Order Number 9003136

Isoparametric finite elements for linear isotropic micropolar plate bending

Suresh, Vallanore K., D.Eng.Sc.

New Jersey Institute of Technology, 1989

U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

Isoparametric Finite Elements For Linear Isotropic Micropolar Plate Bending

by
Vallanore K. Suresh

Dissertation submitted to the Faculty of Graduate School of
New Jersey Institute of Technology in partial fulfillment of
the requirements for the degree of
Doctor of Engineering Science

Approval Sheet

Title of Dissertation: Isoparametric Finite Elements For Linear Isotropic Micropolar Plate Bending

Name of the Candidate: Vallanore K. Suresh
Doctor of Engineering Science

Dissertation and Abstract Approved: ______________________ ____________
Bernard Koplik — Date
Professor
Mechanical Engineering Department

Signatures of other members of the dissertation committee ______________________ ____________ Date

______________________ ____________ Date

______________________ ____________ Date

______________________ ____________ Date

Abstract

Title of Dissertation	: Isoparametric Finite Elements for Linear Isotropic Micropolar Plate Bending.
VALLANORE K. SURESH	: Doctor of Engineering Science, 1989.
Dissertation Directed by	: Dr. Bernard Koplik Professor Department of Mechanical & Industrial Engineering

Finite element analysis programs for bending of linear isotropic micropolar elasticity are developed in this thesis. Isoparametric two-dimensional and three-dimensional elements are used to solve plate bending problems. Two-dimensional and three-dimensional finite element formulations are developed. Corresponding finite element programs are then developed. Patch tests are performed on the two-dimensional elements. Convergence studies are undertaken for the elements. Several plates are considered to evaluate the finite element scheme. The effects of the coupling factor N and characteristic length l are studied. A post processor with graphical display of displacements is developed.

Numerical results obtained for various plates are in good agreement with analytical solutions. Displacements and moments were not affected by the coupling factor for very small values of the characteristic length.

Vita

Name : Vallanore K. Suresh

Degree and Date to be Conferred : Doctor of Engineering Science, 1989.

Education:

College	Dates	Degree	Date of Degree
Regional Engineering College, India.	1976-1981	BE	June, 1981
New Jersey Institute of Technology	1981-1983	MSME	October, 1983
New Jersey Institute of Technology	1983-1989	D.E.Sc.	May, 1989

Major : Mechanical Engineering.

Positions held:
Special Lecturer, 1985-1989, New Jersey Institute of Technology

Acknowledgment

I wish to express my sincere gratitude and thanks to my thesis advisor, Dr. Bernard Koplik, for his valuable guidance and help throughout the period of this study. I would like to thank the members of the dissertation committee, Dr. Harry Herman, Dr. Rong-Yaw Chen, Dr. Benedict Sun and Dr. Sachio Nakamura for all the encouragement they have given me. I thank Dr. Harry Herman for his constructive criticisms and guidance. I am grately indebted to Dr. Sachio Nakamura of AT & T Bell Labs, who was formerly an Assistant Professor of Mechanical Engineering, for his help and suggestions.

I wish to express my gratitude to my parents and my parents-in-law for the encouragement they have given me throughout the period. A special appreciation is extended to my wife, Rama, for all her patience and understanding during this study.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Introductory Comments

The mechanical state of a deformed body as enunciated by Cauchy[1], is completely characterized by stress and strain tensor components. For a micropolar elastic solid, the state is described by additional couple stress and micro-rotation components. Micropolar materials have extra independent degrees of freedom for local rotation. Micropolar theory is expected to find applications in the treatment of the mechanics of granular materials with elongated rigid grains and composite fibrous materials.

The existance of couple stresses was noted as early as 1879 by Thomson and Tait[2]. The renewed interest in the theory in the 1960's lead to some serious research in this area. Most of the work was on establishment and experimentation in confirming the theory by use of classical case studies. Most work was on simple objects. However, analysis of objects of more complex geometry was not possible. The finite element method gives an easy way out in this regard.

It is interesting to note that the fundamental idea of existance of independent couples was noted first in connection with the beam theory developed by Bernoulli and Euler. Yet, the bending aspect of the micropolar material received little attention from researchers. The present study develops a finite element model for the bending analysis of plates based on Micropoar Theory of Elasticity. Several examples of bending of various plates will be considered. Numerical results are compared with existing classical solutions for specific problems in bending. This study also demonstrates the finite element solution methodology for solving problems of bending of Micropolar materials.

1.2 Historical Development of the Micropolar Theory

Early elasticity theory as developed by Navier, Poisson, Cauchy and Lame , started from the concept of central forces of attraction or repulsion between molecules. The equations of elasticity were then derived through integration or summation of these forces over a body, thereby achieving continuous models of elasticity.

The continuum or macroscopic concept, as implied in the Classical Elasticity Theory, is one in which the local structure of solids on the macroscopic scale(granules, crystals, molecules etc...) is not revealed. Ordinary solids when examined microscopically, are found to be composed of granular, crystals, foreign inclusions, cracks, voids and other inhomogeneties. The Classical Elasticity Theory has been successful in a great majority of applications primarily because of the consideration of large bodies where such micro effects are averaged out. But when a significant dimension of a body approaches it's grain size, the classical theory becomes inaccurate.

