeling of space glulam beams. The formulation of this one-dimensional structural element is based on Mindlin's (Timoshenko's) beam theory, and it is consistent with the continuum mechanics equations of motion. The isobeam is particularly suitable for modeling glulam beams for the following reasons:

1. Coordinate interpolations permit the respresentation of straight, curved, and tapered elements by the same formulation. Thus, the element can model the curved geometry of the members of glulam domes.

2. Shear deformations can be modeled, which is important in timber beams because of the high ratio of the longitudinal modulus to the shear modulus ($\frac{E}{G}$) of the material.

3. The formulation can incorporate torsional warping displacements. Thus, the element can accurately represent the torsional stiffness of a space beam, which is of particular importance in stability investigations.

4. It can model axial, biaxial bending, shearing, and torsional deformations (including warping).

5. It can model transversely isotropic beams, a fundamental modeling assumption for glulam beams proposed in this study.

6. It can model geometric and material nonlinearities, which are important in determining the failure modes and ultimate load capacities of timber space frames.

7. Numerical integration allows complete generality at an integration point. Therefore, the element can model glulam beams with stiffer outer layers.

The formulation of the isobeam is given in Section 4.2, and the program development, solution of test problems, and experimental verification of the element are presented in Chapter 5. Sections 4.3 and 4.4 discuss solution techniques in nonlinear analysis and the error controls used in iterative solutions.

4.2 *Formulation of the Isobeam*

The formulation of a nonlinear, isoparametric, 3-node beam element of constant cross-section is presented. The isobeam admits large displacements, large rotations, but small strains, and its formulation is based on the kinematic assumption that plane sections originally normal to the beam axis remain plane and undistorted during deformation, but not necessarily normal to the beam axis. Torsional warping displacements are not modeled, and the torsional stiffness corresponds to Navier's torsion theory. For the accurate torsional representation of a rectangular section warping displacements must be interpolated. The displacement functions to account for warping are given by Bathe and Chaudhary (1982). Figure 4.1 shows the beam finite element in the original (at time 0) and deformed (at time t) configurations. In a static analysis, without time dependent effects, time is only a convenient variable that denotes different levels of load application corresponding to different configurations, and it is used as a fictitious measure of progress along an equilibrium path. The following notation introduced by Bathe is used:

${}^{t+\Delta t}_{t}(\ )^{k}_{i}$ = i-component of a variable at node k measured in the configuration at time t + Δ t with respect to the configuration at time t.

${}^{t}x_{i}$ = global coordinate of a point in the beam of time t

a, b = cross-section dimensions (constant)

${}^{t}V^{k}_{si}$, ${}^{t}V^{k}_{ti}$ = i-components of the unit vectors in the directions s and t, at node k at time t. These unit vectors define the principal axes of the cross section.

h_k = Lagrangian interpolation polynomials (functions of r)

r, s, t = natural coordinates of the parent element

η, ξ, ζ = local element coordinates

${}^{t}u_{i}$ = displacement component at time t

u_i = incremental displacement component

${}^{t}\theta_{i}$ = rotation component at time t

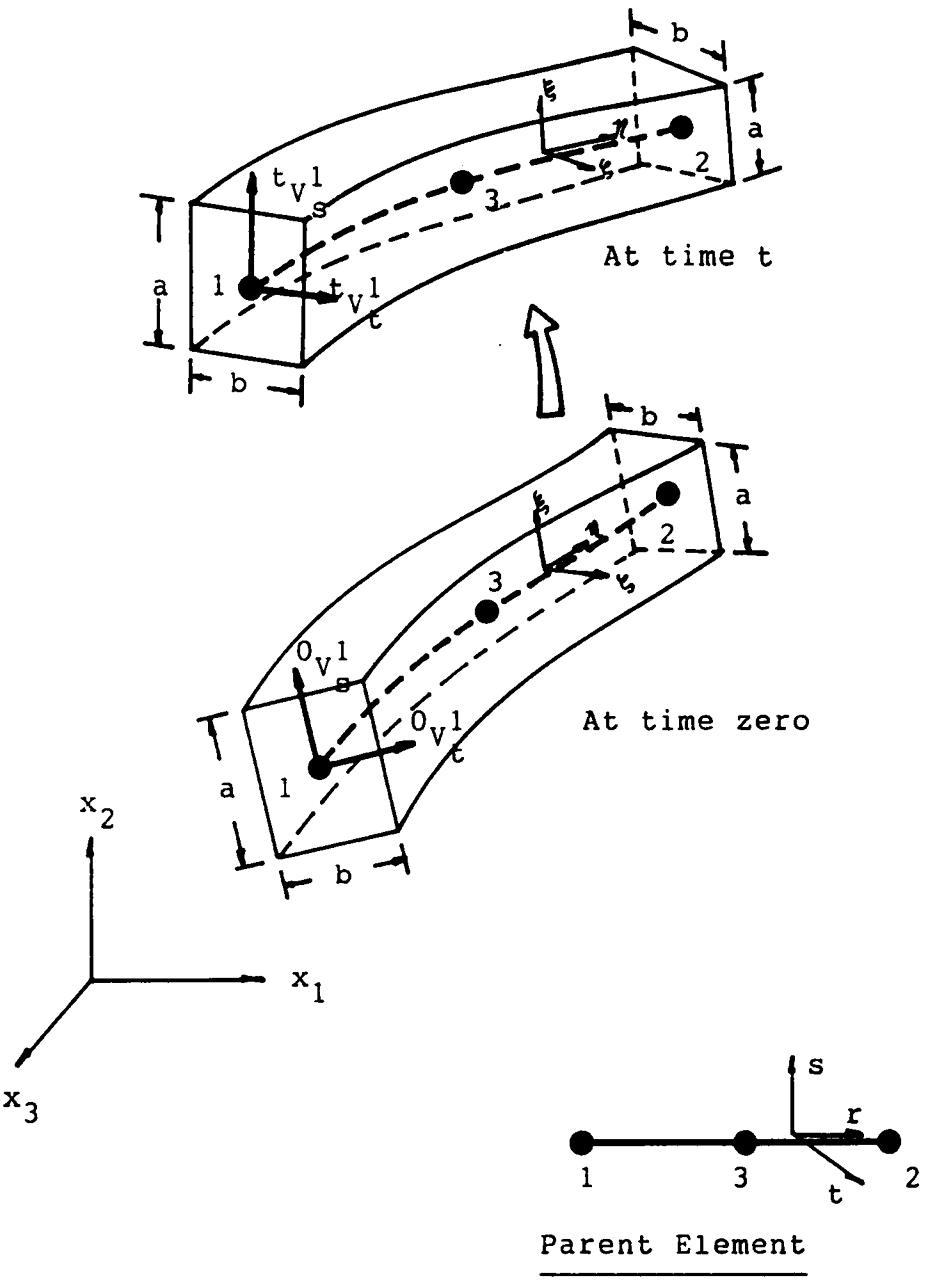


Figure 4.1. 3-d isobeam element in the original (time = 0) and deformed (time = t) configurations.

θ_i^k = incremental rotation component of node k

The cartesian coordinates of a point in the element, as a function of the coordinates r, s, and t are

$$ {}^t x_i = \sum_{k=1}^{3} h_k \, {}^t x_i^k + \frac{sa}{2} \sum_{k=1}^{3} h_k \, {}^t V_{si}^k + \frac{\mathrm{tb}}{2} \sum_{k=1}^{3} h_k \, {}^t V_{ti}^k \tag{4.1} $$

where, i = 1,2,3, and

$$ h_1 = -\frac{r}{2}(1-r), \;\; h_2 = \frac{r}{2}(1+r), \;\; h_3 = (1-r^2) $$

4.2.1 Displacement Derivatives in Global Coordinates, x_i

The displacement and incremental displacement components of any point of the element are (Fig. 4.2)

$$ \begin{aligned} {}^t u_i &= {}^t x_i - {}^o x_i \\ u_i &= {}^{t+\Delta t} x_i - {}^t x_i \end{aligned} \tag{4.2} $$

The incremental components of the unit vectors of the cross section (i.e., the incremental direction cosiness of the principal axes of the cross section) are expressed as follows:

$$ \begin{aligned} V_{si}^k &= {}^{t+\Delta t} V_{si}^k - {}^t V_{si}^k \\ V_{ti}^k &= {}^{t+\Delta t} V_{ti}^k - {}^t V_{ti}^k \end{aligned} \tag{4.3} $$

Substituting Eqs. (4.1) and (4.3) into Eq. (4.2), we obtain expressions for the displacement and incremental displacement components:

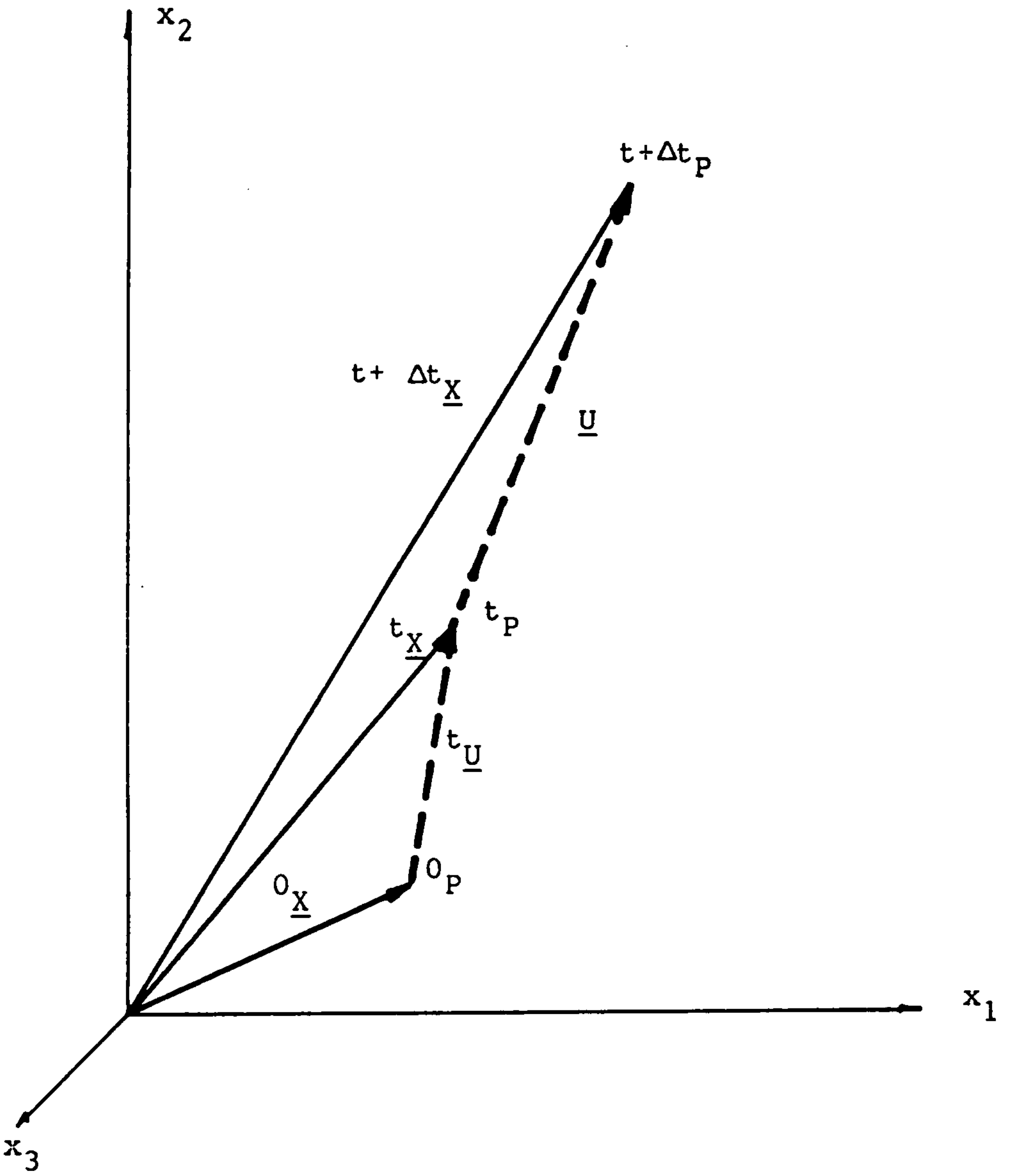


Figure 4.2. Displacement of a point P of the 3-d isobeam element.

$$ {}^t u_i = \sum_{k=1}^{3} h_k\,{}^t u_i^k + \frac{sa}{2}\sum_{k=1}^{3} h_k({}^t V_{si}^k - {}^0 V_{si}^k) + \frac{tb}{2}\sum_{k=1}^{3} h_k({}^t V_{ti}^k - {}^0 V_{ti}^k) \tag{4.4} $$

$$ u_i = \sum_{k=1}^{3} h_k u_i^k + \frac{sa}{2}\sum_{k=1}^{3} h_k V_{si}^k + \frac{tb}{2}\sum_{k=1}^{3} h_k V_{ti}^k \tag{4.5} $$

The incremental direction cosines V_{si}^k and V_{ti}^k can be expressed in terms of the incremental rotations of the nodes θ_i^k about the stationary global cartesian axes:

$$ \begin{aligned} V_s^k &= \theta^k \times {}^t V_s^k \\ V_t^k &= \theta^k \times {}^t V_t^k \end{aligned} \tag{4.6} $$

The cross-products of Eqs. (4.6) are expressed in expanded form as

$$ \begin{bmatrix} V_{s1}^k \\ V_{s2}^k \\ V_{s3}^k \end{bmatrix} = \begin{bmatrix} 0 & {}^t V_{s3}^k & -{}^t V_{s2}^k \\ -{}^t V_{s3}^k & 0 & {}^t V_{s1}^k \\ {}^t V_{s2}^k & -{}^t V_{s1}^k & 0 \end{bmatrix} \begin{bmatrix} \theta_1^k \\ \theta_2^k \\ \theta_3^k \end{bmatrix} $$

$$ \begin{bmatrix} V_{t1}^k \\ V_{t2}^k \\ V_{t3}^k \end{bmatrix} = \begin{bmatrix} 0 & {}^t V_{t3}^k & -{}^t V_{t2}^k \\ -{}^t V_{t3}^k & 0 & {}^t V_{t1}^k \\ {}^t V_{t2}^k & -{}^t V_{t1}^k & 0 \end{bmatrix} \begin{bmatrix} \theta_1^k \\ \theta_2^k \\ \theta_3^k \end{bmatrix} \tag{4.7} $$

Equations (4.6) are valid only for infinitesimal rotations (Synge and Griffith, 1959). To compute the element matrices, which are instantaneous quantities valid for small displacements from the current equilibrium configuration, Eqs. (4.6) are used. However, having calculated the rotations in the finite element solution, the direction cosines of the new nodal vectors at time $t + \Delta t$ must be evaulated as accurately as possible in order to update the configuration. This requires computing the incremental unit vectors of Eqs. (4.6) from the following integral equations (Bathe, 1982):

$$\underline{V}_s^k = \int_{\theta^k} d\theta^k \times {}^{\tau}V_s^k$$

$$\underline{V}_t^k = \int_{\theta^k} d\theta^k \times {}^{\tau}V_t^k \qquad (4.8)$$

The unit vectors at time t + Δ t are:

$$
\begin{aligned}
{}^{t+\Delta t}V_s^k &= {}^{t}V_s^k + \underline{V}_s^k \\
{}^{t+\Delta t}V_t^k &= {}^{t}V_t^k + \underline{V}_t^k
\end{aligned}
$$

In Eqs. (4.8), τ represents a time increment between t and $t + \Delta t$. The integrals can be evaluated using Euler forward integration by setting $d\theta^k$ to be $\frac{1}{n}$ of θ^k (the fraction $\frac{1}{5}$ is used in this study (Kani, 1986)). For each increment of θ^k , the vector products are performed and the vectors ${}^{\tau}V_s^k, {}^{\tau}V_t^k$ are updated and normalized; the unit lengths of ${}^{\tau}V_s^k, {}^{\tau}V_t^k$ must be preserved in Eqs. (4.8). This operation is repeated n times to evaluate the direction cosines of the new nodal vectors, where one integration step (n = 1) corresponds to Eqs. (4.6).

The element matrices of the isobeam are derived using the updated Lagrangian (U.L.) formulation. Thus, the displacements relative to the current configuration at time t are evaluated by substituting Eqs. (4.7) into Eq. (4.5):

$$
\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \sum_{k=1}^{3} h_k \left\{ \begin{bmatrix} u_1^k \\ u_2^k \\ u_3^k \end{bmatrix} + \frac{sa}{2} \begin{bmatrix} 0 & {}^{t}V_{s3}^k & -{}^{t}V_{s2}^k \\ -{}^{t}V_{s3}^k & 0 & {}^{t}V_{s1}^k \\ -{}^{t}V_{s2}^k & -{}^{t}V_{s1}^k & 0 \end{bmatrix} \begin{bmatrix} \theta_1^k \\ \theta_2^k \\ \theta_3^k \end{bmatrix} + \frac{tb}{2} \begin{bmatrix} 0 & {}^{t}V_{t3}^k & -{}^{t}V_{t2}^k \\ {}^{t}V_{t3}^k & 0 & {}^{t}V_{t1}^k \\ {}^{t}V_{t2}^k & -{}^{t}V_{t1}^k & 0 \end{bmatrix} \begin{bmatrix} \theta_1^k \\ \theta_2^k \\ \theta_3^k \end{bmatrix} \right\} \qquad (4.9)
$$

The displacement derivatives are computed directly from Eq. (4.9). For example, u_1, r is written as follows:

$$u_1,_r = \sum_{k=1}^{3} h_k,_r \left[1 \quad 0 \quad \frac{1}{2}(as\,{}^tV^k_{s3} + bt \quad {}^tV^k_{t3}) \quad -\frac{1}{2}(as\,{}^tV^k_{s2} + bt \quad {}^tV^k_{t2}) \right] \begin{bmatrix} u_1^k \\ \theta_1^k \\ \theta_2^k \\ \theta_3^k \end{bmatrix} \tag{4.10}$$

The following three expressions are defined for convenience.

$${}^t(\hat{g})^k = \frac{1}{2}a \begin{bmatrix} 0 & -{}^tV^k_{s3} & {}^tV^k_{s2} \\ {}^tV^k_{s3} & 0 & -{}^tV^k_{s1} \\ -{}^tV^k_{s2} & {}^tV^k_{s1} & 0 \end{bmatrix} \tag{4.11}$$

$${}^t(\bar{g})^k = \frac{1}{2}b \begin{bmatrix} 0 & -{}^tV^k_{t3} & {}^tV^k_{t2} \\ {}^tV^k_{s3} & 0 & -{}^tV^k_{s1} \\ -{}^tV^k_{t2} & {}^tV^k_{t1} & 0 \end{bmatrix} \tag{4.12}$$

$${}^t(g)^k_{ij} = s \; {}^t(\hat{g})^k_{ij} + t \; (\bar{g})^k_{ij} \tag{4.13}$$

Using relations (4.11 - 4.13), the displacement derivatives in natural coordinates r, s, t are

$$\begin{bmatrix} u_{i,r} \\ u_{i,s} \\ u_{i,t} \end{bmatrix} = \sum_{k=1}^{3} M_i^k \begin{bmatrix} u_i^k \\ \theta_1^k \\ \theta_2^k \\ \theta_3^k \end{bmatrix} \quad (i = 1,2,3) \tag{4.14}$$

where M_i^k is a 3x4 matrix corresponding to node k:

$$M_i^k = \begin{bmatrix} h_{k,r}[\,1 \;\; {}^t(g)^k_{1i} \;\; {}^t(g)^k_{2i} \;\; {}^t(g)^k_{3i}\,] \\ h_k \;[\,0 \;\; {}^t(\hat{g})^k_{1i} \;\; {}^t(\hat{g})^k_{2i} \;\; {}^t(\hat{g})^k_{3i}\,] \\ h_k \;[\,0 \;\; {}^t(\bar{g})^k_{1i} \;\; {}^t(\bar{g})^k_{2i} \;\; {}^t(\bar{g})^k_{3i}\,] \end{bmatrix} \tag{4.15}$$

The displacement derivatives in cartesian coordinates ${}^tx_i$, i = 1,2,3 are obtained through the Jacobian transformation

$$\frac{\partial}{\partial^t x} = {}^tJ^{-1}\frac{\partial}{\partial r} \tag{4.16}$$

where

$${}^tJ = \begin{bmatrix} x_{1,r} & x_{2,r} & x_{3,r} \\ x_{1,s} & x_{2,s} & x_{3,s} \\ x_{1,t} & x_{2,t} & x_{3,t} \end{bmatrix} \tag{4.17}$$

The derivatives of the current coordinates ${}^tx_i$, i = 1,2,3 with respect to the natural coordinates r,s,t are evaluated from Eq. (4.1), and the Jacobian is

$${}^tJ = \sum_{k=1}^{3} \begin{bmatrix} \frac{\partial h_k}{\partial r} \; [({}^tx_1^k + \frac{sa}{2}\,{}^tV_{s1}^k + \frac{\mathrm{tb}}{2}\,{}^tV_{t1}^k) & ({}^tx_2^k + \frac{sa}{2}\,{}^tV_{s2}^k + \frac{\mathrm{tb}}{2}\,{}^tV_{t2}^k) & ({}^tx_3^k + \frac{sa}{2}\,{}^tV_{s3}^k + \frac{\mathrm{tb}}{2}\,{}^tV_{t3}^k)] \\ h_k[\; \frac{a}{2}\,{}^tV_{s1}^k & \frac{a}{2}\,{}^tV_{s2}^k & \frac{a}{2}\,{}^tV_{s3}^k \quad] \\ h_k[\; \frac{b}{2}\,{}^tV_{t1}^k & \frac{b}{2}\,{}^tV_{t2}^k & \frac{b}{2}\,{}^tV_{t3}^k \quad] \end{bmatrix} \tag{4.18}$$

The inverse of the Jacobian can be expressed as the adjoint matrix (matrix of cofactors) over the determinant of ${}^tJ$

$${}^tJ^{-1} = \frac{\mathrm{Adj}({}^tJ)}{|{}^tJ|} \tag{4.19}$$

Using Eqs. (4.14) and (4.19), we obtain the displacement derivatives in coordinates ${}^tx_i$.

$$\begin{bmatrix} u_{i,1} \\ u_{i,2} \\ u_{i,3} \end{bmatrix} = \sum_{k=1}^{3} {}^{t}J^{-1}M_i^k \begin{bmatrix} u_i^k \\ \theta_1^k \\ \theta_2^k \\ \theta_3^k \end{bmatrix} \qquad i = 1,2,3, \tag{4.20}$$

where ${}^{t}J^{-1}M_i^k$ is a 3x4 matrix corresponding to node k:

$${}^{t}J^{-1}M_i^k = \begin{bmatrix} {}_{t}h_{k,1} & {}_{t}(G1)_{i1}^k & {}_{t}(G2)_{i1}^k & {}_{t}(G3)_{i1}^k \\ {}_{t}h_{k,2} & {}^{t}(G1)_{i2}^k & {}_{t}(G2)_{i2}^k & {}_{t}(G3)_{i2}^k \\ {}_{t}h_{k,3} & {}_{t}(G1)_{i3}^k & {}_{t}(G2)_{i3}^k & {}_{t}(G3)_{i3}^k \end{bmatrix} \tag{4.21}$$

and

$${}_{t}h_{k,l} = {}^{t}J_{l1}^{-1} h_{k,r}$$

$${}_{t}(Gm)_{ln}^k = ({}^{t}J_{n1}^{-1}\ {}^{t}(g)_{ml}^k)h_{k,r} + ({}^{t}J_{n2}^{-1}\ {}^{t}(\hat{g})_{ml}^k + {}^{t}J_{n3}^{-1}\ {}^{t}(\bar{g})_{ml}^k)h_k$$

The required finite element matrices of the isobeam element corresponding to the U.L. formulation can be derived from Eq. (4.20) and the continuum mechanics equation of equilibrium presented in the next section.

4.2.2 Summary of the Finite Element Matrices in the U.L. Formulation

The linearized governing continuum mechanics equation for the U.L. formulation was introduced in Chapter 2 and it is repeated here for convenience

$$\underbrace{\int_{{}^{t}V} {}_{t}C_{ijrs}\ {}_{t}e_{rs}\ \delta e_{ij}\ {}^{t}d\ V + \int_{{}^{t}V} {}^{t}\tau_{ij}\ \delta_{t}\eta_{ij}\ {}^{t}dV}_{\delta \underline{u}^{T}({}^{t}_{t}\underline{K}_{L} + {}^{t}_{t}\underline{K}_{NL})\Delta \underline{u}}$$

$$= \underbrace{{}^{t+\Delta t}R - \int_{{}^{t}V} {}^{t}\tau_{ij}\ \delta_{t}e_{ij}\ {}^{t}dV}_{\delta \underline{u}^{T}({}^{t+\Delta t}\underline{R} - {}^{t}_{t}\underline{F})} \tag{4.22}$$

where,

${}^{t}_{t}\underline{K}_{L}$ linear strain stiffness matrix

${}^{t}_{t}\underline{K}_{NL}$ nonlinear strain (geometric) stiffness matrix

${}^{t+\Delta t}\underline{R}$ vector of externally applied nodal forces at time t + Δ t

${}^{t}_{t}\underline{F}$ vector of internal nodal forces corresponding to the element stresses at time t.

Considering an incremental displacement u_i, the terms of Eq. (4.22) are expressed in matrix form in Table 4.1. To evaluate ${}^{t}_{t}K_{L,}$ ${}^{t}_{t}K_{NL}$, and ${}^{t}_{t}\underline{F}$, we need to formulate ${}^{t}_{t}\underline{B}_{L}$, ${}^{t}_{t}\underline{B}_{NL}$, ${}^{t}\underline{C}$, ${}^{t}\underline{\tau}$, and ${}^{t}\underline{\hat{\tau}}$, as listed in Table 4.2.

Table 4.1 Governing Equation in the U.L. Formulation

Continuum Mechanics Expression	Matrix Evaluation
$\int_{{}^{t}V} {}_{t}C_{ijrs}\, {}_{t}e_{rs}\, \delta\, {}_{t}e_{ij}\, {}^{t}dV$	$\delta\, \hat{\underline{u}}^{T} \underbrace{\left(\int_{{}^{t}V} {}^{t}_{t}\underline{B}_{L}^{T}\, {}_{t}\underline{C}\, {}^{t}_{t}\underline{B}_{L}\, {}^{t}dV\right)}_{{}^{t}_{t}\underline{K}_{L}} \hat{\underline{u}}$
${}^{t}\underline{e} = {}^{t}_{t}\underline{B}_{L}\hat{\underline{u}}$	vector containing the linear strain increments ${}_{t}e_{ij}$
${}^{t}_{t}\underline{B}_{L}$	linear strain-displacement matrix
${}_{t}\underline{C}$	incremental stress-strain material property matrix
$\hat{\underline{u}}$	incremental nodal displacement vector
$\int_{{}^{t}V} {}^{t}\tau_{ij}\, \delta\, {}_{t}\eta_{ij}\, {}^{t}dV$	$\delta\, \hat{u}^{T} \underbrace{\left(\int_{{}^{t}V} {}^{t}_{t}\underline{B}_{NL}^{T}\, {}^{t}\underline{\tau}\, {}^{t}_{t}\underline{B}_{NL}\, {}^{t}dV\right)}_{{}^{t}_{t}\underline{K}_{NL}} \hat{\underline{u}}$
${}^{t}\underline{\tau}$	matrix containing the cauchy stresses τ_{ij}
${}_{t}\underline{\eta}$	vector containing the nonlinear strain increments ${}_{t}\eta_{ij}$
${}^{t}_{t}\underline{B}_{NL}$	nonlinear strain-displacement matrix
$\int_{{}^{t}V} {}^{t}\tau_{ij}\, \delta\, {}_{t}e_{ij}\, {}^{t}dV$	$\delta\, \hat{\underline{u}}^{T} \underbrace{\left(\int_{{}^{t}V} {}^{t}_{t}\underline{B}_{L}^{T}\, {}^{t}\hat{\underline{\tau}}\, {}^{t}dV\right)}_{{}^{t}_{t}\underline{F}}$
${}^{t}\hat{\underline{\tau}}$	vector containing the cauchy stresses τ_{ij}

Table 4.2 Required Element Matrices

Evaluation of	Requires
${}^{t}_{t}\underline{K}_{L}$	${}^{t}\underline{C}$, ${}^{t}_{t}\underline{B}_{L}$
${}^{t}_{t}\underline{K}_{NL}$	${}^{t}\underline{\tau}$, ${}^{t}_{t}\underline{B}_{NL}$
${}^{t}_{t}\underline{F}$	${}^{t}\hat{\underline{\tau}}$, ${}^{t}_{t}\underline{B}_{L}$

4.2.3 Linear Strain-Displacement Transformation Matrix

The linear strain-displacement matrix ${}^t_t\underline{B}_L$ defines the linear strain increment ${}^t\underline{\varepsilon}$ as a function of the incremental nodal displacements $\hat{\underline{u}}$:

$${}^t\underline{\varepsilon} = {}^t_t\underline{B}_L\hat{\underline{u}}$$

where the linear strain increment is defined as

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

and the displacement derivatives are obtained from Eq. (4.20). For a nodal point k, the linear strain-displacement matrix is

$$\begin{bmatrix} e_{11} \\ e_{22} \\ e_{33} \\ 2e_{12} \\ 2e_{23} \\ 2e_{13} \end{bmatrix} = \underbrace{\begin{bmatrix} {}_th_{k,1} & 0 & 0 & {}_t(G1)^k_{11} & {}_t(G2)^k_{11} & {}_t(G3)^k_{11} \\ 0 & {}_th_{k,2} & 0 & {}_t(G1)^k_{22} & {}_t(G2)^k_{22} & {}_t(G3)^k_{22} \\ 0 & 0 & {}_th_{k,3} & {}_t(G1)^k_{33} & {}_t(G2)^k_{33} & {}_t(G3)^k_{33} \\ {}_th_{k,2} & {}_th_{k,1} & 0 & {}_t(G1)^k_{12} + {}^t(G1)^k_{21} & {}_t(G2)^k_{12} + {}_t(G2)^k_{21} & {}_t(G3)^k_{12} + {}_t(G3)^k_{21} \\ 0 & {}_th_{k,3} & {}_th_{k,2} & {}_t(G1)^k_{23} + {}_t(G1)^k_{32} & {}_t(G2)^k_{23} + {}_t(G2)^k_{32} & {}_t(G3)^k_{23} + {}_t(G3)^k_{32} \\ {}_th_{k,3} & 0 & {}_th_{k,1} & {}_t(G1)^k_{13} + {}_t(G1)^k_{31} & {}_t(G2)^k_{13} + {}_t(G2)^k_{31} & {}_t(G3)^k_{13} + {}_t(G3)^k_{31} \end{bmatrix}}_{{}^t_t\underline{B}^k_L} \begin{bmatrix} u^k_1 \\ u^k_2 \\ u^k_3 \\ \theta^k_1 \\ \theta^k_2 \\ \theta^k_3 \end{bmatrix} \tag{4.23}$$

The strain components of Eq. (4.23) correspond to global axes and are transformed to the beam local axes η, ξ, ζ by means of the transformation matrix ${}^t\underline{T}$. The strain-displacement matrix in local coordinates relates strain increments in local directions to displacement increments in global directions, and it is defined by

$${}^t_t\overline{\underline{B}}_L = {}^t\underline{T}\,{}^t_t\underline{B}_L \tag{4.24}$$

The linear strain increment in local coordinates can now be expressed as

$$
{}^t\bar{\underline{\varepsilon}} = \begin{bmatrix} {}^te_{\eta\eta} \\ 2\,{}^te_{\eta\xi} \\ 2\,{}^te_{\eta\zeta} \end{bmatrix} = {}^t_t\bar{\underline{B}}_L\ \hat{\underline{u}} \tag{4.25}
$$

The transformation matrix ${}^t\underline{T}$ can be obtained by transformation of displacement derivatives (Cook, 1981) or by tensor transformations of strains (Frederick and Chang, 1972). The matrix ${}^t\underline{T}$ can be expressed as a function of the unit vectors at any point of the beam axis:

$$
{}^t\underline{T} = \begin{bmatrix} {}^tV_{r1}^1 & {}^tV_{r2}^2 & {}^tV_{r3}^3 & {}^tV_{r1}\,{}^tV_{r2} & {}^tV_{r2}\,{}^tV_{r3} & {}^tV_{r1}\,{}^tV_{r3} \\ 2\,{}^tV_{r1}\,{}^tV_{s1} & 2\,{}^tV_{r2}\,{}^tV_{s2} & 2\,{}^tV_{r3}\,{}^tV_{s3} & {}^tV_{r2}\,{}^tV_{s1} + {}^tV_{r1}\,{}^tV_{s2} & {}^tV_{r3}\,{}^tV_{s2} + {}^tV_{r2}\,{}^tV_{s3} & {}^tV_{r3}\,{}^tV_{s1} + {}^tV_{r1}\,{}^tV_{s3} \\ 2\,{}^tV_{r1}\,{}^tV_{t1} & 2\,{}^tV_{r2}\,{}^tV_{t2} & 2\,{}^tV_{r3}\,{}^tV_{t3} & {}^tV_{r2}\,{}^tV_{t1} + {}^tV_{r1}\,{}^tV_{t2} & {}^tV_{r3}\,{}^tV_{t2} + {}^tV_{r2}\,{}^tV_{t3} & {}^tV_{r3}\,{}^tV_{t1} + {}^tV_{r1}\,{}^tV_{t3} \end{bmatrix} \tag{4.26}
$$

where

$$
{}^t\underline{V}_r = {}^t\underline{V}_s \times {}^t\underline{V}_t
$$

The unit vectors at any point along the beam element axis can be computed by interpolating the updated nodal unit vectors as follows (Cook, 1981, p. 282):

$$
{}^{t+\Delta t}\underline{V}_s = \sum_{k=1}^{3} h_k\,{}^{t+\Delta t}\underline{V}_s^k
$$

$$
{}^{t+\Delta t}\underline{V}_t = \sum_{k=1}^{3} h_k\,{}^{t+\Delta t}\underline{V}_t^k \tag{4.27}
$$

4.2.4 Nonlinear Strain-Displacement Transformation Matrix

According to Table 4.1, we construct ${}_t\underline{B}_{NL}$ and ${}^t\underline{\tau}$ so that

$$\delta\hat{\underline{u}}^T\ {}^t_t\underline{B}^T_{NL}\ {}^t\underline{\tau}\ {}^t_t\underline{B}_{NL}\ \hat{\underline{u}} = {}^t\tau_{ij}\ \delta_t\eta_{ij} \tag{4.28}$$

Consider the nonlinear strain increment

$${}_t\eta_{ij} = \frac{1}{2}\,{}_tu_{k,i}\ {}_tu_{k,j}$$

As an illustration, the component η_{11} and its first variation $\delta\eta_{11}$ are examined:

$${}_t\eta_{11} = \frac{1}{2}\left(({}_tu_{1,1})^2 + ({}_tu_{2,1})^2 + ({}_tu_{3,1})^2\right)$$

$$\delta_t\eta_{11} = \left[\delta_tu_{1,1}\quad \delta_tu_{2,1}\quad \delta^tu_{3,1}\right]\begin{bmatrix} {}_tu_{1,1} \\ {}_tu_{2,1} \\ {}_tu_{3,1} \end{bmatrix} \tag{4.29}$$

For a nodal point k, we write Eq. (4.29) in terms of the displacement derivatives of Eq. (4.20) as

$$\delta_t\eta_{11} = \delta\hat{\underline{u}}^T\ \underline{A}^T\ \underline{A}\ \hat{\underline{u}} \tag{4.30}$$

where

$$\underline{A} = \begin{bmatrix} {}_th_{k,1} & 0 & 0 & {}_t(G1)^k_{11} & {}_t(G2)^k_{11} & {}_t(G3)^k_{11} \\ 0 & {}_th_{k,1} & 0 & {}_t(G1)^k_{21} & {}_t(G2)^k_{21} & {}_t(G3)^k_{21} \\ 0 & 0 & {}_th_{k,1} & {}_t(G1)^k_{31} & {}_t(G2)^k_{31} & {}_t(G3)^k_{31} \end{bmatrix} \tag{4.31}$$

The rows of matrix A correspond to rows of 1, 4 and 7 of the nonlinear strain-displacement transformation matrix ${}^t_tB_{NL}$. Similarly, by computing $\delta_t\eta_{22}$ and $\delta_t\eta_{33}$, the remaining rows of ${}^t_tB_{NL}$ (rows 2, 5, 8 and 3, 6, 9) are constructed, and it is defined for a node k as

$$
{}^t_tB^k_{NL} = \begin{bmatrix}
{}_th_{k,1} & 0 & 0 & {}_t(G1)^k_{11} & {}_t(G2)^k_{11} & {}_t(G3)^k_{11} \\
{}_th_{k,2} & 0 & 0 & {}_t(G1)^k_{12} & {}_t(G2)^k_{12} & {}_t(G3)^k_{12} \\
{}_th_{k,3} & 0 & 0 & {}_t(G1)^k_{13} & {}_t(G2)^k_{13} & {}_t(G3)^k_{13} \\
0 & {}_th_{k,1} & 0 & {}_t(G1)^k_{21} & {}_t(G2)^k_{21} & {}_t(G3)^k_{21} \\
0 & {}_th_{k,2} & 0 & {}_t(G1)^k_{22} & {}_t(G2)^k_{22} & {}_t(G3)^k_{22} \\
0 & {}_th_{k,3} & 0 & {}_t(G1)^k_{23} & {}_t(G2)^k_{23} & {}_t(G3)^k_{23} \\
0 & 0 & {}_th_{k,1} & {}_t(G1)^k_{31} & {}_t(G2)^k_{31} & {}_t(G3)^k_{31} \\
0 & 0 & {}_th_{k,2} & {}_t(G1)^k_{32} & {}_t(G2)^k_{32} & {}_t(G3)^k_{32} \\
0 & 0 & {}_th_{k,3} & {}_t(G1)^k_{33} & {}_t(G2)^k_{33} & {}_t(G3)^k_{33}
\end{bmatrix} \qquad (4.32)
$$

The Cauchy stresses ${}^t\tau_{ij}$ in global coordinates are expressed as

$$
{}^t\underline{\tau} = \begin{bmatrix}
{}^t\underline{\tau}^* & & \\
 & {}^t\underline{\tau}^* & \\
 & & {}^t\underline{\tau}^*
\end{bmatrix} \qquad (4.33)
$$

where

$$
{}^t\underline{\tau}^* = \begin{bmatrix}
\tau_{11} & \tau_{12} & \tau_{13} \\
\tau_{21} & \tau_{22} & \tau_{23} \\
\tau_{31} & \tau_{32} & \tau_{33}
\end{bmatrix}
$$

The correct construction of ${}^t_tB_{NL}$ can now be checked by expanding the left-hand side of Eq. (4.28) and comparing the result of the matrix products to the right-hand side of the equation. The tensor product of Eq. (4.28) can be written as

$$\begin{aligned} & {}^t\tau_{11}\delta_t\eta_{11} + {}^t\tau_{12}\delta_t\eta_{12} + {}^t\tau_{13}\delta_t\eta_{13} \\ {}^t\tau_{ij}\delta_t\eta_{ij} = & + {}^t\tau_{21}\delta_t\eta_{21} + {}^t\tau_{22}\delta_t\eta_{22} + {}^t\tau_{23}\delta_t\eta_{23} \\ & + {}^t\tau_{31}\delta_t\eta_{31} + {}^t\tau_{32}\delta_t\eta_{32} + {}^t\tau_{33}\delta_t\eta_{33} \end{aligned} \tag{4.34}$$

Although the derivation of ${}^t_tB_{NL}$ did not make use of the components of $\delta\eta_{ij}$ for $i \neq j$, they are automatically included in the matrix product of Eq. (4.28), and the nine terms of Eq. (4.34) are satisfied. This is characteristic of the isoparametric formulation of solid and structural elements which use Lagrangian interpolation functions (Bathe, 1986).

In the local coordinates η, ξ, ζ, only three components of the stress tensors exist. Thus, the Cauchy stress matrix of Eq. (4.33) can be expressed as

$${}^t\underline{\tau} = \begin{bmatrix} {}^t\tilde{\tau} & & \\ & {}^t\tilde{\tau} & \\ & & {}^t\tilde{\tau} \end{bmatrix} \tag{4.35}$$

where

$${}^t\tilde{\underline{\tau}} = \begin{bmatrix} {}^t\tau_{\eta\eta} & {}^t\tau_{\eta\xi} & {}^t\tau_{\eta\zeta} \\ {}^t\tau_{\eta\xi} & 0 & 0 \\ {}^t\tau_{\eta\zeta} & 0 & 0 \end{bmatrix}$$

The matrix ${}^t_tB_{NL}$ can now be transformed to the local beam axis through a transformation matrix $\tilde{T}$. The matrix $\tilde{T}$ is constructed from Eq. (4.26) such that

$$\delta\hat{u}^T \cdot \underbrace{{}^t_tB_{\text{NL}}^T \cdot {}^t\tilde{T}}_{{}^t_t\bar{B}_{\text{NL}}^T} \cdot {}^t\tilde{\tau} \cdot \underbrace{{}^t\tilde{T}\; B_{\text{NL}}}_{{}^t_t\bar{B}_{\text{NL}}} \cdot \hat{u} = {}^t\tilde{\tau}_{ij} \cdot \delta_t\bar{\eta}_{ij} \tag{4.36}$$

where

$$\tilde{T} = \begin{bmatrix} T^* & & \\ & T^* & \\ & & T^* \end{bmatrix} \tag{4.37}$$

$$T^* = \begin{bmatrix} {}^tV_{r1} & {}^tV_{s1} & {}^tV_{t1} \\ {}^tV_{r2} & {}^tV_{s2} & {}^tV_{t2} \\ {}^tV_{r3} & {}^tV_{s3} & {}^tV_{t3} \end{bmatrix}; \quad {}^t\underline{V}_r = {}^t\underline{V}_s \times {}^t\underline{V}_t$$

Moreover, from Table 4.1 the Cauchy stress vector ${}^t\hat{\underline{\tau}}$ is constructed to satisfy the condition

$$\delta\hat{\underline{u}}^T \; {}^t_t\underline{B}_L^T \; {}^t\hat{\underline{\tau}} = {}^t\tau_{ij} \; \delta_t e_{ij}$$

and it is of the following form

$${}^t\hat{\underline{\tau}} = \begin{bmatrix} {}^t\tau_{\eta\eta} \\ {}^t\tau_{\eta\xi} \\ {}^t\tau_{\eta\zeta} \end{bmatrix} \tag{4.38}$$

Finally, the constitutive matrix for linearly elastic material behavior (Eq. 3.4) is simply

$$\bar{C} = \begin{bmatrix} E & 0 & 0 \\ 0 & \frac{1}{\kappa}G & 0 \\ 0 & 0 & \frac{1}{\kappa}G \end{bmatrix}$$

The formulation of the 3-d isobeam is now complete, and the evaluation of the stiffness matrix and the development of the computer program are discussed in Chapter 5.

4.3 Solution Techniques in Nonlinear Finite Element Analysis

The basic problem in a general nonlinear finite element analysis is to trace a nonlinear equilibrium path to critical points and occasionally through critical points. Thus, we need to establish a series of discrete equilibrium points along the equilibrium path. The nonlinear response of a structure may be solved for incremental iterative nodal displacements by taking a series of linear steps. Of the various numerical solution schemes that have been proposed, the Newton-Ralphson and the Riks-Wempner methods are the most widely used. The Newton-Ralphson method and its modifications are included in most commercial programs to trace nonlinear prebuckling paths of structures; however, it cannot trace the response beyond a limit point. Its popularity rests in its simplicity and its efficiency for low convergence tolerance. The Riks-Wempner method has been proposed especially to overcome the difficulty experienced with the Newton-Ralphson method in tracing post-buckling responses. Both of these methods are described next.

4.3.1 Newton-Ralphson Method

The equation of equilibrium for the U.L. formulation was derived through the principle of virtual work, and upon linearization, this equation was expressed as (Eq. 4.22)

$$ {}_{t}^{t}K\Delta u = {}^{t+\Delta t}R - {}_{t}^{t}F \tag{4.39}$$

The right-hand side of this equation is the unbalanced force vector corresponding to a trial configuration, and it is defined by

$$ f = {}^{t+\Delta t}R - {}_{t}^{t}F \tag{4.40}$$

Equation (4.39) is solved through an iterative process. The basic approach is to assume that the solution for time t is known, and that the solution for the discrete time $t+\Delta t$ is required. Thus, the following iterative equation can be written

$$ {}^{t+\Delta t}K^{(i-1)}\ \Delta u^{(i)} = f^{(i-1)} \qquad i = 1,2,3 \ldots ,n \tag{4.41} $$

where

$$ f^{(i-1)} = {}^{t+\Delta t}R - {}^{t+\Delta t}F^{(i-1)} \tag{4.42} $$

$$ {}^{t+\Delta t}u^{(i)} = {}^{t+\Delta t}u^{(i-1)} + \Delta u^{(i)} \tag{4.43} $$

with initial conditions (for i = 1)

$$ \begin{aligned} {}^{t+\Delta t}K^{(0)} &= {}^{t}K \\ {}^{t+\Delta t}F^{(0)} &= {}^{t}F \\ {}^{t+\Delta t}U^{(0)} &= {}^{t}U \end{aligned} $$

The procedure, illustrated for a one-degree-of-freedom system in Fig. 4.3, is as follows: for a given trial configuration $u^{(i-1)}$

1. Establish the tangent stiffness matrix $K^{(i-1)}$

2. Evaluate the equilibrium nodal force vector $F^{(i-1)}$.

3. Compute the vector of unbalanced nodal force $f^{(i-1)}$ from Eq. (4.42).

4. Solve Eq. (4.41) for the vector of incremental nodal displacements $\Delta u^{(i)}$.

5. Update the vector of nodal displacements by Eq. (4.43).

6. Test for convergence (Scction 4.4).

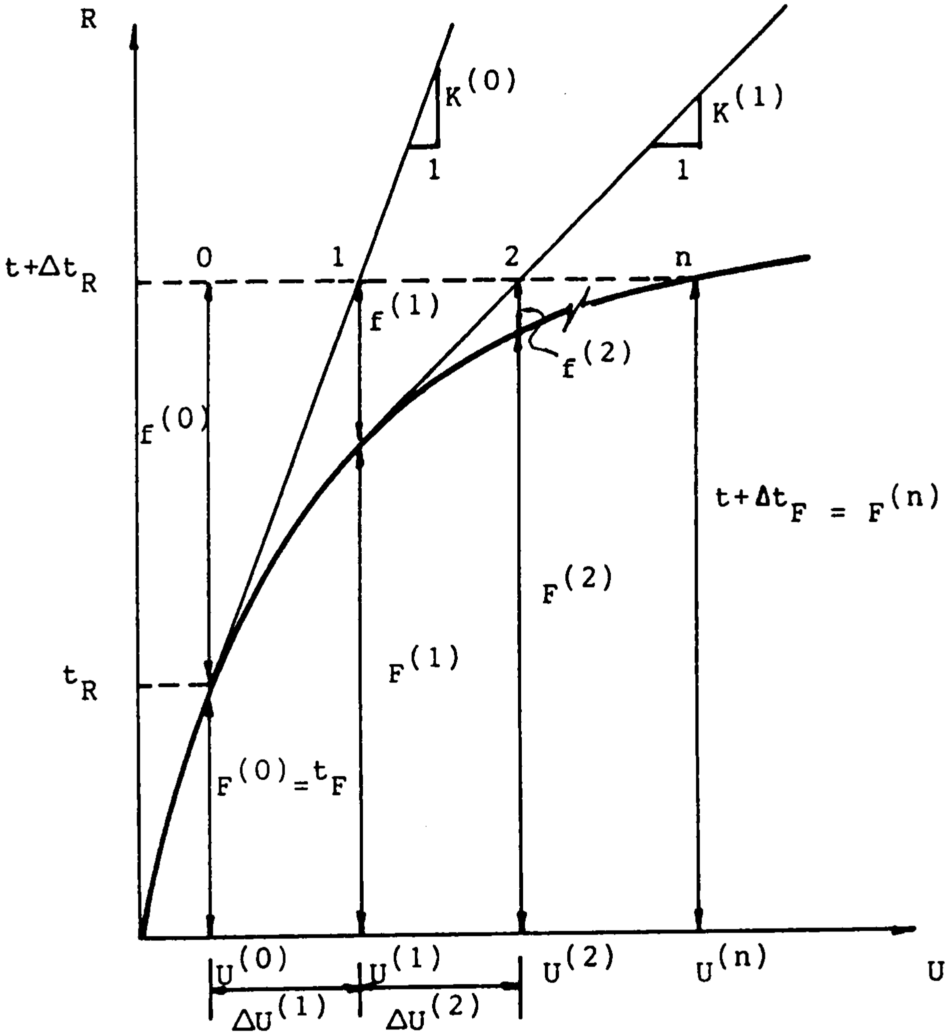


Figure 4.3. Graphical representation of the Newton-Ralphson method.

7. If the process has not converged, return to Step 1. Otherwise, increment the load vector and seek a new solution.

The Newton-Ralphson method requires the stiffness matrix to be updated after each iteration. This can result in high computational cost due to a large number of factorizations of the stiffness matrix. For large systems, especially when the degree of nonlinearity is not very high, it may be more economical to update the stiffness matrix periodically after a certain number of iterations (Bathe and Cimento, 1980). This modified approach will generally require an increased number of iterations for convergence, but the additional cost for extra iterations may be offset by the reduction in the factorization process. The Modified Newton-Ralphson method is presented graphically in Fig. 4.4, and the computational procedure (Fig. 4.5) follows the same steps outlined above, except that ${}^t_t K^{(i-1)} = {}^t K$.

The Newton-Ralphson method cannot trace turning points on the equilibrium path. In the neighborhood of a limit point, the load incrementation procedure does not allow passing through the limit point. As a result, the solution diverges. To solve this problem, the Modified Riks-Wempner method is recommended.

4.3.2 Modified Riks-Wempner Method

The detailed theoretical development of this method is presented by Holzer et al. (1981), and it is included in finite element programs developed by Holzer and his students (Butler, 1983; Jau, 1985; Sage, 1986; Hao, 1986; Ahmed, 1988). Therefore, the details of the method are omitted here.

The modified Riks-Wempner method is based on the Newton-Ralphson method, but differs in the manner in which it traces the equilibrium path. While the Newton-Ralphson method uses a load increment to establish an equilibrium point along the equilibrium path, the Riks-Wempner

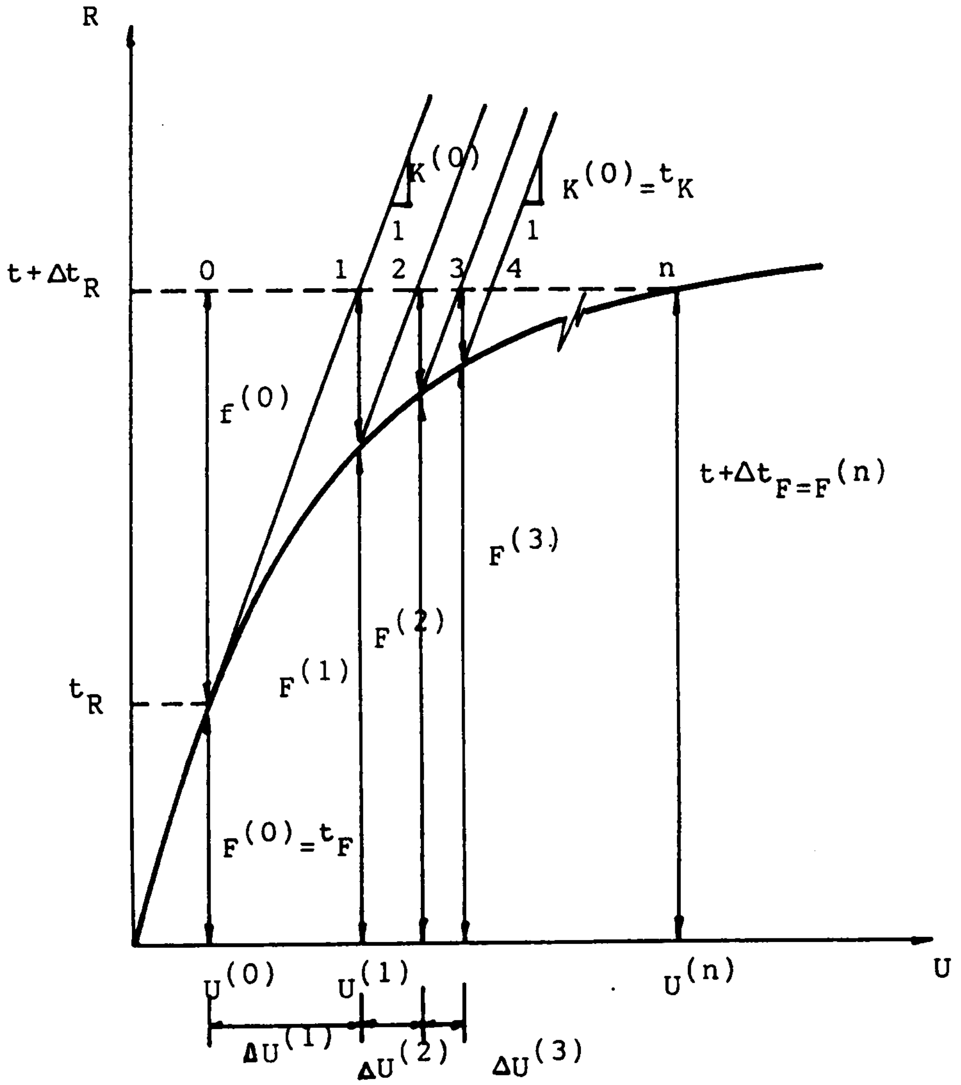


Figure 4.4. Graphical representation of the modified Newton-Ralphson method.

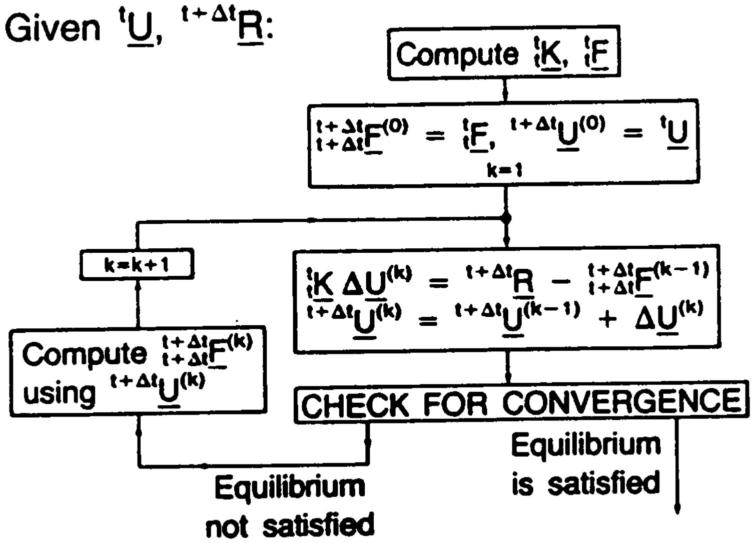

Given ${}^{t}\underline{U}$, ${}^{t+\Delta t}\underline{R}$:
Compute ${}^{t}_{t}\underline{K}$, ${}^{t}_{t}\underline{F}$
${}^{t+\Delta t}_{t+\Delta t}\underline{F}^{(0)} = {}^{t}_{t}\underline{F}$, ${}^{t+\Delta t}\underline{U}^{(0)} = {}^{t}\underline{U}$
k=1
k=k+1
${}^{t}_{t}\underline{K}\,\Delta\underline{U}^{(k)} = {}^{t+\Delta t}\underline{R} - {}^{t+\Delta t}_{t+\Delta t}\underline{F}^{(k-1)}$
${}^{t+\Delta t}\underline{U}^{(k)} = {}^{t+\Delta t}\underline{U}^{(k-1)} + \Delta\underline{U}^{(k)}$
Compute ${}^{t+\Delta t}_{t+\Delta t}\underline{F}^{(k)}$ using ${}^{t+\Delta t}\underline{U}^{(k)}$
CHECK FOR CONVERGENCE
Equilibrium not satisfied
Equilibrium is satisfied

Figure 4.5. Computational procedure of the modified Newton-Ralphson method (Bathe, 1986).

method uses a generalized arc length Δs along the tangent to an equilibrium point to facilitate the procedure of seeking a new solution (Fig. 4.6). Then, the iteration path follows a normal plane to the tangent, and the new equilibrium point will be the intersection of the normal plane with the equilibrium path. In this study, both the modified Newton-Ralphson and the modified Riks-Wempner methods are implemented in the computer program described in Chapter 5.

4.4 Convergence Criteria

In the incremental iterative solution processes described in the previous section, we can get arbitrarily close to the exact solution. The question is, when is the trial solution sufficiently accurate to terminate the process? To define a desired accuracy of the solution, we establish error tolerances. In general, there are three categories of convergence criteria which can be considered (Bathe, 1982):

1. Allowable error in displacements.

2. Allowable error in the residual (out-of-balance) force ${}^{t+\Delta t}R - {}^{t+\Delta t}F^{(i)}$.

3. Error tolerance in the total energy level of the system.

The Euclidean vector norm is used to evaluate the error level during a given iteration, which is defined as

$$||\underline{a}||_2 = \sqrt{\sum_{k=1}^{n} (a_k)^2}$$

The error criterion based on displacements is expressed as the ratio of the norms of the incremental displacement vector for the ith iteration to the exact displacement vector at the time $t + \Delta t$:

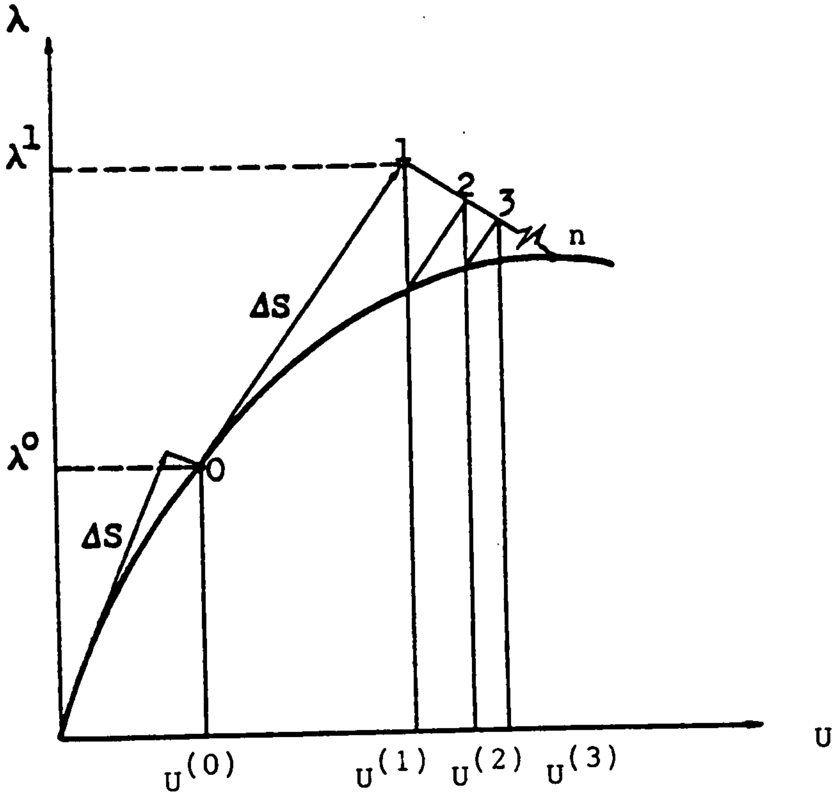


Figure 4.6. Modified Riks-Wempner method: iteration along a normal plane.

$$\frac{||\Delta u^{(i)}||_2}{||{}^{t+\Delta t}u||_2} \leq \varepsilon_D \tag{4.44}$$

The displacements ${}^{t+\Delta t}u$ are not known beforehand. Thus, Bathe (1982) suggests using the current approximation ${}^{t+\Delta t}u^{(i)}$ instead. Cook (1981) suggests a value for ε_D between 10^{-6} and 10^{-2}.

The second criterion specifying an error tolerance in the residual force can be expressed as

$$\frac{||{}^{t+\Delta t}R - {}^{t+\Delta t}F^{(i)}||_2}{||{}^{t+\Delta t}R - {}^{t}F||_2} \leq \varepsilon_F \tag{4.45}$$

where ${}^{t+\Delta t}F^{(i)}$ is the internal force vector for the ith iteration. The suggested value for ε_F is 10^{-3} (Bathe, 1986).

The third criterion is based on the increment in internal energy of the system between trial configurations. The amount of work done by the residual forces is compared to the initial increment in the internal energy (Bathe, 1982). This can be expressed as

$$\frac{\Delta u^{(i)T}({}^{t+\Delta t}R - {}^{t+\Delta t}F^{(i-1)})}{\Delta u^{(i)T}({}^{t+\Delta t}R - {}^{t}F)} \leq \varepsilon_E \tag{4.46}$$

Bathe (1982) recommends a value of $10 \times \varepsilon_D \times \varepsilon_F$.

It is recommended that the error tolerances in displacements (Eq. 4.44) and residual forces (Eq. 4.45) be used simultaneously. Alternatively, the energy criterion alone (Eq. 4.46) should guarantee sufficient accuracy since it includes error controls in displacements and forces (Bathe, 1982). The program described in Chapter 5 allows the user to specify any combination of the above error criteria.

Chapter 5

PROGRAM DEVELOPMENT, TESTING, AND EXPERIMENTAL VERIFICATION OF THE ISOBEAM

5.1 Introduction

Based on important considerations and assumptions for the modeling of glulam space beams (Ch. 3), a 3-d nonlinear isobeam element was formulated (Ch. 4), and in this chapter, the structure of the computer program and testing and experimental verification of the element are presented. To reduce the computational effort in the evaluation of the element stiffness matrix, the matrices that define the linear (K_L) and nonlinear (K_{NL}) components of the stiffness matrix are multiplied and expressed in tensor form before numerical integration is applied (Section 5.2). The numerical integration of the isobeam element requires a 3-point Gauss quadrature along the beam axis. However, by using a 2-point integration rule, the performance of the isobeam element is improved considerably: the computational cost is reduced, shear and membrane locking are eliminated (i.e., the element can model thin and curved beams. See Section 5.3.), and the evaluation of the strains and stresses is optimal at the two Gauss integration points (Section 5.4). The details of the computer program for the 3-node isobeam element are presented in Section 5.5, and the program listing is given in Appendix B. The isobeam element is tested by solving selected problems (Section 5.6), and the accuracy of the element in predicting the linear response of glulam space beams is verified by comparing the analyses and experimental responses of southern pine glulam beams subjected to combined loads (Section 5.7). The application of the isobeam element to the analysis of glulam timber domes is illustrated in Chapter 6.

5.2 Evaluation of the Stiffness Matrix

The stiffness matrix of the isobeam described in Chapter 4 is evaluated by numerical integration. For isoparametric elements with intermediate nodes, the generation and evaluation of the element stiffness matrix can represent more than half of the computational cost in a static analysis (Cook 1981). Therefore, it is important to use efficient schemes to reduce the number of computations in finite element programs, particularly in 3-d analyses. The general form of the finite element stiffness matrix is

$$\underline{K} = \int_V \underline{B}^T \underline{D} \underline{B} dV \tag{5.1}$$

where $\underline{B}$ is the strain-displacement matrix and $\underline{D}$ is the material property matrix. To solve Eq. (5.1) by the conventional numerical integration approach, the weight coefficient of each point of integration is multiplied by the material property matrix, the product $\underline{B}^T\underline{D}\underline{B}$ is evaluated at the integration points through the element, and the individual terms are added to generate the element stiffness matrix. The numbers of additions (Σ_1) and multiplications (Π_1) to evaluate the stiffness matrix of a 3-d element is given by (Gupta and Mohraz, 1972)

$$\begin{aligned} \Sigma_1 &= \frac{1}{2}(63r^2 + 237r)I \\ \Pi_1 &= (27r^2 + 117r + 36)I \end{aligned} \tag{5.2}$$

where r is the number of nodes and I is the number of integration points. Gupta and Mohraz (1972) showed that the number of operations of Eqs. (5.2) can be reduced to one-sixth by expressing the element stiffness in tensor form and by taking the material property tensor outside the integral (for linear material behavior). Using this proposed method, the number of operations for the evaluation of the stiffness matrix of a 3-d element becomes

$$\begin{aligned} \Sigma_2 &= \frac{9}{2}(I + 9)(r^2 + r) \\ \Pi_2 &= \frac{9}{2}(I + 9)(r^2 + r) + 3rI \end{aligned} \tag{5.3}$$

As an illustration, if a 2x3x3 integration rule is used for the 3-node isobeam, the number of operations by Eqs. (5.2) and (5.3) are

$$\begin{aligned} \Sigma_1 &= 11{,}502 \qquad & \Sigma_2 &= 1{,}458 \\ \Pi_1 &= 11{,}340 \qquad & \Pi_2 &= 1{,}620 \end{aligned}$$

For this example, a reduction of 85 percent in computational effort is achieved by using the proposed tensorial method. The computational method of Gupta and Mohraz cannot be readily implemented in the 3-d isobeam of Chapter 4, because of the complexity of the element formulation. Thus, to minimize the computational effort in the evaluation of the element stiffness matrix (K_L and K_{NL} in Table 4.1), the product of $\underline{B}^T\underline{DB}$ was evaluated by hand and then integrated numerically. The details of the formulation of K_L and K_{NL} are given next.

5.2.1 Formulation of $\mathbf{K_L}$

From Table 4.1, the linear stiffness matrix K_L is defined by

$$ {}^t_tK_L = \int_{{}^tV} {}^t_tB_L^T \; {}_tC \; {}^t_tB_L^t \; {}^tdV \tag{5.4}$$

In local coordinates ξ, η, ζ, the strain-displacement matrix for a node k is given by

$$\underset{(3x6)}{\overline{B}_L} = \underset{(3x6)(6x6)}{T \quad B_L} \tag{5.5}$$

where B_L is the strain-displacement matrix in global coordinates x, y, z (Eq. 4.23), and T is the transformation matrix (Eq. 4.26). The product $\overline{B}_L^T\overline{C}\overline{B}_L$ can be written as

$$\underset{(18x18)}{\overline{B}_L^T\overline{C}\overline{B}_L} = \begin{bmatrix} \overline{B}_L^{1T} \\ \overline{B}_L^{2T} \\ \overline{B}_L^{3T} \end{bmatrix} \begin{bmatrix} \overline{C}\overline{B}_L^1 & \overline{C}\overline{B}_L^2 & \overline{C}\overline{B}_L^3 \end{bmatrix} \tag{5.6}$$

or

$$\overline{B}_L^T\overline{C}\overline{B}_L = \begin{bmatrix} \overline{B}_L^{1T}\overline{C}\overline{B}_L^1 & \overline{B}_L^{1T}\overline{C}\overline{B}_L^2 & \overline{B}_L^{1T}\overline{C}\overline{B}_L^3 \\ & \overline{B}_L^{2T}\overline{C}\overline{B}_L^2 & \overline{B}_L^{2T}\overline{C}\overline{B}_L^3 \\ \text{symmetric} & & \overline{B}_L^{3T}\overline{C}\overline{B}_L^3 \end{bmatrix} \tag{5.7}$$

where $\overline{B}_L^{iT}$ and $\overline{C}\overline{B}_L^i$ (i = 1,2,3) are, respectively, (6x3) and (3x6). The (6x6) submatrices of Eq. (5.7) can be generated from the following expression

$$S_{ij}^{kl} = E\;\overline{B}_{1i}^{k}\;\overline{B}_{1j}^{l} + \frac{G}{\kappa}(\overline{B}_{2i}^{k}\;\overline{B}_{2j}^{l} + \overline{B}_{3i}^{k}\;\overline{B}_{3j}^{l}) \tag{5.8}$$

where k, l indicate element node numbers ($k = 1, 2, 3$; $l = 1,2,3$); $i, j = 1,2,3, \ldots ,6$; E and G are the material elastic constants, and κ is the shear correction factor. Using Eq. (5.8), only the upper triangular elements of each (6x6) submatrix of Eq. (5.7) are generated. For k=1, the matrix is symmetric (i.e., the diagonal submatrices of Eq. (5.7); e.g., $\bar{B}_L^{1T}\ \bar{C}\ \bar{B}_L^1$), and therefore, only the upper triangular elements are required. For k ≠l (e.g., $\bar{B}_L^{1T}\ \bar{C}\ \bar{B}_L^2$), the lower triangular elements of the matrix can be generated from Eq. (5.8) by interchanging the indices k and l. Therefore, the elements of $\bar{K}_L$ are completely defined by Eqs. (5.7) and (5.8).

