NASA
Technical
Paper
2767

1988

Continuum Modeling of Large Lattice Structures

Status and Projections

Ahmed K. Noor
The George Washington University
Joint Institute for Advancement of Flight Sciences
Langley Research Center
Hampton, Virginia

Martin M. Mikulas, Jr.
Langley Research Center
Hampton, Virginia

National Aeronautics
and Space Administration

Scientific and Technical
Information Division

Contents

PRECEDING PAGE BLANK NOT FILMED

Summary

The status and some recent developments of continuum modeling for large repetitive lattice structures are summarized. Discussion focuses on a number of aspects including definition of an effective substitute continuum, characterization of the continuum model, and the different approaches for the generation of the properties of the continuum, namely, the constitutive matrix, the matrix of mass densities, and the matrix of thermal coefficients. Also, a simple approach is presented which can be used to generate analytic expressions and/or numerical values of the continuum properties.

The proposed approach is applied to some beamlike and double-layered platelike lattices, currently considered as candidates for large space structures. Future directions of research on continuum modeling are identified. These include needed extensions and applications of continuum modeling as well as computational strategies and modeling techniques.

1. Introduction

Lattice structures have been used for many years in spanning large areas with few intermediate supports. These structures can combine low cost with light weight and an esthetically pleasing appearance. Also, because of their ease of packaging, transporting, and assembling in space, lattice structures have attracted considerable attention for use in large-area space structures such as the Space Station, large space mirrors, antennas, multipurpose platforms, and power systems for supporting space operations. For the large-area lattice structures considered for space applications, a main feature is the repetition of a basic pattern or configuration many times.

The state of the art in the analysis, design, and construction of lattice structures up to 1976 is reviewed in references 1 and 2. The currently used approaches for analyzing large repetitive lattices can be grouped into four classes (see table 1), namely,

1. Direct method
2. Discrete field methods
3. Periodic structure approaches
4. Substitute continuum approaches

In the first approach (direct method) the structure is analyzed as a system of discrete finite elements, and the methods of solving structural framework problems are applied. It has the obvious drawback of being computationally expensive for large lattices. This is particularly true when a buckling, a vibration, or a nonlinear analysis is required. Reduction methods could be applied for buckling and nonlinear problems to remedy this drawback (refs. 3 and 4).

The second approach (discrete field methods) takes advantage of the regularity of the structure and involves writing the equilibrium and compatibility equations at a typical joint of the lattice and either solving the resulting difference equations directly or using truncated Taylor series expansions to replace the difference equations by differential equations (see, for example, refs. 5 to 9). This approach works well for simple lattice configurations, but becomes quite involved for lattices with complex geometry.

The third group of methods, referred to as "periodic structure approaches," are based on either (1) the combined use of finite elements and transfer matrix methods, which is efficient only for rotationally periodic (i.e., cyclically symmetric) structures or lattices with simple geometries (refs. 10 to 12), or (2) the exact representation of the stiffness of an individual member from which the analysis of beamlike lattices with simply supported edges can be performed (refs. 13 to 15).

The fourth approach is based on replacing the actual lattice structure by a substitute continuum model that is equivalent to the original structure in some sense, such as the constitutive relations, strain energy, and/or kinetic energy (see, for example, refs. 16 to 33). The use of continuum models to simulate the behavior of planar lattice beams dates back to the previous century (see ref. 34, p. 483). It has gained popularity only in recent years and has been applied to a variety of other discrete systems and phenomena including solid and liquid crystals, dislocations and defects, composite materials, and biological systems.

The number of publications on continuum modeling of repetitive lattice structures has been steadily increasing. Therefore, there is a need to broaden awareness among practicing engineers and research workers of the recent developments in various aspects of continuum modeling for large lattice structures. The present paper is a modest attempt to fill this void. Specifically, the objectives of this paper are

1. To assess the effectiveness of the currently used approaches for continuum modeling
2. To present a simple and rational approach for development of continuum models for large repetitive lattice structures
3. To identify the future directions of research which have high potential for realizing the advantages of continuum modeling.

The scope of the present study includes thermoelastic stress analysis, buckling, free vibration, and geometrically nonlinear problems of large lattice structures. Beamlike and platelike repetitive lattices with pin and rigid joints are considered. Continuum modeling of lattices with flexible joints is also discussed.

2. Advantages of Continuum Modeling

Before an assessment is made of the different approaches for developing continuum models, the following three advantages of using the continuum modeling approach for analyzing repetitive lattice structures are identified. First, it offers a practical and efficient approach for analyzing large lattice structures. This is particularly true for beamlike and platelike lattices, wherein a dimensionality reduction can result in a substantial reduction in the number of degrees of freedom. Second, it provides a simple means of comparing structural, thermal, and dynamic characteristics of lattices with different configurations and assessing the sensitivity of their responses to variations in material and geometric properties. Third, it provides an effective tool for parameter and system identification and for feedback control system design of lattice structures.

3. Definition and Key Elements of a Substitute Continuum Model

A number of definitions have been given for the substitute continuum model. Herein an *effective continuum model* is defined to be a continuum with the following characteristics:

1. The same amounts of thermoelastic strain and kinetic energies are stored in it as those of the original lattice structure when both are deformed identically.
2. The temperature distribution, loading, and boundary conditions of the continuum simulate those of the original lattice structure.
3. For beamlike and platelike lattices the continuum models are one-dimensional beams and two-dimensional plates, respectively (see fig. 1).
4. Local deformations are accounted for.
5. Lattices with pin joints are modeled as classical continua, and lattices with rigid (and/or flexible) joints are generally modeled as micropolar continua.

The last two characteristics are perhaps the most important in terms of recent developments and are discussed in the next two sections.

3.1. Local Deformations

The local deformations of two axially loaded planar trusses are shown in figure 2. The first truss has double lacing and a single-bay repeating cell. The second truss has single lacing and a double-bay repeating cell. The chord members of the first truss remain straight as shown on the top sketch. On the other hand, the actual deformation of the single-laced truss has the zig-zag pattern shown on the top right sketch. On the average, however, the chord members remain straight. Early continuum models averaged these deformations, thereby substantially overestimating the axial stiffness. Recent continuum models, for lattices with more than one bay in their repeating cells, do account for the local deformations (see refs. 25, 26, and 35).

3.2. Ordinary Versus Micropolar Continua

A contrast between the ordinary and micropolar continua is made in figure 3. For an axially loaded pin-jointed truss member the transverse motion is completely characterized by the joint displacements. The member rotation ψ is related to the joint displacements w_i and w_j. Therefore, the appropriate continuum to use in modeling pin-jointed trusses is the ordinary continuum for which the displacement field completely characterizes the motion of the structure.

On the other hand, for a rigid-jointed member, the transverse motion is characterized by both the joint displacements w_i and w_j and the joint rotations θ_i and θ_j, which are independent degrees of freedom. Therefore, the appropriate continuum to use in modeling rigid-jointed flexural members is one whose motion is characterized by both a displacement field and an independent rotation field (referred to as "micro-rotation" field). The micropolar continuum is such a continuum.

3.3. Characterization of the Substitute Continuum Model

The substitute continuum model is characterized by the thermoelastic constitutive relations and density parameters, which are determined in terms of the geometric and material properties of the original lattice structure. The thermoelastic constitutive relations and density parameters of the continuum can then be used to determine (1) the thermoelastic strain and kinetic energies, (2) the governing differential equations, and whenever appropriate (3) the equivalent discrete finite element model.

3.4. Comments on Continuum Models

The following three comments regarding continuum models seem to be in order:

1. For some lattices the substitute continuum models may not have much resemblance to the continuum theories commonly used in engineering practice. Also, for complicated lattices the continuum models may be fairly complicated, and therefore, not useful for practical applications.
2. The accuracy of the predictions of the con- tinuum approximation increases with the increase in the number of repeating cells (or modules) constituting the original lattice structure.
3. The response of the substitute continuum model (which simulates that of the original lattice structure) can be generated through (1) exact (or analytic) solution of the governing differential equations or (2) application of a discretization technique such as the Rayleigh-Ritz technique or the finite element method.

4. Development of Substitute Continuum Models for Stress Analysis and Free Vibration Problems

A number of approaches have been proposed for developing continuum models and for determining the appropriate constitutive relations and density parameters. These approaches include

1. Relating the force or deformation characteristics (or both) of a small segment of the lattice to those of a small segment of the continuum (refs. 20, 22, and 36 to 42)
2. Using the discrete field method to obtain the governing difference equations of the lattice and either solving them directly or converting them to approximate differential equations (refs. 6 to 8)
3. Applying homogenization techniques based on using multiple-scale asymptotic expansions (refs. 43 to 46)
4. Using energy equivalence concepts, the potential and kinetic energies of a typical (repeating) cell of the lattice are equated to those of the continuum, after expanding the nodal displacements of the lattice in a Taylor series.

The latter approach has been applied to a number of beamlike and platelike lattices. Computerized symbolic manipulation was used to generate analytic expressions for the stiffness and density parameters of the continuum (see refs. 25, 26, 35, 47, and 48). More recently, an equivalent approach was proposed for generating the properties of simplified one- and two-dimensional continuum models of beamlike and platelike lattice trusses with pin joints. This approach does not require the use of computerized symbolic manipulation (see ref. 49). Rather, numerical values of the stiffness and mass coefficients can be obtained by using a small FORTRAN program on a microcomputer. A modified version of this approach is now described.

The three key elements of the foregoing approach are

1. Introduction of kinematic and temperature assumptions to reduce the dimensionality of the continuum
2. Expansion of each nodal displacement, each strain component, and the temperature in a Taylor series
3. Generation of four transformation matrices that relate nodal displacements, axial strains, and temperatures of individual members of the repeating cell to the displacement, strain, and temperature parameters of the continuum.

A schematic representation of the sequence of steps involved in the generation of the stiffness coefficients for simplified one- and two-dimensional continuum models is given in figure 4. The procedure is described subsequently for lattices with pin joints, and the appendix lists the computer program, which was developed by Sandra L. Whitworth and Jeanne M. Peters of the George Washington University Joint Institute for Advancement of Flight Sciences, Hampton, Virginia. The procedure consists of the three major phases described in the following three sections.

4.1. Phase 1—Generation of the Thermoelastic Stiffnesses of a Repeating Cell

Step 1. A repeating cell (or module) is isolated from the lattice grid. The axial strain, temperature, and consistent mass matrix of a typical member k of the repeating cell are given by $\varepsilon^{(k)}$, $T^{(k)}$, and $[\mathrm{m}^{(k)}]$, respectively.

Step 2. The axial strain $\varepsilon^{(k)}$ is expressed in terms of the vector of strain components in the coordinate directions through the following matrix equation:

$$\varepsilon^{(k)} = \left[R^{(k)}\right]\{\varepsilon\}^{(k)} \tag{1}$$

where

$$\{\varepsilon\}^{(k)} = \begin{Bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ \gamma_{12} \\ \gamma_{13} \\ \gamma_{23} \end{Bmatrix} \tag{2}$$

$$\left[R^{(k)}\right] = \left[\ell^2 \quad m^2 \quad n^2 \quad \ell m \quad \ell n \quad mn\right]^{(k)} \tag{3}$$

where ε_{11}, ε_{22}, and ε_{33} are the axial strains in the coordinate directions; γ_{12}, γ_{13}, and γ_{23} are the shearing strains; and $(\ell, m, n)^{(k)}$ are the direction cosines of the member.

For simplicity, in the present study the strain state is assumed to be uniform within each repeating cell. Variation of the strain state within the repeating cell can be accounted for by expanding $\{\varepsilon\}^{(k)}$ in a Taylor series about the center of the repeating cell. The number of terms in the Taylor series expansion is equal to the number of independent deformation modes of the repeating cell.

Step 3. The stiffness matrix and the thermal load vector of the repeating cell are generated as follows:

$$[K] = \sum_{\text{Members}} (EAL)^{(k)} \left[R^{(k)}\right]^t \left[R^{(k)}\right] \tag{4}$$

$$\{P_T\} = \sum_{\text{Members}} (\alpha EAL)^{(k)} \left[R^{(k)}\right]^t T^{(k)} \tag{5}$$

where E, A, L, and α are the elastic modulus, cross-sectional area, length, and coefficient of thermal expansion of member k; and superscript t denotes transposition.

The thermoelastic stiffnesses of the equivalent three-dimensional classical continuum are obtained by dividing the right sides of equations (4) and (5) by the volume of the repeating cell. Note that for members shared by n repeating cells, their cross-sectional areas in equations (4) and (5) are divided by n.

4.2. Phase 2—Generation of Thermoelastic Stiffnesses and Density Parameters of the Beam and Plate Continua

Step 4. Reduction in dimensionality is achieved by introducing kinematic and temperature assumptions regarding the variation of the displacement components, strain components, and temperature in the plane of the cross section (for beamlike lattices) or in the thickness direction (for platelike lattices).

The vectors of nodal displacements and strain components and the temperature of member k are expressed in terms of the corresponding continuum parameters by means of transformation matrices as follows:

$$\{u\}^{(k)} = \left[\Gamma_u^{(k)}\right] \{u\}_c \tag{6}$$

$$\{\varepsilon\}^{(k)} = \left[\Gamma_\varepsilon^{(k)}\right] \{\varepsilon\}_c \tag{7}$$

$$T^{(k)} = \left[\Gamma_T^{(k)}\right] \{T\}_c \tag{8}$$

where $\{u\}_c$, $\{\varepsilon\}_c$, and $\{T\}_c$ are the vectors of displacement parameters, strain parameters, and temperature parameters used in describing the beam (or plate) continuum; $[\Gamma_u^{(k)}]$, $[\Gamma_\varepsilon^{(k)}]$, and $[\Gamma_T^{(k)}]$ are transformation matrices.

Step 5. The stiffness matrix, matrix of density parameters, and thermal load vector of the continuum are given by

$$[K]_c = \frac{1}{\Omega} \sum_{\text{Members}} (EAL)^{(k)} \left[\Gamma_\varepsilon^{(k)}\right]^t \left[R^{(k)}\right]^t \left[R^{(k)}\right] \left[\Gamma_\varepsilon^{(k)}\right] \tag{9}$$

$$[\mathcal{M}]_c = \frac{1}{\Omega} \sum_{\text{Members}} \left[\Gamma_u^{(k)}\right] \left[\mathrm{m}^{(k)}\right] \left[\Gamma_u^{(k)}\right] \tag{10}$$

$$\{P_T\}_c = \frac{1}{\Omega} \sum_{\text{Members}} (\alpha EAL)^{(k)} \left[R^{(k)}\right]^t \left[\Gamma_u^{(k)}\right] \{T\}_c \tag{11}$$

where Ω is a characteristic geometric property of the repeating cell (length of the repeating cell for beamlike lattices and planform area of the repeating cell for platelike lattices).

4.3. Phase 3—Generation of Properties of Simplified Beam and Plate Continua

Step 6. Simplified continuum models are obtained by partitioning the displacement and strain parameters of the continuum as follows:

$$\{\varepsilon\}_c = \begin{Bmatrix} \varepsilon_{c1} \\ \varepsilon_{c2} \\ \varepsilon_{c3} \end{Bmatrix} \tag{12}$$

$$\{u\}_c = \begin{Bmatrix} u_{c1} \\ u_{c2} \end{Bmatrix} \tag{13}$$

where $\{\varepsilon_{c1}\}$ are the strain parameters to be retained in the simplified model, $\{\varepsilon_{c2}\}$ are the strain parameters associated with the neglected stress resultants (or internal forces), $\{\varepsilon_{c3}\}$ are the strain parameters to be omitted, and $\{u_{c1}\}$ and $\{u_{c2}\}$ are the displacement parameters associated with the retained and neglected mass coefficients, respectively. The stiffness matrix, matrix of density parameters, and thermal load vector of the continuum are partitioned accordingly:

$$[K]_c = \begin{bmatrix} K_{11} & K_{12} & K_{13} \\ & K_{22} & K_{23} \\ \text{Symm.} & & K_{33} \end{bmatrix} \tag{14}$$

$$[\mathcal{M}]_c = \begin{bmatrix} \mathcal{M}_{11} & \mathcal{M}_{12} \\ \text{Symm.} & \mathcal{M}_{22} \end{bmatrix} \tag{15}$$

$$\{P_T\}_c = \left\{ \begin{array}{c} P_{T1} \\ P_{T2} \\ P_{T3} \end{array} \right\} \tag{16}$$

The effective thermoelastic coefficients of the simplified continuum models are obtained by deleting the rows and columns associated with $\{\varepsilon_{c3}\}$ and by expressing $\{\varepsilon_{c2}\}$ in terms of $\{\varepsilon_{c1}\}$ (using static condensation). The resulting constitutive matrix and thermal load vector have the following forms:

$$[C] = [K_{11}] - [K_{12}][K_{22}]^{-1}[K_{21}] \tag{17}$$

$$\{C_T\} = \{P_{T1}\} - [K_{12}][K_{22}]^{-1}\{P_{T2}\} \tag{18}$$

The density parameters of the simplified continuum are obtained by neglecting the terms associated with $\{u_{c2}\}$, that is,

$$[m] = [\mathcal{M}_{11}] \tag{19}$$

Step 7. The thermoelastic strain energy density of the continuum can be written in the following form:

$$U_c = U_o - U_1 - U_2 \tag{20}$$

where U_o is the isothermal strain energy density, and U_1 and U_2 are the contributions to the strain energy of the linear and quadratic terms in the temperature parameters. The expressions for U_o and U_1 are

$$\left. \begin{array}{l} U_o = \dfrac{1}{2}\{\varepsilon_{c1}\}^t[C]\{\varepsilon_{c1}\} \\ U_1 = \{\varepsilon_{c1}\}^t\{C_T\} \end{array} \right\} \tag{21}$$

The expression for U_2 is not presented herein since U_2 is inconsequential in the development of the properties of the continuum models.

Step 8. The kinetic energy density of the continuum is given by

$$\mathcal{K} = \frac{1}{2}\{\dot{u}_{c1}\}^t[m]\{\dot{u}_{c1}\} \tag{22}$$

where a dot $(\cdot)$ refers to derivative with respect to time.

The sign convention for the internal forces and generalized displacements, along with the associated direct stiffnesses and mass coefficients, is given in figures 5 and 6 for the simplified beam and plate continua.

The foregoing approach is applied in the succeeding sections to the development of continuum models for beamlike and platelike lattice trusses with pin joints.

5. Application to Beamlike Lattices

The foregoing approach is applied to the beamlike lattices having four longerons and an orthogonal tetrahedral (unsymmetric) configuration (see fig. 7). These trusses are primary candidates for use in the keel beam design of the Space Station (see fig. 8). For the lattice shown in figure 7, the smallest possible repeating cell that can be isolated from the grid extends over one bay of the original structure.

5.1. Kinematic and Temperature Assumptions

The deformed position of any cross section of the lattice is assumed to be specified by 12 displacement parameters. A Timoshenko-type beam theory is obtained by eliminating six of these parameters. The following expressions are used for the displacement field in the plane of the cross section (plane yz):

$$\left\{ \begin{array}{c} u(x,y,z) \\ v(x,y,z) \\ w(x,y,z) \end{array} \right\} = \left\{ \begin{array}{c} u^o \\ v^o \\ w^o \end{array} \right\} + \left[\begin{array}{ccc} \cdot & z & \text{-}y \\ \text{-}z & \cdot & \cdot \\ y & \cdot & \cdot \end{array} \right] \left\{ \begin{array}{c} \phi_x \\ \phi_y \\ \phi_z \end{array} \right\} + \left[\begin{array}{ccc} \cdot & \cdot & \cdot \\ y & \cdot & \frac{1}{2}z \\ \cdot & z & \frac{1}{2}y \end{array} \right] \left\{ \begin{array}{c} \varepsilon_y^o \\ \varepsilon_z^o \\ \gamma_{yz}^o \end{array} \right\} + yz \left\{ \begin{array}{c} \bar{u} \\ \bar{v} \\ \bar{w} \end{array} \right\} \tag{23}$$

where u^o, v^o, and w^o are the displacement components at $y = z = 0$ (chosen to be at the center of the repeating cell); ϕ_x, ϕ_y, and ϕ_z are the rotation components about the coordinate axes; ε_y^o and ε_z^o are the extensional strains in the y- and z-directions; γ_{yz}^o is the shearing strain in the plane of the cross section (plane yz); $\bar{u}$, $\bar{v}$, and $\bar{w}$ represent warping and distortion of the cross section; and the dots refer to zero terms. The sign convention for the displacement and rotation components is shown in figure 5. The 12 parameters u^o, v^o, w^o, ϕ_x, ϕ_y, ϕ_z, ε_y^o, ε_z^o, γ_{yz}^o,$\bar{u}$, $\bar{v}$, and $\bar{w}$ are functions of x only. Note that equations (23) provide an exact representation of the displacement field in the plane of the cross section. As a consequence of the kinematic assumptions (eqs. (23)), the strain components have a bilinear variation in the plane of the cross section (plane yz) as follows:

$$\left.\begin{aligned}
\varepsilon_{11} &= \varepsilon_x^o - y\kappa_y^o + z\kappa_z^o + yz\psi^o \\
\varepsilon_{22} &= \varepsilon_y^o + z\bar{v} \\
\varepsilon_{33} &= \varepsilon_z^o + y\bar{w} \\
\gamma_{12} &= \gamma_{xy}^o + y\,\partial_x\varepsilon_y^o + z\left(-\kappa_t^o + \bar{\kappa}\right) + yz\,\partial_x\bar{v} \\
\gamma_{13} &= \gamma_{xz}^o + y\left(\kappa_t^o + \bar{\kappa}\right) + z\,\partial_x\varepsilon_z^o + yz\,\partial_x\bar{w} \\
\gamma_{23} &= \gamma_{yz}^o + y\bar{v} + z\bar{w}
\end{aligned}\right\} \quad (24)$$

where ε_x^o is the extensional strain of the centerline; κ_y^o and κ_z^o are the curvature changes in the y- and z-directions; κ_t^o is the twist; γ_{xy}^o and γ_{xz}^o are the transverse shear strains; and $\partial_x \equiv d/dx$. The strain parameters ε_x^o, κ_y^o, κ_z^o, $\gamma_{xz}^o, \gamma_{xy}^o$, κ_t^o, ψ^o, and $\bar{\kappa}$ are functions of x only and can be expressed in terms of the displacement and rotation components as follows:

$$\left.\begin{aligned}
\varepsilon_x^o &= \partial_x u^o & \gamma_{xz}^o &= \partial_x w^o + \phi_y \\
\kappa_y^o &= \partial_x\phi_z & \kappa_t^o &= \partial_x\phi_x \\
\kappa_z^o &= \partial_x\phi_y & \psi^o &= \partial_x\bar{u} \\
\gamma_{xy}^o &= \partial_x v^o - \phi_z & \bar{\kappa} &= \frac{1}{2}\partial_x\gamma_{yz}^o + \bar{u}
\end{aligned}\right\} \quad (25)$$

The temperature distribution is also assumed to be bilinear in the plane of the cross section; that is,

$$T(x,y,z) = T^o + y\,\partial_y T^o + z\,\partial_z T^o + yz\,\partial_y\,\partial_z T^o \quad (26)$$

where T^o is the temperature at $y = z = 0$ and the temperature gradients are

$$\partial_y T^o \equiv \frac{\partial T^o}{\partial y} \qquad \partial_z T^o \equiv \frac{\partial T^o}{\partial z} \qquad \partial_y\,\partial_z T^o = \frac{\partial^2 T^o}{\partial y\,\partial z}$$

5.2. Transformation Matrices and Simplified Beam Models

The kinematic assumptions (eqs. (23)) are used to generate the transformation matrix $\Gamma_u^{(k)}$ of the kth member. Equations (24) and (26) are used to generate the transformation matrices $[\Gamma_\varepsilon^{(k)}]$ and $[\Gamma_T^{(k)}]$, with y and z in these equations denoting the coordinates of the center of the kth member.

A simplified (Timoshenko-type) continuum beam model is obtained by retaining six displacement parameters in equations (23) and six strain parameters in equations (24). This is accomplished by defining the partitions of $\{\varepsilon\}_c$ and $\{u\}_c$ as follows:

$$\left.\begin{aligned}
\{\varepsilon_{c1}\}^t &= [\varepsilon_x^o \quad \kappa_y^o \quad \kappa_z^o \quad \gamma_{xy}^o \quad \gamma_{xz}^o \quad \kappa_t^o] \\
\{\varepsilon_{c2}\}^t &= [\bar{\kappa} \quad \gamma_{yz}^o \quad \psi^o \quad \varepsilon_y^o \quad \varepsilon_z^o \quad \bar{v} \quad \bar{w}] \\
\{\varepsilon_{c3}\}^t &= [\partial_x\varepsilon_y^o \quad \partial_x\varepsilon_z^o \quad \partial_x\gamma_{yz}^o \quad \partial_x\bar{v} \quad \partial_x\bar{w}] \\
\{u_{c1}\}^t &= [u^o \quad v^o \quad w^o \quad \phi_x \quad \phi_y \quad \phi_z] \\
\{u_{c2}\}^t &= [\bar{u} \quad \gamma_{yz}^o \quad \varepsilon_y^o \quad \varepsilon_z^o \quad \bar{v} \quad \bar{w}]
\end{aligned}\right\} \quad (27)$$

The constitutive matrix and the matrix of material densities are obtained by following the procedure described in section 4. The numerical values for the coefficients of these matrices are given in table 2 for the lattice structure shown in figure 7. Stiffness coefficients and density parameters for the simplified and higher order continuum models are presented in reference 27 for orthogonal tetrahedral lattices with more than four longerons.

6. Application to Platelike Lattices

In this section applications of the foregoing approach to the tetrahedral and hexahedral grids shown in figures 1 and 9, respectively, are outlined. The tetrahedral grid is chosen because it has many attractive features for application to space structures. The hexahedral grid is chosen because it contains more redundancies and has vertical members that present difficulties in some continuum modeling approaches. In each case, a typical repeating element is isolated from the grid. The continuum model in this case is a two-dimensional plate continuum.

6.1. Kinematic and Temperature Assumptions

The three displacement components u, v, and w are assumed to have a linear variation in the thickness coordinate z; that is,

$$\begin{Bmatrix} u(x,y,z) \\ v(x,y,z) \\ w(x,y,z) \end{Bmatrix} = \begin{Bmatrix} u^o \\ v^o \\ w^o \end{Bmatrix} + z \begin{Bmatrix} \phi_x \\ \phi_y \\ \varepsilon_z^o \end{Bmatrix} \quad (28)$$

where u^o, v^o, and w^o are the displacement components at $z = 0$ (chosen to be at the center of the repeating cell); ϕ_x and ϕ_y are the rotation components; and ε_z^o is the transverse normal strain in the z-direction. The six parameters u^o, v^o, w^o, ϕ_x, ϕ_y, and ε_z^o are functions of x and y only. The sign convention for the displacement and rotation components is shown in figure 6. Equations (28) represent the exact displacement variation in the thickness direction provided that no internal nodes are present.