The fundamental idea of existance of couples in the beam analysis can be traced back all the way to Bernoulli and Euler. In their bar theory, each section of the bar is associated with a deformation vector and a rotation vector and two types of internal loads, namely, tractions and couples. In plate theory there is a similar situation. Thompson and Tait[2] recorded these independent quantities in the context of bar and plate theories. They show distributed twisting couples acting on infinitesimal areas of transverse sections in the plate theory. They also state that the strains due to these twisting couples(couple stresses) when acting on the boundary, rapidly vanish inward from the boundary.

Voight[3] was the first to include the couple stresses and body couples in the equilibrium equations. The complete formulation of the equations of elasticity including couple stresses and body couples was developed by E. & F. Cosserat[4] in 1909. The Cosserats developed a continuum theory in which each point of a medium has the six degrees of freedom of the rigid body. Following significant advances by Mindlin and Tiersten[5] and Toupin[6] in the area of couple stresses, Indeterminate couple stress Theory came to light as a Cosserat Continuum in which the rotation of a point equals the rotaion of the medium and is therefore not independent of the displacements. A new material property appeared, known as the characteristic length(a couple stress constant).

Eringen and Suhubi[8],[9] in 1964 introduced a non-linear theory of micro-elastic solids which fully account for couple stresses and body couples. Their theory includes spin inertia which the couple stress theory does not. Also, in 1964, Mindlin[10] obtained similar results by deriving a linear theory using a variational approach. Eringen in 1966 renamed the Cosserat Continuum theory as 'Micropolar Theory'.

1.3 Scope of the Thesis

Equations of general Micropolar Elasticity Theory are reviewed first in Chapter 2. Plate bending problem is formulated next. A variational formulation of Micropolar Elasticity Theory is presented in Chapter 3 together with the matrix finite element formulation. Isoparametric plate elements for plate bending are developed next.

Numerical examples are presented in Chapter 5. A convergence study is undertaken for the 4-node two-dimensional element and the 8-node three-dimentional element. In Chapter 6, conclusions of the dissertation and possible future research areas are suggested. FORTRAN coding for the plate analysis is presented in the appendix.

2 Micropolar Elasticity Theory

2.1 Introductory Comments

A micropolar elastic solid is described as a composite of microelements embedded in a deformable matrix. These microelements have independent degrees of freedom to rotate. Materials with crystalline structure and composite materials where hard fine particulate matter or fibers are embedded in relatively soft matrix material, are expected to exhibit micropolar behavior. Evidence of micropolar behavior in bones is also noted by J. F. C. Yang and R. S. Lakes[11]. Figure(1) explains the nature of microrotations in an orthotropic material. The harder fibers which are embedded in a relatively soft matrix cause couple stresses to exist. In fiber composite materials, the dimension of the diameters of the fibers are very small compared to the thickness of the material. Hence these so called micro-elements tend to have microrotations in addition to the normal and shear stresses.

The linear isotropic microelastic solid defined in the Eringen-Suhubi Theory[8] is fully characterized by 18 material constants. By requiring the microrotation tensor to be antisymmetric, a couple stress theory is derived where the stress and couple stress tensors are fully determined. The numbrer of elastic constants is reduced to six. These six constants retained in the linear isotropic Micropolar Elasticity Theory are designated λ, μ, κ, α, β and γ . In addition to Lamé constants, there are four extra constants which characterize the microrotation behavior. The

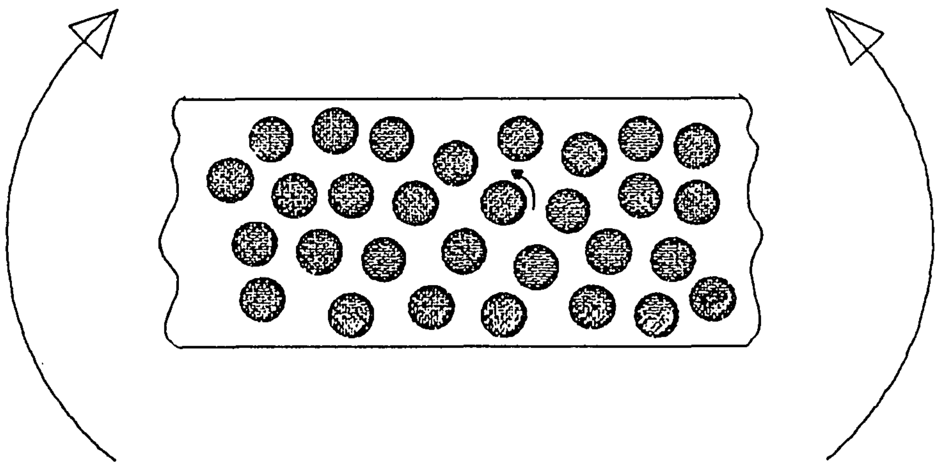


Figure 1: Microrotations in an Orthotropic Material

dimensions of λ, μ, κ are *force/length*^{2} and those of α, β and γ are *force*.

2.2 The Equations of Micropolar Elasticity Theory

Equilibrium Equations

The equilibrium equations in terms of the second order stress and couple stress tensors, t_{ij} and m_{ij} are;

$$t_{ji,j} = 0 \qquad (1)$$

$$m_{ji,j} + e_{ikm}t_{km} = 0 \qquad (2)$$

where e_{ikm} is the permutation tensor.