5.2.2 Formulation of K_{NL}

The nonlinear strain stiffness matrix is (Table 4.1)

$$ {}^t_tK_{NL} = \int_{{}^tV} {}^t_tB^T_{NL}\ {}^t\tau\ {}^t_tB_{NL}\ {}^tdV \tag{5.9}$$

where the product of the matrices of the integral is defined in local coordinates as

$$\underset{(18 \times 18)}{{}^t_t\bar{B}^T_{NL}\ {}^t\bar{\tau}\ {}^t_t\bar{B}_{NL}} = \begin{bmatrix} \bar{B}^{1T}_{NL}\ \bar{\tau}\ \bar{B}^1_{NL} & \bar{B}^{1T}_{NL}\ \bar{\tau}\ \bar{B}^2_{NL} & \bar{B}^{1T}_{NL}\ \bar{\tau}\ \bar{B}^3_{NL} \\ & \bar{B}^{2T}_{NL}\ \bar{\tau}\ \bar{B}^2_{NL} & \bar{B}^2_{NL}\ \bar{\tau}\ \bar{B}^3_{NL} \\ \text{symmetric} & & \bar{B}^{3T}_{NL}\ \bar{\tau}\ \bar{B}^3_{NL} \end{bmatrix} \tag{5.10}$$

The elements of Eq. (5.10) are (6x6) submatrices. For convenience, these matrices are defined as:

$$\bar{Z}^{kl} = \bar{B}^{kT}_{NL}\ \bar{\tau}\ \bar{B}^l_{NL} \qquad k, l = 1,2,3 \tag{5.11}$$

A simple tensorial expression that can generate the terms of the matrices of Eq. (5.11) was not possible, and the upper triangular elements of each (6x6) matrix (Eq. (5.11)) is defined by the following expressions

$$
\begin{aligned}
&{}^{1}a_j = \tau_1 B_{1j} + \tau_2 B_{2j} + \tau_3 B_{3j} \quad ,j = 1,2,3,\ldots,6 \\
&{}^{1}b_j = \tau_2 B_{1j} \\
&{}^{1}c_j = \tau_3 B_{1j}
\end{aligned}
$$

$$Z^{kl}_{1j} = B^{k}_{11}\ {}^{1}a^{l}_j + B^{k}_{21}\ {}^{1}b^{l}_j + B^{k}_{31}\ {}^{1}c^{l}_j$$

$$k, l = 1,2,3; \quad j = 1,2,3,\ldots,6$$

(Note: $Z^{kl}_{12} = Z^{kl}_{13} = Z^{kl}_{14} = 0$)

$$
\begin{aligned}
&{}^{2}a_j = \tau_1 B_{4j} + \tau_2 B_{5j} + \tau_3 B_{6j} \\
&{}^{2}b_j = \tau_2 B_{4j} \\
&{}^{2}c_j = \tau_3 B_{4j}
\end{aligned}
$$

$$Z^{kl}_{2j} = B^{k}_{42}\ {}^{2}a^{l}_j \ + B^{k}_{52}\ {}^{2}b^{l}_j \ + B^{k}_{62}\ {}^{2}c^{l}_j$$

(Note: $Z^{kl}_{23} = Z^{kl}_{25} = 0$)

$$
\begin{aligned}
&{}^{3}a_j = \tau_1 B_{7j} + \tau_2 B_{8j} + \tau_3 B_{9j} \\
&{}^{3}b_j = \tau_2 B_{7j} \\
&{}^{3}c_j = \tau_3 B_{7j}
\end{aligned}
$$

$$Z^{kl}_{3j} = B^{k}_{73}\ {}^{3}a^{l}_j + B^{k}_{83}\ {}^{3}b^{l}_j + B^{k}_{93}\ {}^{3}c^{l}_j$$

(Note: $Z^{kl}_{36} = 0$)

$$\begin{aligned}
Z_{44}^{kl} &= B_{44}^{k}\ {}^{2}a_{4}^{l} + B_{54}\ {}^{2}b_{4}^{l} + B_{64}^{k}\ {}^{2}c_{4}^{l} + B_{74}^{k}\ {}^{3}a_{4}^{l} + B_{84}^{k}\ {}^{3}b_{4}^{l} + B_{94}^{k}\ {}^{3}c_{4}^{l} \\
Z_{45}^{kl} &= B_{74}^{k}\ {}^{3}a_{5}^{l} + B_{84}\ {}^{3}b_{5}^{l} + B_{94}\ {}^{3}c_{5}^{l} \\
Z_{46}^{kl} &= B_{44}^{k}\ {}^{2}a_{6}^{l} + B_{54}\ {}^{2}b_{6}^{l} + B_{64}\ {}^{2}c_{6}^{l} \\
Z_{55}^{kl} &= B_{15}^{k}\ {}^{1}a_{5}^{l} + B_{25}^{k}\ {}^{1}b_{5}^{l} + B_{35}^{k}\ {}^{1}c_{5}^{l} + B_{75}^{k}\ {}^{3}a_{5}^{l} + B_{85}^{k}\ {}^{3}b_{5}^{l} + B_{95}\ {}^{3}c_{5}^{l} \\
Z_{56}^{kl} &= B_{15}^{k}\ {}^{1}a_{6}^{l} + B_{25}^{k}\ {}^{1}b_{6}^{l} + B_{35}^{k}\ {}^{1}c_{6}^{l} \\
Z_{66}^{kl} &= B_{16}^{k}\ {}^{1}a_{6}^{l} + B_{26}^{k}\ {}^{1}b_{6}^{l} + B_{36}^{k}\ {}^{1}c_{6}^{l} + B_{46}^{k}\ {}^{2}a_{6}^{l} + B_{56}^{k}\ {}^{2}b_{6}^{l} + B_{66}^{k}\ {}^{2}c_{6}^{l}
\end{aligned} \tag{5.12}$$

NOTE: the bar denoting local coordinates has been dropped, and $\tau_{1,2,3}$ are the stress components in local axes ξ, η, ζ .

Once again, the diagonal submatrices of Eq. (5.10) are symmetric (i.e., k=l in Eq. (5.11)) and are completely defined by Eqs. (5.12). For k ≠ l (e.g., $\overline{B}_{NL}^{1T}\ \bar{\tau}\ \overline{B}_{NL}^{2}$), the lower triangular elements of the submatrices (Eq. (5.11)) are generated by interchanging k and l in expressions (5.12). At this point, all the element matrices are completely defined, and in the next two sections we discuss briefly the concept of "shear locking," the correct computation of stresses, and the appropriate integration rule to evaluate the stiffness matrix.

5.3 Shear and Membrane Locking

To illustrate shear locking, consider the 2-node isoparametric beam element of Fig. 5.1. The displacements and rotations at any point along the element are expressed as

$$\begin{aligned}
\mathrm{v} &= [h_1\ \ h_2]\begin{bmatrix}\mathrm{v}_1 \\ \mathrm{v}_2\end{bmatrix} \\
\theta &= [h_1\ \ h_2]\begin{bmatrix}\theta_1 \\ \theta_2\end{bmatrix}
\end{aligned} \tag{5.13}$$

where the interpolation polynomials are

$$h_1 = \frac{1}{2}(1-r), \quad h_2 = \frac{1}{2}(1+r)$$

and the strains in the element are given by

$$\varepsilon = -y\theta_{,x} \quad \text{and} \quad \gamma_{xy} = \mathrm{v}_{,x} - \theta \tag{5.14}$$

We can observe from Eq. (5.13) that v and θ are interpolated independently; but they are coupled by the transverse shear strain γ_{xy} (Fig. 5.1c).

The strain energy of the beam element of Fig. (5.1b) is (Cook, 1981, p. 261; Bathe, 1982, p. 239)

$$U = U_b + U_s \tag{5.15}$$

where the bending and shear strain energies are

$$U_b = \frac{1}{2}\left(\frac{Ebt^3}{12}\int_0^L (\theta_{,x})^2 dx\right)$$

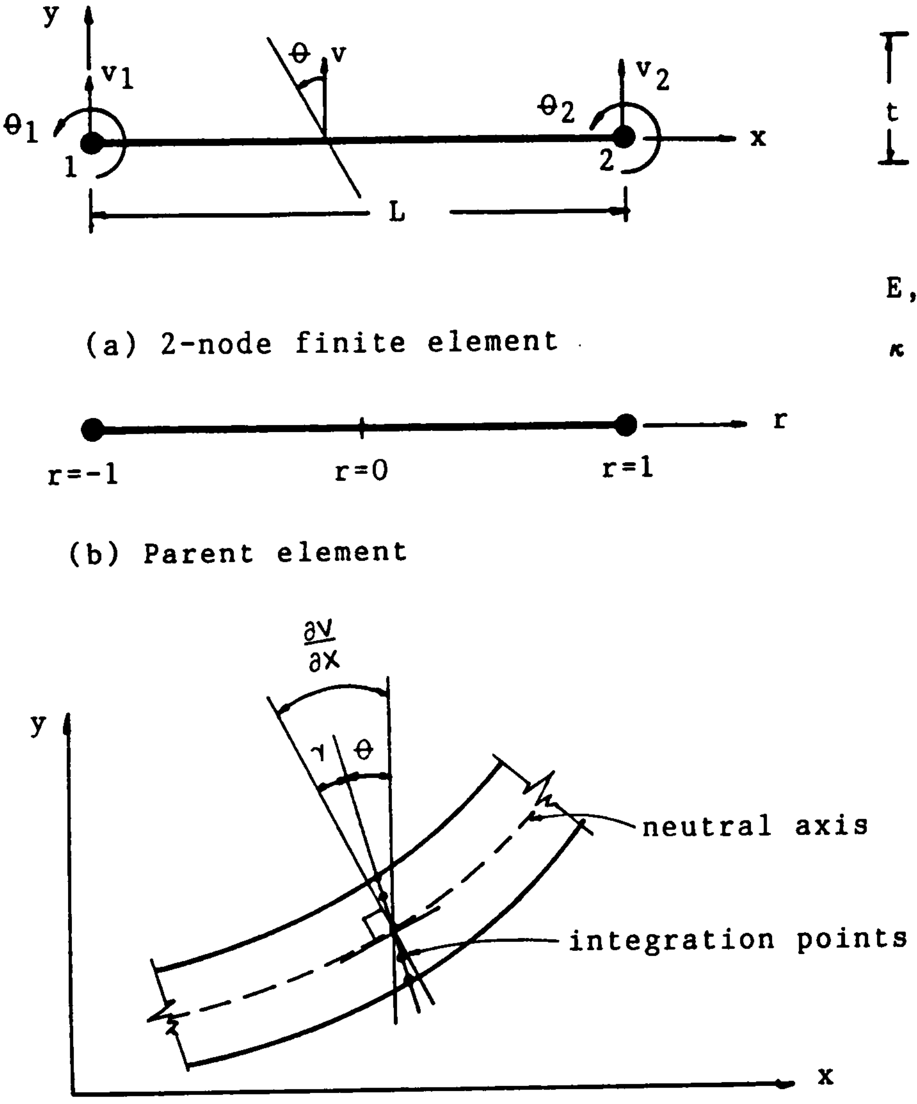
$$U_s = \frac{1}{2}\left(\frac{Gbt}{\kappa}\int_0^L (\mathrm{v}_{,x} - \theta)^2 dx\right) \tag{5.16}$$

Dividing Eq. (5.15) by $\frac{Ebt^3}{24}$, we can write

$$\tilde{U} = \int_0^L (\theta_{,x})^2 dx + \alpha\int_0^L (\mathrm{v}_{,x} - \theta)^2 dx \tag{5.17}$$

where

$$\alpha = \frac{12G}{\kappa E t^2}$$



(a) 2-node finite element

(b) Parent element

(c) Trnasverse shear deformation

Figure 5.1. 2-Node isoparametric beam element.

As the beam becomes thin ($h \rightarrow 0$), the condition of zero shear deformation will be approached, i.e., for $\gamma_{xy} = 0, \theta = v_{,x}$. Thus, the Mindlin beam theory approaches the Bernoulli-Euler beam theory, which neglects shear deformations. However, the finite element assumptions on v and θ must insure that for large values of α (small t) the shearing deformations can be small throughout the element. If this is not the case, the stiffness of the finite element will grossly overestimate the stiffness of the actual beam. This phenomenon is called "shear locking": thin elements being very stiff. To avoid shear locking in isoparametric beam elements, reduced integration is used (Bathe, 1982, p. 237). The procedure consists in integrating the shear strain energy by a lower order than the required exact order of integration (i.e., the shear strain energy is not integrated exactly). For example, the exact integration of the strain energy of the 2-node beam of Fig. 5.1 (Eq. 5.15) requires a 2-point Gaussian quadrature (along the beam axis). However, as the beam becomes thin, the factor α in the shearing term of Eq. (5.17) becomes large and acts as a penalty number. Thus, the element becomes "stiff" and eventually "locks." But if the strain energy of the element is integrated by a 1-point rule, the element works well. This corresponds to assuming a constant transverse shear strain (since the shear strain is only evaluated at the mid-point of the beam), while the bending energy is still evaluated accurately ($\theta_{,x}$ is constant; the 2-node beam element can model only a constant moment).

In practice, a reduced order of numerical integration is used in the evaluation of the element stiffness matrix. To avoid shear locking, Bathe (1986) recommends using the 2-, 3-, and 4-node isobeam elements with 1-, 2-, and 3-point Gauss integration along the beam axes. These beam elements, based on reduced integration, are reliable because they do not possess any spurious zero energy modes (they have only six zero eigenvalues in 3-d analysis, corresponding to the six physical rigid body modes).

In addition to not exhibiting erroneous shear strains, the beam element must also not contain erroneous mid-surface membrane strains in the analysis of curved structures. Bathe (1986) shows that a curved cantilever modeled with a 3-node isobeam element does not exhibit erroneous midsurface membrane strains. He concludes that the isobeam elements with reduced integration also do not "membrane lock."

5.4 Stress Computations at Optimal Points

Finite element models exhibit unique points at which the stresses have higher accuracy than at other points. In particular, the stresses are considerably more accurate at the Gauss integration points than at the nodal points (Barlow 1971). In the displacement-based finite element method, the nodal displacements are the prime variables used in the energy minimization, and the stress field is of a lower degree than the displacement field because stresses are proportional to displacement derivatives. Barlow (1971) rationalizes the reasons for the presence of optimal stress points and suggests the following process to locate them:

1. Let n be the order of the highest complete polynomial displacement field **f** of an element.
2. Impose on the element a complete polynomial displacement field **f′** of order n + 1, where **d′** are the corresponding nodal degrees of freedom.
3. Seek locations in the element where stresses calculated from **f′** are identical to stresses calculated from **f** when nodal degrees of freedom **d′** are imposed.

This technique was applied to the 3-node, quadratic isobeam element, and it was demonstrated that the stresses at the two Gauss quadrature points $\pm \frac{1}{\sqrt{3}}$ are as accurate as the nodal displacements.

In the program implementation of the isobeam element (Section 5.5), the strains and stresses are evaluated at the two Gauss points along the element axis. The 3-node isoparametric beam element integrated with two Gauss points solves exactly a beam segment with linearly varying moment and constant transverse shear force (see for example the simple physical argument presented by Cook, 1981, p. 261). More recently, Jirousek (1983) showed that the same element also solves exactly a beam segment with a parabolic moment and linear transverse shear. Although the 2-point integration rule underestimates by 16.6% the strain energy due to a parabolic moment distribution, the magnitude of the moment and the linearly varying transverse shear match the theoretical values at the two sampling points. In conclusion, by using a reduced order of integration, the performance of the isobeam is improved considerably: since the expense of numerical integration is proportional to the number of integration points, the computational cost is reduced; shear and membrane locking are eliminated (Section 5.3), and the normal and shear stresses are more accurate at the Gauss points. Once the stresses are computed at the Gauss points, they can be extrapolated to the nodes using interpolation functions (Cook, 1981, p. 137) or a least squares method (Hinton and Campbell, 1974).

5.5 *Program Development*

This section discusses the structure of the computer program for the 3-node isobeam element. The program is written in FORTRAN 77 and follows the structure presented by Holzer (1985) for the matrix displacement analysis of space frames. The program is structured and should be easy to follow, particularly by those familiar with Holzer's work (1985).

The program performs linear and nonlinear static analysis of rigid-jointed space frames consisting of straight or curved members of rectangular cross sections. Geometric nonlinearities are included (i.e., large displacements, large rotations, but small strains), and the analysis can be performed by either the modified Newton-Ralphson or the modified Riks-Wempner methods. The material behavior is linear, and the material properties required are the longitudinal (E) and transverse (G) shear moduli (see Eq. 3.4). The shear correction factor κ is set to $\frac{6}{5}$ in the program.

Since the material constants E and G are independent, the element can model either isotropic or transversely isotropic beams. This feature is important in modeling glulam beams (Section 3.6). The applied loads can be either concentrated nodal forces or distributed loads linearly varying over the element length.

5.5.1 Program Input

The data required to run the program includes nodal coordinates, boundary conditions, element material properties and cross-section dimensions, direction cosines of the cross section at the nodes (unit vectors described in Section 4.2), type of analysis (linear or nonlinear), and error tolerances. A detailed user's input manual, including the input file for test problem 1 (Section 5.6), is given in Appendix A.

5.5.2 Program Output

The following output can be requested from the program:

1. Nodal displacements and forces: displacements and applied or reactive forces at the nodes are given in global coordinates.
2. Element forces: moments and forces equilibrating each element are given at its three nodes in global coordinates.
3. Strains and stresses: strains and stresses are given at the two Gauss points of an element. Either a 3x3 (9 points) or a 5x5 (25 points) Newton-Cotes integration rule over the cross section can be specified for the stress output at the location of each Gauss point (see Appendix A).

5.5.3 Program Subroutines

A brief description of the individual program subroutines is presented in this section. The program tree-chart is shown in Fig. 5.2, and the program listing is given in Appendix B. In this study, the equation solver presented by Bathe (1982) and modified by Butler (1983) has been directly incorporated in the program (subroutines SOLVE, FACTOR, REDUCE, and BACKSUB), and therefore is not discussed here. The program subroutines are:

MAIN

It starts the analysis procedure, initializes the variables, and sets up the pointers. If memory allocation is sufficient, it calls STRUCT, LOAD, and LINEAR for linear analysis and STRUCT, LOAD, and NONLIN for nonlinear analysis.

STRUCT

It calculates the size of and allocates the required memory for the system stiffness matrix, reads member incidences, joint constraints, and element properties, and calls CODES, SKYLIN, and PROP.

LOAD

It initializes the load vectors Q and QBAR to zero, reads QIMAX, QI, DQI for nonlinear analysis, or equates QI and QIMAX to 1 for linear analysis. It calls JLOAD and DLOAD.

LINEAR

It is a control module for linear analysis. It calculates the system stiffness matrix and stores it in the already allocated space, performs the analysis and echoes the results, and calls STIFF, POST, SOLVE, and RESULT.

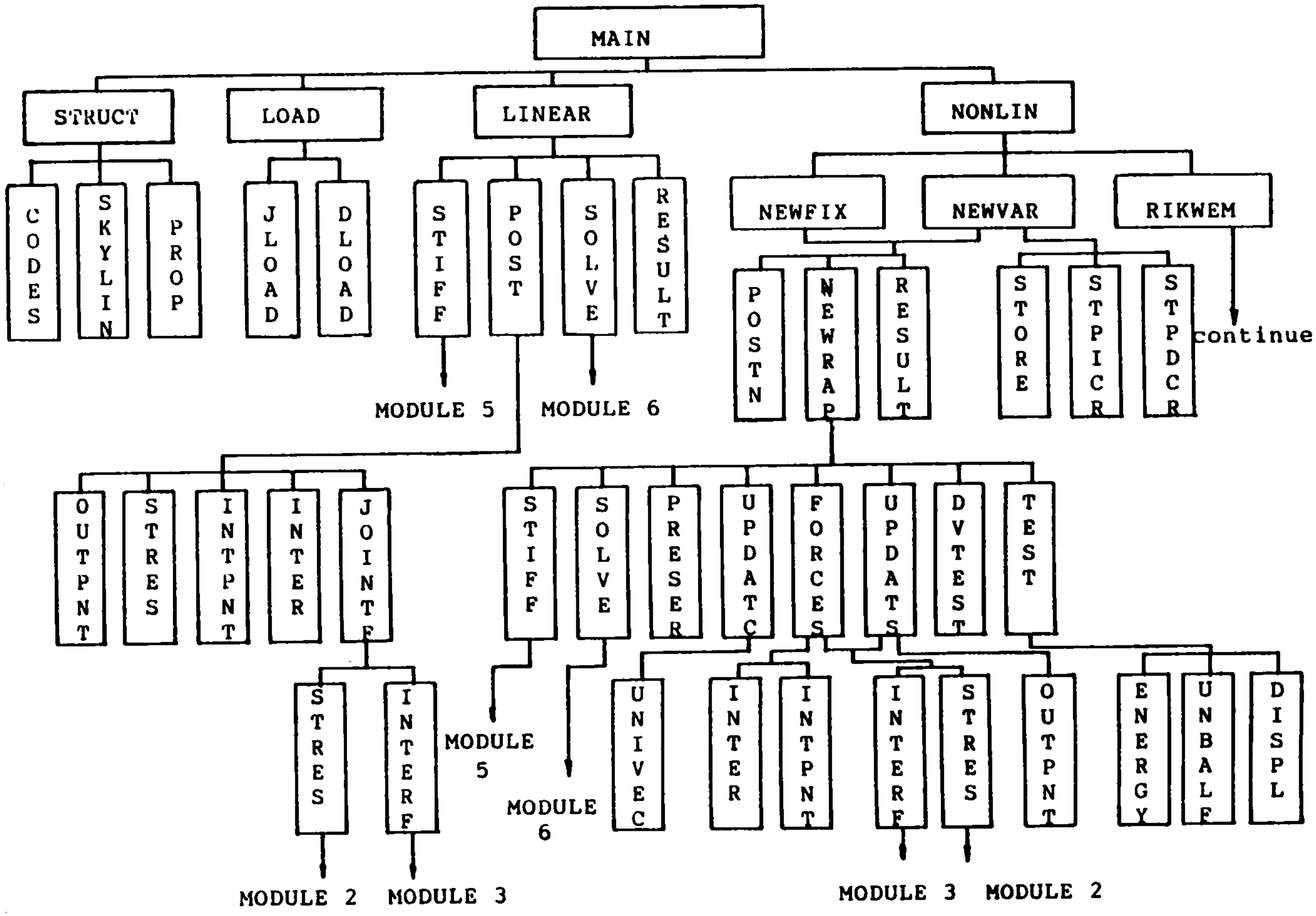
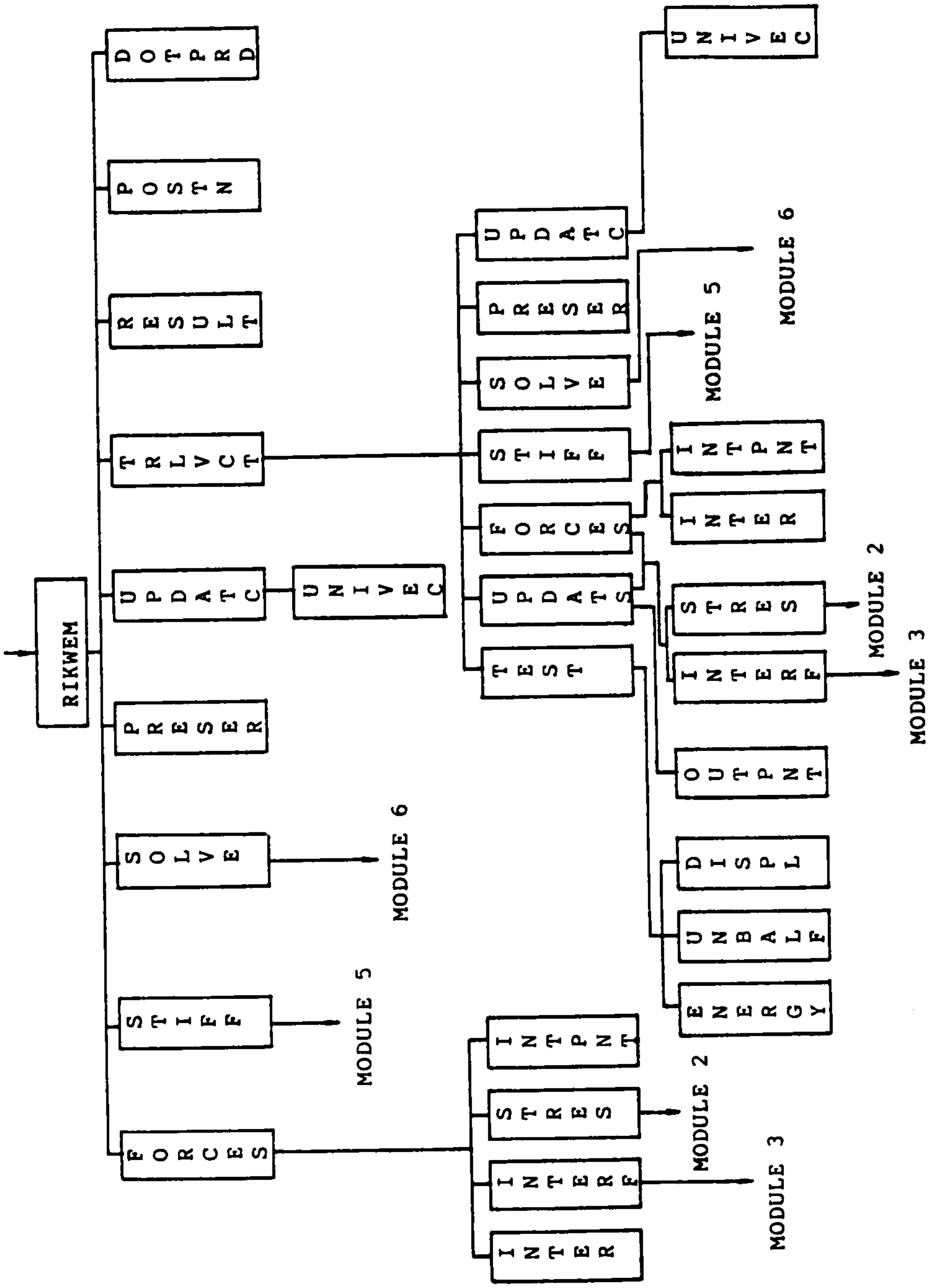
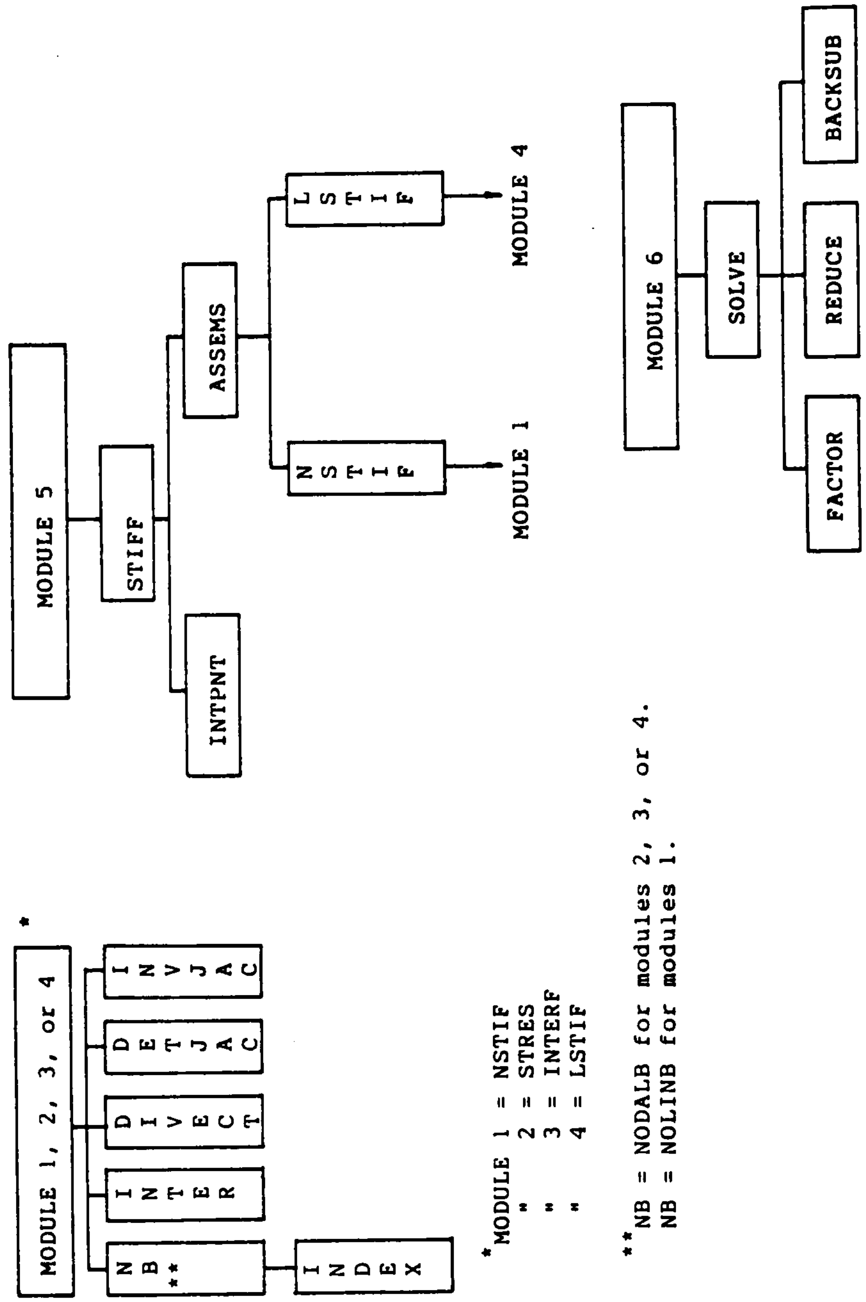


Figure 5.2. Program structure.



Program structure (cont.)



*MODULE 1 = NSTIF
" 2 = STRES
" 3 = INTERF
" 4 = LSTIF

**NB = NODALB for modules 2, 3, or 4.
NB = NOLINB for modules 1.

Program structure (cont.)

CODES

It calculates and saves the joint code matrix JCODE and the member code matrix MCODE.

SKYLIN

It determines the lengths of the tangent stiffness vector SS, the vector which stores the addresses of the main diagonal terms MAXA, and the vector containing the number of elements below the skyline of each column.

PROP

It reads and echoes nodal coordinates, unit vectors, and element properties (width, depth, E, and G).

UNIVEC

It computes the nodal unit vectors in the local t direction.

JLOAD

It reads, echoes, and stores the system nodal loads. It calculates the constant load vector QBAR for nonlinear analysis.

DLOAD

It reads and echoes the member distributed loads, discretizes the loads and stores them in vector form, and calculates the constant load vector for nonlinear analysis.

STIFF

It is a control module for determining and updating the tangent stiffness vector SSS for the current configuration. It initializes SSS and calls INTPNT and ASSEMS.

ASSEMS

It calculates the linear and nonlinear element stiffness contributions and assembles them into a system stiffness vector. It calls LSTIF and NSTIF.

LSTIF

It calculates the linear element stiffness contribution, for the unconstrained degrees of freedom to the system stiffness matrix, and calls INTER, DIVECT, DETJAC, INVJAC, and NODALB.

NSTIF

It calculates the nonlinear element stiffness contributions, for the unconstrained degrees of freedom, to the system stiffness matrix. It calls INTER, DIVECT, DETJAC, INVJAC, and NOLINB.

NODALB

It computes the linear strain-displacement matrix and calls INDEX.

NOLINB

It computes the nonlinear strain-displacement matrix and calls INDEX.

INDEX

It calculates the values of G by Eq. 4.2.1 for an interpolation point.

DETJAC, INVJAC

They calculate the determinant and the inverse of the Jacobian for an interpolation point.

DIVECT

It calculates the unit vectors of the cross section at an interpolation point.

INTPNT, OUTPNT

They assign the values for the Gauss or Newton-Cotes integration rule selected.

POST

It performs the post-processing for linear analysis. It calls OUTPNT, STRES, INPNT, INTERF, and JOINTF.

RESULT

It prints the results in an organized manner.

INTERF

It calculates the element internal forces and assembles the system force vector. It calls INTER, DIVECT, DETJAC, INVJAC, and NODALB.

STRES

It calculates the stresses at a given interpolation point in the element. It calls INTER, DIVECT, DETJAC, INVJAC, and NODALB.

JOINTF

It calculates the reactions (or applied loads) at the requested nodes. and calls STRES and INTERF.

NONLIN

It is a control module for nonlinear analysis. It calls NEWFIX, NEWVAR, and RIKWEM.

NEWFIX

It is a control module for the Modified Newton-Ralphson (N/R) method using fixed time increments. It calls NEWRAP, POSTN, and RESULT.

NEWRAP

It performs the modified N/R analysis until the convergence criteria is met or the number of iterations is exceeded. It calls STIFF, SOLVE, PRESER, UPDATC, FORCES, UPDATS, DVTEST, and TEST.

FORCES

It is a control module for determining element internal forces (in global coordinates) and calculating the residual force vector. It calls INTPNT, INTER, STRES, and INTERF.

UPDATC

It updates the coordinates and nodal vectors of the system after each time increment and calls UNIVEC.

UPDATS

It updates the element stresses after each time interval and calls STRES, INTPNT, OUTPNT.

TEST

It performs convergence tests and calls ENERGY, UNBALF, DISPL.

ENERGY, UNBALF, DISPL

They perform convergence tests in energy, unbalanced force, or displacement.

POSTN

It performs the post-processing of nonlinear analysis for each time increment.

INTER

It calculates the interpolation functions and their derivatives.

RIKWEM

It calculates the generalized arc length in the modified Riks-Wempner method and calls TRLVCT, STIFF, SOLVE, PRESER, UPDATC, RESULT, POSTN, and FORCES.

TRLVCT

It performs iterations to obtain vectors orthogonal to the generalized arc length and calls FORCES, UPDATS, STIFF, SOLVE, PRESER, UPDATC, and TEST.

PRESER

It saves the current configuration before each time increment.

NEWVAR

It is a control module for the modified N/R method using a variable time step. Calls STORE, NEWRAP, STPDCR, STPICR, POSTN, and RESULT.

DVTEST

It estimates the required number of iterations for convergence by the modified N/R method (variable step: NEWVAR).

STPDCR

It reduces the step increment/decrement size by 75% in NEWVAR. It prints an error message and terminates the program if the step increment is already equal to DQIMIN.

STPICR

It increases the time increment by 25% and checks the new value to keep it within the maximum specified (DQIMAX).

STORE

It stores the configuration of the system at time t (if no convergence is achieved in the prescribed number of iterations, it returns to the current configuration and reiterates with a reduced time increment: NEWVAR).

DOTPRD

It is a function to compute the dot product of two vectors DOT1 and DOT2.

5.6 Test Problems

The formulation of the isobeam element was tested by evaluating the response of various structures and comparing the results with known solutions. The following three test problems are reported.

1. **Williams' Toggle Frames**

   Williams (1964) investigated theoretically and experimentally the two frames shown in Fig 5.3. His results are widely used to validate analytical models (e.g., Wood and Zienkiewicz, 1977; Papadrakakis, 1981). In this study, two isobeam elements per member were used in the geometric nonlinear analyses of these frames by the modified Riks-Wempner method. Plots of the load vs. vertical displacement of the apex for the two frames are shown in Fig. 5.3. The analytical results are in close agreement with Williams' experimental values.

2. **Three-dimensional Cantilever Beam of a 45-degree Bend**

   A square-section (1" x 1") cantilever beam of 45° bend, subjected to a tip load and undergoing large displacements, was investigated by Bathe and Bolourchi (1979). The beam lies in the 1-3 plane and its radius of curvature is 100 inches. The concentrated tip load is applied in the 3-direction (Fig. 5.4(a)). Four isobeam elements were used in the geometric nonlinear analysis by the modified Newton-Ralphson method. The tip deflections in the 2-direction, shown in Fig. 5.4(b), compare very well with the solution given by Bathe and Bolourchi (1979).

3. **Twelve-member Space Frame (M.I.T. Dome)**

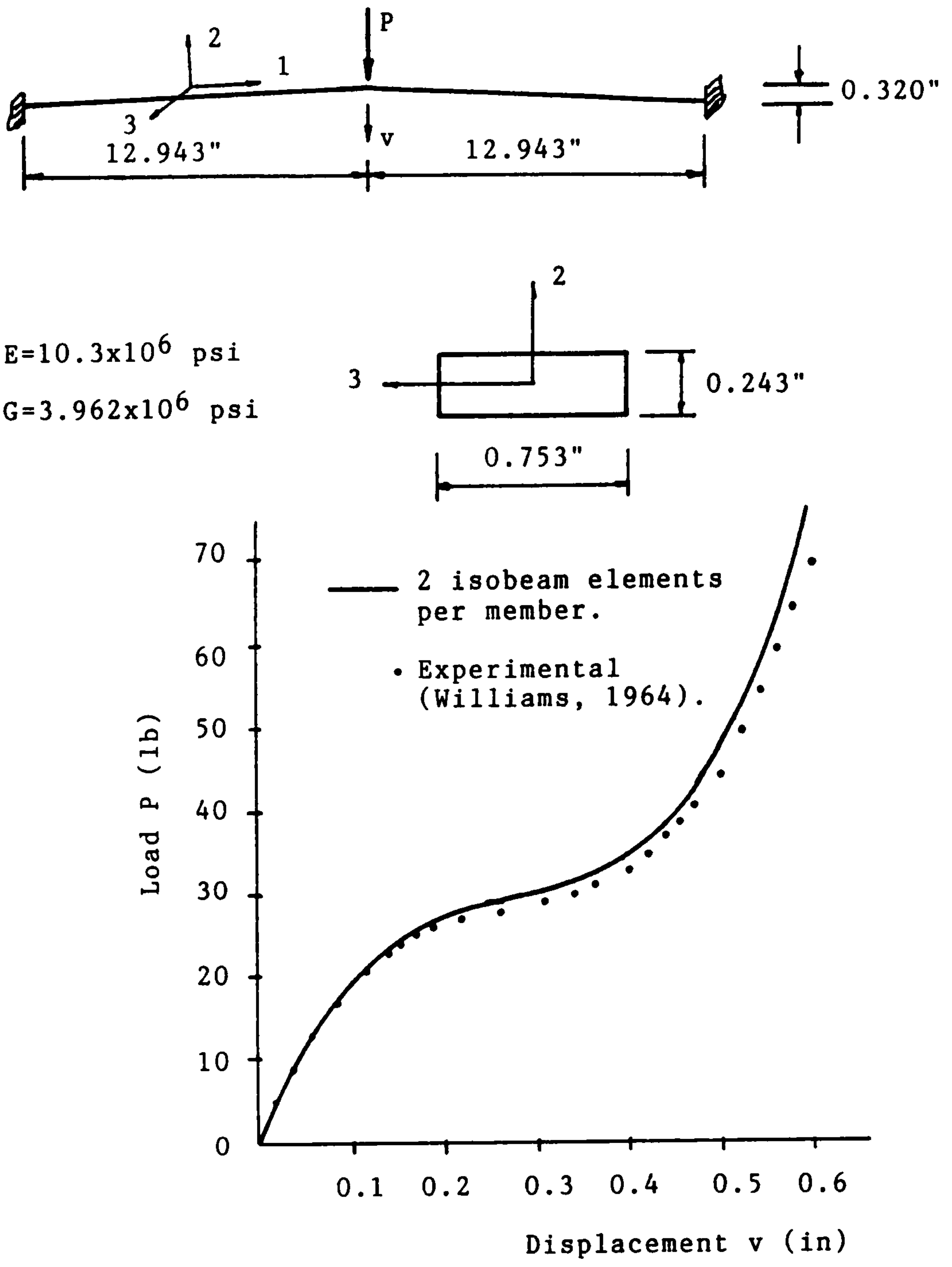


Figure 5.3a. Williams' Toggle Frame 1.

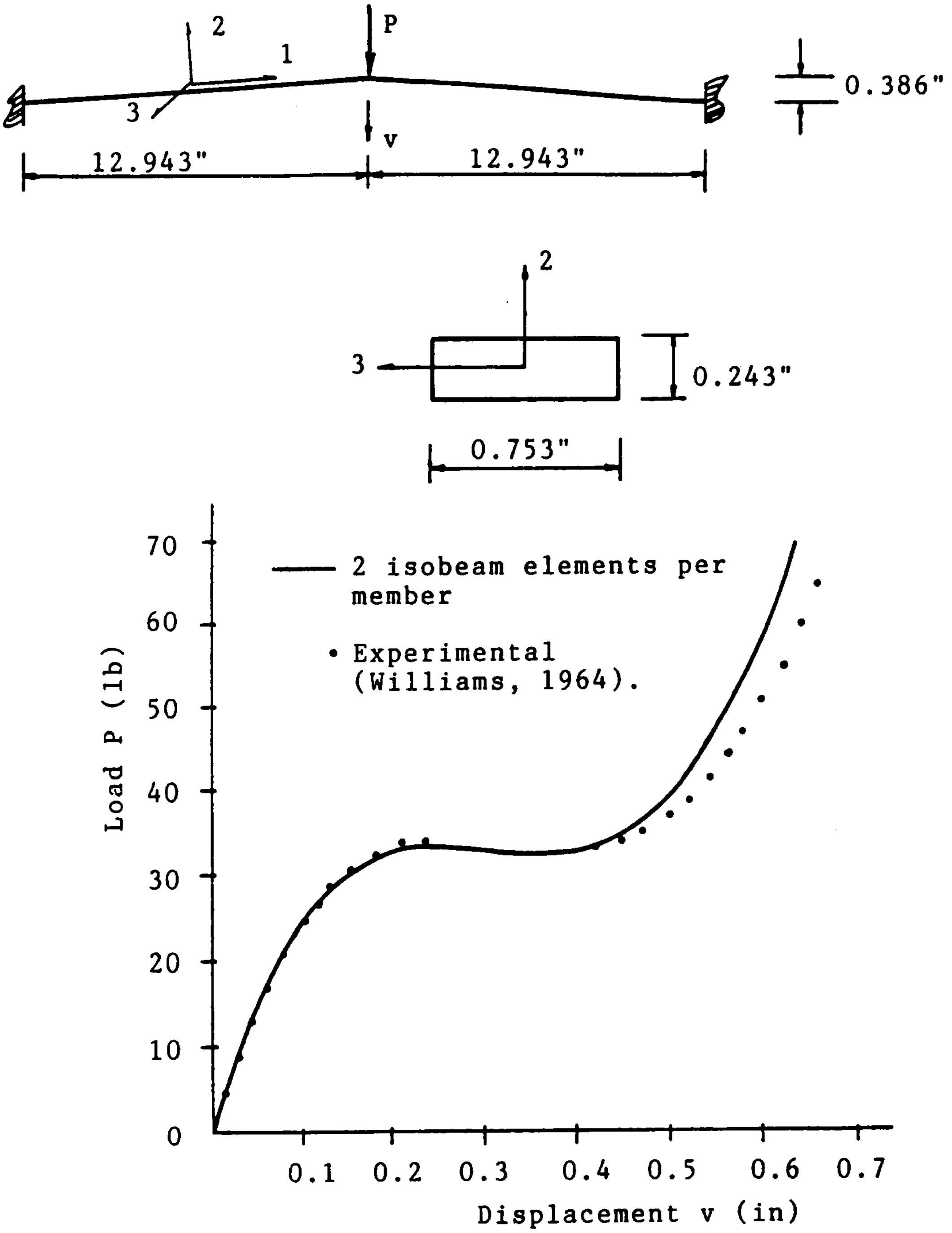


Figure 5.3b. Williams' Toggle Frame 2.

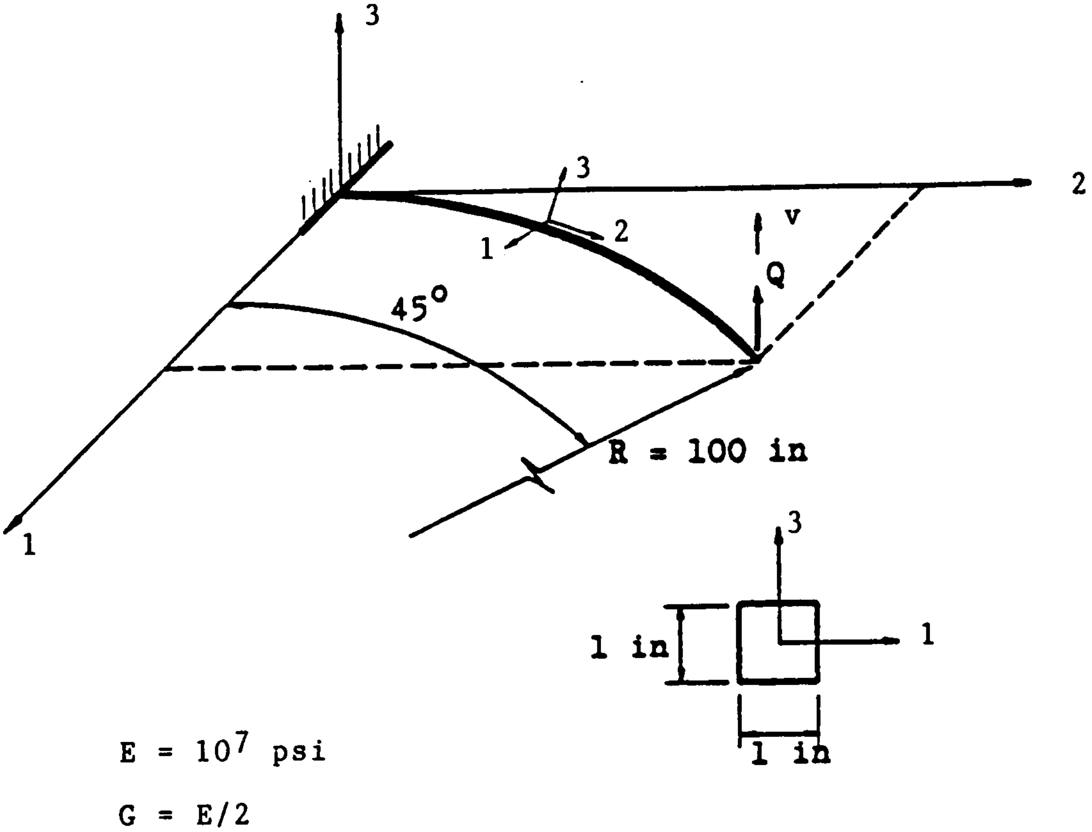


Figure 5.4a. Three-dimensional cantilever beam of 45-degree bend.

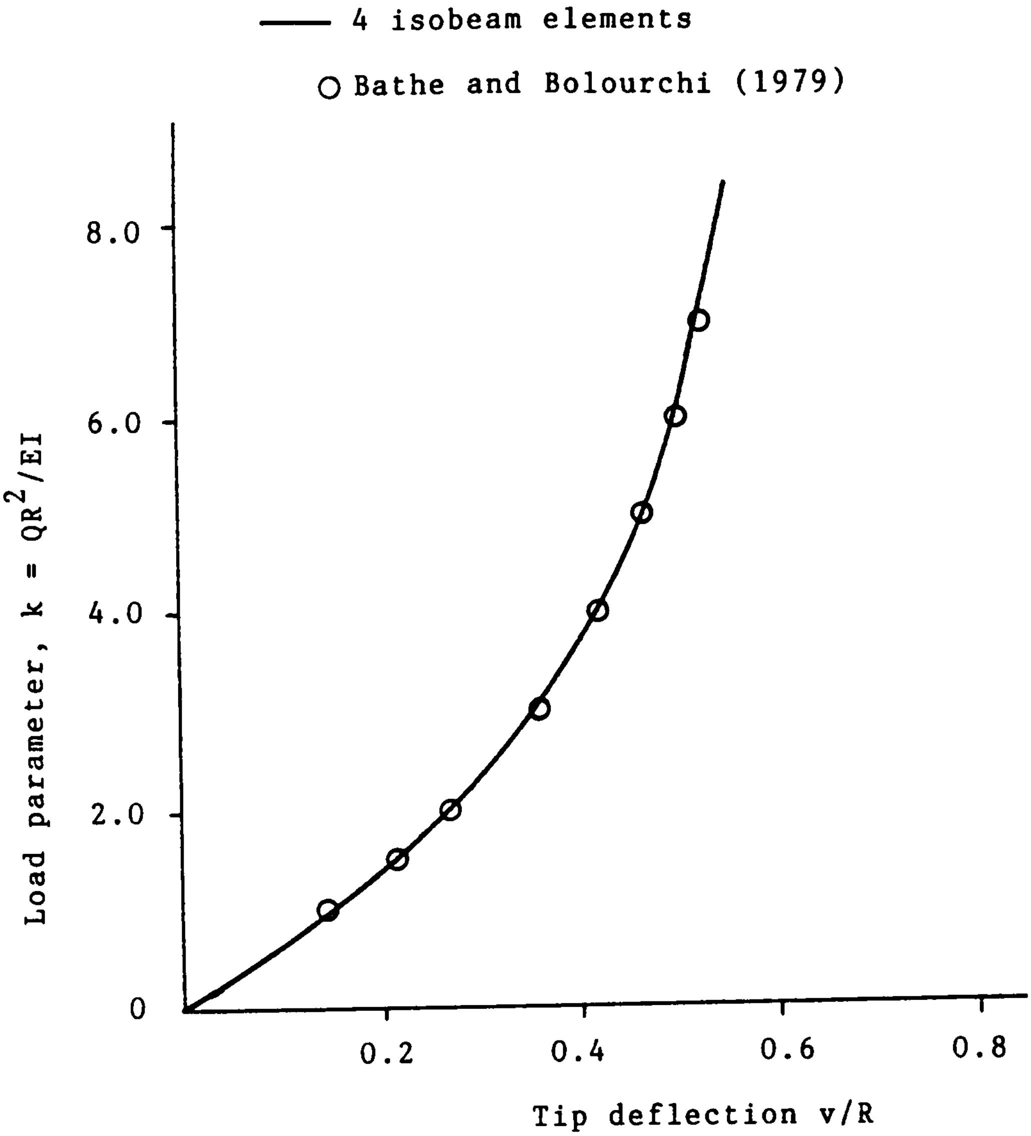


Figure 5.4b. Cantiliver beam of 45-degree bend: load-displacement (in the 3-direction) curve.

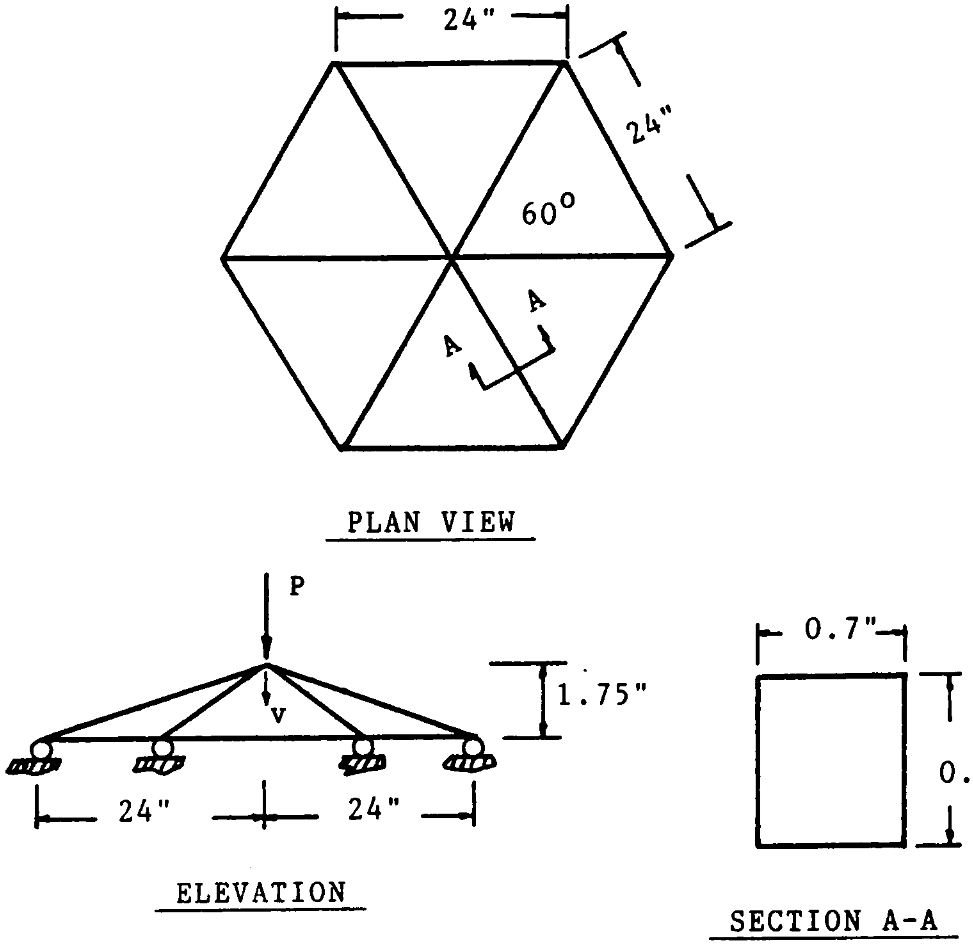
A shallow reticulated space frame consisting of twelve 24-inch members was built and tested at M.I.T. (Griggs, 1966). The members were made of plexiglass and had square cross sections of area 0.494 in^2 (Fig. 5.5(a)). The frame was tested under a concentrated apex load and the experimental results have been used to validate analytical solutions (Connor et al., 1968; Chu and Rampetsreiter, 1972; Papadrakakis, 1981).

In this study, each member of the M.I.T. dome was modeled with one isobeam element. The plot of the load vs. vertical displacement of the apex (Fig. 5.5(b)) illustrates the snap-through behavior of the structure. The first limit load (buckling load) is 3.5% higher than the reported experimental value (Griggs, 1966), and the overall analytical and experimental results are in good agreement.

5.7 *Experimental Verification of the Isobeam*

The 3-node isobeam element can efficiently represent the response of space glulam beams by considering the material transversely isotropic (Section 3.6) and using average material properties through the volume of the element (Section 3.7). In this study, the accuracy of the modeling assumptions for southern pine glulam beams, which are incorporated in the isobeam element, are experimentally tested. Full-size southern pine glulam beams (3 curved and 4 straight) were fabricated in the laboratory and tested in biaxial bending and combined bending and compression. The finite element analysis of the beams was based on the modeling recommendations discussed in Section 3.8. Specifically, the following simplifying assumptions were used:

- The contribution of the gluelines to the response of the beam is neglected, and the material is assumed continuous and homogeneous (Section 3.5).



E = 439,800. psi

G = 159,000. psi

Figure 5.5a. Twelve member space frame (M.I.T. dome)

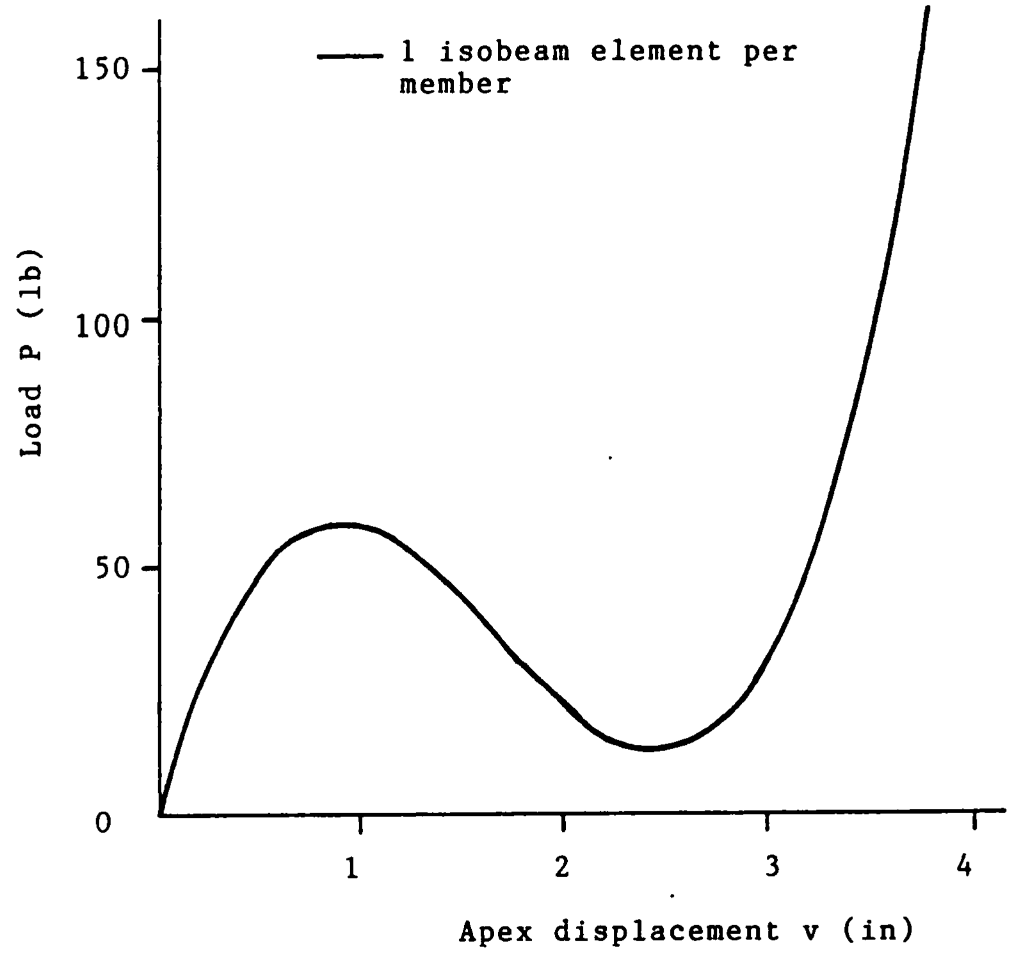


Figure 5.5b. M.I.T. dome: load-vertical displacement of the apex.

- The random orientations of the cross-sectional growth rings of the beam's laminae are assumed to produce an averaging of the properties in the plane of the cross section. Therefore, the beam is modeled as transversely isotropic (Section 3.6), and the shear response is characterized by a single shear modulus, G.

- The material properties are assumed constant through the volume of the beam element. This assumption is quite severe, since it was illustrated in Section 3.7 that the variability in the longitudinal elastic modulus, E_L, along the length of a lamina can be as much as 40%.

Based on these assumptions, it is hypothesized that the isobeam element can accurately represent the overall linear response of the laboratory-built glulam beams subjected to combined loads. The validity of the modeling assumptions are verified by comparing the predicted and measured normal strains and flexural displacements. The description of the testing program is given in Fig. 5.6.

The beams were fabricated in the laboratory following the guidelines given in AITC-117-87-MANUFACTURING. The longitudinal elastic moduli of the beams were obtained from tension tests of the laminae (Section 3.7), and the shear moduli were obtained from torsion tests of small glulam samples as discussed in Section 3.6. The details of the testing procedures and results are discussed by Yadama (1989), and in this section the response of only two beams (one curved and one straight) is briefly investigated. The geometry and material properties of the two beams (C1 and S3) are given in Table 5.1. The beams were subjected to concentrated loads and tested in bending (about both principal axes of the cross section) and combined bending and compression. The strains were recorded with "clip-on" electrical transducers (Loferski et al., 1989), and the displacements were measured with electrical extensometers at three points along the length of the beams. The analytical and experimental results are presented next.

Testing Program for the Southern Pine Glulam Beams

Material properties

Elastic Modulus
Tension tests of the laminae

Shear Modulus
Torsion tests of small glulam samples (Saint-Venant's torsion theory)

Member response: 3 curved and 4 straight beams (several spans tested)

Biaxial Bending

Bending and Compression

Fig. 5.6. Organizational Testing Program for the Verification of the Isobeam

Table 5.1. Characteristics of the test beams C1 and S3.

Characteristics	Beam C1	Beam S3
Width	2.994"	4.865"
Depth	5.984"	6.492"
Span(s)	83"	83", 132"
No. of laminae	9*	5**
Elastic modulus	1.87x10^6 psi*	1.98x10^6 psi**
Shear modulus	1.6x10^5 psi***	1.6x10^5 psi***

* See Table 3.5

** See Table 3.6

*** See Table 3.4

5.7.1 Beam C1

The strains were recorded at the midspan of the beam by seven equally spaced clip gages. The displacements were measured under the bending load application points (L/3 and 2L/3) and at midspan.

1. Bending. Two symmetric concentrated loads were applied at L/3 and 2L/3 (Fig. 5.7(a)). The analytical and experimental strains for two load levels (P = 250, 750 lbs) are plotted in Fig. 5.7(a). The corresponding displacements are compared in Fig. 5.7(b). It can be observed that excellent agreement is obtained between the experimental and analytical results.

2. Bending and compression. First, the beam was subjected to bending by two symmetric concentrated loads of 750 lbs each. Then concentric axial loads were applied while holding the bending load constant. When an axial load was applied, the beam tended to deflect upwards, but the hydraulic ram applying the bending load restrained the points of load application (L/3 and 2L/3) from moving in the vertical direction. Therefore, in the finite element analysis the nodes corresponding to these points were restrained in the 2-direction after the bending load was applied (Fig. 5.8(a)). The analytical and experimental strains for three increments of axial loads (Q = 0, 2000, 6000 lbs) are compared in Fig. 5.8(a). The finite element analysis predicts quite well the overall response of the beam. Although at some points, due to the variability in the material properties, the differences between the predicted and measured strains can be as much as 20%, the experimental values never exceed the analytical values. Therefore, the results of the finite element analysis are conservative. The displacements compare very well and the differences are about 3% (Fig. 5.8(b)).

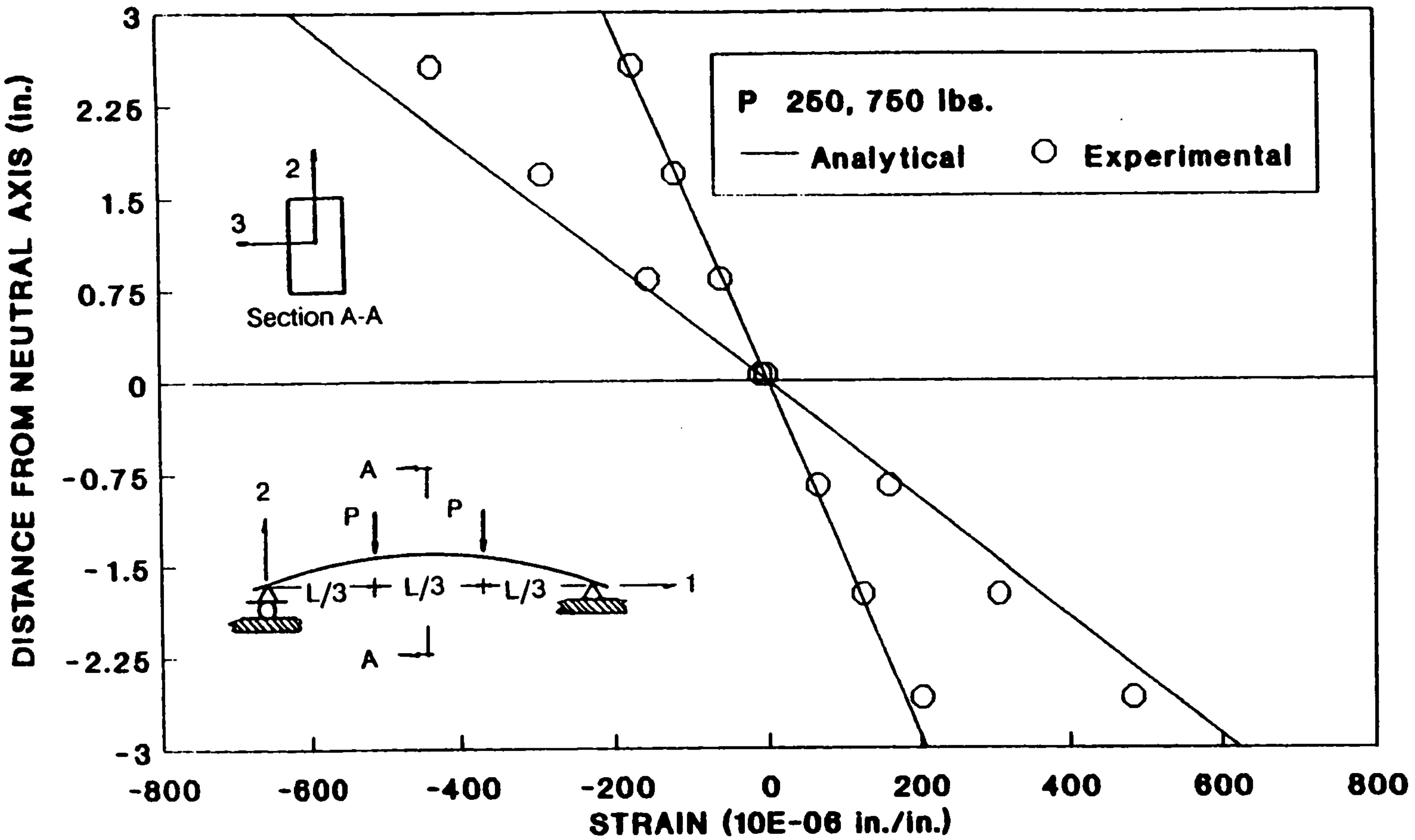


Fig. 5.7 (a) Beam C1, bending Test: Normal strains at midspan. (loads in 2-direction).