As a consequence of the displacement assumptions (eqs. (28)), the strain components have a linear variation across the thickness of the plate as follows:

$$\left\{\begin{array}{c}\varepsilon_{11}\\ \varepsilon_{22}\\ \varepsilon_{33}\\ \gamma_{12}\\ \gamma_{13}\\ \gamma_{23}\end{array}\right\} = \left\{\begin{array}{c}\varepsilon_x^o\\ \varepsilon_y^o\\ \varepsilon_z^o\\ \gamma_{xy}^o\\ \gamma_{xz}^o\\ \gamma_{yz}^o\end{array}\right\} + z\left\{\begin{array}{c}\kappa_x^o\\ \kappa_y^o\\ \cdot\\ 2\kappa_{xy}^o\\ \partial_x\varepsilon_z^o\\ \partial_y\varepsilon_z^o\end{array}\right\} \tag{29}$$

where ε_x^o, ε_y^o, and γ_{xy}^o are the extensional and shearing strains of the middle plane; κ_x^o, κ_y^o, and $2\kappa_{xy}^o$ are the curvature changes and twist; γ_{xz}^o and γ_{yz}^o are the transverse shear strains; and a dot represents zero. The strain measures ε_x^o, ε_y^o, γ_{xy}^o, κ_x^o, κ_y^o, $2\kappa_{xy}^o$, γ_{xz}^o, and γ_{yz}^o are independent of z and can be expressed in terms of the displacement and rotation components of the middle plane as follows:

$$\left.\begin{aligned}\varepsilon_x^o &= \partial_x u^o\\ \varepsilon_y^o &= \partial_y v^o\\ \gamma_{xy}^o &= \partial_x v^o + \partial_y u^o\\ \kappa_x^o &= \partial_x \phi_x\\ \kappa_y^o &= \partial_y \phi_y\\ 2\kappa_{xy}^o &= \partial_x\phi_y + \partial_y\phi_x\\ \gamma_{xz}^o &= \partial_x w^o + \phi_x\\ \gamma_{yz}^o &= \partial_y w^o + \phi_y\end{aligned}\right\} \tag{30}$$

To account for the local transverse deformation of the repeating cell for the platelike lattices considered in the present study, the transverse shear strain expressions must be augmented by their derivatives with respect to x and y as follows:

$$\left.\begin{aligned}\gamma_{13} &= \gamma_{xz}^o + z\,\partial_x\varepsilon_z^o + x\,\partial_x\gamma_{xz}^o + y\,\partial_y\gamma_{xz}^o\\ \gamma_{23} &= \gamma_{yz}^o + z\,\partial_y\varepsilon_z^o + x\,\partial_x\gamma_{yz}^o + y\,\partial_y\gamma_{yz}^o\end{aligned}\right\} \tag{31}$$

The temperature distribution is also assumed to be linear in the z-direction; that is,

$$T(x,y,z) = T^o + z\,\partial_z T^o \tag{32}$$

where T^o is the temperature at $z = 0$, and $\partial_z T^o = \partial T^o/\partial z$ is the temperature gradient in the z-direction. Both T and $\partial_z T^o$ are functions of x and y only.

6.2. Transformation Matrices and Simplified Plate Models

Equations (28), (29), (31), and (32) are used to generate the three transformation matrices $[\Gamma_u^{(k)}]$, $[\Gamma_\varepsilon^{(k)}]$, and $[\Gamma_T^{(k)}]$ for the kth member. A simplified shear-deformation-type plate theory is obtained by defining the partitions of $\{\varepsilon_c\}$ and $\{u_c\}$ as follows:

$$\left.\begin{aligned}\{\varepsilon_{c1}\}^t &= \left[\varepsilon_x^o \quad \varepsilon_y^o \quad \gamma_{xy}^o \quad \kappa_x^o \quad \kappa_y^o \quad 2\kappa_{xy}^o \quad \gamma_{xz}^o \quad \gamma_{yz}^o\right]\\ \{\varepsilon_{c2}\}^t &= \left[\partial_x\gamma_{xz}^o \quad \partial_y\gamma_{xz}^o \quad \partial_x\gamma_{yz}^o \quad \partial_y\gamma_{yz}^o \quad \varepsilon_z^o\right]\\ \{\varepsilon_{c3}\}^t &= \left[\partial_x\varepsilon_z^o \quad \partial_y\varepsilon_z^o\right]\\ \{u_{c1}\}^t &= \left[u^o \quad v^o \quad w^o \quad \phi_x \quad \phi_y\right]\\ \{u_{c2}\}^t &= \left[\varepsilon_z^o\right]\end{aligned}\right\} \tag{33}$$

The constitutive matrix and the matrix of material densities are obtained by following the procedure outlined in section 4. The numerical values for the coefficients of these matrices are given in table 3 for the double-layered hexahedral lattice configuration shown in figure 9. Analytical expressions for the stiffness coefficients and density parameters of simplified plate models are given in reference 25 for the double-layered tetrahedral and hexahedral grids.

7. Comments on the Foregoing Approach for Developing Continuum Models

The following comments regarding the procedure for developing continuum models seem to be in order:

1. The strain parameters $\{\varepsilon_{c1}\}$, $\{\varepsilon_{c2}\}$, and $\{\varepsilon_{c3}\}$ can be selected on the basis of a sensitivity analysis of the response with respect to each of these parameters.

2. For simple lattice configurations, the constitutive matrix $[C]$, the matrix of density parameters $[m]$, and the vector of thermal loads $\{C_T\}$ can be obtained in symbolic form using a computerized symbolic manipulation language such as MACSYMA (see ref. 47). However, for more complicated configurations, it is more convenient to generate $[C]$, $[m]$, and $\{C_T\}$ numerically using the FORTRAN program listed in the appendix on a microcomputer.

3. Higher order continuum models can be obtained by using more strain and displacement parameters than $\{\varepsilon_{c1}\}$ and $\{u_{c1}\}$. The additional parameters may be needed for the accurate prediction of the detailed displacement and stress distribution in the lattice (for example, localized displacements and stresses near a loaded edge).

4. A similar procedure for generating the characteristics of the continuum model was presented in references 23 and 24. In the cited references the lattice structure is considered to be the sum of several arrays of parallel elements, and thermoelastic stiffness coefficients of the continuum model are obtained by applying the tensor transformation relations to the unidirectional stiffness and thermal coefficients of each of the individual members constituting the

repeating cell. The three-dimensional continuum coefficients generated are equal to the stiffness matrix and thermal load vector $[K]$ and $\{P_T\}$ of equations (4) and (5), each divided by the volume of the repeating cell.

8. Numerical Studies

To test and evaluate the accuracy of the predictions of the continuum models developed, a number of thermoelastic stress analysis, free vibration, and bifurcation buckling problems were solved using these models. Comparisons were made with exact solutions based on direct analysis of the actual lattice structure. Two problem sets are presented herein: (1) stress and free vibration analysis of anisotropic beamlike lattices and (2) free vibration analysis of platelike lattices.

8.1. Stress and Free Vibration Analysis of Anisotropic Beamlike Lattices

The first problem set considered is that of the orthogonal tetrahedral truss configuration shown in figure 7, which is a candidate for the primary truss support structure for the Space Station (see fig. 8). Because of the unsymmetry of the basic configuration, the continuum model is anisotropic in the sense that both the extensional and the shear effects, as well as the bending and twisting effects, are coupled. The elastic and dynamic characteristics of the continuum model are listed in table 2. For stress analysis problems, the structure was subjected to longitudinal loadings, transverse loadings, and twisting moments at its free end. Typical results are shown in figures 10 and 11 and in tables 4 and 5.

Table 4 gives the maximum displacements u^o, v^o, and w^o at the free end obtained by the beam models, along with the exact solutions obtained by the direct analysis of the actual structure for lattices with 10 bays. For the sake of comparison, the predictions of the foregoing continuum model are given along with those of the continuum model in which all the coupling stiffness and density parameters are neglected. Table 5 and figure 10 give an indication of the accuracy of the lowest 10 frequencies obtained by the 2 continuum models. Then vibration mode shapes associated with the lowest 10 frequencies of the 10-bay truss are shown in figure 11. As can be seen from table 5 and figure 10, the predictions of the foregoing continuum models are highly accurate. The maximum error in the lowest 10 frequencies for the 10-bay truss is 4.73 percent. The error reduces to less than 1.5 percent when the number of bays increases to 30. On the other hand, neglecting the coupling terms results in overestimating the stiffness of the structure, thereby increasing the vibration frequencies and reducing the maximum displacements. This effect is very pronounced for the vibration frequencies associated with the higher modes. For the 10-bay truss, when the coupling terms are neglected, the maximum errors in the lowest 10 frequencies are over 65 percent.

8.2. Free Vibration Analysis of Platelike Lattice Grids

The second problem considered is that of the free vibration analysis of a cantilevered hexahedral double-layered grid. The characteristics of the grid are shown in figure 9. The boundary nodes at $x = 0$ are completely restrained. The continuum model is taken to be a square, shear-flexible plate with one edge completely restrained. The elastic and dynamic characteristics of the plate are listed in table 3. The continuum solutions presented herein are converged finite element solutions. Typical results are shown in figures 12 and 13 and in table 6.

An indication of the accuracy of the lowest 10 vibration frequencies is given in table 6 and figure 12. The standard of comparison is taken to be the vibration frequencies obtained by the direct finite element solutions of the actual grid. Figure 13 shows the vibration mode shapes associated with the minimum 10 frequencies of the $8L \times 8L$ grid. As can be seen from table 6, for an $8L \times 8L$ grid, the maximum error in the lowest 10 vibration frequencies is 1.1 percent. The error reduces to less than 0.55 percent when a $16L \times 16L$ grid is used.

9. Application of Continuum Models to Stability Problems

An important criterion in the design of lightweight lattice trusses is the onset of elastic instability. For finely divided lattice trusses the failure mode is usually that of general (system) instability rather than local snap-through of a single joint or member buckling. The foregoing continuum models can be extended to the general (global) stability analysis of lattice structures. This is accomplished by equating the potential energy due to initial stresses of the original lattice to that of the continuum, from which expressions for the geometric stiffness coefficients of the continuum are obtained in terms of the geometric and elastic properties of the original lattice structure. The procedure is outlined as follows:

1. A repeating cell is isolated from the lattice grid. The geometric stiffness matrix of a typical member k of the repeating cell referred to the coordinates of the repeating element is given by $[\mathrm{g}^{(k)}]$.

2. The nodal displacements of member k are expressed in terms of the continuum displacement parameters and their derivatives with respect to the spatial coordinates by means of the following transformation:

$$\{u\}^{(k)} = \left[\bar{\Gamma}_{\Delta}^{(k)}\right] \{\Delta\}_c \tag{34}$$

where $\{\Delta\}_c$ is the vector of displacement parameters and their spatial derivatives and $\left[\bar{\Gamma}_{\Delta}^{(k)}\right]$ is a transformation matrix. Equations (34) are obtained by expanding the nodal displacements of member k in a Taylor series about the center of the repeating cell.

3. The geometric stiffness matrix of the continuum is given by

$$[\mathcal{G}]_c = \frac{1}{\Omega} \sum_{\text{Members}} \left[\bar{\Gamma}_{\Delta}^{(k)}\right]^t \left[g^{(k)}\right] \left[\bar{\Gamma}_{\Delta}^{(k)}\right] \tag{35}$$

4. Simplified continuum models are obtained by partitioning the vector $\{\Delta\}_c$ and the matrix $[G]_c$ as follows:

$$\{\Delta\}_c = \begin{Bmatrix} \Delta_{c1} \\ \Delta_{c2} \end{Bmatrix} \tag{36}$$

$$[\mathcal{G}]_c = \begin{bmatrix} \mathcal{G}_{11} & \mathcal{G}_{12} \\ \mathcal{G}_{21} & \mathcal{G}_{22} \end{bmatrix} \tag{37}$$

where $\{\Delta_{c1}\}$ and $\{\Delta_{c2}\}$ are the displacement parameters associated with the retained and neglected geometric stiffness coefficients, respectively. The geometric stiffness coefficients of the simplified continuum are given by

$$[g] = [\mathcal{G}_{11}] \tag{38}$$

Note that the foregoing continuum approach for stability analysis assumes that the individual members of the lattice remain straight and stable during buckling. Local member instability and snap-through buckling of a single joint are not predicted by the present theory.

10. Other Reported Applications of Continuum Modeling

Other reported applications of continuum modeling which have not been discussed in the preceding sections are described subsequently.

10.1. Geometrically Nonlinear Problems of Beamlike Lattices With Pin Joints

In references 50 and 51 an incremental stiffness matrix of the equivalent continuum is used to account for the large-displacement nonlinear effects in beamlike lattices. This approach is applied to static, large-deflection analysis and large-amplitude free vibration analysis, as well as buckling and postbuckling analysis.

10.2. Beamlike Lattices With Material Damping

Planar beamlike lattices with members having viscous damping are modeled as continuum Timoshenko beams in references 52 and 53. The global damping characteristics are determined in terms of the damping coefficients and dimensions of the truss members.

10.3. Stress, Free Vibration, and Buckling Problems of Beamlike Lattices With Rigid and Flexible Joints

Micropolar continuum models are developed in references 48, 54, and 55 for the static, free vibration, and buckling problems of planar and spatial beamlike lattices with rigid joints. The study of reference 48 shows that ordinary shear-flexible continuum beam models are generally adequate for predicting the global response characteristics of lattices that do not need to have rigid joints for their kinematic stability. Such continuum beam models are inadequate however when diagonal members have very small cross-sectional areas. Incorporation of joint flexibility in the continuum model is discussed in reference 56.

10.4. Stress and Free Vibration Problems of Single-Layered Grids

Ordinary continuum plate models are developed in references 20 and 57 for predicting the response of single-layered grids with rigid joints. The transverse motion of the grids is predicted accurately by these models. However, accurate prediction of the in-plane motion of single-layered grids with rigid joints requires the use of a micropolar continuum (see refs. 54, 58, and 59).

10.5. System Identification and Control Law Design of Lattice Structures

Because of the significant reduction in the number of material and structural parameters obtained by replacing the actual lattice structure by a continuum model, a number of recent studies have employed continuum modeling in control design and/or system identification analysis. These studies used either the governing partial differential equations of the continuum or the systems of ordinary differential equations resulting from discretizations of the continuum model (see refs. 60 to 63).

11. Future Directions of Research on Continuum Modeling

Continuum modeling is likely to play a significant role in the analysis, design, and control of large space structures, and accordingly, a number of extensions and computational strategies must be developed.

11.1. Needed Extensions and Future Applications

The needed extensions to the continuum model include incorporating effects of joint flexibility and damping.

New applications of the continuum model include:

1. Random periodic structures (i.e., structures with random variations in material properties, geometric characteristics, or periodicity)
2. Nonlinear structural dynamics and wave propagation problems
3. Multidisciplinary optimization problems (e.g., simultaneous optimization of structures and control systems)

11.2. Computational Strategies and Modeling Techniques

With regard to computational strategies and modeling techniques, the following two areas appear to have high potential for research.

Hybrid modeling approaches. Hybrid modeling approaches include combination of either (1) continuum and repetitive structure approaches for handling lattice structures with arbitrary boundary conditions or (2) continuum and discrete structure approaches for specialized problems such as stress concentration.

Application of operator splitting/reduction technique for analyzing complicated continuum models. Operator splitting/reduction techniques are very effective for generating the response of a complex structure (e.g., anisotropic continuum model) using large perturbations from the response of a simpler structure (e.g., corresponding orthotropic model). These techniques are applied to eigenvalue and nonlinear problems in references 64 and 65.

Reduction techniques have been shown to be related to the preconditioned conjugate gradient iterative method. The preconditioning matrix is taken to be the matrix of the simpler structure; and the preconditioned residuals provide sensitivity information of the response with respect to the complicating factors (e.g., anisotropy).

12. Concluding Remarks

The status and some recent developments in continuum modeling of large repetitive lattice structures are summarized. Discussion focuses on a number of aspects including definition of an effective substitute continuum, characterization of the continuum model, and the different approaches for generating the constitutive matrix and the matrix of mass densities of the continuum. Also, a simple approach is presented for generating the continuum properties. The approach can be used in conjunction with computerized symbolic manipulation to generate analytic expressions for the continuum properties or with a small FORTRAN program to determine the numerical values of these properties. The proposed approach is applied to the generation of continuum properties for beamlike and platelike lattices.

The status of continuum modeling for repetitive lattice structures can be summarized in the following:

1. For pin-jointed beamlike and platelike lattices, effective and verified ordinary continuum models exist for linear thermoelastic stress and free vibration problems. Bifurcation buckling loads associated with general instability (global buckling modes) can also be predicted accurately by continuum models. Applications of continuum modeling to nonlinear static and dynamic problems are limited. Simplified continuum models (associated with known engineering theories such as Timoshenko-type beam theory and Reissner-Mindlin-type plate theory) are adequate for many practical problems. Continuum models associated with higher order theories may be required for predicting localized effects.

2. For lattices with rigid joints, ordinary continuum models are available for the accurate prediction of the global response of lattices with rigid joints, provided that the response is not dominated by local member deformation (e.g., transverse motion of single-layered grids). Micropolar continuum models exist for predicting the in-plane motions of lattices with rigid joints.

Needed extensions of continuum modeling include incorporation of joint flexibility and damping. Also, application of continuum modeling to multidisciplinary optimization problems needs more development. These extensions are recommended as future directions for research. New computational and modeling strategies that have high potential for use in conjunction with continuum modeling include hybrid combination continuum and repetitive structure approaches; hybrid combination of continuum and discrete structure approaches; and the application of operator splitting/reduction technique for analyzing complicated continuum models.

Appendix

FORTRAN Program for Evaluating the Thermoelastic Stiffnesses and Density Parameters of the Continuum Models

This appendix contains the listing of the interactive program used for evaluating the thermoelastic stiffnesses and density parameters of simplified one- and two-dimensional continuum models of beamlike and platelike lattice structures with pin joints. The program is written in Microsoft FORTRAN 3.30 (1985) and works on IBM PC and XT with DOS 3.0 operating system. A tree structure depicting the calling sequence of the different subroutines is given in figure A1.

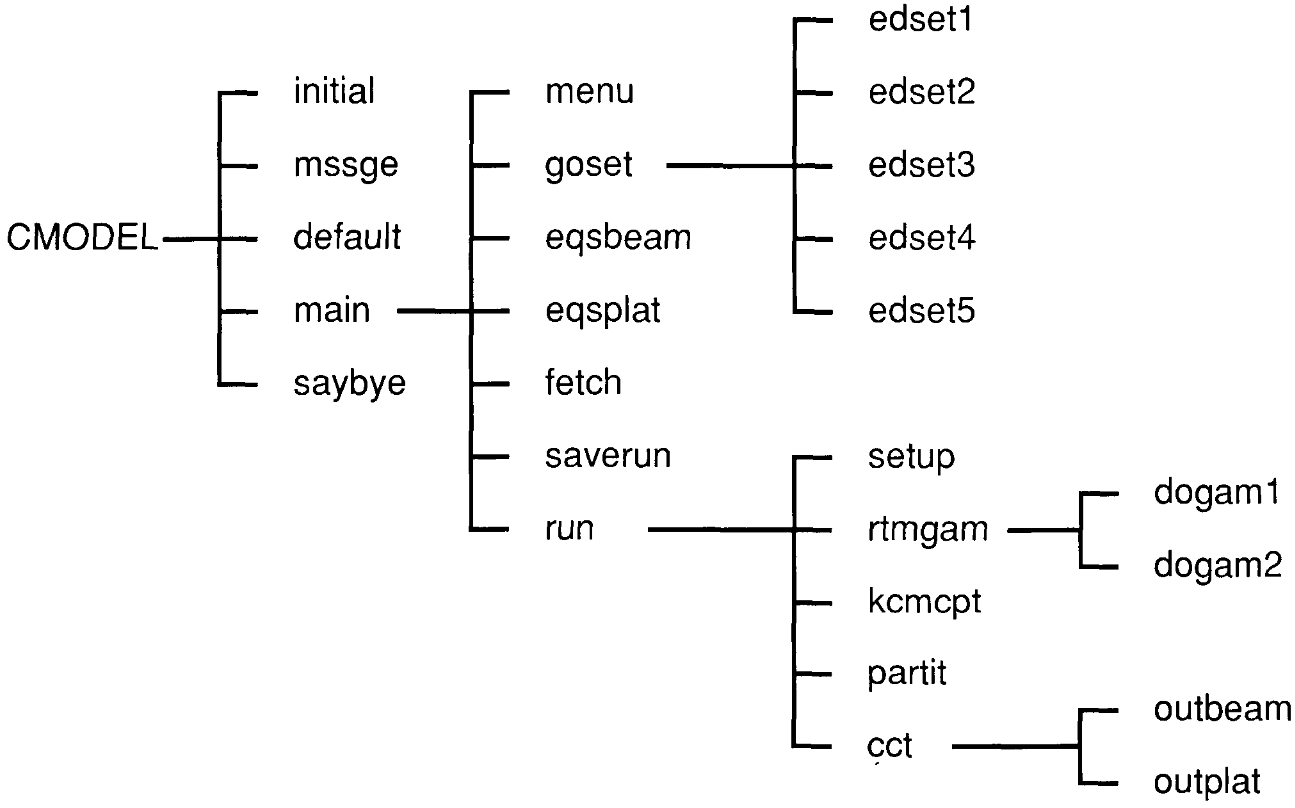


Figure A1. Tree structure for program.

General Comments:

1. There are 5 separate "include" files:
   DBLE, PARAM.CMN, DATA.CMN, CKCOLOR.CMN, GLOBCONS.CMN

2. The following utility routines are used (the routines in part a were obtained from MicroCompatibles, Inc., 301 Prelude Drive, Silver Spring, MD 20901):

```
a) Grafmatic library routines:
   QBORD    - set color of border
   QCLEAR   - clear screen and set color
   QCMOV    - move cursor to specified column and row
b) Other routines for which the source code is not included:
     CKFIELD   - read in value at specified column and row
     CKTABL    - read in table of values
     CLFIELD   - clear a field
     ERRORS    - print error message and beep
     INITIAL   - initialize key definitions and colors
     JCLEAR    - clear screen and move cursor to top left corner
     JLABEL    - print label at specified column and row
     PRFIELD   - print value at specified column and row
     PRTABL    - print table of values
     VTOASC    - convert value to ascii
```

<< file DBLE follows>>

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

<< file PARAM.CMN follows>>

```
C*COMDECK PARAM
C--------------
      INTEGER ZNN, ZNE, ZNPT, ZNSDF, ZNDDF, ZNCN
      PARAMETER (ZNN=18, ZNE=60, ZNPT=8, ZNSDF=18, ZNDDF=12, ZNCN=2)
C
C ZNN  = MAX. NUMBER OF NODES IN BAY
C ZNE  = MAX. NUMBER OF CONNECTING ELEMENTS IN BAY
C ZNPT = MAX. NUMBER OF PROPERTY TYPES
C ZNSDF = MAX. NUMBER OF STRAIN DEGREES OF FREEDOM
C ZNDDF = MAX. NUMBER OF CONTINUUM DISPLACEMENT DEGREES OF FREEDOM
C ZNCN = MAX. NUMBER OF NODES CONNECTED
C
```