Constitutive Equations

The linear constitutive equations are written as;

$$t_{ij} = \lambda\varepsilon_{kk}\delta_{ij} + (\mu+\kappa)\varepsilon_{ij} + \mu\varepsilon_{ji} \qquad (3)$$

$$m_{ij} = \alpha\phi_{kk}\delta_{ij} + \beta\phi_{ij} + \gamma\phi_{ji} \qquad (4)$$

where ε_{ij} is the second order microstrain tensor

ϕ_i is the microrotation vector

δ_{ij} is the Kronecker delta

$\lambda, \mu, \alpha, \beta, \gamma, \kappa$ are the constants introduced in the Micropolar Elasticity Theory

Strain-Displacement Relations

The displacements u_j and microrotations ϕ_k are linked to the microstrain tensor ε_{ij} through the strain-displacement relations;

$$\varepsilon_{ij} \;=\; u_{ji} + \varepsilon_{jik}\phi_k \tag{5}$$

The formulation is now complete with equation (1) through equation (5) which total 33 equations in 33 unknown variables, $t_{ij}, m_{ij}, \varepsilon_{ij}, \phi_i$ and u_j.

Boundary Conditions

Boundary conditions on surface s may be prescribed as;

$$t_{(n)i} \;=\; t_{ji}n_j \tag{6}$$

$$m_{(n)i} \;=\; m_{ji}n_j \tag{7}$$

where $t_{(n)i}, m_{(n)i}$ are surface stress and surface couple vectors,

subscript n refers to the boundary surface whose normal vector is n_j.

In additon, displacements and microrotations may be prescribed over a portion of s in addition to tractions and couples as a mixed boundary conditions.

Compatibility Conditions

To limit arbitrariness with which ε_{ij} and $\phi_{i,j}$ are prescribed, some constraints are to be applied, known as the Compatibility Conditions. They are;

$$\varepsilon_{ik,j} - \varepsilon_{jk,i} + e_{ikm}\phi_{m,j} - e_{jkm}\phi_{m,i} \;=\; 0 \tag{8}$$

Restrictions on Micropolar Elastic Moduli

Eringen[12] uses the principle of non-negative internal energy to restrict values of the micropolar elastic moduli. They are given by;

$$
\begin{gathered}
0 \leq 3\lambda + 2\mu + \kappa \quad , \quad 0 \leq 2\mu + \kappa \\
0 \leq 3\alpha + \beta + \gamma \quad , \quad -\gamma \leq \beta \leq \gamma \\
0 \leq \kappa \quad , \quad 0 \leq \gamma
\end{gathered} \tag{9}
$$

Additional Material Parameters

In addition to the six constants mentioned before, the micropolar theory additionally has two more material parameters. They are (a) characteristic length in bending, l and (b) coupling factor, N. Both the characteristic length and the coupling factor are material specific.

In the linear couple stress theory, Mindlin and Tiersten called the ratio of couple stress to curvature the *bending and twisting modulus.* The ratio of this modulus to the shear modulus has a dimension of $length^2$ and it's square root defines a characteristic length. This material parameter characterises the difference between materials with or without couple stresses.

The coupling factor, N specifies the fractional value of the coupling that exists between the microelements and their surroundings. Figure(2) shows a comparison between various fractions of N. If N=0, the microelements are completely

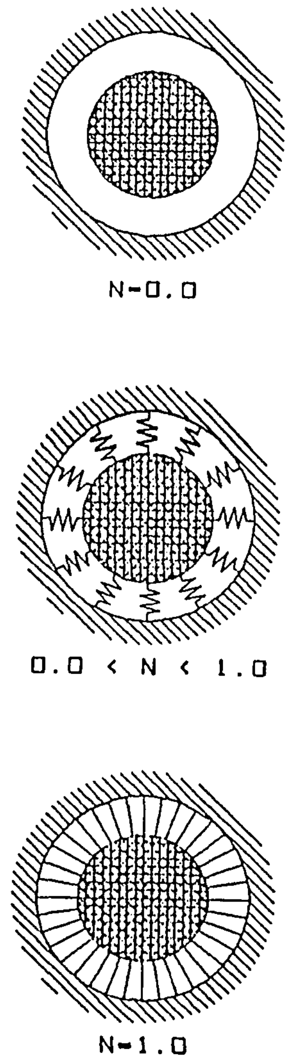


Figure 2: Differences Between Various Values of the Coupling Factor, N for Micropolar Materials

separated from their surroundings. If N=1, they are rigidly connected to the surroundings. If N=0 the spring constant values are zero and if N=1, the spring constants are infinite or rigid connections exist. Any intermediate values for the spring constants may be considered as having a fractional value for N. Cowin[27] defined these parameters in terms of the material constants as,

$$l^2 = \frac{\gamma}{2(2\mu + \kappa)} \tag{10}$$

$$N = \sqrt{\frac{\kappa/2}{\mu + \kappa}} \tag{11}$$

$$\nu = \frac{\lambda}{(2\mu + 2\lambda + \kappa)} \tag{12}$$

The last of the above equations gives a relationship between the micropolar constants and ν , the poisson's ratio of the Classical Elasticity Theory.