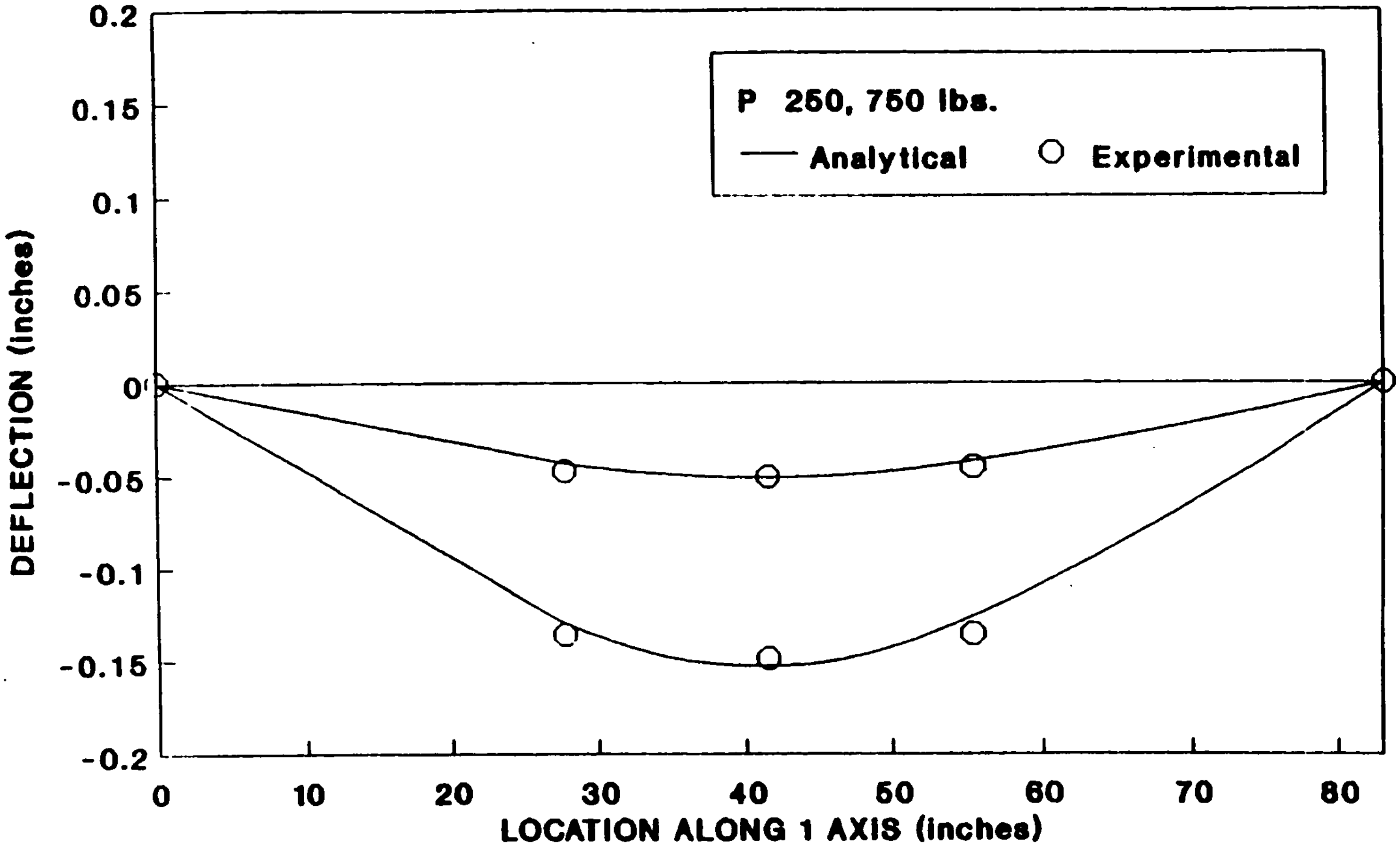


Fig. 5.7(b) Beam C1, Bending Test: Displacements along the beam. (loads in 2-direction).

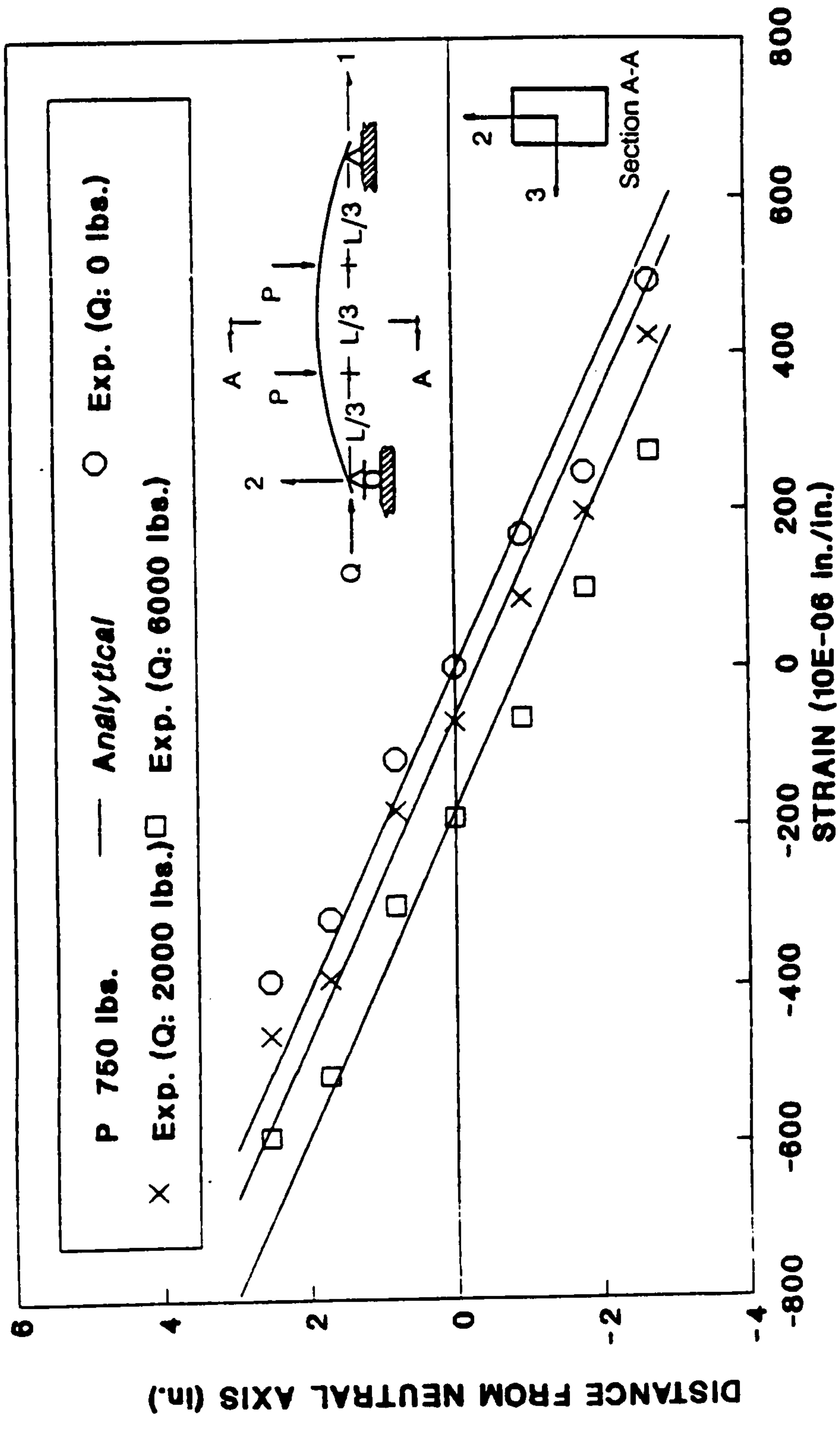


Fig. 5.8(a) Beam C1, Bending and Compression Test: Normal strains at midspan. (loads in 2-direction).

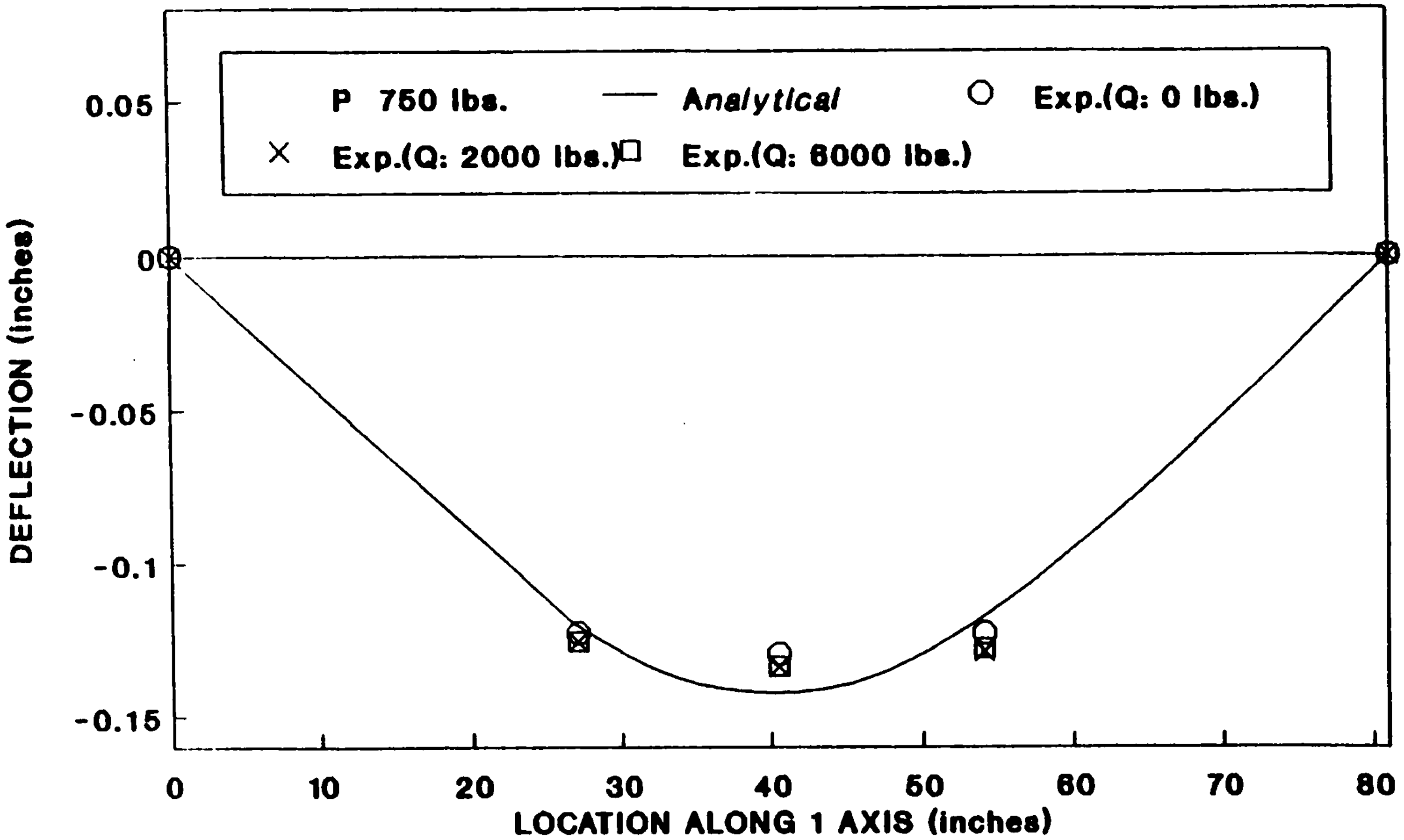


Fig. 5.8(b) Beam C1, Bending and Compression Test: Displacements along the beam.

5.7.2 Beam S3

Two different spans of this straight beam were tested: 83 and 132 inches. For both spans, the beam was tested in bending about both principal axes of the cross section (flatwise and edgewise bending).

1. Bending. The strains were recorded at the midspan, and the deflections were measured at L/3, L/2, and 2L/3. Figs. 5.9(a), (b), (c) and (d) show the analytical and experimental results for the 83-inch span and Figs. 5.10(a), (b), (c), and (d) for the 132-inch span. The finite element analyses predict quite well the average experimental responses of the beam. The displacement predictions are in excellent agreement with the measured values. Discrepancies in normal strains (of about 20%) occur only at isolated points (compression side of Fig. 5.9(a) and tension side of Fig. 5.10(a)). Disregarding such points, the overall differences are within 10 to 15%, which are usually acceptable for timber.

2. Bending and compression. Only the results for the 130-inch span are discussed here. The beam was loaded in bending by two 750 lbs concentrated loads placed unsymmetrically at 74.25" and 102.25" from the left support (Fig. 5.11(a)). The bending load was held constant and axial loads were applied in increments. The results for three load levels are shown in Figs. 5.11(a), (b), (c), and (d). Once again, the finite element analysis characterizes the overall response of the beam quite well, and significant discrepancies in normal strains are observed only at a few points in the cross-section. The maximum differences in displacements are about 15%.

Based on the results shown in this section and those discussed by Yadama (1989), it is concluded that the isobeam element can accurately predict the overall linear response of curved and straight southern pine glulam beams. The application of the isobeam to the analysis of glulam timber domes is illustrated in the next chapter.

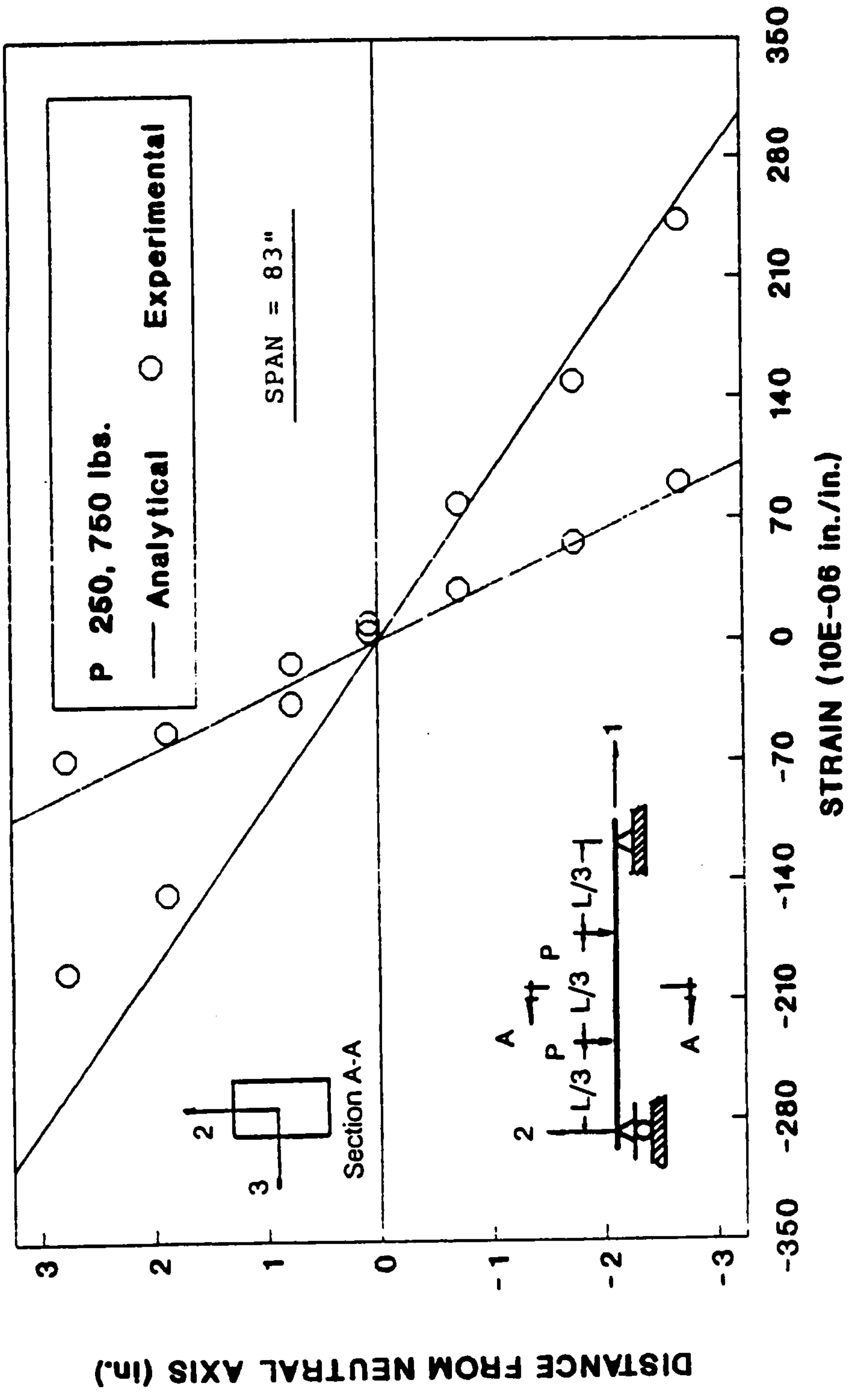


Fig. 5.9(a) Beam S3, Bending Test: Normal strains at midspan. (loads in 2-direction).

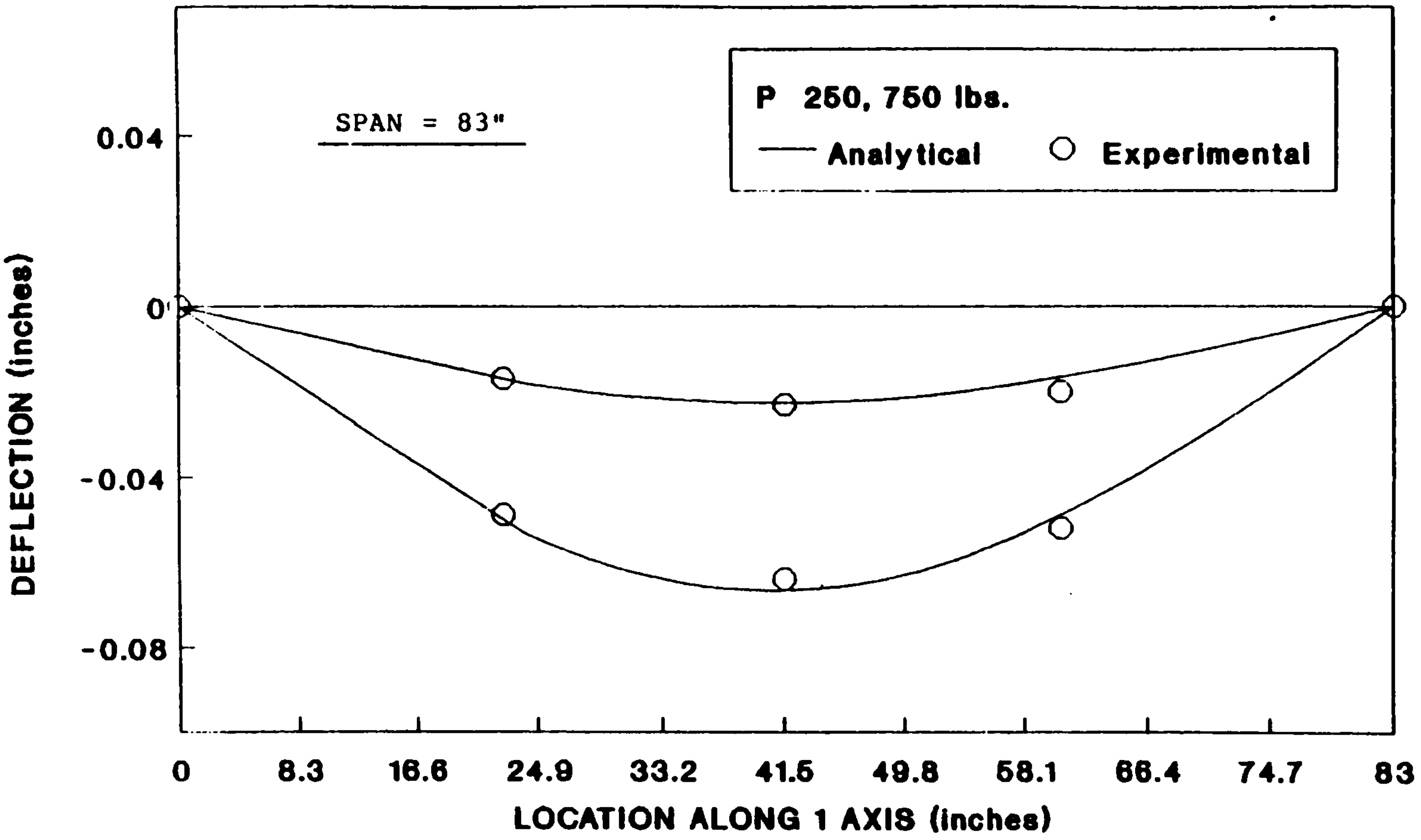


Fig. 5.9(b) Beam S3, Bending Test: Displacements along the beam.
(loads in 2-direction).

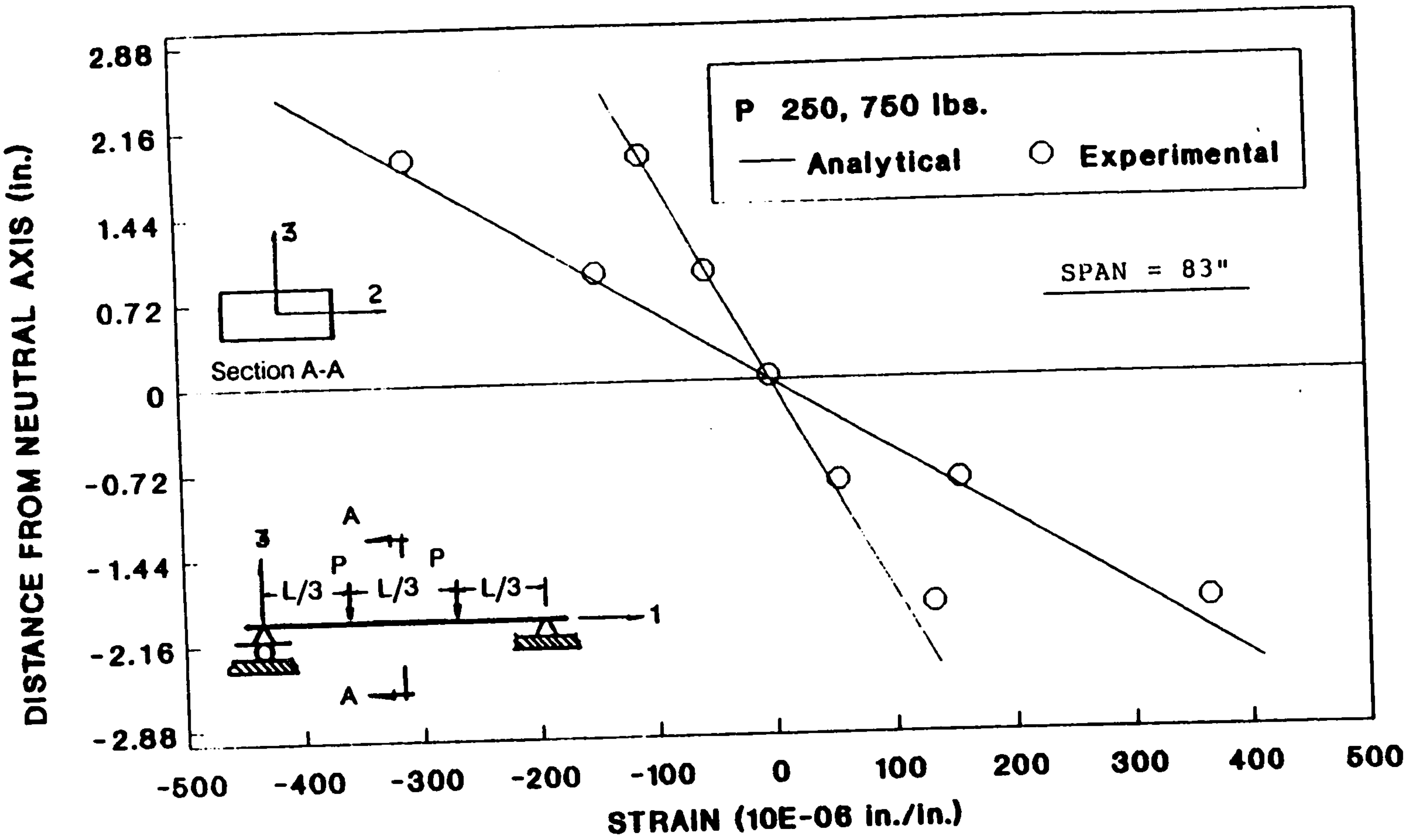


Fig. 5.9(c) Beam S3, Bending Test: Normal strains at midspan.
(loads in 3-direction).

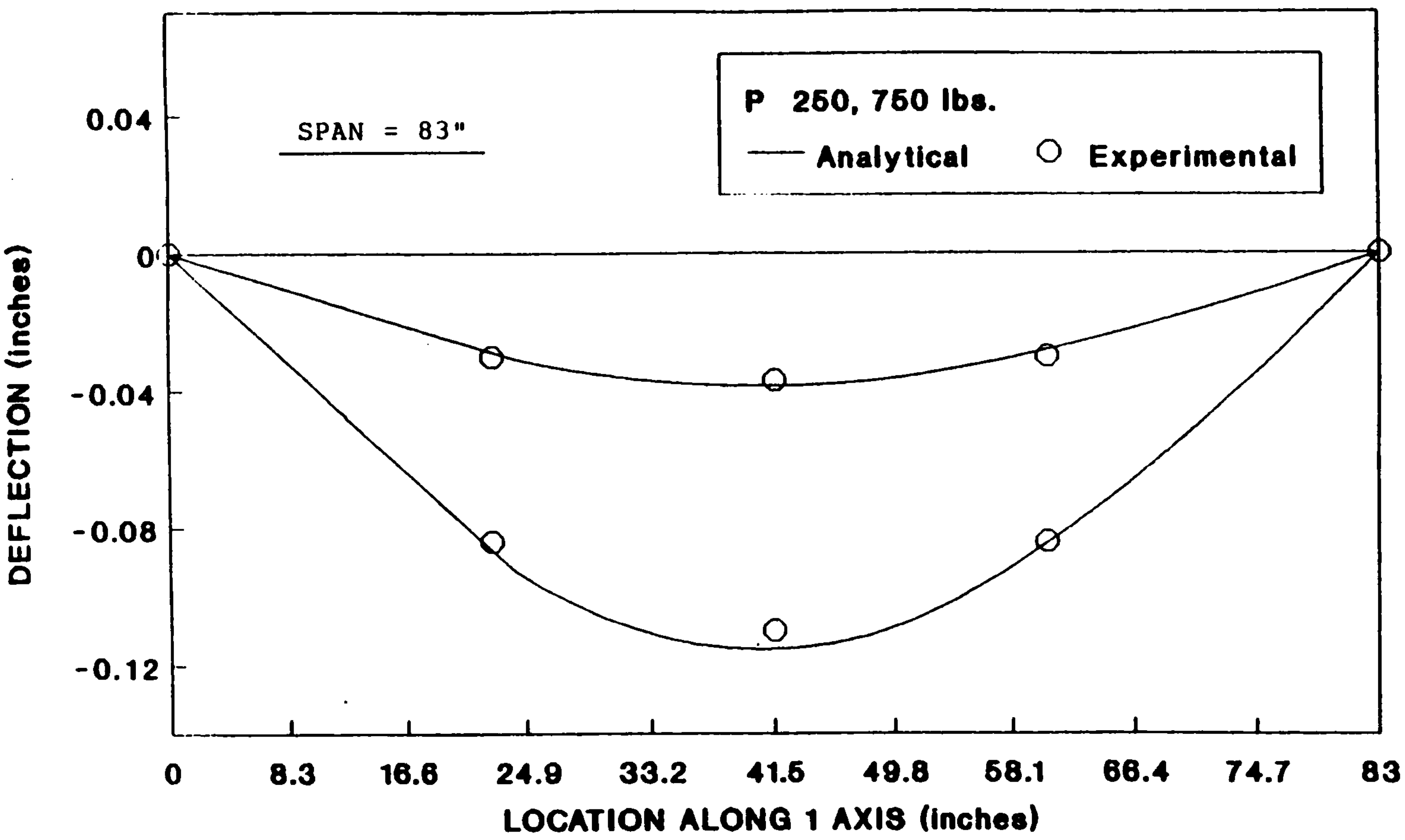


Fig. 5.9(d) Beam S3, Bending Test: Displacements along the beam. (loads in 3-direction).

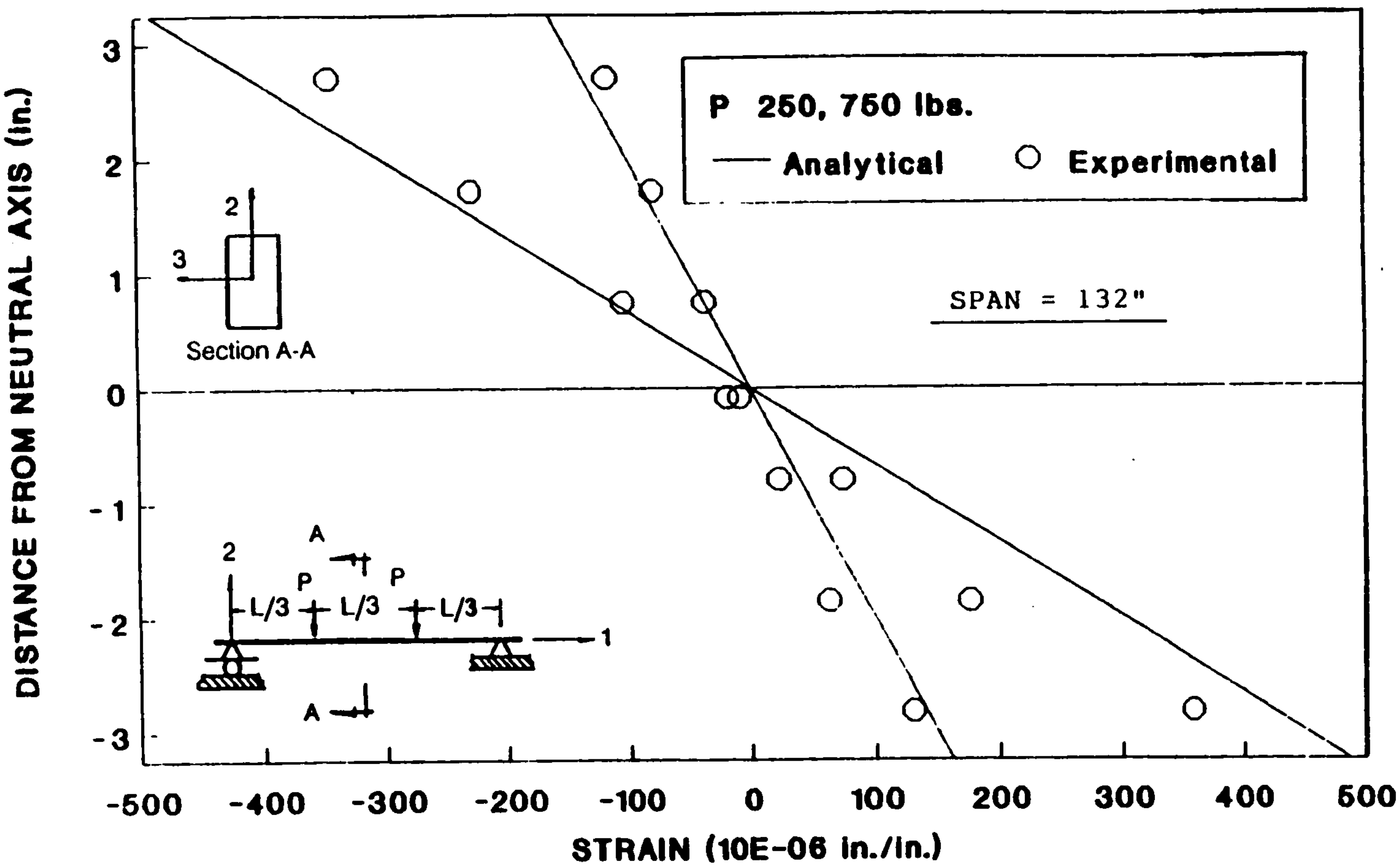


Fig. 5.10(a) Beam S3, Bending Test: Normal strains at midspan. (loads in 2-direction).

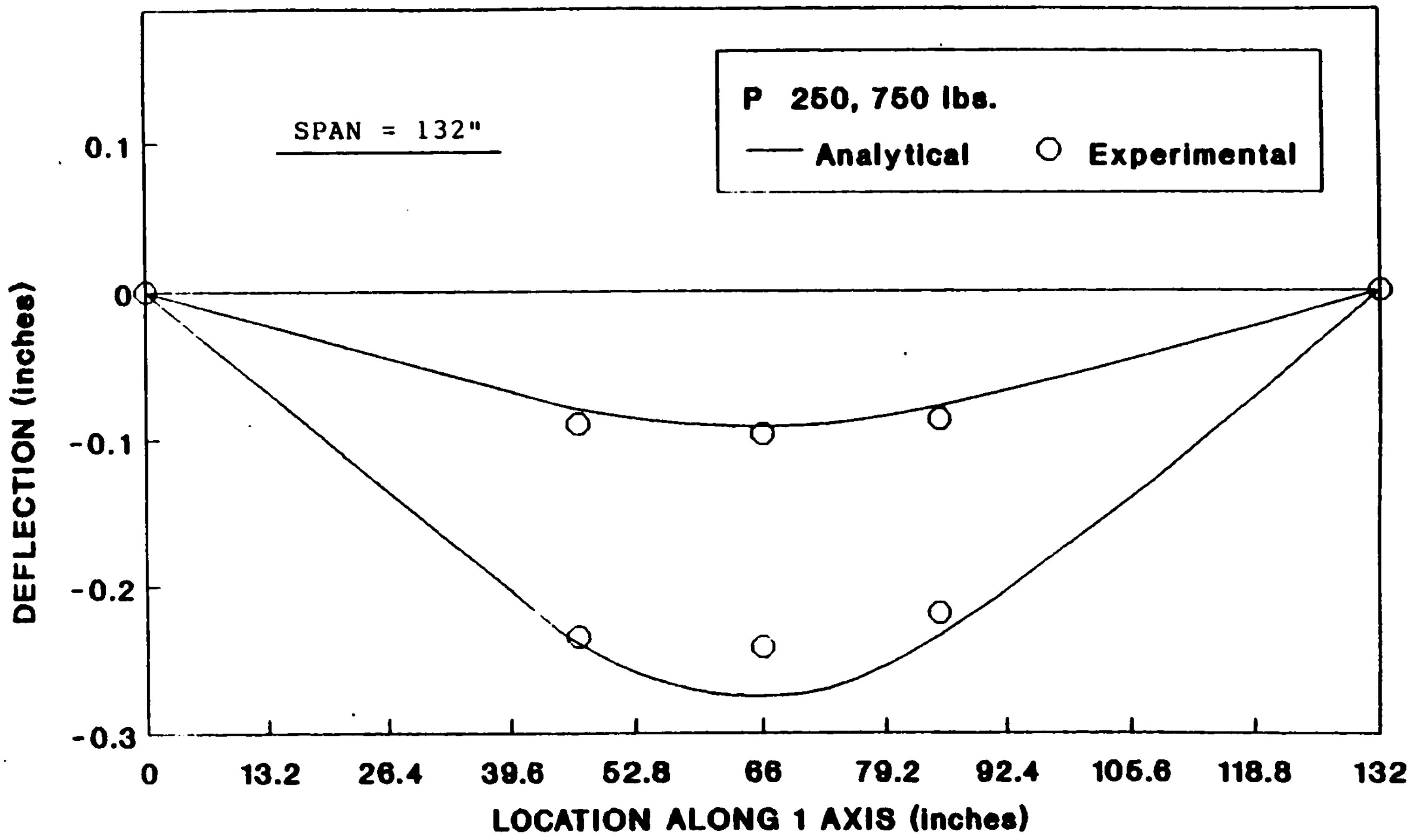


Fig. 5.10(b) Beam S3, Bending Test: Displacements along the beam.
(loads in 2-direction).

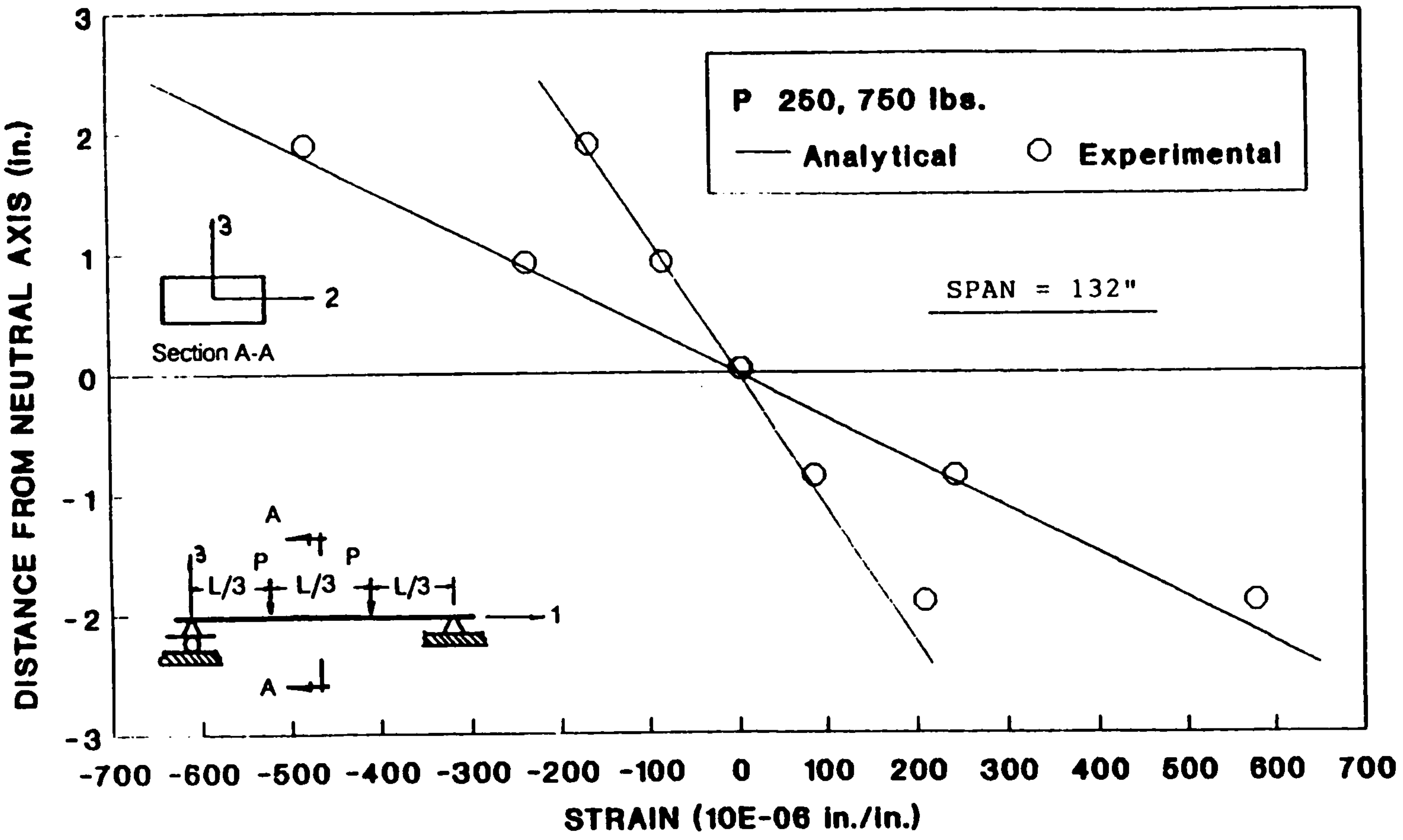


Fig. 5.10(c) Beam S3, Bending Test: Normal strains at midspan. (loads in 3-direction).

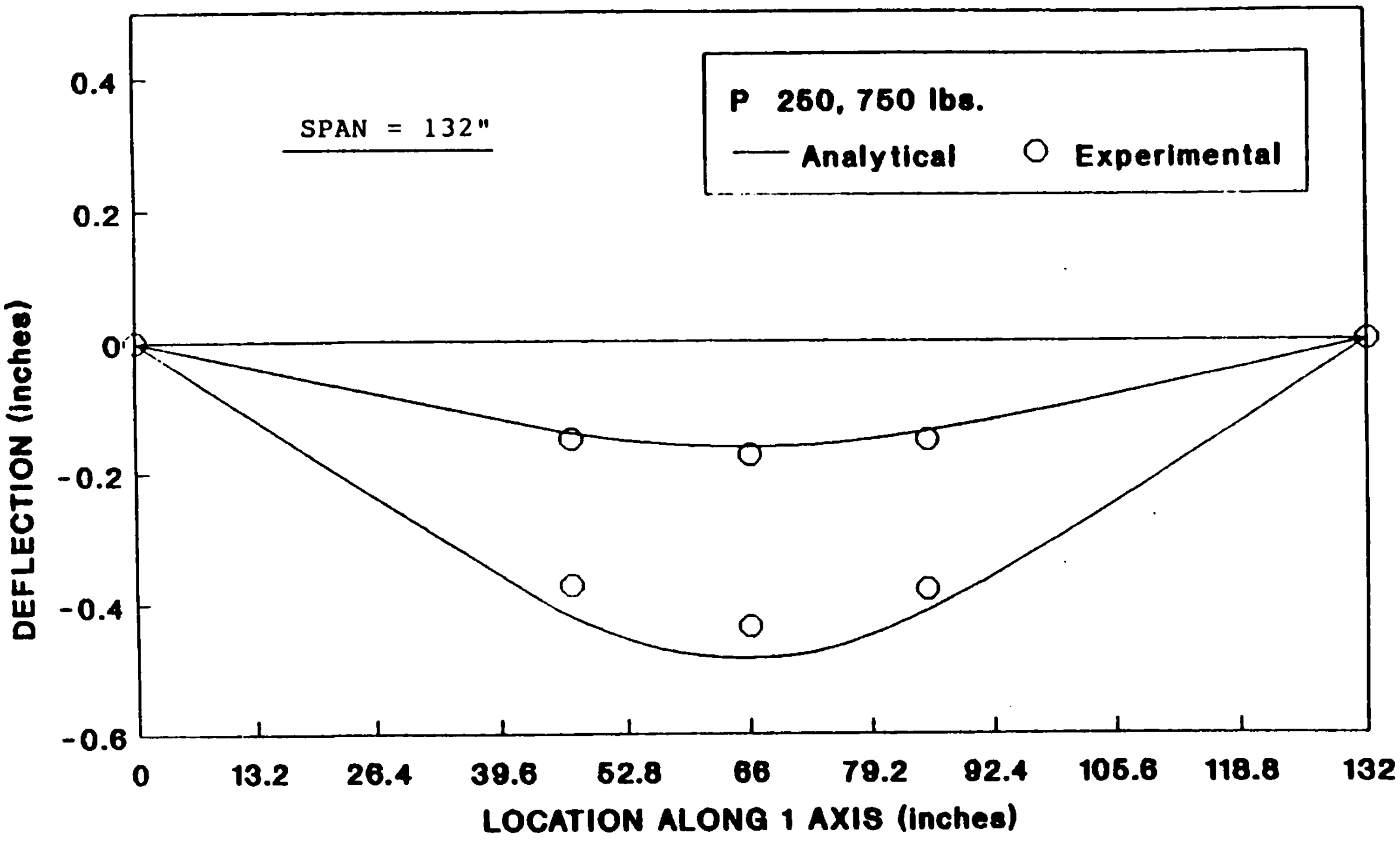


Fig. 5.10(d) Beam S3, Bending Test: Displacements along the beam. (loads in 3-direction).

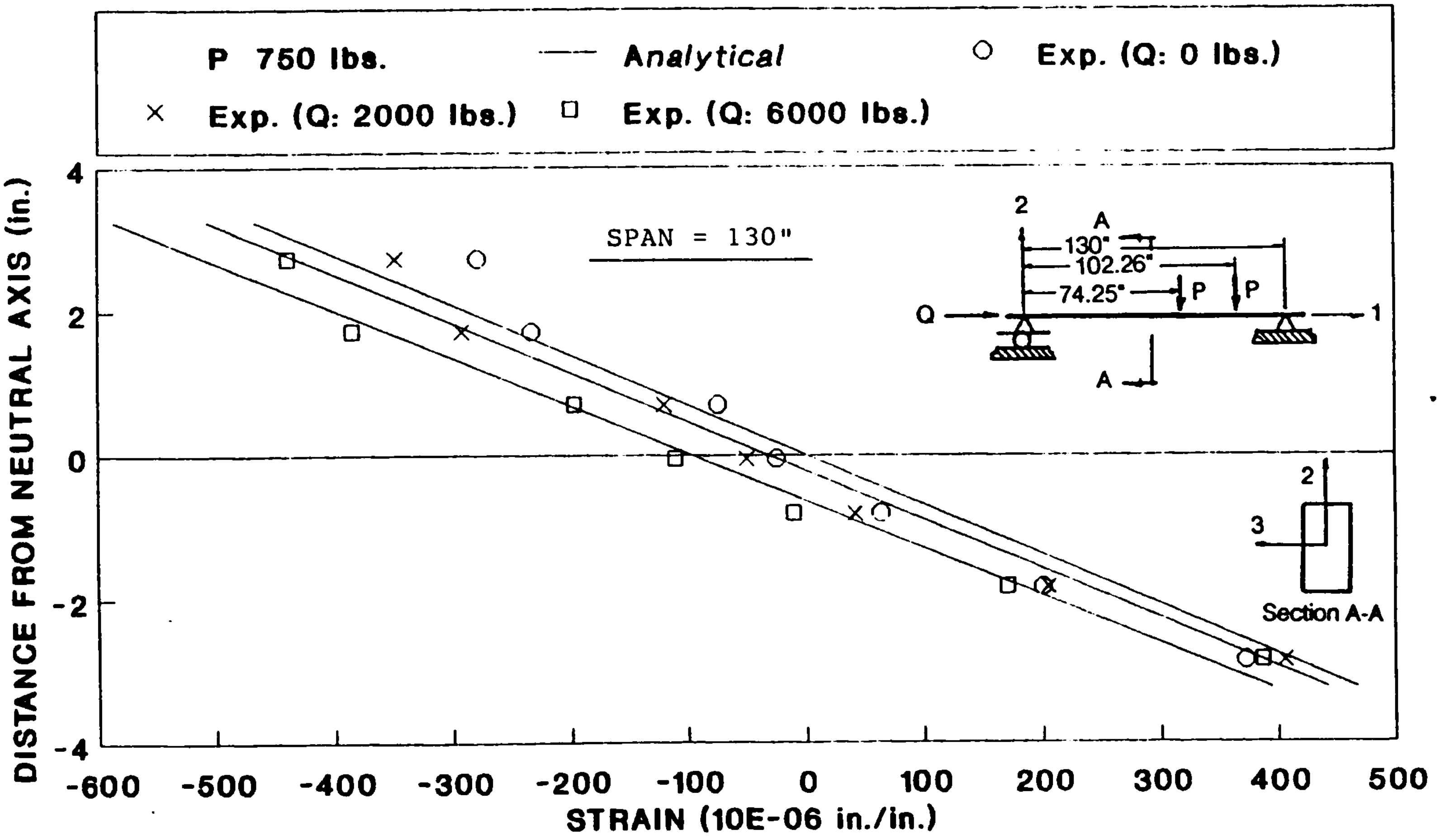


Fig. 5.11 (a) Beam S3, Bending and Compression Test: Normal strains at midspan.
(loads in 2-direction).

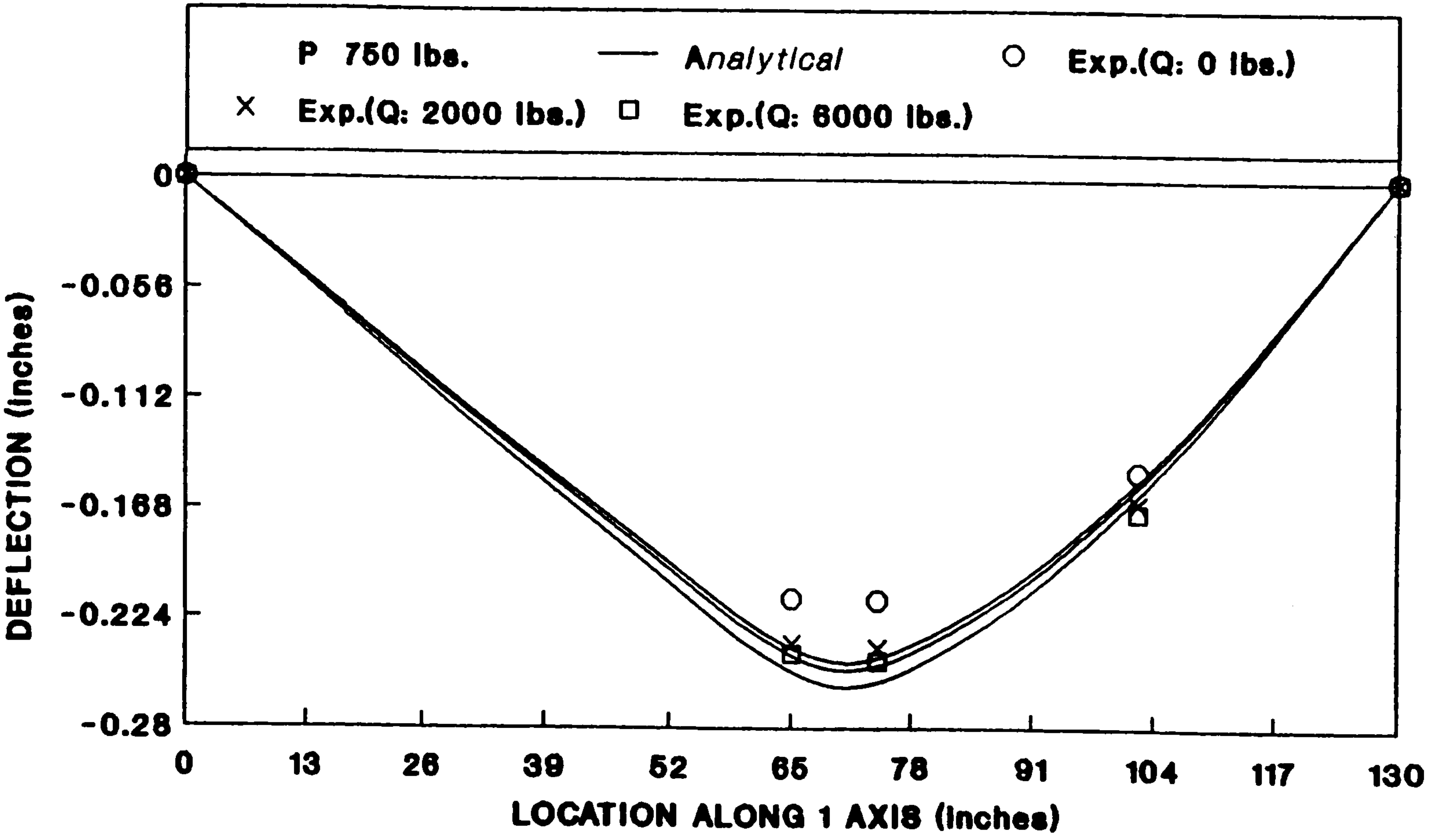


Fig. 5.11(b) Beam S3, Bending and Compression Test: Displacements along the beam.
(loads in 2-direction).

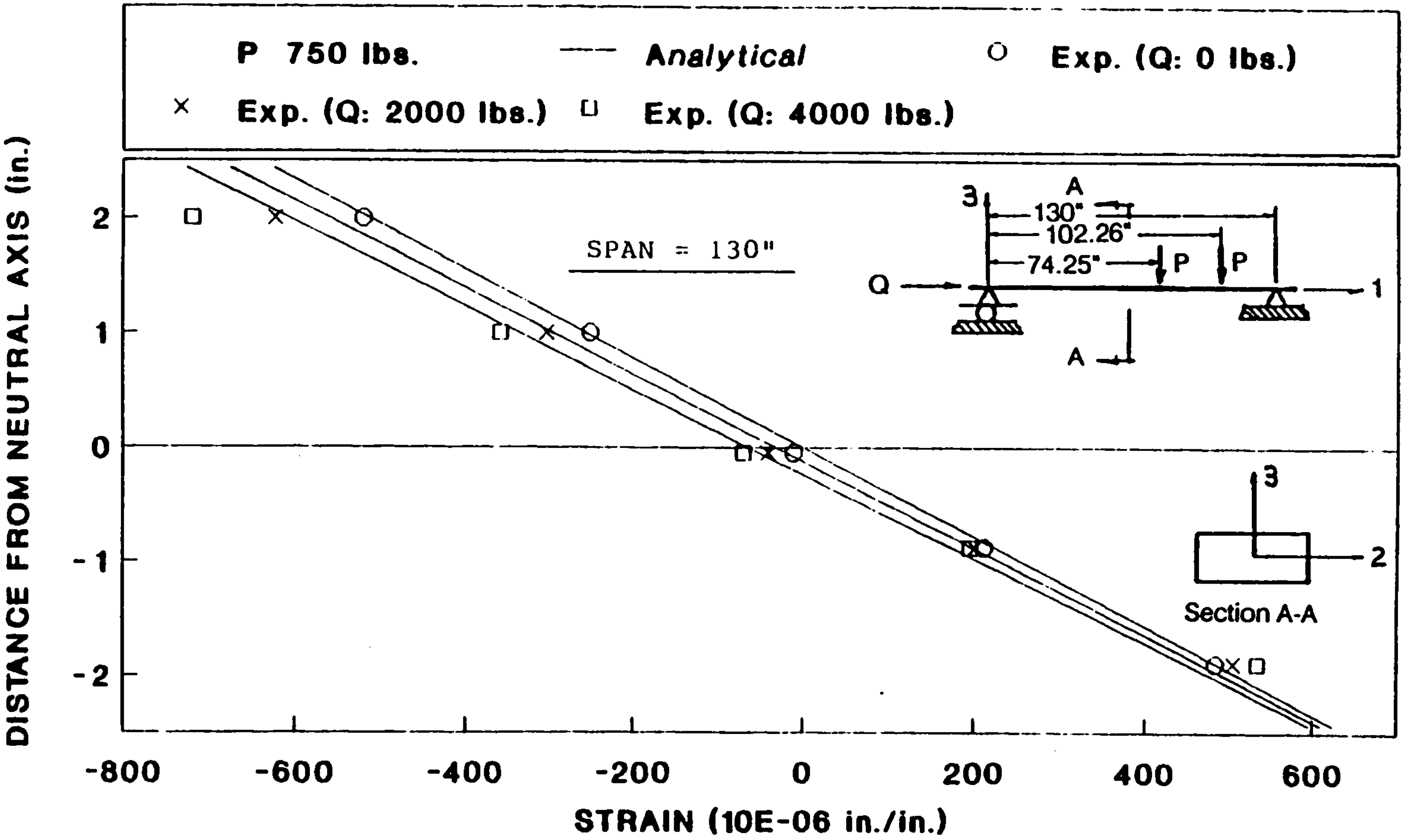


Fig. 5.11(c) Beam S3, Bending and Compression Test: Normal strains at midspan.
(loads in 3-direction).

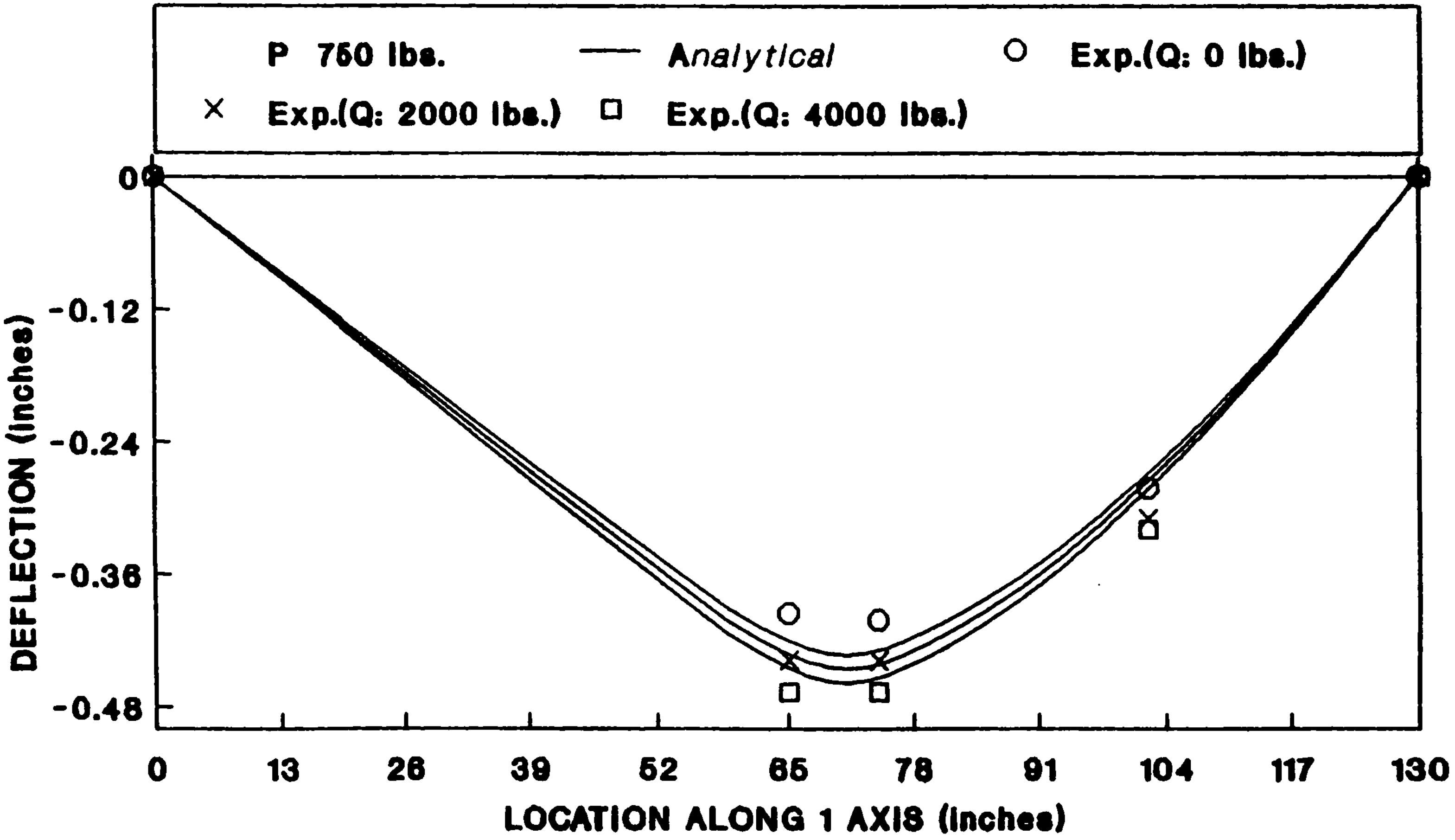


Fig. 5.11(d) Beam S3, Bending and Compression Test: Displacements along the beam.
(loads in 3-direction).

Chapter 6

ANALYSIS OF GLULAM TIMBER DOMES

6.1 Introduction

This chapter presents the finite element analysis results of a glued-laminated timber dome model, whose geometry is based on a Triax glulam dome built in Raleigh, North Carolina in 1975. Glued-laminated timber domes, such as the Varax and Triax grid domes, are single-layer lattice (reticulated) structures whose nodes lie on a spherical surface.

In this chapter, the general geometry and characteristics of lattice domes are presented (Section 6.2.1), including a detailed discussion of Triax and Varax domes (Section 6.2.2). Elastic stability is the dominant failure mode of single-layer metallic space frames (Section 6.2.3). Thus, a fundamental question that is answered in this study is whether or not the nonlinear response of the glulam dome model is governed by instability (Section 6.7.2). The analysis of the dome model requires defining the geometry of the dome (Section 6.3.1), modeling the components, such as beams, purlins, joints, and tension ring (Section 6.3.2), defining the boundary conditions for the dome and a sector of the dome (Section 6.3.3), specifying the required material properties (Section 6.3.4), and determining the finite element mesh (Section 6.3.5).

Important design loads for glulam domes are dead load, live loads, snow loads, wind loads, earthquake loads, and snow concentrations and special hanging loads (Section 6.4). Various load combinations must be considered in the analysis of glulam domes (Section 6.5). Glulam timber domes are covered with a tongue-and-groove wood decking nailed to the beams and purlins. In this study, the load-carrying-capacity of the decking is not considered. Thus, the distributed loads applied to the decking are transformed into discrete member loads (Section 6.6). The analysis of the dome model consists of three distinct processes: a linear analysis, an eigenvalue buckling prediction, and an incremental iterative nonlinear analysis (Section 6.7.1). The response evaluation of the dome model for three load conditions and a discussion of the results is presented (Section 6.7.2).

The information presented in this chapter, particularly Sections 6.2 and 6.7.1, follows closely the writings of Holzer (Holzer and Loferski, 1987; Holzer et al., 1989; Holzer and Huang, 1989) and the USDA proposal on which this study is based (Holzer and Loferski, 1986).

6.2 Glulam Timber Domes

In this section, the general geometry and characteristics of lattice domes are presented (Section 6.2.1). Varax and Triax glulam domes are single-layer reticulated structures whose nodes lie on a spherical surface (Section 6.2.2). As illustrated in Section 6.2.3, the dominant failure mode of single-layer metallic domes is elastic instability, which is governed by geometrically nonlinear behavior. Thus, it is expected that the ultimate load capacity of glulam domes be governed by geometric nonlinearities rather than by material failure. The results of the finite element analysis of a glulam dome model in Section 6.5 show that this is the case.

The information presented in this section is based in part on a paper by Holzer and Loferski (1987).

6.2.1 Lattice Domes

Lattice domes, reticulated domes, or braced domes are three-dimensional networks of elements with their nodes contained on a surface of revolution, usually a spherical surface. The essence of their design is the three-dimensional load-carrying mechanism. Lattice domes can be classified on the basis of their configurations as Schwedler, lamella grid, parallel lamella, and geodesic domes (Fig. 6.1). The relative merits of these structures are discussed in the literature (Holzer and Loferski, 1987).

The elements of lattice domes are made of steel, aluminum, or glued-laminated timber. The connectors range from the sophisticated Mero connector, which provides a rigid concentric connection, to simple bolted connections that provide little flexural or torsional resistance (Holzer and Loferski, 1987). The selection of appropriate connectors is fundamental to the successful design and construction of space structures (Tsuboi, et al., 1984).

Most lattice domes are single-layer domes with short spans. However, there are a few examples of single-layer domes with spans exceeding 330 ft. (Makowski, 1984). Surprisingly, the Tacoma glulam timber dome, the largest timber dome and the largest clear-span wood roof structure in the world, has a clear span of 530 ft. (Pacific Builder and Engineer, 1983).

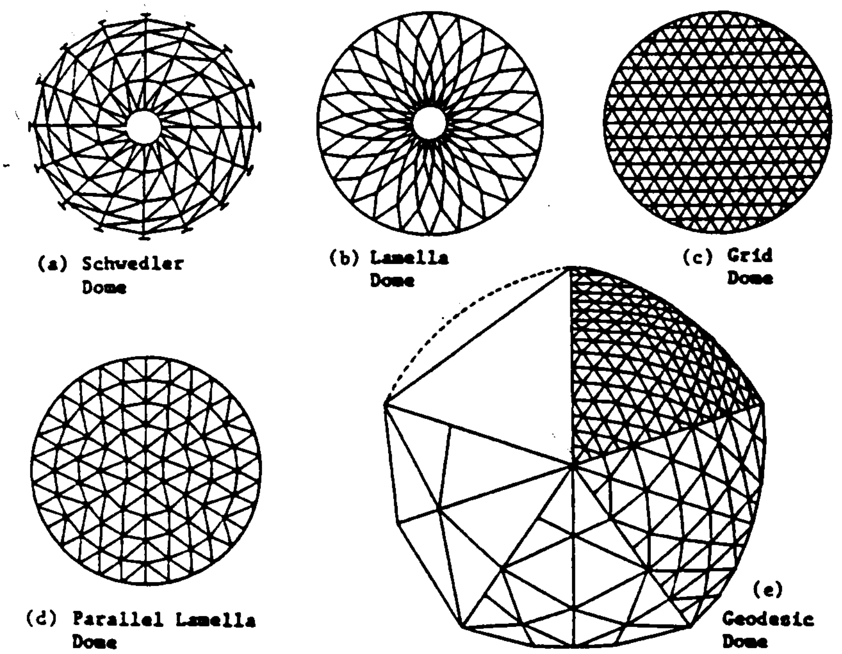


Figure 6.1. Classification of lattice domes (Tsuboi et al., 1984). Plan views

6.2.2 Triax and Varax Domes

Triax (Neal, 1973) and Varax (Eshelby and Evans, 1988) domes consist of curved glulam timber beams arranged in a network of triangular modules interconnected by patented steel hubs (Fig. 6.2). The basic geometry of these domes is the three-way grid pattern which is obtained by projecting a network of equilateral triangles, contained in the plane of the tension ring, onto the spherical surface of the dome by rays originating at the center of the sphere. Thus, all members lie in great circle planes. This procedure generates fairly uniform networks of curved elements on the surface of the sphere, but it is limited to relatively shallow domes (e.g., the 502 ft Skydome of Northern Arizona State University at Flagstaff). For example, the development of the geometry of the Tacoma dome (530 ft) required a modified procedure which consisted in adding a zone of equilateral triangles to the outer edge of a basic grid pattern. The triax geometry has been extended to rectangular base domes by using projecting rays with different origins and radii. This results in sets of members, rather than all members, with identical radii. This idea was used by Eshelby (1988), the chief designer of the Tacoma dome, to design the Charles Wright Academy gymnasium dome (140 ft by 220 ft rectangle with 30 ft radius corners) in Tacoma, Washington, which replaced an air-supported roof structure.

The differences in the design of the steel hub connector mark the differences between the Varax and the Triax domes. The Varax joint (Fig. 6.3(a)) has flexural and rotational stiffness. Based on tests carried out at the University of Washington, the flexural stiffness of the connector is taken as 85% of a rigid connection (Eshelby and Evans, 1988). The Triax joint (Fig. 6.3(b)) transmits only compressive forces. For both domes, the tongue-and-groove decking (Fig. 6.2) curved to the dome's radius and nailed to beams and purlins, forms an integral part of the design. Steel or reinforced concrete tension rings at the base of the dome complete the design.