<< file DATA.CMN follows>>

```
C*COMDECK CONN
C------------
      COMMON /CONN/ NOP(ZNE,ZNCN)
C
C NOP - CONNECTIVITY TABLE, NE X NCN
C
C*COMDECK COORD
C-------------
      COMMON /COORD/ XCOORD(ZNN), YCOORD(ZNN), ZCOORD(ZNN)
C
```

```
C YCOORD - COORDINATES IN Y DIRECTION
C ZCOORD - COORDINATES IN Z DIRECTION
C
C*COMDECK DIRCOS
C--------------
      COMMON /DIRCOS/ DCOSX(ZNE), DCOSY(ZNE), DCOSZ(ZNE)
C
C DCOSX - DIRECTION COSINE IN X DIRECTION (=L IN EQUATIONS)
C DCOSY - DIRECTION COSINE IN Y DIRECTION (=M IN EQUATIONS)
C DCOSZ - DIRECTION COSINE IN Z DIRECTION (=N IN EQUATIONS)
C
C*COMDECK ECUC
C------------
      INTEGER EC, UC, ECI, UCI
      COMMON /ECUC/ EC(ZNSDF), UC(ZNDDF), ECI(ZNSDF), UCI(ZNDDF),
     *             NEC1, NEC2, NEC3, NUC1, NUC2

C NOTE: PARAMETERS HAVE DEFAULT ORDERING AND ARE NUMBERED AS SUCH
C      EC,UC,ECI,UCI  TAKE CARE OF PERMUTATION OF THE PARAMETERS
C
C  EC = (EC1, EC2, EC3), WHERE EC1, EC2, EC3 CORRESPOND TO RETAINED,
C                        ASSOCIATED W/NEGLECTED, AND NEGLECTED
C                        PARAMETERS, RESPECTIVELY
C                        EC(K)  = N, THE NTH PARAMETER IS IN KTH POSITION
C ECI (INVERSE OF EC)    ECI(N) = K,     "         "         "      "
C  UC = (UC1, UC2),       WHERE UC1,UC2 CORRESPOND TO RETAINED AND
C                        NEGLECTED PARAMETERS, RESPECTIVELY
C                        UC(K)  = N, THE NTH PARAMETER IS IN KTH POSITION
C UCI (INVERSE OF UC)    UCI(N) = K,     "         "         "      "

C  NEC1,NEC2,NEC3,NUC1,NUC2 - NUMBER OF PARAMETERS IN EACH GROUP
C
C
C*COMDECK ECOORD
C--------------
      COMMON /ECOORD/ EX(ZNE), EY(ZNE), EZ(ZNE)
C
C EX - X COORDINATE AT CENTER OF ELEMENT
C EY - Y COORDINATE AT CENTER OF ELEMENT
C EZ - Z COORDINATE AT CENTER OF ELEMENT
C
C*COMDECK ELEQS
C-------------
      REAL*8 M
      COMMON /ELEQS/ K, R(6), T(4), M(6,6), GAM1(6,ZNSDF), GAM2(6,ZNDDF)
C
C K - WHICH NUMBER ELEMENT WORKING ON
C [R],[T],[M],[GAM1],[GAM2] - ELEMENTAL EQUATIONS
C
C*COMDECK FULLEQS
C---------------
      PARAMETER (NSS=ZNSDF*ZNSDF, NS4=ZNSDF*4, NDD=ZNDDF*ZNDDF)
      REAL*8 KC, MC, KC11, KC12, KC21, KC22, MC11
      COMMON /FULLEQS/ KC(NSS), PT(NS4), MC(NDD), C(NSS), CT(NS4),
     *                KC11(NSS), KC12(NSS),
                       KC21(NSS), KC22(NSS), MC11(NDD),
                       PT1(NS4), PT2(NS4)
C
C*COMDECK LENS
C------------
      REAL*8 LEN
      COMMON /LENS/ LEN(ZNE)
C
C LEN - LENGTH OF EACH ELEMENT
C
C*COMDECK NUMBER
C--------------
      COMMON /NUMBER/ ITRUSS, OMEGA, NN, NE, NSDF, NDDF, NPT, NCN, NDIMT
      CHARACTER*80 TITLE
      COMMON /TITLEC/ TITLE
C
C TITLE  - TITLE OF DATA
C ITRUSS - =1, BEAMLIKE  LATTICE TRUSS
C          =2, PLATELIKE LATTICE TRUSS
C OMEGA  - =LENGTH OF REPEATING CELL FOR BEAMLIKE LATTICE (ITRUSS=1)
C          =PLANFORM AREA OF REPEATING CELL FOR PLATELIKE LATTICE (ITRUSS=2)
C NN     - NUMBER OF NODES
C NE     - NUMBER OF ELEMENTS
C NSDF   - NUMBER OF STRAIN DEGREES OF FREEDOM
C NDDF   - NUMBER OF DISPL. DEGREES OF FREEDOM
C NPT    - NUMBER OF PROPERTY TYPES
C NCN    - NUMBER OF NODES CONNECTED IN EACH ELEMENT
C NDIMT  - DIMENSION OF ELEMENTAL MATRIX [T], EITHER=4 OR 2
C
C*COMDECK PROP
C------------
      COMMON /PROP/ AREA(ZNPT), E(ZNPT), RHO(ZNPT), ALPHA(ZNPT),
                   IPROP(ZNE)
C
C AREA  - CROSS-SECTIONAL AREA OF STRUCTURE
C E     - YOUNG'S MODULUS
C RHO   - MASS DENSITY
C ALPHA - COEFFICIENT OF THERMAL EXPANSION
C IPROP - FOR EACH ELEMENT, WHICH PROPERTY TYPE
C
<< File GLOBCONS.CMN follows >>
        common /globcons/  SOLID, DASHED, DOTTED, LEFT, RIGHT, MIDDLE,
*                          ARROW, CROSS, BLOCK, POINT, HRGLAS,
*                          YES, NO, ON, OFF, TRUE, FALSE,
*                          BLACK, BLUE, GREEN, CYAN, RED, MAGENT,
*                          BROWN, LGRAY, DGRAY, LBLUE, LGREEN, LCYAN,
*                          LRED, LMAGEN, YELLOW, WHITE, ULINE
        integer SOLID, DASHED, DOTTED, RIGHT, ARROW, CROSS, BLOCK,
*               POINT, HRGLAS, YES, ON, OFF, TRUE, FALSE,
*               BLACK, BLUE, GREEN, CYAN, RED, BROWN, DGRAY, YELLOW,
*               WHITE, ULINE


<< file CKCOLOR.CMN follows >>
c
```

```
c BG=BACKGROUND, FG=FOREGROUND, T=TEXT, F=FIELD, W=WINDOW,
c ICOLOR=1 FOR COLOR MONITOR, 0 FOR MONOCHROME
c
    common /ckcolor/ IBGT, IFGT, IBGF, IFGF, IBGTW, IFGTW,
 *                    IBGFW, IFGFW, ICOLOR
```

```
$ STORAGE:2
C*********************************************************************
C CONTINUUM MODEL FOR REPETITIVE LATTICE STRUCTURES:
C
C           BEAMLIKE LATTICE TRUSSES
C           PLATELIKE LATTICE TRUSSES
C
C *** USES DOUBLE PRECISION ***
C
C GENERAL EQUATIONS:
C-----------
C
C FOR EACH ELEMENT -
C
C     [R] = [L*L, M*M, N*N, L*M, L*N, M*N]
C                                                [1 0 0]
C     [M] = 1/6 * RHO*AREA*LEN * [2*I3     I3]   I3=[0 1 0]
C                                [  I3  2*I3],      [0 0 1]
C
C     [T] = [1, YC, ZC, YC*ZC], BEAMLIKE
C     [T] = [1, ZC]           , PLATELIKE
C
C     [GAM1], DEPENDS ON BEAM-OR-PLATE, CHOICE OF
C     [GAM2], EC1,EC2,EC3,UC1,UC2
C
C
C [KC] = SUM OF ELEMENTS ([R]*[GAM1])T * ([R]*[GAM1]
C [PT] = SUM OF ELEMENTS ([R]*[GAM1])T * [T]
C [MC] = SUM OF ELEMENTS 1/OMEGA * [GAM2]T*[M]*[GAM2]
C
C [KC] = [ KC11  KC12  KC13 ]       [MC] = [MC11 MC12]
C        [       KC22  KC23 ]              [MC21 MC22]
C        [ SYM.        KC33 ]
C
C [C]  = 1/OMEGA * [KC11 - (KC12*KC22$^{-1}$) * KC21]
C
C [CT] = 1/OMEGA * [PT1] - [KC12*KC22$^{-1}$] * [PT2]
C
C
C FILES :
C -------
C     UNIT 5  - SCREEN INPUT
C     UNIT 6  - PROBLEM OUTPUT
C     UNIT 7  - DEBUG FILE (NO LONGER USED)
C     UNIT 8  - FILE TO READ DATA (OR TO WRITE DATA AND RESULTS)
C**********************************************************************
$INCLUDE:'DBLE'

C     unit 5 - screen
C     unit 6 - screen
      OPEN (5,FILE='CON:')
      OPEN (6,FILE='CON:')

c       initialize key definitions and colors .
```

```
      CALL INITIAL

c     . write cover message .
      CALL MSSGE

c     . initial variables and set defaults .
      CALL DEFAULT

c     . start main job .
      CALL MAIN

c     . write ending message .
      CALL SAYBYE
      END


      SUBROUTINE MSSGE
C***********************************************************************
C     Writes cover message and any instructions
C***********************************************************************
$INCLUDE:'DBLE'
$INCLUDE:'GLOBCONS.CMN'
$INCLUDE:'CKCOLOR.CMN'
      CALL QCLEAR (BLUE,YELLOW)
      CALL QBORD (GREEN)
      CALL QCMOV (0,24)

      CALL JLABEL (12, 20, 'Continuum Models for Repetitive Lattice Stru
     .ctures $')
      CALL JLABEL (12, 19, '--------------------------------------------
     .------ $')
      CALL JLABEL ( 5, 17, 'Welcome ! This program will compute the foll
     .owing matrices : $')
      CALL JLABEL (15, 15,'stiffness matrix of equivalent continuum,
     .[C] $')
      CALL JLABEL (15, 14, 'thermal coefficients of equivalent continuum
     ., [CT] $')
      CALL JLABEL (15, 13, 'matrix of density parameters of equivalent c
     .ontinuum, [m]$')
      CALL JLABEL ( 5, 11, 'for the following structures : $')
      CALL JLABEL (15,  9, 'beamlike lattice trusses $')
      CALL JLABEL (15,  8, 'platelike lattice trusses $')

      CALL QCMOV (0,2)
      PAUSE
      RETURN
      END


      SUBROUTINE DEFAULT
C***********************************************************************
C     Sets defaults (or do initializing) for data
C***********************************************************************
$INCLUDE:'DBLE'
$INCLUDE:'PARAM.CMN'
```

```
$INCLUDE:'DATA.CMN'

C     --- data set 1 ---
      ITRUSS = -1
      OMEGA  = -1
      NN     = -1
      NE     = -1
      NPT    = -1
      NCN    = 2
      NSDF   = 0
      NDDF   = 0

C     --- data set 2 ---
      DO 10 I = 1, ZNE
      DO 10 J = 1, ZNCN
         NOP(I,J) = 0
  10  CONTINUE
      DO 20 I = 1, ZNE
         IPROP(I) = 1
  20  CONTINUE
      DO 30 I = 1, ZNPT
         AREA(I) = 0.
         E(I)    = 0.
         RHO(I)  = 0.
         ALPHA(I)= 0.
  30  CONTINUE

C     --- data set 3 ---
      NEC1 = 0
      NEC2 = 0
      NEC3 = 0
      NUC1 = 0
      NUC2 = 0
      DO 40 I = 1, ZNSDF
         EC(I) = I
  40  CONTINUE
      DO 50 I = 1, ZNDDF
         UC(I) = I
  50  CONTINUE

C     --- data set 4 ---
      DO 60 I = 1, ZNN
         XCOORD(I) = 0.
         YCOORD(I) = 0.
         ZCOORD(I) = 0.
  60  CONTINUE
      RETURN
      END

      SUBROUTINE MAIN
C*****************************************************************
C
C     Main subroutine to control outer menu and response
C
C*****************************************************************
```

```
$INCLUDE:'DBLE'
$INCLUDE:'GLOBCONS.CMN'
$INCLUDE:'CKCOLOR.CMN'
      CHARACTER CHOICE*5, YN*1, FIRST*1
      INTEGER INUM(1)
      COMMON /LINES/ LINES(10)
      DATA CHOICE / '     '/

 1    CALL MENU
 2    CALL CLFIELD (9,24,71,IBGF,IFGF)
      CALL JLABEL (9,24,'What do you want to do ? $')
      CALL CKFIELD (34,24,5,CHOICE,'CH')

C     convert 1st character to ascii to see if number
      FIRST = CHOICE(1:1)
      CALL VTOASC (INUM(1), 1, 'CH', FIRST)
      INUM(1) = INUM(1) - 48
      IF (INUM(1).GE.1 .AND. INUM(1).LE.5) THEN
         CALL GOSET(INUM(1))
         GOTO 1
      ELSE IF (CHOICE .EQ. 'B' .OR. CHOICE .EQ. 'b') THEN
         CALL EQSBEAM
         GOTO 1
      ELSE IF (CHOICE .EQ. 'P' .OR. CHOICE .EQ. 'p') THEN
         CALL EQSPLAT
         GOTO 1
      ELSE IF (CHOICE .EQ. 'F' .OR. CHOICE .EQ. 'f') THEN
         CALL FETCH
         GOTO 1
      ELSE IF (CHOICE .EQ. 'R' .OR. CHOICE .EQ. 'r') THEN
         CALL RUN
         GOTO 1
      ELSE IF (CHOICE .EQ. 'S' .OR. CHOICE .EQ. 's') THEN
         CALL SAVERUN
         GOTO 1
      ELSE IF (CHOICE .EQ. 'QUIT ' .OR. CHOICE .EQ. 'quit ') THEN
         LINES(1) = 0
         CALL CLFIELD (9,24,71,IBGF,IFGF)
         CALL JLABEL(9,24, 'Are you sure you want to quit (Y/N) ? $')
         CALL CKFIELD(50,24,1,YN,'CH')
         IF (YN .NE. 'Y' .AND. YN .NE. 'y') GOTO 2
      ELSE
         LINES(1) = 0
         CALL ERRORS('That was not a choice ! Hit enter & try again.$')
         CALL WAIT
         GOTO 2
      ENDIF
      RETURN
      END
      SUBROUTINE MENU
C*****************************************************************
C
C     subroutine to print the menu screen
C
C*****************************************************************
$INCLUDE:'DBLE'
$INCLUDE:'GLOBCONS.CMN'
$INCLUDE:'CKCOLOR.CMN'

      CALL JCLEAR
      CALL JLABEL(0,24,'MENU - $')
      CALL JLABEL(0,20,'ENTER : $')
      CALL JLABEL(9,19,'EITHER number of data set to EDIT data$')

      DO 100 IROW = 17,13,-1
         CALL CLFIELD(13,IROW,53,BLACK,YELLOW)
 100  CONTINUE
      CALL JLABEL(15,17, 'SET # 1 --- DESCRIPTION OF TRUSS$')
      CALL JLABEL(15,16, 'SET # 2 --- ELEMENT DATA & PROPERTY TYPES$')
      CALL JLABEL(15,15, 'SET # 3 --- COORDINATES$')
      CALL JLABEL(15,14, 'SET # 4 --- CONTINUUM STRAIN PARAMETERS$')
      CALL JLABEL(15,13, 'SET # 5 --- CONTINUUM DISPL.  PARAMETERS$')

      CALL JLABEL(9,10,'OR  $')
      CALL JLABEL(9, 8,'"b"    - to display equations for BEAMlike latti
     .ce truss $')
      CALL JLABEL(9, 7,'"p"    - to display equations for PLATElike latt
     .ice truss $')
      CALL JLABEL(9, 6,'"f"    - to FETCH data from a file $')
      CALL JLABEL(9, 5,'"r"    - to RUN data $')
      CALL JLABEL(9, 4,'"s"    - to SAVE data and results to file $')
      CALL JLABEL(9, 3,'"quit" - to QUIT $')
      RETURN
      END


      SUBROUTINE WAIT
C***********************************************************************
C     waits until <enter> is hit
C***********************************************************************
      CHARACTER CH*1
      CALL CKFIELD (78, 24, 1, CH, 'CH')
      RETURN
      END


      SUBROUTINE EQSBEAM
C***********************************************************************
C     displays equations assumed for beamlike lattice truss
C***********************************************************************
$INCLUDE:'DBLE'
$INCLUDE:'GLOBCONS.CMN'
$INCLUDE:'CKCOLOR.CMN'
```

```
      CALL JCLEAR
      CALL JLABEL (15,22, '***** EQUATIONS for BEAMLIKE LATTICE TRUSS **
     .*** $')
      CALL JLABEL (0, 20, 'Displacement Assumptions $')
      CALL JLABEL (0, 19, '------------------------ $')
      CALL JLABEL (0, 18, 'u = uo - y*phiz  + z*phiy + y*z*ubar $')
      CALL JLABEL (0, 17, 'v = vo + y*epsyo + z*(- phix+gammayzo/2) + y*z
     .*vbar $')
      CALL JLABEL (0, 16, 'w = wo + z*epszo + y*(phix+gammayzo/2)   + y*z
     .*wbar $')

      CALL JLABEL (0, 14, 'Strain Assumptions $')
      CALL JLABEL (0, 13, '------------------ $')
      CALL JLABEL (0, 12, 'eps11 = epsxo - y*kappayo + z*kappazo + y*z*p
     .si$')
      CALL JLABEL (0, 11, 'eps22 = epsyo + z*vbar $')
      CALL JLABEL (0, 10, 'eps33 = epszo + y*wbar $')
      CALL JLABEL (0,  9, 'gamma12 = gammaxyo + y*d(epsyo)/dx + z*(-kapp
     .ato+kappabr) + y*z*d(vbar)/dx $')
      CALL JLABEL (0,  8, 'gamma13 = gammaxzo + z*d(epszo)/dx + y*( kapp
     .ato+kappabr) + y*z*d(wbar)/dx $')
      CALL JLABEL (0,  7, 'gamma23 = gammayzo + y*vbar + z*wbar $')

      CALL JLABEL (0,  5, 'Temperature Assumptions $')
      CALL JLABEL (0,  4, '----------------------- $')
      CALL JLABEL (0,  3, 'T = To + y*d(To)/dy + z*d(To)/dz + yz*d2(To)/
     .dydz $')

      CALL CLFIELD (0,24,80,IBGF,IFGF)
      CALL JLABEL (0,24,'Hit enter to continue ... $')
      CALL WAIT
      CALL JCLEAR
      RETURN
      END


      SUBROUTINE EQSPLAT
C***********************************************************************
C    displays equations assumed for platelike lattice truss
C***********************************************************************
$INCLUDE:'DBLE'
$INCLUDE:'GLOBCONS.CMN'
$INCLUDE:'CKCOLOR.CMN'

      CALL JCLEAR
      CALL JLABEL (15,22, '***** EQUATIONS for PLATELIKE LATTICE TRUSS *
     .**** $')
      CALL JLABEL (0, 20, 'Displacement Assumptions $')
      CALL JLABEL (0, 19, '------------------------ $')
      CALL JLABEL (0, 18, 'u = uo + z*phix $')
      CALL JLABEL (0, 17, 'v = vo + z*phiy $')
      CALL JLABEL (0, 16, 'w = wo + z*epsz $')

      CALL JLABEL (0, 14, 'Strain Assumptions $')
      CALL JLABEL (0, 13, '------------------ $')
      CALL JLABEL (0, 12, 'eps11 = epsxo + z*kappaxo $')
      CALL JLABEL (0, 11, 'eps22 = epsyo + z*kappayo $')
      CALL JLABEL (0, 10, 'eps33 = epszo              $')
      CALL JLABEL (0,  9, 'gamma12 = gammaxyo + z*2kappaxyo $')
      CALL JLABEL (0,  8, 'gamma13 = gammaxzo + z*d(epszo)/dx + x*d(gamm
     .axzo)/dx + y*d(gammaxzo)/dy $')
      CALL JLABEL (0,  7, 'gamma23 = gammayzo + z*d(epszo)/dy + x*d(gamm
     .ayzo)/dx + y*d(gammayzo)/dy $')

      CALL JLABEL (0,  5, 'Temperature Assumptions $')
      CALL JLABEL (0,  4, '---------------------- $')
      CALL JLABEL (0,  3, 'T = To + z*d(To)/dz      $')

      CALL CLFIELD (0,24,80,IBGF,IFGF)
      CALL JLABEL (0,24,'Hit enter to continue ... $')
      CALL WAIT
      CALL JCLEAR
      RETURN
      END


      SUBROUTINE FETCH
C***********************************************************************
C     Fetches a data set on a file
C     (data in same format as written to in 'SAVERUN')
C***********************************************************************
$INCLUDE:'DBLE'
$INCLUDE:'GLOBCONS.CMN'
$INCLUDE:'CKCOLOR.CMN'
$INCLUDE:'PARAM.CMN'
$INCLUDE:'DATA.CMN'
      CHARACTER FILENAM*12, STRNG1*38, STRNG2*8, STRNG3*17
      LOGICAL OLDFILE
      COMMON /LINES/ LINES(10)

c     --- read in name of file with data ---
      CALL JCLEAR
      CALL JLABEL (0,16,'Name of the file with data : $')
  5   CALL PRFIELD (40,16,12,'              ','CH',IBGF,IFGF)
      CALL CKFIELD (40,16,12,FILENAM,'CH')
      INQUIRE (FILE=FILENAM, EXIST=OLDFILE, IOSTAT=IERR)
      IF (.NOT.OLDFILE) THEN
         LINES(1) = 0
         CALL ERRORS ('That file does not exist! Hit enter. $')
         CALL WAIT
         RETURN
      ENDIF
      OPEN (8,FILE=FILENAM,FORM='FORMATTED',STATUS='OLD')

c     --- read data to file ---
      READ(8,6) STRNG1, ITRUSS
      READ(8,6) STRNG1, NN
      READ(8,6) STRNG1, NE
      READ(8,6) STRNG1, NPT
      READ(8,7) STRNG1, OMEGA
      READ(8,10) (I, (NOP(I,J), J=1,NCN), IPROP(I), I=1,NE)
      READ(8,20) (I, AREA(I), E(I), RHO(I), ALPHA(I), I=1,NPT)
      READ(8,30) STRNG2, NEC1
      READ(8,40) STRNG3, (EC(I), I=1,NEC1)
      READ(8,30) STRNG2, NEC2
      READ(8,40) STRNG3, (EC(I), I=NEC1+1,NEC1+NEC2)
      READ(8,30) STRNG2, NEC3
      READ(8,40) STRNG3, (EC(I), I=NEC1+NEC2+1,NSDF)
      READ(8,30) STRNG2, NUC1
      READ(8,40) STRNG3, (UC(I), I=1,NUC1)
      READ(8,30) STRNG2, NUC2
      READ(8,40) STRNG3, (UC(I), I=NUC1+1,NUC1+NUC2)
      READ(8,50) (J, XCOORD(J), YCOORD(J), ZCOORD(J), J=1,NN)
      CLOSE (8)
    6 FORMAT (A, I5)
    7 FORMAT (A, E17.10)
   10 FORMAT (1X/ (I5, 3X, 2I5, 10X, I5))
   20 FORMAT (1X/1X/ (I6,2X,4E17.10))
   30 FORMAT (A, I5)
   40 FORMAT (A, (20I4))
   50 FORMAT (1X/ (I5, 5X, 3E17.10))

C     --- set other variables ---
      IF (ITRUSS.EQ.1) THEN
         NSDF = 18
         NDDF = 12
      ELSE
         NSDF = 15
         NDDF =  6
      ENDIF

      CALL CLFIELD (0,24,80,IBGF,IFGF)
      CALL JLABEL (9,24, 'Data read in. Hit enter.$')
      CALL WAIT
      RETURN
      END


      SUBROUTINE SAVERUN
C*************************************************************************
C     saves data and results of run to a file
C*************************************************************************
$INCLUDE:'DBLE'
$INCLUDE:'GLOBCONS.CMN'
$INCLUDE:'CKCOLOR.CMN'
$INCLUDE:'PARAM.CMN'
$INCLUDE:'DATA.CMN'
      CHARACTER FILENAM*12, ANS*3
      LOGICAL OLDFILE
      COMMON /LINES/ LINES(10)

c     --- read in name to call file ---
      CALL JCLEAR
      CALL JLABEL (0,16,'Name of the file to save results to : $')
    5 CALL PRFIELD (40,16,12,'            ','CH',IBGF,IFGF)
      CALL CKFIELD (40,16,12,FILENAM,'CH')
      INQUIRE (FILE=FILENAM, EXIST=OLDFILE, IOSTAT=IERR)
      IF (OLDFILE) THEN
         LINES(1) = 0
         CALL ERRORS ('That file already exists! Hit enter and try again
     .$')
         CALL WAIT
         GOTO 5
      ENDIF
      OPEN (8,FILE=FILENAM,FORM='FORMATTED',STATUS='NEW')

c     --- write data to file ---
      WRITE(8,6) 'ITRUSS (Beamlike=1, Platelike=2)    = ', ITRUSS
      WRITE(8,6) 'Number of nodes in repeating cell   = ', NN
      WRITE(8,6) 'Number of connecting elements       = ', NE
      WRITE(8,6) 'Number of property types           = ', NPT
      WRITE(8,7) 'Characteristic dimension of cell    = ', OMEGA
      WRITE(8,10) (I, (NOP(I,J), J=1,NCN), IPROP(I), I=1,NE)
      WRITE(8,20) (I, AREA(I), E(I), RHO(I), ALPHA(I), I=1,NPT)
      WRITE(8,30) NEC1, (EC(I), I=1,NEC1)
      WRITE(8,31) NEC2, (EC(I), I=NEC1+1,NEC1+NEC2)
      WRITE(8,32) NEC3, (EC(I), I=NEC1+NEC2+1,NSDF)
      WRITE(8,40) NUC1, (UC(I), I=1,NUC1)
      WRITE(8,41) NUC2, (UC(I), I=NUC1+1,NUC1+NUC2)
      WRITE(8,50) (J, XCOORD(J), YCOORD(J), ZCOORD(J), J=1,NN)
      WRITE(8,*) ' '
    6 FORMAT (A, I5)
    7 FORMAT (A, E17.10)
   10 FORMAT ('OELEMENT      NOP          PROPERTY TYPE'/
     .       (I5, 3X, 2I5, 10X, I5))
   20 FORMAT ('OPROPERTY TYPE - '/
     .        ' TYPE NO.   AREA              E RHO
     .     ALPHA'/
     .       (I6,2X,4E17.10))
   30 FORMAT ('ONEC1 = ', I5/ ' VECTOR (EC1) IS ', (20I4))
   31 FORMAT ('ONEC2 = ', I5/ ' VECTOR (EC2) IS ', (20I4))
   32 FORMAT ('ONEC3 = ', I5/ ' VECTOR (EC3) IS ', (20I4))
   40 FORMAT ('ONUC1 = ', I5/ ' VECTOR (UC1) IS ', (20I4))
   41 FORMAT ('ONUC2 = ', I5/ ' VECTOR (UC2) IS ', (20I4))
   50 FORMAT ('ONODAL COORDINATES (X,Y,Z)'/ (I5, 5X, 3E17.10))

c     --- write results to file ---
      CALL OUTMATX (C,  NEC1, NEC1, NEC1,  10, 8, '[C] :     ')
      CALL OUTMATX (CT, NEC1, NEC1, NDIMT, 10, 8, '[CT] :    ')
      CALL OUTMATX (MC11,NUC1,NUC1, NUC1,  10, 8, '[m] :     ')
      CALL CLFIELD (0,24,80,IBGF,IFGF)
      CALL JLABEL (9,24, 'Data and results saved. Hit enter.$')
      CALL WAIT
      CLOSE(8)
      RETURN
      END


      SUBROUTINE SAYBYE
```

```
C************************************************************************
C     writes ending message and clears screen
C************************************************************************
$INCLUDE:'DBLE'
$INCLUDE:'GLOBCONS.CMN'
$INCLUDE:'CKCOLOR.CMN'

      CALL QCLEAR (BLUE, YELLOW)
      CALL QBORD (GREEN)
      CALL QCMOV (0,24)

      CALL JLABEL (19,10,'Hope this program has been helpful ! $')
      CALL JLABEL (28, 8,'Have a good day ! $')
      CALL QCMOV (0,2)
      PAUSE
      CALL QCLEAR (BLACK, WHITE)
      CALL QBORD (BLACK)
      CALL QCMOV (0,24)
      RETURN
      END
```

```
$STORAGE:2
      SUBROUTINE GOSET(ISET)
C*************************************************************************
C
C     Subroutine to control data set number ISET
C
C*************************************************************************
$INCLUDE:'DBLE'
      CHARACTER*9 SET(5)

      SET(1) = 'SET 1 : $'
      SET(2) = 'SET 2 : $'
      SET(3) = 'SET 3 : $'
      SET(4) = 'SET 4 : $'
      SET(5) = 'SET 5 : $'
      IF (ISET .LT. 1 .OR. ISET .GT. 5) THEN
         WRITE(5,*) 'ERROR ! ISET = ',ISET
         STOP
      ENDIF
      CALL JCLEAR
      CALL JLABEL(0,24,SET(ISET))

      GOTO (1,2,3,4,5) ISET
    1 CALL EDSET1
      GOTO 90

    2 CALL EDSET2
      GOTO 90

    3 CALL EDSET3
      GOTO 90

    4 CALL EDSET4
      GOTO 90

    5 CALL EDSET5
   90 RETURN
      END

      SUBROUTINE EDSET1
C*************************************************************************
C     edits data set 1 - description of truss
C*************************************************************************
$INCLUDE:'DBLE'
$INCLUDE:'GLOBCONS.CMN'
$INCLUDE:'CKCOLOR.CMN'
$INCLUDE:'PARAM.CMN'
$INCLUDE:'DATA.CMN'
      COMMON /LINES/ LINES(10)

      CALL JLABEL (0,18,'1 - Beamlike truss=1, Platelike truss=2 $')
      CALL JLABEL (0,16,'2 - Number of nodes in typical repeating cell,
     .NN         (max=18) $')
      CALL JLABEL (0,14,'3 - Number of connecting elements in repeating
     .cell, NE   (max=60) $')
```

```
      CALL JLABEL (0,12,'4 - Number of property types, NPT
     .           (max=8)  $')
      CALL JLABEL (0,10,'5 - Characteristic dimension of typical repeati
     .ng cell             $')
      CALL JLABEL (0, 9,'  (length for beamlike, planform area for plate
     .like lattice truss) $')

      ICOL = 68
      IF (ITRUSS.EQ.-1) CALL PRFIELD (ICOL, 18, 3,'    ', 'CH',IBGF,IFGF)
      IF (NN.EQ.-1)     CALL PRFIELD (ICOL, 16, 3,'    ', 'CH',IBGF,IFGF)
      IF (NE.EQ.-1)     CALL PRFIELD (ICOL, 14, 3,'    ', 'CH',IBGF,IFGF)
      IF (NPT.EQ.-1)    CALL PRFIELD (ICOL, 12, 3,'    ', 'CH',IBGF,IFGF)
      IF (OMEGA.EQ.-1)  CALL PRFIELD (ICOL, 10,10,' ','CH',
                         IBGF,IFGF)

      IF (ITRUSS.NE.-1) CALL PRFIELD (ICOL, 18, 3,ITRUSS,'I2',IBGF,IFGF)
      IF (NN.NE.-1)     CALL PRFIELD (ICOL, 16, 3, NN,'I2',IBGF,IFGF)
      IF (NE.NE.-1)     CALL PRFIELD (ICOL, 14, 3, NE,'I2',IBGF,IFGF)
      IF (NPT.NE.-1)    CALL PRFIELD (ICOL, 12, 3, NPT,'I2',IBGF,IFGF)
      IF (OMEGA.NE.-1)  CALL PRFIELD (ICOL, 10, 10,OMEGA,'R8',IBGF,IFGF)

  10  CALL CKFIELD (ICOL, 18, 3, ITRUSS, 'I2')
      IF (ITRUSS.NE.1 .AND. ITRUSS.NE.2) THEN
         LINES(1) = 0
         CALL ERRORS ('Please input 1 or 2. Hit enter to redo.  $')
         CALL WAIT
         GOTO 10
      ENDIF
      IF (ITRUSS.EQ.1) THEN
         NSDF = 18
         NDDF = 12
      ELSE
         NSDF = 15
         NDDF =  6
      ENDIF

  20  CALL CKFIELD (ICOL, 16, 3, NN, 'I2')