2.3 Formulation of Micropolar Plate Bending

The treatment of a plate of uniform thickness has been undertaken by Timoshenko and Krieger[12]. In the classical case, the displacement field is assumed to contain three displacements w, ψ_x, ψ_y along the principal axes. The strain field is computed as;

$$\begin{aligned} \varepsilon_{xx} &= z\frac{\partial^2 w}{\partial x^2} \\ \varepsilon_{yy} &= z\frac{\partial^2 w}{\partial y^2} \\ \varepsilon_{xy} &= z\frac{\partial^2 w}{\partial x \partial y} \end{aligned} \tag{13}$$

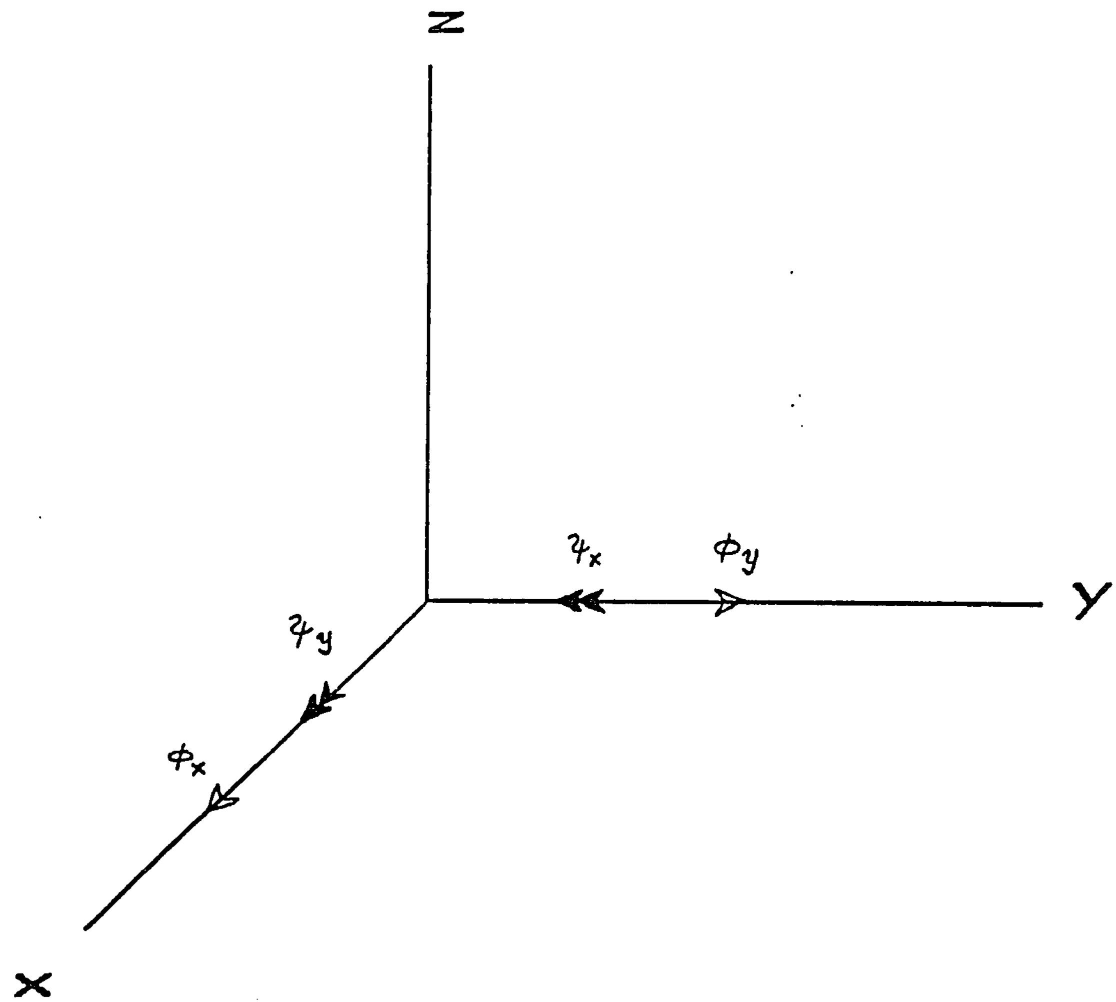


Figure 3: Sign Conventions Used for the Nodal Field Variables.

Mindlin's plate theory incorporates the effects of transverse shear stress and the strain field with shear as given by;

$$
\begin{aligned}
\varepsilon_{xx} &= \frac{\partial \psi_x}{\partial x} \\
\varepsilon_{yy} &= z\frac{\partial \psi_y}{\partial y} \\
\varepsilon_{xy} &= \frac{z}{2}\left(\frac{\partial \psi_x}{\partial y} + \frac{\partial \psi_y}{\partial x}\right) \\
\varepsilon_{xz} &= \frac{1}{2}\left(\frac{\partial w}{\partial x} - \psi_x\right) \\
\varepsilon_{yz} &= \frac{1}{2}\left(\frac{\partial w}{\partial y} - \psi_y\right)
\end{aligned}
\tag{14}
$$

The present study uses a plate model similar to the Mindlin's plate theory, i.e., transverse shear effects are included in the present study. Sign conventions for the plate bending are shown in Figure (3).