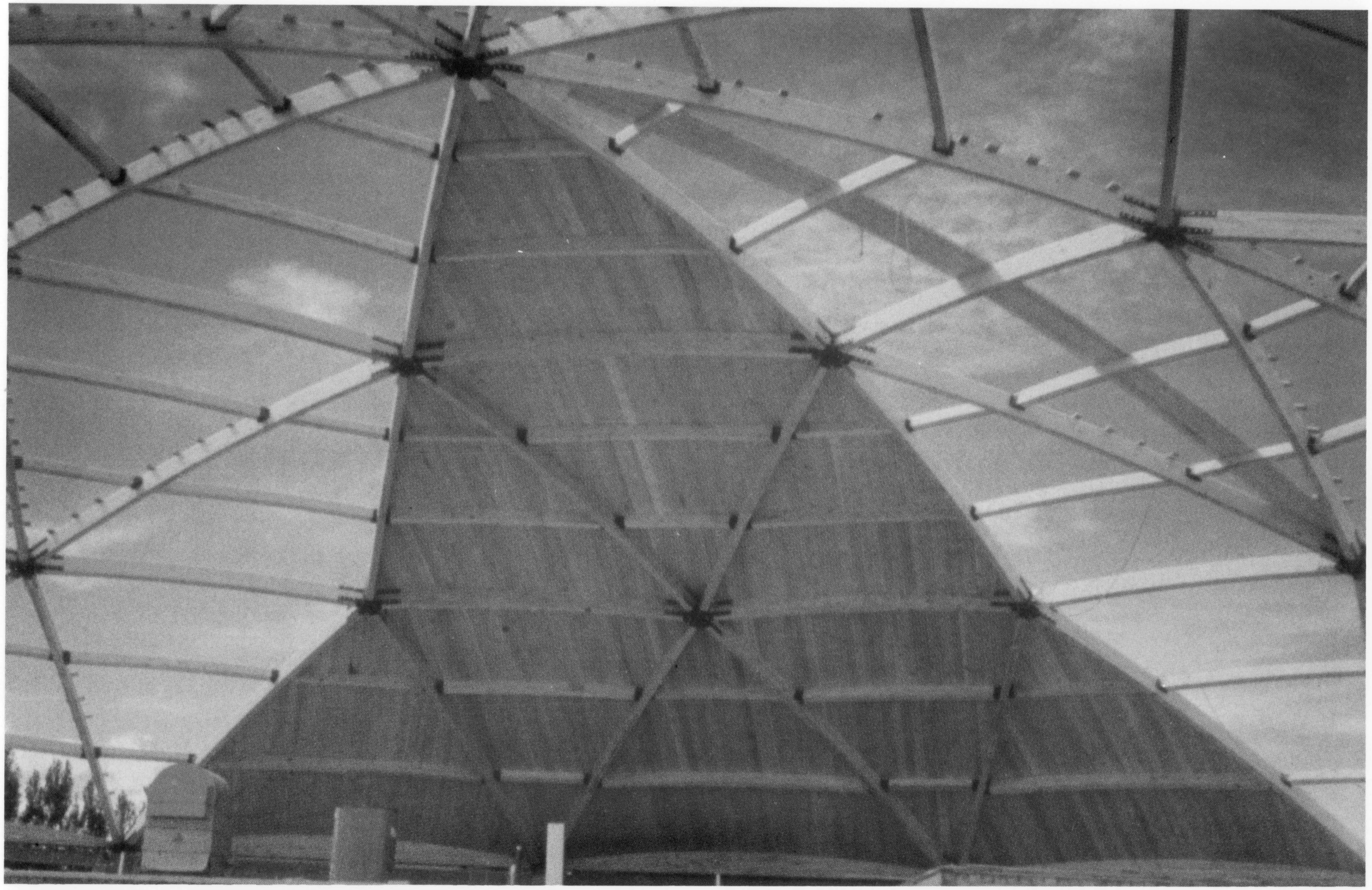
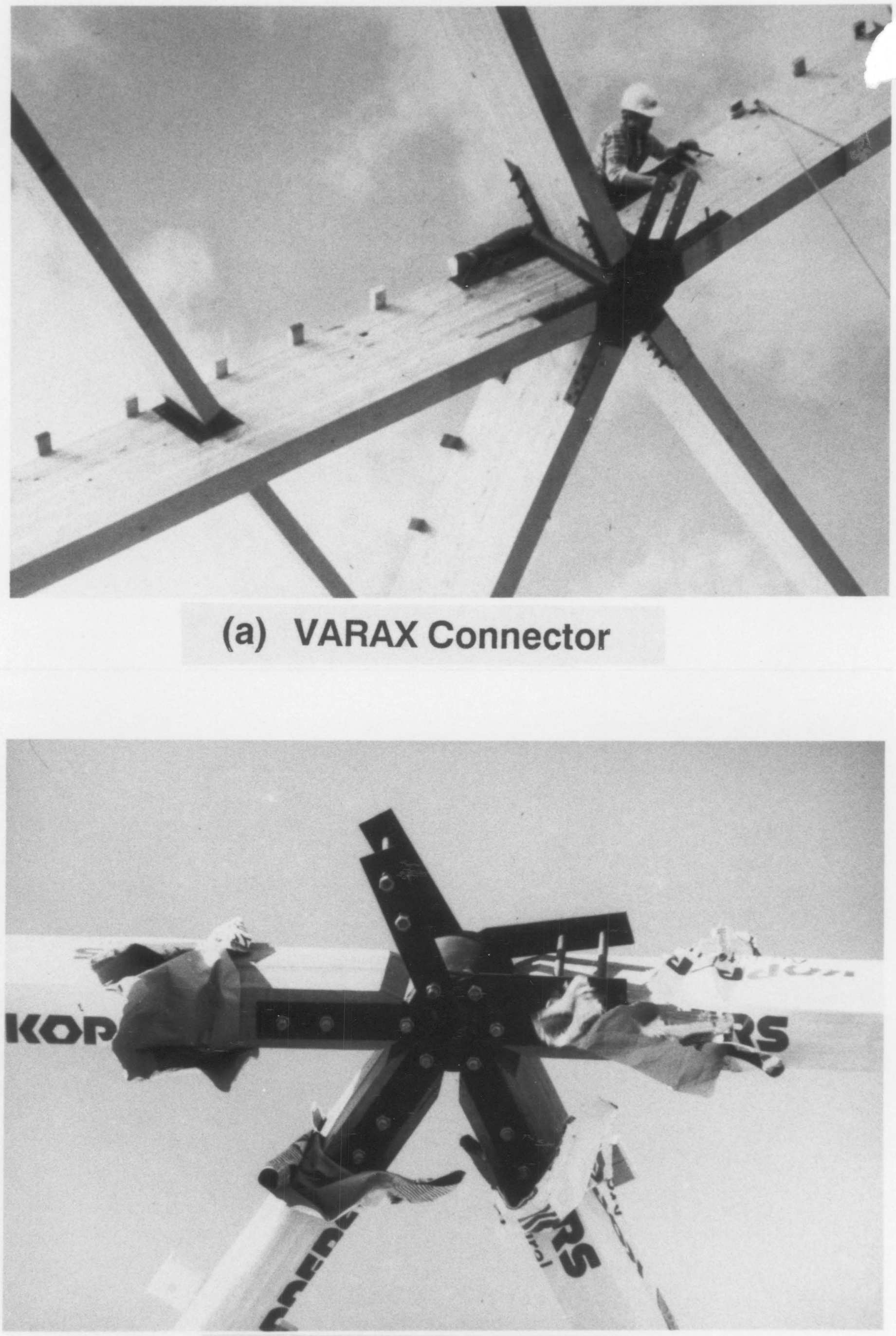


Figure 6.2. Details of a Varax glulam dome.



(a) VARAX Connector

(b) TRIAX Connector

Figure 6.3. Details of the Varax and Triax steel hub connectors.

6.2.3 Failure Modes of Lattice Structures

The ultimate load carrying capacity (i.e., the failure mode) of lattice structures is governed by three kinds of behavior (Tsuboi, et al., 1984, p. 48): elastic buckling, elasto-plastic buckling, and limit strength. These types of behavior are influenced by the geometric and material nonlinearities of the structure. The first type is characteristic of single-layer lattice structures and the last two types correspond to double-layer lattice structures (Fig. 6.4).

For single-layer domes, stability represents an important design criterion (Holzer and Loferski, 1987). For example, experts concluded that the failure of the Bucharest dome (see Section 6.4) was induced by elastic instability (Soare, 1984). Moreover, in a recent state-of-the-art report on space frames (Tsuboi et al., 1984), it was stressed that geometrically nonlinear behavior plays the key role in elastic buckling, the dominant failure mode of single-layer domes (Fig. 6.4). Since glulam timber domes (e.g., Triax and Varax domes) are single-layer space frames, the failure mode of these structures should also be governed by elastic instability. Thus, it is expected that geometric nonlinearities, rather than material nonlinearities, will dominate the ultimate load capacity of glulam lattice domes. This is a fundamental assumption to be tested in this study (Section 6.5).

6.3 Finite Element Modeling

The finite element modeling of glulam timber domes involves:

- the generation of the geometry of the dome, including the orientations of the interconnected space beams;

- the selection of finite elements to model the beams, connections, decking, and tension ring;

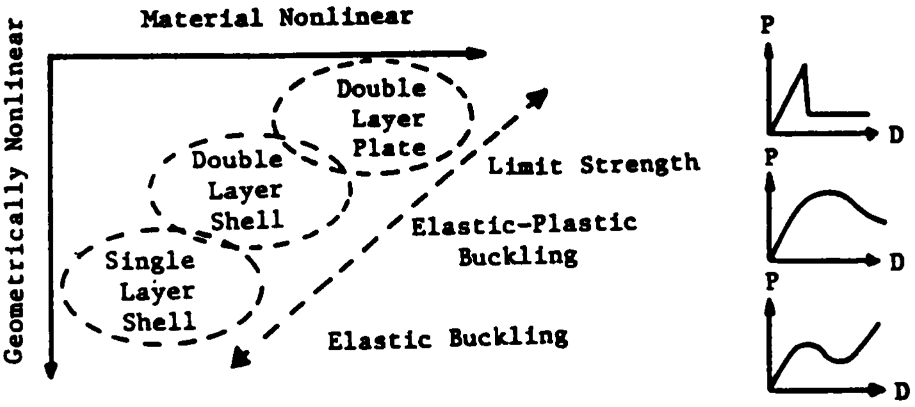


Figure 6.4. Nonlinearity and failure modes of lattice structures (Tsuboi et al., 1984).

- the specification of appropriate boundary conditions for the entire dome and for symmetric substructures (e.g., a sector of the dome);

- the specification of material properties;

- the determination of a sufficiently accurate mesh.

These modeling considerations are discussed in this section in relation to a dome model whose geometry is based on a Triax dome built in Raleigh, North Carolina in 1975.

6.3.1 Dome Model

In this study, the finite element modeling and analysis of glulam timber domes is limited to a dome model whose geometry is based on the Crafts Pavilion Triax dome built in Raleigh, NC in 1975. The dome has a span of 133 ft and a rise above the tension ring of 18 ft (Fig. 6.5). It consists of a 3-d grid system of six identical sectors composed of curved southern pine members: 132 beams and 108 purlins. The dimensions of the beams and purlins are given in Figure 6.6. The beams are interconnected by 61 steel hubs, and the purlins are connected to the beams by 216 steel hangers (see Fig. 6.3). In the finite element analysis, the joints are modeled with connector elements (Section 6.3.2). The dome model rests on a steel tension ring of 1″x12″ (Fig. 6.6). The Crafts Pavilion Triax dome is covered with a 2″ tongue-and-groove wood decking fastened to the beams and purlins with nails. In this study, the effect of the decking, which is expected to be significant, is not included. In the analysis of the Tacoma Dome (Eshelby and Evans, 1988), the effect of the decking was simulated by constraining each member to deform only in its radial plane.

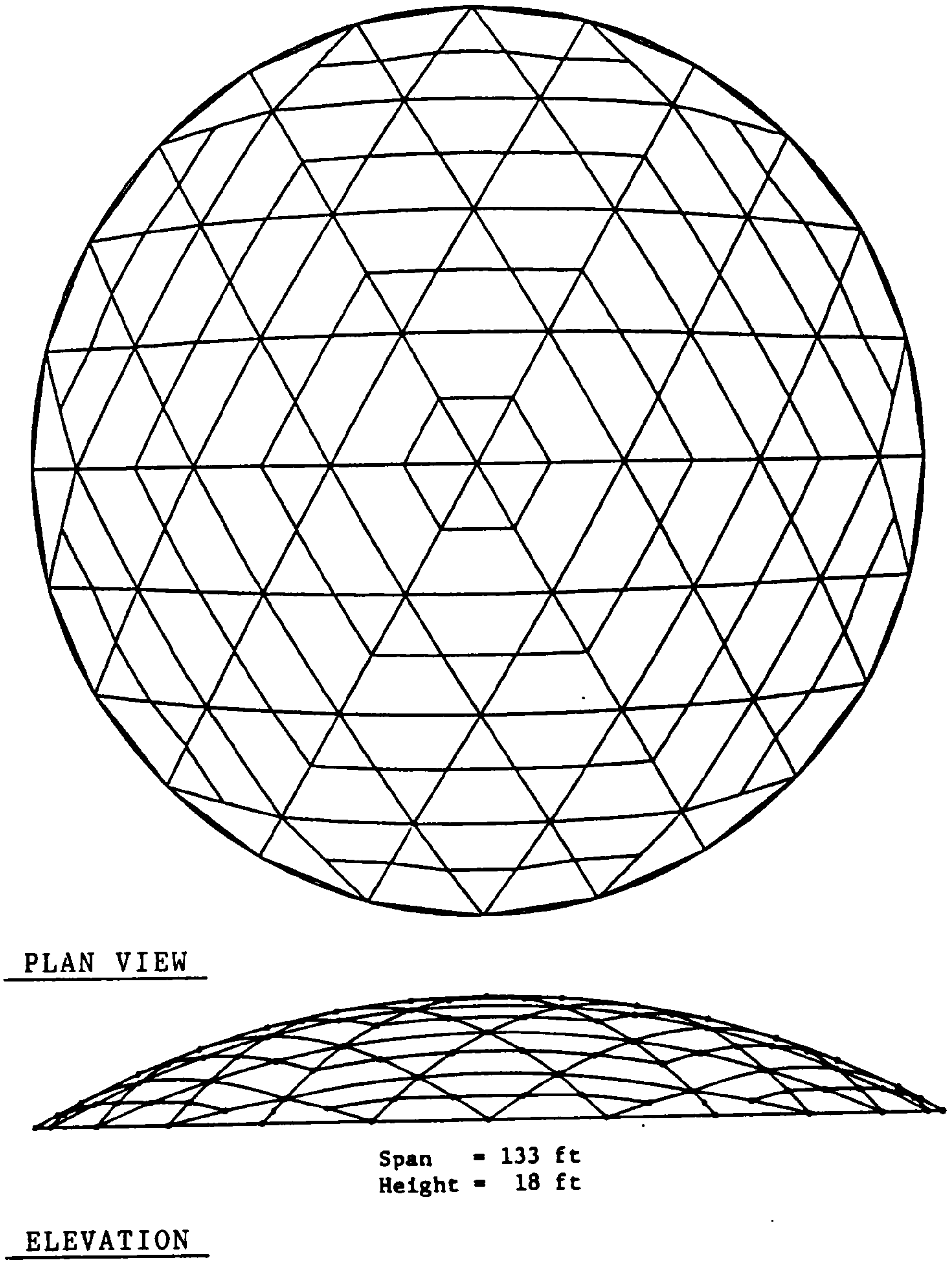


Figure 6.5. Geometry of the dome model: Crafts Pavillion Triax dome, Raleigh, North Carolina (1975).

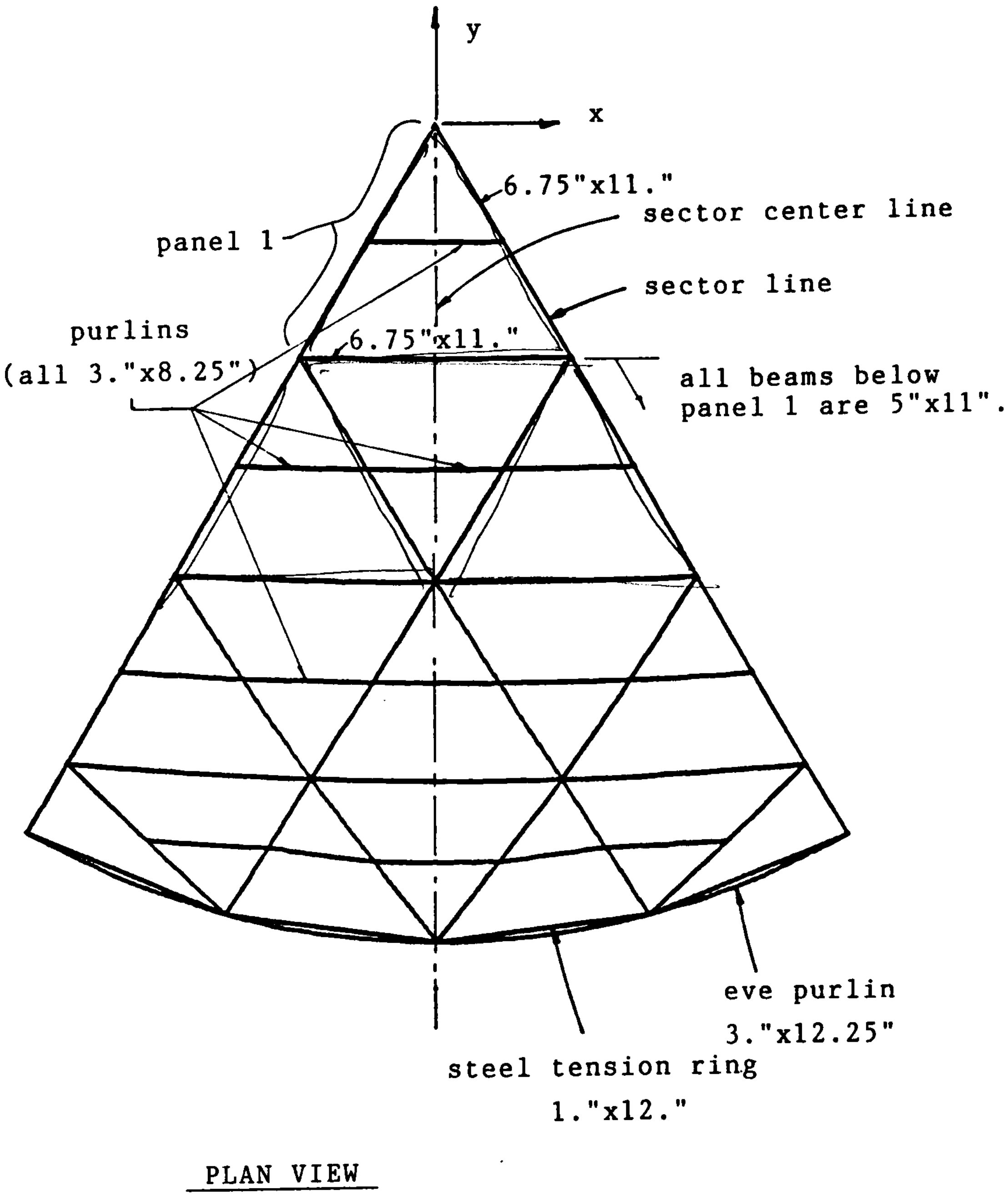


Figure 6.6. Dimensions of beams, purlins, and tension ring of one sector of the Triax dome model.

6.3.2 Modeling of Components

The finite element analysis of the dome model requires the selection of the following finite elements: a beam element to model the curved glulam beams and purlins, connector elements to simulate the flexibility of the joints, and a truss element to model the steel tension ring. Since the program developed in Chapters 4 and 5 can only analyze rigid-jointed-frames and does not include a truss element, the commercial finite element program ABAQUS (version 4.7-1, 1989) is used in this study. This powerful general purpose program contains a library of elements that can satisfy the modeling requirements of the dome. ABAQUS is especially tailored for nonlinear analysis, and some of its key features are automatic time stepping and iteration. It is regarded as one of the easiest large-scale finite element programs to learn and use, and one of the most computationally efficient, even in simple linear applications (Kardestuncer and Norrie, 1987, p. 4-263). The elements selected to model the dome are described next.

1. **Space beam element.** The 3-node isobeam element described in Chapters 4 and 5 is contained in ABAQUS, and it is, therefore, selected to model the beams and purlins. A mesh of 2-elements per member is used (Section 6.3.5).

2. **Connector element.** A 2-node isobeam element is used to model the beam-to-beam and purlin-to-beam connections. Since each steel hub connector joins six beams, the hub connecting the space beams can be modeled by six 2-node isobeam elements joined at a common node (Fig. 6.7(a)). Similarly, a 2-node element can be used to simulate the steel hanger connecting a purlin to a beam. The excentricity between the axes of the beam and the purlin (which is about 1.6") is ignored, and the 2-node element connects the center of the purlin's cross section to the beam axis (Fig. 6.7(b)). ABAQUS permits the user to specify the following parameters for the 2-node isobeam: the longitudinal and shear moduli E_L and G, the area of the cross section A, the cross-sectional principal moments of inertia I_1 and I_2, and the torsional constant J (polar moment of inertia for a circular section). Thus, the axial, flexural, and

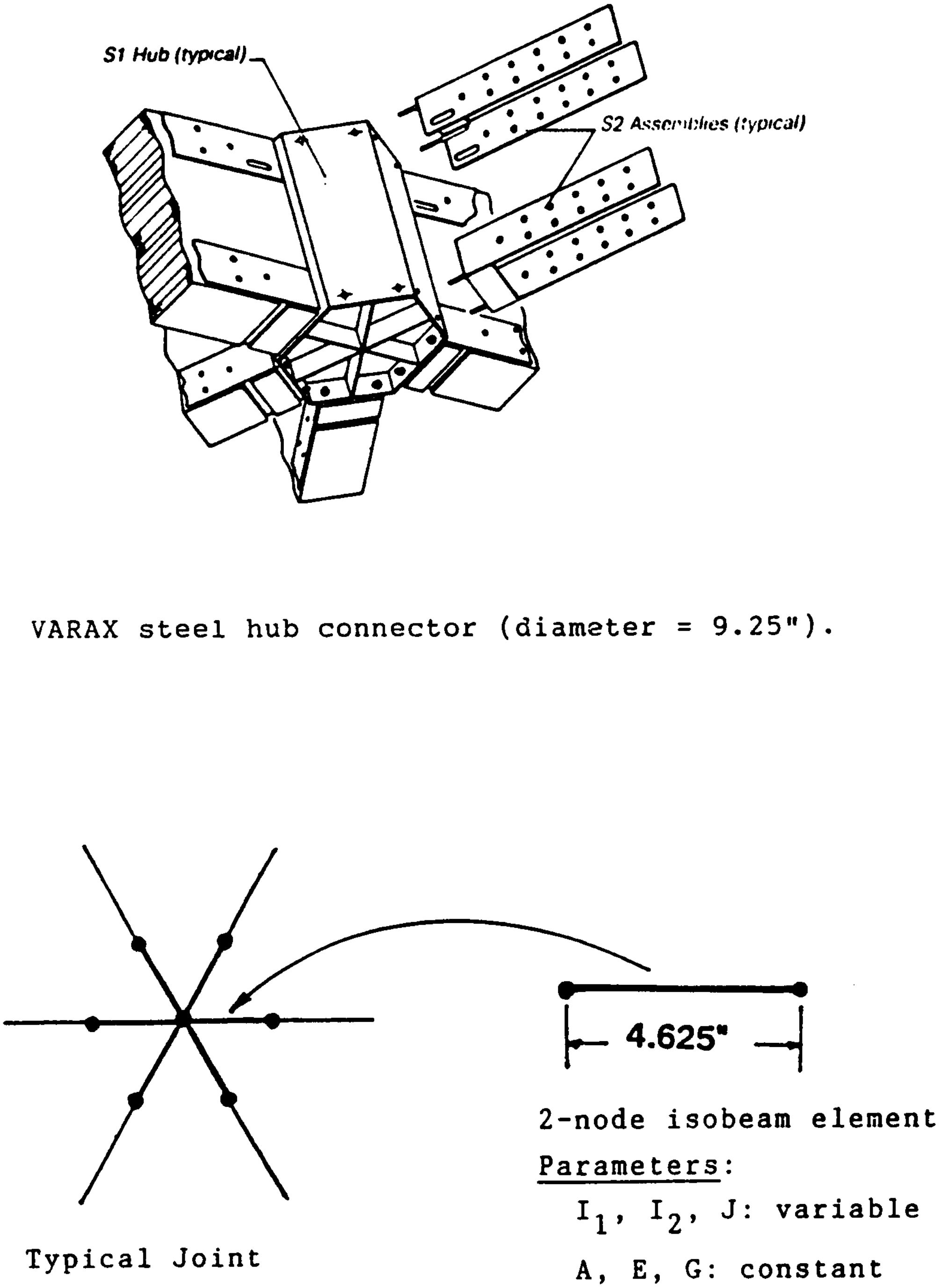
torsional stiffnesses of the connector element can be controlled at will. In this study, the axial stiffness of each connector is the same as that of the connecting member (i.e., A and E are kept constant), whereas the flexural and torsional stiffnesses are decreased, by specifying reduced values for I_1, I_2, and J, to simulate flexible joints.

3. **Truss element.** A three-dimensional, 2-node truss element is used to model the steel tension ring of the dome. Twenty-four elements are used to model the entire tension ring, with each element spanning between two consecutive supports (Fig. 6.6).

6.3.3 Boundary Conditions and Symmetry

One must be careful in specifying constraints to remove rigid-body motions for large frame structures. For example, in the analysis of the M.I.T. dome shown in Fig. 5.5, it was sufficient to impose constraints at the apex of the dome; however, when the same constraints were used for the glulam dome model, they caused singularity due to large rigid-body rotations of the base of the dome. To eliminate rigid-body motions, the supports coinciding with the 1-axis (nodes 1 and 13, Fig. 6.8(a)) are constrained in the 2 and 3 directions, and those along the 2-axis (nodes 7 and 19, Fig. 6.8(a)) are constrained in the 1 and 3 directions. The remaining supports along the base of the dome are constrained in the 3-direction. These constraints allow the tension ring to move radially and are sufficient to avoid rigid-body translations and rotations of the dome.

The dome model consists of six identical sectors. Thus, the geometry of the dome exhibits cyclic symmetry. If the loads are the same for all five sectors (e.g., a uniform distributed load over the dome), it is easier and more efficient to model only one sector of the dome, particularly in linear analysis. In nonlinear analysis, a sector may not represent the response of the dome, because the unsymmetric buckling modes are excluded by virtue of the boundary conditions imposed on the sector (see Section 6.5). To exploit cyclic symmetry, the following boundary conditions along the



VARAX steel hub connector (diameter = 9.25").

Typical Joint

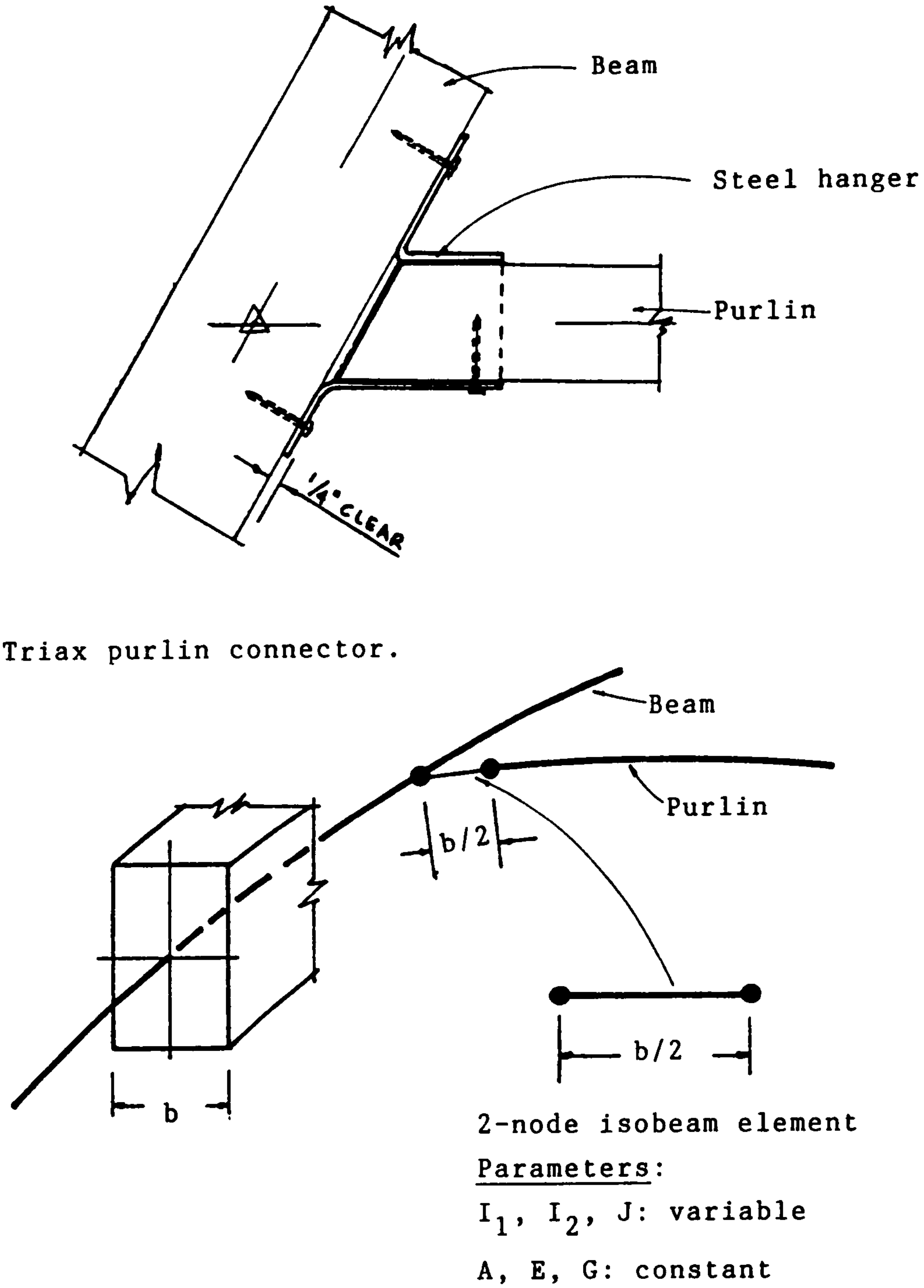
2-node isobeam element

Parameters:

I_1, I_2, J: variable

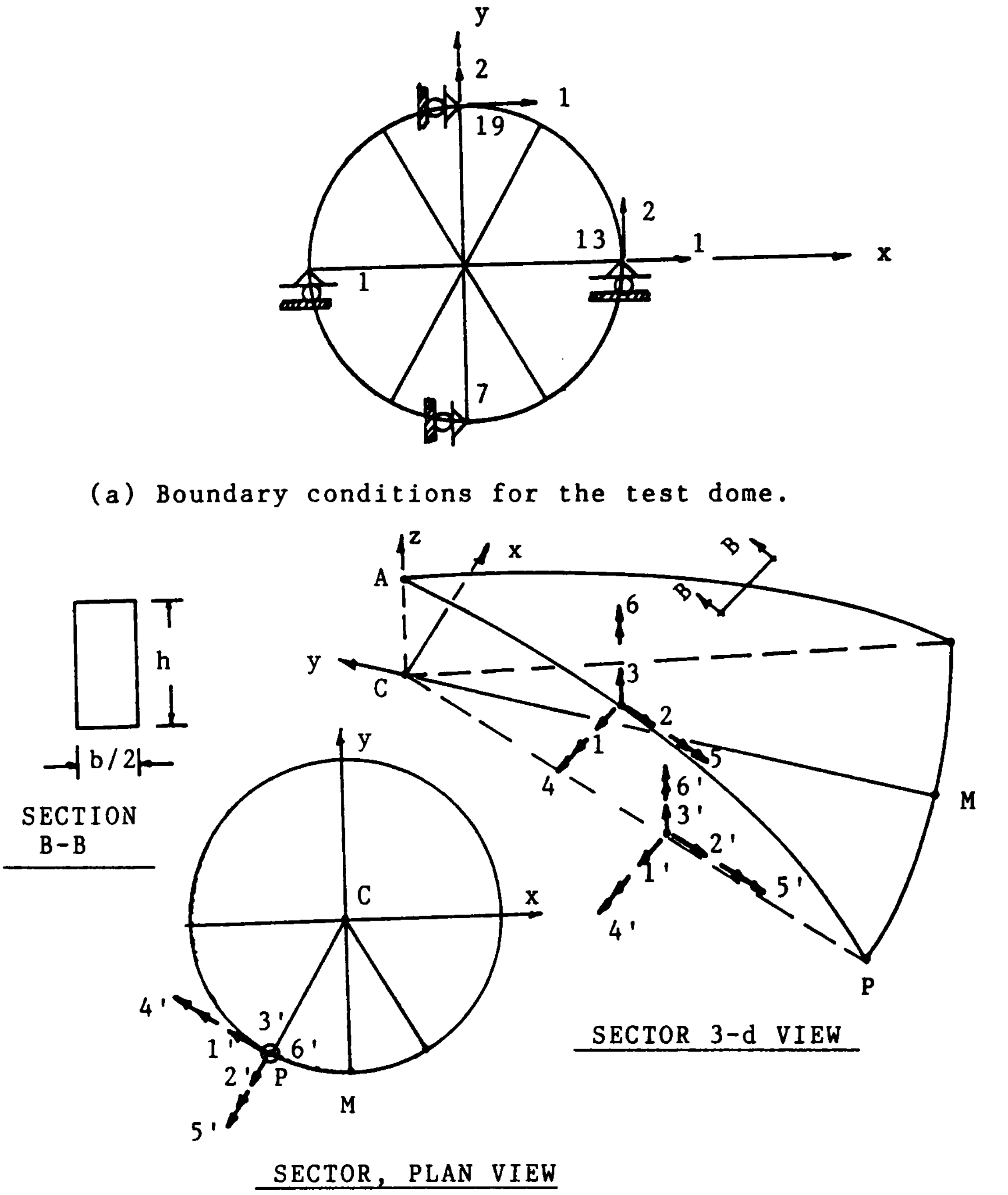
A, E, G: constant

Figure 6.7a. Modeling of the space joints.



Triax purlin connector.

Figure 6.7b. Modeling of the purlin-to-beam connection.



(a) Boundary conditions for the test dome.

(b) Local coordinate systems for a sector

Figure 6.8. Boundary conditions for the whole dome and for one sector.

edges of the sector (sector lines) can be specified (Fig. 6.8(b)): for any node along the sector line, except the node at the support, a local nodal coordinate system (1,2,3) must be defined with the 2-axis tangent to the sector line. At each node, the local coordinate system is unique and the degrees of freedom 1, 5, and 6 are constrained with respect to the nodal frame of reference. The node at the support (node P in Fig. 6.8(b)) is constrained with respect to a local coordinate system $(1', 2', 3')$ that defines the plane of the tension ring. The $2'$-axis coincides with the radial line, CP, and the $3'$-axis with the global 3 direction. Node P is constrainted in the $1', 3', 5'$, and $6'$ directions.

In this study, a simplification is used to define the boundary conditions for a sector. The node at the support (node P) is constrained with respect to the primed local axes. The constraints 1, 5, and 6 for the nodes along the sector line can also be defined in the primed frame of reference (i.e., constraints $1', 5'$, and $6'$). Therefore, only the primed local coordinate system is used to specify boundary conditions along the sector line. The equivalency of the primed and unprimed constraints is based on the following argument: if rotations 5 and 6 are zero, then rotations $5'$ and $6'$ should also be zero (Fig. 6.8(b)). The node at the apex is constrained in global coordinates and it is allowed to displace only in the global 3-direction. The remaining four nodes along the base of the sector are constrained in the global 3-direction. When modeling a sector, the width of the beams along the sector lines must be reduced by half.

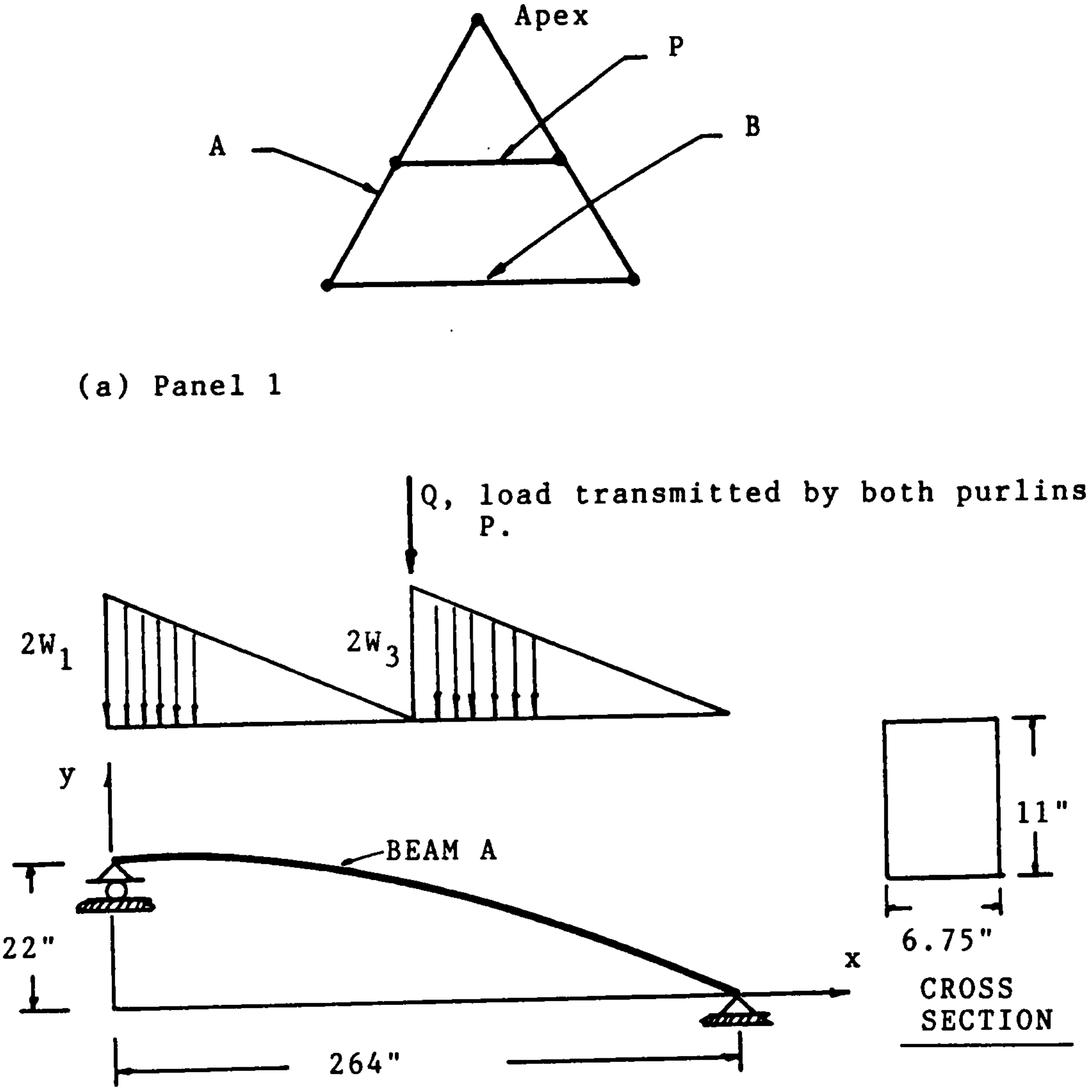
6.3.4 Element Orientation and Material Properties

The cross section of a beam element is oriented in space by specifying the direction cosines of a nodal unit vector contained in the plane of the cross section and parallel to the longer face of the beam (the depth of the cross section). Since all the members above the tension ring lie on great circles of the sphere that defines the surface of the dome, the unit cross-sectional vectors are oriented along radial lines that join the center of the sphere with the node of interest. The eve purlins located on the outside of the tension ring (see Fig. 6.6) lie in the plane defined by the tension ring; their cross-sectional unit vectors are parallel to the global 3-axis.

Based on information provided by the manufacturer of the dome, the glulam beams and purlins are classified as E-rated 56 southern pine (NDS, 1986). Accordingly, the longitudinal elastic modulus E_L is taken as 1.8 x10^6 psi. Tabulated values of the shear modulus of southern pine glulam beams are not available. Therefore, the shear modulus obtained from torsion tests of small glulam samples (Section 3.6), G = 1.6 x 10^5 psi, is used for the beams of the dome model.

6.3.5 Mesh Convergence

To determine the minimum number of elements per member that can be used in the analysis of the dome model, the panel near the apex (panel 1) can be studied (Fig. 6.9(a)). The members comprising this panel are the largest in the dome and may be susceptible to large displacements and local buckling for some load conditions (e.g., a uniform distributed load over the inner half of the dome). To determine the optimal mesh, the panel loads are transformed into member loads by a tributary-area method discussed in Section 6.4. Each member of panel 1 carries part of the load imposed over this panel and over the adjacent panels. The response of the members of panel 1 are studied for different boundary conditions and increasingly refined meshes. For example, beam A of panel 1 is shown in Fig. 6.9(b) loaded with triangular loads that correspond to a uniformly distributed load over the two adjacent panels and a concentrated load transmitted by the two purlins resting on the beam. The response of beam A for 2-element and 4-element meshes is summarized in Table 6.1. Similar results were obtained for beam B and purlin P of Fig. 6.9(a). Based on the response of the members of panel 1, it is concluded that a mesh of 2-elements per member is accurate enough to characterize the response of the dome model.



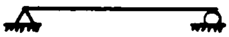
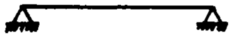
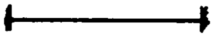
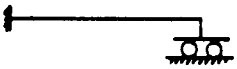
(a) Panel 1

Note: For DL=16 psf and LL=20 psf, $W_1=W_3=7.145$ lb/in and Q=2,836 lbs.

(b) Beam A shown in its radial plane with applied panel loads

Figure 6.9. Mesh convergence study of the beams of panel 1.

Table 6.1 Mesh Convergence Study for Beam A of Panel 1*

	% Difference in Response Between a Two and a Four Element Mesh		
Support conditions	Reactions	Displacements @ quarter points	Stresses @ midspan
	0.0	0.6	0.2
	0.0	0.9	1.0
	0.3	1.3	0.0
	0.3	2.4	3.3

* See Fig. 6.6

6.4 Design Loads

Important design loads for lattice domes are wind and snow loads (Baker, 1984; Eshelby and Evans, 1988; Makowski, 1984; Soare, 1984); the failure of several metallic lattice domes occurred during snow storms (ENR; Holzer et al., 1984; Soare, 1984). The U.S. codes of practice (ANSI, 1982; BOCA, 1984; UBC, 1985) do not provide design wind pressures for domes, and the few international codes that do are not consistent (Baker, 1984). Generally, codes recommend that wind pressure distributions for domes be determined by wind-tunnel tests. However, Baker (1984) states that the problem of obtaining reliable wind load data on any dome is so great, even by wind-tunnel testing, that expert advice is often to simply overdesign on the basis of the best available information. In regard to snow loads, investigators have attributed the collapse of some domes to heavy local snow concentrations (ENR; Holzer et al., 1984; Soare, 1984). This was illustrated by the snap-through buckling of the Bucharest dome (Soare, 1984). The collapse of the dome was attributed to a local snow concentration of approximately 30% of the total design snow load over 15% of the horizontal projection of the dome. This increased the intensity of the pressure by a factor of 3.5 over the uniform design pressure. A historical summary of progressive collapse of light framed structures during snow storms is given by Lorenzen (1980). Wreckage data collected over a period of 12 years (1966-1978) indicates that the failure of a minor part of a roof generally precipitates the complete collapse of the roof frame. Unfortunately, current codes of practice do not seem to reflect the possibility of local snow accumulations on domes during snow storms.

The design loads for the test dome considered in this study are estimated following the guidelines given by the American Institute of Timber Construction (1985), the American National Standards Institute (1982), the Uniform Building Code (1985), and the European Convention for Constructional Steelwork (1978). The loads considered include dead load, live load, snow loads, wind loads, and earthquake loads.

6.4.1 Dead Load

The dead load for timber domes consists of the weight of the glulam beams and purlins, the steel connector hubs, the tongue-and-groove decking, the insulation and roofing material, and other mechanical and electrical components permanently attached to the roof. The dead load for the dome model is estimated as follows:

Beams and purlins	=	2 psf (based on 40 pcf specific weight)
2" decking	=	5 psf (AITC, Table 7.1, p. 7-664)
Steel hubs, roofing and insulation	=	9 psf
Total	=	16 psf

6.4.2 Live Load

Live load consists of the weight imposed over the structure during construction, roofing, and reroofing. It also accounts for the weight of firemen and equipment in the event of fire. Live load for domes can be computed from ANSI A58.2-82, Sect. 4.10, p. 12. The same information is presented in tabular form in UBC-85 (Table 23-C, p. 128). The live load computed for the dome model is 20 psf.

6.4.3 Snow Load

Guidelines for design snow loads are given in AITC (1985) and ANSI A58.1-82. Using these sources, the snow load for the dome model is estimated as 14 psf uniformly distributed over the entire dome and a nonuniform load over half the dome that varies from 7 psf at the crown to 28 psf at the eve.

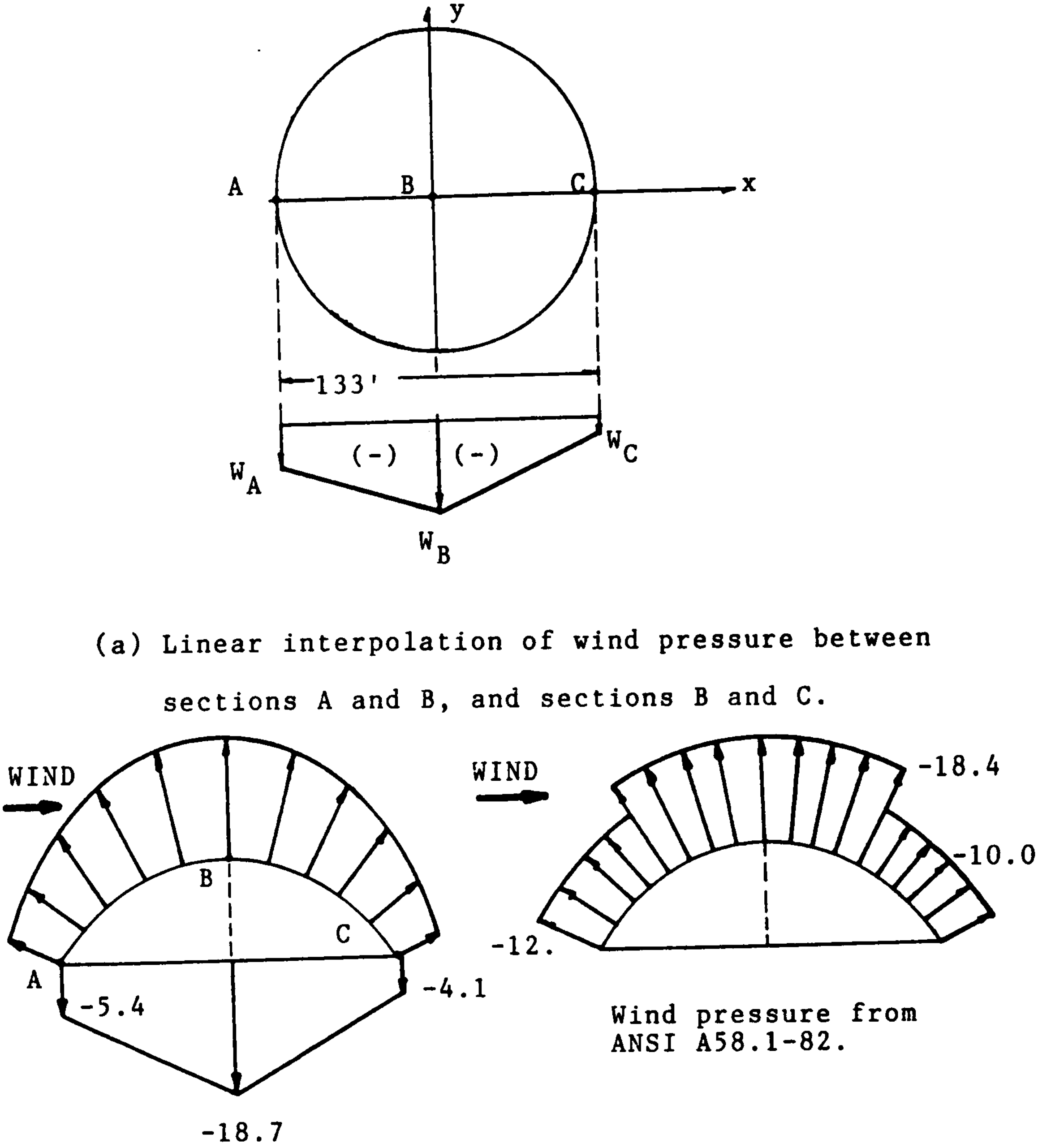
6.4.3 Wind Load

The information contained in the *Wind Loading Handbook* (Newberry and Eaton, 1974) and the ECCS (1978) appears to be the best available. Since the ECCS pressure distributions are explicitly defined and simple to apply, they would be the preferred wind load pressures in design provided they are sufficiently accurate. The ECCS specifies pressure values at three points along the midsection of the dome in the wind direction, the points A, B, C in Fig. 6.10(a). The pressure along any section normal to the wind is constant, and values between the points A, B, C are obtained by linear interpolation.

The wind load for the dome model is determined from ECCS by using the basic wind speed given in ANSI A58.1-82, p. 36. The pressure distribution over the dome is shown in Fig. 6.10(b) and corresponds to a height of the tension ring of 15 ft above the ground. The wind load for the test dome could also be computed from ANSI A58.1-82 using the pressure coefficients for arched roofs. By this approach, it is assumed that pressure distributions are uniform over the windward quarter, center half, and leeward quarter of the dome. However, the jumps or discontinuities from region to region over the roof are unrealistic (Fig. 6.10(b)).

6.4.4 Seismic Load

Seismic loads are specified, for example, in the Uniform Building Code (1985). For the Tacoma dome, the seismic load condition was not critical (Eshelby and Evans, 1988), and therefore, it is not considered in this study.



(a) Linear interpolation of wind pressure between sections A and B, and sections B and C.

Wind pressure from ANSI A58.1-82.

Wind pressure from ECCS.

(b) Wind pressure distribution for the test dome.

Figure 6.10. Wind Load

6.5 Load Combinations

Various load combinations must be considered in the analysis of glulam domes (AITC, 1985). The following load combinations are typical (Varax):

1. Combined DL and LL on entire dome.

2. Combined DL + LL on 1/2 dome terminating on sector line.

3. Combined DL + LL on 1/2 dome terminating on sector center line.

4. Combined DL + LL on inner 1/2 dome.

5. Combined DL + LL on outer 1/2 dome.

6. Combined DL + WL.

7. Combined DL + WL + one-half LL on 1/2 dome (leeward side), LL terminating on sector line.

8. Combined DL + WL + one-half LL on 1/2 dome (leeward side), LL terminating along sector center line.

9. Snow concentrations and special hanging loads.

The test dome will be analyzed for load combinations 1, 2, and 4.

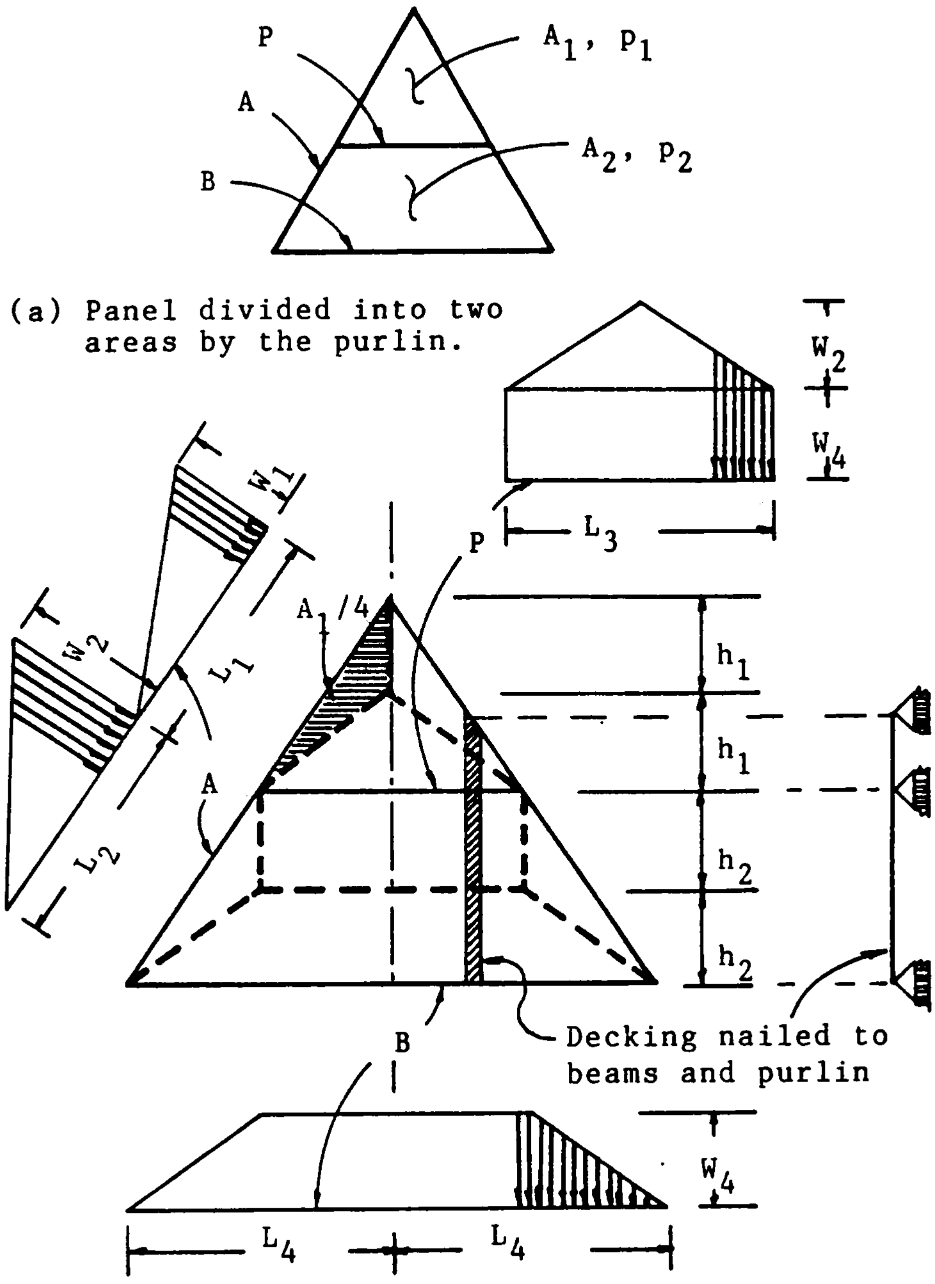
6.6 Discretization of Loads

Each sector of the Crafts Pavilion Triax dome is covered with a tongue-and-groove wood decking. The 2"x6" boards are nailed to the beams and purlins in a parallel direction to the sector center line. In this study, the load-carrying capacity of the decking is not considered. Thus, the distributed loads applied to the decking are transformed into member loads by assigning load-tributary areas to the interconnected members. Furthermore, the distributed member loads are transformed into equivalent concentrated nodal loads. This transformation can be performed internally by the finite element program ABAQUS. However, since the structure consists of hundreds of elements, the task of assembling the load contributions from adjacent panels becomes easier when concentrated nodal loads are specified.

6.6.1 Transformation of Panel Loads into Member Loads

Each sector of the dome contains a number of triangles consisting of three beams and a purlin. Each of these triangles is called a panel. Since the radius of curvature of the beams is relatively large (133 ft), the arc and chord lengths are approximately equal. Thus, for the transformation of loads, each panel can be treated as a "flat triangle". The panel is divided by the purlin into areas A_1 and A_2, and the load acting on each area can be averaged and considered uniformly distributed (Fig. 6.11(a)).

To convert the panel-load into member-loads, it can be assumed that a strip of decking distributes the load equally to the supporting members to which it is nailed. By applying this concept to successive strips, it is possible to divide the panel surface area into member tributary areas as shown in Fig. 6.11(b). The pressure acting over each tributary area can be converted into distributed member loads. The magnitude of these loads is a function of the geometry of the panel and



(a) Panel divided into two areas by the purlin.

Decking nailed to beams and purlin

(b) Load tributary areas.

Figure 6.11. Transformation of panel loading into member actions.

the pressures acting over the corresponding tributary areas. As an example, consider the load acting over the length L_1 of member A in Fig. 6.11(b). The equivalent load over the length L_1 is $W_1 \frac{L_1}{2}$ and is equal to the force $P_1(\frac{A_1}{4})$ due to the pressure P_1 acting over the tributary area $\frac{A_1}{4}$. Thus,

$$\frac{W_1 L_1}{2} = \frac{P_1 A_1}{4}$$

or

$$W_1 = \frac{P_1 A_1}{2L_1}$$

In a similar manner, the loads for all the members of the panel are computed. This load transformation is used for the entire sector of the dome with some modifications in the procedure for the panels near the tension ring.

6.6.2 Transformation of Distributed Element Loads into Concentrated Nodal Loads

The representation of the input data of the applied loads becomes easier if the element distributed loads are converted into concentrated nodal loads. The loads are transformed using the load-discretization finite element approach which is consistent with the formulation of the isobeam element (Ch. 4). To illustrate the procedure, consider the isobeam element in the plane of Fig. 6.12: the coordinates, displacements, and rotations are interpolated in terms of the corresponding nodal values as

$$x = \sum_{i=1}^{3} h_i x_i$$
$$v = \sum_{i=1}^{3} h_i v_i \qquad (6.1)$$
$$\theta = \sum_{i=1}^{3} h_i \theta_i$$

where h_i are the Lagrange interpolation polynomials introduced in Chapter 4 (Eq. 4.1):

$h_1 = -\frac{r}{2}(1-r), \quad h_2 = \frac{r}{2}(1+r), \quad h_3 = (1-r^2)$

The displacements v and rotations θ are interpolated independently (Eq. 6.1), but they are related by the shear strain γ_{xy} (see Fig. 5.1(c)). Similarly, the distributed element forces and moments are uncoupled and can be converted into discrete nodal values by integrating the corresponding load function over the element. Therefore, the equivalent nodal forces and moments are computed from

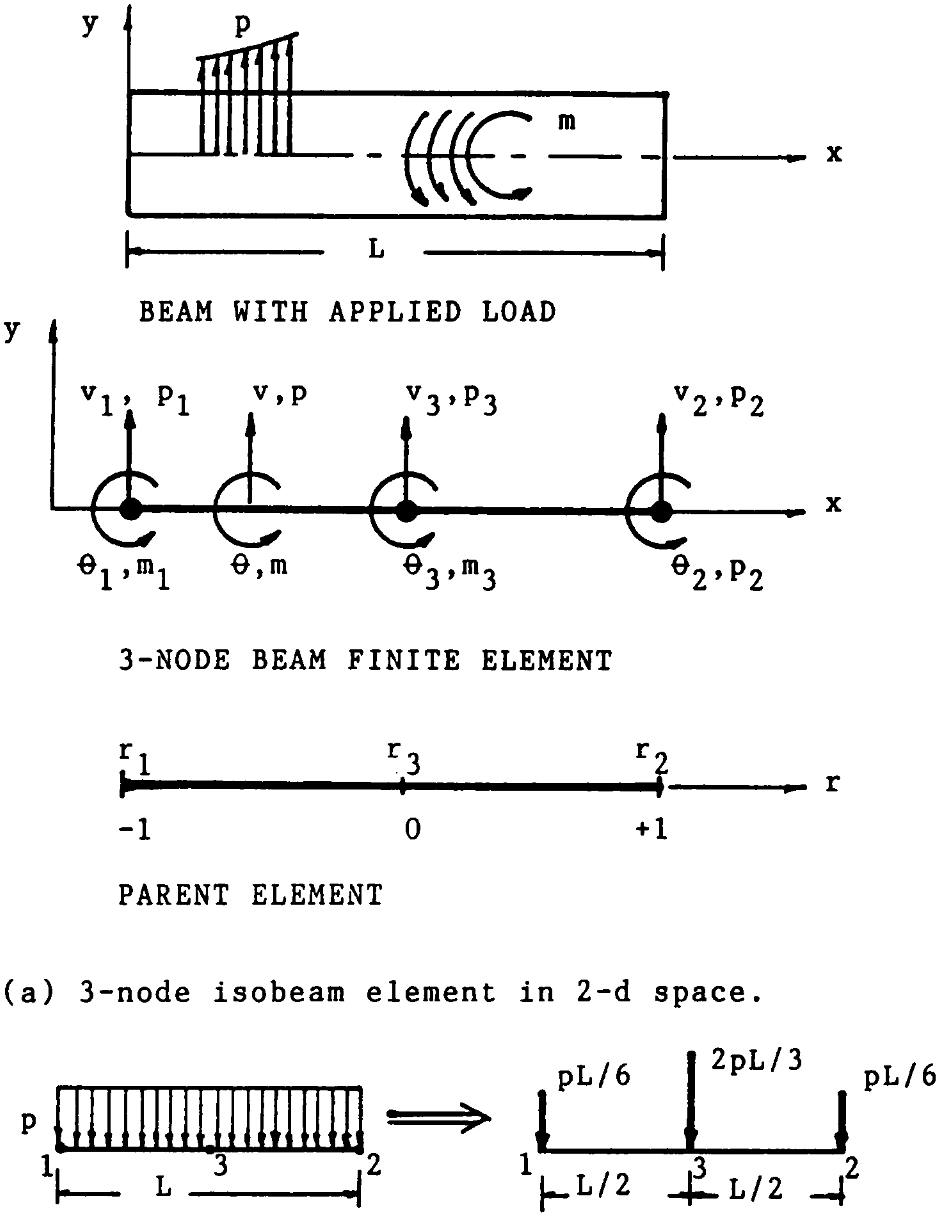
$$f_{pi} = \int_{x_1}^{x_2} h_i \; p(x) \; dx \qquad (6.2)$$

$$f_{mi} = \int_{x_1}^{x_2} h_i \; m(x) \; dx \qquad (6.3)$$

For a straight element, dx can be expressed as

$$dx = \sum_{i=1}^{3} h_{i,r} \; x_i \; dr = \frac{L}{2} dr \qquad (6.4)$$

For example, a uniform distributed load over the element can be converted into nodal loads through Eqs. (6.2) and (6.4). In this case, the load function is a constant p and the integral is simply



(a) 3-node isobeam element in 2-d space.

(b) Load discretization

Figure 6.12. Transformation of element loads into nodal loads.

$$f_{pi} = p\frac{L}{2} \int_{-1}^{1} h_i dr \qquad i = 1,2,3$$

The equivalent nodal forces are

$$f_{p1} = \frac{pL}{2}\frac{1}{2}[\frac{r^3}{3} - \frac{r^2}{2}]_{-1}^{1} = \frac{1}{6}pL$$

$$f_{p2} = \frac{pL}{2}\frac{1}{2}[\frac{r^3}{3} + \frac{r^2}{2}]_{-1}^{1} = \frac{1}{6}pL$$

$$f_{p3} = \frac{pL}{2}\frac{1}{2}[r - \frac{r^3}{3}]_{-1}^{1} = \frac{2}{3}pL$$

as shown in Fig. 6.12(b).

Once the finite element mesh is established, the values of the nodal forces for each panel can be tabulated as functions of the applied panel pressures. To compute the total load at a node, the load contributions from adjacent panels are added. For example, the nodal loads for the panel near the apex (panel 1) of the test dome are given in Table 6.2 for a mesh of two elements per member. The definition of the nodes and panel pressures are given in a sketch below Table 6.2.

6.7 Finite Element Analysis

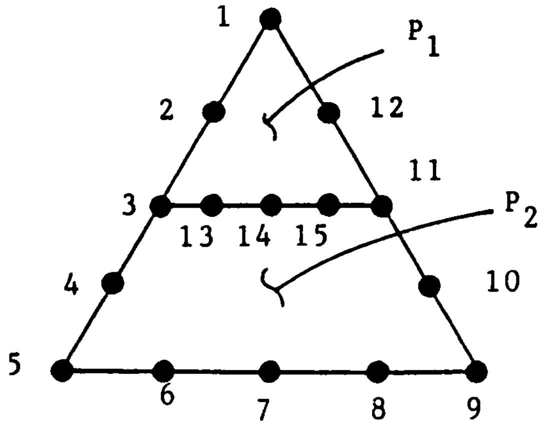
The finite element analysis of lattice domes involves the selection of finite elements (see Section 6.3.2), the formulation of the equations of equilibrium (see Chapter 2), the application of a solution process (Section 6.7.1), and the evaluation of the structural response (Section 6.7.2.).

Table 6.2 Equivalent nodal loads for panel 1 shown below.

Nodes*	Nodal Loads (lbs)
1	λP_1
2, 12	λP_1
3, 11	λP_2
4, 10	λP_2
5, 9	$0.1254 \lambda P_2$
6, 8	$3.25 \lambda P_2$
7	$2.25 \lambda P_2$
13, 15	$\lambda P_1 + 2.0 \lambda P_2$
14	$\lambda P_1 + \lambda P_2$

Note: $\lambda = 1{,}260.4$; P_1, P_2 = panel pressures in psi

*Node numbering for a mesh of 2 elements per member



6.7.1 Analysis Procedure

The dome model is analyzed for three load conditions as described in Sections 6.5 and 6.7.2. For each load condition, three distinct analyses are performed: a linear analysis to check the design adequacy of the structure, a combined linear/nonlinear buckling load prediction analysis, and an incremental iterative nonlinear analysis.

Linear Analysis

In the linear analysis, the member design criteria is checked. The program output gives the extreme-fiber stresses of the cross section at the location of the integration points along the beam axis of each beam element. From these, the bending and axial components of stresses are computed by a postprocessing program (Huang, 1989). The induced member stresses are then checked against the design criteria established by the National Design Specifications for Wood Construction (1986). Specifically, the interaction equations for combined flexural and axial loading of NDS Section 3.10 are used. The allowable stresses for the southern pine beams of the test dome were obtained from NDS (1986) for E-rated 56 southern pine glulam.

Nonlinear Analysis

At equilibrium points obtained in the solution process, the lattice dome model is tested for local and global failure modes. This includes the computation of strains and stresses, the identification of excessive deformations, and the investigation of the stability of equilibrium to search for critical points. In the stability analysis of the test dome, two different methods are used: a linearized eigenvalue buckling prediction and an incremental iterative nonlinear analysis by the modified Riks-Wempner method. The details of these methods are presented next.

Eigenvalue buckling prediction. ABAQUS permits a combined linear and nonlinear analysis for estimating elastic buckling by eigenvalue extraction. The procedure consists of computing two tangent stiffness matrices at distinct load levels (e.g., zero load and a small increment of the design load). The change in stiffness is assumed to be proportional to the change in load. This leads to an eigenvalue problem, and the lowest eigenvalue provides an estimate of the buckling load. This estimate is useful for "stiff" structures, where the prebuckling response is almost linear (ABAQUS). Moreover, it helps to predict the length of the nonlinear equilibrium path that must be traced to locate the first critical point in a nonlinear analysis. According to Holzer and Huang (1989), "the combined analysis may have originated with Dupuis et al. (1970) who used it to compute a bifurcation point (Kardestuncer et al., 1987, p. 2.256), and it is one of the linear buckling formulations considered by Brendel, Kempen, and Ramm (1982)." The procedure is as follows (Holzer and Huang, 1989):

1. A base load ${}^bR$ is applied and a nonlinear analysis is performed to obtain the corresponding tangent stiffness matrix ${}^bK$.

2. A small load increment ΔR is added to define a reference load

$$ {}^rR = {}^bR + \Delta R \tag{6.5} $$

and the corresponding tangent stiffness matrix ${}^rK$ is computed.

3. It is assumed that the change in stiffness, ΔK, going from the base load ${}^bR$ to the reference load ${}^rR$, is proportional to the change in load, ΔR . Thus,

$$ \Delta K = {}^rK - {}^bK \tag{6.6} $$

4. Based on this assumption, for any load

$$ R = {}^bR + \lambda \Delta R \tag{6.7} $$

the stiffness of the structure is defined by

$$K = {}^{b}K + \lambda \Delta K \tag{6.8}$$

5. At a critical load $K(\lambda)$ is singular, and the problem of solving for λ, such that $\det(K) = 0$, leads to the solution of the eigenvalue problem

$$({}^{b}K + \lambda \Delta K)\phi = 0 \tag{6.9}$$

which yields the critical load parameter, λ_{cr}, the corresponding buckling mode shape, ϕ_{cr}, and the buckling load

$$R_{cr} = {}^{b}R + \lambda_{cr} \Delta R \tag{6.10}$$

For proportional loading, ΔR, is a multiple of ${}^{b}R$ and Eq. 6.7 can be written as

$$R = {}^{b}R + \lambda(c\ {}^{b}R), \quad c < 1$$

or

$$R = \lambda_b\ {}^{b}R, \quad 0 \leq \lambda \leq \lambda_{cr} \tag{6.11}$$

where, according to Holzer and Huang (1989), "...the base load ${}^{b}R$ is magnified as λ_b varies from 1 to λ_{cr}. If linear buckling analyses are performed for increasing values of λ_b , a sequence of critical load parameter predictions are obtained that converge to λ_{cr} as λ_b approaches λ_{cr}. This is illustrated in Fig. 6.13(a), where the critical load prediction curve intersects the non-linear equilibrium path at the actual critical point. The critical load prediction curve in Fig. 6.13(a) is not unique because it depends on the degree of freedom selected for its presentation. This problem can be eliminated by selecting the base load parameter λ_b rather than a degree of freedom u_i to plot the critical load prediction curve, as shown in Fig. 6.13(b). The resulting critical load prediction curve intersects the 45° line ($\lambda = \lambda_b$) at the critical load parameter λ_{cr}."