      IF (NN.LT.1 .OR. NN.GT.ZNN) THEN
         LINES(1) = ZNN
         LINES(2) = 0
         CALL ERRORS ('Please redo: hit enter and input an integer betwe
     .en 1 and  $')
         CALL WAIT
         GOTO 20
      ENDIF

  30  CALL CKFIELD (ICOL, 14, 3, NE, 'I2')
      IF (NE.LT.1 .OR. NE.GT.ZNE) THEN
         LINES(1) = ZNE
         LINES(2) = 0
         CALL ERRORS ('Please redo: hit enter and input an integer betwe
     .en 1 and  $')
         CALL WAIT
         GOTO 30
      ENDIF
  40  CALL CKFIELD (ICOL, 12, 3, NPT, 'I2')
      IF (NPT.LT.1 .OR. NPT.GT.ZNPT) THEN
         LINES(1) = ZNPT
         LINES(2) = 0
         CALL ERRORS ('Please redo: hit enter and input an integer betwe
     .en 1 and  $')
         CALL WAIT
         GOTO 40
      ENDIF

  50  CALL CKFIELD (ICOL, 10, 10, OMEGA, 'R8')
      RETURN
      END


      SUBROUTINE EDSET2
C***********************************************************************
C     edits data set 2 - connectivities and prop.types
C***********************************************************************
$INCLUDE:'DBLE'
$INCLUDE:'GLOBCONS.CMN'
$INCLUDE:'CKCOLOR.CMN'
$INCLUDE:'PARAM.CMN'
$INCLUDE:'DATA.CMN'
      PARAMETER (NCOLS1=3, NCOLS2=4)
      PARAMETER (NVAL1=ZNE*ZNCN+ZNE, NVAL2=ZNPT*NCOLS2)
      COMMON /LINES/ LINES(10)
      INTEGER I1COLSP(NCOLS1), FLP1(NCOLS1), VAL1(NVAL1)
     .       ,I2COLSP(NCOLS2), FLP2(NCOLS2)
      REAL*8 VAL2(NVAL2)
      CHARACTER TYPEP1(NCOLS1)*2, TYPEP2(NCOLS2)*2
      LOGICAL HOMO, ERR1, ERR2
      DATA I1COLSP /18,24,40/
      DATA FLP1 /3,3,3/

c     --- check that number of elements is defined ---
      IF (NE.EQ.-1) THEN
         LINES(1) = 0
         CALL ERRORS ('Please define NE in data set #1 first. Hit enter
     .to continue $')
         CALL WAIT
         RETURN
      ENDIF

c     --- print note ---
      CALL JLABEL (50,11, '       *** NOTE ***          $')
      CALL JLABEL (50,10, 'For elements shared by n     $')
      CALL JLABEL (50, 9, 'repeating cells, the cross-$')
      CALL JLABEL (50, 8, 'sectional areas are to be    $')
      CALL JLABEL (50, 7, 'divided by n.                $')

c     --- property table ---
      CALL JLABEL (14, 22, 'cross sectional  Young's modulus mass dens
     .ity  coefficient of thermal$')
```

```
      CALL JLABEL (0,21,'Prop.type no.                      Area E
     .   Rho        expansion Alpha$')
      DO 30 I = 1, NPT
         VAL2(I)       = AREA(I)
         VAL2(I+NPT)   = E(I)
         VAL2(I+2*NPT) = RHO(I)
         VAL2(I+3*NPT) = ALPHA(I)
   30 CONTINUE
      HOMO = .TRUE.
      TYPEP2(1)  = 'R8'
      I2COLSP(1) = 18
      FLP2(1)  = 14
      NROWS   = NPT
      NLINES  = 4
      CALL PRTABL (0, 17, NLINES, NROWS, NCOLS2, I2COLSP, FLP2, VAL2,
     .             TYPEP2, HOMO)
      CALL CKTABL (0, 17, NLINES, NROWS, NCOLS2, I2COLSP, FLP2, VAL2,
     .             TYPEP2, HOMO)
      DO 40 I = 1, NPT
         AREA(I) = VAL2(I)
         E(I)    = VAL2(I+NPT)
         RHO(I)  = VAL2(I+2*NPT)
         ALPHA(I)= VAL2(I+3*NPT)
   40 CONTINUE

c     --- element table ---
      CALL JLABEL (0,14,'Element no.  Connectivities   Property type no.
     .$')
      DO 10 I = 1, NE
         VAL1(I) = NOP(I,1)
         VAL1(I+NE) = NOP(I,2)
         VAL1(I+2*NE) = IPROP(I)
   10 CONTINUE
      HOMO = .FALSE.
      DO 20 I = 1, NCOLS1
         TYPEP1(I) = 'I2'
   20 CONTINUE
      NROWS = NE
      NLINES = 10
   22 CALL PRTABL (0, 4, NLINES, NROWS, NCOLS1, I1COLSP, FLP1, VAL1,
     .             TYPEP1, HOMO)
      CALL CKTABL (0, 4, NLINES, NROWS, NCOLS1, I1COLSP, FLP1, VAL1,
     .             TYPEP1, HOMO)
      ERR1 = .FALSE.
      ERR2 = .FALSE.
      DO 25 I = 1, NE
         NOP(I,1) = VAL1(I)
         NOP(I,2) = VAL1(I+NE)
         IPROP(I) = VAL1(I+2*NE)
         IF (NOP(I,1).LT.1 .OR. NOP(I,1).GT.NN .OR.
     .       NOP(I,2).LT.1 .OR. NOP(I,2).GT.NN)  ERR1=.TRUE.
         IF (IPROP(I).LT.1 .OR. IPROP(I).GT.NPT) ERR2=.TRUE.
   25 CONTINUE
      IF (ERR1) THEN
         LINES(1) = NN
         LINES(2) = 0
         CALL ERRORS ('Please redo: connectivity numbers should be integ
     .ers between 1 and  $')
         CALL WAIT
         GOTO 22
      ENDIF
      IF (ERR2) THEN
         LINES(1) = NPT
         LINES(2) = 0
         CALL ERRORS ('Please redo: property numbers should be integers
     .between 1 and  $')
         CALL WAIT
         GOTO 22
      ENDIF
      RETURN
      END


      SUBROUTINE EDSET3
C***********************************************************************
C     edits data set 3 - coordinates
C***********************************************************************
$INCLUDE:'DBLE'
$INCLUDE:'GLOBCONS.CMN'
$INCLUDE:'CKCOLOR.CMN'
$INCLUDE:'PARAM.CMN'
$INCLUDE:'DATA.CMN'
      PARAMETER (NCOLS=3, NVAL=NCOLS*ZNN)
      INTEGER ICOLSP(NCOLS), FLP(NCOLS)
      CHARACTER TYPEP(NCOLS)*2
      REAL*8 VAL(NVAL)
      LOGICAL HOMO
      COMMON /LINES/ LINES(10)

c     --- check that NN is defined ---
      IF (NN.EQ.-1) THEN
         LINES(1) = 0
         CALL ERRORS ('Please define NN in data set #1 first. Hit enter
     .to continue $')
         CALL WAIT
         RETURN
      ENDIF

c     --- table of coordinates ---
      CALL JLABEL (0, 22, 'Coordinates $')
      CALL JLABEL (0, 21, 'node no.        X                      Y Z $')
      CALL JLABEL (46,18, 'For Beamlike Lattice  $')
      CALL JLABEL (46,17, '--------------------  $')
      CALL JLABEL (46,16, 'X   = coordinate along the length$')
      CALL JLABEL (46,15, 'Y,Z = coordinates in the plane  $')
      CALL JLABEL (52,14, 'of the cross section $')
      CALL JLABEL (46,12, 'For Platelike Lattice  $')
      CALL JLABEL (46,11, '---------------------  $')
      CALL JLABEL (46,10, 'X,Y = surface coordinates       $')
      CALL JLABEL (46, 9, 'Z   = coordinate in the         $')
      CALL JLABEL (52, 8, 'transverse direction $')
      CALL JLABEL (46, 6, 'NOTE : origin must be at center $')
      CALL JLABEL (53, 5, 'of repeating cell $')
      DO 10 I = 1, NN
        VAL(I)      = XCOORD(I)
        VAL(I+NN)   = YCOORD(I)
        VAL(I+2*NN) = ZCOORD(I)
  10  CONTINUE
      HOMO = .TRUE.
      TYPEP(1)  = 'R8'
      ICOLSP(1) = 13
      FLP(1)    = 10
      NROWS  = NN
      NLINES = 18
  20  CALL PRTABL (0,3,NLINES,NROWS,NCOLS,ICOLSP,FLP,VAL,TYPEP,HOMO)
      CALL CKTABL (0,3,NLINES,NROWS,NCOLS,ICOLSP,FLP,VAL,TYPEP,HOMO)
      DO 30 I = 1, NN
        XCOORD(I) = VAL(I)
        YCOORD(I) = VAL(I+NN)
        ZCOORD(I) = VAL(I+2*NN)
  30  CONTINUE
      RETURN
      END


      SUBROUTINE EDSET4
C***********************************************************************
C     edits data set 4 - continuum strain parameters
C***********************************************************************
$INCLUDE:'DBLE'
$INCLUDE:'GLOBCONS.CMN'
$INCLUDE:'CKCOLOR.CMN'
$INCLUDE:'PARAM.CMN'
$INCLUDE:'DATA.CMN'
      PARAMETER (NCOLS=3, NVAL=NCOLS*ZNSDF)
      INTEGER ICOLSP(NCOLS), FLP(NCOLS), VAL(NVAL)
      CHARACTER TYPEP(NCOLS)*2
      LOGICAL HOMO, ERR
      COMMON /LINES/ LINES(10)

c     --- check that itruss is defined ---
      IF (ITRUSS.EQ.-1) THEN
        LINES(1) = 0
        CALL ERRORS ('Please define ITRUSS in data set #1 first.  Hit en
     .ter to continue $')
        CALL WAIT
        RETURN
      ENDIF

c     --- set defaults according to itruss ---
      IF (ITRUSS.EQ.1) THEN
        NEC1 = 6
        NEC2 = 7
        NEC3 = 5
      ELSE IF (ITRUSS.EQ.2) THEN
```

```
         NEC1 = 8
         NEC2 = 5
         NEC3 = 2
      ENDIF

c     --- list strain parameters for itruss=1 ---
      CALL JLABEL (0, 22, 'Each strain parameter will be identified with
     . a number :$')
      IF (ITRUSS.EQ.1) THEN
      CALL JLABEL (2, 21, 'epsxo  =1 gammaxyo=4 kappabr =7 epsyo=10 wbar
     .       =13 d(gammayzo)/dx=16$')
      CALL JLABEL (2, 20, 'kappayo=2 gammaxzo=5 gammayzo=8 epszo=11 d(ep
     .syo)/dx=14 d(vbar)/dx    =17$')
      CALL JLABEL (2, 19, 'kappazo=3 kappat  =6 psio    =9 vbar =12 d(ep
     .szo)/dx=15 d(wbar)/dx    =18$')
      CALL JLABEL (55, 9, 'Defaults correspond $')
      CALL JLABEL (55, 8, 'to Timoshenko-type  $')
      CALL JLABEL (55, 7, 'beam theory         $')
      ENDIF

c     --- list strain parameters for itruss=2 ---
      IF (ITRUSS.EQ.2) THEN
      CALL JLABEL (2, 21, 'epsxo   =1 kappaxo  =4 gammaxzo      =7 d(gam
     .maxzo)/dy=10 epszo      =13$')
      CALL JLABEL (2, 20, 'epsyo   =2 kappayo  =5 gammayzo      =8 d(gam
     .mayzo)/dx=11 d(epszo)/dx=14$')
      CALL JLABEL (2, 19, 'gammaxyo=3 2kappaxyo=6 d(gammaxzo)/dx=9 d(gam
     .mayzo)/dy=12 d(epszo)/dy=15$')
      CALL JLABEL (55, 9, 'Defaults correspond $')
      CALL JLABEL (55, 8, 'to Reissner-Mindlin $')
      CALL JLABEL (55, 7, 'type beam theory    $')
      ENDIF

c     --- edit nec1,nec2,nec3 ---
      CALL JLABEL (0, 17, 'Number of strain parameters retained, NEPSC1$
     .')
      CALL JLABEL (0, 16, 'Number of strain parameters associated w/ neg
     .lected forces, NEPSC2 $')
      CALL JLABEL (0, 15, 'Number of strain parameters neglected, NEPSC3
     . $')
      ICOL = 67
 10   CALL PRFIELD (ICOL, 17, 2, NEC1, 'I2', IBGF, IFGF)
      CALL PRFIELD (ICOL, 16, 2, NEC2, 'I2', IBGF, IFGF)
      CALL PRFIELD (ICOL, 15, 2, NEC3, 'I2', IBGF, IFGF)
      CALL CKFIELD (ICOL, 17, 2, NEC1, 'I2')
      CALL CKFIELD (ICOL, 16, 2, NEC2, 'I2')
      CALL CKFIELD (ICOL, 15, 2, NEC3, 'I2')
      LINES(1) = NSDF
      LINES(2) = 0
      IF (NEC1.LT.1 .OR. NEC1.GT.NSDF) THEN
         CALL ERRORS ('Please redo: hit enter and input an integer betwe
     .en 1 and  $')
         CALL WAIT
         GOTO 10
      ENDIF
      IF (NEC2.LT.0 .OR. NEC2.GT.NSDF .OR.
     .    NEC3.LT.0 .OR. NEC3.GT.NSDF) THEN
         CALL ERRORS ('Please redo: hit enter and input an integer betwe
     .en 0 and  $')
         CALL WAIT
         GOTO 10
      ENDIF
      IF (NEC1+NEC2+NEC3.NE.NSDF) THEN
         CALL ERRORS ('Please redo: NEPSC1+NEPSC2+NEPSC3 should equal  $
     .')
         CALL WAIT
         GOTO 10
      ENDIF

c     --- table of strain parameters ---
      CALL JLABEL (0, 13, 'Choose which strain parameters are in each gr
     .oup $')
      CALL JLABEL (0, 12, 'No. parameter  retained assoc.w/negl          neg
     .lected $')
      DO 15 I = 1, NVAL
         VAL(I) = 0
  15  CONTINUE
      DO 20 I = 1, NEC1
         VAL(I) = EC(I)
  20  CONTINUE
      DO 30 I = 1, NEC2
         VAL(I+NSDF) = EC(I+NEC1)
  30  CONTINUE
      DO 40 I = 1, NEC3
         VAL(I+2*NSDF) = EC(I+NEC1+NEC2)
  40  CONTINUE
      HOMO = .FALSE.
      DO 50 I = 1, NCOLS
         TYPEP(I) = 'I2'
         ICOLSP(I) = 20+(I-1)*14
         FLP(I)    = 2
  50  CONTINUE
      NROWS  = NSDF
      NLINES = 10
  55  CALL PRTABL (0,2,NLINES,NROWS,NCOLS,ICOLSP,FLP,VAL,TYPEP,HOMO)
      CALL CKTABL (0,2,NLINES,NROWS,NCOLS,ICOLSP,FLP,VAL,TYPEP,HOMO)
      DO 60 I=1,NEC1
         EC(I) = VAL(I)
  60  CONTINUE
      DO 70 I = 1, NEC2
         EC(I+NEC1) = VAL(I+NSDF)
  70  CONTINUE
      DO 80 I = 1, NEC3
         EC(I+NEC1+NEC2) = VAL(I+2*NSDF)
  80  CONTINUE
      ERR=.FALSE.
      DO 90 I = 1, NSDF
         IF (EC(I).LT.1 .OR. EC(I).GT.NSDF) ERR=.TRUE.
  90  CONTINUE
      IF (ERR) THEN
```

```
         LINES(1) = NSDF
         LINES(2) = 0
         CALL ERRORS ('Please redo : table numbers should be integers be
     .tween 1 and  $')
         CALL WAIT
         GOTO 55
      ENDIF
      RETURN
      END


      SUBROUTINE EDSET5
C***************************************************************************
C     edits data set 5 - continuum displ. parameters
C***************************************************************************
$INCLUDE:'DBLE'
$INCLUDE:'GLOBCONS.CMN'
$INCLUDE:'CKCOLOR.CMN'
$INCLUDE:'PARAM.CMN'
$INCLUDE:'DATA.CMN'
      PARAMETER (NCOLS=2, NVAL=NCOLS*ZNDDF)
      INTEGER ICOLSP(NCOLS), FLP(NCOLS), VAL(NVAL)
      CHARACTER TYPEP(NCOLS)*2
      LOGICAL HOMO, ERR
      COMMON /LINES/ LINES(10)

c     --- check that itruss is defined ---
      IF (ITRUSS.EQ.-1) THEN
         LINES(1) = 0
         CALL ERRORS ('Please define ITRUSS in data set #1 first.  Hit en
     .ter to continue $')
         CALL WAIT
         RETURN
      ENDIF

c     --- set defaults according to itruss ---
      IF (ITRUSS.EQ.1) THEN
         NUC1 = 6
         NUC2 = 6
      ELSE IF (ITRUSS.EQ.2) THEN
         NUC1 = 5
         NUC2 = 1
      ENDIF

c     --- list displacement parameters for itruss=1 ---
      CALL JLABEL (0, 22, 'Each displacement parameter will be identifie
     .d with a number :$')
      IF (ITRUSS.EQ.1) THEN
      CALL JLABEL (5, 21, 'uo=1  phix=4   ubar    =7   epszo=10 $')
      CALL JLABEL (5, 20, 'vo=2  phiy=5   gammayzo=8   vbar =11 $')
      CALL JLABEL (5, 19, 'wo=3  phiz=6   epsyo   =9   wbar =12 $')
      CALL JLABEL (55, 9, 'Defaults correspond $')
      CALL JLABEL (55, 8, 'to Timoshenko-type  $')
      CALL JLABEL (55, 7, 'beam theory          $')
      ENDIF
```

```
c     --- list strain parameters for itruss=2 ---
      IF (ITRUSS.EQ.2) THEN
      CALL JLABEL (5, 21, 'uo=1   phix=4 $')
      CALL JLABEL (5, 20, 'vo=2   phiy=5 $')
      CALL JLABEL (5, 19, 'wo=3   epsz=6 $')
      CALL JLABEL (55, 9, 'Defaults correspond $')
      CALL JLABEL (55, 8, 'to Reissner-Mindlin $')
      CALL JLABEL (55, 7, 'type beam theory    $')
      ENDIF

c     --- edit nuc1,nuc2 ---
      CALL JLABEL (0,17,'Number of displacement parameters retained, NUC
     .1 $')
      CALL JLABEL (0,16,'Number of displacement parameters neglected,NUC
     .2 $')
      ICOL = 50
  10  CALL PRFIELD (ICOL, 17, 2, NUC1, 'I2', IBGF, IFGF)
      CALL PRFIELD (ICOL, 16, 2, NUC2, 'I2', IBGF, IFGF)
      CALL CKFIELD (ICOL, 17, 2, NUC1, 'I2')
      CALL CKFIELD (ICOL, 16, 2, NUC2, 'I2')
      LINES(1) = NDDF
      LINES(2) = 0
      IF (NUC1.LT.1 .OR. NUC1.GT.NDDF) THEN
        CALL ERRORS ('Please redo: hit enter and input an integer betwe
     .en 1 and  $')
        CALL WAIT
        GOTO 10
      ENDIF
      IF (NUC2.LT.0 .OR. NUC2.GT.NDDF) THEN
        CALL ERRORS ('Please redo: hit enter and input an integer betwe
     .en 0 and  $')
        CALL WAIT
        GOTO 10
      ENDIF
      IF (NUC1+NUC2.NE.NDDF) THEN
         CALL ERRORS ('Please redo: NUC1+NUC2 should equal  $')
         CALL WAIT
         GOTO 10
      ENDIF

c     --- table of displ. parameters ---
      CALL JLABEL (0, 13, 'Choose which displacement parameters are in e
     .ach group $')
      CALL JLABEL (0, 12, 'No. parameter   retained   neglected $')
      DO 15 I = 1, NVAL
         VAL(I) = 0
  15  CONTINUE
      DO 20 I = 1, NUC1
         VAL(I) = UC(I)
  20  CONTINUE
      DO 30 I = 1, NUC2
         VAL(I+NDDF) = UC(I+NUC1)
  30  CONTINUE
      HOMO = .FALSE.
      DO 50 I = 1, NCOLS
         TYPEP(I) = 'I2'
         ICOLSP(I) = 20+(I-1)*14
         FLP(I)    = 2
   50 CONTINUE
      NROWS  = NDDF
      NLINES = 10
   55 CALL PRTABL (0,2,NLINES,NROWS,NCOLS,ICOLSP,FLP,VAL,TYPEP,HOMO)
      CALL CKTABL (0,2,NLINES,NROWS,NCOLS,ICOLSP,FLP,VAL,TYPEP,HOMO)
      DO 60 I = 1, NUC1
         UC(I) = VAL(I)
   60 CONTINUE
      DO 70 I = 1, NUC2
         UC(I+NUC1) = VAL(I+NDDF)
   70 CONTINUE
      ERR=.FALSE.
      DO 90 I = 1, NDDF
         IF (UC(I).LT.1 .OR. UC(I).GT.NDDF) ERR=.TRUE.
   90 CONTINUE
      IF (ERR) THEN
         LINES(1) = NDDF
         LINES(2) = 0
         CALL ERRORS ('Please redo : table numbers should be integers be
     .tween 1 and  $')
         CALL WAIT
         GOTO 55
      ENDIF
      RETURN
      END

$STORAGE:2
      SUBROUTINE RUN
C***********************************************************************
C     Runs data, does all computation
C***********************************************************************
$INCLUDE:'DBLE'
$INCLUDE:'PARAM.CMN'
$INCLUDE:'DATA.CMN'

C     . set up .
CCC   OPEN (7,FILE='CMDAY')
CCC   WRITE(7,*) 'SETUP'
      CALL SETUP

C     . loop over the elements .
      CALL ZEROV (KC, ZNSDF*ZNSDF)
      CALL ZEROV (PT, ZNSDF*4)
      CALL ZEROV (MC, ZNDDF*ZNDDF)
CCC   WRITE(7,*) 'LOOP OVER ELEMENTS'
      DO 10 K = 1, NE

C        . compute element of [R],[T],[M],[GAM1], and [GAM2] .
CCC      WRITE(7,*) 'RTMGAM, K = ', K
         CALL RTMGAM
```

```
C         . compute element of [KC],[MC], and [PT]; add in .
CCC       WRITE(7,*) 'KCMCPT, K = ', K
          CALL KCMCPT
   10  CONTINUE
CCC    CALL OUTMATX (KC, NSDF, NSDF, NSDF, 10, 7, 'KC :      ')
CCC    CALL OUTMATX (PT, NSDF, NSDF,NDIMT, 10, 7, 'PT :      ')
CCC    CALL OUTMATX (MC, NDDF, NDDF, NDDF, 10, 7, 'MC :      ')

C      . partition [KC] AND [MC] .
CCC    WRITE(7,*) 'PARTITION'
       CALL PARTIT

C      . compute [C] AND [CT] .
CCC    WRITE(7,*) '[C],[CT]'
       CALL CCT
       RETURN
       END


       SUBROUTINE SETUP
C***********************************************************************
C      do all setting up :
C
C      ... compute inverse permutations ECI and UCI for parameters
C      ... for each element, compute length
C      ... for each element, compute direction cosines X,Y,Z
C      ... for each element, compute center coordinates X,Y,Z
C***********************************************************************
$INCLUDE:'DBLE'
$INCLUDE:'GLOBCONS.CMN'
$INCLUDE:'CKCOLOR.CMN'
$INCLUDE:'PARAM.CMN'
$INCLUDE:'DATA.CMN'
       PARAMETER (NCOLS=7, NVAL=NCOLS*ZNE)
       INTEGER ICOLSP(NCOLS), FLP(NCOLS)
       REAL*8 VAL(NVAL)
       CHARACTER TYPEP(NCOLS)*2
       LOGICAL HOMO

       CALL JCLEAR
       CALL CLFIELD (0,24,80,IBGF,IFGF)
       CALL JLABEL (0,24,'Setting up ... $')
C
C      . COMPUTE THE INVERSE OF (EC) AND (UC) .
       DO 2 K = 1, NSDF
          N = EC(K)
          ECI(N) = K
    2  CONTINUE
       DO 4 K = 1, NDDF
          N = UC(K)
          UCI(N) = K
    4  CONTINUE
C
C      . LOOP OVER THE ELEMENTS .
       DO 10 K = 1, NE
```

```
C
C        . FIND WHICH NODES THIS ELEMENT CONNECTS .
         I = NOP(K,1)
         J = NOP(K,2)
CCC      WRITE(7,*) 'ELEMENT ',K,' CONNECTS NODES ',I,J
C
C        . COMPUTE LENGTH .
         XX = XCOORD(J) - XCOORD(I)
         YY = YCOORD(J) - YCOORD(I)
         ZZ = ZCOORD(J) - ZCOORD(I)
CCC      WRITE(7,*) '   XX,YY,ZZ = ',XX,YY,ZZ
         LEN(K) = SQRT (XX*XX + YY*YY + ZZ*ZZ)
CCC      WRITE(7,*) '   LEN      = ',LEN(K)
C
C        . COMPUTE X,Y,Z DIRECTION COSINES .
         DCOSX(K) = XX / LEN(K)
         DCOSY(K) = YY / LEN(K)
         DCOSZ(K) = ZZ / LEN(K)
CCC      WRITE(7,*) '   DCOSX,Y,Z= ', DCOSX(K),DCOSY(K),DCOSZ(K)
C
C        . COMPUTE X,Y,Z CENTER COORDINATE .
         EX(K) = .5 * (XCOORD(I) + XCOORD(J))
         EY(K) = .5 * (YCOORD(I) + YCOORD(J))
         EZ(K) = .5 * (ZCOORD(I) + ZCOORD(J))
  10  CONTINUE
C
C     . WRITE OUT ELEMENT DATA .
CCC   WRITE(7,*) ' '
CCC   WRITE(7,*) ' ELEMENT  LENGTH           DCOSX       DCOSY
CCC  .         ' DCOSZ            EX            EY          EZ'
CCC   WRITE(7,20) (K,LEN(K),DCOSX(K),DCOSY(K),DCOSZ(K),EX(K),EY(K),
CCC  .              EZ(K), K=1,NE)
  20  FORMAT (I5, E13.6, 5X, 3E13.6, 5X, 3E13.6)
C
C     . SET UP TABLE TO PRINT .
      DO 30 I = 1, NE
         VAL(I)      = LEN(I)
         VAL(I+1*NE) = DCOSX(I)
         VAL(I+2*NE) = DCOSY(I)
         VAL(I+3*NE) = DCOSZ(I)
         VAL(I+4*NE) = EX(I)
         VAL(I+5*NE) = EY(I)
         VAL(I+6*NE) = EZ(I)
   30 CONTINUE
      HOMO=.TRUE.
      TYPEP(1) = 'R8'
      ICOLSP(1)= 10
      FLP(1)   =  9
      NROWS = NE
      NLINES = MIN0(NE,20)
      IROW   = 21-NLINES
      CALL JLABEL (19,22,'direction cosines              coordinates of c
     .enter $')
      CALL JLABEL (0,21,'Elem.no. LENGTH    l         m         n
     .  Xc        Yc        Zc$')
```

```
      CALL PRTABL (0,IROW,NLINES,NROWS,NCOLS,ICOLSP,FLP,VAL,TYPEP,HOMO)
      CALL CKTABL (0,IROW,NLINES,NROWS,NCOLS,ICOLSP,FLP,VAL,TYPEP,HOMO)
      CALL CLFIELD (0,24,80,IBGF,IFGF)
      CALL JLABEL (0,24,'Hit enter to continue ... $')
      CALL WAIT
      CALL JCLEAR
C
C     ... EXIT
      RETURN
      END


      SUBROUTINE RTMGAM
C***********************************************************************
C     computes for an element the following matrices :
C
C     [R] = [L*L, M*M, N*N, L*M, L*N, M*N]
C                                                 [1 0 0]
C     [M] = 1/6 * RHO*AREA*LEN * [2*I3   I3]   I3=[0 1 0]
C                                [ I3  2*I3],     [0 0 1]
C
C     [T] = [1, YC, ZC, YC*ZC], BEAMLIKE
C     [T] = [1, ZC]          , PLATELIKE
C
C     [GAM1], DEPENDS ON BEAM-OR-PLATE, CHOICE OF
C     [GAM2], EC1,EC2,EC3,UC1,UC2
C***********************************************************************
$INCLUDE:'DBLE'
$INCLUDE:'GLOBCONS.CMN'
$INCLUDE:'CKCOLOR.CMN'
$INCLUDE:'PARAM.CMN'
$INCLUDE:'DATA.CMN'

      CALL CLFIELD (0,24,80,IBGF,IFGF)
      CALL JLABEL (0,24,'Computing elements number $')
      CALL PRFIELD (28, 24, 3, K, 'I2', IBGF, IFGF)
C
C     --- COMPUTE [R] ---
      R(1) = DCOSX(K)*DCOSX(K)
      R(2) = DCOSY(K)*DCOSY(K)
      R(3) = DCOSZ(K)*DCOSZ(K)
      R(4) = DCOSX(K)*DCOSY(K)
      R(5) = DCOSX(K)*DCOSZ(K)
      R(6) = DCOSY(K)*DCOSZ(K)
CCC   IF (K.EQ.1.OR.K.EQ.NE) WRITE(7,10) '0[R], ELEMENT=',K,(R(I),I=1,6)
  10  FORMAT (A,I5/ 10E13.6)
C
C     --- COMPUTE [T] ---
      IF (ITRUSS.EQ.1) THEN
         NDIMT=4
         T(1) = 1
         T(2) = EY(K)
         T(3) = EZ(K)
         T(4) = EY(K)*EZ(K)
      ELSE
```

```
         NDIMT=2
         T(1) = 1
         T(2) = EZ(K)
      ENDIF
CCC   IF (K.EQ.1.OR.K.EQ.NE) WRITE(7,10) '0[T], ELEMENT= ',K,(T(I),I=1,
CCC   .    NDIMT)
C
C     --- COMPUTE [M] ---
      CALL ZEROV (M, 6*6)
      IP = IPROP(K)
      VALUE = RHO(IP)*AREA(IP)*LEN(K) / 6.
      DO 20 I=1,6
         M(I,I) = VALUE*2.
  20  CONTINUE
      M(1,4) = VALUE
      M(2,5) = VALUE
      M(3,6) = VALUE
      M(4,1) = VALUE
      M(5,2) = VALUE
      M(6,3) = VALUE
CCC   IF (K.EQ.1.OR.K.EQ.NE) CALL OUTMATX (M,6,6,6,10,7,'[M] :      ')
C
C     --- COMPUTE [GAM1] ---
      CALL DOGAM1
CCC   IF (K.EQ.1.OR.K.EQ.NE)
CCC  .CALL OUTMATX(GAM1,6,6,NSDF,10,7,'[GAM1] :  ')
C
C     --- COMPUTE [GAM2] ---
      CALL DOGAM2
CCC   IF (K.EQ.1.OR.K.EQ.NE)
CCC  .CALL OUTMATX(GAM2,6,6,NDDF,10,7,'[GAM2] :  ')
C
C     ... EXIT
      RETURN
      END


      SUBROUTINE DOGAM1
C***********************************************************************
C     COMPUTES ELEMENTAL MATRIX [GAM1]
C
C     DEPENDS ON BEAMLIKE (ITRUSS=1) OR PLATELIKE (ITRUSS=2),
C     AND ON CHOICE OF EC1,EC2,EC3
C
C NOTE THAT EC(K) = N,  WHERE THE NTH PARAMETER IS IN THE KTH POSITION
C           ECI(N)= K,
C THEREFORE, WE NEED TO USE ECI HERE
C**********************************************************************
$INCLUDE:'DBLE'
$INCLUDE:'GLOBCONS.CMN'
$INCLUDE:'CKCOLOR.CMN'
$INCLUDE:'PARAM.CMN'
$INCLUDE:'DATA.CMN'
C
C     --- BEAMLIKE TRUSS ---
```

```
      CALL ZEROV (GAM1, 6*NSDF)
      IF (ITRUSS.EQ.1) THEN
         GAM1(1,ECI(1)) =  1
         GAM1(1,ECI(2)) = -EY(K)
         GAM1(1,ECI(3)) =  EZ(K)
         GAM1(1,ECI(9)) =  EY(K)*EZ(K)

         GAM1(2,ECI(10)) =  1
         GAM1(2,ECI(12)) =  EZ(K)

         GAM1(3,ECI(11)) =  1
         GAM1(3,ECI(13)) =  EY(K)

         GAM1(4,ECI(4))  =  1
         GAM1(4,ECI(6))  = -EZ(K)
         GAM1(4,ECI(7))  =  EZ(K)
         GAM1(4,ECI(14)) =  EY(K)
         GAM1(4,ECI(17)) =  EY(K)*EZ(K)

         GAM1(5,ECI(5))  =  1
         GAM1(5,ECI(6))  =  EY(K)
         GAM1(5,ECI(7))  =  EY(K)
         GAM1(5,ECI(15)) =  EZ(K)
         GAM1(5,ECI(18)) =  EY(K)*EZ(K)

         GAM1(6,ECI(8))  =  1
         GAM1(6,ECI(12)) =  EY(K)
         GAM1(6,ECI(13)) =  EZ(K)
C
C     --- PLATELIKE TRUSS ---
      ELSE
         GAM1(1,ECI(1)) =  1
         GAM1(1,ECI(4)) =  EZ(K)

         GAM1(2,ECI(2)) =  1
         GAM1(2,ECI(5)) =  EZ(K)

         GAM1(3,ECI(13)) =  1