2.3.1 Pure Bending of a Rectangular Micropolar Plate

In the case of pure bending of a rectangular plate , Gauthier[13] has derived the equations for plate bending for micropolar materials without shear effects. By adding shear effects to his equations, the following strain-displacement relations are obtained:

$$
\begin{aligned}
\varepsilon_{xx} &= z\frac{\partial \psi_x}{\partial x} \\
\varepsilon_{yy} &= z\frac{\partial \psi_y}{\partial y} \\
\varepsilon_{xy} &= z\frac{\partial^2 w}{\partial x \partial y} - \phi_z
\end{aligned}
\tag{15}
$$

$$\varepsilon_{yx} = z\frac{\partial^2 w}{\partial x \partial y} + \phi_z$$
$$\varepsilon_{xz} = \frac{\partial w}{\partial x} + \phi_y$$
$$\varepsilon_{yz} = \frac{\partial w}{\partial y} - \phi_x$$

where ϕ_x, ϕ_y and ϕ_z are microrotations about x, y, z axes.

In the finite element formulation, ϕ_z is assumed to be zero. Gauthier derived the plate equations for pure bending without shear effects. These relations are given by

$$\begin{aligned}
t_{xx} &= \frac{E}{(1-\nu^2)}\frac{M_x z}{(D+\gamma h)} \\
t_{yy} &= \frac{E\nu}{(1-\nu^2)}\frac{M_x z}{(D+\gamma h)} \\
m_{xy} &= \frac{\gamma M_x}{D+\gamma h} \\
m_{yx} &= \frac{\beta M_x}{D+\gamma h} \\
\varepsilon_{xx} &= \frac{M_x z}{D+\gamma h} \\
\varepsilon_{zz} &= \frac{\nu M_x z}{(1-\nu)(D+\gamma h)} \\
u_x &= \frac{M_x x z}{D+\gamma h} \\
u_z &= -\frac{M_x}{2(D+\gamma h)}\left(x^2 + \frac{\nu z^2}{1-\nu}\right) \\
\phi_y &= \frac{M_x x}{D+\gamma h}
\end{aligned} \tag{16}$$

In the above equations, D is the symbol for flexural rigidity of the Classical

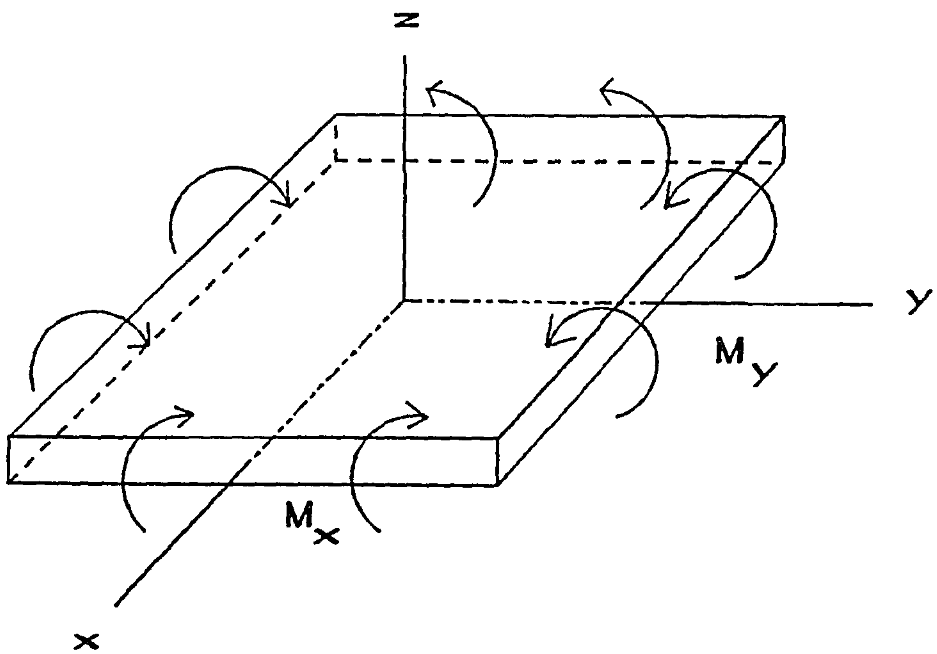


Figure 4: Pure Bending of a Micropolar Plate

Elasticity Theory and is given by

$$D = \frac{Eh^3}{12(1-\nu^2)}$$

These equations are for pure cylindrical bending of a micropolar plate. The lateral moment M_y necessary for cylindrical bending can be found from the second and forth equations above as,

$$M_y = \frac{(D\nu - \beta h)M_x}{D + \gamma h}$$

3 Variational and Finite Element Formulations of Micropolar Elasticity

3.1 Introductory Comments

A variational formulation of Micropolar Elasticity Theory originally developed by Nakamura et. al.[16], is presented here. A formulation for orthotropic micropolar elasticity is developed. The total potential energy for a body composed of an anisotropic micropolar linear elastic material is developed and is used to formulate a displacement type finite element method of analysis. Several types of finite elements are then presented.