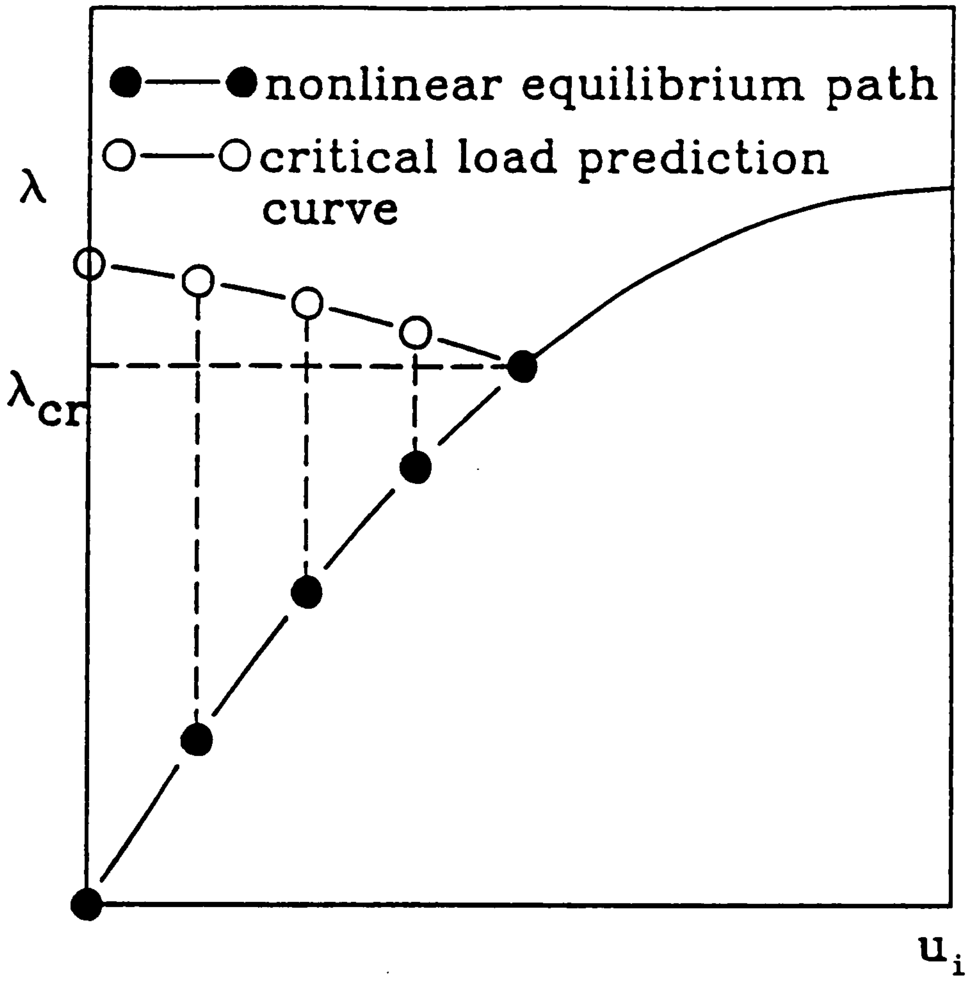


Figure 6.13a. Load-displacement curves (Holzer and Huang, 1989).

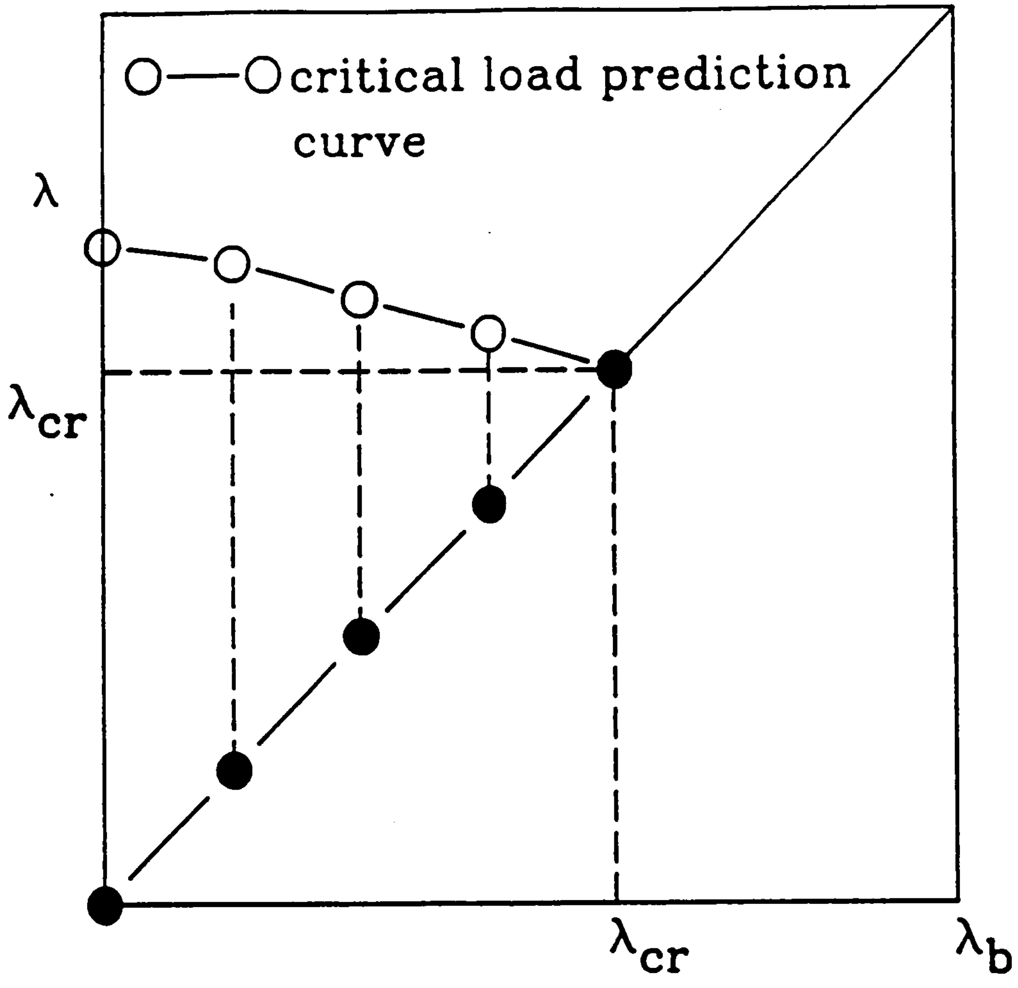


Figure 6.13b. λ - λ_b curves (Holzer and Huang, 1989).

The combined analysis is illustrated for the truss problem of Fig. 6.14(a). The equilibrium equation can be written as (ABAQUS, 4.5, test problem 3.2.3)

$$P = \frac{AEL^2}{l^2}(u-h)\ln(\frac{l}{L}) \tag{6.12}$$

The stiffness of the system at any time is

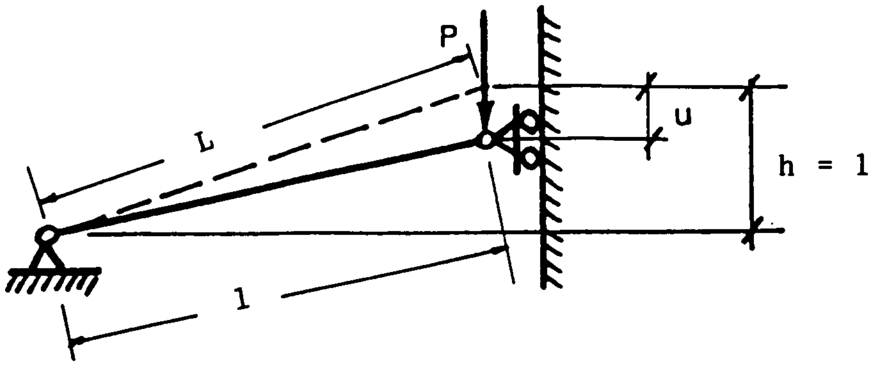
$$K = \frac{dP}{du} = \frac{AEL}{l^2}[\ln\frac{l}{L} + \frac{(u-h)^2}{l^2}(1-2\ln\frac{l}{L})] \tag{6.13}$$

For the dimensions chosen, the first limit point corresponds to

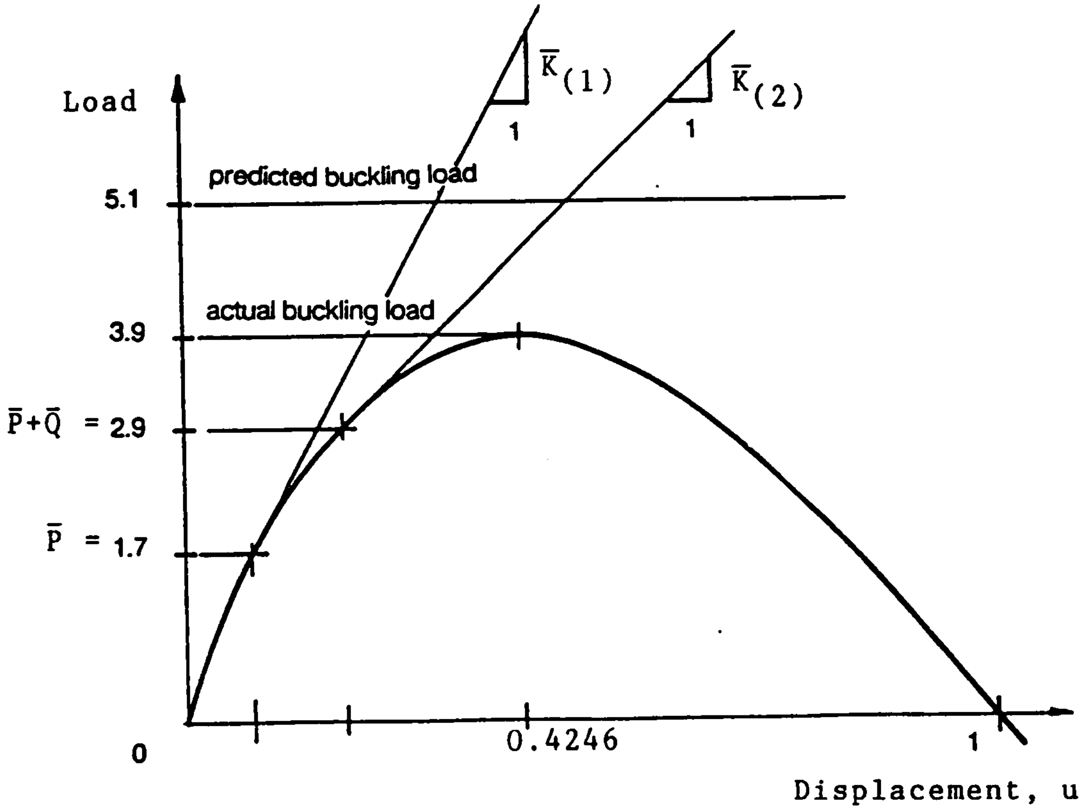
$$u = 0.4246, \quad P_{cr} = 3.89x10^{-3}\frac{AE}{2L} \tag{6.14}$$

as shown in Fig. 6.14(a). Calculations of linearized buckling estimates are given in Table 6.3 for three load increments that correspond to displacements u = 0.1, 0.2 and 0.3 (Eq. (6.12)). The graphical representation of the second critical load estimate of Table 6.3 is shown in Fig. 6.14(a). The critical load prediction curve is shown in Fig. 6.14(b), from which the critical load can be approximated by intersecting the 45° line with a straight line joining two buckling prediction points. For the choice of points shown in Fig. 6.14(b), the predicted buckling load is within 5% of the actual buckling load. The buckling loads of the dome model can be predicted by the linear function interpolation/extrapolation procedure illustrated in this example.

Incremental iterative nonlinear analysis. In the nonlinear analysis, the nonlinear equations of equilibrium can be solved by various incremental iterative procedures that result in a series of points on the equilibrium path; each equilibrium point defines the values of the nodal displacements and the corresponding load parameter. In several comparison studies (Holloway and Rustum, 1984; Holzer et al., 1981), the modified Riks-Wempner method, the choice of this study, was found to be most effective in tracing nonlinear equilibrium paths through bifurcation and limit (turning) points along the curve.



Truss problem (ABAQUS 4.5, test problem 3.2.3)



Graphical representation of the second critical load estimate given in Table 6.3.

Figure 6.14a. Example: Buckling load prediction.

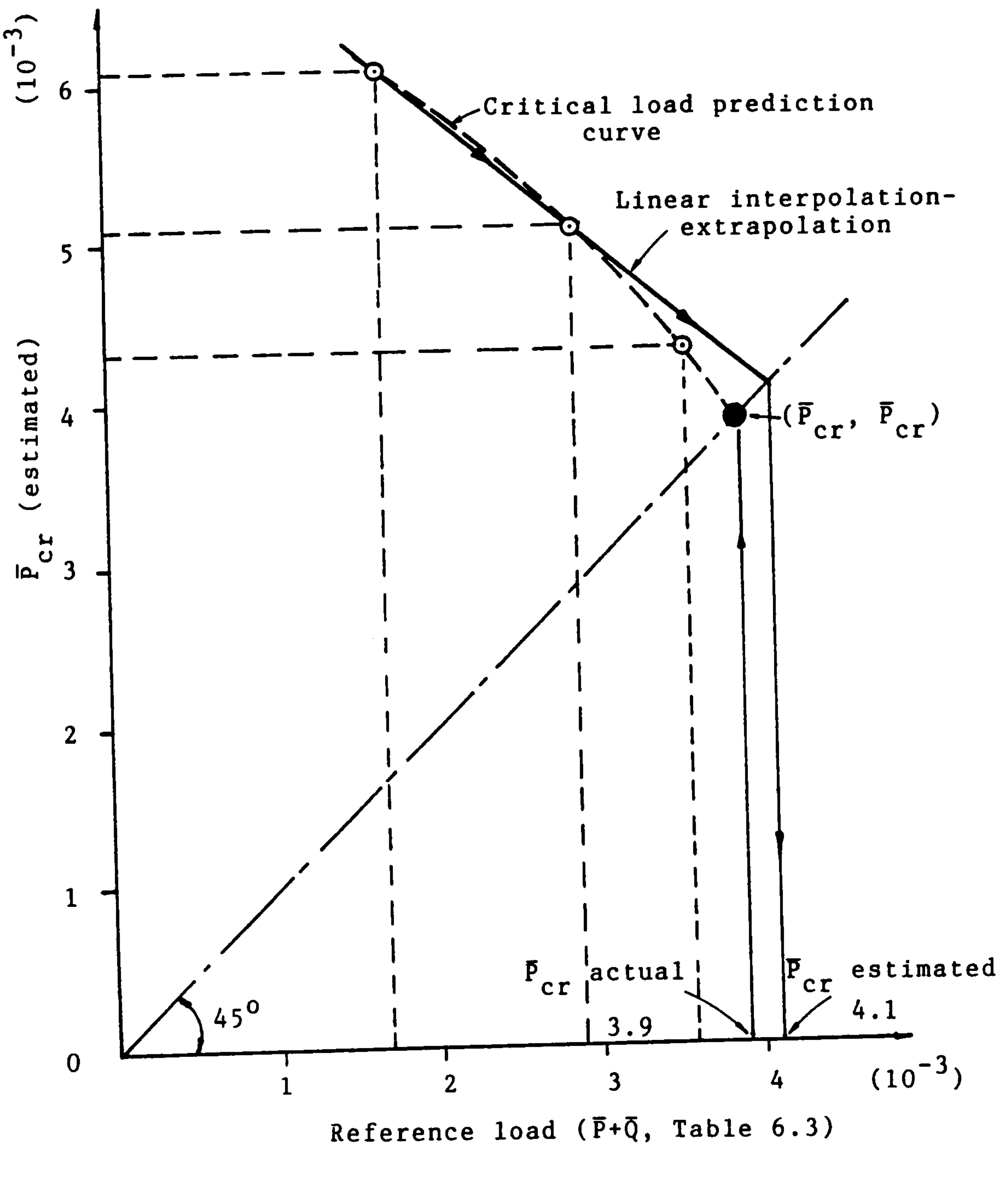


Figure 6.14b. Critical load estimate.

Table 6.3 Eigenvalues buckling predictions for the truss problem of Figure 6.14(a).

	(1)	(2)*					
Prediction	$\underline{P}$	$(\overline{P}+\overline{Q})$	$\overline{K}_{(1)}$	$\overline{K}_{(2)}$	$\Delta\overline{K}=\overline{K}_{(2)}-\overline{K}_{(1)}$	$\overline{P}_{cr}^{**}=\overline{P}+\lambda_{cr}\overline{Q}$	$\lambda_{cr}^{**}=-\frac{K_{(1)}}{\Delta K}$
1	0.0	$1.715\text{x}10^{-3}$	$2.0\text{x}10^{-2}$	$1.439\text{x}10^{-2}$	$-0.561\text{x}10^{-2}$	$6.114\text{x}10^{-3}$	3.565
2	$1.715\text{x}10^{-3}$	$2.896\text{x}10^{-3}$	$1.439\text{x}10^{-2}$	$9.319\text{x}10^{-3}$	$-5.068\text{x}10^{-3}$	$5.068\text{x}10^{-3}$	2.839
3	$2.896\text{x}10^{-3}$	$3.596\text{x}10^{-3}$	$9.319\text{x}10^{-3}$	$4.812\text{x}10^{-3}$	$-4.507\text{x}10^{-3}$	$4.347\text{x}10^{-3}$	2.068

NOTE: The bar indicates a quantity divided by $\frac{AE}{2L}$.

* The loads correspond to displacements u = 0.1, 0.2, and 0.3.

** estimated

The stability of the structure can be determined from the eigenvalues of the tangent stiffness matrix, which is formulated at each equilibrium point in the solution process. Equilibrium is stable if all eigenvalues are positive, critical if one or more eigenvalues are zero, and unstable if one or more eigenvalues are negative (Holzer et al., 1981). For computational efficiency, stability is usually determined from the diagonal matrix obtained in the factorization of the tangent stiffness matrix during Gaussian elimination, a necessary step in the solution process.

It is an important property of the tangent stiffness matrix that the effects of all deformations modeled by the finite elements and linked in the assembly are reflected by its stability characteristics. For example, the appearance of a single negative eigenvalue signifies the loss of stiffness, i.e., the possibility of a rigid body displacement in one direction. This may correspond, for example, to the snap-through of a single joint or the torsional-flexural buckling of some elements. Thus, any form of instability is reflected by the tangent stiffness matrix.

At each equilibrium point obtained in the solution process, the state of the glulam dome model is assessed. This includes the investigation of the stability of equilibrium at the given load level; the computation and evaluation of normal and shearing stresses, and the identification of excessive deformations and possible material failures (crushing and rupture). A very important question that the response evaluation of the test dome must answer is whether the mode of failure is governed by elastic instability or by material strength. Therefore, the induced stresses are checked against the proportional-limit and ultimate stresses recommended by the Wood Handbook (1987) for longleaf pine (Table 6.4).

6.7.2 Response Evaluation of the Dome Model

Based on the modeling assumptions and analysis considerations discussed in the preceding sections, the dome model is analyzed for three load conditions:

Table 6.4 Maximum parallel to grain stresses (psi) for longleaf pine (Wood Handbook, 1987, Tables 4-2 and 4-8).

Compression		Tension	Shear
proportional limit	ultimate compressive stress*	ultimate tensile stress**	ultimate shearing stress
5,896	8,470	14,500	1,510

* Maximum crushing stress

** Modulus of rupture from static bending tests.

1. Dead load plus uniform snow load over the entire dome.

2. Dead load plus snow load on the inner circular half of the horizontal projection of the dome.

3. Dead load plus snow load over the lower half of the horizontal projection of the dome, terminating on sector lines.

For load condition 1, the whole dome and a sector of the dome are analyzed. The analyses for the three load conditions are presented first, and then a discussion of the results is presented.

Load Condition 1

The first load condition consists of dead load (16 psf) and uniform snow load over the entire dome (20 psf). Since the geometry and the load are symmetric, a sector of the dome is analyzed first to test the analyses procedures (Section 6.7.1) and the required boundary conditions to model symmetry (Section 6.3.3). The geometry of the sector is shown in Fig. 6.6, and the boundary conditions are explained in Section 6.8.3 and shown in Fig. 6.8(b). The width of the beams along the sector lines are cut in half. The analysis of the sector is used as a guide for the analysis of the whole dome. The geometry of the dome is presented in Fig. 6.5, and the boundary conditions are shown in Fig. 6.8(a). Three-node isoparametric beam elements are used to model the beams and purlins, with a mesh of 2-elements per member (Section 6.3.5). The connectors are modeled with 2-node beam elements, and the tension ring is modeled with truss elements (Section 6.3.2). The joints are first assumed rigid (including the purlin connectors). Then, flexible joints are specified by reducing the flexural and torsional stiffnesses of the connectors. With flexible joints, the purlin connectors are modeled as pinned by reducing the principal moments of inertia (I_1 and I_2) and the torsional constant (J) of the connectors by a factor of 256. By reducing the stiffness by a facotr of 256, the connector can simulate a pinned connection, and yet, the reduced stiffness is sufficient to avoid causing singularity of the stiffness matrix. The results of the analyses of the sector and the whole dome are summarized in Tables 6.5 and 6.6.

Table 6.5. Load Condition 1: DL plus uniform SL over the entire dome. Results of the anlayses of a sector.

	Nonlinear Analysis					
	Linear Analysis		Maximum Stress[3]			
Joint Stiffness	Max. Stress[2] % of Allowable . Design Stress	Eigenvalue Buckling Prediction (λ)	Buckling Load (λ)	% of Proportional Limit	% of Ultimate Compressive Stress	% of Ultimate Tensile Stress
Rigid	70	3.611	4.188* (6.738)**	128	89	34
50% (84%)[1]	75	---	3.035 (4.663)	133	92	57
25% (70%)[1]	75	---	2.848 (4.326)	137	95	59
5% (50%)[1]	78	---	1.935 (2.683)	156	108	51

[1] % of original width and height

[2] National Design Specification (1986)

[3] Wood Handbook (1987)

* $Q = \lambda(Q_D + Q_L)$

** $Q = Q_D + \lambda Q_L$

Note: For flexible joints the purlins are pinned (I_1, I_2 and J are reduced by a factor of 256).

Table 6.6. Load Condition 1: DL plus uniform SL over the entire dome. Results of the anlayses of the whole dome.

		Nonlinear Analysis				
	Linear Analysis			Maximum Stress[3]		
Joint Stiffness	Max. Stress[2] % of Allowable Design Stress	Eigenvalue Buckling Prediction (λ)	Buckling Load (λ)	% of Proportional Limit	% of Ultimate Compressive Stress	% of Ultimate Tensile Stress
Rigid	70	2.8207	3.279* (5.0986)**	87	60	22
75% (93%)[1]	72	---	2.618 (3.912)	90	62	18
50% (84%)[1]	75	---	2.49 (3.682)	61	42	10
25% (70%)[1]	75	---	2.249 (3.2482)	67	46	12

[1] % of original width and height

[2] National Design Specification (1986)

[3] Wood Handbook (1987)

* $Q = \lambda(Q_D + Q_L)$

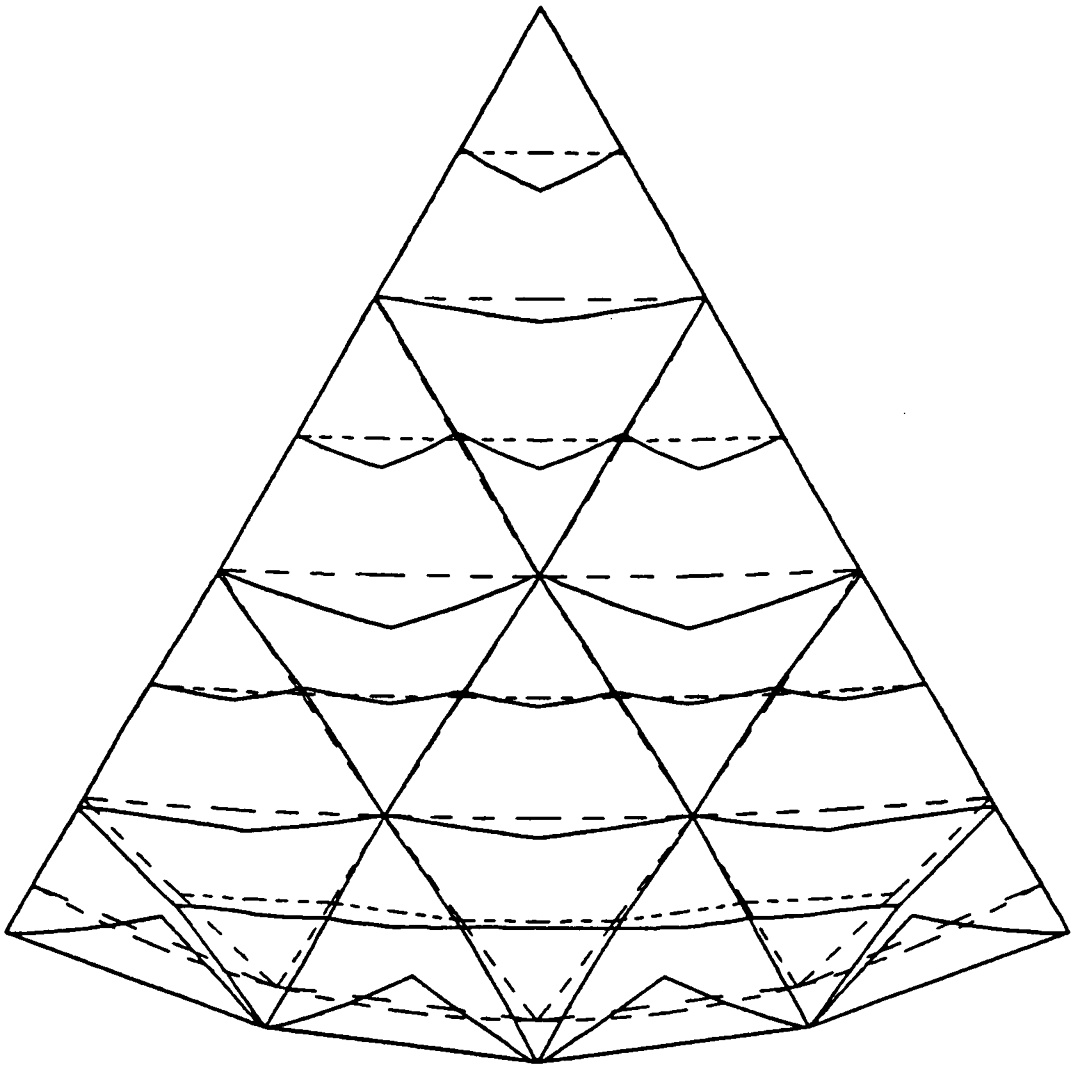
** $Q = Q_D + \lambda Q_L$

Note: For flexible joints the purlins are pinned (I_1, I_2 and J are reduced by a factor of 256).

In the linear analysis, the results showed that the response of the sector corresponds precisely to the response of the whole dome (i.e., the displacements and stresses for both structures are the same). The induced stresses in the members are below the allowable stresses specified by the code (NDS, 1986). In the buckling analysis, the response of the sector does not reflect the response of the dome. The reason is that the boundary conditions to model symmetry constrain the sector, along the sector lines, from displacing laterally, and only vertical displacements are possible along the sector lines. Thus, the sector is "somewhat stiffer" than the whole dome. Nevertheless, the nonlinear analysis of the sector is used as a guide for the analysis of the dome. The eigenvalue buckling prediction analysis for rigid joints is performed by computing the tangent stiffness matrices at zero load and at the design load. The linearized buckling analysis underestimates the buckling load for the sector and the whole dome by only 13%. The buckling loads for the sector are 25% higher than those for the dome. As the joint stiffness decreases, the buckling load decreases. For example, by decreasing the joint stiffness by 50% (with respect to a rigid joint), the buckling load decreases by 32% (see Table 6.6). The deformed shapes of the sector and the dome for a 50% joint stiffness are shown in Figures 6.15 and 6.16, respectively. The buckling analysis of the sector overestimates the induced element stresses by more than 100%. Therefore, they are not reliable. The results of the buckling analysis of the dome show that stresses in all the members remain below the proportional limit, and therefore, below the ultimate allowable values specified by the Wood Handbook (1987). Thus, for load condition 1, the mode of failure of the dome is elastic instability. A detailed discussion of results is presented at the end of this chapter.

Load Condition 2

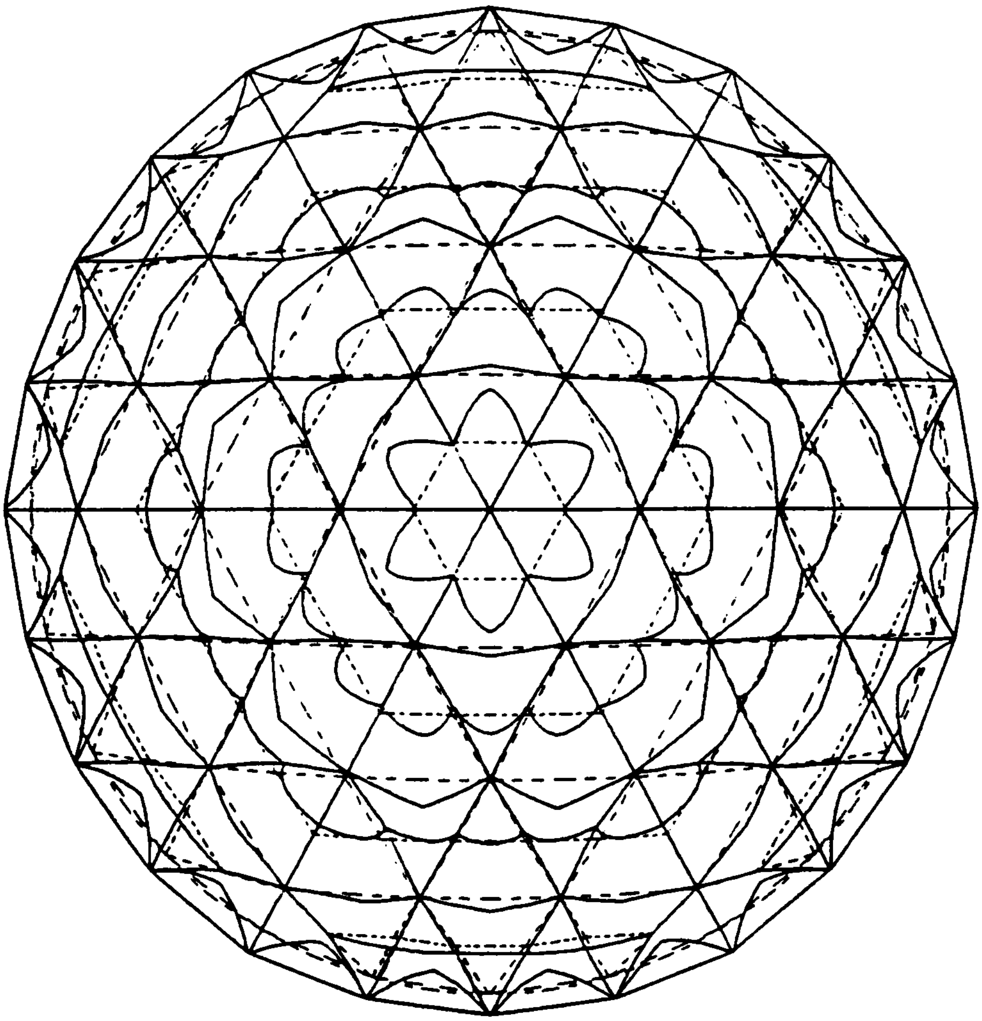
This load condition consists of dead laod (16 psf) and uniform snow load (20 psf) on the horizontal projection of the circular inner half of the dome. The dome model is analyzed for rigid joints and for a 50% reduction in the joint stiffnesses of the connectors. The results of the analyses are summarized in Table 6.7. In the linear analysis, the stresse are about 50% below the allowable design values computed from NDS (1986). In the linearized buckling prediction analysis, for the



PLAN VIEW

solid lines - displaced mesh
dashed lines - original mesh
magnification factor - 490.

Figure 6.15. Load condition 1: buckling analysis of a sector for 50% joint stiffnesses and pinned purlins.



PLAN VIEW

solid lines - displaced mesh
dashed lines - original mesh
magnification factor - 450.

Figure 6.16. **Load condition 1: buckling analysis of the whole dome for 50% joint stiffnesses and pinned purlins.**

condition of rigid joints, two tangent stiffness matrices are computed: one at zero load and one at the design load. The eigenvalue solution, as described in Section 6.7.1, provides an estimate of the buckling load, which is only 7% below the buckling load given by the incremental nonlinear analysis (Table 6.7). When the joint stiffnesses are reduced by 50%, the reduction in buckling load, from the condition of rigid joints, is approximately 6%. Thus, it appears that for this load condition, a reduction of 50% in the joint stiffness does not cause a significant reduction of the buckling load. In the buckling analysis, the maximum stresses are below the proportional limit, and the mode of failure of the dome model is elastic instability. The deformed shapes for rigid and 50% joint stiffnesses are shown in Figures 6.17 and 6.18.

Load Condition 3

This load condition consists of dead load (16 psf) and a uniform snow load (20 psf) on half the dome. The load terminates on sector lines. The results of the anlayses for rigid and 50% joint stiffnesses are presented in Table 6.8. Under the design load (linear analysis), the stresses are about 50% below the allowable design values (NDS, 1986). The eigenvalue buckling prediction for rigid joints is accomplished by computing the tangent stiffness matrices at zero and at the design load. The buckling load estimate is 2% below the buckling load given by the incremental nonlinear analysis. A 50% reduction in the stiffness of the joints results in a 23% reduction of the buckling load. Thus, for this load condition, the flexibility of the joints has a significant effect on the buckling response of the structure. At the buckling load, the maximum stresses remain below the proportional limit, and once again, the failure mode of the structure is elastic buckling. The deformed configurations are shown in Figs. 6.19 and 6.20.

Table 6.7. Load Condition 2: Results of DL plus SL on the inner-half of the dome.

	Linear Analysis	Nonlinear Analysis				
				Maximum Stress[3]		
Joint Stiffness	Max. Stress[2] % of Allowable Design Stress	Eigenvalue Buckling Prediction (λ)	Buckling Load (λ)	% of Proportional Limit	% of Ultimate Compressive Stress	% of Ultimate Tensile Stress
Rigid	45	3.043* (4.677)**	3.267* (5.081)**	83	58	26
50% (84%)[1]	53	---	3.073 (4.731)	96	67	23

[1] % of original width and height

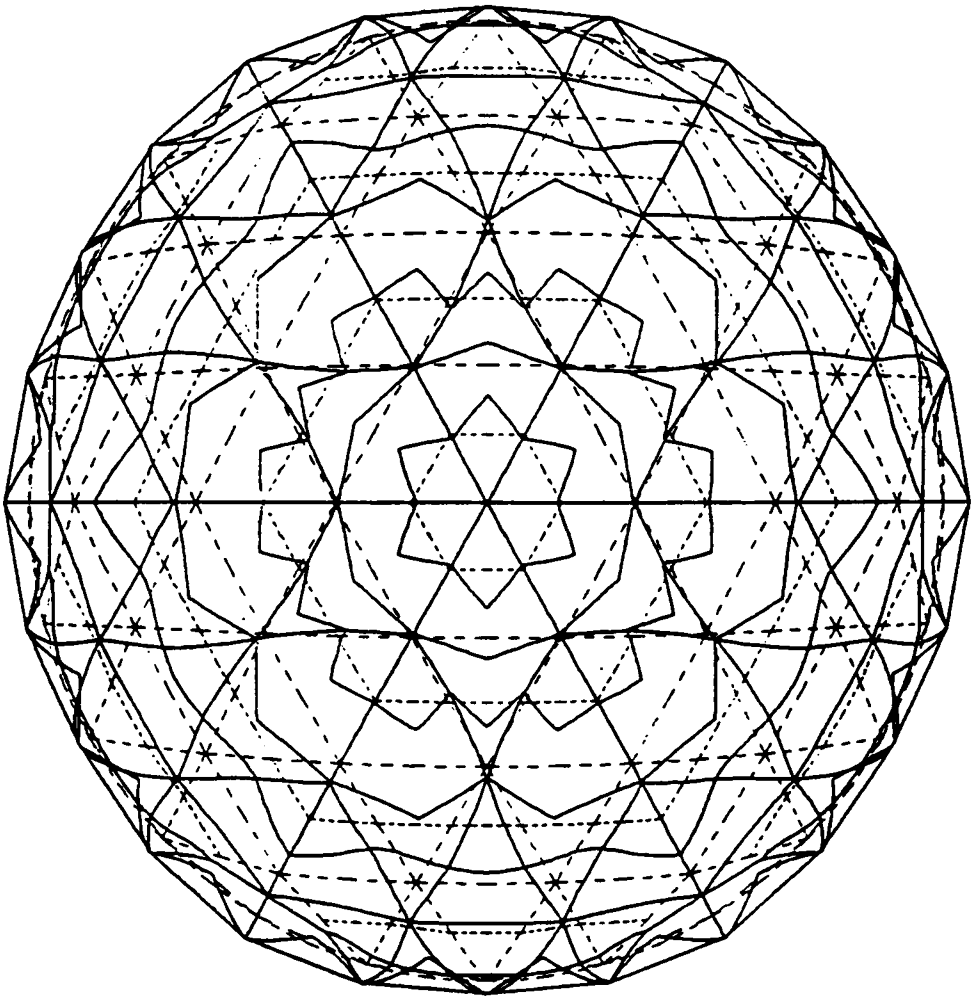
[2] National Design Specification (1986)

[3] Wood Handbook (1987)

* $Q = \lambda(Q_D + Q_L)$

** $Q = Q_D + \lambda Q_L$

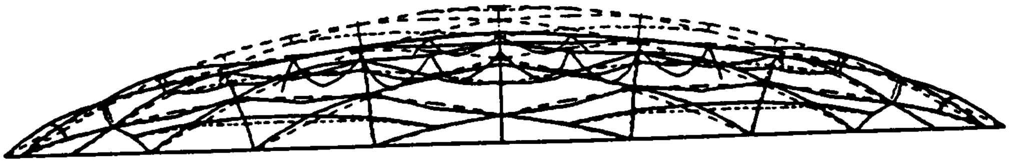
Note: For flexible joints the purlins are pinned (I_1, I_2 and J are reduced by a factor of 256).



PLAN VIEW

solid lines - displaced mesh
dashed lines - original mesh
magnification factor - 860.

Figure 6.17. Load Condition 2: buckling anlaysis for rigid joints (including purlins).



ELEVATION

solid lines - displaced mesh
dashed lines - original mesh
magnification factor - 180.

Figure 6.18. Load Condition 2: buckling analysis for 50% joint stiffnesses and pinned purlins.

Table 6.8. Load Condition 3: Results of DL plus SL on half the dome.

	Linear Analysis	Nonlinear Analysis				
				Maximum Stress[3]		
Joint Stiffness	Max. Stress[2] % of Allowable Design Stress	Eigenvalue Buckling Prediction (λ)	Buckling Load (λ)	% of Proportional Limit	% of Ultimate Compressive Stress	% of Ultimate Tensile Stress
Rigid	50	2.452* (3.614)**	2.492* (3.686)**	82	58	24
50% (84%)[1]	55	---	2.022 (2.840)	51	36	46

[1] % of original width and height

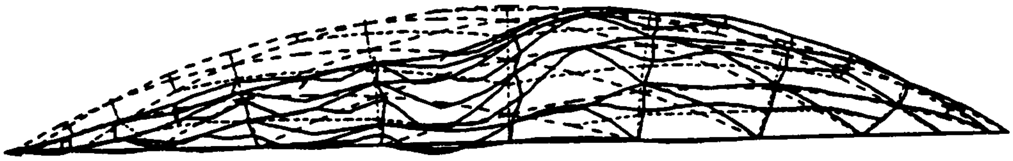
[2] National Design Specification (1986)

[3] Wood Handbook (1987)

* $Q = \lambda(Q_D + Q_L)$

** $Q = Q_D + \lambda Q_L$

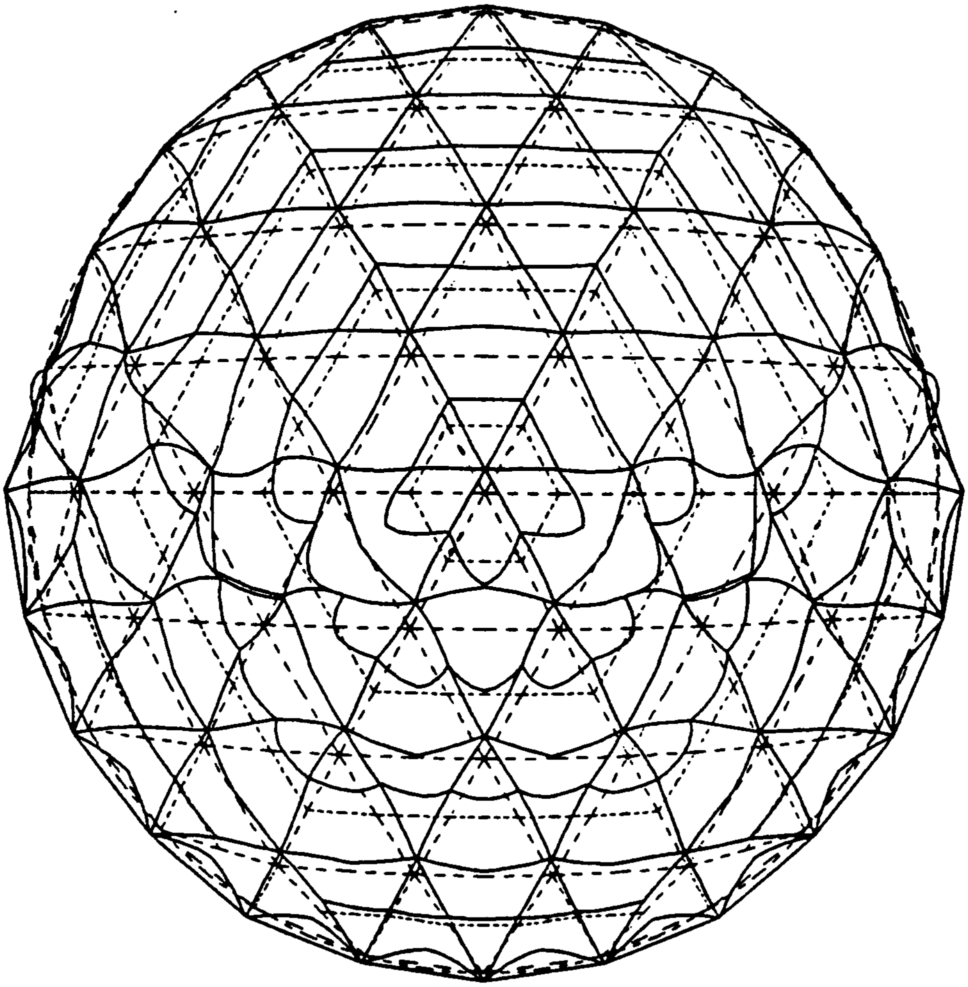
Note: For flexible joints the purlins are pinned (I_1, I_2 and J are reduced by a factor of 256).



ELEVATION

solid lines - displaced mesh
dashed lines - original mesh
magnification factor - 270.

Figure 6.19. Load condition 3: buckling analysis for rigid joints (including purlins).



PLAN VIEW

solid lines - displaced mesh
dashed lines - original mesh
magnification factor - 430.

Figure 6.20. Load condition 3: buckling analysis for 50% joint stiffnesses and pinned purlins

6.8 Discussion of Results

From the results of the analyses of the dome model, the following observations can be made:

1. For load conditions 1 (uniform load over the whole dome) and 2 (uniform load on the inner half of the dome), the linear response of the dome can be studied by considering a sector with appropriate boundary conditions (Section 6.3.3).

2. For the three load conditions considered in this study, the stresses corresponding to the design load (linear analysis) are below the design values specified by the code (NDS, 1986). Of the three load conditions, load condition 1 (uniform load over the whole dome) produces the largest stresses in the members, and therefore, it is more critical in allowable stress design.

3. The linearized eigenvalue prediction gives a good estimate of the buckling load for rigid connections. The tangent stiffness matrices are computed at zero load and at the deisgn load, which results in a simple and straightforward application of the solution procedure (Section 6.7.1). In this study, the eigenvalue buckling prediction for rigid joints was used as a guide (an upper limit) in the computation of the buckling loads for flexible joints.

4. The buckling analyses of the three load conditions show that the lowest buckling load corresponds to load condition 3 (uniform snow load on half of the dome). For 50% joint stiffness, the buckling load for this condition is, respectively, 23% and 52% less than for load conditions 1 and 2. In all three cases, the maximum stresses remain below the proportional limit of the material (Wood Handbook, 1987). Therefore, the failure of the dome model is governed by elastic instability and not by material strength.

5. The buckling response of the test model is influenced by the flexibility of the joints. For example, the reductions in the buckling loads from rigid to 50% joint stiffness for load conditions 1, 2, and 3 are, respectively, 32%, 6%, and 23%.

Chapter 7

CONCLUSIONS AND RECOMMENDATIONS

A finite element modeling and geometrically nonlinear static analysis of glued-laminated timber domes is presented. The modeling assumptions and analysis procedures are applied to a dome model whose geometry is based on a triax dome built in Raleigh, NC in 1975. This spherical dome has a span of 133 ft and a rise of 18 ft above the tension ring. It consists of triangulated networks of curved southern pine glulam beams and purlins connected by steel hubs. All the members have the same radius of curvature and lie on great circles of a spherical surface of 133.3 ft radius. The dome is covered with a 2-inch tongue-and-groove wood decking. In this study, the load carrying capacity of the decking is not considered.

The glulam members of the test dome are modeled with one-dimensional, 3-node, isoparametric structural beam elements which incorporate shear deformations. Material properties are assumed to be constant throughout the volume of the element. The material is assumed to be continuous, homogeneous, and transversely isotropic. The transverse isotropy assumption for glulam beams, proposed in this study, is validated by testing small southern pine samples in torsion. The torsion experiments also allow the computation of the shear modulus of the material. Based on these simplifying assumptions for glulam beams, a geometrically nonlinear 3-node isoparametric beam element is formulated. The accuracy of the modeling assumptions for glulam beams is verified by comparing the analytical and experimental linear responses of southern pine glulam beams subjected to combined loads. The results show that the isobeam element can accurately represent the overall linear response of the beams.

The dome model is analyzed for three load conditions using the commercial finite element program ABAQUS. The following guidelines for the modeling and analysis of glulam domes are included in this study:

1. the specification of boundary conditions for the whole dome and for a symmetric sector of the dome;

2. the determination of design loads, particularly wind and snow loads, and the specification of important load combinations;

3. the modeling of the space joints and the purlin-to-beam connectors;

4. the selection of finite elements to model the beams, connections, and tension ring;

5. the specification of material properties;

6. the selection of solution procedures to evaluate the complete structural response of lattice domes up to collapse. Specifically, three distinct analyses are illustrated: a linear analysis to check the member design criteria specified by the design codes (NDS, 1986), a linearized buckling prediction analysis to estimate the buckling load, and an incremental iterative geometric nonlinear analysis by the modified Riks-Wempner method.

Since the decking of the dome is not modeled in this study, the applied pressures are transformed into discrete member loads. The response of the dome model is evaluated for rigid and flexible joints, and a discussion of the results is presented.

7.1 Conclusions

1. The results of torsion tests of small solid and glulam southern pine samples show that the assumption of transverse isotropy for southern pine glulam beams is acceptable. Moreover, the same experiments can be used to obtain the shear modulus of the material.

2. The 3-d isoparametric beam finite element can characterize the overall linear response of southern pine glulam beams fairly accurately.

3. Since the formulation of the isoparametric beam element is based on Timoshenko's beam theory, which includes shear deformations, the displacements and rotations are interpolated independently. Therefore, the applied distributed element loads and moments are uncoupled and can be transformed into applied discrete nodal forces by integrating the corresponding load parameter over the length of the element. In the analysis of large glulam domes, it is more manageable to discretize the loads independently of the main analysis program and to specify nodal concentrated forces.

4. For uniform loads, the linear response of glulam domes can be studied by considering a sector of the dome.

5. For the condition of rigid joints, the linearized buckling prediction analysis can estimate fairly well the ultimate load capacity of the test model; for flexible joints, this value can be used as an upper estimate of the critical load.

6. For the three load conditions considered in the study of the test model, the failure mode of the dome is governed by elastic instability. At ultimate load, the maximum stresses remain below the proportional limit of the material. Thus, geometric nonlinearities dominate the buckling response of the dome. However, the ends of the beams attached to the steel hub connector may be subjected to high localized stresses, and failure may be initiated by perpendicular-to-grain splitting of the beams.

7. The unsymmetric load condition (uniform snow load over half the dome model) produces the most critical buckling load of the three load conditions considered.

8. The flexibility of the joints have a significant influence on the buckling response of the dome model. The ultimate load decreases as the flexibility of the joints increases.

7.2 Recommendations

1. The isobeam element can represent accurately the 3-d response of space glulam beams. However, its accuracy depends on how well the material is modeled; specifically:

   - Torsion tests of full-size glulam beams of various species are needed in order to:

     i) assess the validity of transverse isotropy;

     ii) compute the shear modulus of the material;

     iii) study the torsional response of glulam beams to establish proportional limit and ultimate shear stresses, to evaluate the influence of warping displacements on the torsional stiffnesses (of particular importance in geometric nonlinear analysis), to identify torsional failure mechanisms, and to develop a material torsional model for glulam.

- In order to specify constant properties for the volume of the beam element, experimental tests and statistical analyses are needed for the accurate estimate of the elastic and shear moduli of glulam beams.

2. The response of the dome model considered in this study should be evaluated for other load conditions, including snow/wind load combinations, snow concentrations, special hanging loads, and earthquake loads. Critical load conditions can then be identified.

3. The development of pre- and postprocessing programs are required for the efficient analysis of glulam domes.

4. Experimental tests of full or scaled models of glulam space frames and lattice domes are required in order to verify the analytical results.

5. The analysis of glulam domes must include the effect of the decking.

6. The effect of variations in the joint stiffnesses, the tension ring stiffness, and the depth-to-width ratios of the beams should be studied.

7. A probabilistic sensitivity study of the effects of variations in material properties on the behavior of the space beams should be conducted.

8. Time dependent effects of creep and moisture change should be evaluated.

9. Other lattice dome geometries, such as the parallel lamella dome, can be modeled with glulam members to study possible advantages over the Triax geometry.

10. Most importantly, thc study should be extended to other glulam dome configurations to determine if elastic instability is the governing design criterion for the range of dome configurations characteristic of existing domes.

11. The research results can be transformed into design and analysis guidelines for codes of practice.

REFERENCES

ABAQUS, Hibbitt, Karlsson & Sorensen, Inc., 100 Medway Street, Providence, RI.

Ahmed, Z., "Nonlinear Finite Element Creep Analysis of a Truss," Internal Report, Department of Civil Engineering, Structures Division, Virginia Polytechnic Institute and State University, Dec. 1988.

AITC 117-87-DESIGN, *Laminating Specifications*, American Institute for Timber Construction.

AITC 117-87-MANUFACTURING, *Standard Specifications for Structural Glued Laminated Timber of Softwood Species*, American Institute of Timber Construction.

American Institute of Timber Construction, *Timber Construction Manual*, 3rd ed., John Wiley & Sons, Inc., New York, 1985.

American National Standard ANSI/AITC A190.1-1983. *Structural Glued Laminated Timber*, American National Standards Institute.

Baker, S., "A Comparison of the Codes of Practice Used in Different Countries for the Determination of Wind Loads on Domes," *Analysis, Design and Construction of Braced Domes*, Z. S. Makowski, Ed., Nichols Publishing Company, New York, 1984, pp. 315-336.

Barlow, J., "Optimal Stress Locations in Finite Element Models," *International Journal for Numerical Methods in Engineering*, Vol. 10, 1976, pp. 243-251.

Bathe, K. J., *Finite Element Procedures in Engineering Analysis*, Prentice-Hall, Englewood Cliffs, NJ, 1982.

Bathe, K. J., "Finite Element Procedures for Solids and Structures - Nonlinear Analysis," Video Course Study Guide, *MIT Center for Advanced Engineering Study*, Reorder No. 73-2200, 1986.

Bathe, K. J. and Bolourchi, S., "Large Displacement Analysis of Three-Dimensional Beam Structures," *International Journal for Numerical Methods in Engineering*, Vol. 14, 1979, pp. 961-986.

Bathe, K. J. and Cimento, A. P., "Some Practical Procedures for the Solution of Nonlinear Finite Element Equations," *Journal of Computer Methods in Applied Mechanics and Engineering*, vol. 22, 1980, pp. 59-85.

Bathe, K. J. and Chandhary, A., "On the Displacement Formulation of Torsion of Shafts with Rectangular Cross-sections," *International Journal for Numerical Methods in Engineering*, Vol. 18, 1982, pp. 1565-1568.

Belytschko, T., Schwer, L., Klein, M. J., "Large Displacement, Transient Analysis of Space Frames," *International Journal for Numerical Methods in Engineering*, Vol. 11, 1977, pp. 65-84.

Biblis, E. J., "Shear Deflection of Wood Beams," *Forest Products Journal*, November 1965, pp. 492-498.

Bodig, J. and Jayne, B. A., *Mechanics of Wood and Wood Composites*, Van Nostrand Reinhold Company, New York, 1982.

Broker, F. W. and Schwab, E., "Torsionprufung von Holz," *Holz als Roh- und Werkstoff*, vol. 46, 1988, pp. 47-52.

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, VA, December 1983.

Chu, Kuang-Han and Rampetsreiter, R. H., "Large Deflection Buckling of Space Frames," *Journal of the Structural Division*, ASCE, vol. 98 (ST12), December 1972, pp. 2701-2722.

Cohen, D., "Effects of Thermal Cycling on Matrix Cracking and Stiffness Changes in Composite Tubes," M.S. Thesis, Virginia Polytechnic Institute and State University, August 1984, pp. 126-129.

Conners, T. E., "Segmented Models for Stress-Strain Diagrams," *Wood Science and Technology*, Vol. 23, 1989, pp. 65-73.

Connor, J. J., Logcher, R. D. and Chan, S. C., "Nonlinear Analysis of Elastic Framed Structures," *Journal of the Structural Division*, ASCE, vol. 94 (ST6), June 1968, pp. 1525-1545.

Cook, R. D., *Concepts and Applications of Finite Element Analysis*, 2nd ed., John Wiley and Sons, New York, 1981.

Crisfield, M. A., "A Fast Incremental/Iterative Solution Procedure That Handles Snap-Through," *Computers & Structures*, vol. 13, 1981, pp. 55-62.

Davalos, J. F., "Background for Finite Element Analysis and Experimental Testing of Glued-Laminated Space Beams," M.S. Thesis, Virginia Polytechnic Institute and State University, July 1987.

Davalos, J. F., Loferski, J. R., and Holzer, S. M., "Verification of a 3-D Glulam Beam Finite Element," *Proceedings of the 1988 International Conference on Timber Engineering*, Seattle, WA, Vol. 2, pp. 194-204.

Dewey, G. R., "Analysis and Testing of Single and Double-Tapered Glulam Beams," M.S. Thesis, Colorado State University, November 25, 1984.

Engineering News Record:
"Collapses, Failures Take Heavy Toll," May 19, 1966.

"Snow, Wind Triggered Dome Collapse," May 19, 1966.
"Winds Topple 207 Tons of Steel at Auditorium," November 5, 1970.
"Geodesic Gymnasium Sags in Severe Weather," February 4, 1971.
"Geodesic Gymnasium Sags Again Before Previous Damage is Repaired," February 29, 1971.
"Space Frame Roofs Collapse Following Heavy Snowfall," January 26, 1978.

Eshelby, R. W. and Evans, R. J., "Design Procedures for Reticulated Timber Domes," *Proceedings of the 1988 International Conference on Timber Engineering*, Seattle, WA, Vol. 1, pp. 283-287.

European Convention for Constructional Steelwork, Recommendations for the Calculation of Wind Effects on Buildings and Structures, Technical Committee T12: Wind Effects, September 1978.

Foschi, R. O. and Barrett, J. D., "Glued-Laminated Beam Strength: A Model," *Journal of the Structural Division*, ASCE, Vol. 106, (No. ST8), 1980, pp. 1735-1755.

Frederick, D. and Chang, C. S., *Continuum Mechanics*, Scientific Publishers, Inc., Cambridge, 1972.

Fung, Y. C., *A First Course in Continuum Mechanics*, Prentice-Hall Inc., Englewood Cliffs, NJ, 1969.

Goodman, J. and Bodig, J., "Orthotropic Elastic Properties of Wood, *Journal of the Structural Division*, ASCE, Vol. 96, No. ST 11, November 1970, pp. 2301-2319.

Grossman, P. U. A., "Constitutive Equations Governing the Rheological Behavior of Wood," *Proceedings of the Fourth Australasian Conference on the Mechanics of Structures and Materials*, Queensland, Australia, August 1973, pp. 97-99.

Gupta, A. K., and Mohraz, B., "A Method of Computing Numerically Integrated Stiffness Matrices," *International Journal for Numerical Methods in Engineering*, Vol. 5, 1972, pp. 83-89.

Gurfinkel, G., *Wood Engineering*, 2nd ed., Kendall/Hunt, 1973.

Hao, N. A., "Parallel Lamella Dome Under Wind and Snow Loads," M.S. Thesis, Virginia Polytechnic Institute and State University, April 1986.

Hinton, E. and Cambell, J. S., "Local and Global Smoothing of Discontinuous Finite Element Functions Using a Least Squares Method," *International Journal for Numerical Methods in Engineering*, Vol. 8, 1974, pp. 461-480.

Holloway, L. and Rustum, A., "Buckling Analysis of a Shallow Dome Manufactured from Pultruded Fibre/Matrix Composites," *Proceedings of the Third International Conference on Space Structures*, H. Nooshin, ed., Elsevier Applied Science Publishers, London and New York, 1984, pp. 545-553.

Holzer, S. M. et al., "Dynamic Considerations in Latticed Structures," *Journal of the Structural Division*, ASCE, Vol. 110, No. 10, October 1984, pp. 2547-2550.

Holzer, S. M. and Huang, C. Y., "Stability Analyses by the Finite Element Method," Report No. CE/VPI-ST89/05, Virginia Polytechnic Institute and State University, Blacksburg, VA, August, 1989.

Holzer, S. M. and Loferski, J. R., "Analysis of Glued-Laminated Timber Space Frames and Lattice Domes," U.S.D.A. proposal, May 1986.

Holzer, S. M. and Loferski, J. R., "Background for Research on Glulam Lattice Domes," *Proceedings of the Sixth Annual Structures Congress*, ASCE Structural Division, Orlando, FL, August 1987, pp. 305-318.

Holzer, S. M., Dillard, D. A., and Loferski, J. R., "A Review of Creep in Wood: Concepts Relevant to Develop Long-Term Behavior Predictions for Wood Structures," *Wood and Fiber Science*, 1989, to appear.

Holzer, S. M., Huang, C. Y., Davalos, J. F. and Loferski, J. R., "Analysis of Glulam Lattice Dome," *Proceedings of the Seventh Structures Congress*, ASCE Structural Division, San Francisco, CA, May 1989, pp. 914-920.

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, MO, October 1981, pp. 179-209.

Hsu, T. T. C., *Torsion of Reinforced Concrete*, Van Nostrand Reinhold Col, New York, 1984, pp. 3-49.

Jau, J. J., *Geometrically Nonlinear Finite Element Analysis of Space Frames*, Ph.D. Dissertation, Virginia Polytechnic Institute and State University, 1985.

Jirousek, J., "On the Capacity of the Isoparametric Quadratic Beam Element to Solve Accurately a Uniformly Loaded Beam Segment," Technical Note, *Computers and Structures*, Vol. 19, No. 4, 1984, pp. 693-695.

Jones, R. M., *Mechanics of Composite Materials*, Hemisphere Publishing Corporation, New York, 1984, pp. 32-35.

Kani, I. M., "A Theoretical and Experimental Investigation of the Collapse of Shallow Reticulated Domes," Ph.D. Dissertation, University of Cambridge, February 1986.

Kardestuncer, H. and Norrie, D. H., eds., *Finite Element Handbook*, McGraw-Hill Book Company, 1987.

Kline, D. E., Woeste, F. E., and Bendtsen, B. A., "Stocastic Model for Modulus of Elasticity of Lumber," *Wood and Fiber Sci.*, Vol. 18, No. 2, 1986, pp. 228-238.

Langhaar, H. L., *Energy Methods in Applied Mechanics*, Wiley, New York, 1962.

Lekhnitskii, S. G., *Theory of Elasticity of an Anisotropic Body*, Holden Day, San Francisco, CA, 1963, pp. 197-204.

Loferski, J. R., Davalos, J. R. and Yadama, V., "A Laboratory Built Clip-on Strain Gage Transducer for Testing Wood," *Forest Products Journal*, 1989, accepted for publication.

Lorenzen, R. T., "Progressive Collapse of Clearspan Structures," *American Society of Agricultural Engineers Summer Meeting*, San Antonio, TX, 1980, paper No. 80-4001.

Makowski, Z. S., "A History of the Development of Domes and a Review of Recent Achievements World-Wide," *Analysis, Design and Construction of Braced Domes*, Z. S. Makowski, ed., Nichols Publishing Company, New York, 1984, pp. 1-85.

Malvern, L. E., *Introduction to the Mechanics of a Continuous Medium*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1969.

Minimum Design Loads for Buildings and Other Structures, ANSI A58.1-1982, American National Standards Institute, Inc., 1430 Broadway, NY, 10018, 1982.

Mohler, K. and Hemmer, K., "Verformungs- und Festigkeitsverhalten von Nadelvoll- und Brettschichtholz bei Torsionbeanspruchung," *Holz als Roh- und Werkstoff*, Vol. 35, 1977, pp. 473-478.

National Forest Products Association, 1986, *National Design Specifications for Wood Construction*, Washington, D.C.

Natterer, J., "The Future of Timber Structures," *Civil Engineering*, Oct. 1987, p. 6.

Neal, D. W., *The Triax Dome*, Culbertson, Noren, and Neal, Consulting Engineers, 1410 S. W. Morrison St., Portland, OR 97205, March 1973.

Newberry, C. W. and Eaton, K. J., *Wind Loading Handbook*, Building Research Establishment Report, HM Stationery Office, London, 1974.

Newlin, J. A. and Trayer, G. W., "Deflections of Beams with Especial Reference to Shear Deformations," *National Advisory Committee for Aeronautics Rpt. No. 180*, Washington, D.C., 1924.

O'Halloran, M. R., *A Curvilinear Stress-Strain Model for Wood in Compression*, Ph.D. Dissertation, Colorado State University, Fort Collins, CO, 1973.

Papadrakakis, M., "Post-buckling Analysis of Spatial Structures by Vector Iteration Methods," *Computers & Structures*, vol. 14, 1981, pp. 393-402.

Roark, R. J., and Warren, C. Y., *Formulas for Stress and Strain*, fifth ed., McGraw-Hill Book Company, NY, 1982.

Sage, W. M., "Nonlinear Analysis of Cable Structures," M.S. Thesis, Virginia Polytechnic Institute and State University, Blacksburg, December 1986.

Scharr, G., "Beitrag zur Torsionselastizitat von Holzern in Abhangigkeit von der Holztemperatur und der Belastungszeit," *Holz als Roh- und Werkstoff*, Vol. 44, 1986, pp. 57-60.

Schniewind, A. P., "Mechanical Behavior and Properties of Wood," *Wood as a Structural Material*, A. G. H. Dietz, et al., eds., EMMSE, Materials Research Laboratory, Pennsylvania State University, University Park, PA 16802, 1982, p. 59.

Semenov, P. I., "Determination of Shear Moduli of Orthotropic Materials From Torsion Tests," *Mekhanika Polimerov*, Vol. 2, No. 1, 1966, pp. 27-33.

Soare, M. V., "Investigation of the Collapse of a Large-Span Braced Dome," *Analysis Design and Construction of Braced Domes*, Z. S. Makowski, ed., Nichols Publishing Company, New York, 1984, pp. 161-173.

Synge, J. L.,and Griffith, B. A., *Principles of Mechanics*, third ed., McGraw-Hill Book Company, NY, 1959.

Tacoma Dome, *Pacific Builder and Engineer*, February 7, 1983.

Tang, R. C., Adams, S. F., and Mark, R. E., "Moduli of Rigidity and Torsional Strength of Scarlet Oak Related to Moisture Content," *Wood Science*, Vol. 3, No. 4, April 1971, pp. 238-244.

The BOCA Basic/National Building Code, Building Officials and Code Administrators International, Inc., Homewood, IL, 1984.

Timoshenko, S. P., *Strength of Materials, Part I, Elementary Theory and Problems*, 3rd ed., Van Nostrand, New York, 1955.

Timoshenko, S. P. and Goodier, J. N., *Theory of Elasticity* Third Edition, McGraw-Hill Book Company, 1970, pp. 291-315.

Trayer, G. W. and March, H. W., *The Torsion of Members Having Sections Common in Aircraft Construction*, National Advisory Committee for Aeronautics, Report No. 334, 1930.

Tsuboi, Y., et al., "Analysis, Design and Realization of Space Frames," A State-of-the-Art Report by the I.A.S.S. Working Group on Spatial Steel Structures, *Bulletin of the International Association of Shell and Spatial Structures*, n. 84/85, Vol. XXV-1/2, April-August 1984.

Uniform Building Code, 1985, International Conference of Building Officials, Whittier, CA.

United States Department of Agriculture, 1987, *Wood Handbook: Wood as an engineering material*, U.S. Forest Products Laboratory, Agriculture Handbook No. 72, U.S. Government Printing Office, Washington, D.C.

Van Erp, G. M., "The Stability of Timber Portal Frames and Arch Frames," *Heron*, Vol. 30, No. 2, 1985, pp. 1-52.

Varax Domes, Specifications, Section 13500 Varax Dome Structure and Roof Decking, Western Wood Structures Inc., Tualatin, OR.

Williams, F. W., "An Approach to the Non-Linear Behavior of the Members of a Rigid Jointed Plane Framework with Finite Deflections," *Quarterly Journal of Mechanics and Applied Mathematics*, vol. 17, part 4, 1964, pp. 451-469.

Wood, R. D. and Zienkiewicz, O. C., "Geometrically Non-linear Finite Element Analysis of Beams, Frames, Arches and Axisymmetric Shells," *Computers & Structures*, vol. 7, 1977, pp. 725-735.

Yadama, V., "Experimental Verification of a 3-d Glulam Beam Finite Element Model," M.S. Thesis, Virginia Polytechnic Institute and State University, 1989.

Yang, Y. B., "Linear and Nonlinear Analysis of Space Frames with Nonuniform Torsion Using Interactive Computer Graphics," Ph.D. Dissertation, Cornell University, 1984.