         GAM1(4,ECI(3))  =  1
         GAM1(4,ECI(6))  =  EZ(K)

         GAM1(5,ECI(7))  =  1
         GAM1(5,ECI(9))  =  EX(K)
         GAM1(5,ECI(10)) =  EY(K)
         GAM1(5,ECI(14)) =  EZ(K)

         GAM1(6,ECI(8))  =  1
         GAM1(6,ECI(11)) =  EX(K)
         GAM1(6,ECI(12)) =  EY(K)
         GAM1(6,ECI(15)) =  EZ(K)
      ENDIF
C
C     ... EXIT
      RETURN
      END


      SUBROUTINE DOGAM2
C***********************************************************************
C     COMPUTES ELEMENTAL MATRIX [GAM2]
C
C     DEPENDS ON BEAMLIKE (ITRUSS=1) OR PLATELIKE (ITRUSS=2),
C     AND ON CHOICE OF UC1,UC2
C
C NOTE THAT UC(K) = N,  WHERE THE NTH PARAMETER IS IN THE KTH POSITION
C           UCI(N)= K,
C THEREFORE, WE NEED TO USE UCI HERE
C***********************************************************************
$INCLUDE:'DBLE'
$INCLUDE:'PARAM.CMN'
$INCLUDE:'DATA.CMN'
      REAL*8 X(2), Y(2), Z(2)
C
C     . FIND WHICH NODES THIS ELEMENT CONNECTS .
      I = NOP(K,1)
      J = NOP(K,2)
      X(1) = XCOORD(I)
      X(2) = XCOORD(J)
      Y(1) = YCOORD(I)
      Y(2) = YCOORD(J)
      Z(1) = ZCOORD(I)
      Z(2) = ZCOORD(J)
C
C     . DO FIRST  USING X,YCOORD OF ITH NODE .
C     . DO SECOND USING X,YCOORD OF JTH NODE .
      CALL ZEROV (GAM2, 6*NDDF)
      DO 20 N=1,2
C
C     --- BEAMLIKE TRUSS ---
      IF (ITRUSS.EQ.1) THEN
         GAM2(1+3*(N-1),UCI(1)) =  1
         GAM2(1+3*(N-1),UCI(5)) =  Z(N)
         GAM2(1+3*(N-1),UCI(6)) = -Y(N)
         GAM2(1+3*(N-1),UCI(7)) =  Y(N)*Z(N)

         GAM2(2+3*(N-1),UCI(2)) =  1
         GAM2(2+3*(N-1),UCI(4)) = -Z(N)
         GAM2(2+3*(N-1),UCI(8)) = .5*Z(N)
         GAM2(2+3*(N-1),UCI(9)) =  Y(N)
         GAM2(2+3*(N-1),UCI(11))=  Y(N)*Z(N)

         GAM2(3+3*(N-1),UCI(3)) =  1
         GAM2(3+3*(N-1),UCI(4)) =  Y(N)
         GAM2(3+3*(N-1),UCI(8)) = .5*Y(N)
         GAM2(3+3*(N-1),UCI(10))=  Z(N)
         GAM2(3+3*(N-1),UCI(12))=  Y(N)*Z(N)
C
C     --- PLATELIKE TRUSS ---
      ELSE
```

```
            GAM2(1+3*(N-1),UCI(1)) =  1
            GAM2(1+3*(N-1),UCI(4)) =  Z(N)

            GAM2(2+3*(N-1),UCI(2)) =  1
            GAM2(2+3*(N-1),UCI(5)) =  Z(N)

            GAM2(3+3*(N-1),UCI(3)) =  1
            GAM2(3+3*(N-1),UCI(6)) =  Z(N)
         ENDIF

   20 CONTINUE
C
C     ... EXIT
      RETURN
      END


      SUBROUTINE KCMCPT
C***********************************************************************
C     COMPUTES AN ELEMENT OF [KC], [MC], AND [PT]
C
C     ELEMENT [KC] =       E*AREA*LEN * ([R]*[GAM1])T * ([R]*[GAM1])
C     ELEMENT [PT] = ALPHA*E*AREA*LEN * ([R]*[GAM1])T * [T]
C     ELEMENT [MC] =          1/OMEGA * [GAM2]T*[M]*[GAM2]
C
C     SUM OF ELEMENTS IS DONE BY ADDING EACH ELEMENT IN TO MATRIX
C***********************************************************************
$INCLUDE:'DBLE'
$INCLUDE:'GLOBCONS.CMN'
$INCLUDE:'CKCOLOR.CMN'
$INCLUDE:'PARAM.CMN'
$INCLUDE:'DATA.CMN'
      REAL*8 RGAM1(ZNSDF), MGAM2(6,ZNDDF)
      REAL*8 KCTEMP(NSS), PTTEMP(NS4), MCTEMP(NDD)
C
C     --- ELEMENT [KC] ---
      CALL MATMUL (R, GAM1, RGAM1, 1, 6, NSDF)
      CALL MATMUT (RGAM1, RGAM1, KCTEMP, 1, NSDF, NSDF)
CCC   IF (K.EQ.1.OR.K.EQ.NE)
CCC  .CALL OUTMATX (KCTEMP, NSDF, NSDF, NSDF, 10, 7, 'ELEM.KC : ')
      IP = IPROP(K)
      VALUE = E(IP)*AREA(IP)*LEN(K)
CCC   WRITE(7,*) 'KC MULTIPLIER = ', VALUE
      DO 10 I = 1, NSDF*NSDF
         KC(I) = KC(I) + VALUE*KCTEMP(I)
   10 CONTINUE
C
C     --- ELEMENT [PT] ---
      CALL MATMUT (RGAM1, T, PTTEMP, 1, NSDF, NDIMT)
CCC   IF (K.EQ.1.OR.K.EQ.NE)
CCC  .CALL OUTMATX (PTTEMP, NSDF, NSDF, NDIMT, 10, 7, 'ELEM.PT : ')
      VALUE = ALPHA(IP)*VALUE
CCC   WRITE(7,*) 'PT MULTIPLIER = ', VALUE
      DO 20 I = 1, NSDF*NDIMT
         PT(I) = PT(I) + VALUE*PTTEMP(I)
   20 CONTINUE
C
C     --- ELEMENT [MC] ---
      CALL MATMUL (M, GAM2, MGAM2, 6, 6, NDDF)
      CALL MATMUT (GAM2, MGAM2, MCTEMP, 6, NDDF, NDDF)
CCC   IF (K.EQ.1.OR.K.EQ.NE)
CCC  .CALL OUTMATX (MCTEMP, NDDF, NDDF, NDDF, 10, 7, 'ELEM.MC : ')
      DO 30 I = 1, NDDF*NDDF
         MC(I) = MC(I) + MCTEMP(I)/OMEGA
   30 CONTINUE
C
C     ... EXIT
  100 FORMAT (A,I5/ (10E13.6))
      RETURN
      END


      SUBROUTINE PARTIT
C***********************************************************************
C     PARTITION [KC], [MC], AND [PT]
C
C     [KC] = [ KC11 KC12 KC13 ]   [MC] = [MC11 MC12]   [PT] = [PT1]
C            [      KC22 KC23 ]          [MC21 MC22]          [PT2]
C            [ SYM.      KC33 ]                               [PT3]
C
C     ONLY NEED KC11,KC12,KC21,KC22,MC11,PT1,PT2
C***********************************************************************
$INCLUDE:'DBLE'
$INCLUDE:'GLOBCONS.CMN'
$INCLUDE:'CKCOLOR.CMN'
$INCLUDE:'PARAM.CMN'
$INCLUDE:'DATA.CMN'
      IK(I,J) = (J-1)*NSDF + I
      IM(I,J) = (J-1)*NDDF + I
      IP(I,J) = (J-1)*NSDF + I
      IK1(I,J) = (J-1)*NEC1 + I
      IK2(I,J) = (J-1)*NEC2 + I
      IM1(I,J) = (J-1)*NUC1 + I

      CALL CLFIELD (0,24,80,IBGF,IFGF)
      CALL JLABEL (0,24,'Removing neglected parameters ... $')
C
C     --- PARTITION [KC] ---
      DO 10 J = 1, NEC1
      DO 10 I = 1, NEC1
         KC11(IK1(I,J)) = KC(IK(I,J))
   10 CONTINUE

      DO 20 J = 1, NEC2
      DO 20 I = 1, NEC1
         JJ = J+NEC1
         II = I
         KC12(IK1(I,J)) = KC(IK(II,JJ))
         KC21(IK2(J,I)) = KC(IK(JJ,II))
   20 CONTINUE
```

```
      DO 30 J = 1, NEC2
      DO 30 I = 1, NEC2
         JJ = J+NEC1
         II = I+NEC1
         KC22(IK2(I,J)) = KC(IK(II,JJ))
   30 CONTINUE
CCC   CALL OUTMATX (KC12, NEC1, NEC1, NEC2, 10, 7, 'KC12 :     ')
CCC   CALL OUTMATX (KC21, NEC2, NEC2, NEC1, 10, 7, 'KC21 :     ')
CCC   CALL OUTMATX (KC11, NEC1, NEC1, NEC1, 10, 7, 'KC11 :     ')
CCC   CALL OUTMATX (KC22, NEC2, NEC2, NEC2, 10, 7, 'KC22 :     ')
C
C     --- PARTITION [PT] ---
      DO 40 J = 1, NDIMT
      DO 40 I = 1, NEC1
         PT1(IK1(I,J)) = PT(IP(I,J))
   40 CONTINUE
      DO 50 J = 1, NDIMT
      DO 50 I = 1, NEC2
         II = I+NEC1
         PT2(IK2(I,J)) = PT(IP(II,J))
   50 CONTINUE
CCC   CALL OUTMATX (PT1, NEC1, NEC1,NDIMT, 10, 7, 'PT1 :      ')
CCC   CALL OUTMATX (PT2, NEC2, NEC2,NDIMT, 10, 7, 'PT2 :      ')
C
C     --- PARTITION [MC] ---
      DO 60 J = 1, NUC1
      DO 60 I = 1, NUC1
         MC11(IM1(I,J)) = MC(IM(I,J))
   60 CONTINUE
CCC   CALL OUTMATX (MC11, NUC1, NUC1, NUC1, 10, 7, 'MC11 :     ')
C
C     ... EXIT
      RETURN
      END


      SUBROUTINE CCT
C*********************************************************************
C     COMPUTES FINAL MATRICES
C STIFFNESS MATRIX [C]  = 1/OMEGA * [KC11 - KC12*KC22$^{-1}$*KC21]
C
C THERMAL COEFF.  [CT] = 1/OMEGA * [PT1  - KC12*KC22$^{-1}$*PT2]
C
C NOTES
C ... TEMP = [KC12*KC22INV]
C ... IF NEC2=0, SKIP COMPUTING TEMP*KC21 AND TEMP*PT2
C ... USES "MATINV" TO DO MATRIX INVERSION
C*********************************************************************
$INCLUDE:'DBLE'
$INCLUDE:'GLOBCONS.CMN'
$INCLUDE:'CKCOLOR.CMN'
$INCLUDE:'PARAM.CMN'
$INCLUDE:'DATA.CMN'
      COMMON /LINES/ LINES(10)
      PARAMETER (NS2=ZNSDF*2)
      PARAMETER (MNCOLS=ZNSDF, NVAL=ZNSDF*ZNSDF)
      PARAMETER (MN2COLS=ZNDDF, NVAL2=ZNDDF*ZNDDF)
      REAL*8 B(ZNSDF), IPIVOT(ZNSDF), IWK(NS2), TEMP(NSS)
      REAL*8 VAL(NVAL), VAL2(NVAL)
      INTEGER ICOLSP(MNCOLS), FLP(MNCOLS)
      INTEGER I2COLSP(MN2COLS), FLP2(MN2COLS)
      CHARACTER TYPEP(MNCOLS)*2, TYPEP2(MN2COLS)*2
      LOGICAL HOMO
      IC(I,J) = (J-1)*NEC1 + I
      IM(I,J) = (J-1)*NUC1 + I
      IK2(I,J)= (J-1)*NEC2 + I

      CALL CLFIELD (0,24,80,IBGF,IFGF)
      CALL JLABEL (0,24,'Computing [C] and [CT] ... $')
C
C     . ZERO OUT MATRICES .
      CALL ZEROV (C, ZNSDF*ZNSDF)
      CALL ZEROV (CT, ZNSDF*4)
C
C     *** IF NEC2=0, SKIP COMPUTE KC22 INVERSE AND TEMP ***
      IF (NEC2.EQ.0) GOTO 5
C
C     --- CHECK IF ROW (& COLUMN) ARE ALL ZEROS, IF SO ---
C     --- REMOVE SINGULARITY WITH A ONE ON DIAGONAL    ---
      DO 4 J = 1, NEC2
         DO 3 I = 1, NEC2
            IF (KC22(IK2(I,J)).NE.0.) GOTO 4
   3     CONTINUE
C        . SINGULAR AT KC22(J,J) .
         KC22(IK2(J,J)) = 1.
         CALL CLFIELD (0,24,80,IBGF,IFGF)
         LINES(1) = NEC1 + J
         LINES(2) = 0
         CALL ERRORS ('WARNING ! These strain parameters have no contrib
  .ution : $')
         CALL WAIT
         CALL CLFIELD (0,24,80,IBGF,IFGF)
   4  CONTINUE
C
C     --- INVERT [KC22] ---
      MROWS = 0
      IOP=0
      CALL MATINV (NEC2,NEC2,KC22,MROWS,B,IOP,DETERM,ISCALE,IPIVOT,IWK)
CCC   WRITE(7,*) ' INVERTING KC22, DETERM = ', DETERM
      IF (DETERM.EQ.0) STOP 'KC22 SINGULAR !'
CCC   CALL OUTMATX (KC22, NEC2, NEC2, NEC2, 10, 7, 'KCINV :    ')
C
C     --- COMPUTE [C] ---
      CALL MATMUL (KC12, KC22, TEMP, NEC1, NEC2, NEC2)
CCC   CALL OUTMATX (TEMP, NEC1, NEC1, NEC2, 10, 7, 'TEMP :     ')
      CALL MATMUL (TEMP, KC21,    C, NEC1, NEC2, NEC1)
CCC   CALL OUTMATX (C,    NEC1, NEC1, NEC1, 10, 7, '12*22*21 :')
   5  DO 10 I = 1, NEC1
      DO 10 J = 1, NEC1
```

```
        C(IC(I,J)) = (KC11(IC(I,J)) - C(IC(I,J))) / OMEGA
  10  CONTINUE
CCC   CALL OUTMATX (C, NEC1, NEC1, NEC1, 10, 7, '[C] :     ')
C
C     --- COMPUTE [CT] ---
      IF (NEC2.NE.0) CALL MATMUL (TEMP, PT2, CT, NEC1, NEC2, NDIMT)
      DO 20 I = 1, NEC1
      DO 20 J = 1, NDIMT
        CT(IC(I,J)) = (PT1(IC(I,J)) -CT(IC(I,J))) / OMEGA
  20  CONTINUE
CCC   CALL OUTMATX (CT, NEC1, NEC1, NDIMT, 10, 7, '[CT] :    ')
C
C     --- PRINT OUT TABLE OF [C] ---
      IF (ITRUSS.EQ.1) CALL OUTBEAM
      IF (ITRUSS.EQ.2) CALL OUTPLAT
      DO 30 I = 1, NEC1
      DO 30 J = 1, NEC1
        VAL(IC(I,J)) = C(IC(I,J))
  30  CONTINUE
      HOMO      = .TRUE.
      TYPEP(1) = 'R8'
      ICOLSP(1)= 4
      FLP(1)    = (77-(NEC1-1))/NEC1
      FLP(1)    = MINO(FLP(1),15)
      NCOLS     = NEC1
      NROWS     = NEC1
      NLINES    = NEC1
      IROW      = 22-NLINES
      CALL JLABEL (0,22,'[C] : $')
      CALL PRTABL (0,IROW,NLINES,NROWS,NCOLS,ICOLSP,FLP,VAL,TYPEP,HOMO)
      CALL CLFIELD (0,24,80,IBGF,IFGF)
      CALL JLABEL (0,24,'Hit enter to continue ... $')
      CALL WAIT
      CALL JCLEAR
C
C     --- PRINT OUT TABLE OF [CT] ---
      DO 40 I = 1, NEC1
      DO 40 J = 1, NDIMT
        VAL(IC(I,J)) = CT(IC(I,J))
  40  CONTINUE
      FLP(1) = 12
      NCOLS  = NDIMT
      CALL JLABEL (0,22,'[CT] : $')
      CALL PRTABL (0,IROW,NLINES,NROWS,NCOLS,ICOLSP,FLP,VAL,TYPEP,HOMO)
      CALL CLFIELD (0,24,80,IBGF,IFGF)
      CALL JLABEL (0,24,'Hit enter to continue ... $')
      CALL WAIT
      CALL JCLEAR
C
C     --- PRINT OUT TABLE OF [MC11] ---
      DO 50 I = 1, NUC1
      DO 50 J = 1, NUC1
       VAL2(IM(I,J)) = MC11(IM(I,J))
  50  CONTINUE
      HOMO = .TRUE.
      TYPEP2(1) = 'R8'
      FLP2(1)   = 12
      I2COLSP(1)= 4
      FLP(1)    = (77-(NUC1-1))/NUC1
      FLP(1)    = MINO(FLP(1),15)
      NCOLS     = NUC1
      NROWS     = NUC1
      NLINES    = NUC1
      IROW      = 22-NLINES
      CALL JLABEL (0,22,'[m] : $')
      CALL PRTABL (0,IROW,NLINES,NROWS,NCOLS,I2COLSP,FLP2,VAL2,TYPEP2,
     .             HOMO)
      CALL CLFIELD (0,24,80,IBGF,IFGF)
      CALL JLABEL (0,24,'Hit enter to continue ... $')
      CALL WAIT
      CALL JCLEAR
C
C     ... EXIT
      RETURN
      END


      SUBROUTINE OUTBEAM
C***********************************************************************
C     displays output informations for beamlike lattice truss
C***********************************************************************
$INCLUDE:'DBLE'
$INCLUDE:'GLOBCONS.CMN'
$INCLUDE:'CKCOLOR.CMN'

      CALL JLABEL (0, 22, '[C] for default set of strain and displ. set
     .of parameters $')
      CALL JLABEL (0, 21, '------------------------------------------------
     .-------------- $')
      CALL JLABEL (0, 20, 'C11     = extensional stiffness           $')
      CALL JLABEL (0, 19, 'C22,C33 = bending stiffnesses             $')
      CALL JLABEL (0, 18, 'C44,C55 = transverse shear stiffnesses   $')
      CALL JLABEL (0, 17, 'C66     = torsional stiffness             $')
      CALL JLABEL (0, 16, 'C14,C15 = extensional-shear stiffnesses  $')
      CALL JLABEL (0, 15, 'C23     = coupling bending stiffness      $')
      CALL JLABEL (0, 14, 'C26,C36 = bending-torsional stiffnesses  $')
      CALL JLABEL (0, 12, '[CT] for default set of strain and displ. set
     . of parameters $')
      CALL JLABEL (0, 11, '------------------------------------------------
     .--------------- $')
      CALL JLABEL (0, 10, 'This matrix is multiplied by the vector $')
      CALL JLABEL (4,  9, '(    To     ) $')
      CALL JLABEL (4,  8, '( d(To)/dy  ) $')
      CALL JLABEL (4,  7, '( d(To)/dz  ) $')
      CALL JLABEL (4,  6, '(d2(To)/dydz) $')

      CALL CLFIELD (0,24,80,IBGF,IFGF)
      CALL JLABEL (0,24,'Hit enter to continue ... $')
      CALL WAIT
      CALL JCLEAR
      RETURN
      END


      SUBROUTINE OUTPLAT
C************************************************************************
C     displays output information for platelike lattice truss
C************************************************************************
$INCLUDE:'DBLE'
$INCLUDE:'GLOBCONS.CMN'
$INCLUDE:'CKCOLOR.CMN'

      CALL JLABEL (0, 22, 'Cij for default set of strain and displ. set
     .of parameters $')
      CALL JLABEL (0, 21, '--------------------------------------------
     .------------- $')
      CALL JLABEL (0, 20, 'i=1,3 j=1,3  extensional stiffnesses      $')
      CALL JLABEL (0, 19, 'i=4,6 j=4,6  bending stiffnesses          $')
      CALL JLABEL (0, 18, 'i=7,8 j=7,8  transverse shear stiffnesses $')
      CALL JLABEL (0, 17, 'i=1,3 j=4,6  bending-extensional coupling sti
     .ffnesses $')
      CALL JLABEL (0, 16, 'i=4,6 j=1,3        "        "       "     $')
      CALL JLABEL (0, 12, '[CT] for default set of strain and displ. set
     . of parameters $')
      CALL JLABEL (0, 11, '--------------------------------------------
     .-------------- $')
      CALL JLABEL (0, 10, 'This matrix is multiplied by the vector $')
      CALL JLABEL (4,  9, '(   To    ) $')
      CALL JLABEL (4,  8, '(d(To)/dz) $')

      CALL CLFIELD (0,24,80,IBGF,IFGF)
      CALL JLABEL (0,24,'Hit enter to continue ... $')
      CALL WAIT
      CALL JCLEAR
      RETURN
      END

$STORAGE:2
C--------------------------------C
C GENERAL MATRIX SUBROUTINES :   C
C (DOUBLE PRECISION)              C
C                                C
C     MATMUL    ZEROV             C
C     MATMUT    MATINV            C
C     OUTMATX                     C
C------------------------------ C


      SUBROUTINE MATMUL (A,B,C,N,M,K)
C************************************************************************
C
C     MATRIX MULTIPLY: C[N,K] = A[N,M] * B[M,K]
C
C************************************************************************
$INCLUDE:'DBLE'
```

```
      DIMENSION A(N,M), B(M,K), C(N,K)
      DO 10 I = 1, N
      DO 10 J = 1, K
         C(I,J) = 0.
         DO 10 L = 1, M
         C(I,J) = C(I,J) + A(I,L)*B(L,J)
  10  CONTINUE
      RETURN
      END


      SUBROUTINE MATMUT (A,B,C,M,L,N)
C***********************************************************************
C
C     MATRIX TRANSPOSE MULTIPLY: C[L,N] = A[M,L]**T * B[M,N]
C
C***********************************************************************
$INCLUDE:'DBLE'
      DIMENSION A(M,L), B(M,N), C(L,N)
      DO 30 K = 1, N
      DO 20 I = 1, L
         SUM = 0.
         DO 10 J = 1, M
            SUM = SUM + A(J,I)*B(J,K)
  10     CONTINUE
         C(I,K) = SUM
  20  CONTINUE
  30  CONTINUE
      RETURN
      END


      SUBROUTINE OUTMATX (K, MNROWS, NROWS, NCOLS, NFMT, IOUTF, LABEL)
C***********************************************************************
C     OUTPUTS MATRIX [K] ON TAPE'IOUTF' WITH 'NFMT' NUMBERS PER LINE
C     LABEL MUST BE 10 CHARACTERS
C***********************************************************************
$INCLUDE:'DBLE'
      REAL*8 K(MNROWS,NCOLS)
      CHARACTER*10 LABEL

      WRITE(IOUTF,10) LABEL, NROWS, NCOLS
  10  FORMAT (1X,A10/ '  NROWS =',I5,' NCOLS=',I5)
      IEND = 0
      NNN = NCOLS/NFMT
      IF (MOD(NCOLS,NFMT).NE.0) NNN = NNN+1
      DO 40 II = 1, NNN
         IST = IEND+1
         IEND = IST+NFMT-1
         IEND = MIN0(IEND,NCOLS)
         WRITE(IOUTF,50) (I, I=IST,IEND)
         DO 30 IR = 1, NROWS
            WRITE(IOUTF,45) IR, (K(IR,J), J=IST,IEND)
  30     CONTINUE
  40  CONTINUE
```

```
   45 FORMAT (I6, 10E13.6)
   50 FORMAT (1X,5X,10I12)
      RETURN
      END


      SUBROUTINE ZEROV (V, N)
C*****************************************************************
C     ZERO OUT VECTOR (V) OF LENGTH N
C*****************************************************************
$INCLUDE:'DBLE'
      REAL*8 V(N)
      DO 10 I = 1, N
         V(I) = 0.
   10 CONTINUE
      RETURN
      END


      SUBROUTINE MATINV (MAX,N,A,M,B,IOP,DETERM,ISCALE,IPIVOT,IWK)
C*********************************************************************
C     NOTE : DOUBLE PRECISION, SINGLE COMMENTED OUT WITH C*
C
C     PURPOSE - MATINV INVERTS A REAL SQUARE MATRIX A.
C               IN ADDITION THE ROUTINE SOLVES THE MATRIX
C               EQUATION AX=B,WHERE B IS A MATRIX OF CONSTANT
C               VECTORS. THERE IS ALSO AN OPTION TO HAVE THE
C               DETERMINANT EVALUATED.
C
C     USE     - CALL MATINV(MAX,N,A,M,B,IOP,DETERM,ISCALE,IPIVOT,IWK)
C
C                 MAX - THE MAXIMUM ORDER OF A AS STATED IN THE
C                       DIMENSION STATEMENT OF THE CALLING PROGRAM.
C
C                 N   - THE ORDER OF A, 1.LE.N.LE.MAX.
C
C                 A   - A TWO-DIMENSIONAL ARRAY OF THE COEFFICIENTS.
C                       ON RETURN TO THE CALLING PROGRAM, A INVERSE
C                       IS STORED IN A.
C                       A MUST BE DIMENSIONED IN THE CALLING PROGRAM
C                       WITH FIRST DIMENSION MAX AND SECOND DIMENSION
C                       AT LEAST N.
C
C                 M   - THE NUMBER OF COLUMN VECTORS IN B.
C                       M=0 SIGNALS THAT THE SUBROUTINE IS
C                       USED SOLELY FOR INVERSION,HOWEVER,
C                       IN THE CALL STATEMENT AN ENTRY CORRE-
C                       SPONDING TO B MUST BE PRESENT.
C
C                 B   - A TWO-DIMENSIONAL ARRAY OF THE CONSTANT
C                       VECTOR B. ON RETURN TO CALLING PROGRAM,
C                       X IS STORED IN B. B SHOULD HAVE ITS FIRST
C                       DIMENSION MAX AND ITS SECOND AT LEAST M.
C
C                 IOP - COMPUTE DETERMINANT OPTION.
```

```
C                        IOP=0 COMPUTES THE MATRIX INVERSE AND
C                              DETERMINANT.
C                        IOP=1 COMPUTES THE MATRIX INVERSE ONLY.
C
C               DETERM- FOR IOP=0-IN CONJUNCTION WITH ISCALE
C                       REPRESENTS THE VALUE OF THE DETERMINANT
C                       OF A, DET(A),AS FOLLOWS.
C                       DET(A)=(DETERM)(10**100(ISCALE))
C                       THE COMPUTATION DET(A) SHOULD NOT BE
C                       ATTEMPTED IN THE USER PROGRAM SINCE IF
C                       THE ORDER OF A IS LARGER AND/OR THE
C                       MAGNITUDE OF ITS ELEMENTS ARE LARGE(SMALL),
C                       THE DET(A) CALCULATION MAY CAUSE OVERFLOW
C                       (UNDERFLOW). DETERM SET TO ZERO FOR
C                       SINGULAR MATRIX CONDITION, FOR EITHER
C                       IOP=1,OR 0. SHOULD BE CHECKED BY PROGRAMER
C                       ON RETURN TO MAIN PROGRAM.
C
C             ISCALE  - A SCALE FACTOR COMPUTED BY THE
C                       SUBROUTINE TO AVOID OVERFLOW OR
C                       UNDERFLOW IN THE COMPUTATION OF
C                       THE QUANTITY,DETERM.
C
C             IPIVOT  - A ONE DIMENSIONAL INTEGER ARRAY
C                       USED BY THE SUBPROGRAM TO STORE
C                       PIVOTAL INFORMATION. IT SHOULD BE
C                       DIMENSIONED AT LEAST N. IN GENERAL
C                       THE USER DOES NOT NEED TO MAKE USE
C                       OF THIS ARRAY.
C
C                IWK  - A TWO-DIMENSIONAL INTEGER ARRAY OF
C                       TEMPORARY STORAGE USED BY THE ROUTINE.
C                       IWK SHOULD HAVE ITS FIRST DIMENSION
C                       MAX, AND ITS SECOND 2.
C
C     REQUIRED ROUTINES-
C
C     REFERENCE       -FOX,L, AN INTRODUCTION TO NUMERICAL
C                              LINEAR ALGEBRA
C
C     STORAGE         - 542 OCTAL LOCATIONS
C
C     LANGUAGE        -FORTRAN
C
C     FORTRAN
C     LIBRARY FUNCTIONS -ABS
C
C     LATEST REVISION - JULY 1973 -CMPB
C
C     MATRIX INVERSION WITH ACCOMPANYING SOLUTION OF LINEAR EQUATIONS
C****************************************************************************
$INCLUDE:'DBLE'

      DIMENSION IPIVOT(N), A(MAX,N), B(MAX,N), IWK(MAX,2)
      EQUIVALENCE (IROW,JROW), (ICOLUM,JCOLUM), (AMAX, T, SWAP)
```

```
C
C     INITIALIZATION
C
      ISCALE=0
      R1 = 1.D100
      R1 = 1.E30
      R2=1.0/R1
      DETERM=1.0
      DO 20 J=1,N
   20 IPIVOT(J)=0
      DO 550 I=1,N
C
C     SEARCH FOR PIVOT ELEMENT
C
      AMAX=0.0
      DO 105 J=1,N
      IF (IPIVOT(J)-1) 60, 105, 60
   60 DO 100 K=1,N
      IF (IPIVOT(K)-1) 80, 100, 740
   80 IF (DABS(AMAX)-DABS(A(J,K)))85,100,100
C* 80 IF ( ABS(AMAX)- ABS(A(J,K)))85,100,100
   85 IROW=J
      ICOLUM=K
      AMAX=A(J,K)
  100 CONTINUE
  105 CONTINUE
      IF (AMAX) 110,106,110
  106 DETERM=0.0
      ISCALE=0
      GO TO 740
  110 IPIVOT(ICOLUM)=IPIVOT(ICOLUM)+1
C
C     INTERCHANGE ROWS TO PUT PIVOT ELEMENT ON DIAGONAL
C
      IF (IROW-ICOLUM) 140, 260, 140
  140 DETERM=-DETERM
      DO 200 L=1,N
      SWAP=A(IROW,L)
      A(IROW,L)=A(ICOLUM,L)
  200 A(ICOLUM,L)=SWAP
      IF(M) 260, 260, 210
  210 DO 250 L=1, M
      SWAP=B(IROW,L)
      B(IROW,L)=B(ICOLUM,L)
  250 B(ICOLUM,L)=SWAP
  260 IWK(I,1)=IROW
      IWK(I,2)=ICOLUM
      PIVOT=A(ICOLUM,ICOLUM)
      IF(IOP.EQ.1) GO TO 321
      IF(PIVOT)1000,106,1000
C
C     SCALE THE DETERMINANT
C
 1000 PIVOTI=PIVOT
      IF(DABS(DETERM)-R1)1030,1010,1010
```

```
C*    IF( ABS(DETERM)-R1)1030,1010,1010
 1010 DETERM=DETERM/R1
      ISCALE=ISCALE+1
      IF(DABS(DETERM)-R1)1060,1020,1020
C*    IF( ABS(DETERM)-R1)1060,1020,1020
 1020 DETERM=DETERM/R1
      ISCALE=ISCALE+1
      GO TO 1060
 1030 IF(DABS(DETERM)-R2)1040,1040,1060
C*1030 IF(ABS(DETERM)-R2)1040,1040,1060
 1040 DETERM=DETERM*R1
      ISCALE=ISCALE-1
      IF(DABS(DETERM)-R2)1050,1050,1060
C*    IF( ABS(DETERM)-R2)1050,1050,1060
 1050 DETERM=DETERM*R1
      ISCALE=ISCALE-1
 1060 IF(DABS(PIVOTI)-R1)1090,1070,1070
C*1060 IF(ABS(PIVOTI)-R1)1090,1070,1070
 1070 PIVOTI=PIVOTI/R1
      ISCALE=ISCALE+1
      IF(DABS(PIVOTI)-R1)320,1080,1080
C*    IF( ABS(PIVOTI)-R1)320,1080,1080
 1080 PIVOTI=PIVOTI/R1
      ISCALE=ISCALE+1
      GO TO 320
 1090 IF(DABS(PIVOTI)-R2)2000,2000,320
C*1090 IF(ABS(PIVOTI)-R2)2000,2000,320
 2000 PIVOTI=PIVOTI*R1
      ISCALE=ISCALE-1
      IF(DABS(PIVOTI)-R2)2010,2010,320
C*    IF( ABS(PIVOTI)-R2)2010,2010,320
 2010 PIVOTI=PIVOTI*R1
      ISCALE=ISCALE-1
  320 DETERM=DETERM*PIVOTI
C
C     DIVIDE PIVOT ROW BY PIVOT ELEMENT
C
  321 IF(PIVOT)330,106,330
  330 A(ICOLUM,ICOLUM)=1.0
      DO 350 L=1,N
  350 A(ICOLUM,L)=A(ICOLUM,L)/PIVOT
      IF(M) 380, 380, 360
  360 DO 370 L=1,M
  370 B(ICOLUM,L)=B(ICOLUM,L)/PIVOT
C
C     REDUCE NON-PIVOT ROWS
C
  380 DO 550 L1=1,N
      IF(L1-ICOLUM) 400, 550, 400
  400 T=A(L1,ICOLUM)
      A(L1,ICOLUM)=0.0
      DO 450 L=1,N
  450 A(L1,L)=A(L1,L)-A(ICOLUM,L)*T
      IF(M) 550, 550, 460
  460 DO 500 L=1,M
  500 B(L1,L)=B(L1,L)-B(ICOLUM,L)*T
  550 CONTINUE
C
C     INTERCHANGE COLUMNS
C
      DO 710 I=1,N
      L=N+1-I
      IF (IWK(L,1)-IWK(L,2))630,710,630
  630 JROW=IWK(L,1)
      JCOLUM=IWK(L,2)
      DO 705 K=1,N
      SWAP=A(K,JROW)
      A(K,JROW)=A(K,JCOLUM)
      A(K,JCOLUM)=SWAP

  705 CONTINUE
  710 CONTINUE
  740 RETURN
      END
```