3.2 Variational Formulation of Micropolar Elasticity

3.2.1 Principle of Virtual Work

Under virtual displacement and virtual microrotation, the virtual work done by the external load may be written,

$$\delta W_{virt.} = \int\int_V \{G_i \delta u_i + C_i \delta\phi_i\}\ dV + \int_{S_t} \left\{T_i^{(v)} \delta u_i + M_i^{(v)} \delta\phi_i\right\} ds \qquad (17)$$

where G_i and C_i are the applied body force and body couple,

T_i and M_i are the surface force and surface couple tractions,

S_t is the portion of the surface boundary of the body where surface tractions and surface couples are prescribed,

u_i is the displacement for the i-direction

ϕ_i is the microrotation for the i- direction

Using Cauchy's formula and the Divergence Theorem, the virtual work, extended to the entire surface s will become;

$$\begin{aligned}\delta W_{virt.} &= \int\int_V \{(\tau_{ji,j} + G_i)\delta U_i + (m_{ji,j} + e_{ijk}\tau_{jk} + C_i)\delta\phi_i \\ &\quad +(\tau_{ji}\delta U_{i,j} + m_{ji}\delta\phi_{i,j} - e_{ijk}\tau_{jk}\delta\phi_i)\}\, dV\end{aligned}$$

The equilibrium equations for Cosserat Elasticity are given by:

$$\begin{aligned}\tau_{ji,j} + B_i &= 0 \\ e_{ijk}\tau_{jk} + m_{ji,j} + C_i &= 0\end{aligned}$$

Substituting the equilibrium equations in to the expression for virtual work, the first and second terms vanish, leaving

$$\begin{aligned}\delta W_{virt.} &= \int\int_V \{G_i\delta u_i + C_i\delta\phi_i\}\ dV + \int_{S_t} \left\{T_i^{(v)}\delta u_i + M_i^{(v)}\delta\phi_i\right\} ds \qquad (18) \\ &= \int\int_V \{\tau_{ji}\delta_{i,j} + m_{ji}\delta\phi_{i,j} - e_{ijk}\tau_{jk}\delta\phi_i\}\, dV\end{aligned}$$

The virtual work for small deformations is finally given as;

$$\delta W_{virt.} = \int\int_V \{\tau_{ji}\delta\varepsilon_{ji} + m_{ji}\delta\phi_{i,j}\}\, dV \qquad (19)$$

This is the principle of virtual work that gives the relationships between a deformation field $(\delta u_i, \delta\varepsilon_{ij}, \delta\phi_i, \delta\phi_{ij})$ and the stress field, (τ_{ji}, m_{ji}) for any constitutive material of small deformation.

3.2.2 Principle of Minimum Potential Energy

Assuming existence of strain energy density,

$$U_0 \;=\; U_0(\varepsilon_{ji}, \phi_{i,j})$$

such that

$$\tau_{ji} = \frac{\partial U_0}{\partial \varepsilon_{ji}} \quad , \quad m_{ji} = \frac{\partial U_0}{\partial \phi_{i,j}} \tag{20}$$

By Gibb's theorem, we know that this strain energy is positive definite. By equation(19),

$$\int\int_V \{\tau_{ji}\delta\varepsilon_{ji} + m_{ji}\delta\phi_{i,j}\}\, dV - \int\int_V \{G_i\delta u_i + C_i\delta\phi_i\}\; dV \tag{21}$$
$$- \int_{S_t} \left\{T_i^{(v)}\delta u_i + M_i^{(v)}\delta\phi_i\right\} ds \;=\; 0$$

Therefore this can be written,

$$\delta U - \delta V \;=\; 0 \tag{22}$$

where δU is given by the first term of equation(21) and δV is given by the last two terms of equation(21). The total potential energy π of an elastic body is defined as:

$$\pi \;=\; U \;-\; V \tag{23}$$

where U is the strain energy and V is the work done by the external loads acting on the body. Hence equation(22) in fact represents variation of potential energy set to zero.

$$\delta\pi = 0 \tag{24}$$

For linear constitutive relations, from Equation(21),

$$U = \frac{1}{2}\int\int_V (\tau_{ji}\varepsilon_{ji} + m_{ji}\phi_{i,j})dV \tag{25}$$

This is the strain energy expression of micropolar elasticity. Equation(24) says stationary point of total potential energy $\pi = U - V$ gives the solution of equilibrium equations and Cauchy's equations. Since strain energy U is positive definite, the stationary point is actually a minimum point of π.

3.3 Finite Element Formulation of Micropolar Elasticity

A finite element formulation based on general micropolar elasticity is presented here. This is reviewed from the work of Nakamura[18]. From the previous section, we have a minimum potential energy functional of Micropolar Elasticity Theory as

$$\begin{aligned}\pi = {} & \frac{1}{2}\int\int_V \{\tau_{ji}\delta\varepsilon_{ji} + m_{ji}\delta\phi_{i,j}\}\,dV - \int\int_V \{G_i\delta u_i + C_i\delta\phi_i\}\ dV \\ & - \int_{S_t}\left\{T_i^{(v)}\delta u_i + M_i^{(v)}\delta\phi_i\right\}ds\end{aligned} \tag{26}$$

Subdividing the entire domain V as $V = \sum_{k=1}^{n} V_k$, the minimum potential energy functional for each subdomain or finite element, can be defined as,

$$\pi_k = \frac{1}{2}\int\int_{V_k} \{\tau_{ji}\delta\varepsilon_{ji} + m_{ji}\delta\phi_{i,j}\}\, dV - \int\int_{V_k} \{G_i\delta u_i + C_i\delta\phi_i\}\ dV - \int_{S_k} \left\{T_i^{(v)}\delta u_i + M_i^{(v)}\delta\phi_i\right\} ds \qquad (27)$$

Hence the total potential energy is given by

$$\pi = \sum_{k=1}^{n} \pi_k \qquad (28)$$

The problem now is to minimize π_k for each finite element V_k with surface s_k. Hence by introducing appropriate shape functions N_u and N_ϕ, we have for all $x \in V_k$,

$$u(x) = N_u(x)u^e \quad , \quad \phi(x) = N_\phi(x)\phi^e \qquad (29)$$

where u_x and ϕ_x are displacement and microrotation field variable vectors inside V_k , respectively, and

u^e and ϕ^e are nodal field variable vectors of the finite element V_k .