Appendix A

Data Sequence

```
----------------------------MAIN--------------------------------------
DATE      ( FORMAT = --/--/-- )
HEADING   (MAXIMUM 80 CHARACTERS )
NE NJ NLC
TYPE
---------------------------STRUCT------------------------------------
MINC(I,J) I = 1,3  J = 1,NE -------- ( MEMBER INCIDENCES )
JNUM JDIR   END WITH 0 0 ------------ ( JOINT CONSTRAINTS )
-------------------------------PROP----------------------------------
NJ X1 X2 X3 VS1 VS2 VS3 VT1 VT2 VT3   END WITH 0 0 0... (NODE PROPS.)
IELEM WIDTH DEPTH EMOD GMOD           END WITH 0  (ELEMENT PROPS.)
----------------------------- LOAD ---------------------------------
IF TYPE = 'NONLIN' THEN QIMAX QI DQI
--------------------------- JLOAD ----------------------------------
FOR EACH LOAD CONDITION:
JNUM JDIR FORCE      END WITH 0 0 0.D0
--------------------------- DLOAD ----------------------------------
FOR EACH LOAD CONDITION:
ENUM PA1 PA2 PA3 PB1 PB2 PB3 RADIUS (RADIUS = 0.D0 FOR STRAIGHT ELEMENT)
           END WITH A LINE OF EIGHT ZEROES
--------------------------- NONLIN ---------------------------------
          ( ONLY FOR NONLINEAR ANALYSIS )
TOLENE,TOLFOR,TOLDIS
ALGO
IRES
MAXIT,ITEUPD
*************** FOR NEWVAR ADD THE FOLLOWING 3 VALUES AT THE END OF THE
*************** FILE
DQIMAX,DQIMIN,DTOL
----------------------------- DPOST ------------------------------
JNUM ( FOR NODES WHOSE DISPLACEMENTS ARE TO BE PRINTED ) < END WITH 0 >
ELEMN ( FOR ELEMENTS WHOSE STRESSES AND STRAINS ARE TO BE PRINTED )
                         < END WITH 0 >
ELEMN ( FOR ELEMENTS WHOSE INTERNAL FORCES ARE TO BE PRINTED )
```

```
                         < END WITH 0 >
JNUM ( FOR NODES WHOSE NODAL FORCES  ARE TO BE PRINTED ) < END WITH 0 >

------------------------------ END ---------------------------------
*******************************************************************
*                                                                 *
*EXAMPLE: INPUT DATA FILE FOR TEST PROBLEM 1: WILLIAMS TOGGLE FRAME 1.*
*                                                                 *
*******************************************************************
```

```
07/29/89
WILLIAMS TOGGLE FRAME1
2 5 1
NONLIN
1 3 2
3 5 4
1 1
1 2
1 3
1 4
1 5
1 6
5 1
5 3
5 4
5 5
5 6
0 0
1 0.D0 0.D0 0.D0 -0.024716D0 0.9996945D0 0.D0 0.D0 0.D0 1.D0
2 3.236D0 0.08D0 0.D0 -0.024716D0 0.9996945D0 0.D0 0.D0 0.D0 1.D0
3 6.472D0 0.16D0 0.D0 -0.024716D0 0.9996945D0 0.D0 0.D0 0.D0 1.D0
4 9.7073D0 0.24D0 0.D0 -0.024716D0 0.9996945D0 0.D0 0.D0 0.D0 1.D0
5 12.943D0 0.32D0 0.D0 -0.024716D0 0.9996945D0 0.D0 0.D0 0.D0 1.D0
0 0.D0 0.D0 0.D0 0.D0 0.D0 0.D0 0.D0 0.D0 0.D0
1 0.753D0 0.243D0 10.36D06 3.9615D06
2 0.753D0 0.243D0 10.36D06 3.9615D06
0 0.D0 0.D0 0.D0 0.D0
10.D0 1.D0 0.3D0
5 2 -35.D0
0 0 0.D0
0 0.D0 0.D0 0.D0 0.D0 0.D0 0.D0 0.D0
5.01D0 3.1D0 0.01D0
RIKWEM
1
100,5
5
0
1
0
1
0
1
0
```