References

1. Bibliography on Latticed Structures. *J. Struct. Div., ASCE,* vol. 98, no. ST7, July 1972, pp. 1545–1566.
2. Latticed Structures: State-of-the-Art Report. *J. Struct. Div., ASCE,* vol. 102, no. ST11, Nov. 1976, pp. 2197–2230.
3. Noor, Ahmed K.: On Making Large Nonlinear Problems Small. *Comput. Methods Appl. Mech. & Eng.*, vol. 34, nos. 1–3, Sept. 1982, pp. 955–985.
4. Noor, Ahmed K.; and Peters, Jeanne M.: Recent Advances in Reduction Methods for Instability Analysis of Structures. *Comput. & Struct.*, vol. 16, no. 1–4, 1983, pp. 67–80.
5. Dean, Donald L.; and Avent, R. Richard: State of the Art of Discrete Field Analysis of Space Structures. *2nd International Conference on Space Structures,* Dep. of Civil Engineering, Univ. of Surrey (Guildford, England), 1975, pp. 7–16.
6. Dean, Donald L.: *Discrete Field Analysis of Structural Systems.* Springer-Verlag, c.1976.
7. Renton, J. D.: The Related Behaviour of Plane Grids, Space Grids and Plates. *Space Structures,* R. M. Davies, ed., Blackwell Scientific Publ. (Oxford and Edinburgh, England), c.1967, pp. 19–32.
8. Renton, John D.: The Beam-Like Behavior of Space Trusses. *AIAA J.*, vol. 22, no. 2, Feb. 1984, pp. 273–280.
9. Wah, Thein; and Calcote, Lee R.: *Structural Analysis by Finite Difference.* Van Nostrand Reinhold, 1970.
10. McDaniel, T. J.; and Chang, K. J.: Dynamics of Rotationally Periodic Large Space Structures. *J. Sound & Vib.*, vol. 68, no. 3, Feb. 8, 1980, pp. 351–368.
11. Williams, F. W.: An Algorithm for Exact Eigenvalue Calculations for Rotationally Periodic Structures. *Int. J. Numer. Methods Eng.*, vol. 23, no. 4, Apr. 1986, pp. 609–622.
12. Williams, F. W.: Exact Eigenvalue Calculations for Structures With Rotationally Periodic Substructures. *Int. J. Numer. Methods Eng.*, vol. 23, no. 4, Apr. 1986, pp. 695–706.
13. Anderson, Melvin S.: Buckling of Periodic Lattice Structures. *AIAA J.*, vol. 19, no. 6, June 1981, pp. 782–788.
14. Anderson, Melvin S.: Vibration of Prestressed Periodic Lattice Structures. *AIAA J.*, vol. 20, no. 4, Apr. 1982, pp. 551–555.
15. Anderson, M. S.; and Williams, F. W.: Natural Vibration and Buckling of General Periodic Lattice Structures. *AIAA J.*, vol. 24, no. 1, Jan. 1986, pp. 163–169.
16. Abrate, S.: Continuum Modeling of Latticed Structures. *Shock & Vib. Dig.*, vol. 17, no. 1, Jan. 1985, pp. 15–21.
17. Dow, John O.; Su, Z. W.; Feng, C. C.; and Bodley, Carl: Equivalent Continuum Representation of Structures Composed of Repeated Elements. *AIAA J.*, vol. 23, no. 10, Oct. 1985, pp. 1564–1569.
18. Dow, John O.; and Huyer, Stephen A.: An Equivalent Continuum Analysis Procedure for Space Station Lattice Structures. *AIAA/ASME/ASCE/AHS 28th Structures, Structural Dynamics and Materials Conference, Part 1—A Collection of Technical Papers,* pp. 110–122. (Available as AIAA-87-0724.)
19. Flower, William R.; and Schmidt, Lewis C.: Analysis of Space Truss as Equivalent Plate. *J. Struct. Div., ASCE,* vol. 97, no. ST12, Dec. 1971, pp. 2777–2789.
20. Heki, Koichiro: On the Effective Rigidities of Lattice Plates. *Recent Researches of Structural Mechanics,* H. Tanaka and S. Kawamata, eds., Uno Shoten (Tokyo, Japan), 1968, pp. 31–46.
21. Kleiber, M.: The Approximate Methods in the Theory of Elastic Lattice-Type Shells. *Archit. Mech.*, vol. 25, no. 2, 1973, pp. 195–211.
22. Kollár, L.; and Hegedűs, I.: *Analysis and Design of Space Frames by the Continuum Method.* Elsevier Science Publ., 1985.
23. Nayfeh, Adnan H.; and Hefzy, Mohamed S.: Continuum Modeling of Three-Dimensional Truss-Like Space Structures. *AIAA J.*, vol. 16, no. 8, Aug. 1978, pp. 779–787.
24. Nayfeh, Adnan H.; and Hefzy, Mohamed S.: Continuum Modeling of the Mechanical and Thermal Behavior of Discrete Large Structures. *AIAA J.*, vol. 19, no. 6, June 1981, pp. 766–773.
25. Noor, Ahmed K.; Anderson, Melvin S.; and Greene, William H.: Continuum Models for Beam- and Platelike Lattice Structures. *AIAA J.*, vol. 16, no. 12, Dec. 1978, pp. 1219–1228.
26. Noor, Ahmed K.; and Weisstein, Larry S.: Stability of Beamlike Lattice Trusses. *Comput. Methods Appl. Mech. & Eng.*, vol. 25, no. 2, Feb. 1981, pp. 179–193.
27. Noor, Ahmed K.: Assessment of Current State of the Art in Modeling Techniques and Analysis Methods for Large Space Structures. *Modelling, Analysis, and Optimization Issues for Large Space Structures,* Larry D. Pinson, Anthony K. Amos, and V. B. Venkayya, compilers, NASA CP-2258, 1983, pp. 5–32.
28. Soare, Mircea V.: Application of the Equivalent Continuum Method to the Analysis of Double-Layer Parallel Square Mesh Grids. *Buletinul Stiintific al Institutului de Constructii Bucuresti*, Anul XIV, nr. 1–2, 1971, pp. 251–269.
29. Soare, Mircea V.: Statica şi Dinamica Reţelelor Spaţiale Planar Pătrate Simple Prin Aplicarea Metodei Mediului Continuu Echivalent. *Studii si Cercetari de Mechanica Aplicata*, Tomul 31, nr. 3, 1972, pp. 673–702.
30. Sun, C. T.; and Yang, T. Y.: A Continuum Approach Toward Dynamics of Gridworks. *J. Appl. Mech.*, vol. 40, ser. E, no. 1, Mar. 1973, pp. 186–192.
31. Sun, C. T.; Kim, B. J.; and Bogdanoff, J. L.: On the Derivation of Equivalent Simple Models for Beam- and Plate-Like Structures in Dynamic Analysis. *A Collection of Technical Papers, Part 2—AIAA/ASME/ASCE/AHS 22nd Structures, Structural Dynamics & Materials Conference and AIAA Dynamics Specialists Conference*, 1981, pp. 523–532. (Available as AIAA-81-0624.)
32. Weisstein, Larry S.: Introduction and Survey on Continuum Models for Repetitive Lattice Structures. *Proceedings of the Workshop on Applications of Distributed System Theory to the Control of Large Space Structures*, G. Rodriguez, ed., JPL Publ. 83-46, California Inst. of Technology, July 1, 1983, pp. 63–70.

33. Wright, D. T.: A Continuum Analysis for Double Layer Space Frame Shells. *Publ. Int. Assoc. Bridge & Struct. Eng.* (Zurich), vol. 26, 1966, pp. 593–610.

34. Timoshenko, Stephen P.; and Gere, James M.: *Theory of Elastic Stability, Second ed.* McGraw-Hill Book Co., 1961.

35. Noor, Ahmed K.; and Andersen, C. M.: Analysis of Beam-Like Lattice Trusses. *Comput. Methods Appl. Mech. & Eng.*, vol. 20, no. 1, Oct. 1979, pp. 53–70.

36. Heki, K.; and Fujitani, Y.: The Stress Analysis of Grids Under the Action of Bending and Shear. *Space Structures*, R. M. Davies, ed., Blackwell Scientific Publ. (Oxford and Edinburgh, England), c.1967, pp. 33–43.

37. Heki, Koichiro: The Effect of Shear Deformation on Double Layer Lattice Plates and Shells. *2nd International Conference on Space Structures*, Dep. of Civil Engineering, Univ. of Surrey (Guildford, England), 1975, pp. 189–198.

38. Kollár, L.; and Hegedus, I.: Solution of Double-Layer Space Trusses of General Triangular Grid by the Equivalent Continuum Method. *Acta Technica Scientiarum Hungaricae*, Tomus 74 (3–4), 1973, pp. 363–381.

39. Kollár, L.: Continuum Method of Analysis for Double-Layer Space Trusses with Upper and Lower Chord Planes of Different Rigidities. *Acta Technica Academiae Scientiarum Hungaricae*, Tomus 76 (1–2), 1974, pp. 53–63.

40. Kollár, L.: Analysis of Double-Layer Space Trusses With Diagonally Square Mesh by the Continuum Method. *Acta Technica Academiae Scientiarum Hungaricae*, Tomus 76 (3–4), 1974, pp. 273–292.

41. Kollar, L.: Continuum Method of Analysis for Double-Layer Space Trusses of Hexagonal Over Triangular Mesh. *Acta Technica*, vol. 86, no. 1–2, 1978, pp. 55–77.

42. Saka, Toshitsugu; and Heki, Koichiro: On the Effective Rigidities of Bar-Like Plane Trusses. *Mem. Fac. Eng., Osaka City Univ. (Osaka, Japan)*, vol. 22, Dec. 1981, pp. 167-173.

43. Artola, Michel; and Duvaut, Georges: Élasticité—Homogénéisation d'une Plaque Renforcée. *Comptes Rendus des Séances de L'Académie des Sciences*, Tome 284, ser. A, no. 12, Mar. 21, 1977, pp. 707–710.

44. Babuška, Ivo: Homogenization and Its Application. Mathematical and Computational Problems. *Numerical Solution of Partial Differential Equations—III, SYNSPADE 1975*, Bert Hubbard, ed., Academic Press, Inc., 1976, pp. 89–116.

45. Bensoussan, Alain; Lions, Jacques Louis; and Papanicolaou, George: *Asymptotic Analysis for Periodic Structures.* North-Holland Publ. Co., 1978.

46. Lutoborski, Adam: Homogenization of Linear Elastic Shells. *J. Elast.*, vol. 15, no. 1, Mar. 1985, pp. 69–87.

47. Noor, Ahmed K.; and Andersen, C. M.: Computerized Symbolic Manipulation in Structural Mechanics—Progress and Potential. *Comput. & Struct.*, vol. 10, nos. 1/2, Apr. 1979, pp. 95–118.

48. Noor, Ahmed K.; and Nemeth, Michael P.: Analysis of Spatial Beamlike Lattices With Rigid Joints. *Comput. Methods Appl. Mech. & Eng.*, vol. 24, no. 1, Oct. 1980, pp. 35–59.

49. Noor, Ahmed K.; and Russell, William C.: Anisotropic Continuum Models for Beamlike Lattice Trusses. *Comput. Methods Appl. Mech. & Eng.*, vol. 57, no. 3, Sept. 1986, pp. 257–277.

50. Abrate, S.; and Sun, C. T.: Dynamic Analysis of Geometrically Nonlinear Truss Structures. *Comput. & Struct.*, vol. 17, no. 4, 1983, pp. 491–497.

51. Berry, D. T.; and Yang, T. Y.: Simplified Lattice Beam Elements for Geometrically Nonlinear Static, Dynamic, and Postbuckling Analysis. *AIAA J.*, vol. 24, no. 8, Aug. 1986, pp. 1346–1347.

52. Abrate, S.; and Sun, C. T.: Continuum Modelling of Damping in Large Space Structures. *Proceedings of the Second International Conference on Recent Advances in Structural Dynamics*, M. Petyt and H. F. Wolfe, eds., Volume II, Inst. of Sound and Vibration Research, Univ. of Southampton, Apr. 1984, pp. 877–885. (Available from DTIC as AD P003 715.)

53. Sun, C. T.; and Juang, J. N.: Modeling Global Structural Damping in Trusses Using Simple Continuum Models. *AIAA/ASME/ASCE/AHS 24th Structures, Structural Dynamics and Materials Conference, Part 1—A Collection of Technical Papers,* 1983, pp. 722–729. (Available as AIAA-83-1008.)

54. Bazant, Z. P.; and Christensen, M.: Analogy Between Micropolar Continuum and Grid Frameworks Under Initial Stress. *Int. J. Solids & Struct.*, vol. 8, Mar. 1972, pp. 327–346.

55. Noor, Ahmed K.; and Nemeth, Michael P.: Micropolar Beam Models for Lattice Grids With Rigid Joints. *Comput. Methods Appl. Mech. & Eng.*, vol. 21, no. 2, Feb. 1980, pp. 249–263.

56. Russell, William C.: Continuum Modeling Theories for Repetitive Lattice Space Structures. M.S. Thesis, George Washington Univ., Oct. 1985.

57. Nemeth, Michael P.: Continuum Models for Repetitive Lattice Structures With Rigid Joints. M.S. Thesis, George Washington Univ., Sept. 12, 1979.

58. Sun, C. T.; and Yang, T. Y.: A Couple-Stress Theory for Gridwork-Reinforced Media. *J. Elast.*, vol. 5, no. 1, Mar. 1975, pp. 45–58.

59. Chen, C. C.; Sun, C. T.; Bogdanoff, J. L.; and Lo, H.: *Simple Models for Computing Dynamic Responses of Complex Frame Structures.* NSF/RA-780529, Applied Science & Research Applications, National Science Found., Apr. 1978. (Available from NTIS as PB 292 798.)

60. Banks, H. T.; and Crowley, J. M.: *Parameter Identification in Continuum Models.* NASA CR-172132, 1983.

61. Berry, D. T.; Yang, T. Y.; and Skelton, R. E.: Dynamics and Control of Lattice Beams Using Simplified Finite Element Models. *J. Guid., Control, & Dyn.*, vol. 8, no. 5, Sept.-Oct. 1985, pp. 612-619.

62. Juang, J. N.; and Sun, C. T.: System Identification of Large Flexible Structures by Using Simple Continuum Models. *J. Astronaut. Sci.*, vol. 31, no. 1, Jan.-Mar. 1983, pp. 77-98.

63. Lamberson, S. E.; and Yang, T. Y.: Integrated Design of Space Structures Using Lattice Plate Finite Elements. *J. Guid., Control, & Dyn.*, vol. 9, no. 4, July-Aug. 1986, pp. 478-484.

64. Noor, Ahmed K.; and Peters, Jeanne M.: Nonlinear Analysis of Anisotropic Panels. *AIAA J.*, vol. 24, no. 9, Sept. 1986, pp. 1545-1553.