Using the expressions for the micropolar strain tensor

$$\varepsilon = Lu_x + M\phi_x \qquad (30)$$

where L is a differential operator and M is a permutation matrix. Substituting in Equation(29) yields

$$\varepsilon = L(N_u u^e) + M(N_\phi \phi^e)$$

$$= [LN_u, MN_\phi] \left\{ \begin{array}{c} u^e \\ \phi^e \end{array} \right\}$$

$$\varepsilon = B_0 U^e \tag{31}$$

Similarly, for $\phi_{i,j}$,

$$\nabla\phi_{i,j} = B_1 U^e \tag{32}$$

Hence the constitutive equations become

$$\tau = D_0\varepsilon = D_0 B_0 U^e \tag{33}$$

$$m = D_1 \nabla\phi = D_1 B_1 U^e$$

Using these and by taking the first variation of π with respect to nodal field variables and equating to zero, the discretized equilibrium equations are obtained.

$$k^e U^e = F_V^e + F_S^e \tag{34}$$

and

$$k^e = \int\int_V (B_0^T D_0 B_0 + B_1^T D_1 B_1) dV \tag{35}$$

where k^e is the element stiffness matrix of micropolar elasticity and is a generalized form of the classical case where the second term was missing. The B_0, B_1, D_0 and D_1 matrices for the stiffness are given in the next chapter where isoparametric finite element implementations are presented.

4 Isoparametric Elements for Plate Bending

4.1 Introductory Comments

A displacement type finite element method is employed in this thesis for studying the micropolar effects in bending of plates. The displacements at the nodes are assumed as the unknown variables. In this approach the compatibility conditions in and among the elements are satisfied initially. Then the governing equations are written for each node using the equilibrium conditions in terms of nodal displacements. It must be noted that the thin plate thoery itself gives approximate solutions. Any finite element approximation doubles the difference from the actual values. However, several simple and efficient finite elements are in use which give satisfactory results for plate bending problems.

Isoparametric finite elements are considered in this work. In isoparametric formulation, the relationship between the element displacement at any point and the element nodal displacements is obtained directly through the use of a shape function. The nodal coordinates and the nodal displacements are expressed using the same interpolation functions using the natural coordinate system of each element. The natural local coordinate system permits specification of a point within the element by a set of coordinates whose values lie between -1 and +1. As such this generalizes and simplifies the formulation and facilitates numerical integration which is a requirement in the isoparametric formulation. This is useful in applications where curved elements are to be used.

The sampling points and weights in the Gauss-Legendre numerical integration scheme are shown in Table 1. Values upto the order 4 only are shown since the maximum integration order used in this thesis is 3. For two-dimensional elements, numerical integration is carried out in the x and y directions only and in the z direction, explicit integration is used. These steps are shown later, in the stiffness matrix listings. A general three-dimensional element is developed initially in this thesis with eight nodes and six degrees of freedom at each node. Simple two-dimensional elements are considered next with four and eight nodes and five degrees of freedom at each node.

4.2 Convergence Criteria

For monotonic convergence of a finite element, the element should be *complete* and *compatible*. Completeness of an element requires that the displacement functions of the element must be able to represent the rigid body displacements and the constant strain states. If more and more elements are added, or consequently in the limit as each element approaches very small size, the strain in each element must approach a constant value. This, together with the element's ability to undergo displacement modes without developing stresses, constitutes completeness of the element.

Gauss-Integration Order	r_i	s_i
1	+0.00000 00000 00000	2.00000 00000 00000
2	+0.57735 02691 89626	1.00000 00000 00000
3	+0.77459 66692 41483	0.55555 55555 55556
	+0.00000 00000 00000	0.88888 88888 88889
4	+0.86113 63115 94053	0.34785 48451 37454
	+0.33998 10435 84856	0.65214 51548 62546

Table 1: Sampling Points and Weights in Gauss- Legendre Numerical Integration

Compatibility means that the displacements within the elements and across the element boundaries must be continuous. For plane stress problems, only the displacements must be continuous - hence, they are termed C^0 - class problems. On the other hand, for plate bending problems, in addition to the displacements, the derivatives of the displacements must be continuous for the two-dimensional analysis. These elements are termed C^1 - continuity elements. Elements which are complete and compatible are called *conforming elements.*

For two-dimensional plane-stress, plane-strain, axi-symmetric analyses and in three-dimensional analysis, where only u, v, w degrees of freedom are used as nodal point field variables, compatibility can be obtained in a relatively easy way. But the compatibility requirements are difficult to satisfy in plate bending analysis. However, research shows that a number of non-conforming plate bending elements can be used to yield very good results.

For two-dimensional analysis, C^1 continuity requires that the derivatives of the transverse displacement must be included as degrees of freedom at each node of the element. Alternatively, so called $p-class$ elements may be used, where a higher order polynomial is used for interpolation. Since the micropolar theory already has microrotations as additional degrees of freedom, the present study considers C^0 elements for the bending problem to reduce the total number of degrees of freedom at each node of the element.