Appendix B

Program Listing

```
C***********************************************************************
C                    MAIN
C***********************************************************************
C
C   THIS PROGRAM EMPLOYS A 3-D, 3-NODE, ISOPARAMETRIC  BEAM ELEMENT
C   FOR LINEAR AND NONLINEAR ANALYSIS OF RIGID-JOINTED SPACE FRAMES
C   CONSISTING OF EITHER STRAIGHT OR CURVED MEMBERS.
C -----------------------------**********-------------------------
C   'MAIN' STARTS THE ANALYSIS PROCEDURE. IT INITIALIZES THE VARIABLES
C   SETS UP THE POINTERS  AND CALLS 'STRUCT', 'LOAD' AND 'LINEAR' FOR
C   LINEAR ANALYSIS AND 'STRUCT', 'LOAD', AND 'NONLIN' FOR NONLINEAR
C   ANALYSIS. IT ALSO CHECKS THE MEMORY ALLOCATIONS AGAINST THE MEMORY
C   REQUIREMENTS BEFORE CALLING THESE SUBROUTINES.
C
      IMPLICIT REAL*8(A-H,O-Z)
      CHARACTER*(*) TYPE*6,ALGO*6,HEAD*27,DATE*8
      PARAMETER (LIMIT = 100000)
      DIMENSION A(LIMIT)
      READ(5,'(A)') DATE
      READ(5,'(A)') HEAD
      READ(5,*) NE,NJ,NLC
      READ(5,'(A)') TYPE
C
      WRITE(6,13) HEAD
 13   FORMAT(T20,A)
      WRITE(6,15) DATE
 15   FORMAT(T20,A)
      IF(TYPE .EQ. 'NONLIN') THEN
        NLC = 1
      ENDIF
C
      WRITE(6,10) 'CONTROL VARIABLES:',
     .        'NUMBER OF ELEMENTS',NE,'NUMBER OF NODES',NJ,
     .        'NO. OF LOAD CONDITIONS:',NLC
```

```
  10 FORMAT(///T5,A/T5,18('_')//2(T5,A,T30,I4/),T5,A,I6)
     WRITE(6,20) TYPE
  20 FORMAT(/T10,'TYPE OF ANALYSIS:   ',A)
C
C---  SET POINTERS FOR STRUCT.
C
     NMINC = 1
     NJCODE = NMINC + 3*NE
     NMCODE = NJCODE + 6*NJ
     NWIDTH = NMCODE + 18*NE
     NDEPTH = NWIDTH + NE
     NEMOD = NDEPTH + NE
     NGMOD = NEMOD + NE
     NX = NGMOD + NE
     NVR = NX + 3*NJ
     NVS = NVR + 3*NJ
     NVT = NVS + 3*NJ
     NMAXA = NVT + 3*NJ
C
     NEQ = 6 * NJ
C
     NKHT = NMAXA + (NEQ + 1)
     NMAX = NKHT + NEQ - 1
C
C---  CHECK MEMORY REQUIREMENTS
C
     IF(NMAX .LE. LIMIT) THEN
       CALL STRUCT(A(NMINC),A(NJCODE),A(NMCODE),A(NWIDTH),A(NDEPTH),
     .          A(NEMOD),A(NGMOD),A(NX),A(NVR),A(NVS),A(NVT),
     .          A(NKHT),A(NMAXA),NE,NJ,NEQ,LSS)
     ELSE
       WRITE(6,25) NMAX
 25     FORMAT(/'INADEQUATE MEMORY FOR STRUCT; INCREASE MEMORY TO',I9)
     STOP
     ENDIF
C
C---  SET POINTERS FOR LOAD.
C
     NQ = NMAXA + NEQ + 1
     NQBAR = NQ + NEQ
     NMAX =  NQBAR + NEQ
C
     IF(NMAX .GT. LIMIT) THEN
       WRITE(6,30) NMAX
 30     FORMAT(/'INADEQUATE MEMORY FOR LOAD; INCREASE MEMORY TO',I9)
       STOP
     ENDIF
C
C---  SET POINTERS FOR LINEAR AND NONLINEAR ANALYSIS
C
       NSSS = NQBAR + NEQ
       NR = NSSS + LSS
       NS = NR + 3
       NT = NS + 5
       NWTR = NT + 5
       NWTS = NWTR + 3
       NWTT = NWTS + 3
       NB = NWTT + 3
       NH = NB + 3*3*6
       NDH = NH + 3
       NG = NDH + 3
       NXJINV = NG + 3*3*3
       NTTT = NXJINV + 3*3
```

```
        ND = NTTT + NEQ
        NEPS = ND + NEQ
        NSTRS = NEPS + 3
        NDNL = NSTRS + 3
        NFINT = NDNL + 6 * 3
        NP = NFINT + 18
        NVVR = NP + 6 * NJ
        NVVS = NVVR + 3
        NVVT = NVVS + 3
        NBNL = NVVT + 3
        NDS1 = NBNL + 2 * 2 * 2 * 3 * 9 * 6
        NDS2 = NDS1 + 2 * 2 * 2
        NDS3 = NDS2 + 2 * 2 * 2
        NMAX = NDS3 + 2 * 2 * 2
C
C--- SET ADDITIONAL POINTERS FOR NONLINEAR ANALYSIS
C
      IF(TYPE .EQ. 'NONLIN') THEN
        NDD = NDS3 + 2 * 2 * 2
        NDD0 = NDD + NEQ
        NF = NDD0 + NEQ
        NFP = NF + NEQ
        NFPI = NFP + NEQ
        NFORCE = NFPI + NEQ
        NRES = NFORCE + NEQ
        NDD01 = NRES + NEQ
        NDDT = NDD01 + NEQ
        NDDP = NDDT + NEQ
        NDD1 = NDDP + NEQ
        NDD2 = NDD1 + NEQ
        NXP = NDD2 + NEQ
        NVRP = NXP + 3 * NJ
        NVSP = NVRP + 3 * NJ
        NVTP = NVSP + 3 * NJ
        NXPP = NVTP + 3 * NJ
        NVRPP = NXPP + 3 * NJ
        NVSPP = NVRPP + 3 * NJ
        NVTPP = NVSPP + 3 * NJ
        NMAX = NVTPP + 3 * NJ
      ENDIF
C
      IF(NMAX .GT. LIMIT) THEN
        WRITE(6,40) NMAX
 40     FORMAT(/'INADEQUATE MEMORY FOR ANALYSIS; INCREASE MEMORY TO',
     .       I9)
        STOP
      ENDIF
C
      DO 50 LC = 1,NLC
        CALL LOAD(A(NQ),A(NQBAR),A(NJCODE),NEQ,QIMAX,QI,DQI,TYPE,LC,
     .          A(NMINC),A(NX) )
C
C--- CHECK MEMORY REQUIREMENTS AND CALL LINEAR OR NONLIN AS THE CASE
C    MAY BE.
C
        IF(TYPE .EQ. 'LINEAR') THEN
          CALL LINEAR(NEQ,QIMAX,A(NQ),A(NQBAR),TYPE,A(NMINC),NE,
     .     A(NMAXA),A(NSSS),A(NJCODE),A(NR),A(NS),A(NT),A(NWTR),A(NWTS),
     .      A(NWTT),LSS,A(NB),A(NEMOD),A(NGMOD),A(NH),A(NDH),A(NDEPTH),
     .      A(NWIDTH),A(NG),A(NVR),A(NVS),A(NVT),A(NXJINV),A(NX),LC,
     .      A(NTTT),A(ND),A(NMCODE),A(NEPS),A(MSTRS),A(NDNL),A(NFINT),
     .    A(NP),A(NVVR),A(NVVS),A(NVVT),A(NBNL),A(NDS1),A(NDS2),A(NDS3))
        ELSE
        CALL NONLIN(A(NQ),A(NQBAR),QI,QIMAX,DQI,A(ND),A(NDD),A(NR),
     .          A(NS),A(NT),A(NEPS),A(NSTRS),A(NJCODE),A(NB),
     .          A(NBNL),A(NH),A(NDH),A(NXJINV),A(NDEPTH),
     .          A(NWIDTH),A(NX),A(NVR),A(NVS),A(NVT),A(NG),
     .          A(NDNL),A(NGMOD),A(NEM0D),A(NWTR),A(NWTS),A(NWTT),
     .          A(NFINT),A(NP),NE,A(NMCODE),A(NVVR),A(NVVS),
     .          A(NVVT),A(NF),A(NFP),A(NFPI),A(NFORCE),A(NTTT),
     .          A(NRES),A(NSSS),A(NMAXA),NEQ,TYPE,LSS,A(NMINC),
     .          A(NDD0),NJ,ALGO,A(NDS1),A(NDS2),A(NDS3),A(NDD01),
     .          A(NDDT),A(NDDP),A(NDD1),A(NDD2),A(NXP),A(NVRP),
     .          A(NVSP),A(NVTP),A(NXPP),A(NVRPP),A(NVSPP),
     .          A(NVTPP) )
       ENDIF
 50   CONTINUE
C
      END
C*******************************************************************
C*                       STRUCT                       *
C*******************************************************************
C
C THIS SUBROUTINE CALCULATES THE SIZE AND ALLOCATES MEMORY FOR THE
C THE SYSTEM STIFFNESS MATRIX.
C IT ALSO READS MEMBER INCIDENCES,JOINT CONSTRAINTS, AND ELEMENT
C PROPERTIES. IT CALL 'CODES', 'SKYLIN', AND 'PROP'.
C
      SUBROUTINE STRUCT(MINC,JCODE,MCODE,WIDTH,DEPTH,EMOD,GMOD,X,VR,VS,
     .           VT,KHT,MAXA,NE,NJ,NEQ,LSS)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION JCODE(6,*),EMOD(*),KHT(*),MAXA(*),MCODE(18,*),
     .       MINC(3,*),X(3,*),VR(3,*),VS(3,*),VT(3,*),WIDTH(*),
     .       DEPTH(*),GMOD(*)
C
C--- READ AND ECHO MEMBER INCIDENCES (MINC)
C
      WRITE(6,10) 'MEMBER INCIDENCES:','MEMBER NO.','A-END','B-END',
     .        'MIDDLE'
  10 FORMAT(//T5,A/T5,18('_'),//T5,A,T20,A,T30,A,T40,A)
C
      DO 30 I = 1,NE
        READ(5,*) MINC(1,I),MINC(2,I),MINC(3,I)
        WRITE(6,20)I,MINC(1,I),MINC(2,I),MINC(3,I)
  20    FORMAT(T8,I3,T21,I3,T31,I3,T41,I3)
  30 CONTINUE
C
      DO 50 I = 1,NE
        DO 40 L = 1,18
          MCODE(L,I) = 0
  40    CONTINUE
  50 CONTINUE
C
      DO 70 I = 1,NJ
        DO 60 L = 1,6
          JCODE(L,I) = 1
  60    CONTINUE
  70 CONTINUE
C
C--- READ AND ECHO JOINT CONSTRAINTS.
C
      READ(5,*) JNUM,JDIR
      IF(JNUM .NE. 0) THEN
        WRITE(6,80) 'NODE CONSTRAINTS','NODE','DIRECTION'
 80     FORMAT(//T10,A/T10,A,T30,A)
```

```
      ENDIF
C
 90   IF(JNUM .NE. 0) THEN
        WRITE(6,100) JNUM,JDIR
 100    FORMAT(T11,I3,T32,I2)
        JCODE(JDIR,JNUM) = 0
        READ(5,*) JNUM,JDIR
        GO TO 90
      ENDIF
C
      CALL CODES(JCODE,MCODE,MINC,NJ,NE,NEQ)
C
      CALL SKYLIN(KHT,MCODE,MAXA,NEQ,LSS,NE)
C
      CALL PROP(WIDTH,DEPTH,EMOD,GMOD,X,VR,VS,VT,NE,NJ)
C
      RETURN
      END
C**********************************************************************
C*                       LOAD                               *
C**********************************************************************
C LOAD IS A CONTROLLING MODULE FOR INPUT OF LOADING CONDITIONS. IT
C INITILIAZES LOAD VECTORS Q AND QBAR TO ZERO. IT READS QIMAX, QI,DQI
C FOR NONLINEAR CONDITION OTHERWISE EQUATES QI AND QIMAX TO 1. IT CALLS
C 'JLOAD' AND 'DLOAD'.
C
      SUBROUTINE LOAD(Q,QBAR,JCODE,NEQ,QIMAX,QI,DQI,TYPE,LC,MINC,X)
      IMPLICIT REAL*8(A-H,O-Z)
      CHARACTER*(*) TYPE*6
      DIMENSION JCODE(6,*),QBAR(*),Q(*),MINC(3,*),X(3,*)
C
      WRITE(6,5) 'LOAD CONDITION',LC
  5   FORMAT(//T18,A,T35,I1/T18,18('_'))
C
C---  INITIALIZE THE VECTOR 'QBAR' WHICH CONTAINS THE CONSTANT
C     PROPORTION OF JOINT LOADS.
C
      DO 10 I = 1,NEQ
        Q(I) = 0.D0
        QBAR(I) = 0.D0
  10  CONTINUE
C
C---  READ THE MAX VALUE OF LAMBDA,INITIAL VALUE OF LAMBDA AND THE
C     INITIAL INCREMENT; LAMBDA = LOADING PARAMETER.   ( FOR NONLINEAR )
C
      IF(TYPE .NE. 'LINEAR') THEN
        READ(5,*) QIMAX,QI,DQI
      ELSE
        QI = 1.D0
        QIMAX = 1.D0
      ENDIF
C
C---  CALL SUBROUTINE FOR JOINT LOADS.
C
      CALL JLOAD(Q,QBAR,JCODE,QIMAX,NEQ)
C
C---  CALL SUBROUTINE FOR DISTRIBUTED LOADS.
C
      CALL DLOAD(Q,QBAR,JCODE,QIMAX,NEQ,MINC,X)
C
C---  PRINT THE VALUES OF THE LOADING PARAMETERS.
C
      IF(TYPE .NE. 'LINEAR') THEN
      WRITE(6,20) QIMAX,QI,DQI
   20 FORMAT(//T5,'LOADING PARAMETER (LAMBDA):'/T5,27('_')/
     .       T5,'MAXIMUM VALUE: ',T24,F8.3/T5,'INITIAL VALUE: ',T24,F8.3
     .      /T5,'INITIAL INCREMENT: ',T24,F8.3)
      ENDIF
C
      RETURN
      END
C*********************************************************************
C                        LINEAR
C*********************************************************************
C
C-- 'LINEAR' IS A CONTROL MODULE FOR LINEAR ANALYSIS. THE SYSTEM
C    MATRIX IS CALCULATED AND STORED IN THE ALREADY ALLOCATED SPACE.
C    ANALYSIS PERFORMED AND THE RESULTS ECHOED. IT CALLS 'STIFF',
C    'POST', 'SOLVE', AND 'RESULT'.
C
      SUBROUTINE LINEAR (NEQ,QIMAX,Q,QBAR,TYPE,MINC,NE,MAXA,SSS,JCODE,R,
     .                S,T,WTR,WTS,WTT,LSS,B,EMOD,GMOD,H,DH,DEPTH,
     .                WIDTH,G,VR,VS,VT,XJINV,X,LC,TTT,D,MCODE,EPS,
     .                STRESS,DNL,FINT,P,VVR,VVS,VVT,BNL,DS1,DS2,
     .                DS3)
      IMPLICIT REAL*8(A-H,O-Z)
      CHARACTER*(*) TYPE*6
      DIMENSION Q(*),QBAR(*),MINC(3,*),MAXA(*),SSS(*),JCODE(6,*),R(*),
     .          S(*),T(*),WTR(*),WTS(*),WTT(*),B(3,3,*),EMOD(*),GMOD(*),
     .          H(*),DH(*),DEPTH(*),WIDTH(*),G(3,3,*),VR(3,*),VS(3,*),
     .          VT(3,*),XJINV(3,*),X(3,*),TTT(*),D(*),MCODE(18,*),
     .          STRESS(*),DNL(6,*),FINT(*),P(6,*),EPS(*),VVR(*),VVS(*),
     .          VVT(*),BNL(2,2,2,3,9,*),DS1(2,2,*),DS2(2,2,*),
     .          DS3(2,2,*)
C
      DO 10 I = 1,NEQ
        Q(I) = QBAR(I) * QIMAX
   10 CONTINUE
C
      CALL STIFF (NE,MINC,R,S,T,WTR,WTS,WTT,TYPE,JCODE,SSS,MAXA,
     .             LSS,B,EMOD,GMOD,H,DH,VR,VS,VT,DEPTH,WIDTH,G,
     .             XJINV,X,VVR,VVS,VVT,BNL,DS)
C
      DO 20 I = 1,NEQ
        TTT(I) = Q(I)
   20 CONTINUE
      CALL SOLVE(SSS,TTT,MAXA,NEQ,LC)
C
      DO 30 I = 1,NEQ
        D(I) = TTT(I)
   30 CONTINUE
C
C--- PERFORM POSTPROCESSING.
C
      CALL POST(R,S,T,EPS,STRESS,MINC,JCODE,D,B,H,DH,XJINV,DEPTH,
     .        WIDTH,X,VR,VS,VT,G,DNL,GMOD,EMOD,WTR,WTS,WTT,
     .        FINT,P,NE,MCODE,VVR,VVS,VVT)
C
C--- ECHO THE RESULTS
C
      CALL RESULT(1.D0)
C
      RETURN
      END
C*********************************************************************
C*                        CODES
```

```
C*******************************************************************
C
C    'CODES' CALCULATES AND SAVES THE JOINT CODE MATRIX JCODE AND THE
C    MEMBER CODE MATRIX MCODE.
C
      SUBROUTINE CODES(JCODE,MCODE,MINC,NJ,NE,NEQ)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION JCODE(6,*),MCODE(18,*),MINC(3,*)
      NEQ = 0
C
C--- GENERATE JCODE
C
      DO 20 J = 1,NJ
        DO 10 L = 1,6
          K = JCODE(L,J)
          IF(K .NE. 0) THEN
            NEQ = NEQ + 1
            JCODE(L,J) = NEQ
          ENDIF
   10   CONTINUE
   20 CONTINUE
      DO 40 I = 1,NE
        IM = MINC(1,I)
        JM = MINC(2,I)
        KM = MINC(3,I)
        DO 30 L = 1,6
          MCODE(L,I) = JCODE(L,IM)
          MCODE(L+6,I) = JCODE(L,JM)
          MCODE(L+12,I) = JCODE(L,KM)
   30   CONTINUE
   40 CONTINUE
C
      RETURN
      END
C*******************************************************************
C*                       SKYLIN
C*******************************************************************
C
C    IT DETERMINES THE LENGTH OF THE TANGENT STIFFNESS VECTOR SS, THE
C    VECTOR WHICH STORES THE ADDRESSES OF THE MAIN DIAGONAL TERMS MAXA,
C    AND THE NUMBER OF ELEMENTS BELOW THE SKYLINE OF EACH COLUMN.
C
      SUBROUTINE SKYLIN(KHT,MCODE,MAXA,NEQ,LSS,NE)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION KHT(*),MAXA(*),MCODE(18,*)
C
C--- INITIALIZE KHT(K) = 0
C
      DO 10 I = 1,NEQ
      KHT(I) = 0
   10 CONTINUE
C
C--- GENERATE KHT
C
      DO 40 I = 1,NE
        MIN = NEQ
        DO 20 J = 1,18
          N = MCODE(J,I)
          IF(N .NE. 0 .AND. N .LT. MIN) THEN
            MIN = N
          ENDIF
   20   CONTINUE
        DO 30 J =1,18
```

```
          K = MCODE(J,I)
          IF(K .NE. 0 .AND. K .NE. MIN) THEN
            KHT(K) = MAX0(KHT(K),K-MIN)
          ENDIF
   30   CONTINUE
   40 CONTINUE
      WRITE(6,50)
   50 FORMAT(//'-',11X,'I',10X,'KHT(I)',10X,'MAXA(I)')
C
C--- GENERATE MAXA
C
      MAXA(1) = 1
      DO 70 I = 1,NEQ
        WRITE(6,60)I,KHT(I),MAXA(I)
   60   FORMAT('0',7X,I5,9X,I5,11X,I5)
        MAXA(I+1) = MAXA(I) + KHT(I) +1
   70 CONTINUE
      LSS = MAXA(NEQ+1) - 1
      I = NEQ + 1
      WRITE(6,80) I,MAXA(I),LSS
   80 FORMAT('0',7X,I5,25X,I5//7X,'LSS = ',I5)
C
      RETURN
      END
C**********************************************************************
C*                      PROP
C**********************************************************************
C
      SUBROUTINE PROP(WIDTH,DEPTH,EMOD,GMOD,X,VR,VS,VT,NE,NJ)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION WIDTH(*),DEPTH(*),EMOD(*),GMOD(*),X(3,*),VR(3,*),
     .        VS(3,*),VT(3,*)
C
C--- READ AND ECHO JOINT COORDINATES FOR THE STRUCTURE. IF JNT. NO. = 0
C    TERMINATE THE READING.
C
      WRITE(6,10) 'INITIAL STATE','NODAL COORDINATES','NODE',
     .          'DIRECTION-1','DIRECTION-2','DIRECTION-3'
   10 FORMAT(///T30,19('#')/T30,'#',T33,A,T47,'#'/T30,19('#')//T10,A/
     .     T10,18('_')/T10,A,T22,A,T35,A,T48,A)
C
      READ(5,*) J,XX1,XX2,XX3,VSS1,VSS2,VSS3,VTT1,VTT2,VTT3
   30 IF(J .NE. 0) THEN
        X(1,J) = XX1
        X(2,J) = XX2
        X(3,J) = XX3
        VS(1,J) = VSS1
        VS(2,J) = VSS2
        VS(3,J) = VSS3
        VT(1,J) = VTT1
        VT(2,J) = VTT2
        VT(3,J) = VTT3
        CALL UNIVEC(VR,VS,VT,J)
        READ(5,*) J,XX1,XX2,XX3,VSS1,VSS2,VSS3,VTT1,VTT2,VTT3
        GO TO 30
      ENDIF
      DO 50 J = 1,NJ
        WRITE(6,40) J,X(1,J),X(2,J),X(3,J)
   40   FORMAT(T13,I3,T20,F8.2,T33,F8.2,T46,F8.2)
   50 CONTINUE
C
C--- ECHO THE UNIT VECTORS AT EACH NODE.
C
```

```
      WRITE(6,20) 'UNIT NODAL VECTORS','NODE','VR1','VR2','VR3','VS1',
     .            'VS2','VS3','VT1','VT2','VT3'
   20 FORMAT(//T66,A/T5,A,T19,A,T34,A,T49,A,T64,A,T79,A,T94,A,T109,A,
     .        T124,A,T139,A)
C
      DO 70 J = 1,NJ
        WRITE(6,60) J,VR(1,J),VR(2,J),VR(3,J),VS(1,J),VS(2,J),VS(3,J),
     .              VT(1,J),VT(2,J),VT(3,J)
   60   FORMAT(T5,I4,T12,F12.6,T27,F12.6,T42,F12.6,T57,F12.6,
     .      T72,F12.6,T87,F12.6,T102,F12.6,T117,F12.6,T132,F12.6)
   70 CONTINUE
C
C--- READ AND ECHO ELEMENT PROPERTIES... SC = SHEAR CORRECTION FACTOR.
C
      SC = 5.D0 / 6.D0
      WRITE(6,80) 'ELEMENT PROPERTIES','ELEMENT','WIDTH','DEPTH',
     .           'MOD. OF EL.','SHEAR MOD.'
   80 FORMAT(///T66,A/T20,A,T40,A,T60,A,T80,A,T100,A)
      READ(5,*) I,WIDTHI,DEPTHI,EMODI,GMODI
   90 IF(I .NE. 0) THEN
        WRITE(6,100) I,WIDTHI,DEPTHI,EMODI,GMODI
  100   FORMAT(T20,I4,T36,F10.3,T56,F10.3,T80,E9.2,T100,E9.2)
        WIDTH(I) = WIDTHI
        DEPTH(I) = DEPTHI
        EMOD(I) = EMODI
        GMOD(I) = GMODI * SC
        READ(5,*) I,WIDTHI,DEPTHI,EMODI,GMODI
        GO TO 90
      ENDIF
C
      RETURN
      END
C**********************************************************************
C                        UNIVEC
C**********************************************************************
C
C     IT COMPUTES THE CROSS PRODUCT OF THE NODAL UNIT VECTORS IN LOCAL
C     S AND T DIRECTIONS TO COMPUTE THE UNIT VECTOR  IN LOCAL R
C     DIRECTIONS.
C
      SUBROUTINE UNIVEC(VR,VS,VT,J)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION VR(3,*),VS(3,*),VT(3,*)
C
      VR(1,J) = VS(2,J) * VT(3,J) - VS(2,J) * VT(2,J)
      VR(2,J) = VS(3,J) * VT(1,J) - VS(1,J) * VT(3,J)
      VR(3,J) = VS(1,J) * VT(2,J) - VS(2,J) * VT(1,J)
C
      RETURN
      END
C**********************************************************************
C*                       JLOAD
C**********************************************************************
C
C     IT READS AND ECHOES THE SYSTEM JOINT LOADS AND  STORES THEM IN
C     A VECTOR FORM. IT ALSO CALCULATES THE CONSTANT LOAD VECTOR QBAR
C     FOR NONLINEAR ANALYSIS.
C
      SUBROUTINE JLOAD(Q,QBAR,JCODE,QIMAX,NEQ)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION JCODE(6,*),Q(*),QBAR(*)
      READ(5,*) JNUM,JDIR,FOR
      IF(JNUM .NE. 0) THEN
```

```
        WRITE(6,10) 'APPLIED NODAL LOADS','NODE NO.',
     .            'DIRECTION', 'FORCE'
  10    FORMAT(/T5,A,
     .      /T5,24('_')/T5,A,T15,A,T28,A)
      ENDIF
  20 IF(JNUM .NE. 0) THEN
        WRITE(6,30) JNUM,JDIR,FOR
  30    FORMAT(T8,I3,T19,I1,T26,F10.4)
        K = JCODE(JDIR,JNUM)
        IF(K .NE. 0) THEN
          Q(K) = FOR
        ENDIF
        READ(5,*) JNUM,JDIR,FOR
        GO TO 20
      ENDIF
C
C--- COMPUTATION OF QBAR VECTOR.
C
      DO 40 I = 1,NEQ
        QBAR(I) = Q(I)/QIMAX
        Q(I) = 0.D0
 40   CONTINUE
C
      RETURN
      END
C**********************************************************************
C*                         DLOAD
C**********************************************************************
C
C   IT READS AND ECHOES THE DISTRIBUTED LOADS ON THE MEMBERS,
C   DISCRETIZES THEM AND STORES THEM IN A VECTOR FORM. IT ALSO
C   CALCULATES THE CONSTANT LOAD VECTOR FOR NONLINEAR ANALYSIS.
C
      SUBROUTINE DLOAD(Q,QBAR,JCODE,QIMAX,NEQ,MINC,X)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION JCODE(6,*),Q(*),QBAR(*),MINC(3,*),X(3,*)
C
      READ(5,*) IE,PA1,PA2,PA3,PB1,PB2,PB3,RADIUS
C
      IF(IE .NE. 0) THEN
        WRITE(6,10) 'APPLIED DISTRIBUTED LOADS','ELEM NO.','*A - END*',
     .         '*B - END*','DIRECTION 1','DIRECTON 2','DIRECTION 3',
     .          'DIRECTION 1','DIRECTION 2','DIRECTION 3'
  10    FORMAT(/T60,A/T60,30('_')/T5,A,T46,A,T101,A//T23,A,T43,A,T63,A,
     .     T88,A,T108,A,T128,A)
      ENDIF
C
 20   IF(IE .NE. 0) THEN
        KG1 = MINC(1,IE)
        KG2 = MINC(2,IE)
        KG3 = MINC(3,IE)
        ELENG = DSQRT ( (X(1,KG2) - X(1,KG1))**2 +
     .       (X(2,KG2) - X(2,KG1))**2 + (X(3,KG2) - X(3,KG1))**2 )
        IF(RADIUS .NE. 0.D0) THEN
          ELENG = RADIUS * 2.D0 * DASIN (ELENG / 2.D0 / RADIUS)
        ENDIF
C
        K1 = JCODE(1,KG1)
        K2 = JCODE(1,KG2)
        K3 = JCODE(1,KG3)
C
        IF(K1 .NE. 0) THEN
          Q(K1) = Q(K1) + PA1 * ELENG / 6.D0
        ENDIF
C
        IF(K2 .NE. 0) THEN
          Q(K2) = Q(K2) + PB1 * ELENG / 6.D0
        ENDIF
C
        IF(K3 .NE. 0) THEN
          Q(K3) = Q(K3) + ELENG * (PA1 + PB1) / 3.D0
        ENDIF
C
        K1 = JCODE(2,KG1)
        K2 = JCODE(2,KG2)
        K3 = JCODE(2,KG3)
C
        IF(K1 .NE. 0) THEN
          Q(K1) = Q(K1) + PA2 * ELENG / 6.D0
        ENDIF
C
        IF(K2 .NE. 0) THEN
          Q(K2) = Q(K2) + PB2 * ELENG / 6.D0
        ENDIF
C
        IF(K3 .NE. 0) THEN
          Q(K3) = Q(K3) + ELENG * (PA2 + PB2) / 3.D0
        ENDIF
C
        K1 = JCODE(3,KG1)
        K2 = JCODE(3,KG2)
        K3 = JCODE(3,KG3)
C
        IF(K1 .NE. 0) THEN
          Q(K1) = Q(K1) + PA3 * ELENG / 6.D0
        ENDIF
C
        IF(K2 .NE. 0) THEN
          Q(K2) = Q(K2) + PB3 * ELENG / 6.D0
        ENDIF
C
        IF(K3 .NE. 0) THEN
          Q(K3) = Q(K3) + ELENG * (PA3 + PB3) / 3.D0
        ENDIF
C
        WRITE(6,30) IE,PA1,PA2,PA3,PB1,PB2,PB3
 30      FORMAT(T8,I3,T20,E16.10,T40,E16.10,T60,E16.10,T85,E16.10,T105,
     .       E16.10,T125,E16.10)
C
        READ(5,*) IE,PA1,PA2,PA3,PB1,PB2,PB3,RADIUS
        GO TO 20
      ENDIF
C
C---  COMPUTATION OF QBAR VECTOR.
C
      DO 40 I = 1,NEQ
        QBAR(I) = Q(I)/QIMAX + QBAR(I)
        Q(I) = 0.D0
 40   CONTINUE
C
      RETURN
      END
C*****************************************************************
C                      STIFF
C*****************************************************************
C
```

```
C   IT IS A CONTROL MODULE FOR DETERMINING AND UPDATING THE TANGENT
C   STIFFNESS VECTOR SSS FOR THE CURRENT CONFIGURATION. IT INITIALIZES
C   SSS AND CALLS 'INTPNT' AND 'ASSEMS'.
C
      SUBROUTINE STIFF (NE,MINC,R,S,T,WTR,WTS,WTT,TYPE,JCODE,SSS,MAXA,
     .            LSS,B,EMOD,GMOD,H,DH,VR,VS,VT,DEPTH,WIDTH,G,
     .            XJINV,X,VVR,VVS,VVT,BNL,STRESS,DS1,DS2,DS3)
      IMPLICIT REAL*8(A-H,O-Z)
      CHARACTER*(*) TYPE*6
      DIMENSION MINC(3,*),R(*),S(*),T(*),WTR(*),WTS(*),WTT(*),MAXA(*),
     .      JCODE(6,*),SSS(*),B(3,3,*),EMOD(*),GMOD(*),H(*),DH(*),
     .      VR(3,*),VS(3,*),VT(3,*),DEPTH(*),WIDTH(*),G(3,3,*),
     .      XJINV(3,*),X(3,*),VVR(*),VVS(*),VVT(*),
     .      STRESS(*),DS1(2,2,*),DS2(2,2,*),DS3(2,2,*),
     .      BNL(2,2,2,3,9,*)
C
      REWIND (23)
      DO 10 I = 1,LSS
        SSS(I) = 0.D0
   10 CONTINUE
C
      NR = 2
      NS = 2
      NT = 2
      CALL INTPNT(R,S,T,NR,NS,NT,WTR,WTS,WTT)
C
      DO 50 N = 1,NE
      IF (TYPE .EQ. 'NONLIN') THEN
        DO 46 IRR = 1,2
          DO 44 ISS = 1,2
            DO 42 ITT = 1,2
              READ (23) ST1,ST2,ST3
                DS1(IRR,ISS,ITT) = ST1
                DS2(IRR,ISS,ITT) = ST2
                DS3(IRR,ISS,ITT) = ST3
 42         CONTINUE
 44       CONTINUE
 46     CONTINUE
      ENDIF
        DO 40 M = 1,3
          DO 30 L = 1,3
           IM = MINC(M,N)
           IL = MINC(L,N)
      CALL ASSEMS (M,L,N,IM,IL,JCODE,MAXA,SSS,TYPE,B,EMOD,GMOD,
     .            H,DH,VR,VS,VT,R,S,T,WTR,WTS,WTT,DEPTH,WIDTH,G,
     .            XJINV,X,MINC,LSS,VVR,VVS,VVT,BNL,STRESS,DS1,
     .            DS2,DS3)
   30     CONTINUE
   40   CONTINUE
   50 CONTINUE
C
      RETURN
      END
C*********************************************************************
C*                     SOLVE                                         *
C*********************************************************************
      SUBROUTINE SOLVE(SS,Q,MAXA,NEQ,LC)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION SS(*),Q(*),MAXA(*)
C
C   SOLVE DETERMINES THE SOLUTION TO THE SYSTEM EQUATIONS BY COMPACT
C   GAUSSIAN ELIMINATION (HOLZER, PP. 290, 296, 307) BASED ON THE
C   SUBROUTINE COLSOL (BATHE P. 721) AND THE MODIFICATION BY MICHAEL
```

```
C     BUTLER (MS 1984): FOR THE FIRST LOAD CONDITION, LC = 1, CALL
C     FACTOR,REDUCE,AND BACSUB; FOR SUBSEQUENT LOAD CONDITIONS, LC > 1,
C     CALL REDUCE AND BACSUB.
C
      IF (LC.EQ.1) THEN
         CALL FACTOR(SS,MAXA,NEQ)
      END IF
         CALL REDUCE(SS,Q,MAXA,NEQ)
         CALL BACSUB(SS,Q,MAXA,NEQ)
C
      RETURN
      END
C*********************************************************************
C*                     FACTOR                         *
C*********************************************************************
C
C     FACTOR PERFORMS THE LDU FACTORIZATION OF THE STIFFNESS MATRIX.
C
      SUBROUTINE FACTOR(SS,MAXA,NEQ)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION SS(*),MAXA(*)
C
      DO 80 N=1,NEQ
       KN=MAXA(N)
       KL=KN+1
       KU=MAXA(N+1)-1
       KH=KU-KL
       IF(KH) 70,50,10
   10  K=N-KH
       IC=0
       KLT=KU
       DO 40 J=1,KH
        IC=IC+1
        KLT=KLT-1
        KI=MAXA(K)
        ND=MAXA(K+1)-KI-1
        IF(ND) 40,40,20
   20   KK=MIN0(IC,ND)
        C=0.D0
        DO 30 L=1,KK
   30   C=C+SS(KI+L)*SS(KLT+L)
        SS(KLT)=SS(KLT)-C
   40  K=K+1
   50  K=N
       B=0.D0
       DO 60 KK=KL,KU
        K=K-1
        KI=MAXA(K)
        C=SS(KK)/SS(KI)
        B=B+C*SS(KK)
   60  SS(KK)=C
       SS(KN)=SS(KN)-B
C
C     STOP EXECUTION IF A ZERO PIVOT IS DETECTED
C
   70 IF(SS(KN).EQ.0.D0) THEN
        PRINT 75,N,SS(KN)
   75   FORMAT('-STIFFNESS MATRIX IS NOT POSITIVE DEFINITE'/'0PIVOT IS
     .  ZERO FOR D.O.F. ',I4/'0PIVOT = ',E15.8)
        STOP
       END IF
C
   80 CONTINUE
```

```
      RETURN
      END
C
C**********************************************************************
C*                        REDUCE                                      *
C**********************************************************************
C
C     REDUCE REDUCES THE RIGHT-SIDE LOAD VECTOR.
C
      SUBROUTINE REDUCE(SS,Q,MAXA,NEQ)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION SS(*),Q(*),MAXA(*)
C
      DO 20 N=1,NEQ
        KL=MAXA(N)+1
        KU=MAXA(N+1)-1
        KH=KU-KL
        IF(KH.GE.0) THEN
          K=N
          C=0.D0
          DO 10 KK=KL,KU
            K=K-1
            C=C+SS(KK)*Q(K)
   10     CONTINUE
          Q(N)=Q(N)-C
        END IF
   20 CONTINUE
      RETURN
      END
C
C**********************************************************************
C*                        BACSUB                                      *
C**********************************************************************
C
C     BACSUB PERFORMS BACK-SUBSTITUTION TO OBTAIN THE SOLUTION.
C
      SUBROUTINE BACSUB(SS,Q,MAXA,NEQ)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION SS(*),Q(*),MAXA(*)
C
      DO 10 N=1,NEQ
        K=MAXA(N)
        Q(N)=Q(N)/SS(K)
   10 CONTINUE
      IF(NEQ.EQ.1) RETURN
      N=NEQ
      DO 30 L=2,NEQ
        KL=MAXA(N)+1
        KU=MAXA(N+1)-1
        KH=KU-KL
        IF(KH.GE.0)THEN
          K=N
          DO 20 KK=KL,KU
            K=K-1
            Q(K)=Q(K)-SS(KK)*Q(N)
   20     CONTINUE
        END IF
        N=N-1
   30 CONTINUE
      RETURN
      END
C**********************************************************************
C                         ASSEMS
```

```
C***********************************************************************
C
C    IT CALCULATES THE LINEAR AND NONLINEAR STIFFNESS CONTRIBUTIONS OF
C    THE ELEMENTS AND ASSEMBLES THEM INTO A SYSTEM STIFFNESS VECTOR.
C    IT CALLS 'LSTIF' AND 'NSTIF'.
C
      SUBROUTINE ASSEMS (M,L,N,IM,IL,JCODE,MAXA,SSS,TYPE,B,EMOD,GMOD,
     .           H,DH,VR,VS,VT,R,S,T,WTR,WTS,WTT,DEPTH,WIDTH,G,
     .           XJINV,X,MINC,LSS,VVR,VVS,VVT,BNL,STRESS,DS1,
     .           DS2,DS3)
      IMPLICIT REAL*8(A-H,O-Z)
      CHARACTER*(*) TYPE*6
      DIMENSION R(*),S(*),T(*),WTR(*),WTS(*),WTT(*),B(3,3,*),MAXA(*),
     .      EMOD(*),GMOD(*),H(*),DH(*),VR(3,*),VS(3,*),VT(3,*),
     .      SSS(*),DEPTH(*),WIDTH(*),G(3,3,*),XJINV(3,*),X(3,*),
     .      JCODE(6,*),MINC(3,*),VVR(*),VVS(*),VVT(*),
     .      BNL(2,2,3,9,*),STRESS(*),DS1(2,2,*),
     .      DS2(2,2,*),DS3(2,2,*)
C
C--- GET THE ELEMENT DEGREE OF FREEDOM WHICH CONTRIBUTES TO THE SYSTEM
C    STIFFNESS.
C
      DO 20 JE = 1,6
        J = JCODE(JE,IM)
        IF(J .NE. 0) THEN
          DO 10 IE = 1,6
            L1 = JCODE(IE,IL)
              IF(L1 .NE. 0 .AND. L1 .LE. J) THEN
                K = MAXA(J) + J - L1
C
C--- CALCULATE THE LINEAR STIFFNESS CONTRIBUTION.
C
                CALL LSTIF(R,S,T,WTR,WTS,WTT,IM,IL,M,L,N,IE,JE,B,
     .              EMOD,GMOD,H,DH,VR,VS,VT,DEPTH,WIDTH,G,XJINV,X,
     .              MINC,SL,VVR,VVS,VVT)
C
C--- GET THE NONLINEAR STIFFNESS CONTRIBUTION (FOR NONLINEAR ANALYSIS
C    ONLY).
C
                IF(TYPE .EQ. 'NONLIN') THEN
                CALL NSTIF(R,S,T,WTR,WTS,WTT,IM,IL,M,L,N,IE,JE,BNL,
     .            STRESS,VVR,VVS,VVT,
     .            H,DH,VR,VS,VT,DEPTH,WIDTH,G,XJINV,X,MINC,SN,
     .            DS1,DS2,DS3)
                ENDIF
C
C--- ASSEMBLE THE SYSTEM STIFFNESS VECTOR.
C
                SSS(K) = SSS(K) + SL + SN
              ENDIF
   10     CONTINUE
        ENDIF
   20 CONTINUE
C
      RETURN
      END
C***********************************************************************
C                       LSTIF
C***********************************************************************
C
C    IT CALCULATES THE LINEAR GLOBAL STIFFNESS CONTRIBUTION FOR EACH
C    ELEMENT FOR ITS UNCONSTRAINED DEGREE OF FREEDOM. IT CALLS
```

```
C     'INTER', 'DIVECT', 'DETJAC', 'INVJAC', AND 'NODALB'.
C
      SUBROUTINE LSTIF(R,S,T,WTR,WTS,WTT,IM,IL,M,L,N,IE,JE,B,EMOD,GMOD,
     .           H,DH,VR,VS,VT,DEPTH,WIDTH,G,XJINV,X,MINC,SL,
     .           VVR,VVS,VVT)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION R(*),S(*),T(*),WTR(*),WTS(*),WTT(*),EMOD(*),
     .      GMOD(*),H(*),DH(*),VR(3,*),VS(3,*),VT(3,*),DEPTH(*),
     .      WIDTH(*),G(3,3,*),XJINV(3,*),X(3,*),B(3,3,*),MINC(3,*),
     .      VVR(*),VVS(*),VVT(*)
C
      SL = 0.D0
      DO 40 IR = 1,2
        DO 30 IS = 1,2
          DO 20 IT = 1,2
            RR = R(IR)
            WR = WTR(IR)
            SS = S(IS)
            WS = WTS(IS)
            TT = T(IT)
            WT = WTT(IT)
            CALL INTER(H,DH,RR)
            CALL DIVECT(VR,VS,VT,VVR,VVS,VVT,H,MINC,N)
           CALL DETJAC(H,DH,RR,SS,TT,DEPTH,WIDTH,VS,VT,XJAC,X,N,MINC)
          CALL INVJAC(H,DH,RR,DEPTH,WIDTH,VS,VT,XJAC,X,N,XJINV,
     .           SS,TT,MINC)
C
            DO 10 LL = 1,3
              CALL NODALB(H,DH,RR,SS,TT,XJINV,DEPTH,WIDTH,XJAC,X,BB,
     .           VR,VS,VT,G,N,IM,M,LL,JE,VVR,VVS,VVT)
              B(M,LL,JE) = BB
              CALL NODALB(H,DH,RR,SS,TT,XJINV,DEPTH,WIDTH,XJAC,X,BB,
     .           VR,VS,VT,G,N,IL,L,LL,IE,VVR,VVS,VVT)
              B(L,LL,IE) = BB
 10          CONTINUE
          SSL = (EMOD(N) * B(M,1,JE) * B(L,1,IE) + GMOD(N) *
     .        (B(M,2,JE) * B(L,2,IE) + B(M,3,JE) * B(L,3,IE))) *
     .         XJAC * WR * WS * WT
          SL = SL + SSL
C
 20       CONTINUE
 30     CONTINUE
 40   CONTINUE
C
      RETURN
      END
C***********************************************************************
C                        NSTIF
C***********************************************************************
C
C     IT CALCULATES THE NONLINEAR GLOBAL STIFFNESS CONTRIBUTION FOR
C     EACH ELEMENT FOR ITS UNCONSTRAINED DEGREE OF FREEDOM. IT CALLS
C     'INTER', 'DIVECT', 'DETJAC', 'INVJAC', AND 'NOLINB'.
C
      SUBROUTINE NSTIF(R,S,T,WTR,WTS,WTT,IM,IL,M,L,N,IE,JE,BNL,
     .           STRESS,VVR,VVS,VVT,
     .           H,DH,VR,VS,VT,DEPTH,WIDTH,G,XJINV,X,MINC,SN,
     .           DS1,DS2,DS3)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION R(*),S(*),T(*),WTR(*),WTS(*),WTT(*),
     .      H(*),DH(*),VR(3,*),VS(3,*),VT(3,*),DEPTH(*),
     .      WIDTH(*),G(3,3,*),XJINV(3,*),X(3,*),BNL(2,2,2,3,9,*),
     .      VVR(*),VVS(*),VVT(*),MINC(3,*),DS1(2,2,*),DS2(2,2,*),
     .      STRESS(*),DS3(2,2,*)
C
      SN = 0.D0
      DO 400 IR = 1,2
        DO 390 IS = 1,2
          DO 380 IT = 1,2
            RR = R(IR)
            WR = WTR(IR)
C
C--- CALCULATE THE INTERPOLATION FUNCTIONS AT THE GIVEN INTERPOLATION
C    POINTS.
C
            CALL INTER(H,DH,RR)
            SS = S(IS)
            WS = WTS(IS)
            TT = T(IT)
            WT = WTT(IT)
C
C--- CALCULATE THE UNIT VECTORS AT THE INTERPOLATION POINTS.
C
            CALL DIVECT(VR,VS,VT,VVR,VVS,VVT,H,MINC,N)
C
C--- CALCULATE THE DETERMINANT OF THE JACOBIAN MATRIX.
C
            CALL DETJAC(H,DH,RR,SS,TT,DEPTH,WIDTH,VS,VT,
     .             XJAC,X,N,MINC)
C
C--- CALCULATE THE INVERSE OF THE JACOBIAN MATRIX.
C
            CALL INVJAC(H,DH,RR,DEPTH,WIDTH,VS,VT,XJAC,X,N,XJINV,
     .             SS,TT,MINC)
C
C---   CALCULATE THE NONLINEAR STRAIN-DISPLACEMENT MATRIX.
C
            IF(N .NE. NP) THEN
              CALL NOLINB(IR,IS,IT,RR,SS,TT,XJINV,DEPTH,WIDTH,H,DH,
     .             XJAC,X,VR,VS,VT,G,N,VVR,VVS,VVT,MINC,BNL)
            ENDIF
C
            STRESS(1) = DS1(IR,IS,IT)
            STRESS(2) = DS2(IR,IS,IT)
            STRESS(3) = DS3(IR,IS,IT)
C
C--- ACCUMULATE THE NONLINEAR STIFFNESS CONTRIBUTIONS FOR EACH GLOBAL
C    DEGREE OF FREEDOM.
C
            IF(IE .EQ. 1) THEN
              IF(JE .EQ. 1) THEN
              SN = SN + ( BNL(IR,IS,IT,M,1,1) *  ( STRESS(1) *
     .          BNL(IR,IS,IT,L,1,1) + STRESS(2) *
     .          BNL(IR,IS,IT,L,2,1) + STRESS(3) *
     .          BNL(IR,IS,IT,L,3,1) ) + BNL(IR,IS,IT,M,2,1) *
     .          STRESS(2) * BNL(IR,IS,IT,L,1,1) +
     .          BNL(IR,IS,IT,M,3,1) * STRESS(3) *
     .          BNL(IR,IS,IT,M,1,1) ) * XJAC * WR * WS * WT
C
              ELSEIF(JE .EQ. 5) THEN
              SN = SN + ( BNL(IR,IS,IT,M,1,1) * ( STRESS(1) *
     .          BNL(IR,IS,IT,L,1,5) + STRESS(2) *
     .          BNL(IR,IS,IT,L,2,5) + STRESS(3) *
     .          BNL(IR,IS,IT,L,3,5) ) +
     .          BNL(IR,IS,IT,M,2,1) * STRESS(2) *
     .          BNL(IR,IS,IT,L,1,5) +
     .      BNL(IR,IS,IT,M,3,1) * STRESS(3) *
     .      BNL(IR,IS,IT,M,1,5) ) * XJAC * WR * WS * WT
C
       ELSEIF(JE .EQ. 6) THEN
       SN = SN + ( BNL(IR,IS,IT,M,1,1) *
     .      ( STRESS(1) * BNL(IR,IS,IT,L,1,6) +
     .      STRESS(2) * BNL(IR,IS,IT,L,2,6) + STRESS(3) *
     .      BNL(IR,IS,IT,L,3,6) ) +
     .      BNL(IR,IS,IT,M,2,1) * STRESS(2) *
     .      BNL(IR,IS,IT,L,1,6) + BNL(IR,IS,IT,M,3,1) *
     .      STRESS(3) * BNL(IR,IS,IT,M,1,6) ) * XJAC
     .      * WR * WS * WT
C
       ENDIF
      ELSEIF(IE .EQ. 2) THEN
       IF(JE .EQ. 2) THEN
         SN = SN + ( BNL(IR,IS,IT,M,4,2) * ( STRESS(1) *
     .        BNL(IR,IS,IT,L,4,2) + STRESS(2) *
     .        BNL(IR,IS,IT,L,5,2) + STRESS(3) *
     .        BNL(IR,IS,IT,L,6,2) ) + BNL(IR,IS,IT,M,5,2) *
     .        STRESS(2) * BNL(IR,IS,IT,L,4,2) +
     .        BNL(IR,IS,IT,M,6,2) * STRESS(3) *
     .        BNL(IR,IS,IT,M,4,2) ) * XJAC * WR * WS * WT
C
       ELSEIF(JE .EQ. 4) THEN
         SN = SN + ( BNL(IR,IS,IT,M,4,2) * ( STRESS(1) *
     .        BNL(IR,IS,IT,L,4,4) + STRESS(2)
     .       * BNL(IR,IS,IT,L,5,4) + STRESS(3) *
     .        BNL(IR,IS,IT,L,6,4) ) + BNL(IR,IS,IT,M,5,2)
     .       * STRESS(2) * BNL(IR,IS,IT,L,4,4) +
     .        BNL(IR,IS,IT,M,6,2) * STRESS(3) *
     .        BNL(IR,IS,IT,M,4,4) ) * XJAC * WR * WS * WT
C
       ELSEIF(JE .EQ. 6) THEN
         SN = SN + ( BNL(IR,IS,IT,M,4,2) * ( STRESS(1) *
     .        BNL(IR,IS,IT,L,4,6) + STRESS(2)
     .       * BNL(IR,IS,IT,L,5,6) + STRESS(3) *
     .        BNL(IR,IS,IT,L,6,6) ) + BNL(IR,IS,IT,M,5,2) *
     .        STRESS(2) * BNL(IR,IS,IT,L,4,6) +
     .        BNL(IR,IS,IT,M,6,2) * STRESS(3) *
     .        BNL(IR,IS,IT,M,4,6) ) * XJAC * WR * WS * WT
       ENDIF
C
      ELSEIF(IE .EQ. 3) THEN
       IF(JE .EQ. 3) THEN
         SN = SN + ( BNL(IR,IS,IT,M,7,3) *
     .        ( STRESS(1) * BNL(IR,IS,IT,L,7,3) +
     .        STRESS(2) * BNL(IR,IS,IT,L,8,3) + STRESS(3) *
     .        BNL(IR,IS,IT,L,9,3) ) + BNL(IR,IS,IT,M,8,3) *
     .        STRESS(2) * BNL(IR,IS,IT,L,7,3) +
     .        BNL(IR,IS,IT,M,9,3) * STRESS(3) *
     .        BNL(IR,IS,IT,M,7,3) ) * XJAC * WR * WS * WT
C
       ELSEIF(JE .EQ. 4) THEN
         SN = SN + ( BNL(IR,IS,IT,M,7,3) * ( STRESS(1) *
     .        BNL(IR,IS,IT,L,7,4) + STRESS(2) *
     .        BNL(IR,IS,IT,L,8,4) + STRESS(3) *
     .        BNL(IR,IS,IT,L,9,4) ) + BNL(IR,IS,IT,M,8,3) *
     .        STRESS(2) * BNL(IR,IS,IT,L,7,4) +
     .        BNL(IR,IS,IT,M,9,3) * STRESS(3) *
     .        BNL(IR,IS,IT,M,7,4) ) * XJAC * WR * WS * WT
C
       ELSEIF(JE .EQ. 5) THEN
```

```
            SN = SN + ( BNL(IR,IS,IT,M,7,3) * ( STRESS(1) *
     .          BNL(IR,IS,IT,L,7,5) + STRESS(2) *
     .          BNL(IR,IS,IT,L,8,5) + STRESS(3) *
     .          BNL(IR,IS,IT,L,9,5) ) + BNL(IR,IS,IT,M,8,3)
     .          * STRESS(2) * BNL(IR,IS,IT,L,7,5) +
     .          BNL(IR,IS,IT,M,9,3) * STRESS(3) *
     .          BNL(IR,IS,IT,M,7,5) ) * XJAC * WR * WS * WT
          ENDIF
C
        ELSEIF(IE .EQ. 4) THEN
          IF(JE .EQ. 2) THEN
            SN = SN + ( BNL(IR,IS,IT,M,4,4) * ( STRESS(1) *
     .          BNL(IR,IS,IT,L,4,2) + STRESS(2) *
     .          BNL(IR,IS,IT,L,5,2) + STRESS(3) *
     .          BNL(IR,IS,IT,L,6,2) ) + BNL(IR,IS,IT,M,5,4)
     .          * STRESS(2) * BNL(IR,IS,IT,L,4,2) +
     .          BNL(IR,IS,IT,M,6,4) * STRESS(3) *
     .          BNL(IR,IS,IT,M,4,2) ) * XJAC * WR * WS * WT
C
          ELSEIF(JE .EQ. 3) THEN
            SN = SN + ( BNL(IR,IS,IT,M,7,4) * ( STRESS(1) *
     .          BNL(IR,IS,IT,L,7,3) + STRESS(2) *
     .          BNL(IR,IS,IT,L,8,3) + STRESS(3) *
     .          BNL(IR,IS,IT,L,9,3) ) + BNL(IR,IS,IT,M,8,4) *
     .          STRESS(2) * BNL(IR,IS,IT,L,8,3) +
     .          BNL(IR,IS,IT,M,9,4) * STRESS(3) *
     .          BNL(IR,IS,IT,M,7,3) ) * XJAC * WR * WS * WT
C
          ELSEIF(JE .EQ. 4) THEN
            SN = SN + ( BNL(IR,IS,IT,M,4,4) * ( STRESS(1) *
     .          BNL(IR,IS,IT,L,4,4) + STRESS(2) *
     .          BNL(IR,IS,IT,L,5,4) + STRESS(3) *
     .          BNL(IR,IS,IT,L,6,4) ) + BNL(IR,IS,IT,M,5,4) *
     .          STRESS(2) * BNL(IR,IS,IT,L,4,4) +
     .          BNL(IR,IS,IT,M,6,4) * STRESS(3) *
     .          BNL(IR,IS,IT,M,4,4) + BNL(IR,IS,IT,M,7,4) *
     .          ( STRESS(1) * BNL(IR,IS,IT,L,7,4) +
     .          STRESS(2) * BNL(IR,IS,IT,L,8,4) + STRESS(3) *
     .          BNL(IR,IS,IT,L,9,4) ) + BNL(IR,IS,IT,M,8,4) *
     .          STRESS(2) * BNL(IR,IS,IT,L,7,4) +
     .          BNL(IR,IS,IT,M,9,4) * STRESS(3) *
     .          BNL(IR,IS,IT,M,7,4) ) * XJAC * WR * WS * WT
C
          ELSEIF(JE .EQ. 5) THEN
            SN = SN + ( BNL(IR,IS,IT,M,7,4) * ( STRESS(1) *
     .          BNL(IR,IS,IT,L,7,5) + STRESS(2) *
     .          BNL(IR,IS,IT,L,8,5) + STRESS(3) *
     .          BNL(IR,IS,IT,L,9,5) ) + BNL(IR,IS,IT,M,8,4)
     .         * STRESS(2) * BNL(IR,IS,IT,L,7,5) +
     .          BNL(IR,IS,IT,M,9,4) * STRESS(3) *
     .          BNL(IR,IS,IT,M,7,5) ) * XJAC * WR * WS * WT
C
          ELSEIF(JE .EQ. 6) THEN
            SN = SN + ( BNL(IR,IS,IT,M,4,4) * ( STRESS(1) *
     .          BNL(IR,IS,IT,L,4,6) + STRESS(2) *
     .          BNL(IR,IS,IT,L,5,6) + STRESS(3) *
     .          BNL(IR,IS,IT,L,6,6) ) + BNL(IR,IS,IT,M,5,4) *
     .          STRESS(2) * BNL(IR,IS,IT,L,4,6) +
     .          BNL(IR,IS,IT,M,6,4) * STRESS(3) *
     .          BNL(IR,IS,IT,M,4,6) ) * XJAC * WR * WS * WT
C
        ELSEIF(IE .EQ. 5) THEN
          IF(JE .EQ. 1) THEN
```

```
              SN = SN + ( BNL(IR,IS,IT,M,1,5) * ( STRESS(1) *
     .           BNL(IR,IS,IT,L,1,1) + STRESS(2) *
     .           BNL(IR,IS,IT,L,2,1) + STRESS(3) *
     .           BNL(IR,IS,IT,L,3,1) ) + BNL(IR,IS,IT,M,2,5) *
     .           STRESS(2) * BNL(IR,IS,IT,L,1,1) +
     .           BNL(IR,IS,IT,M,3,5) * STRESS(3) *
     .           BNL(IR,IS,IT,M,1,1) ) * XJAC * WR * WS * WT
C
              ELSEIF(JE .EQ. 3) THEN
                SN = SN + ( BNL(IR,IS,IT,M,7,5) * ( STRESS(1) *
     .             BNL(IR,IS,IT,L,7,3) + STRESS(2) *
     .             BNL(IR,IS,IT,L,8,3) + STRESS(3) *
     .             BNL(IR,IS,IT,L,9,9) ) + BNL(IR,IS,IT,M,8,5) *
     .             STRESS(2) * BNL(IR,IS,IT,L,7,3) +
     .             BNL(IR,IS,IT,M,9,5) * STRESS(3) *
     .             BNL(IR,IS,IT,M,7,3) ) * XJAC * WR * WS * WT
              ENDIF
C
              ELSEIF(JE .EQ. 4) THEN
                SN = SN + ( BNL(IR,IS,IT,M,7,5) * ( STRESS(1) *
     .             BNL(IR,IS,IT,L,7,4) + STRESS(2) *
     .             BNL(IR,IS,IT,L,8,4) + STRESS(3) *
     .             BNL(IR,IS,IT,L,9,4) ) + BNL(IR,IS,IT,M,8,5) *
     .             STRESS(2) * BNL(IR,IS,IT,L,7,4) +
     .             BNL(IR,IS,IT,M,9,5) * STRESS(3) *
     .             BNL(IR,IS,IT,M,7,4) ) * XJAC * WR * WS * WT
C
              ELSEIF(JE .EQ. 5) THEN
                SN = SN + ( BNL(IR,IS,IT,M,1,5) * ( STRESS(1) *
     .             BNL(IR,IS,IT,L,1,5) + STRESS(2) *
     .             BNL(IR,IS,IT,L,2,5) + STRESS(3) *
     .             BNL(IR,IS,IT,L,3,5) ) + BNL(IR,IS,IT,M,2,5) *
     .             STRESS(2) * BNL(IR,IS,IT,L,1,5) +
     .             BNL(IR,IS,IT,M,3,5) * STRESS(3) *
     .             BNL(IR,IS,IT,M,1,5) + BNL(IR,IS,IT,M,7,5) *
     .             ( STRESS(1) * BNL(IR,IS,IT,L,7,5) +
     .             STRESS(2) * BNL(IR,IS,IT,L,8,5) + STRESS(3) *
     .             BNL(IR,IS,IT,L,9,5) ) + BNL(IR,IS,IT,M,8,5) *
     .             STRESS(2) * BNL(IR,IS,IT,L,7,5) +
     .             BNL(IR,IS,IT,M,9,5) * STRESS(3) *
     .             BNL(IR,IS,IT,M,7,5) ) * XJAC * WR * WS * WT
C
              ELSEIF(JE .EQ. 6) THEN
              SN = SN + ( BNL(IR,IS,IT,M,1,5) * ( STRESS(1) *
     .           BNL(IR,IS,IT,L,1,6) + STRESS(2) *
     .           BNL(IR,IS,IT,L,2,6) + STRESS(3) *
     .           BNL(IR,IS,IT,L,3,6) ) + BNL(IR,IS,IT,M,2,5) *
     .           STRESS(2) * BNL(IR,IS,IT,L,1,6) +
     .           BNL(IR,IS,IT,M,3,5) * STRESS(3) *
     .           BNL(IR,IS,IT,M,1,6) ) * XJAC * WR * WS * WT
              ENDIF
C
            ELSEIF(IE .EQ. 6) THEN
              IF(JE .EQ. 1) THEN
                SN = SN + ( BNL(IR,IS,IT,M,1,6) * ( STRESS(1) *
     .             BNL(IR,IS,IT,L,1,1) + STRESS(2) *
     .             BNL(IR,IS,IT,L,2,1) + STRESS(3) *
     .             BNL(IR,IS,IT,L,3,1) ) + BNL(IR,IS,IT,M,2,6) *
     .             STRESS(2) * BNL(IR,IS,IT,L,1,1) +
     .             BNL(IR,IS,IT,M,3,6) * STRESS(3) *
     .             BNL(IR,IS,IT,M,1,1) ) * XJAC * WR * WS * WT
C
                ELSEIF(JE .EQ. 2) THEN
                SN = SN + ( BNL(IR,IS,IT,M,4,6) * ( STRESS(1) *
     .               BNL(IR,IS,IT,L,4,2) + STRESS(2) *
     .               BNL(IR,IS,IT,L,5,2) + STRESS(3) *
     .               BNL(IR,IS,IT,L,6,2) ) + BNL(IR,IS,IT,M,5,6)
     .             * STRESS(2) * BNL(IR,IS,IT,L,4,2) +
     .               BNL(IR,IS,IT,M,6,6) * STRESS(3) *
     .               BNL(IR,IS,IT,M,4,2) ) * XJAC * WR * WS * WT
C
            ELSEIF(JE .EQ. 4) THEN
              SN = SN + ( BNL(IR,IS,IT,M,4,6) * ( STRESS(1) *
     .            BNL(IR,IS,IT,L,4,4) +
     .            STRESS(2) * BNL(IR,IS,IT,L,5,4) + STRESS(3) *
     .            BNL(IR,IS,IT,L,6,4) ) + BNL(IR,IS,IT,M,5,6)
     .          * STRESS(2) * BNL(IR,IS,IT,L,4,4) +
     .            BNL(IR,IS,IT,M,6,6) * STRESS(3) *
     .            BNL(IR,IS,IT,M,4,4) ) * XJAC * WR * WS * WT
C
            ELSEIF(JE .EQ. 5) THEN
            SN = SN + ( BNL(IR,IS,IT,M,1,6) * ( STRESS(1) *
     .         BNL(IR,IS,IT,L,1,5) + STRESS(2) *
     .         BNL(IR,IS,IT,L,2,5) + STRESS(3) *
     .         BNL(IR,IS,IT,L,3,5) ) + BNL(IR,IS,IT,M,2,6)
     .       * STRESS(2) * BNL(IR,IS,IT,L,1,5) +
     .         BNL(IR,IS,IT,M,3,6) * STRESS(3) *
     .         BNL(IR,IS,IT,M,1,5) ) * XJAC * WR * WS * WT
C
            ELSEIF(JE .EQ. 6) THEN
              SN = SN + ( BNL(IR,IS,IT,M,1,6) * ( STRESS(1) *
     .            BNL(IR,IS,IT,L,1,6) + STRESS(2) *
     .            BNL(IR,IS,IT,L,2,6) + STRESS(3) *
     .            BNL(IR,IS,IT,L,3,6) ) + BNL(IR,IS,IT,M,2,6) *
     .            STRESS(2) *
     .            BNL(IR,IS,IT,L,1,6) + BNL(IR,IS,IT,M,3,6) *
     .            STRESS(3) * BNL(IR,IS,IT,M,1,6)
     .            + BNL(IR,IS,IT,M,4,6) * ( STRESS(1) *
     .            BNL(IR,IS,IT,L,4,6) +
     .            STRESS(2) * BNL(IR,IS,IT,L,5,6) + STRESS(3) *
     .            BNL(IR,IS,IT,L,6,6) ) + BNL(IR,IS,IT,M,5,6) *
     .            STRESS(2) *
     .            BNL(IR,IS,IT,L,4,6) + BNL(IR,IS,IT,M,6,6) *
     .            STRESS(3) *
     .            BNL(IR,IS,IT,M,4,6) ) * XJAC * WR * WS * WT
C
            ENDIF
C
          ENDIF
  380     CONTINUE
  390   CONTINUE
  400 CONTINUE
C
      NP = N
      RETURN
      END
C**********************************************************************
C                       NODALB
C**********************************************************************
C
C COMPUTES THE LINEAR STRAIN-DISPLACEMENT MATRIX. IT CALLS 'INDEX'.
C
      SUBROUTINE NODALB(H,DH,RR,SS,TT,XJINV,DEPTH,WIDTH,XJAC,X,BB,
     .           VR,VS,VT,G,N,KG,KL,LL,IJE,VVR,VVS,VVT)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION H(*),DH(*),XJINV(3,*),DEPTH(*),WIDTH(*),VR(3,*),VS(3,*),
     .       VT(3,*),G(3,3,*),VVR(*),VVS(*),VVT(*)
C
C--  GET THE 'G' VALUE (EQ.  ) FOR THE CORRESPONDING INTERPOLATION POIN
C    AND ELEMENT.
C
      CALL INDEX(II,DH,SS,TT,XJINV,DEPTH,WIDTH,VR,VS,VT,G,N,KG,KL)
C
      IF(LL .EQ. 1) THEN
        IF(IJE .EQ. 1) THEN
          BB = DH(KL) * (VVR(1) * VVR(1) * XJINV(1,1) + VVR(1) *
     .       VVR(2) * XJINV(2,1) + VVR(1) * VVR(3) *
     .       XJINV(3,1) )
C
        ELSEIF(IJE .EQ. 2) THEN
          BB = DH(KL) * (VVR(2) * VVR(2) * XJINV(2,1) + VVR(1) *
     .       VVR(2) * XJINV(1,1) + VVR(2) * VVR(3) *
     .       XJINV(3,1) )
C
        ELSEIF(IJE .EQ. 3) THEN
          BB = DH(KL) * (VVR(3) * VVR(3) * XJINV(3,1) + VVR(2) *
     .       VVR(3) * XJINV(2,1) + VVR(1) * VVR(3) *
     .       XJINV(1,1) )
C
        ELSEIF(IJE .EQ. 4) THEN
          BB = VVR(2) * VVR(2) * G(1,2,2) + VVR(3) * VVR(3) *
     .       G(1,3,3) + VVR(1) * VVR(2) * G(1,2,1) + VVR(2) *
     .       VVR(3) * ( G(1,2,3) + G(1,3,2) ) + VVR(1) *
     .       VVR (3) * G(1,3,1)
C
        ELSEIF(IJE .EQ. 5) THEN
          BB = VVR(1) * VVR(1) * G(2,1,1) + VVR(3) * VVR(3) *
     .       G(2,3,3) + VVR(1) * VVR(2) * G(2,1,2) + VVR(2) *
     .       VVR(3) * G(2,3,2) + VVR(1) * VVR (3) *
     .       ( G(2,1,3) + G(2,3,1) )
C
        ELSEIF(IJE .EQ. 6) THEN
          BB = VVR(1) * VVR(1) * G(3,1,1) + VVR(2) * VVR(2) *
     .       G(3,2,2) + VVR(1) * VVR(2) * ( G(3,1,2) + G(3,2,1))
     .       + VVR(2) * VVR(3) * G(3,2,3) + VVR(1) * VVR (3)
     .       * G(3,1,3)
C
        ENDIF
      ELSEIF(LL .EQ. 2) THEN
        IF(IJE .EQ. 1) THEN
          BB = DH(KL) * ( 2.D0 * VVR(1) * VVS(1) * XJINV(1,1)
     .     + ( VVR(2) * VVS(1) + VVR(1) * VVS(2) ) * XJINV(2,1)
     .    + ( VVR(3) * VVS(1) + VVR(1) * VVS(3) ) * XJINV(3,1))
C
        ELSEIF(IJE .EQ. 2) THEN
          BB = DH(KL) * ( 2.D0 * VVR(2) * VVS(2) * XJINV(2,1)
     .     + ( VVR(2) * VVS(1) + VVR(1) * VVS(2) ) * XJINV(1,1)
     .    + ( VVR(3) * VVS(2) + VVR(2) * VVS(3) ) * XJINV(3,1))
C
        ELSEIF(IJE .EQ. 3) THEN
          BB = DH(KL) * ( 2.D0 * VVR(3) * VVS(3) * XJINV(3,1)
     .     + ( VVR(3) * VVS(2) + VVR(2) * VVS(3) ) * XJINV(2,1)
     .    + ( VVR(3) * VVS(1) + VVR(1) * VVS(3) ) * XJINV(1,1))
C
        ELSEIF(IJE .EQ. 4) THEN
          BB = 2.D0 * VVR(2) * VVS(2) * G(1,2,2) + 2.D0 * VVR(3)
     .       * VVS(3) * G(1,3,3) + ( VVR(2) * VVS(1) +
     .       VVR(1) * VVS(2) ) * G(1,2,1) + ( VVR(3) *
     .       VVS(2) + VVR(2) * VVS(3) ) * ( G(1,2,3) +
     .       G(1,3,2) ) + ( VVR(3) * VVS(1) + VVR(1) *
     .       VVS(3) ) * G(1,3,1)
C
        ELSEIF(IJE .EQ. 5) THEN
          BB = 2.D0 * VVR(1) * VVS(1) * G(2,1,1) + 2.D0 * VVR(3)
     .       * VVS(3) * G(2,3,3) + ( VVR(2) * VVS(1) +
     .       VVR(1) * VVS(2) ) * G(2,1,2) + ( VVR(3) *
     .       VVS(2) + VVR(2) * VVS(3) ) * G(2,3,2)
     .       + ( VVR(3) * VVS(1) + VVR(1) *
     .       VVS(3) ) * ( G(2,1,3) + G(2,3,1) )
C
        ELSEIF(IJE .EQ. 6) THEN
          BB = 2.D0 * VVR(1) * VVS(1) * G(3,1,1) + 2.D0 * VVR(2)
     .       * VVS(2) * G(3,2,2) + ( VVR(2) * VVS(1) +
     .       VVR(1) * VVS(2) ) * ( G(3,1,2) + G(3,2,1) ) +
     .       ( VVR(3) * VVS(2) + VVR(2) * VVS(3) )
     .       * G(3,2,3) + ( VVR(3) * VVS(1) + VVR(1) *
     .       VVS(3) ) * G(3,1,3)
        ENDIF
      ELSEIF(LL .EQ. 3) THEN
        IF(IJE .EQ. 1) THEN
          BB = DH(KL) * ( 2.D0 * VVR(1) * VVT(1) * XJINV(1,1)
     .     + ( VVR(2) * VVT(1) + VVR(1) * VVT(2) ) * XJINV(2,1)
     .    + ( VVR(3) * VVT(1) + VVR(1) * VVT(3) ) * XJINV(3,1))
C
        ELSEIF(IJE .EQ. 2) THEN
          BB = DH(KL) * ( 2.D0 * VVR(2) * VVT(2) * XJINV(2,1)
     .     + ( VVR(2) * VVT(1) + VVR(1) * VVT(2) ) * XJINV(1,1)
     .    + ( VVR(3) * VVT(2) + VVR(2) * VVT(3) ) * XJINV(3,1))
C
        ELSEIF(IJE .EQ. 3) THEN
          BB = DH(KL) * ( 2.D0 * VVR(3) * VVT(3) * XJINV(3,1)
     .     + ( VVR(3) * VVT(2) + VVR(2) * VVT(3) ) * XJINV(2,1)
     .    + ( VVR(3) * VVT(1) + VVR(1) * VVT(3) ) * XJINV(1,1))
C
        ELSEIF(IJE .EQ. 4) THEN
          BB = 2.D0 * VVR(2) * VVT(2) * G(1,2,2) + 2.D0 * VVR(3)
     .       * VVT(3) * G(1,3,3) + ( VVR(2) * VVT(1) +
     .       VVR(1) * VVT(2) ) * G(1,2,1) + ( VVR(3) *
     .       VVT(2) + VVR(2) * VVT(3) ) * ( G(1,2,3) +
     .       G(1,3,2) ) + ( VVR(3) * VVT(1) + VVR(1) *
     .       VVT(3) ) * G(1,3,1)
        ELSEIF(IJE .EQ. 5) THEN
          BB = 2.D0 * VVR(1) * VVT(1) * G(2,1,1) + 2.D0 * VVR(3)
     .       * VVT(3) * G(2,3,3) + ( VVR(2) * VVT(1) +
     .       VVR(1) * VVT(2) ) * G(2,1,2) + ( VVR(3) *
     .       VVT(2) + VVR(2) * VVT(3) ) * G(2,3,2)
     .       + ( VVR(3) * VVT(1) + VVR(1) *
     .       VVT(3) ) * ( G(2,1,3) + G(2,3,1) )
        ELSEIF(IJE .EQ. 6) THEN
          BB = 2.D0 * VVR(1) * VVT(1) * G(3,1,1) + 2.D0 * VVR(2)
     .       * VVT(2) * G(3,2,2) + ( VVR(2) * VVT(1) +
     .       VVR(1) * VVT(2) ) * ( G(3,1,2) + G(3,2,1) ) +
     .       ( VVR(3) * VVT(2) + VVR(2) * VVT(3) )
     .       * G(3,2,3) + ( VVR(3) * VVT(1) + VVR(1) *
     .       VVT(3) ) * G(3,1,3)
        ENDIF
      ENDIF
C
      RETURN
      END
C**********************************************************************
C                     DETJAC
```

```
C***********************************************************************
C
C     IT CALCULATES THE DETERMINANT OF THE JACOBIAN FOR AN INTERPOLATION
C     POINT.
C
      SUBROUTINE DETJAC(H,DH,RR,SS,TT,DEPTH,WIDTH,VS,VT,XJAC,X,N,MINC)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION H(*),DH(*),DEPTH(*),WIDTH(*),VS(3,*),VT(3,*),X(3,*),
     .          MINC(3,*)
C
      XJAC = 0.D0
C
      KG1 = MINC(1,N)
      KG2 = MINC(2,N)
      KG3 = MINC(3,N)
C
      A = DEPTH(N) * WIDTH(N) / 4.D0
      B = DH(1) * X(1,KG1) + DH(2) * X(1,KG2) + DH(3) * X(1,KG3)
      C = DH(1) * X(2,KG1) + DH(2) * X(2,KG2) + DH(3) * X(2,KG3)
      D = DH(1) * X(3,KG1) + DH(2) * X(3,KG2) + DH(3) * X(3,KG3)
C
      E = H(1) * VS(2,KG1) + H(2) * VS(2,KG2) + H(3) * VS(2,KG3)
      F = H(1) * VS(3,KG1) + H(2) * VS(3,KG2) + H(3) * VS(3,KG3)
      G = H(1) * VS(1,KG1) + H(2) * VS(1,KG2) + H(3) * VS(1,KG3)
      O = H(1) * VT(1,KG1) + H(2) * VT(1,KG2) + H(3) * VT(1,KG3)
      P = H(1) * VT(2,KG1) + H(2) * VT(2,KG2) + H(3) * VT(2,KG3)
      Q = H(1) * VT(3,KG1) + H(2) * VT(3,KG2) + H(3) * VT(3,KG3)
C
      XJAC1 = A * ( B * (E * Q - F * P) - C * (G * Q - F * O) + D *
     .    (G * P - E * O) )
C
      A = DEPTH(N) * DEPTH(N) * WIDTH(N) * SS / 8.D0
      B = H(1) * VT(1,KG1) + H(2) * VT(1,KG2) + H(3) * VT(1,KG3)
      C = H(1) * VT(2,KG1) + H(2) * VT(2,KG2) + H(3) * VT(2,KG3)
      D = H(1) * VT(3,KG1) + H(2) * VT(3,KG2) + H(3) * VT(3,KG3)
C
      E = DH(1) * H(2) * VS(2,KG1) * VS(3,KG2) + DH(1) * H(3) *
     .  VS(2,KG1) * VS(3,KG3) + DH(2) * H(1) * VS(2,KG2) * VS(3,KG1)
     .   + DH(2) * H(3) * VS(2,KG2) * VS(3,KG3) + DH(3) * H(1) *
     .  VS(2,KG3) * VS(3,KG1) + DH(3) * H(2) * VS(2,KG3) *
     .  VS(3,KG2)
C
      F = DH(1) * H(2) * VS(2,KG2) * VS(3,KG1) + DH(1) * H(3) *
     .  VS(2,KG3) * VS(3,KG1) + DH(2) * H(1) * VS(2,KG1) * VS(3,KG2)
     .   + DH(2) * H(3) * VS(2,KG3) * VS(3,KG2) + DH(3) * H(1) *
     .  VS(2,KG1) * VS(3,KG3) + DH(3) * H(2) * VS(2,KG2) *
     .  VS(3,KG3)
C
      G = DH(1) * H(2) * VS(1,KG1) * VS(3,KG2) + DH(1) * H(3) *
     .  VS(1,KG1) * VS(3,KG3) + DH(2) * H(1) * VS(1,KG2) * VS(3,KG1)
     .   + DH(2) * H(3) * VS(1,KG2) * VS(3,KG3) + DH(3) * H(1) *
     .  VS(1,KG3) * VS(3,KG1) + DH(3) * H(2) * VS(1,KG3) *
     .  VS(3,KG2)
C
      O = DH(1) * H(2) * VS(1,KG2) * VS(3,KG1) + DH(1) * H(3) *
     .  VS(1,KG3) * VS(3,KG1) + DH(2) * H(1) * VS(1,KG1) * VS(3,KG2)
     .   + DH(2) * H(3) * VS(1,KG3) * VS(3,KG2) + DH(3) * H(1) *
     .  VS(1,KG1) * VS(3,KG3) + DH(3) * H(2) * VS(1,KG2) *
     .  VS(3,KG3)
C
      P = DH(1) * H(2) * VS(1,KG1) * VS(2,KG2) + DH(1) * H(3) *
     .  VS(1,KG1) * VS(2,KG3) + DH(2) * H(1) * VS(1,KG2) * VS(2,KG1)
     .   + DH(2) * H(3) * VS(1,KG2) * VS(2,KG3) + DH(3) * H(1) *
     .  VS(1,KG3) * VS(2,KG1) + DH(3) * H(2) * VS(1,KG3) *
     .  VS(2,KG2)
C
        Q = DH(1) * H(2) * VS(1,KG2) * VS(2,KG1) + DH(1) * H(3) *
     .  VS(1,KG3) * VS(2,KG1) + DH(2) * H(1) * VS(1,KG1) * VS(2,KG2)
     .   + DH(2) * H(3) * VS(1,KG3) * VS(2,KG2) + DH(3) * H(1) *
     .  VS(1,KG1) * VS(2,KG3) + DH(3) * H(2) * VS(1,KG2) *
     .  VS(2,KG3)
C
        XJAC2 = A * ( B * (E - F) + C * (O - G) + D * (P - Q) )
C
        A = DEPTH(N) * WIDTH(N) * WIDTH(N) * TT / 8.D0
C
        E = DH(1) * H(1) * VS(3,KG1) * VT(2,KG1) + DH(2) * H(2) *
     .  VS(3,KG2) * VT(2,KG2) + DH(3) * H(3) * VS(3,KG3) * VT(2,KG3) +
     .  DH(1) * H(2) * VS(3,KG2) * VT(2,KG1) + DH(1) * H(3) *
     .  VT(2,KG1) * VS(3,KG3) + DH(2) * H(1) * VT(2,KG2) * VS(3,KG1)
     .   + DH(2) * H(3) * VT(2,KG2) * VS(3,KG3) + DH(3) * H(1) *
     .  VT(2,KG3) * VS(3,KG1) + DH(3) * H(2) * VT(2,KG3) *
     .  VS(3,KG2)
C
        F = DH(1) * H(1) * VT(3,KG1) * VS(2,KG1) + DH(2) * H(2) *
     .  VT(3,KG2) * VS(2,KG2) + DH(3) * H(3) * VT(3,KG3) * VS(2,KG3) +
     .  DH(1) * H(2) * VT(3,KG1) * VS(2,KG2) + DH(1) * H(3) *
     .  VT(3,KG1) * VS(2,KG3) + DH(2) * H(1) * VT(3,KG2) * VS(2,KG1)
     .   + DH(2) * H(3) * VT(3,KG2) * VS(2,KG3) + DH(3) * H(1) *
     .  VT(3,KG3) * VS(2,KG1) + DH(3) * H(2) * VT(3,KG3) *
     .  VS(2,KG2)
C
        G = DH(1) * H(1) * VS(3,KG1) * VT(1,KG1) + DH(2) * H(2) *
     .  VS(3,KG2) * VT(1,KG2) + DH(3) * H(3) * VS(3,KG3) * VT(1,KG3) +
     .  DH(1) * H(2) * VS(3,KG2) * VT(1,KG1) + DH(1) * H(3) *
     .  VT(1,KG1) * VS(3,KG3) + DH(2) * H(1) * VT(1,KG2) * VS(3,KG1)
     .   + DH(2) * H(3) * VT(1,KG2) * VS(3,KG3) + DH(3) * H(1) *
     .  VT(1,KG3) * VS(3,KG1) + DH(3) * H(2) * VT(1,KG3) *
     .  VS(3,KG2)
C
        O = DH(1) * H(1) * VT(3,KG1) * VS(1,KG1) + DH(2) * H(2) *
     .  VT(3,KG2) * VS(1,KG2) + DH(3) * H(3) * VT(3,KG3) * VS(1,KG3) +
     .  DH(1) * H(2) * VT(3,KG1) * VS(1,KG2) + DH(1) * H(3) *
     .  VS(1,KG3) * VT(3,KG1) + DH(2) * H(1) * VS(1,KG1) * VT(3,KG2)
     .   + DH(2) * H(3) * VS(1,KG3) * VT(3,KG2) + DH(3) * H(1) *
     .  VS(1,KG1) * VT(3,KG3) + DH(3) * H(2) * VS(1,KG2) *
     .  VT(3,KG3)
C
        P = DH(1) * H(1) * VS(2,KG1) * VT(1,KG1) + DH(2) * H(2) *
     .  VS(2,KG2) * VT(1,KG2) + DH(3) * H(3) * VS(2,KG3) * VT(1,KG3) +
     .  DH(1) * H(2) * VS(2,KG2) * VT(1,KG1) + DH(1) * H(3) *
     .  VT(1,KG1) * VS(2,KG3) + DH(2) * H(1) * VT(1,KG2) * VS(2,KG1)
     .   + DH(2) * H(3) * VT(1,KG2) * VS(2,KG3) + DH(3) * H(1) *
     .  VT(1,KG3) * VS(2,KG1) + DH(3) * H(2) * VT(1,KG3) *
     .  VS(2,KG2)
C
        Q = DH(1) * H(1) * VS(1,KG1) * VT(2,KG1) + DH(2) * H(2) *
     .  VT(2,KG2) * VS(1,KG2) + DH(3) * H(3) * VT(2,KG3) * VS(1,KG3) +
     .  DH(1) * H(2) * VT(2,KG1) * VS(1,KG2) + DH(1) * H(3) *
     .  VS(1,KG3) * VT(2,KG1) + DH(2) * H(1) * VS(1,KG1) * VT(2,KG2)
     .   + DH(2) * H(3) * VS(1,KG3) * VT(2,KG2) + DH(3) * H(1) *
     .  VS(1,KG1) * VT(2,KG3) + DH(3) * H(2) * VS(1,KG2) *
     .  VT(2,KG3)
C
        XJAXC3 = A * ( B *(E - F) + C * (O - G) + D * (P - Q) )
C
```

```
      XJAC = XJAC1 + XJAC2 + XJAC3
C
      RETURN
      END
C***********************************************************************
C                        NOLINB
C***********************************************************************
C
C   IT CALCULATES THE NONLINEAR STRAIN-DISPLACEMENT MATRIX. IT CALLS
C   'INDEX'.
C
      SUBROUTINE NOLINB(IR,IS,IT,RR,SS,TT,XJINV,DEPTH,WIDTH,H,DH,
     .           XJAC,X,VR,VS,VT,G,N,VVR,VVS,VVT,MINC,BNL)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION H(*),DH(*),XJINV(3,*),DEPTH(*),WIDTH(*),VR(3,*),VS(3,*),
     .       VT(3,*),G(3,3,*),VVR(*),VVS(*),VVT(*),X(3,*),MINC(3,*),
     .       BNL(2,2,2,3,9,*)
C
      DO 10 KL = 1,3
        KG = MINC(KL,N)
        CALL INDEX(H,DH,SS,TT,XJINV,DEPTH,WIDTH,VR,VS,VT,G,N,KG,KL)
C
        BNL(IR,IS,IT,KL,1,1) = VVR(1) * XJINV(1,1) * DH(KL) + VVR(2) *
     .  XJINV(2,2) * DH(KL) + VVR(3) * XJINV(3,1) * DH(KL)
C
        BNL(IR,IS,IT,KL,1,2) = 0.D0
C
        BNL(IR,IS,IT,KL,1,3) = 0.D0
C
        BNL(IR,IS,IT,KL,1,4) = VVR(1) * G(1,1,1) + VVR(2) * G (1,1,2) +
     .                     VVR(3) * G(1,1,3)
C
        BNL(IR,IS,IT,KL,1,5) = VVR(1) * G(2,1,1) + VVR(2) *
     .                     G(2,1,2) + VVR(3) * G(2,1,3)
C
        BNL(IR,IS,IT,KL,1,6) = VVR(1) * G(3,1,1) + VVR(2) * G(3,1,2) +
     .                     VVR(3) * G(3,1,3)
C
        BNL(IR,IS,IT,KL,2,1) = VVS(1) * XJINV(1,1) * DH(KL) + VVS(2) *
     .                     XJINV(2,2) * DH(KL) +
     .                     VVS(3) * XJINV(3,1) * DH(KL)
C
        BNL(IR,IS,IT,KL,2,2) = 0.D0
C
        BNL(IR,IS,IT,KL,2,3) = 0.D0
C
        BNL(IR,IS,IT,KL,2,4) = VVS(1) * G(1,1,1) + VVS(2) * G(1,1,2) +
     .                     G(1,1,3) * VVS(3)
C
        BNL(IR,IS,IT,KL,2,5) = VVS(1) * G(2,1,1) + VVS(2) *
     .                     G(2,1,2) + VVS(3) * G(2,1,3)
C
        BNL(IR,IS,IT,KL,2,6) = VVS(1) * G(3,1,1) + VVS(2) * G(3,1,2) +
     .                     VVS(3) * G(3,1,3)
C
        BNL(IR,IS,IT,KL,3,1) = VVT(1) * XJINV(1,1) * DH(KL) + VVT(2)
     .                     * XJINV(2,2) * DH(KL)
     .                     + VVT(3) * XJINV(3,1) * DH(KL)
C
        BNL(IR,IS,IT,KL,3,2) = 0.D0
C
        BNL(IR,IS,IT,KL,3,3) = 0.D0
C
```

```
      BNL(IR,IS,IT,KL,3,4) = VVT(1) * G(1,1,1) + VVT(2) * G(1,1,2) +
     .                G(1,1,3) * VVT(3)
C
      BNL(IR,IS,IT,KL,3,5) = VVT(1) * G(2,1,1) + VVT(2) *
     .                G(2,1,2) + VVT(3) * G(2,1,3)
C
      BNL(IR,IS,IT,KL,3,6) = VVT(1) * G(3,1,1) + VVT(2) * G(3,1,2) +
     .                VVT(3) * G(3,1,3)
C
      BNL(IR,IS,IT,KL,4,1) = 0.D0
C
      BNL(IR,IS,IT,KL,4,2) = VVR(1) * XJINV(1,1) * DH(KL) + VVR(2) *
     .                XJINV(2,2) * DH(KL) + VVR(3)
     .                * XJINV(3,1) * DH(KL)
C
      BNL(IR,IS,IT,KL,4,3) = 0.D0
C
      BNL(IR,IS,IT,KL,4,4) = VVR(1) * G(1,2,1) + VVR(2) * G(1,2,2) +
     .                G(1,2,3) * VVR(3)
C
      BNL(IR,IS,IT,KL,4,5) = VVR(1) * G(2,2,1) + VVR(2) * G(2,2,2) +
     .                G(2,2,3) * VVR(3)
C
      BNL(IR,IS,IT,KL,4,6) = VVR(1) * G(3,2,1) + VVR(2) * G(3,2,2) +
     .                G(3,2,3) * VVR(3)
C
      BNL(IR,IS,IT,KL,5,1) = 0.D0
C
      BNL(IR,IS,IT,KL,5,2) = VVS(1) * XJINV(1,1) * DH(KL) + VVS(2)
     .               * XJINV(2,2) * DH(KL) + VVS(3)
     .               * XJINV(3,1) * DH(KL)
C
      BNL(IR,IS,IT,KL,5,3) = 0.D0
C
      BNL(IR,IS,IT,KL,5,4) = VVS(1) * G(1,2,1) + VVS(2) * G(1,2,2) +
     .                G(1,2,3) * VVS(3)
C
      BNL(IR,IS,IT,KL,5,5) = VVS(1) * G(2,2,1) + VVS(2) * G(2,2,2) +
     .                G(2,2,3) * VVS(3)
C
      BNL(IR,IS,IT,KL,5,6) = VVS(1) * G(3,2,1) + VVS(2) * G(3,2,2) +
     .                G(3,2,3) * VVS(3)
C
      BNL(IR,IS,IT,KL,6,1) = 0.D0
C
      BNL(IR,IS,IT,KL,6,2) = VVT(1) * XJINV(1,1) * DH(KL) + VVT(2)
     .               * XJINV(2,2) * DH(KL) + VVT(3) *
     .               XJINV(3,1) * DH(KL)
C
      BNL(IR,IS,IT,KL,6,3) = 0.D0
C
      BNL(IR,IS,IT,KL,6,4) = VVT(1) * G(1,2,1) + VVT(2) * G(1,2,2) +
     .                G(1,2,3) * VVT(3)
C
      BNL(IR,IS,IT,KL,6,5) = VVT(1) * G(2,2,1) + VVT(2) * G(2,2,2) +
     .                G(2,2,3) * VVT(3)
C
      BNL(IR,IS,IT,KL,6,6) = VVT(1) * G(3,2,1) + VVT(2) * G(3,2,2) +
     .                G(3,2,3) * VVT(3)
C
      BNL(IR,IS,IT,KL,7,1) = 0.D0
C
```

```
      BNL(IR,IS,IT,KL,7,2) = 0.D0
C
      BNL(IR,IS,IT,KL,7,3) = VVR(1) * XJINV(1,1) * DH(KL) + VVR(2)
     .              * XJINV(2,2) * DH(KL) + VVR(3) *
     .              XJINV(3,1) * DH(KL)
C
      BNL(IR,IS,IT,KL,7,4) = VVR(1) * G(1,3,1) + VVR(2) * G(1,3,2) +
     .               G(1,3,3) * VVR(3)
C
      BNL(IR,IS,IT,KL,7,5) = VVR(1) * G(2,3,1) + VVR(2) * G(2,3,2) +
     .               G(2,3,3) * VVR(3)
C
      BNL(IR,IS,IT,KL,7,6) = VVR(1) * G(3,3,1) + VVR(2) * G(3,3,2) +
     .               G(3,3,3) * VVR(3)
C
      BNL(IR,IS,IT,KL,8,1) = 0.D0
C
      BNL(IR,IS,IT,KL,8,2) = 0.D0
C
      BNL(IR,IS,IT,KL,8,3) = VVS(1) * XJINV(1,1) * DH(KL) + VVS(2)
     .              * XJINV(2,2) * DH(KL) + VVS(3) *
     .              XJINV(3,1) * DH(KL)
C
      BNL(IR,IS,IT,KL,8,4) = VVS(1) * G(1,3,1) + VVS(2) * G(1,3,2) +
     .              G(1,3,3) * VVS(3)
C
      BNL(IR,IS,IT,KL,8,5) = VVS(1) * G(2,3,1) + VVS(2) * G(2,3,2) +
     .               G(2,3,3) * VVS(3)
C
      BNL(IR,IS,IT,KL,8,6) = VVS(1) * G(3,3,1) + VVS(2) * G(3,3,2) +
     .              G(3,3,3) * VVS(3)
C
      BNL(IR,IS,IT,KL,9,1) = 0.D0
C
      BNL(IR,IS,IT,KL,9,2) = 0.D0
C
      BNL(IR,IS,IT,KL,9,3) = VVT(1) * XJINV(1,1) * DH(KL) + VVT(2)
     .              * XJINV(2,2) * DH(KL) + VVT(3)
     .              * XJINV(3,1) * DH(KL)
C
      BNL(IR,IS,IT,KL,9,4) = VVT(1) * G(1,3,1) + VVT(2) * G(1,3,2) +
     .               G(1,3,3) * VVT(3)
C
      BNL(IR,IS,IT,KL,9,5) = VVT(1) * G(2,3,1) + VVT(2) * G(2,3,2) +
     .               G(2,3,3) * VVT(3)
C
      BNL(IR,IS,IT,KL,9,6) = VVT(1) * G(3,3,1) + VVT(2) * G(3,3,2) +
     .               G(3,3,3) * VVT(3)
C
 10   CONTINUE
      RETURN
      END
C*****************************************************************
C                         INVJAC
C*****************************************************************
C
C     CALCULATES THE INVERSE OF JACOBIAN MATRIX FOR THE GIVEN
C     INTERPOLATION POINTS.
C
      SUBROUTINE INVJAC(H,DH,RR,DEPTH,WIDTH,VS,VT,XJAC,X,N,XJINV,
     .          SS,TT,MINC)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION H(*),DH(*),DEPTH(*),WIDTH(*),VS(3,*),VT(3,*),X(3,*),
     .     XJINV(3,*),MINC(3,*)
C
C--- INITIALIZATION OF THE XJINV MATRIX.
C
      DO 20 JX = 1,3
        DO 10 IX = 1,3
          XJINV(IX,JX) = 0.D0
 10     CONTINUE
 20   CONTINUE
C
C--- COMPUTATION OF THE ADJ J MATRIX
C
      KG1 = MINC(1,N)
      KG2 = MINC(2,N)
      KG3 = MINC(3,N)
C
      A = H(1) * VS(2,KG1) + H(2) * VS(2,KG2) + H(3) * VS(2,KG3)
      B = H(1) * VT(3,KG1) + H(2) * VT(3,KG2) + H(3) * VT(3,KG3)
      C = H(1) * VS(3,KG1) + H(2) * VS(3,KG2) + H(3) * VS(3,KG3)
      D = H(1) * VT(2,KG1) + H(2) * VT(2,KG2) + H(3) * VT(2,KG3)
C
      XJINV(1,1) = WIDTH(N) * DEPTH(N) / 4.D0 * (A * B - C * D)
C
      A = H(1) * VS(1,KG1) + H(2) * VS(1,KG2) + H(3) * VS(1,KG3)
      D = H(1) * VT(1,KG1) + H(2) * VT(1,KG2) + H(3) * VT(1,KG3)
C
      XJINV(2,1) = - DEPTH(N) * WIDTH(N) / 4.D0 * (A * B - C * D)
C
      B = H(1) * VT(2,KG1) + H(2) * VT(2,KG2) + H(3) * VT(2,KG3)
      C = H(1) * VS(2,KG1) + H(2) * VS(2,KG2) + H(3) * VS(2,KG3)
C
      XJINV(3,1) = DEPTH(N) * WIDTH(N) / 4.D0 * (A * B - C * D)
C
      A = DH(1) * ( X(1,KG1) + DEPTH(N) * SS * 0.5D0 * VS(1,KG1) +
     .    WIDTH(N) * TT * 0.5D0 * VT(1,KG1) ) + DH(2) * ( X(1,KG2) +
     .    0.5D0 * DEPTH(N) * SS * VS(1,KG2) + WIDTH(N) * TT / 2.D0 *
     .    VT(1,KG2) ) + DH(3) * ( X(1,KG3) + 0.5D0 * DEPTH(N) * SS *
     .    VS(1,KG3) + WIDTH(N) * TT / 2.D0 * VT(1,KG3) )
C
      C = DH(1) * ( X(2,KG1) + DEPTH(N) * SS * 0.5D0 * VS(2,KG1) +
     .    WIDTH(N) * TT * 0.5D0 * VT(2,KG1) ) + DH(2) * ( X(2,KG2) +
     .    0.5D0 * DEPTH(N) * SS * VS(2,KG2) + WIDTH(N) * TT / 2.D0 *
     .    VT(2,KG2) ) + DH(3) * ( X(2,KG3) + 0.5D0 * DEPTH(N) * SS *
     .    VS(2,KG3) + WIDTH(N) * TT / 2.D0 * VT(2,KG3) )
C
      XJINV(3,2) = - WIDTH(N) / 2.D0 * (A * B - C * D)
C
      D = H(1) * VS(1,KG1) + H(2) * VS(1,KG2) + H(3) * VS(1,KG3)
      B = H(1) * VS(2,KG1) + H(2) * VS(2,KG2) + H(3) * VS(2,KG3)
C
      XJINV(3,3) = DEPTH(N) / 2.D0 * (A * B - C * D)
C
      B = H(1) * VS(3,KG1) + H(2) * VS(3,KG2) + H(3) * VS(3,KG3)
      C = DH(1) * ( X(3,KG1) + DEPTH(N) * SS * 0.5D0 * VS(3,KG1) +
     .    WIDTH(N) * TT * 0.5D0 * VT(3,KG1) ) + DH(2) * ( X(3,KG2) +
     .    0.5D0 * DEPTH(N) * SS * VS(3,KG2) + WIDTH(N) * TT / 2.D0 *
     .    VT(3,KG2) ) + DH(3) * ( X(3,KG3) + 0.5D0 * DEPTH(N) * SS *
     .    VS(3,KG3) + WIDTH(N) * TT / 2.D0 * VT(3,KG3) )
C
      XJINV(2,3) = - DEPTH(N) / 2.D0 * (A * B - C * D)
C
      B = H(1) * VT(3,KG1) + H(2) * VT(3,KG2) + H(3) * VT(3,KG3)
      D = H(1) * VT(1,KG1) + H(2) * VT(1,KG2) + H(3) * VT(1,KG3)
C
      XJINV(2,2) = WIDTH(N) / 2.D0 * (A * B - C * D)
C
      A = DH(1) * ( X(2,KG1) + DEPTH(N) * SS * 0.5D0 * VS(2,KG1) +
     .  WIDTH(N) * TT * 0.5D0 * VT(2,KG1) ) + DH(2) * ( X(2,KG2) +
     .  0.5D0 * DEPTH(N) * SS * VS(2,KG2) + WIDTH(N) * TT / 2.D0 *
     .  VT(2,KG2) ) + DH(3) * ( X(2,KG3) + 0.5D0 * DEPTH(N) * SS *
     .  VS(2,KG3) + WIDTH(N) * TT / 2.D0 * VT(2,KG3) )
C
      D = H(1) * VT(2,KG1) + H(2) * VT(2,KG2) + H(3) * VT(2,KG3)
C
      XJINV(1,2) = - WIDTH(N) / 2.D0 * (A * B - C * D)
C
      B = H(1) * VS(3,KG1) + H(2) * VS(3,KG2) + H(3) * VS(3,KG3)
      D = H(1) * VS(2,KG1) + H(2) * VS(2,KG2) + H(3) * VS(2,KG3)
C
      XJINV(1,3) = DEPTH(N) / 2.D0 * (A * B - C * D)
C
C---  COMPUTATION OF THE INVERSE OF J MATRIX.
C
      DO 40 JX = 1,3
        DO 30 IX = 1,3
          XJINV(IX,JX) = XJINV(IX,JX) / XJAC
   30   CONTINUE
   40 CONTINUE
C
      RETURN
      END
C**********************************************************************
C                        INDEX
C**********************************************************************
C
C     IT CALCULATES THE G VALUES (SEE EQ. 4.21) AT A GIVEN
C     INTERPOLATION POINT FOR AN ELEMENT.
C
      SUBROUTINE INDEX(H,DH,SS,TT,XJINV,DEPTH,WIDTH,VR,VS,VT,G,N,KG,KL)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION H(*),DH(*),XJINV(3,*),DEPTH(*),WIDTH(*),VR(3,*),VS(3,*),
     .          VT(3,*),G(3,3,*)
C
      DO 30 IX = 1,3
        DO 20 JX = 1,3
          DO 10 KX = 1,3
            G(IX,JX,KX) = 0.D0
   10     CONTINUE
   20   CONTINUE
   30 CONTINUE
C
C--- G(I,J,K).....I = 1, J = 1.......ALL ZEROES
C
C--- G(I,J,K).....I = 1, J = 2.......
C
      A = DEPTH(N) / 2.D0 * VS(3,KG)
      B = WIDTH(N) / 2.D0 * VT(3,KG)
C
      DO 40 KX = 1,3
        G(1,2,KX) = -XJINV(KX,1) * (A * SS + B *TT) * DH(KL)
     .          - (XJINV(KX,2) * A + XJINV(KX,3) * B) * H(KL)
   40 CONTINUE
C
C--- G(I,J,K).....I = 1, J = 3.......
C
```

```
      A = DEPTH(N) / 2.D0 * VS(2,KG)
      B = WIDTH(N) / 2.D0 * VT(2,KG)
C
      DO 50 KX = 1,3
        G(1,3,KX) = XJINV(KX,1) * (A * SS + B *TT) * DH(KL)
     .          + (XJINV(KX,2) * A + XJINV(KX,3) * B) * H(KL)
 50   CONTINUE
C
C--- G(I,J,K).....I = 2, J = 1.......
C
      A = DEPTH(N) / 2.D0 * VS(3,KG)
      B = WIDTH(N) / 2.D0 * VT(3,KG)
C
      DO 60 KX = 1,3
        G(2,1,KX) = XJINV(KX,1) * (A * SS + B *TT) * DH(KL)
     .          + (XJINV(KX,2) * A + XJINV(KX,3) * B) * H(KL)
 60   CONTINUE
C
C--- G(I,J,K).....I = 2, J = 2.......ALL ZEROES
C
C--- G(I,J,K).....I = 2, J = 3.......
C
      A = DEPTH(N) / 2.D0 * VS(1,KG)
      B = WIDTH(N) / 2.D0 * VT(1,KG)
C
      DO 70 KX = 1,3
        G(2,3,KX) = -XJINV(KX,1) * (A * SS + B *TT) * DH(KL)
     .          - (XJINV(KX,2) * A + XJINV(KX,3) * B) * H(KL)
 70   CONTINUE
C
C--- G(I,J,K).....I = 3, J = 1.......
C
      A = DEPTH(N) / 2.D0 * VS(2,KG)
      B = WIDTH(N) / 2.D0 * VT(2,KG)
C
      DO 80 KX = 1,3
        G(3,1,KX) = -XJINV(KX,1) * (A * SS + B *TT) * DH(KL)
     .          - (XJINV(KX,2) * A + XJINV(KX,3) * B) * H(KL)
 80   CONTINUE
C
C--- G(I,J,K).....I = 3, J = 2.......
C
      A = DEPTH(N) / 2.D0 * VS(1,KG)
      B = WIDTH(N) / 2.D0 * VT(1,KG)
C
      DO 90 KX = 1,3
        G(3,2,KX) = XJINV(KX,1) * (A * SS + B *TT) * DH(KL)
     .          + (XJINV(KX,2) * A + XJINV(KX,3) * B) * H(KL)
 90   CONTINUE
C
C--- G(I,J,K).....I = 3, J = 3.......ALL ZEROES
C
      RETURN