65. Noor, Ahmed K.; and Whitworth, Sandra L.: Model-Size Reduction for the Buckling and Vibration Analyses of Anisotropic Panels. *J. Eng. Mech. Div., ASCE,* vol. 113, no. 2, Feb. 1987, pp. 170-185.

Symbols

A	cross-sectional area of a member of the repeating cell
$[C]$	matrix of stiffness coefficients of the simplified continuum
$C_{11}, C_{12}, C_{13}, \ldots, C_{88}$	stiffness coefficients of the simplified continuum (see tables 2 and 3 and figs. 5 and 6)
$\{C_T\}$	thermal load vector of the simplified continuum
$d_1, d_2, \ldots, d_6$	generalized displacements (see figs. 5 and 6)
E	elastic modulus of the material of a member of the repeating cell
$F_1, F_2, \ldots, F_6$	generalized internal forces in the continuum beam model (see fig. 5)
$\mathcal{G}_{11}, \mathcal{G}_{12}, \mathcal{G}_{22}$	partitions of the matrix $[\mathcal{G}]_c$ (see eqs. (37))
$[\mathcal{G}]_c$	geometric stiffness matrix of the continuum
$[g]$	geometric stiffness matrix of the simplified continuum
$[\mathsf{g}^{(k)}]$	geometric stiffness matrix of member k of the repeating cell
$[K]$	stiffness matrix of the repeating cell
$[K]_c$	stiffness matrix of the continuum
$K_{11}, K_{12}, K_{13}, K_{22}, K_{23}, K_{33}$	partitions of the matrix $[K]_c$ (see eqs. (14))
$\mathcal{K}$	kinetic energy density of the continuum
L	length of a member of the repeating cell
$(\ell, m, n)^{(k)}$	direction cosines of member k
M_2, M_3, M_t	bending and twisting moments in continuum beam model (see fig. 5)
M_{11}, M_{22}, M_{12}	bending and twisting stress resultants in continuum plate model (see fig. 6)
$\mathcal{M}_{11}, \mathcal{M}_{12}, \mathcal{M}_{22}$	partitions of the matrix $[\mathcal{M}]_c$ (see eqs. (15))
$[\mathcal{M}]_c$	matrix of density parameters of the continuum
$[m]$	matrix of density parameters of the simplified continuum model
$m_{11}, m_{12}, m_{13}, \ldots, m_{66}$	density parameters of the simplified continuum (see tables 2 and 3 and figs. 5 and 6)
$[\mathsf{m}^{(k)}]$	consistent mass matrix of member k of the repeating cell
$\bar{N}$	axial force in beamlike lattices (see fig. 5)
N_{11}, N_{22}, N_{12}	extensional stress resultants in continuum plate model (see fig. 6)
$\{P_T\}$	thermal load vector of the repeating cell
$\{P_T\}_c$	thermal load vector of the continuum
$\{P_{T1}\}, \{P_{T2}\}, \{P_{T3}\}$	partitions of the vector $\{P_T\}_c$ (see eqs. (16))
Q_{12}, Q_{13}	transverse shearing forces in the y- and z-directions in continuum beam model
Q_1, Q_2	transverse shear stress resultants in continuum plate model
$[R^{(k)}]$	transformation matrix whose entries are products of direction cosines of member k
$S_1, S_2, \ldots, S_8$	stress resultants in the continuum plate model (see fig. 6)
T^o	temperature parameter (see eqs. (26) and (32))
T	temperature of a member of the repeating cell
$\{T\}_c$	vector of temperature parameters used in describing the continuum
U_c	thermoelastic strain energy density of the continuum

U_1, U_2	contributions to the strain energy of the linear and quadratic terms in the temperature parameters
U_o	isothermal strain energy
u, v, w	displacement components in the coordinate directions
$\bar{u}, \bar{v}, \bar{w}$	displacement parameters characterizing warping and cross-sectional distortions
u^o, v^o, w^o	displacement parameters in the coordinate directions
$\{u\}^{(k)}$	vector of nodal displacements of member k of the repeating cell
$\{u\}_c$	vector of displacement parameters used in describing the continuum
$\{u_{c1}\}, \{u_{c2}\}$	partitions of the vector $\{u_c\}$ (see eqs. (13))
x, y, z	Cartesian coordinates
α	coefficient of thermal expansion of a member of the repeating cell
$[\Gamma_u^{(k)}], [\Gamma_\varepsilon^{(k)}],$ $[\Gamma_T^{(k)}], [\bar{\Gamma}_\Delta^{(k)}]$	transformation matrices (see eqs. (6), (7), (8), and (34))
$\gamma_{xy}^o, \gamma_{xz}^o, \gamma_{yz}^o$	shearing strain parameters in the coordinate planes
$\gamma_{12}, \gamma_{13}, \gamma_{23}$	shearing strains in the coordinate planes
$\{\Delta\}_c$	vector of displacement parameters and their spatial derivatives for the continuum
$\{\Delta_{c1}\}, \{\Delta_{c2}\}$	partitions of the vector $\{\Delta\}_c$ (see eqs. (36))
$\varepsilon^{(k)}$	axial strain of member k of the repeating cell
$\varepsilon_{11}, \varepsilon_{22}, \varepsilon_{33}$	axial strains in the coordinate directions
$\varepsilon_x^o, \varepsilon_y^o, \varepsilon_z^o$	extensional strain parameters in the coordinate directions
$\{\varepsilon\}_c$	vector of strain parameters used in describing the continuum
$\{\varepsilon\}^{(k)}$	vector of strain components in the coordinate directions used in the expansion of $\varepsilon^{(k)}$
$\{\varepsilon_{c1}\}, \{\varepsilon_{c2}\},$ $\{\varepsilon_{c3}\}$	partitions of the vector $\{\varepsilon\}_c$ (see eqs. (12))
$\bar{\kappa}$	strain parameter (see eqs. (25))
$\kappa_x^o, \kappa_y^o, \kappa_z^o,$ $\kappa_t^o, 2\kappa_{xy}^o$	curvature changes and twist parameters
ρ	mass density of the material (see fig. 7)
ϕ_x, ϕ_y, ϕ_z	rotation components
ψ^o	strain parameter (see eqs. (25))
Ω	characteristic geometric property of the repeating cell of the lattice (length of repeating cell for beamlike lattices and planform area of repeating cell for platelike lattices)
ω	frequency of vibration (see figs. 10 to 13)

Subscripts:

b	bottom surface bars
d	diagonals
l	longerons
t	top surface bars
v	verticals

Partial derivatives are denoted as follows:

$$\partial_x \equiv \partial/\partial x, \quad \partial_y \equiv \partial/\partial y, \quad \text{and} \quad \partial_z \equiv \partial/\partial z$$

Superscript t denotes matrix transposition. Superscript (k) denotes member k of the repeating cell.

Table 1. Approaches for Analyzing Repetitive Lattice Structures

Approach	Drawbacks	Possible remedial actions
Direct method: Structure analyzed as a system of discrete finite elements	Computationally very expensive for large lattices	Use of reduction methods for buckling and nonlinear problems (refs. 43 and 44)
Discrete field methods: Equilibrium and compatibility equations are written at a typical joint Taylor series expansions are used to develop differential equations	Development and solution can be substantial for complex lattice configurations	
Periodic structure approaches: Substructuring, combined use of finite elements, and transfer matrix methods Exact representation of stiffness of individual members	Not efficient for complicated configurations and/or for transient response calculations Limited to beamlike lattices with simply supported ends or ring configuration with rigid central mast	Combine with substitute continuum approach
Substitute continuum approach	Local deformation effects not accounted for Ordinary continuum not suitable for lattices with rigid joints	Include local deformation modes in the development Use micropolar continua

Table 2. Stiffness Coefficients and Density Parameters for the Continuum Model of the Beamlike Lattice Structure Shown in Figure 7

Coefficient[a]	Value
C_{11}	2.967×10^8 N
$C_{14} = C_{15} = C_{44} = C_{55}$	3.126×10^7 N
$C_{22} = C_{33}$	1.561×10^9 N-m^2
C_{23}	9.770×10^7 N-m^2
$C_{26} = C_{36} = -\frac{1}{2}C_{66}$	-1.954×10^8 N-m^2
$m_{11} = m_{22} = m_{33}$	6.200 kg/m
$m_{44} = 2m_{55} = 2m_{66}$	56.096 kg-m
m_{56}	1.212 kg-m

[a] C_{11} is the extensional stiffness; C_{22} and C_{33} are the bending stiffnesses; C_{44} and C_{55} are the transverse shear stiffnesses; C_{66} is the torsional stiffness; and C_{14}, C_{15}, C_{23}, C_{26}, and C_{36} are coupling coefficients.

Table 3. Stiffness Coefficients and Density Parameters for the Continuum Model of the Platelike Lattice Structure Shown in Figure 9

Coefficient[a]	Value
$C_{11} = C_{22}$	1.682×10^6 N/m
$C_{12} = C_{33}$	0.4394×10^6 N/m
$C_{14} = C_{25}$	1.456×10^6 N
$C_{15} = C_{24} = C_{36}$	0.3802×10^6 N
$C_{44} = C_{55}$	2.366×10^7 N
$C_{45} = C_{66}$	0.6179×10^7 N-m
$C_{77} = C_{88}$	3.380×10^4 N/m
$m_{11} = m_{22} = m_{33}$	0.1779 kg/m^2
$m_{14} = m_{25}$	0.1418 kg/m
$m_{44} = m_{55}$	2.370 kg

[a] C_{11} and C_{22} are the extensional stiffnesses in the x- and y-directions; C_{33} is the in-plane shear stiffness; C_{44} and C_{55} are the bending stiffnesses in the x- and y-directions; C_{66} is the torsional stiffness; C_{77} and C_{88} are the transverse shear stiffnesses; and C_{14}, C_{15}, C_{24}, C_{25}, and C_{36} are bending-extensional coupling stiffnesses.

Table 4. Maximum Displacements From Continuum Beam Models Compared With Exact Solutions for the Cantilevered Lattice Structure Shown in Figure 7

[Number of repeating cells = 10]

	Continuum model		
Normalized displacements[a]	Coupling terms neglected	Coupling terms included	Exact
$u^o/\bar{N}$, m/N	1.685×10^{-7}	2.135×10^{-7}	2.135×10^{-7}
w^o/Q_{13}, m/N	2.828×10^{-5}	3.028×10^{-5}	2.969×10^{-5}
v^o/Q_{12}, m/N	2.828×10^{-5}	3.028×10^{-5}	2.969×10^{-5}
w^o/M_t, N^{-1}	0	4.270×10^{-7}	4.276×10^{-7}
w^o/M_2, N^{-1}	8.005×10^{-7}	8.540×10^{-7}	8.442×10^{-7}
v^o/M_3, N^{-1}	8.005×10^{-7}	8.540×10^{-7}	8.442×10^{-7}

[a] $\bar{N}$ is the axial force; Q_{13} and Q_{12} are the transverse shearing forces; M_t is the twisting moment; and M_2 and M_3 are the bending moments (see fig. 5).

Table 5. Minimum Vibration Frequencies From Continuum Beam Models Compared With Exact Solutions for the Cantilevered Lattice Structure Shown in Figure 7

[Number of repeating cells = 10]

	Frequency from continuum model, Hz		
Mode	Coupling terms neglected	Coupling terms included	Exact frequency, Hz
1	3.385	3.252	3.280
2	3.385	3.274	3.325
3	13.197	12.977	13.136
4	17.059	15.853	16.353
5	17.059	16.877	17.172
6	38.794	31.608	31.641
7	38.794	35.468	36.116
8	62.185	35.817	37.596
9	62.185	41.704	42.996
10	86.209	56.732	59.198

Table 6. Minimum Vibration Frequencies From Continuum Plate Models Compared With Exact Solutions for the Cantilevered Double-Layered Grid Shown in Figure 9

	$8L \times 8L$ grid		$16L \times 16L$ grid	
Mode	Continuum frequency, Hz	Exact frequency, Hz	Continuum frequency, Hz	Exact frequency, Hz
1	1.253	1.248	0.3887	0.3878
2	2.064	2.087	0.7152	0.7178
3	4.260	4.249	1.703	1.700
4	4.768	4.792	2.182	2.189
5	5.527	5.573	2.346	2.345
6	6.303	6.298	2.387	2.391
7	8.273	8.347	3.436	3.455
8	8.377	8.396	3.670	3.664
9	9.090	9.144	3.979	3.984
10	9.856	9.809	4.360	4.354

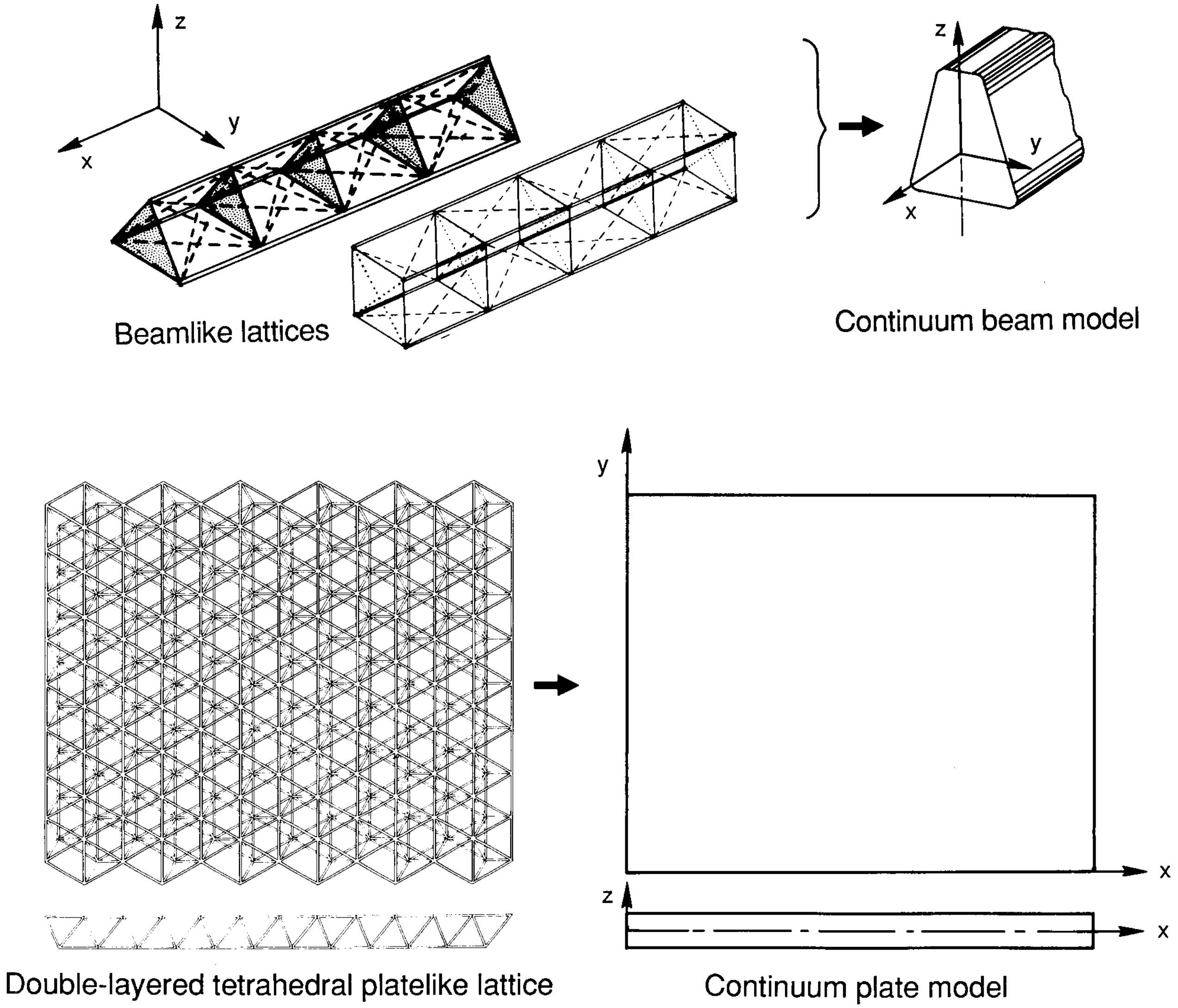


Figure 1. Continuum models for beamlike and platelike lattice structures.

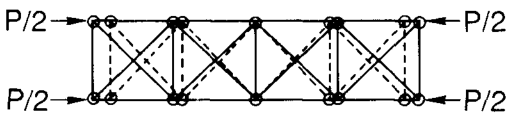
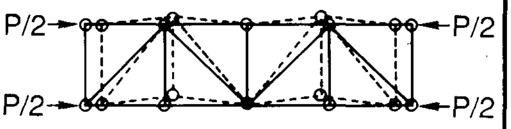
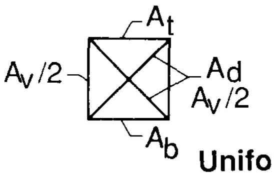
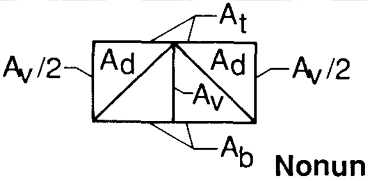
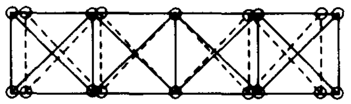
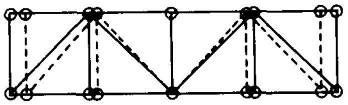
	Double-laced lattice truss	Single-laced lattice truss
Lattice structure and actual deformation		
Repeating cell and strain state within repeating cell	 Uniform	 Nonuniform
Deformation predicted by classical continuum (based on uniform strain state within repeating cell)		

Figure 2. Local deformations in planar lattice trusses subjected to axial loading *P*.

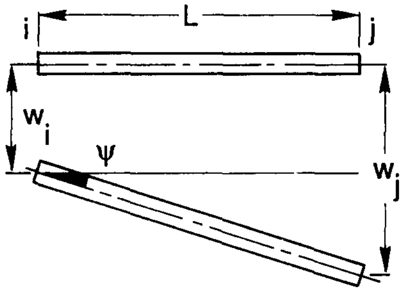
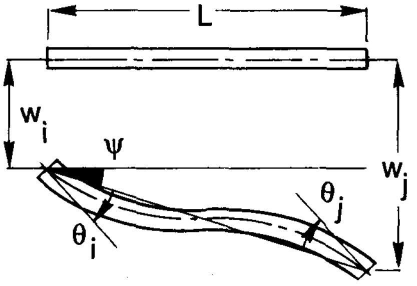
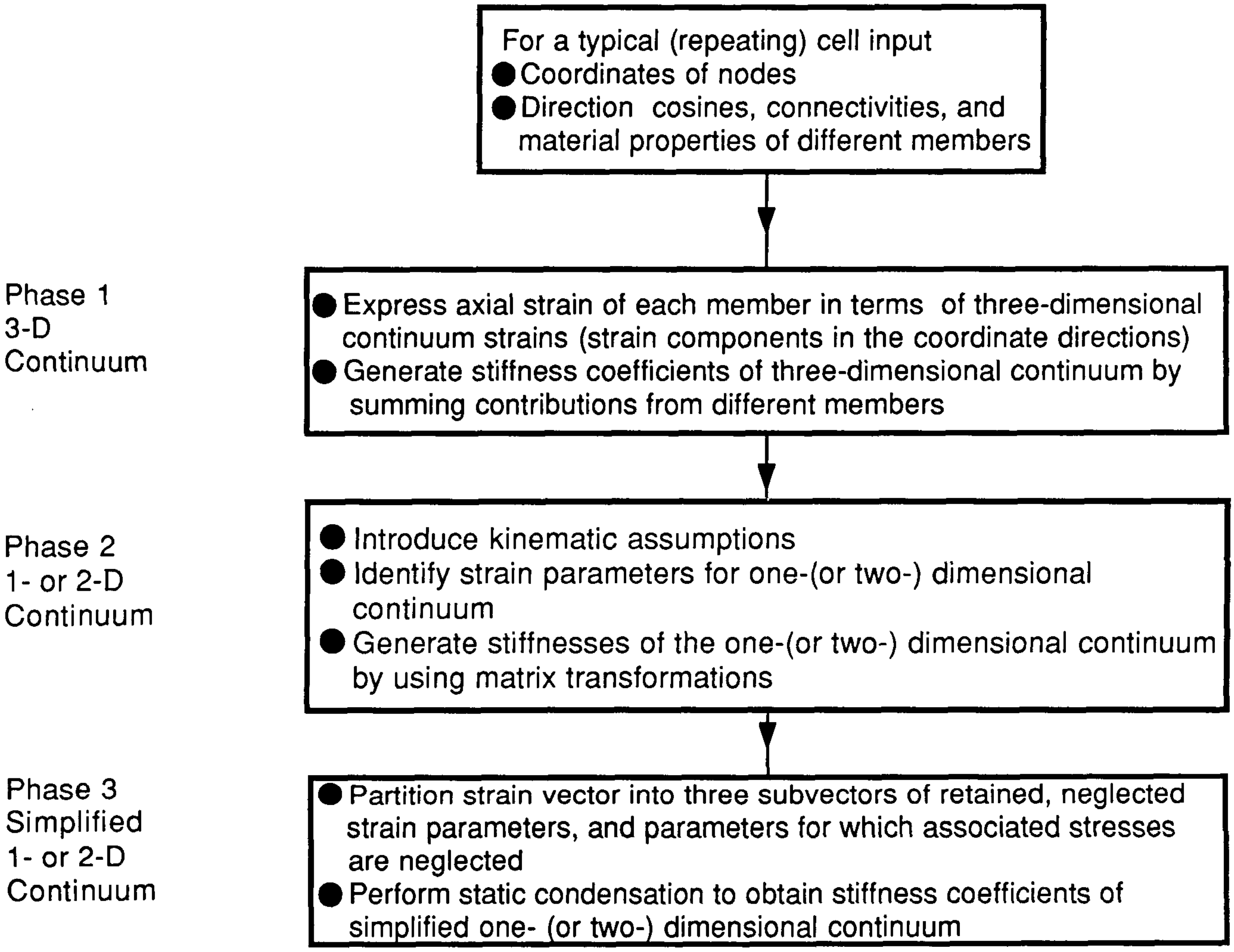
	Truss member (pin joints)	Beam member (rigid joints)
Deformation pattern		
Rotations	$\psi = \frac{1}{L}(w_j - w_i)$	$\psi = \frac{1}{L}(w_j - w_i)$ θ_i, θ_j (joint rotations)
Appropriate continuum	Ordinary (displacement field only)	Micropolar (independent displacement and rotation fields)

Figure 3. Deformation patterns for pin-jointed and rigid-jointed one-dimensional members.



For a typical (repeating) cell input

- ● Coordinates of nodes
- ● Direction cosines, connectivities, and material properties of different members

Phase 1
3-D
Continuum

- ● Express axial strain of each member in terms of three-dimensional continuum strains (strain components in the coordinate directions)
- ● Generate stiffness coefficients of three-dimensional continuum by summing contributions from different members

Phase 2
1- or 2-D
Continuum

- ● Introduce kinematic assumptions
- ● Identify strain parameters for one-(or two-) dimensional continuum
- ● Generate stiffnesses of the one-(or two-) dimensional continuum by using matrix transformations

Phase 3
Simplified
1- or 2-D
Continuum

- ● Partition strain vector into three subvectors of retained, neglected strain parameters, and parameters for which associated stresses are neglected
- ● Perform static condensation to obtain stiffness coefficients of simplified one- (or two-) dimensional continuum

Figure 4. Steps involved in the generation of the stiffness coefficients for simplified one- and two-dimensional continuum models of beamlike and platelike lattice structures.

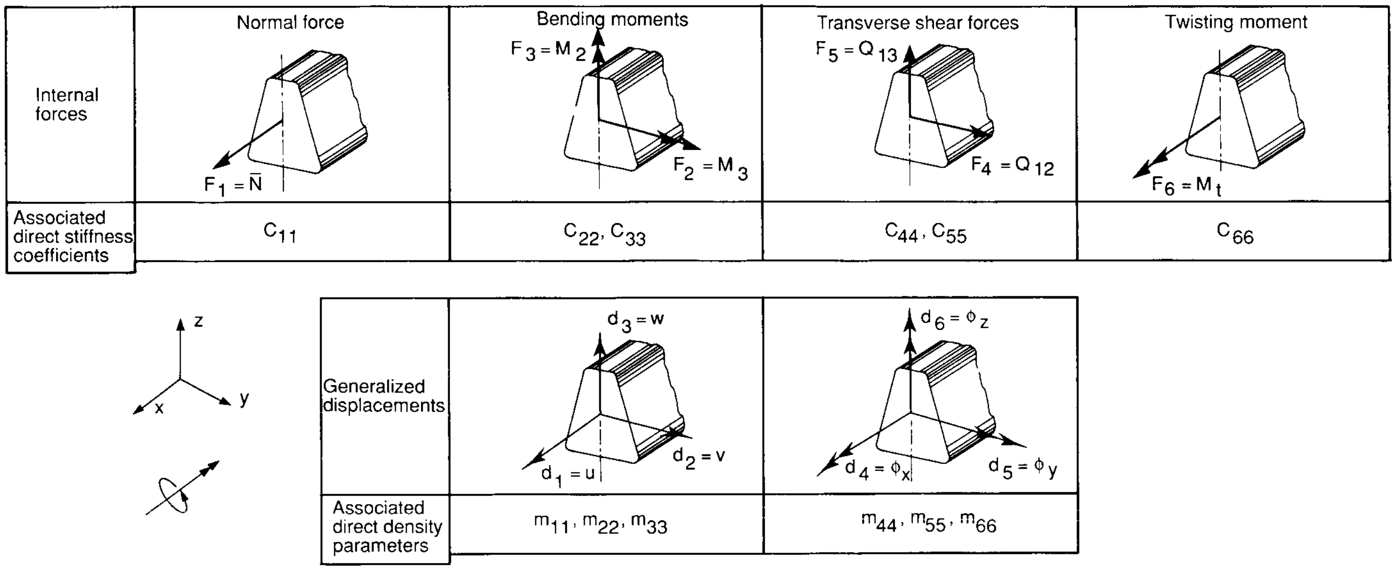
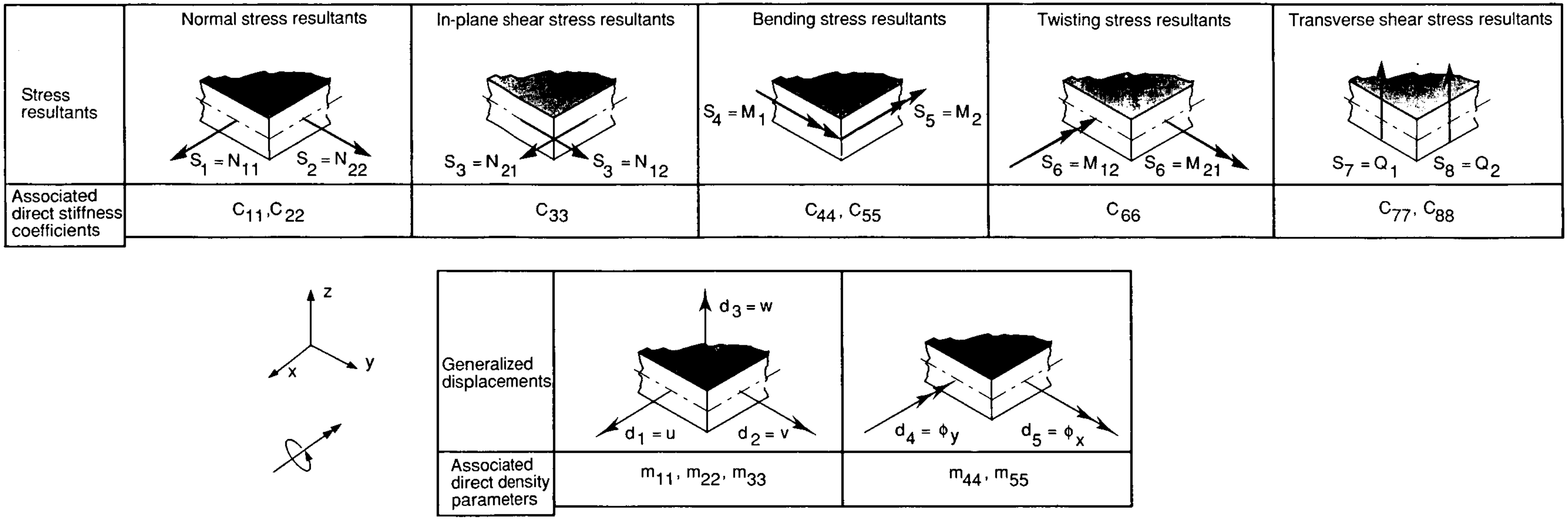


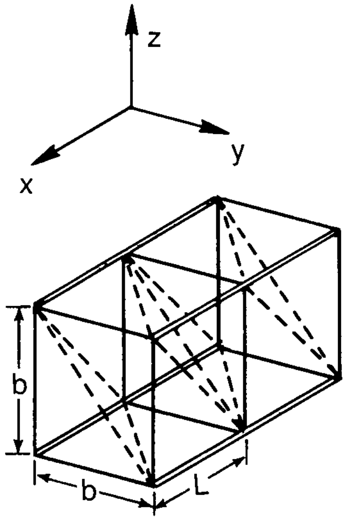
Figure 5. Simplified continuum beam element, sign convention for internal forces and generalized displacements, and associated stiffness and density parameters.