      END
C**********************************************************************
C                        INTPNT
C**********************************************************************
C
C     IT FIRST INITIALIZES THE WEIGHT AND COORDINATE VECTORS FOR THE
C     INTERPOLATION POINTS AND THEN ASSIGNS THE PROPER VALUES BASED
C     ON THE NUMBER OF INTEGRATION POINTS (GAUSS RULE).
C
      SUBROUTINE INTPNT(R,S,T,IR,IS,IT,WTR,WTS,WTT)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION R(*),S(*),T(*),WTR(*),WTS(*),WTT(*)
C
      DO 10 I = 1,3
        R(I) = 0.D0
        WTR(I) = 0.D0
        WTS(I) = 0.D0
        WTT(I) = 0.D0
 10   CONTINUE
C
      DO 20 I = 1,5
        S(I) = 0.D0
        T(I) = 0.D0
 20   CONTINUE
C
      IF(IR .EQ. 3) THEN
        R(1) = -0.774596669241483D0
        R(2) = 0.D0
        R(3) = 0.774596669241483D0
C
        WTR(1) = 0.555555555555556D0
        WTR(2) = 0.888888888888889D0
        WTR(3) = 0.555555555555556D0
C
      ELSEIF(IR .EQ. 2) THEN
        R(1) = - 0.577350269189626D0
        R(2) = 0.577350269189626D0
C
        WTR(1) = 1.D0
        WTR(2) = 1.D0
      ENDIF
C
      IF(IS .EQ. 2) THEN
        S(1) = - 0.577350269189626D0
        S(2) = 0.577350269189626D0
C
        WTS(1) = 1.D0
        WTS(2) = 1.D0
C
      ELSEIF(IS .EQ. 3) THEN
C
        S(1) = -0.774596669241483D0
        S(2) = 0.D0
        S(3) = 0.774596669241483D0
C
        WTS(1) = 0.555555555555556D0
        WTS(2) = 0.888888888888889D0
        WTS(3) = 0.555555555555556D0
C
      ENDIF
C
      IF(IT .EQ. 2) THEN
        T(1) = - 0.577350269189626D0
        T(2) = 0.577350269189626D0
C
        WTT(1) = 1.D0
        WTT(2) = 1.D0
C
      ELSEIF(IT .EQ. 3) THEN
        T(1) = -0.774596669241483D0
        T(2) = 0.D0
        T(3) = 0.774596669241483D0
C
```

```
        WTT(1) = 0.555555555555556D0
        WTT(2) = 0.888888888888889D0
        WTT(3) = 0.555555555555556D0
C
      ENDIF
      RETURN
      END
C**********************************************************************
C                       OUTPNT
C**********************************************************************
C     IT FIRST INITIALIZES THE COORDINATE VECTORS FOR THE
C     INTERPOLATION POINTS AND THEN ASSIGNS THE PROPER VALUES BASED
C     ON THE NUMBER OF INTEGRATION POINTS (NEWTON COTES).
C
      SUBROUTINE OUTPNT(R,S,T,IR,IS,IT)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION R(*),S(*),T(*)
C
      DO 10 I = 1,3
        R(I) = 0.D0
   10 CONTINUE
C
      DO 20 I = 1,5
        S(I) = 0.D0
        T(I) = 0.D0
   20 CONTINUE
C
      R(1) = -1.D0
      R(2) = 0.D0
      R(3) = 1.D0
C
      IF(IS .EQ. 2) THEN
        S(1) = -1.D0
        S(2) = 1.D0
      ELSEIF(IS .EQ. 3) THEN
        S(1) = -1.D0
        S(2) = 0.D0
        S(3) = 1.D0
      ELSEIF(IS .EQ. 5) THEN
        S(1) = -1.D0
        S(2) = -0.5D0
        S(3) = 0.D0
        S(4) = 0.5D0
        S(5) = 1.D0
      ENDIF
C
      IF(IT .EQ. 2) THEN
        T(1) = -1.D0
        T(2) = 1.D0
      ELSEIF(IT .EQ. 3) THEN
        T(1) = -1.D0
        T(2) = 0.D0
        T(3) = 1.D0
      ELSEIF(IT .EQ. 5) THEN
        T(1) = -1.D0
        T(2) = -0.5D0
        T(3) = 0.D0
        T(4) = 0.5D0
        T(5) = 1.D0
      ENDIF
C
      RETURN
      END
C***********************************************************************
C                      POST
C***********************************************************************
C     IT PERFORMS POST-PROCESSING FOR LINEAR ANALYSIS. IT CALLS
C     'OUTPNT', 'STRES', 'INTPNT', 'INTERF', AND 'JOINTF'.
C
      SUBROUTINE POST(R,S,T,EPS,STRESS,MINC,JCODE,D,B,H,DH,XJINV,DEPTH,
     .          WIDTH,X,VR,VS,VT,G,DNL,GMOD,EMOD,WTR,WTS,WTT,
     .          FINT,P,NE,MCODE,VVR,VVS,VVT)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION R(*),S(*),T(*),EPS(*),STRESS(*),MINC(3,*),JCODE(6,*),
     .      D(*),B(3,3,*),H(*),DH(*),XJINV(3,*),DEPTH(*),WIDTH(*),
     .      X(3,*),VR(3,*),VS(3,*),VT(3,*),G(3,3,*),DNL(6,*),
     .      GMOD(*),EMOD(*),WTR(*),WTS(*),WTT(*),FINT(*),P(6,*),
     .      MCODE(18,*),VVR(*),VVS(*),VVT(*)
C
      REWIND (3)
      REWIND (4)
      REWIND (9)
      REWIND (10)
      REWIND (11)
      REWIND (12)
      REWIND (13)
      REWIND (14)
      REWIND (15)
C
C---  READ THE NODE WHOSE DISPLACEMENTS ARE TO BE ECHOED.
C
      READ(5,*) IND
      WRITE (3) IND
C
 10   IF(IND .NE. 0) THEN
        DO 20 KK = 1,6
          NGLO = JCODE(KK,IND)
          IF(NGLO .NE. 0) THEN
            DEF = D(NGLO)
          ELSE
            DEF = 0.D0
          ENDIF
C
          WRITE (4) DEF
 20     CONTINUE
          READ(5,*) IND
          WRITE (3) IND
        GO TO 10
      ENDIF
C
C---  READ THE ELEMENT NUMBERS WHOSE STRESSES AND STRAINS ARE TO BE
C     ECHOED.
C
        READ (5,*) IES
        WRITE (9) IES
C
      IF(IES .NE. 0) THEN
C
C...  SET THE PARAMETERS FOR THE NUMBER OF OUTPUT POINTS. ( IR = NO OF
C     POINTS IN R DIRECTION, IS = NO. OF POINTS IN S DIRECTION, IT = NO.
C     OF POINTS IN T DIRECTION ).
      IR = 3
      IS = 3
      IT = 3
      WRITE (15) IR,IS,IT
```

```
        CALL OUTPNT(R,S,T,IR,IS,IT)
      ENDIF
C
 30   IF(IES .NE. 0) THEN
        DO 300 IRR = 1,IR
          RR = R(IRR)
          DO 200 ISS = 1,IS
            SS = S(ISS)
            DO 100 ITT = 1,IT
              TT = T(ITT)
              CALL STRES(EPS,STRESS,IES,MINC,JCODE,D,B,H,DH,
     .                   RR,SS,TT,XJINV,DEPTH,WIDTH,
     .                   XJAC,X,VR,VS,VT,G,DNL,GMOD,EMOD,VVR,VVS,
     .                   VVT)
              WRITE (10) IRR,ISS,ITT,EPS(1),EPS(2),EPS(3),
     .                   STRESS(1),STRESS(2),STRESS(3)
 100        CONTINUE
 200      CONTINUE
 300    CONTINUE
          READ(5,*) IES
          WRITE (9) IES
C
        GO TO 30
      ENDIF
C
C---  READ THE ELEMENT NOS. WHOSE INTERNAL FORCES ARE TO BE ECHOED.
C
        READ(5,*) IEF
        WRITE (11) IEF
C
C...  SET THE PARAMETERS FOR THE NUMBER OF INTEGRATION POINTS.
C      ( IR = NO OF POINTS IN R DIRECTION, IS = NO. OF POINTS IN S
C       DIRECTION, IT = NO. OF POINTS IN T DIRECTION ).
      IR = 3
      IS = 2
      IT = 2
        CALL INTPNT(R,S,T,IR,IS,IT,WTR,WTS,WTT)
 35   IF(IEF .NE. 0) THEN
        DO 40 III = 1,18
          FINT(III) = 0.D0
 40     CONTINUE
C
        DO 600 IRR = 1,IR
          RR = R(IRR)
          WR = WTR(IRR)
          DO 500 ISS = 1,IS
            SS = S(ISS)
            WS = WTS(ISS)
            DO 400 ITT = 1,IT
              TT = S(ITT)
              WT = WTT(ITT)
              CALL STRES(EPS,STRESS,IEF,MINC,JCODE,D,B,H,
     .                   DH,RR,SS,TT,XJINV,DEPTH,
     .                   WIDTH,XJAC,X,VR,VS,VT,G,DNL,GMOD,EMOD,VVR,
     .                   VVS,VVT)
              CALL INTERF(WR,WS,WT,H,DH,RR,SS,TT,DEPTH,WIDTH,XJAC,X,
     .                    VR,VS,VT,IEF,MINC,XJINV,B,G,FINT,STRESS,
     .                    VVR,VVS,VVT)
 400        CONTINUE
 500      CONTINUE
 600    CONTINUE
        WRITE (12) FINT(1),FINT(2),FINT(3),FINT(4),FINT(5),FINT(6)
        WRITE (12) FINT(7),FINT(8),FINT(9),FINT(10),FINT(11),FINT(12)
      WRITE (12) FINT(13),FINT(14),FINT(15),FINT(16),FINT(17),
   .        FINT(18)
      READ(5,*) IEF
      WRITE (11) IEF
      GO TO 35
    ENDIF
C
C--- JOINT NOS. WHOSE JOINT FORCES ARE TO BE ECHOED.
C
      READ(5,*) IP
      WRITE (13) IP
C
    CALL JOINTF(P,NE,WTR,WTS,WTT,H,DH,R,S,T,DEPTH,WIDTH,
   .           XJAC,X,VR,VS,VT,MINC,XJINV,B,G,FINT,STRESS,
   .           IR,IS,IT,MCODE,EPS,JCODE,D,DNL,GMOD,EMOD,
   .           VVR,VVS,VVT)
 50  IF(IP .NE. 0) THEN
      WRITE (14) P(1,IP),P(2,IP),P(3,IP),P(4,IP),P(5,IP),P(6,IP)
        READ(5,*) IP
        WRITE (13) IP
C
      GO TO 50
    ENDIF
C
    IFLAG = IFLAG + 1
    RETURN
    END
C*************************************************************************
C                    RESULT
C*************************************************************************
C
C    IT PRINTS OUT THE RESULTS IN AN ORGANISED MANNER.
C
    SUBROUTINE RESULT(RATIO)
    IMPLICIT REAL*8(A-H,O-Z)
C
    REWIND (3)
    REWIND (4)
    REWIND (9)
    REWIND (10)
    REWIND (11)
    REWIND (12)
    REWIND (13)
    REWIND (14)
    REWIND (15)
C
C--- ECHO THE NORMALIZED LOADING PARAMETER... (QI / QIMAX )
C
    WRITE(6,5) 'NORMALIZED LOADING PARAMETER',RATIO
  5  FORMAT(//T20,A,2X,E16.10)
C
C--- PRINT THE DISPLACEMENTS FOR THE REQUESTED NODES...
C
    READ (3) IND
C
    IF(IND .NE. 0) THEN
      WRITE(6,10) 'N O D A L   D I S P L A C E M E N T S','NODE',
   .         'DISPL 1','DISPL 2','DISPL 3','DISPL 4','DISPL 5',
   .         'DISPL 6'
 10   FORMAT(///T46,A/T46,37('_')//T3,A,T16,A,T37,A,T58,A,T79,A,
   .      T100,A,T121,A)
    ENDIF
C
```

```
 20  IF(IND .NE. 0) THEN
         READ (4) DEF1
         READ (4) DEF2
         READ (4) DEF3
         READ (4) DEF4
         READ (4) DEF5
         READ (4) DEF6
         WRITE(6,30) IND,DEF1,DEF2,DEF3,DEF4,DEF5,DEF6
 30       FORMAT(T3,I3,5X,E16.10,5X,E16.10,5X,E16.10,5X,E16.10,5X,
    .           E16.10,5X,E16.10,5X,E16.10)
       READ (3) IND
       GO TO 20
     ENDIF
C
C--- PRINT THE STRESSES AND STRAINS FOR THE DESIRED ELEMENTS...
C
     READ (9) IES
     IF(IES .NE. 0) THEN
       READ (15) IR,IS,IT
       WRITE(6,40) 'E L E M E N T   O U T P U T','ELEMENT','OUTPUT',
    .           'POINT','STRAIN 1','STRAIN 1-2','STRAIN 1-3',
    .           'STRESS 1','STRESS 1-2','STRESS 1-3',
    .           'NUMBER','R','S','T'
 40  FORMAT(///T51,A/T51,27('_')/T4,A,T13,A,T20,A,T31,A,T47,A,T65,
    .     A,T85,
    .     A,T101,A,T119,A/T4,A,T15,A,T18,A,T21,A)
     ENDIF
 50  IF(IES .NE. 0) THEN
       DO 90 JR = 1,IR
       DO 80 JS = 1,IS
       DO 70 JT = 1,IT
       READ (10) IRR,ISS,ITT,EPS1,EPS2,EPS3,STRES1,STRES2,STRES3
       WRITE(6,60) IES,IRR,ISS,ITT,EPS1,EPS2,EPS3,STRES1,STRES2,STRES3
 60     FORMAT(T4,I4,T15,I1,T18,I1,T21,I1,6(6X,E12.5) )
 70     CONTINUE
 80     CONTINUE
 90     CONTINUE
       READ (9) IES
       GO TO 50
     ENDIF
C
C--- ECHO THE LOCAL INTERNAL ELEMENT FORCES...
C
     READ (11) IEF
     IF(IEF .NE. 0) THEN
       WRITE(6,100) 'INTERNAL ELEMENT FORCES AT LOCAL NODES'
 100    FORMAT(///T46,A/T46,38('_'))
     ENDIF
C
 110 IF(IEF .NE. 0) THEN
       WRITE(6,120) 'ELEMENT NUMBER :',IEF
 120    FORMAT(//T10,A,2X,I4)
       READ (12) FINT1,FINT2,FINT3,FINT4,FINT5,FINT6
       READ (12) FINT7,FINT8,FINT9,FINT10,FINT11,FINT12
       READ (12) FINT13,FINT14,FINT15,
    .          FINT16,FINT17,FINT18
C
       WRITE(6,130) 'COMPONENT 1','COMPONENT 2','COMPONENT 3',
    .          'COMPONENT 4','COMPONENT 5','COMPONENT 6'
 130    FORMAT(/T11,A,T31,A,T51,A,T71,A,T91,A,T111,A)
       WRITE(6,140) FINT1,FINT2,FINT3,FINT4,FINT5,FINT6
 140    FORMAT(T8,E16.10,T28,E16.10,T48,E16.10,T68,E16.10,T88,E16.10,
    .      T108,E16.10)
```

```
         WRITE(6,150) 'COMPONENT 7','COMPONENT 8','COMPONENT 9',
     .             'COMPONENT 10','COMPONENT 11','COMPONENT 12'
 150       FORMAT(/T11,A,T31,A,T51,A,T70,A,T90,A,T110,A)
         WRITE(6,160) FINT7,FINT8,FINT9,FINT10,FINT11,FINT12
 160       FORMAT(T8,E16.10,T28,E16.10,T48,E16.10,T68,E16.10,T88,E16.10,
     .        T108,E16.10)
         WRITE(6,170) 'COMPONENT 13','COMPONENT 14','COMPONENT 15',
     .             'COMPONENT 16','COMPONENT 17','COMPONENT 18'
 170       FORMAT(/T10,A,T30,A,T50,A,T70,A,T90,A,T110,A)
         WRITE(6,180) FINT13,FINT14,FINT15,FINT16,FINT17,FINT18
 180       FORMAT(T8,E16.10,T28,E16.10,T48,E16.10,T68,E16.10,T88,E16.10,
     .        T108,E16.10)
         READ (11) IEF
         GO TO 110
       ENDIF
C
C--- ECHO THE JOINT FORCES FOR THE SPECIFIED NODES...
C
       READ (13) IP
       IF(IP .NE. 0) THEN
         WRITE(6,190) 'EXTERNAL NODAL FORCES (REACTIONS / APPLIED )'
 190       FORMAT(///T44,A/T44,44('_')/)
         WRITE(6,195) 'NODE','COMPONENT 1','COMPONENT 2','COMPONENT 3',
     .             'COMPONENT 4','COMPONENT 5','COMPONENT 6'
 195       FORMAT(T5,A,T16,A,T35,A,T54,A,T73,A,T92,A,T111,A)
       ENDIF
C
 210   IF(IP .NE. 0) THEN
         READ (14) P1,P2,P3,P4,P5,P6
         WRITE(6,220) IP,P1,P2,P3,P4,P5,P6
 220       FORMAT(T5,I4,T14,E16.10,T33,E16.10,T52,E16.10,T71,E16.10,
     .        T90,E16.10,T109,E16.10)
          READ (13) IP
          GO TO 210
       ENDIF
C
       RETURN
       END
C*********************************************************************
C                       INTERF
C*********************************************************************
C
C    IT CALCULATES THE INTERNAL FORCES IN EACH ELEMENT AND ASSEMBLES
C    THE SYSTEM FORCE VECTOR.  IT CALLS 'INTER', 'DIVECT', 'DETJAC',
C    'INVJAC', AND 'NODALB'.
C
       SUBROUTINE INTERF(WR,WS,WT,H,DH,RR,SS,TT,DEPTH,WIDTH,
     .             XJAC,X,VR,VS,VT,IE,MINC,XJINV,B,G,FINT,STRESS,
     .             VVR,VVS,VVT)
       IMPLICIT REAL*8(A-H,O-Z)
       DIMENSION STRESS(*),MINC(3,*),B(3,3,*),H(*),DH(*),
     .       XJINV(3,*),DEPTH(*),WIDTH(*),X(3,*),VR(3,*),
     .       FINT(*),VS(3,*),VT(3,*),G(3,3,*),VVR(*),VVS(*),VVT(*)
C
        CALL INTER(H,DH,RR)
        CALL DIVECT(VR,VS,VT,VVR,VVS,VVT,H,MINC,IE)
           CALL DETJAC(H,DH,RR,SS,TT,DEPTH,WIDTH,VS,VT,XJAC,
     .            X,IE,MINC)
           CALL INVJAC(H,DH,RR,DEPTH,WIDTH,VS,VT,XJAC,X,IE,XJINV,
     .            SS,TT,MINC)
           DO 70 NDOF = 1,6
             DO 60 IEPS = 1,3
               DO 50 KL = 1,3
```

```
                 KG = MINC(KL,IE)
                 CALL NODALB(H,DH,RR,SS,TT,XJINV,DEPTH,WIDTH,
     .                XJAC,X,BB,VR,VS,VT,G,IE,KG,KL,IEPS,NDOF,
     .                VVR,VVS,VVT)
                 B(KL,IEPS,NDOF) = BB
 50            CONTINUE
 60          CONTINUE
 70        CONTINUE
C
C--- ASSEMBLE THE INTERNAL FORCE VECTOR.
C
      DO 80 NDOF = 1,6
           FINT(NDOF) = FINT(NDOF) + ( B(1,1,NDOF) * STRESS(1) +
     .              B(1,2,NDOF) * STRESS(2) + B(1,3,NDOF) *
     .              STRESS(3) ) * XJAC * WR * WS * WT
          FINT(NDOF + 6) = FINT(NDOF + 6) + (B(2,1,NDOF) * STRESS(1)
     .               + B(2,2,NDOF) * STRESS(2) + B(2,3,NDOF) *
     .               STRESS(3) ) * XJAC * WR * WS * WT
           FINT(NDOF + 12) = FINT(NDOF + 12) + (B(3,1,NDOF) *
     .                 STRESS(1) + B(3,2,NDOF) * STRESS(2) +
     .             B(3,3,NDOF) * STRESS(3)) * XJAC * WR * WS * WT
 80      CONTINUE
C
      RETURN
      END
C**********************************************************************
C                     STRES
C**********************************************************************
C
C    CALCULATES THE STRESSES AT THE GIVEN INTERPOLATION POINTS IN THE
C    ELEMENTS. IT CALLS 'INTER', 'DIVECT', 'DETJAC', 'INVJAC', AND
C    'NODALB'.
C
      SUBROUTINE STRES(EPS,STRESS,IE,MINC,JCODE,D,B,H,DH,RR,SS,TT,
     .          XJINV,DEPTH,WIDTH,XJAC,X,VR,VS,VT,G,DNL,GMOD,EMOD,
     .          VVR,VVS,VVT)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION EPS(*),STRESS(*),MINC(3,*),JCODE(6,*),D(*),B(3,3,*),H(*)
     .      ,DH(*),XJINV(3,*),DEPTH(*),WIDTH(*),
     .       X(3,*),VR(3,*),VS(3,*),VT(3,*),G(3,3,*),DNL(6,*),
     .       GMOD(*),EMOD(*),VVR(*),VVS(*),VVT(*)
C
      DO 20 I = 1,3
        EPS(I) = 0.D0
        DO 10 L = 1,6
          DNL(L,I) = 0.D0
 10     CONTINUE
 20   CONTINUE
C
        CALL INTER(H,DH,RR)
           CALL DETJAC(H,DH,RR,SS,TT,DEPTH,WIDTH,VS,VT,XJAC,
     .             X,IE,MINC)
           CALL INVJAC(H,DH,RR,DEPTH,WIDTH,VS,VT,XJAC,X,IE,XJINV,
     .             SS,TT,MINC)
           DO 70 IEPS = 1,3
             DO 60 KL = 1,3
             KG = MINC(KL,IE)
               DO 50 NDOF = 1,6
                 J = JCODE(NDOF,KG)
                 IF(J .NE. 0) THEN
                   DNL(NDOF,KL) = D(J)
                 ELSE
                   DNL(NDOF,KL) = 0.D0
                 ENDIF
                 CALL DIVECT(VR,VS,VT,VVR,VVS,VVT,H,MINC,IE)
                 CALL NODALB(H,DH,RR,SS,TT,XJINV,DEPTH,WIDTH,
     .                XJAC,X,BB,VR,VS,VT,G,IE,KG,KL,IEPS,NDOF,
     .                    VVR,VVS,VVT)
                 B(KL,IEPS,NDOF) = BB
                 EPS(IEPS) = EPS(IEPS) + B(KL,IEPS,NDOF)
     .                   * DNL(NDOF,KL)
                 IF(IEPS .EQ. 1) THEN
                   STRESS(IEPS) = EMOD(IE) * EPS(IEPS)
                 ELSE
                   STRESS(IEPS) = GMOD(IE) * EPS(IEPS)
                 ENDIF
   50           CONTINUE
   60         CONTINUE
   70       CONTINUE
C
      RETURN
      END
C**********************************************************************
C                     JOINTF
C**********************************************************************
C
C     IT CALCULATES THE REACTIONS OR THE EXTERNAL APPLIED LOADS AT
C     THE SPECIFIED NODES.
C     IT CALLS 'STRES' AND 'INTERF'.
C
      SUBROUTINE JOINTF(P,NE,WTR,WTS,WTT,H,DH,R,S,T,DEPTH,WIDTH,
     .           XJAC,X,VR,VS,VT,MINC,XJINV,B,G,FINT,STRESS,
     .           IR,IS,IT,MCODE,EPS,JCODE,D,DNL,GMOD,EMOD,
     .           VVR,VVS,VVT)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION STRESS(*),MINC(3,*),B(3,3,*),H(*),DH(*),R(*),S(*),T(*),
     .      XJINV(3,*),DEPTH(*),WIDTH(*),X(3,*),VR(3,*),P(6,*),
     .      WTR(*),WTS(*),WTT(*),FINT(*),VS(3,*),VT(3,*),G(3,3,*),
     .      MCODE(18,*),EPS(*),JCODE(6,*),D(*),DNL(6,*),GMOD(*),
     .      EMOD(*),VVR(*),VVS(*),VVT(*)
C
      DO 20 J = 1,NJ
        DO 10 L = 1,6
          P(L,J) = 0.D0
   10   CONTINUE
   20 CONTINUE
C
      DO 70 IE = 1,NE
        DO 30 III = 1,18
          FINT(III) = 0.D0
   30   CONTINUE
C
        DO 60 IRR = 1,IR
          RR = R(IRR)
          WR = WTR(IRR)
          DO 50 ISS = 1,IS
            SS = S(ISS)
            WS = WTS(ISS)
            DO 40 ITT = 1,IT
              TT = T(ITT)
              WT = WTT(ITT)
              CALL STRES(EPS,STRESS,IE,MINC,JCODE,D,B,H,DH,
     .               RR,SS,TT,XJINV,DEPTH,WIDTH,
     .               XJAC,X,VR,VS,VT,G,DNL,GMOD,EMOD,VVR,VVS,
     .               VVT)
              CALL INTERF(WR,WS,WT,H,DH,RR,SS,TT,DEPTH,WIDTH,
     .              XJAC,X,VR,VS,VT,IE,MINC,XJINV,B,G,FINT,STRESS,
     .              VVR,VVS,VVT)
 40          CONTINUE
 50        CONTINUE
 60      CONTINUE
        DO 65 NDOF = 1,18
          J = MCODE(NDOF,IE)
C
          IF(NDOF .GE. 1 .AND. NDOF .LE. 6) THEN
            KG = MINC(1,IE)
            P(NDOF,KG) = P(NDOF,KG) + FINT(NDOF)
          ELSEIF(NDOF .GE. 7 .AND. NDOF .LE. 12) THEN
            KG = MINC(2,IE)
            P((NDOF - 6),KG) = P((NDOF - 6),KG) + FINT(NDOF)
          ELSEIF(NDOF .GE. 13 .AND. NDOF .LE. 18) THEN
            KG = MINC(3,IE)
            P((NDOF - 12),KG) = P((NDOF - 12),KG) + FINT(NDOF)
          ENDIF
 65      CONTINUE
 70    CONTINUE
C
      RETURN
      END
C**********************************************************************
C                         DIVECT
C**********************************************************************
C
C     IT CALCULATES THE UNIT VECTORS OF THE CROSS SECTIONS AT THE
C     GIVEN INTERPOLATION POINTS.
C
      SUBROUTINE DIVECT(VR,VS,VT,VVR,VVS,VVT,H,MINC,IE)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION VR(3,*),VS(3,*),VT(3,*),VVR(*),VVS(*),VVT(*),H(*),
     .          MINC(3,*)
C
      DO 10 I = 1,3
        VVR(I) = 0.D0
        VVS(I) = 0.D0
        VVT(I) = 0.D0
 10   CONTINUE
C
      KG1 = MINC(1,IE)
      KG2 = MINC(2,IE)
      KG3 = MINC(3,IE)
C
      DO 20 I = 1,3
        VVR(I) = H(1) * VR(I,KG1) + H(2) * VR(I,KG2) + H(3) * VR(I,KG3)
        VVS(I) = H(1) * VS(I,KG1) + H(2) * VS(I,KG2) + H(3) * VS(I,KG3)
        VVT(I) = H(1) * VT(I,KG1) + H(2) * VT(I,KG2) + H(3) * VT(I,KG3)
 20   CONTINUE
C
      RETURN
      END
C**********************************************************************
C                         NONLIN
C**********************************************************************
C
C     IT IS A CONTROL MODULE FOR NONLINEAR ANALYSIS. IT CALLS 'NEWFIX',
C     'NEWVAR', AND 'RIKWEM'
C
      SUBROUTINE NONLIN(Q,QBAR,QI,QIMAX,DQI,D,DD,R,S,T,EPS,STRESS,
     .             JCODE,B,BNL,H,DH,XJINV,DEPTH,WIDTH,X,VR,VS,VT,
     .             G,DNL,GMOD,EMOD,WTR,WTS,WTT,FINT,P,NE,MCODE,
     .         VVR,VVS,VVT,F,FP,FPI,FORCE,TTT,RES,SSS,MAXA,
     .         NEQ,TYPE,LSS,MINC,DD0,NJ,ALGO,DS1,DS2,DS3,
     .         DD01,DDT,DDP,DD1,DD2,XP,VRP,VSP,VTP,XPP,VRPP,
     .         VSPP,VTPP)
      IMPLICIT REAL*8(A-H,O-Z)
      CHARACTER*(*) TYPE*6,ALGO*6
      DIMENSION Q(*),QBAR(*),MINC(3,*),MAXA(*),SSS(*),JCODE(6,*),R(*),
     .      S(*),T(*),WTR(*),WTS(*),WTT(*),B(3,3,*),EMOD(*),GMOD(*),
     .      H(*),DH(*),DEPTH(*),WIDTH(*),G(3,3,*),VR(3,*),VS(3,*),
     .      VT(3,*),XJINV(3,*),X(3,*),TTT(*),D(*),MCODE(18,*),
     .      STRESS(*),DNL(6,*),FINT(*),P(6,*),EPS(*),VVR(*),VVS(*),
     .      VVT(*),DD(*),BNL(2,2,2,3,9,*),F(*),FP(*),FPI(*),
     .      FORCE(*),DD01(*),DDT(*),DDP(*),DD1(*),DD2(*),
     .      RES(*),DD0(*),DS1(2,2,*),DS2(2,2,*),DS3(2,2,*),
     .      XP(3,*),VRP(3,*),VSP(3,*),VTP(3,*),
     .      XPP(3,*),VRPP(3,*),VSPP(3,*),VTPP(3,*)
      DO 22 IK = 1,NEQ
        FP(IK) = 0.D0
 22   CONTINUE
C
C---  READ THE PARAMETERS FOR THE CONVERGENCE TESTS...
C     TOLENE = ENERGY CONVERGENCE TOLERENCE
C     TOLFOR = FORCE CONVERGENCE TOLERENCE
C     TOLDIS = DISPLACEMENT CONVERGENCE TOLERENCE
C
      READ(5,*) TOLENE,TOLFOR,TOLDIS
C
C---  READ THE TYPE OF ITERATIVE ALGORITHM TO BE ADOPTED FOR ANALYSIS...
C     ALGO = NEWFIX/NEWVAR/RIKWEM
C
      READ(5,'(A)') ALGO
C
C---  READ THE NUMBER OF TIME STEPS AFTER WHICH THE RESULTS ARE TO BE
C     PRINTED...
C
      READ(5,*) IRES
C
C---  READ MAX. ALLOWED ITERATIONS AND NO. OF ITERATIONS BEFORE STIFFNES
C     IS UPDATED...
C
      READ(5,*) MAXIT,ITEUPD
C
C---  WRITE HEADINGS FOR NONLINEAR ANALYSIS
C
      WRITE(6,10) 'N O N L I N E A R   F I N I T E   E L E M E N T',
     .        'A N A L Y S I S','RESULTS PRINTED AT EVERY',IRES,
     .        'LOAD STEP.'
 10   FORMAT(///T32,A,3X,A/T32,65('_')//T10,A,T37,I3,2X,A)
C
C---  ECHO INFORMATION FOR CONVERGENCE CRITERIA...
C
      IF(TOLENE .GT. 1.D0) THEN
        WRITE(6,200) 'ENERGY CRITERION NOT CONSIDERED'
 200    FORMAT(//T20,A)
      ELSE
        WRITE(6,210) 'ENERGY TOLERENCE: ',TOLENE
 210    FORMAT(//T20,A,T40,E16.10)
      ENDIF
C
      IF(TOLFOR .GT. 1.D0) THEN
        WRITE(6,220) 'FORCE CRITERION NOT CONSIDERED'
 220    FORMAT(T20,A)
      ELSE
```

```
        WRITE(6,230) 'FORCE TOLERENCE: ',TOLFOR
 230      FORMAT(T20,A,T40,E16.10)
      ENDIF
C
      IF(TOLDIS .GT. 1.D0) THEN
        WRITE(6,240) 'DISPLACEMENT CRITERION NOT CONSIDERED'
 240      FORMAT(T20,A)
      ELSE
        WRITE(6,250) 'DISPL. TOLERENCE: ',TOLDIS
 250      FORMAT(T20,A,T40,E16.10)
      ENDIF
C
C--- ECHO INFORMATION FOR STIFFNESS UPDATING...
C
      WRITE(6,260) 'STIFFNESS UPDATED WITHIN',ITEUPD,'ITERATIONS.'
 260  FORMAT(//T20,A,2X,I2,1X,A)
C
      REWIND (3)
      READ(5,*) IND
      WRITE (3) IND
C
C--- READ AND STORE NODE NOS. WHOSE OUTPUT ARE REQUIRED
C
   30 IF (IND .NE. 0) THEN
        READ (5,*) IND
        WRITE (3) IND
        GO TO 30
      END IF
C
      REWIND (9)
      REWIND (15)
      REWIND (10)
C
      READ (5,*) IES
      WRITE (9) IES
      IR=3
      IS=3
      IT=3
C
C-------------------- STRESSES ------------------------
C
      WRITE (15) IR,IS,IT
   40 IF(IES .NE. 0) THEN
        DO 70 IRR = 1,IR
          DO 60 ISS = 1, IS
            DO 50 ITT = 1,IT
              WRITE (10) IRR,ISS,ITT,0.D0,0.D0,0.D0,0.D0,0.D0,0.D0
   50       CONTINUE
   60     CONTINUE
   70   CONTINUE
        READ (5,*) IES
        WRITE (9) IES
        GO TO 40
      ENDIF
C
C------------------ INTERNAL FORCES ------------------
C
      REWIND (11)
      READ (5,*) IEF
      WRITE (11) IEF
   80 IF (IEF .NE. 0) THEN
        READ (5,*) IEF
        WRITE (11) IEF
```

```
       GO TO 80
     ENDIF
C
C------------------ NODAL FORCES ----------------------
C
     READ (5,*) IP
     WRITE (13) IP
  90 IF (IP .NE. 0) THEN
       READ (5,*) IP
       WRITE (13) IP
       GO TO 90
     ENDIF
C
     REWIND (20)
C
     DO 130 IE = 1,NE
       DO 120 IRR = 1,2
         DO 110 ISS = 1,2
           DO 100 ITT = 1,2
             WRITE (20) 0.D0, 0.D0, 0.D0
 100         CONTINUE
 110       CONTINUE
 120     CONTINUE
 130 CONTINUE
C
     REWIND (23)
C
     DO 170 IE = 1,NE
       DO 160 IRR = 1,2
         DO 150 ISS = 1,2
           DO 140 ITT = 1,2
             WRITE (23) 0.D0,0.D0,0.D0
 140         CONTINUE
 150       CONTINUE
 160     CONTINUE
 170 CONTINUE
C
C--  CALL NEWTON RALPHSON METHOD WITH FIXED TIME INTERVALS IF THE ALGO
C    SPECIFIED IS NEWFIX
C
     IF(ALGO .EQ. 'NEWFIX') THEN
        WRITE(6,20) 'ALGORITHM USED:','NEWTON RALPHSON METHOD (FIXED)'
     WRITE(6,270)'MAXIMUM NO. OF ITERATIONS ALLOWED IN EACH TIME STEP:'
    .           ,MAXIT
 270    FORMAT(T10,A,2X,I2)
  20    FORMAT(T10,A,T37,A)
        CALL NEWFIX(Q,QBAR,QI,QIMAX,DQI,D,DD,R,S,T,EPS,STRESS,MINC,
    .               JCODE,B,BNL,H,DH,XJINV,DEPTH,WIDTH,X,VR,VS,VT,
    .               G,DNL,GMOD,EMOD,WTR,WTS,WTT,FINT,P,NE,MCODE,
    .               VVR,VVS,VVT,IRES,TOLENE,TOLFOR,TOLDIS,F,FP,FPI,
    .               FORCE,TTT,RES,SSS,MAXA,NEQ,TYPE,LSS,
    .               DD0,NJ,MAXIT,ITEUPD,DS1,DS2,DS3,
    .               XP,VRP,VSP,VTP,ALGO,DD2)
C
C--  CALL NEWTON RALPHSON METHOD WITH VARIABLE TIME INTERVALS IF THE
C    ALGO SPECIFIED IS NEWVAR
C
     ELSEIF(ALGO .EQ. 'NEWVAR') THEN
      WRITE(6,43) 'ALGORITHM USED:','NEWTON RALPHSON METHOD (VARIABLE)'
 43    FORMAT(T10,A,T37,A)
      READ(5,*) DQIMAX,DQIMIN,DTOL
     WRITE(6,53)'MAXIMUM NO. OF ITERATIONS ALLOWED IN EACH TIME STEP:'
    .          ,MAXIT
 53     FORMAT(T10,A,2X,I2)
      WRITE(6,63) 'MAX SIZE OF TIME STEP INCREMENT: ',DQIMAX,
     .        'MIN SIZE OF TIME STEP INCREMENT: ',DQIMIN,
     .        'TOLERANCE FOR DIVERGENCE TEST:   ',DTOL
 63   FORMAT(/T10,A,2X,D16.10/T10,A,2X,D16.10/T10,A,D16.10)
C
      CALL NEWVAR(Q,QBAR,QI,QIMAX,DQI,D,DD,R,S,T,EPS,STRESS,MINC,
     .            JCODE,B,BNL,H,DH,XJINV,DEPTH,WIDTH,X,VR,VS,VT,
     .            G,DNL,GMOD,EMOD,WTR,WTS,WTT,FINT,P,NE,MCODE,
     .            VVR,VVS,VVT,IRES,TOLENE,TOLFOR,TOLDIS,F,FP,FPI,
     .            FORCE,TTT,RES,SSS,MAXA,NEQ,TYPE,LSS,
     .            DD0,NJ,MAXIT,ITEUPD,DS1,DS2,DS3,XP,VRP,VSP,VTP,
     .            ALGO,DD2,XPP,VRPP,VSPP,VTPP,DD1,DQIMAX,DQIMIN,
     .            DTOL)
C
C--- USE MODIFIED RIKS WEMPNER METHOD IF THE ALGO SPECIFIED IS RIKWEM
C
      ELSEIF(ALGO .EQ. 'RIKWEM') THEN
C
C--- SELECT THE DESIRED NO. OF ITERATIONS IN THE MODIFIED RIKS WEMPNER
C    METHOD FOR EACH TIME STEP (ITEDES)..
C
      ITEDES = 5
C
        WRITE(6,35) 'ALGORITHM USED:','MODIFIED RIKS-WEMPNER METHOD'
 35     FORMAT(T10,A,T37,A)
      WRITE(6,470)'MAXIMUM NO. OF ITERATIONS ALLOWED IN EACH TIME STEP:'
     .        ,MAXIT
 470    FORMAT(T10,A,2X,I2)
        WRITE(6,480) 'DESIRED NO. OF ITERATIONS IN EACH TIME STEP: ',
     .            ITEDES
 480    FORMAT(T10,A,4X,I2)
        CALL RIKWEM(D,DD,DD0,F,FP,FPI,ICONV,NEQ,NJ,Q,TOLENE,TOLFOR,
     .            TOLDIS,QI,QIMAX,DQI,FORCE,VR,VS,VT,EPS,STRESS,
     .            MINC,JCODE,B,XJINV,DEPTH,WIDTH,X,G,DNL,GMOD,
     .            EMOD,VVR,VVS,VVT,FINT,MCODE,RES,QBAR,DD01,
     .            H,DH,NE,WTR,WTS,WTT,R,S,T,P,DDT,DDP,DD1,DD2,
     .            TTT,SSS,MAXA,TYPE,LSS,BNL,DS1,DS2,DS3,ITEUPD,
     .            IRES,ITEDES,MAXIT,XP,VRP,VSP,VTP)
      ENDIF
C
      RETURN
      END
C**********************************************************************
C*                     NEWFIX                          *
C**********************************************************************
C
C    IT IS A CONTROL MODULE FOR NONLINEAR ANALYSIS BY NEWTON RALPHSON
C    METHOD BY USING FIXED TIME INCREMENTS. IT CALLS 'NEWRAP','POSTN',
C    AND 'RESULT'
C
      SUBROUTINE NEWFIX(Q,QBAR,QI,QIMAX,DQI,D,DD,R,S,T,EPS,STRESS,MINC,
     .            JCODE,B,BNL,H,DH,XJINV,DEPTH,WIDTH,X,VR,VS,VT,
     .            G,DNL,GMOD,EMOD,WTR,WTS,WTT,FINT,P,NE,MCODE,
     .            VVR,VVS,VVT,IRES,TOLENE,TOLFOR,TOLDIS,F,FP,FPI,
     .            FORCE,TTT,RES,SSS,MAXA,NEQ,TYPE,LSS,
     .            DD0,NJ,MAXIT,ITEUPD,DS1,DS2,DS3,XP,VRP,VSP,VTP,
     .            ALGO,RP)
      IMPLICIT REAL*8(A-H,O-Z)
      CHARACTER*(*) TYPE*6,ALGO*6
C
      DIMENSION Q(*),QBAR(*),D(*),DD(*),R(*),S(*),T(*),EPS(*),STRESS(*),
     .        MINC(3,*),JCODE(6,*),B(3,3,*),BNL(2,2,2,3,9,*),H(*),
     .         DEPTH(*),WIDTH(*),X(3,*),VR(3,*),VS(3,*),VT(3,*),
     .         G(3,3,*),DNL(6,*),GMOD(*),EMOD(*),WTR(*),WTS(*),WTT(*),
     .         FINT(*),P(6,*),MCODE(18,*),VVR(*),VVS(*),VVT(*),DH(*),
     .         F(*),FP(*),FPI(*),FORCE(*),TTT(*),RES(*),XJINV(3,*),
     .         SSS(*),MAXA(*),DD0(*),DS1(2,2,*),DS2(2,2,*),DS3(2,2,*),
     .         XP(3,*),VRP(3,*),VSP(3,*),VTP(3,*),RP(*)
C
      IFLAG = 0
      ISTEP = 0
      ICONV = 0
C
    5 IF(QI .LE. QIMAX .AND. IFLAG .LE. 1) THEN
         DO 10 I = 1,NEQ
            Q(I) = QI*QBAR(I)
   10    CONTINUE
C
C--- CALL NEWTON RALPHSON METHOD
C
         CALL NEWRAP(SSS,TTT,MAXA,NEQ,NE,MINC,R,S,T,WTR,WTS,WTT,TYPE,
     .               JCODE,LSS,B,EMOD,GMOD,H,DH,VR,VS,VT,DEPTH,WIDTH,
     .               G,XJINV,X,VVR,VVS,VVT,FORCE,MCODE,FINT,RES,Q,
     .               EPS,STRESS,D,DNL,DD,DD0,F,FP,FPI,ICONV,NJ,
     .               TOLDIS,TOLENE,TOLFOR,MAXIT,ITEUPD,BNL,P,DS1,
     .               DS2,DS3,XP,VRP,VSP,VTP,IDIVER,ITECNT,ALGO,RP,
     .               1.D0)
         IF(ICONV .NE. 0) THEN
            WRITE(6,20) '******ERROR******','SOLUTION FAILS TO CONVERGE'
     .          ,'IN GIVEN NUMBER OF ITERATIONS'
   20       FORMAT(///T30,A//T12,A,1X,A)
            STOP
         ELSE
            ISTEP = ISTEP + 1
            IF(ISTEP .EQ. IRES .OR. QI .GE. QIMAX) THEN
C
C--- PERFORM POSTPROCESSING.
C
               CALL POSTN(JCODE,D,P)
C
               RATIO = QI / QIMAX
C
C--- PRINT THE RESULTS AT THE PRESENT TIME INTERVAL.
C
               CALL RESULT(RATIO)
               ISTEP = 0
            ENDIF
C
            DO 30 I = 1,NEQ
               FP(I) = FORCE(I)
   30       CONTINUE
         ENDIF
C
C--- INCREMENT VARIABLE LOAD PROPORTIONALITY FACTOR
C
         QI = QI + DQI
         IF(QI .GT. QIMAX) THEN
            IFLAG = IFLAG + 1
            QI = QIMAX
         ENDIF
C
         GO TO 5
      ENDIF
C
      RETURN
      END
C*********************************************************************
C                         NEWRAP                           *
C*********************************************************************
C
C    IT PERFORMS NEWTON RALPHSON METHOD ON A NONLINEAR SYSTEM UNTIL THE
C    CONVERGENCE CRITERIA ARE MET OR THE MAXIMUM NUMBER OF ITERATIONS
C    ARE SURPASSED. IT CALLS 'STIFF', 'SOLVE', 'PRESER', 'UPDATC',
C    'FORCES', 'UPDATS', 'DVTEST', AND 'TEST'.
C
      SUBROUTINE NEWRAP(SSS,TTT,MAXA,NEQ,NE,MINC,R,S,T,WTR,WTS,WTT,TYPE,
     .           JCODE,LSS,B,EMOD,GMOD,H,DH,VR,VS,VT,DEPTH,WIDTH,
     .           G,XJINV,X,VVR,VVS,VVT,FORCE,MCODE,FINT,RES,Q,
     .           EPS,STRESS,D,DNL,DD,DD0,F,FP,FPI,ICONV,NJ,
     .           TOLDIS,TOLENE,TOLFOR,MAXIT,ITEUPD,BNL,P,DS1,
     .           DS2,DS3,XP,VRP,VSP,VTP,IDIVER,ITECNT,ALGO,RP,
     .           DTOL)
C
      IMPLICIT REAL*8(A-H,O-Z)
      CHARACTER*(*) TYPE*6,ALGO*6
C
      DIMENSION SSS(*),TTT(*),MAXA(*),MINC(3,*),R(*),S(*),T(*),WTR(*),
     .      WTS(*),WTT(*),JCODE(6,*),B(3,3,*),EMOD(*),GMOD(*),H(*),
     .      DH(*),VR(3,*),VS(3,*),VT(3,*),DEPTH(*),WIDTH(*),DD(*),
     .      G(3,3,*),XJINV(3,*),X(3,*),VVR(*),VVS(*),VVT(*),FINT(*),
     .      FORCE(*),MCODE(18,*),RES(*),EPS(*),STRESS(*),D(*),
     .      DD0(*),F(*),FP(*),FPI(*),DNL(6,*),Q(*),BNL(2,2,2,3,9,*),
     .      P(6,*),DS1(2,2,*),DS2(2,2,*),DS3(2,2,*),XP(3,*),
     .      VRP(3,*),VSP(3,*),VTP(3,*),RP(*)
C
      DO 10 I = 1,NEQ
        TTT(I) = 0.D0
        RES(I) = Q(I) - FP(I)
   10 CONTINUE
      ITENUM = ITEUPD
      ITECNT = 1
      ICONV = 1
C
   15 IF(ICONV .NE. 0 .AND. ITECNT .LE. MAXIT) THEN
C
C---  UPDATE STIFFNESS MATRIX.
C
        IF(ITENUM .GE. ITEUPD) THEN
          CALL STIFF (NE,MINC,R,S,T,WTR,WTS,WTT,TYPE,JCODE,SSS,MAXA,
     .             LSS,B,EMOD,GMOD,H,DH,VR,VS,VT,DEPTH,WIDTH,G,
     .             XJINV,X,VVR,VVS,VVT,BNL,STRESS,DS1,DS2,DS3)
          ITENUM = 0
        ENDIF
C
        DO 50 I = 1,NEQ
          TTT(I) = 0.D0
          TTT(I) = RES(I)
   50   CONTINUE
        IF(ITENUM .EQ. 0) THEN
        CALL SOLVE(SSS,TTT,MAXA,NEQ,1)
        ELSE
        CALL SOLVE(SSS,TTT,MAXA,NEQ,2)
        ENDIF
        DO 60 I = 1,NEQ
          DD(I) = TTT(I)
          TTT(I) = 0.D0
   60   CONTINUE
        DO 80 I = 1,NEQ
        D(I) = D(I) + DD(I)
  80   CONTINUE
C
C--- SAVE THE PREVIOUS CONFIGURATION.
C
       CALL PRESER(X,VR,VS,VT,XP,VRP,VSP,VTP,NJ)
C
C--- UPDATE COORDINATES
C
       CALL UPDATC(X,D,VR,VS,VT,DD,NEQ,NJ,NE,MINC,JCODE)
C
     IF(ITECNT .LE. 1) THEN
       DO 85 IJ = 1,NEQ
         DD0(IJ) = DD(IJ)
  85   CONTINUE
     ENDIF
C
       DO 90 I = 1,NEQ
         IF(ITECNT .NE. 0) THEN
           FPI(I) = FORCE(I)
         ELSE
           FPI(I) = FP(I)
         ENDIF
  90   CONTINUE
C
     DO 95 IK = 1,NEQ
       RP(I) = RES(I)
  95 CONTINUE
     CALL FORCES(NJ,P,FORCE,NEQ,R,S,T,WTR,WTS,WTT,NE,H,DH,EPS,
    .            STRESS,MINC,JCODE,DD,B,XJINV,DEPTH,WIDTH,X,
    .            VR,VS,VT,G,DNL,GMOD,EMOD,VVR,VVS,
    .            VVT,FINT,MCODE,RES,Q,XP,VRP,VSP,VTP)
C
C--- UPDATE STRESSES
C
     CALL UPDATS(EPS,STRESS,MINC,JCODE,DD,B,H,DH,R,S,T,WTR,WTS,
    .           WTT,XJINV,DEPTH,WIDTH,X,VR,VS,VT,G,DNL,GMOD,
    .           EMOD,VVR,VVS,VVT,XP,VRP,VSP,VTP)
       DO 100 I = 1,NEQ
         F(I) = FORCE(I)
 100   CONTINUE
C
C
       IF(ITECNT .GE. 4 .AND. ALGO .EQ. 'NEWVAR') THEN
         CALL DVTEST(DTOL,IDIVER,ITECNT,MAXIT,NEQ,NTRY,RES,RP)
         IF(IDIVER .GE. 1) THEN
           RETURN
         ENDIF
       ELSE
         IDIVER = 0
       ENDIF
C
C--- TEST THE SOLUTION FOR THE CONVERGENCE CRITERIA.
C
       CALL TEST(D,DD,DD0,F,FP,FPI,ICONV,JCODE,NEQ,NJ,Q,TOLDIS
    .       ,TOLENE,TOLFOR)
       ITENUM = ITENUM + 1
       ITECNT = ITECNT + 1
       GO TO 15
     ENDIF
C
     RETURN
```

```
      END
C**********************************************************************
C                        FORCES
C**********************************************************************
C
C     IT IS A CONTROL MODULE FOR DETERMINING INTERNAL FORCES IN THE
C     GLOBAL DEGREES OF FREEDOM FOR EACH MEMBFR AND CALCULATE THE
C     RESIDUAL FORCE VECTOR. IT CALLS 'INTPNT', 'INTER', 'STRES', AND
C     'INTERF'.
C
      SUBROUTINE FORCES(NJ,P,FORCE,NEQ,R,S,T,WTR,WTS,WTT,NE,H,DH,EPS,
     .           STRESS,MINC,JCODE,DD,B,XJINV,DEPTH,WIDTH,X,
     .           VR,VS,VT,G,DNL,GMOD,EMOD,VVR,VVS,
     .           VVT,FINT,MCODE,RES,Q,XP,VRP,VSP,VTP)
      IMPLICIT REAL*8(A-H,O-Z)
C
      DIMENSION P(6,*),FORCE(*),R(*),S(*),T(*),WTR(*),WTS(*),WTT(*),
     .      H(*),DH(*),EPS(*),STRESS(*),MINC(3,*),JCODE(6,*),DD(*),
     .      B(3,3,*),XJINV(3,*),DEPTH(*),WIDTH(*),X(3,*),FINT(*),
     .      VR(3,*),VS(3,*),VT(3,*),G(3,3,*),DNL(6,*),MCODE(18,*),
     .      EMOD(*),VVR(*),VVS(*),VVT(*),RES(*),GMOD(*),Q(*),
     .      XP(3,*),VRP(3,*),VSP(3,*),VTP(3,*)
C
      REWIND (11)
      REWIND (12)
      REWIND (20)
      REWIND (21)
C
      DO 20 J = 1,NJ
        DO 10 I = 1,6
          P(I,J) = 0.D0
 10     CONTINUE
 20   CONTINUE
C
      DO 30 I = 1,NEQ
        FORCE(I) = 0.D0
 30   CONTINUE
C
      IR = 2
      IS = 2
      IT = 2
      CALL INTPNT(R,S,T,IR,IS,IT,WTR,WTS,WTT)
C
      DO 70 IE = 1,NE
        DO 60 IRR = 1,IR
          RR = R(IRR)
          WR = WTR(IRR)
          CALL INTER(H,DH,RR)
          DO 50 ISS = 1,IS
            SS = S(ISS)
            WS = WTS(ISS)
            DO 40 ITT = 1,IT
              TT = T(ITT)
              WT = WTT(ITT)
              CALL STRES(EPS,STRESS,IE,MINC,JCODE,DD,B,H,DH,RR,SS,TT
     .              ,XJINV,DEPTH,WIDTH,XJAC,XP,VRP,VSP,VTP,G,
     .              DNL,GMOD,EMOD,VVR,VVS,VVT)
C
              READ (20) STP1,STP2,STP3
              STRESS(1) = STRESS(1) + STP1
              STRESS(2) = STRESS(2) + STP2
              STRESS(3) = STRESS(3) + STP3
          WRITE (21) STRESS(1),STRESS(2),STRESS(3)
C
 40         CONTINUE
 50        CONTINUE
 60       CONTINUE
 70      CONTINUE
C
      REWIND (20)
      REWIND (21)
C
      DO 140 IE = 1,NE
       DO 80 KK = 1,18
        FINT(KK) = 0.D0
 80      CONTINUE
       DO 110 IRR = 1,IR
        RR = R(IRR)
        WR = WTR(IRR)
        DO 100 ISS = 1,IS
         SS = S(ISS)
         WS = WTS(ISS)
         DO 90 ITT = 1,IT
          TT = T(ITT)
          WT = WTT(ITT)
          READ (21) STRESS(1),STRESS(2),STRESS(3)
          WRITE (20) STRESS(1),STRESS(2),STRESS(3)
C
          CALL INTERF(WR,WS,WT,H,DH,RR,SS,TT,DEPTH,WIDTH,
     .              XJAC,X,VR,VS,VT,IE,MINC,XJINV,B,G,FINT,
     .              STRESS,VVR,VVS,VVT)
 90         CONTINUE
100        CONTINUE
110       CONTINUE
C
       READ (11) IEF
120     IF(IEF .NE. 0) THEN
        IF(IEF .EQ. IE) THEN
         WRITE (12) FINT(1),FINT(2),FINT(3),FINT(4),FINT(5),
     .   FINT(6)
         WRITE (12) FINT(7),FINT(8),FINT(9),FINT(10),FINT(11),
     .   FINT(12)
         WRITE (12) FINT(13),FINT(14),FINT(15),FINT(16),FINT(17),
     .   FINT(18)
C
        ENDIF
        READ (11) IEF
        GO TO 120
       ELSE
        REWIND (11)
       ENDIF
C
       DO 130 NDOF = 1,18
        J = MCODE(NDOF,IE)
        IF(J .NE. 0) THEN
         FORCE(J) = FORCE(J) + FINT(NDOF)
        ENDIF
C
        IF(NDOF .GE. 1 .AND. NDOF .LE. 6) THEN
         KG = MINC(1,IE)
         IDOF = NDOF
        ELSEIF(NDOF .GE. 7 .AND. NDOF .LE. 12) THEN
         KG = MINC(2,IE)
         IDOF = NDOF - 6
        ELSEIF(NDOF .GE. 13 .AND. NDOF .LE. 18) THEN
```

```
            KG = MINC(3,IE)
            IDOF = NDOF - 12
          ENDIF
C
          P(IDOF,KG) = P(IDOF,KG) + FINT(NDOF)
C
 130     CONTINUE
 140   CONTINUE
C
      DO 150 I = 1,NEQ
        RES(I) = Q(I) - FORCE(I)
 150   CONTINUE
C
      RETURN
      END
C***********************************************************************
C*                     UPDATC                          *
C***********************************************************************
C
C    IT UPDATES THE COORDINATES AND NODAL VECTORS OF THE SYSTEM
C    AFTER EACH TIME INCREMENT. IT CALLS 'UNIVEC'.
C
      SUBROUTINE UPDATC(X,D,VR,VS,VT,DD,NEQ,NJ,NE,MINC,JCODE)
      IMPLICIT REAL*8(A-H,O-Z)
C
      DIMENSION DD(*),JCODE(6,*),X(3,*),D(*),VR(3,*),VS(3,*),VT(3,*),
     .      MINC(3,*)
C
      DO 20 J = 1,NJ
        DO 10 L = 1,6
          K = JCODE(L,J)
          IF(K .NE. 0) THEN
            X(L,J) = X(L,J) + DD(K)
          ENDIF
  10     CONTINUE
  20   CONTINUE
C
C--- SET THE VALUE BY WHICH THE INCREMENTAL ROTATIONS (NODAL) ARE TO
C    BE DIVIDED IN THE EULER'S FORWARD INTEGRATION, TO UPDATE THE
C    INCREMENTAL NODAL VECTORS. A VALUE OF 5 IS SUGGESTED.
C    SET THE VALUE IN NDIV
C
      NDIV = 5
      DIV = DFLOAT (NDIV)
      DO 40 JNUM = 1,NJ
        K = JCODE(4,JUNM)
        TH1 = DD(K) / DIV
        K = JCODE(5,JUNM)
        TH2 = DD(K) / DIV
        K = JCODE(6,JUNM)
        TH3 = DD(K) / DIV
        DO 30 KK = 1,NDIV
          DUM1 = VS(1,JNUM) + TH2 * VS(3,JNUM) - TH3 * VS(2,JNUM)
          DUM2 = VS(2,JNUM) + TH3 * VS(1,JNUM) - TH1 * VS(3,JNUM)
          DUM3 = VS(3,JNUM) + TH1 * VS(2,JNUM) - TH2 * VS(1,JNUM)
C
          VMAG = DSQRT (DUM1 * DUM1 + DUM2 * DUM2 + DUM3 * DUM3)
          VS(1,JNUM) = DUM1 / VMAG
          VS(2,JNUM) = DUM2 / VMAG
          VS(3,JNUM) = DUM3 / VMAG
C
          DUM1 = VT(1,JNUM) + TH2 * VT(3,JNUM) - TH3 * VT(2,JNUM)
          DUM2 = VT(2,JNUM) + TH3 * VT(1,JNUM) - TH1 * VT(3,JNUM)
        DUM3 = VT(3,JNUM) + TH1 * VT(2,JNUM) - TH2 * VT(1,JNUM)
C
        VMAG = DSQRT (DUM1 * DUM1 + DUM2 * DUM2 + DUM3 * DUM3)
        VT(1,JNUM) = DUM1 / VMAG
        VT(2,JNUM) = DUM2 / VMAG
        VT(3,JNUM) = DUM3 / VMAG
C
 30    CONTINUE
C
C--- PERFORM CROSS PRODUCT TO OBTAIN VR
C
      CALL UNIVEC(VR,VS,VT,JNUM)
 40 CONTINUE
C
     RETURN
     END
C***********************************************************************
C                       UPDATS
C***********************************************************************
C
C    UPDATE THE STRESSES IN THE ELEMENTS AFTER EACH TIME INTERVAL.
C    IT CALLS 'STRES', 'INTPNT', 'OUTPNT'.
C
     SUBROUTINE UPDATS(EPS,STRESS,MINC,JCODE,DD,B,H,DH,R,S,T,WTR,WTS,
    .           WTT,XJINV,DEPTH,WIDTH,X,VR,VS,VT,G,DNL,GMOD,
    .           EMOD,VVR,VVS,VVT,XP,VRP,VSP,VTP)
C
     IMPLICIT REAL*8(A-H,O-Z)
     DIMENSION R(*),S(*),T(*),EPS(*),STRESS(*),MINC(3,*),JCODE(6,*),
    .      DD(*),B(3,3,*),H(*),DH(*),XJINV(3,*),DEPTH(*),WIDTH(*),
    .      X(3,*),VR(3,*),VS(3,*),VT(3,*),G(3,3,*),DNL(6,*),
    .      GMOD(*),EMOD(*),WTR(*),WTS(*),WTT(*),XP(3,*),VRP(3,*),
    .      VVR(*),VVS(*),VVT(*),VSP(3,*),VTP(3,*)
C
     REWIND (9)
     REWIND (10)
     REWIND (22)
     REWIND (15)
C
C--- READ THE ELEMENTS  WHOSE STRESSES & STRAINS ARE TO BE UPDATED.
C
     READ (9) IES
C
     IF(IES .NE. 0) THEN
       READ (15) IR,IS,IT
       CALL OUTPNT(R,S,T,IR,IS,IT)
     ENDIF
C
  10 IF(IES .NE. 0) THEN
       DO 40 IRR = 1,IR
         RR = R(IRR)
         DO 30 ISS = 1,IS
           SS = S(ISS)
           DO 20 ITT = 1,IT
             TT = T(ITT)
             CALL STRES(EPS,STRESS,IES,MINC,JCODE,DD,B,H,DH,RR,SS,
    .           TT,XJINV,DEPTH,WIDTH,XJAC,XP,VRP,VSP,VTP,G,DNL,
    .           GMOD,EMOD,VVR,VVS,VVT)
             READ (10) III,JJJ,KKK,E1,E2,E3,S1,S2,S3
             EPS(1) = EPS(1) + E1
             EPS(2) = EPS(2) + E2
             EPS(3) = EPS(3) + E3
C
```

```
              STRESS(1) = STRESS(1) + S1
              STRESS(2) = STRESS(2) + S2
              STRESS(3) = STRESS(3) + S3
C
              WRITE (22) IRR,ISS,ITT,EPS(1),EPS(2),EPS(3),STRESS(1),
     .                STRESS(2),STRESS(3)
 20         CONTINUE
 30       CONTINUE
 40     CONTINUE
        READ (9) IES
        GO TO 10
      ENDIF
C
C--- TRANSFER THE DATA FROM UNIT 22 TO UNIT 10 FOR FUTURE USE...
C
      REWIND (9)
      REWIND (10)
      REWIND (22)
C
      READ (9) IES
C
 50   IF(IES .NE. 0) THEN
        DO 80 IRR = 1,IR
          DO 70 ISS = 1,IS
            DO 60 ITT = 1,IT
              READ (22) I,J,K,E1,E2,E3,S1,S2,S3
              WRITE (10) I,J,K,E1,E2,E3,S1,S2,S3
 60         CONTINUE
 70       CONTINUE
 80     CONTINUE
        READ (9) IES
        GO TO 50
      ENDIF
C
C--- COMPUTE AND STORE THE NEW STRESS VALUES AT THE 2 X 2 X 2 GAUSS
C    POINTS FOR THE REDUCED INTEGRATION TO CALCULATE THE GEOMETRIC
C    STIFNESS MATRIX...
C
      REWIND (23)
      REWIND (24)
C
      IR = 2
      IS = 2
      IT = 2
C
      CALL INTPNT(R,S,T,IR,IS,IT,WTR,WTS,WTT)
C
          DO 120 IE = 1,NE
            DO 110 IRR = 1,IR
              RR = R(IRR)
              WR = WTR(IRR)
              DO 100 ISS = 1,IS
                SS = T(ISS)
                WS = WTS(ISS)
                DO 90 ITT = 1,IT
                  TT = T(ITT)
                  WT = WTT(ITT)
                  CALL STRES(EPS,STRESS,IE,MINC,JCODE,DD,B,
     .               H,DH,RR,SS,TT,XJINV,DEPTH,WIDTH,
     .               XJAC,XP,VRP,VSP,VTP,G,DNL,GMOD,EMOD,VVR,VVS,
     .               VVT)
C
                  READ (23) S1,S2,S3
                STRESS(1) = STRESS(1) + S1
                STRESS(2) = STRESS(2) + S2
                STRESS(3) = STRESS(3) + S3
                WRITE (24) STRESS(1),STRESS(2),STRESS(3)
 90           CONTINUE
100         CONTINUE
110       CONTINUE
120     CONTINUE
C
C--- TRANSFER DATA FROM UNIT 24 TO 23 FOR FUTURE USE...
C
      DO 160 IE = 1,NE
        DO 150 IRR = 1,IR
          DO 140 ISS = 1,IS
            DO 130 ITT = 1,IT
              READ (24) S1,S2,S3
              WRITE (23) S1,S2,S3
130         CONTINUE
140       CONTINUE
150     CONTINUE
160   CONTINUE
C
      RETURN
      END
C ******************************************************************
C                         TEST                              *
C*******************************************************************
C
C     PERFORM CONVERGENCE TESTS, ASSUME THAT CONVERGENCE IS REACHED,
C     ( SETTING INCONV=0 ) UNTIL IT IS PROVED THE CONTRARY. CALL
C     'ENERGY', 'UNBALF', 'DISPL'.
C
      SUBROUTINE TEST(D,DD,DD0,F,FP,FPI,INCONV,JCODE,NEQ,NJ,
     .          Q,TOLDIS,TOLENE,TOLFOR)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION  D(*),DD(*),DD0(*),F(*),FP(*),FPI(*),JCODE(6,*),Q(*)
C
      INCONV = 0
      IF (TOLENE.LE.1.D0) THEN
        CALL ENERGY (DD,DD0,FP,FPI,INCONV,NEQ,Q,TOLENE)
      END IF
C
      IF (TOLFOR.LT.1.D0) THEN
        CALL UNBALF (F,FP,INCONV,NEQ,Q,TOLFOR,FC)
      END IF
C
      IF (TOLDIS.LT.1.D0) THEN
        CALL DISPL(D,DD,INCONV,NEQ,TOLDIS)
      END IF
C
      RETURN
      END
C ******************************************************************
C                        ENERGY                             *
C*******************************************************************
C
C     PERFORM THE ENERGY CONVERGENCE TEST. IF DIVER IS DIFFERENT FROM
C     ZERO, TEST DIVERGENCE.
C
C     TEST FOR DIVERGENCE IN ENERGY CAN ALSO BE MADE BY PUTTING THE
C     PARAMETER DIVER = 1.D0  IF THIS TEST OF DIVERGENCE IN ENERGY IS NOT
C     REQUIRED KEEP DIVER = 0.D0
C
```

```
      SUBROUTINE ENERGY (DD,DD0,FP,FPI,INCONV,NEQ,Q,TOLENE)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION DD(*),DD0(*),FP(*),FPI(*),Q(*)
C
      WORKP=0.D0
      WORKI=0.D0
      DIVER = 1.D0
C
C     INTERNAL ENERGY ..............................................
      DO 10 I=1,NEQ
        WORKI= WORKI + DD(I)*(Q(I)-FPI(I))
        WORKP= WORKP + DD0(I)*(Q(I)-FP(I))
   10 CONTINUE
      IF ( DIVER.NE.0 ) THEN
C
C        TEST FOR DIVERGENCE ........................................
         IF ( WORKI.GT.DIVER*WORKP ) THEN
           WRITE(6,100)
           STOP
         END IF
      END IF
C
C     CHECK WITH TOLERANCES ........................................
      IF ( TOLENE.LT.1.D00 ) THEN
        IF ( WORKI.GT.TOLENE*WORKP ) THEN
          INCONV = INCONV + 1000
        END IF
      END IF
C
  100 FORMAT(/T10,'***** ERROR: DIVERGENCE IN ENERGY *****')
C
      RETURN
      END
C*********************************************************************
C                          UNBALF                          *
C*********************************************************************
C
C     PERFORM A CONVERGENCE TEST FOR THE UNBALANCE FORCE.
C
      SUBROUTINE UNBALF ( F,FP,INCONV,NEQ,Q,TOLFOR)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION F(*),FP(*),Q(*)
C
      UNBFI =0.D00
      UNBFP =0.D00
C
C     COMPUTE THE UNBALANCE FORCE ..................................
      DO 10 I=1,NEQ
        UNBFI = UNBFI + (Q(I)-F(I))**2
        UNBFP = UNBFP + (Q(I)-FP(I))**2
   10 CONTINUE
C
C     CHECK WITH TOLERANCES ........................................
      IF ( UNBFP.NE.0.D0 ) THEN
        C= ( DSQRT(UNBFI) ) / ( DSQRT(UNBFP) )
C
        IF ( C.GT.TOLFOR ) THEN
          INCONV = INCONV + 100
        END IF
      ELSE
        INCONV = INCONV + 100
      END IF
C
```

```
      RETURN
      END
C*********************************************************************
C                         DISPL                          *
C*********************************************************************
C
C    PERFORM THE DISPLACEMENT CONVERGENCE TEST USING THE EUCLIDEAN
C    VECTOR NORM OF DISPLACEMENTS.
C
      SUBROUTINE DISPL ( D,DD,INCONV,NEQ,TOLDIS )
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION D(*),DD(*)
C
      DELTAD = 0.D00
      TOTALD = 0.D00
C
C    EUCLIDIAN VECTOR NORM OF DISPLACEMENTS ...........................
      DO 10 I=1,NEQ
       DELTAD = DELTAD + (DD(I))**2
       TOTALD = TOTALD + (D(I))**2
   10 CONTINUE
C
C    CHECK WITH TOLERANCES ..........................................
      IF ( TOTALD.NE.0 ) THEN
        C = ( DSQRT(DELTAD) ) / ( DSQRT(TOTALD) )
        IF ( C.GT.TOLDIS ) 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                        POSTN
C*********************************************************************
C
C    IT PERFOMS THE POSTPROCESSING OF NONLINEAR ANALYSIS FOR EACH
C    TIME INCREMENT
C
      SUBROUTINE POSTN(JCODE,D,P)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION JCODE(6,*),D(*),P(6,*)
C
      REWIND (3)
      REWIND (4)
      REWIND (13)
      REWIND (14)
C
C--- READ THE NODE WHOSE DISPLACEMENTS ARE TO BE ECHOED.
C
      READ (3) IND
C
   10 IF(IND .NE. 0) THEN
        DO 20 KK = 1,6
          NGLO = JCODE(KK,IND)
          IF(NGLO .NE. 0) THEN
            DEF = D(NGLO)
          ELSE
          DEF = 0.D0
        ENDIF
C
        WRITE (4) DEF
 20    CONTINUE
        READ (3) IND
      GO TO 10
    ENDIF
C
C--- JOINT NOS. WHOSE JOINT FORCES ARE TO BE ECHOED.
C
    READ (13) IP
 50 IF(IP .NE. 0) THEN
       WRITE (14) P(1,IP),P(2,IP),P(3,IP),P(4,IP),P(5,IP),P(6,IP)
       READ (13) IP
       GO TO 50
    ENDIF
C
    RETURN
    END
C*********************************************************************
C                    INTER
C*********************************************************************
C
C   IT CALCULATES THE INTERPOLATION FUNCTIONS AND THEIR DERIVATIVES
C   AT THE GIVEN INTERPOLATION POINTS.
C
    SUBROUTINE INTER(H,DH,RR)
    IMPLICIT REAL*8(A-H,O-Z)
    DIMENSION H(*),DH(*)
C
    H(1) = - RR/2.D0 * (1.D0 - RR)
    H(2) =   RR/2.D0 * (1.D0 + RR)
    H(3) =   1.D0 - RR*RR
C
    DH(1) = - 0.5D0 + RR
    DH(2) =   0.5D0 + RR
    DH(3) = - 2.D0 * RR
C
    RETURN
    END
C*********************************************************************
C*                    RIKWEM                             *
C*********************************************************************
C
C   IT CALCULATES THE GENERALIZED ARC LENGTH IN THE MODIFIED RIKS
C   WEMPNER METHOD. IT CALLS 'TRLVCT', 'STIFF', 'SOLVE', 'PRESER',
C   'UPDATC', 'RESULT', 'POSTN' AND 'FORCES'.
C
    SUBROUTINE RIKWEM(D,DD,DD0,F,FP,FPI,ICONV,NEQ,NJ,Q,TOLENE,TOLFOR,
   .           TOLDIS,QI,QIMAX,DQI,FORCE,VR,VS,VT,EPS,STRESS,
   .           MINC,JCODE,B,XJINV,DEPTH,WIDTH,X,G,DNL,GMOD,
   .           EMOD,VVR,VVS,VVT,FINT,MCODE,RES,QBAR,DD01,
   .           H,DH,NE,WTR,WTS,WTT,R,S,T,P,DDT,DDP,DD1,DD2,
   .           TTT,SSS,MAXA,TYPE,LSS,BNL,DS1,DS2,DS3,ITEUPD,
   .           IRES,ITEDES,MAXIT,XP,VRP,VSP,VTP)
C
    IMPLICIT REAL*8(A-H,O-Z)
    CHARACTER*(*) TYPE*6
C
    DIMENSION D(*),DD(*),DD0(*),F(*),FP(*),FPI(*),Q(*),FORCE(*),
   .      VR(3,*),VS(3,*),VT(3,*),EPS(*),STRESS(*),MINC(3,*),
   .      JCODE(6,*),B(3,3,*),XJINV(3,*),DEPTH(*),WIDTH(*),X(3,*),
     .      G(3,3,*),DNL(6,*),GMOD(*),EMOD(*),VVR(*),VVS(*),VVT(*),
     .      FINT(*),MCODE(18,*),RES(*),QBAR(*),H(*),DH(*),WTR(*),
     .      WTS(*),WTT(*),R(*),S(*),T(*),P(6,*),DDT(*),DDP(*),
     .      DD1(*),DD2(*),TTT(*),SSS(*),MAXA(*),BNL(2,2,2,3,9,*),
     .      DS1(2,2,*),DS2(2,2,*),DS3(2,2,*),DD01(*),XP(3,*),
     .      VRP(3,*),VSP(3,*),VTP(3,*)
C
      DO 10 I = 1,NEQ
        TTT(I) = 0.D0
        DDT(I) = 0.D0
   10 CONTINUE
      ITECNT = 0
      JTECNT = 0
      IFLAG = 0
      NTRY = 0
C
   20 IF(QI .LE. QIMAX .AND. IFLAG .LE. 1) THEN
        CALL STIFF (NE,MINC,R,S,T,WTR,WTS,WTT,TYPE,JCODE,SSS,MAXA,
     .              LSS,B,EMOD,GMOD,H,DH,VR,VS,VT,DEPTH,WIDTH,G,
     .              XJINV,X,VVR,VVS,VVT,BNL,STRESS,DS1,DS2,DS3)
C
        DO 30 I = 1,NEQ
          TTT(I) = QBAR(I)
   30   CONTINUE
C
        CALL SOLVE(SSS,TTT,MAXA,NEQ,1)
C
        DO 40 I = 1,NEQ
          DD01(I) = TTT(I)
          TTT(I) = 0.D0
   40   CONTINUE
C
        IF(ITECNT .EQ. 0) THEN
          DS = DQI*DSQRT(DOTPRD(DD01,DD01,NEQ) + 1.D0)
          DSMAX = DS*2.D0
        ELSE
          DQI = DS/DSQRT(DOTPRD(DD01,DD01,NEQ) + 1.D0)
          TEMP = DQI*(DOTPRD(DD01,DDP,NEQ)+DQI1)
          IF(TEMP .GT. 0.D0) THEN
            SGN = 1.D0
          ELSE
            SGN = -1.D0
          ENDIF
          DQI = SGN*DQI
        ENDIF
C
C--- SAVE THE VALUES FOR THE FIRST TRIAL CONFIGURATION;
C
        DQI1 = DQI
        DO 50 I = 1,NEQ
          DDP(I) = 0.D0
          DD0(I) = DQI * DD01(I)
          DDP(I) = DDP(I) + DD0(I)
C
C--- CALCULATE THE TOTAL DISPLACEMENTS.
C
          D(I) = D(I) + DD0(I)
          DDT(I) = DD0(I) + DDT(I)
   50   CONTINUE
C
        CALL PRESER(X,VR,VS,VT,XP,VRP,VSP,VTP,NJ)
        CALL UPDATC(X,D,VR,VS,VT,DD0,NEQ,NJ,NE,MINC,JCODE)
C
```

```
      QI = QI + DQI
      IF(QI .GT. QIMAX) THEN
        QI = QIMAX
        IFLAG = IFLAG + 1
      ENDIF
C
      CALL TRLVCT(D,DD,DD0,F,FP,FPI,ICONV,NEQ,NJ,Q,TOLENE,TOLFOR,
     .          TOLDIS,QI,QIMAX,DQI,FORCE,VR,VS,VT,EPS,STRESS,
     .          MINC,JCODE,B,XJINV,DEPTH,WIDTH,X,G,DNL,GMOD,EMOD,
     .          VVR,VVS,VVT,FINT,MCODE,RES,QBAR,H,DH,NE,WTR,WTS,
     .          WTT,R,S,T,P,DDT,DDP,DD1,DD2,DQI1,TTT,SSS,MAXA,
     .          TYPE,LSS,BNL,DS1,DS2,DS3,ITEUPD,IT,MAXIT,XP,
     .          VRP,VSP,VTP)
C
      DO 55 I = 1,NEQ
        FP(I) = F(I)
   55 CONTINUE
      ITECNT = ITECNT + 1
      JTECNT = JTECNT + 1
      IRES = IRES + 1
      IF(ICONV .NE. 0) THEN
        WRITE(6,60)'******ERROR******','SOLUTION FAILS TO CONVERGE'
   60   FORMAT(///T30,A//T27,A)
        STOP
      ELSE
        IF(IRES .GE. JTECNT .OR. QI .GE. QIMAX) THEN
          RATIO = QI / QIMAX
          CALL POSTN(JCODE,D,P)
          CALL RESULT(RATIO)
          JTECNT = 0
        ENDIF
      ENDIF
      DS = DS*DSQRT(DFLOAT(ITEDES)/DFLOAT(IT))
      IF(DABS(DS) .GT. DSMAX) THEN
        IF(DS. LT. 0) THEN
          DS = -DSMAX
        ELSE
          DS = DSMAX
        ENDIF
      ENDIF
      GO TO 20
    ENDIF
C
    RETURN
    END
C**********************************************************************
C*                      TRLVCT                              *
C**********************************************************************
C
C   IT PERFORMS ITERATIONS TO OBTAIN VECTORS ORTHOGONAL TO THE
C   GENERALIZED ARC LENGTH. IT CALLS 'FORCES', 'UPDATS', 'STIFF',
C   'SOLVE', 'PRESER', 'UPDATC', AND 'TEST'.
C
    SUBROUTINE TRLVCT(D,DD,DD0,F,FP,FPI,ICONV,NEQ,NJ,Q,TOLENE,TOLFOR,
     .          TOLDIS,QI,QIMAX,DQI,FORCE,VR,VS,VT,EPS,STRESS,
     .          MINC,JCODE,B,XJINV,DEPTH,WIDTH,X,G,DNL,GMOD,EMOD,
     .          VVR,VVS,VVT,FINT,MCODE,RES,QBAR,H,DH,NE,WTR,WTS,
     .          WTT,R,S,T,P,DDT,DDP,DD1,DD2,DQI1,TTT,SSS,MAXA,
     .          TYPE,LSS,BNL,DS1,DS2,DS3,ITEUPD,IT,MAXIT,XP,
     .          VRP,VSP,VTP)
C
    IMPLICIT REAL*8(A-H,O-Z)
```

```
      CHARACTER*(*) TYPE*6
C
      DIMENSION D(*),DD(*),DD0(*),F(*),FP(*),FPI(*),Q(*),FORCE(*),
     .      VR(3,*),VS(3,*),VT(3,*),EPS(*),STRESS(*),MINC(3,*),
     .      JCODE(6,*),B(3,3,*),XJINV(3,*),DEPTH(*),WIDTH(*),X(3,*),
     .      G(3,3,*),DNL(6,*),GMOD(*),EMOD(*),VVR(*),VVS(*),VVT(*),
     .      FINT(*),MCODE(18,*),RES(*),QBAR(*),H(*),DH(*),WTR(*),
     .      WTS(*),WTT(*),R(*),S(*),T(*),P(6,*),DDT(*),DDP(*),
     .      DD1(*),DD2(*),TTT(*),SSS(*),MAXA(*),BNL(2,2,2,3,9,*),
     .      DS1(2,2,*),DS2(2,2,*),DS3(2,2,*),XP(3,*),VRP(3,*),
     .      VSP(3,*),VTP(3,*)
C
      CALL FORCES(NJ,P,FORCE,NEQ,R,S,T,WTR,WTS,WTT,NE,H,DH,EPS,
     .         STRESS,MINC,JCODE,DD0,B,XJINV,DEPTH,WIDTH,X,
     .         VR,VS,VT,G,DNL,GMOD,EMOD,VVR,VVS,
     .         VVT,FINT,MCODE,RES,Q,XP,VRP,VSP,VTP)
C
      CALL UPDATS(EPS,STRESS,MINC,JCODE,DD0,B,H,DH,R,S,T,WTR,WTS,
     .       WTT,XJINV,DEPTH,WIDTH,X,VR,VS,VT,G,DNL,GMOD,
     .       EMOD,VVR,VVS,VVT,XP,VRP,VSP,VTP)
      DO 55 I = 1,NEQ
        F(I) = FORCE(I)
 55   CONTINUE
C
      IT = 0
      ICONV = 1
      ITENUM = ITEUPD
C
 10   IF(IT .LE. MAXIT .AND. ICONV .NE. 0) THEN
        DO 20 I = 1,NEQ
          FPI(I) = F(I)
            Q(I) = QBAR(I) * QI
 20     CONTINUE
C
C--- UPDATE THE STIFFNESS MATRIX.
C
        IF(ITENUM .GE. ITEUPD) THEN
          CALL STIFF (NE,MINC,R,S,T,WTR,WTS,WTT,TYPE,JCODE,SSS,MAXA,
     .          LSS,B,EMOD,GMOD,H,DH,VR,VS,VT,DEPTH,WIDTH,G,
     .          XJINV,X,VVR,VVS,VVT,BNL,STRESS,DS1,DS2,DS3)
          ITENUM = 0
        ENDIF
C
        DO 30 I = 1,NEQ
          TTT(I) = QBAR(I)
 30     CONTINUE
C
        IF(ITENUM .EQ. 0) THEN
          CALL SOLVE(SSS,TTT,MAXA,NEQ,1)
        ELSE
          CALL SOLVE(SSS,TTT,MAXA,NEQ,2)
        ENDIF
C
        DO 40 I = 1,NEQ
          DD1(I) = TTT(I)
          TTT(I) = 0.D0
 40     CONTINUE
C
        DO 50 I = 1,NEQ
          TTT(I) = Q(I) - F(I)
 50     CONTINUE
C
```

```
          CALL SOLVE(SSS,TTT,MAXA,NEQ,2)
C
        DO 60 I = 1,NEQ
          DD2(I) = TTT(I)
          TTT(I) = 0.D0
  60    CONTINUE
C
        DQI = -( DOTPRD(DD0,DD2,NEQ) )/( DOTPRD(DD0,DD1,NEQ) + DQI1 )
C
        DO 70 I = 1,NEQ
          DD(I) =DQI*DD1(I) +DD2(I)
  70    CONTINUE
C
        DO 80 I = 1,NEQ
          D(I) = D(I) + DD(I)
          DDP(I) = DDP(I) + DD(I)
          DDT(I) = DDT(I) + DD(I)
  80    CONTINUE
C
C--- UPDATE THE COORDINATES.
C
        CALL PRESER(X,VR,VS,VT,XP,VRP,VSP,VTP,NJ)
        CALL UPDATC(X,D,VR,VS,VT,DD,NEQ,NJ,NE,MINC,JCODE)
C
        CALL FORCES(NJ,P,FORCE,NEQ,R,S,T,WTR,WTS,WTT,NE,H,DH,EPS,
     .          STRESS,MINC,JCODE,DD,B,XJINV,DEPTH,WIDTH,X,
     .          VR,VS,VT,G,DNL,GMOD,EMOD,VVR,VVS,
     .          VVT,FINT,MCODE,RES,Q,XP,VRP,VSP,VTP)
C
        CALL UPDATS(EPS,STRESS,MINC,JCODE,DD,B,H,DH,R,S,T,WTR,WTS,
     .          WTT,XJINV,DEPTH,WIDTH,X,VR,VS,VT,G,DNL,GMOD,
     .          EMOD,VVR,VVS,VVT,XP,VRP,VSP,VTP)
C
        DO 90 I = 1,NEQ
          F(I) = FORCE(I)
  90    CONTINUE
C
        QI = QI + DQI
C
        IF(QI .GT. QIMAX) THEN
          QI = QIMAX
        ENDIF
C
C--- CALL TEST TO CHECK THE CONVERGENCE OF THE SOLUTION.
C
        CALL TEST(D,DD,DD0,F,FP,FPI,ICONV,JCODE,NEQ,NJ,
     .           Q,TOLDIS,TOLENE,TOLFOR)
C
        ITENUM = ITENUM + 1
        IT = IT + 1
        GO TO 10
      ENDIF
      RETURN
      END
C
C           ******************************
C--------------*        DOTPRD         *----------------------
C           ******************************
C
C   FUNCTION THAT COMPUTES THE DOT PRODUCT OF DOT1 AND DOT2.
C
      FUNCTION DOTPRD(DOT1,DOT2,N)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION DOT1(1),DOT2(1)
C
      DOTPRD=0.D00
      DO 10 I=1,N
        DOTPRD=DOTPRD+DOT1(I)*DOT2(I)
   10 CONTINUE
      RETURN
      END
C**********************************************************************
C                         PRESER
C**********************************************************************
C
C   IT SAVES THE CURRENT CONFIGURATION BEFORE EACH NEW TIME INCREMENT.
C
      SUBROUTINE PRESER(X,VR,VS,VT,XP,VRP,VSP,VTP,NJ)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X(3,*),XP(3,*),VR(3,*),VRP(3,*),VS(3,*),VSP(3,*),
     .          VT(3,*),VTP(3,*)
C
      DO 30 J = 1,NJ
        DO 20 I = 1,3
          XP(I,J) = X(I,J)
          VRP(I,J) = VR(I,J)
          VSP(I,J) = VS(I,J)
          VTP(I,J) = VT(I,J)
   20   CONTINUE
   30 CONTINUE
C
      RETURN
      END
C**********************************************************************
C*                         NEWVAR
C**********************************************************************
C
C   IT IS A CONTROL MODULE FOR NONLINEAR ANALYSIS BY NEWTON RALPHSON
C   METHOD USING A VARIABLE TIME STEP. IT CALLS 'STORE', 'NEWRAP',
C   'STPDCR', 'STPICR', 'POSTN', AND 'RESULT'.
C
      SUBROUTINE NEWVAR(Q,QBAR,QI,QIMAX,DQI,D,DD,R,S,T,EPS,STRESS,MINC,
     .            JCODE,B,BNL,H,DH,XJINV,DEPTH,WIDTH,X,VR,VS,VT,
     .            G,DNL,GMOD,EMOD,WTR,WTS,WTT,FINT,P,NE,MCODE,
     .            VVR,VVS,VVT,IRES,TOLENE,TOLFOR,TOLDIS,F,FP,FPI,
     .            FORCE,TTT,RES,SSS,MAXA,NEQ,TYPE,LSS,
     .            DD0,NJ,MAXIT,ITEUPD,DS1,DS2,DS3,XP,VRP,VSP,VTP,
     .            ALGO,RP,XPP,VRPP,VSPP,VTPP,DP,DQIMAX,DQIMIN,
     .            DTOL)
      IMPLICIT REAL*8(A-H,O-Z)
      CHARACTER*(*) TYPE*6,ALGO*6
C
      DIMENSION Q(*),QBAR(*),D(*),DD(*),R(*),S(*),T(*),EPS(*),STRESS(*),
     .      MINC(3,*),JCODE(6,*),B(3,3,*),BNL(2,2,2,3,9,*),H(*),
     .      DEPTH(*),WIDTH(*),X(3,*),VR(3,*),VS(3,*),VT(3,*),
     .      G(3,3,*),DNL(6,*),GMOD(*),EMOD(*),WTR(*),WTS(*),WTT(*),
     .      FINT(*),P(6,*),MCODE(18,*),VVR(*),VVS(*),VVT(*),DH(*),
     .      F(*),FP(*),FPI(*),FORCE(*),TTT(*),RES(*),XJINV(3,*),
     .      SSS(*),MAXA(*),DD0(*),DS1(2,2,*),DS2(2,2,*),DS3(2,2,*),
     .      XP(3,*),VRP(3,*),VSP(3,*),VTP(3,*),RP(*),XPP(3,*),
     .      VRPP(3,*),VSPP(3,*),VTPP(3,*),DP(*)
C
      IFLAG = 0
      ISTEP = 0
      ICONV = 0
      NTRY = 0
```

```
        DO 10 I = 1,NEQ
           Q(I) = QI*QBAR(I)
 10      CONTINUE
C
        CALL STORE(X,VR,VS,VT,D,XPP,VRPP,VSPP,VTPP,DP,NJ,NEQ,NE)
C
        CALL NEWRAP(SSS,TTT,MAXA,NEQ,NE,MINC,R,S,T,WTR,WTS,WTT,TYPE,
     .             JCODE,LSS,B,EMOD,GMOD,H,DH,VR,VS,VT,DEPTH,WIDTH,
     .             G,XJINV,X,VVR,VVS,VVT,FORCE,MCODE,FINT,RES,Q,
     .             EPS,STRESS,D,DNL,DD,DD0,F,FP,FPI,ICONV,NJ,
     .             TOLDIS,TOLENE,TOLFOR,MAXIT,ITEUPD,BNL,P,DS1,
     .             DS2,DS3,XP,VRP,VSP,VTP,IDIVER,ITECNT,ALGO,RP,
     .             DTOL)
C
        IF(IDIVER .EQ. 1) THEN
          CALL STPDCR(X,XPP,VR,VRPP,VS,VSPP,VT,VTPP,NJ,NEQ,D,DP,QI,
     .             DQI,DQIMIN,NE)
          NTRY = 0
          ICONV = 0
        ELSE
          KTECNT = ITECNT - 1
          MAX = MAXIT / 2
          IF(KTECNT .LE. MAX) THEN
            NTRY = NTRY + 1
          ELSE
            NTRY = 0
          ENDIF
          DO 15 IEQ = 1,NEQ
            FP(IEQ) = FORCE(IEQ)
 15        CONTINUE
        ENDIF
C
        IF(NTRY .EQ. 2) THEN
          CALL STPICR(DQI,DQIMAX,NTRY)
        ENDIF
C
        IF(INCONV .NE. 0) THEN
          WRITE(6,20)
 20        FORMAT(///T1,80('#')//T27,'SOLUTION FAILED TO CONVERGE'//T1,
     .         80('#'))
          STOP
        ELSE
          IF(IDIVER .EQ. 0) THEN
            ISTEP = ISTEP + 1
          ENDIF
          IF(ISTEP .EQ. IRES .OR. QI .GE. QIMAX) THEN
            CALL POSTN(JCODE,D,P)
            RATIO = QI / QIMAX
            CALL RESULT(RATIO)
          ENDIF
       ENDIF
C
        QI = QI + DQI
        IF(QI .GT. QIMAX) THEN
          QI = QIMAX
          IFLAG = IFLAG + 1
        ENDIF
        GO TO 5
      ENDIF
C
      RETURN
      END
C***************************************************************************
```

```
C*                    DVTEST                    *
C*********************************************************************
C
C  FOR A GIVEN LOAD STEP, ESTIMATE THE REQUIRED NUMBER OF ITERATIONS
C  FOR CONVERGENCE BY THE N/R METHOD.
C
      SUBROUTINE DVTEST(DTOL,IDIVER,ITECNT,MAXIT,NEQ,NTRY,R,RP)
      IMPLICIT REAL*8(A-H,O-Z)
C
      DIMENSION R(*),RP(*)
C
      IDIVER = 0
      RMAX = 0.D0
      RPREV = 0.D0
C
      DO 10 I = 1,NEQ
        IF(ABS (R(I)) .GT. RMAX) THEN
          RMAX = ABS (R(I))
        ENDIF
   10 CONTINUE
C
      DO 20 I = 1,NEQ
        IF(ABS (RP(I)) .GT. RPREV) THEN
          RPREV = ABS(RP(I))
        ENDIF
   20 CONTINUE
      ESTN = ITECNT + DLOG(DTOL/RMAX)/DLOG(RMAX/RPREV)
      IF((RMAX/RPREV) .GT. 1) THEN
C        ESTN = MAXIT + 1.D0
      ENDIF
      IF(ESTN .GT. MAXIT) THEN
        IDIVER = 1
      ENDIF
C
      RETURN
      END
C*********************************************************************
C*                    STPDCR                    *
C*********************************************************************
C
C   IT REDUCES THE STEP INCREMENT/DECREMENT SIZE BY 75 PERCENT IN THE
C   VARIABLE TIME STEP N/R METHOD. PRINT AN ERROR MESSAGE AND ABORT
C   THE PROGRAM IF THE STEP INCREMENT IS ALREADY EQUAL TO DQIMIN.
C
      SUBROUTINE STPDCR(X,XPP,VR,VRPP,VS,VSPP,VT,VTPP,NJ,NEQ,D,DP,QI,
     .            DQI,DQIMIN,NE)
      IMPLICIT REAL*8(A-H,O-Z)
C
      DIMENSION X(3,*),XPP(3,*),VR(3,*),VRPP(3,*),VS(3,*),VSPP(3,*),
     .       VT(3,*),VTPP(3,*),D(*),DP(*)
C
      DO 20 J = 1,NJ
        DO 10 L = 1,3
          X(L,J) = XPP(L,J)
          VR(L,J) = VRPP(L,J)
          VS(L,J) = VSPP(L,J)
          VT(L,J) = VTPP(L,J)
   10   CONTINUE
   20 CONTINUE
C
      DO 30 I = 1,NEQ
        D(I) = DP(I)
```

```
 30  CONTINUE
C
     QI = QI - DQI
C
     DO 40 I = 1,NEQ
       D(I) = DP(I)
 40  CONTINUE
C
     REWIND (9)
     REWIND (10)
     REWIND (15)
     REWIND (18)
     REWIND (19)
     REWIND (22)
     REWIND (23)
     REWIND (24)
C
     READ (15) IR,IS,IT
     READ (9) IES
  50 IF(IES .NE. 0) THEN
       DO 80 IRR = 1,IR
         DO 70 ISS = 1,IS
           DO 60 ITT = 1,IT
             READ (18) IA,IB,IC,E1,E2,E3,S1,S2,S3
             WRITE (10) IA,IB,IC,E1,E2,E3,S1,S2,S3
 60        CONTINUE
 70      CONTINUE
 80    CONTINUE
       READ (9) IES
       GO TO 50
     ENDIF
C
     DO 120 IE = 1,NE
       DO 110 IRR = 1,2
         DO 100 ISS = 1,2
           DO 90 ITT = 1,2
             READ (19) IEE,S1,S2,S3
             WRITE (23) IEE,S1,S2,S3
  90        CONTINUE
 100      CONTINUE
 110    CONTINUE
 120 CONTINUE
C
     IF(DQI .LE. DQIMIN) THEN
      WRITE(6,130) '***ERROR***','REQUIRED STEP INCREMENT IS SMALLER',
     .            'THAN ALLOWABLE'
 130  FORMAT(//T35/T15,A,1X,A)
       STOP
C
     ELSE
       DQI = DQI*0.25
     ENDIF
C
     IF(DQI .LT. DQIMIN) THEN
       DQI = DQIMIN
     ENDIF
C
     RETURN
     END
C**********************************************************************
C*                         STPICR                              *
C**********************************************************************
C
```

```
C   IT INCREASES THE TIME INCREMENT BY 25 PERCENT AND CHECKS THE
C   NEW VALUE TO KEEP IT WITHIN THE MAXIMUM SPECIFIED (DQIMAX).
C
      SUBROUTINE STPICR(DQI,DQIMAX,NTRY)
      IMPLICIT REAL*8(A-H,O-Z)
C
      DQI = DQI*1.25D0
      IF(DQI .GE. DQIMAX) THEN
        DQI = DQIMAX
      ENDIF
      NTRY = 0
C
      RETURN
      END
C*********************************************************************
C                        STORE
C*********************************************************************
C
C   IT STORES THE CONFIGURATION OF THE SYSTEM AT TIME T. IF NO
C   CONVERGENCE IS ACHIEVED IN THE PRESCRIBED NUMBER OF ITERATIONS
C   THE PROGRAM RETURNS TO THE CURRENT CONFIGURATION AND REITERATES
C   WITH A REDUCED TIME INCREMENT.
C
      SUBROUTINE STORE(X,VR,VS,VT,D,XPP,VRPP,VSPP,VTPP,DP,NJ,NEQ,NE)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X(3,*),XPP(3,*),VR(3,*),VRPP(3,*),VS(3,*),D(*),DP(*),
     .         VTPP(3,*),VSPP(3,*),VT(3,*)
C
      DO 20 J = 1,NJ
        DO 10 I = 1,3
          XPP(I,J) = X(I,J)
          VRPP(I,J) = VR(I,J)
          VSPP(I,J) = VS(I,J)
          VTPP(I,J) = VT(I,J)
 10     CONTINUE
 20   CONTINUE
C
      DO 30 I = 1,NEQ
        DP(I) = D(I)
 30   CONTINUE
C
      REWIND (9)
      REWIND (10)
      REWIND (18)
      REWIND (19)
      REWIND (23)
      REWIND (15)
C
      READ (15) IR,IS,IT
      READ (9) IES
 50   IF(IES .NE. 0) THEN
        DO 80 IRR = 1,IR
          DO 70 ISS = 1,IS
            DO 60 ITT = 1,IT
              READ (10) IA,IB,IC,E1,E2,E3,S1,S2,S3
              WRITE (18) IA,IB,IC,E1,E2,E3,S1,S2,S3
 60         CONTINUE
 70       CONTINUE
 80     CONTINUE
        READ (9) IES
        GO TO 50
      ENDIF
C
```

```
      DO 120 IE = 1,NE
        DO 110 IRR = 1,2
          DO 100 ISS = 1,2
            DO 90 ITT = 1,2
              READ (23) S1,S2,S3
              WRITE (19) S1,S2,S3
 90         CONTINUE
100       CONTINUE
110     CONTINUE
120   CONTINUE
C
      RETURN
      END
```