Stress resultants	Normal stress resultants	In-plane shear stress resultants	Bending stress resultants	Twisting stress resultants	Transverse shear stress resultants
	$S_1 = N_{11}$ $S_2 = N_{22}$	$S_3 = N_{21}$ $S_3 = N_{12}$	$S_4 = M_1$ $S_5 = M_2$	$S_6 = M_{12}$ $S_6 = M_{21}$	$S_7 = Q_1$ $S_8 = Q_2$
Associated direct stiffness coefficients	C_{11}, C_{22}	C_{33}	C_{44}, C_{55}	C_{66}	C_{77}, C_{88}

Generalized displacements	$d_1 = u$ $d_2 = v$ $d_3 = w$	$d_4 = \phi_y$ $d_5 = \phi_x$
Associated direct density parameters	m_{11}, m_{22}, m_{33}	m_{44}, m_{55}

Figure 6. Simplified continuum plate element, sign convention for stress resultants and generalized displacements, and associated stiffness and density parameters.



	Cross section area	Length	Designation
Longerons	A_l	L	═══
Diagonals	A_d	d	- - - -
Battens	A_b	b	———

$b = L = 5\ \mathrm{m}$

$A = 2.359 \times 10^{-4}\ \mathrm{m}^2$

$\rho = 1743.8\ \mathrm{kg/m}^3$

$E = 2.482 \times 10^{11}$ Pa (longerons and battens)

$= 2.557 \times 10^{11}$ Pa (diagonals)

Figure 7. Orthogonal tetrahedral beamlike lattice.

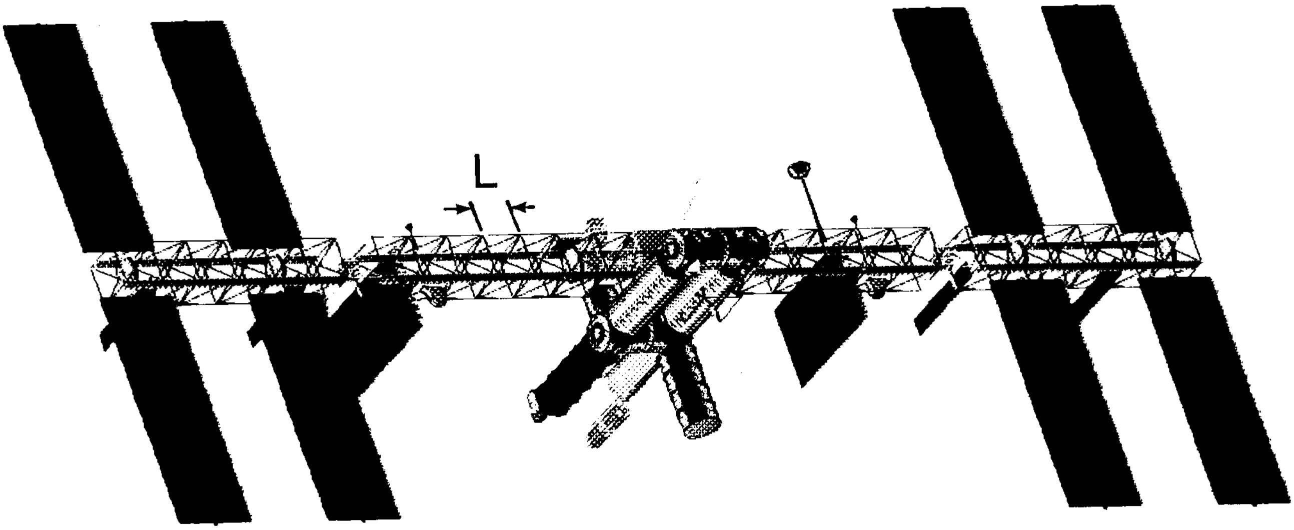
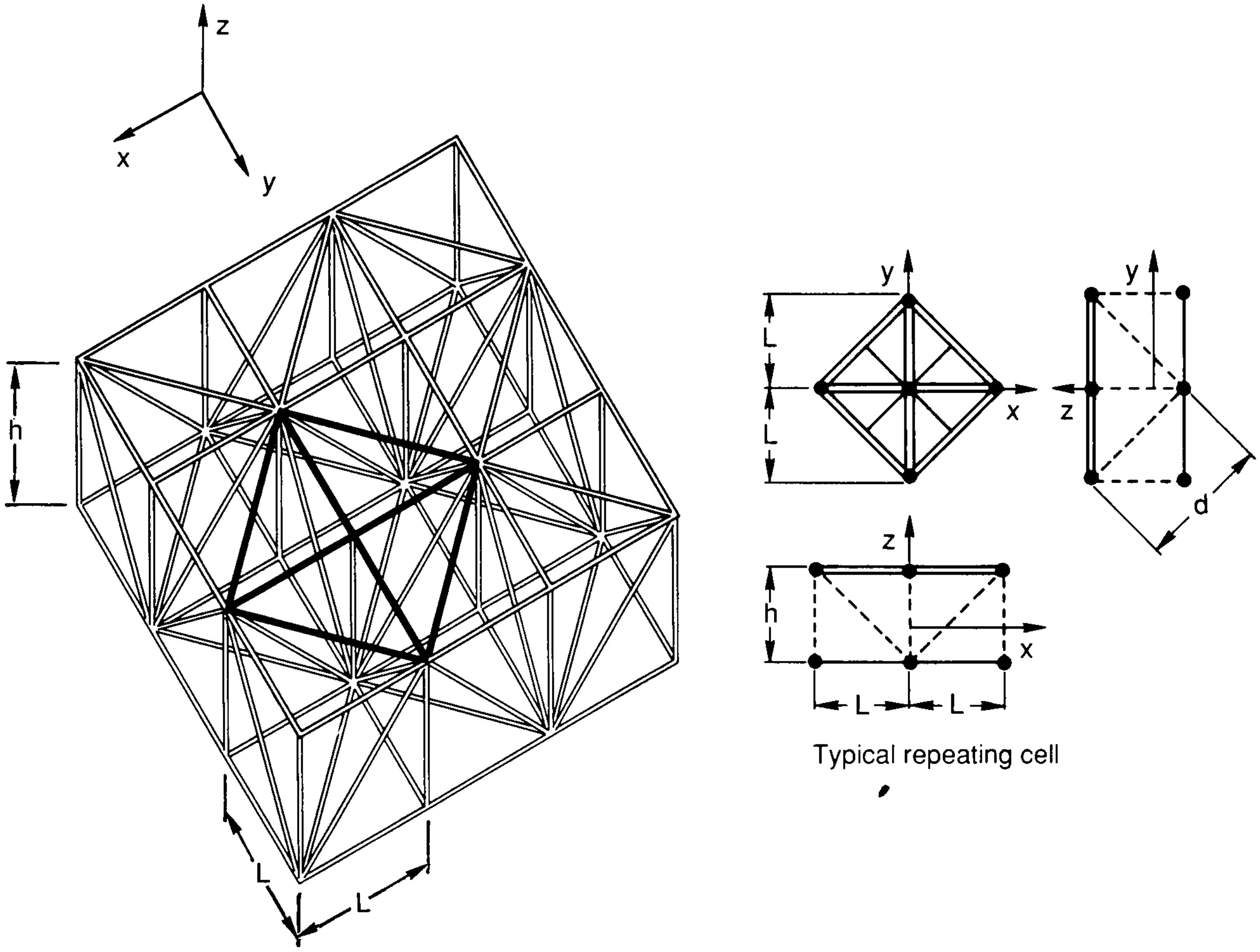


Figure 8. Candidate Space Station configuration (phase I).



Typical repeating cell

$A_t = 80 \times 10^{-6}\ m^2$, $E = 71.7 \times 10^9$ Pa

$A_b = 50 \times 10^{-6} m^2$

$A_d = 10 \times 10^{-6} m^2$

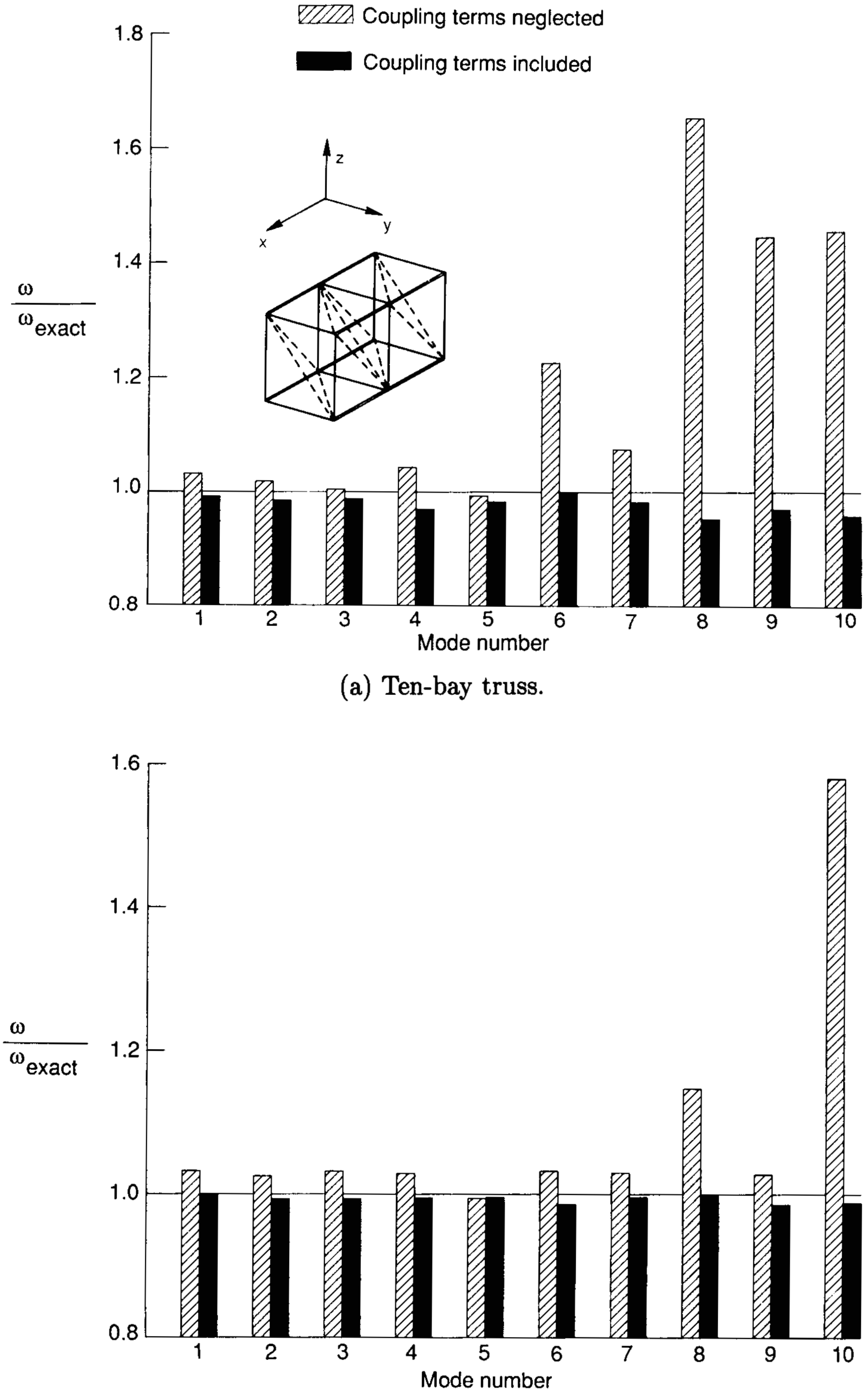
$A_v = 10 \times 10^{-6} m^2$, $\rho_t = \rho_b = \rho_d = \rho_v$

L = 7.5 m $\quad = 2768\ kg/m^3$

h = 7.5 m

	Cross section area	Length	Desig.
Top surf. bars	A_t	$L, \sqrt{2 L}$	═════
Bottom surf. bars	A_b	$L, \sqrt{2 L}$	———
Verticals	A_v	h	- - - - -
Diagonals	A_d	d	- - - - -

Figure 9. Double-layered hexahedral platelike lattice truss.



(a) Ten-bay truss.

(b) Thirty-bay truss.

Figure 10. Accuracy of minimum frequencies from continuum beam models for the cantilevered beamlike lattice structure shown in figure 8.

$\omega_1 = 3.252$ (3.280) $\omega_2 = 3.274$ (3.325)

$\omega_3 = 12.98$ (13.14) $\omega_4 = 15.85$ (16.35)

$\omega_5 = 16.88$ (17.17) $\omega_6 = 31.61$ (31.64)

$\omega_7 = 35.47$ (36.12) $\omega_8 = 35.82$ (37.60)

$\omega_9 = 41.70$ (43.00) $\omega_{10} = 56.73$ (59.20)

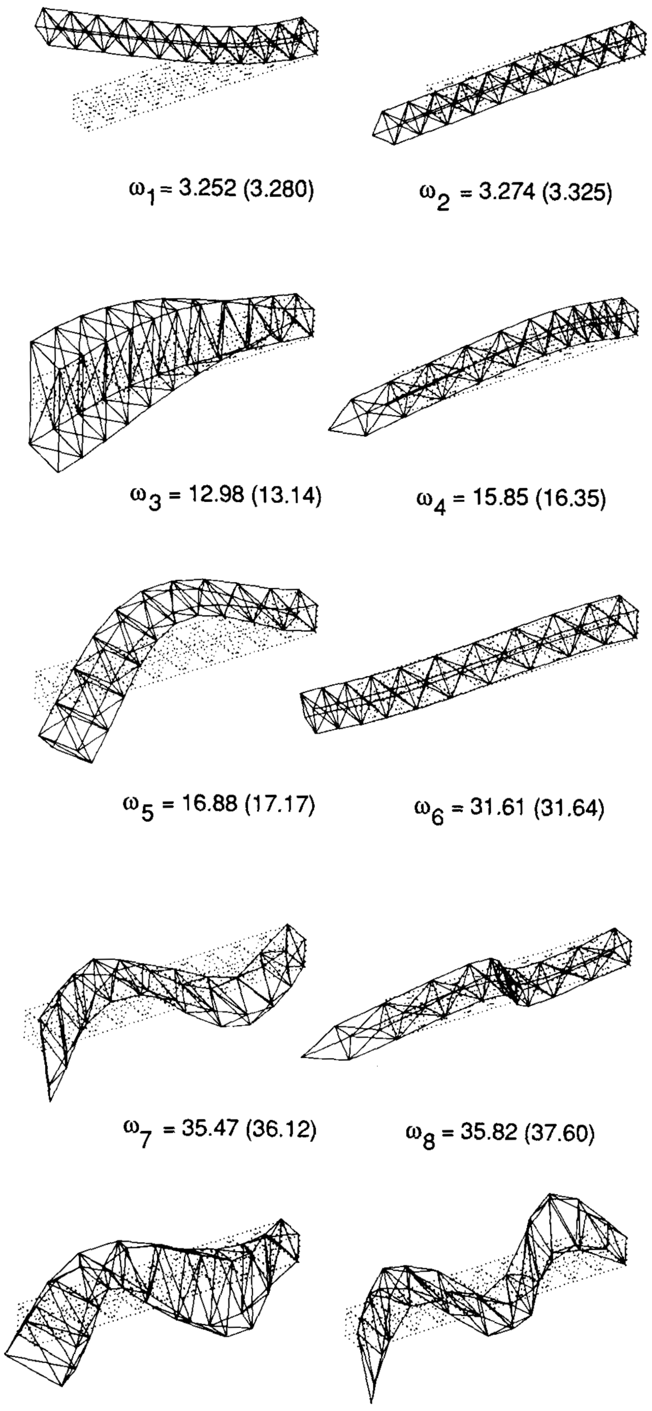
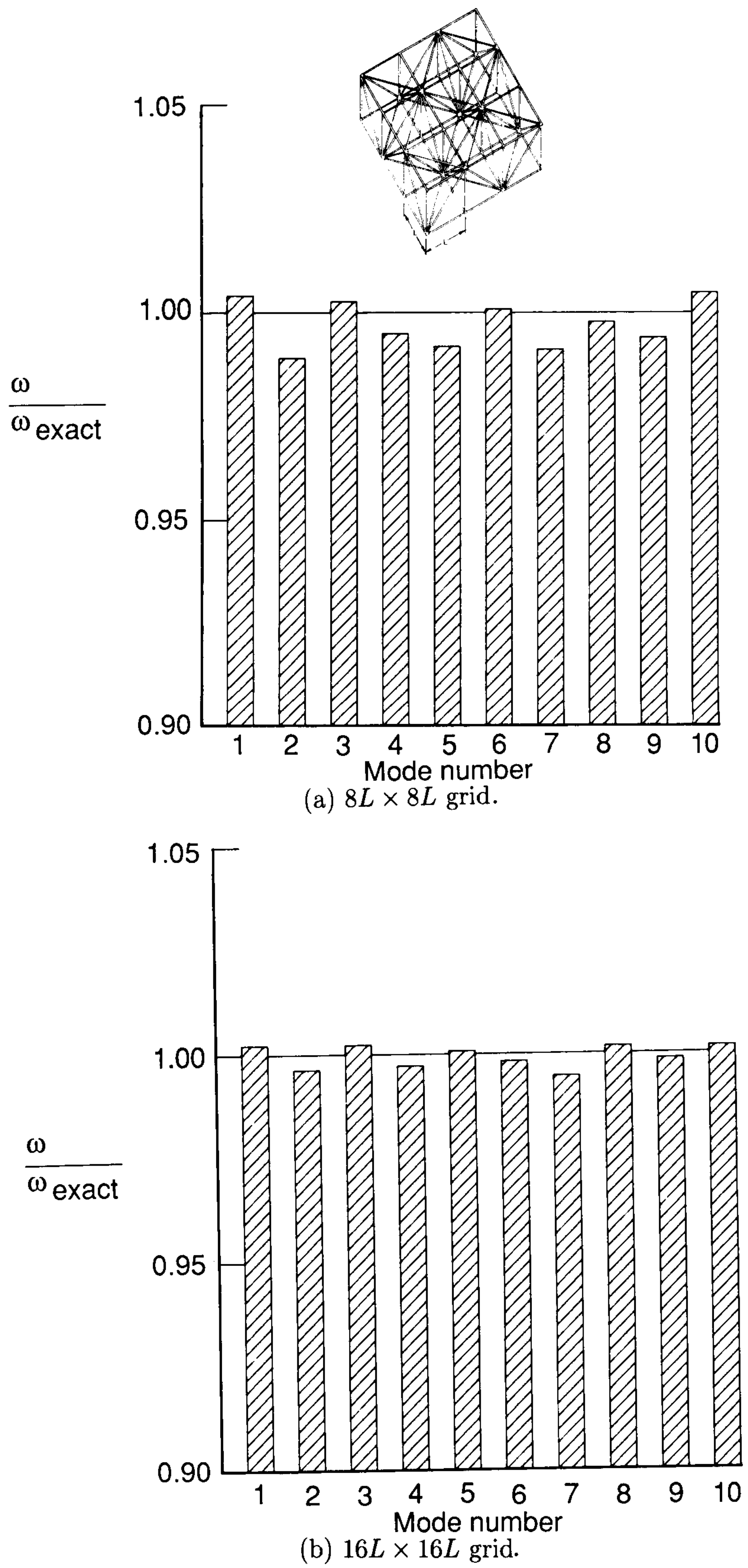


Figure 11. Vibration mode shapes for the cantilevered lattice 10-bay truss shown in figure 8. Numbers in parentheses are exact frequencies (in hertz); other numbers are continuum beam frequencies.



(a) 8L × 8L grid.

(b) 16L × 16L grid.

Figure 12. Accuracy of minimum frequencies from continuum plate models for the cantilevered platelike lattic truss shown in figure 10.

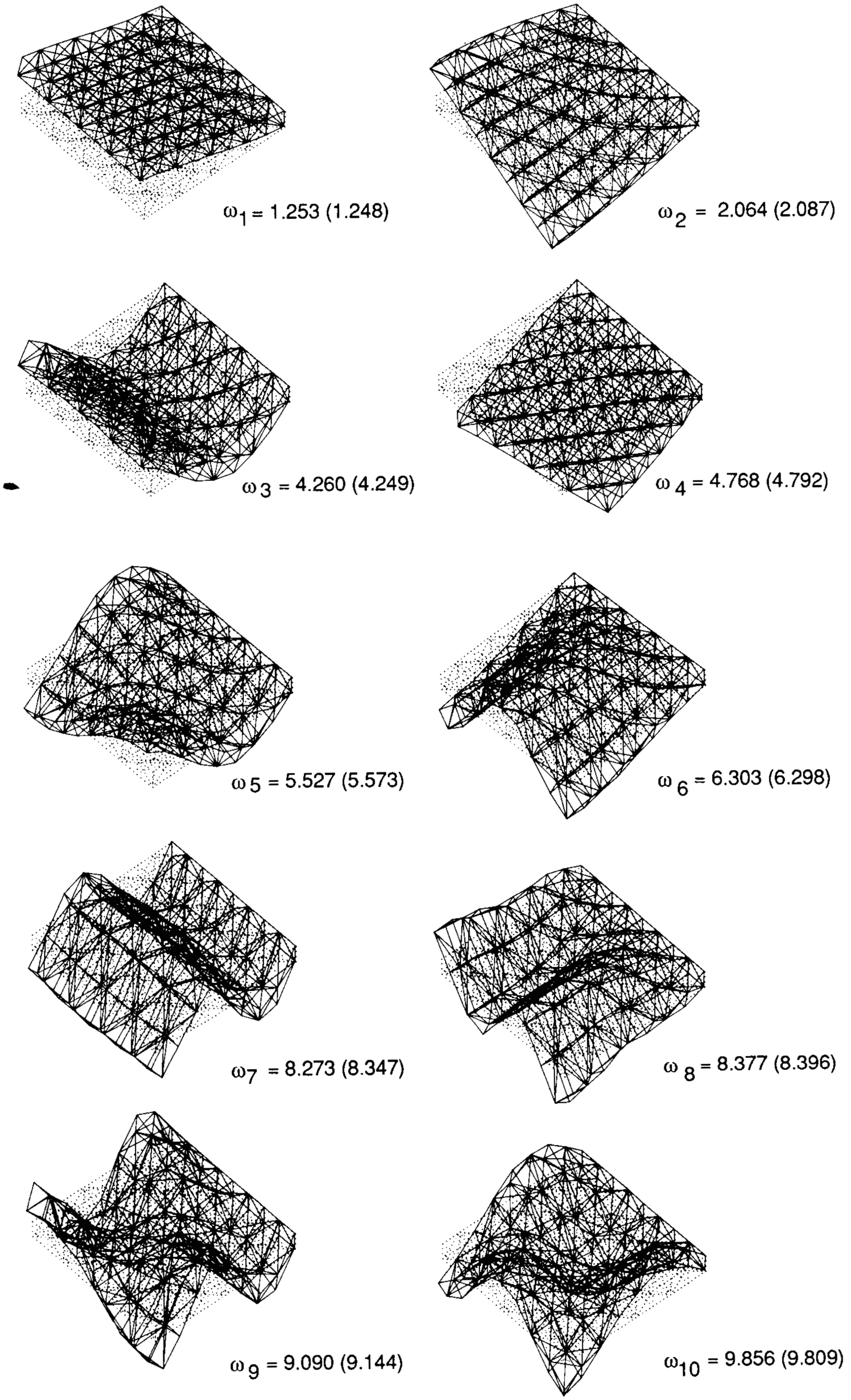

ω_1 = 1.253 (1.248)
ω_2 = 2.064 (2.087)
ω_3 = 4.260 (4.249)
ω_4 = 4.768 (4.792)
ω_5 = 5.527 (5.573)
ω_6 = 6.303 (6.298)
ω_7 = 8.273 (8.347)
ω_8 = 8.377 (8.396)
ω_9 = 9.090 (9.144)
ω_{10} = 9.856 (9.809)

Figure 13. Vibration mode shapes for the cantilevered hexahedral grid shown in figure 10. Numbers in parentheses are exact frequencies (in hertz); other numbers are continuum plate frequencies.

NASA
National Aeronautics and
Space Administration

Report Documentation Page

1. Report No.
NASA TP-2767

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
Continuum Modeling of Large Lattice Structures—Status and Projections

5. Report Date
February 1988

6. Performing Organization Code

7. Author(s)
Ahmed K. Noor and Martin M. Mikulas, Jr.

8. Performing Organization Report No.
L-16360

9. Performing Organization Name and Address
NASA Langley Research Center
Hampton, VA 23665-5225

10. Work Unit No.
506-43-41-02

11. Contract or Grant No.

12. Sponsoring Agency Name and Address
National Aeronautics and Space Administration
Washington, DC 20546-0001

13. Type of Report and Period Covered
Technical Paper

14. Sponsoring Agency Code

15. Supplementary Notes
Ahmed K. Noor: The George Washington University, Joint Institute for Advancement of Flight Sciences, Langley Research Center, Hampton, Virginia. Work supported under NASA Grant NAG1-740.
Martin M. Mikulas, Jr.: Langley Research Center, Hampton, Virginia.

16. Abstract
The status and some recent developments of continuum modeling for large repetitive lattice structures are summarized. Discussion focuses on a number of aspects including definition of an effective substitute continuum, characterization of the continuum model, and the different approaches for generating the properties of the continuum, namely, the constitutive matrix, the matrix of mass densities, and the matrix of thermal coefficients. Also, a simple approach is presented for generating the continuum properties. The approach can be used to generate analytic and/or numerical values of the continuum properties.

17. Key Words (Suggested by Authors(s))
Continuum modeling
Large lattice structures

18. Distribution Statement
Unclassified—Unlimited

Subject Category 18

19. Security Classif.(of this report)
Unclassified

20. Security Classif.(of this page)
Unclassified

21. No. of Pages
77

22. Price
A05

NASA-Langley, 1988
For sale by the National Technical Information Service, Springfield, Virginia 22161-2171