4.3 Three-Dimensional Eight Node Element for Bending

The three-dimensional element considered in this thesis has eight nodes. There are six degrees of freedom at each node - three displacements and three microrotations. The element is shown in Figure(5). The displacement and microrotation fields can be interpolated by using the following shape functions:

$$\begin{aligned}
N_1 &= \frac{1}{8}(1+r)(1+s)(1+t) \\
N_2 &= \frac{1}{8}(1+r)(1-s)(1-t) \\
N_3 &= \frac{1}{8}(1+r)(1+s)(1-t) \\
N_4 &= \frac{1}{8}(1-r)(1+s)(1-t) \\
N_5 &= \frac{1}{8}(1-r)(1-s)(1+t) \\
N_6 &= \frac{1}{8}(1+r)(1-s)(1+t) \\
N_7 &= \frac{1}{8}(1+r)(1+s)(1+t) \\
N_8 &= \frac{1}{8}(1-r)(1+s)(1+t)
\end{aligned} \tag{36}$$

The displacement and microrotation fields can be interpolated as:

$$\begin{aligned}
u &= \sum_{i=1}^{q} N_i u_i \\
v &= \sum_{i=1}^{q} N_i v_i \\
w &= \sum_{i=1}^{q} N_i w_i \\
\phi_x &= \sum_{i=1}^{q} N_i \phi_{xi}
\end{aligned} \tag{37}$$

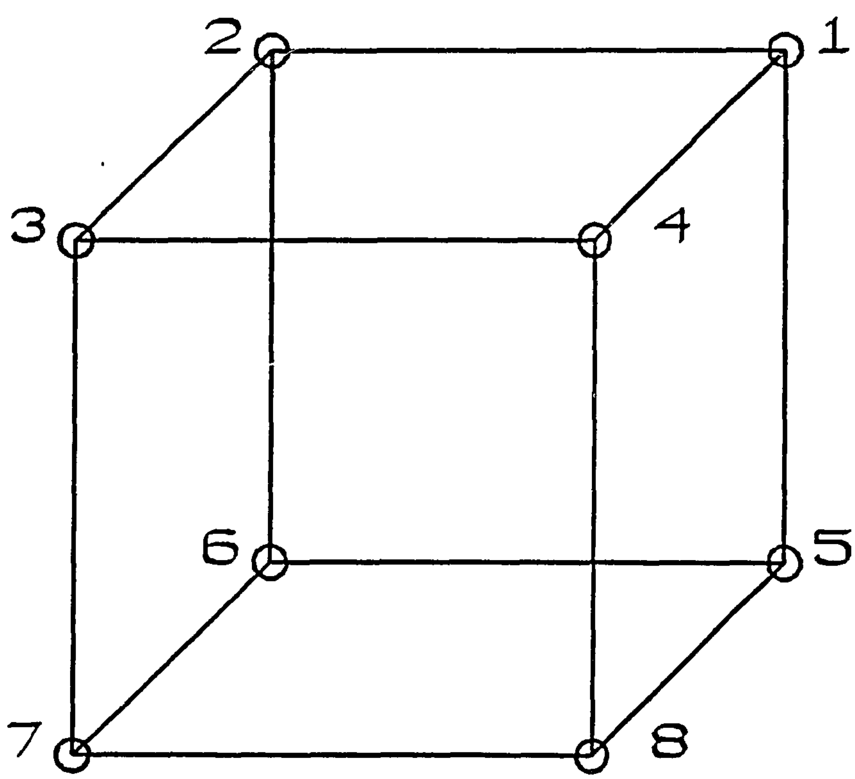


Figure 5: Three Dimensional Eight Node Element with Six Degrees of Freedom at Each Node.

$$\phi_y = \sum_{i=1}^{q} N_i \phi_{yi}$$

$$\phi_z = \sum_{i=1}^{q} N_i \phi_{zi}$$

where $u,\ v,\ w$ are the displacements

$\phi_x,\ \phi_y,\ \phi_z$ are the microrotations.

Here q is the number of nodes of the element. Defining nodal displacement vector U^e as,

$$U^e = [u_1, v_1, w_1, \cdots, \phi_{xq}, \phi_{yq}, \phi_{zq}]^T \tag{38}$$

Equation(37) can be expressed in a compact form as;

$$\left\{ \begin{array}{c} u(r,s,t) \\ v(r,s,t) \\ w(r,s,t) \\ \phi_x(r,s,t) \\ \phi_y(r,s,t) \\ \phi_z(r,s,t) \end{array} \right\} = \left[\begin{array}{ccc} N_1 00000 & ... & N_q 00000 \\ 0N_1 0000 & ... & 0N_q 0000 \\ 00N_1 000 & ... & 00N_q 000 \\ 000N_1 00 & ... & 000N_q 00 \\ 0000N_1 0 & ... & 0000N_q 0 \\ 00000N_1 & ... & 00000N_q \end{array} \right] U^e \tag{39}$$

The N matrix on the right hand side of Equation(39) is a 6 by 48 matrix since the element has eight nodes and each node has six degrees of freedom. The shape functions in the above equations are in terms of the element natural coordinate system. Since the strain-displacement relations involve derivatives of the displacement field in global coordinates, a transformation from natural to the global coordinate system is required. This is accomplished by the *Jacobian* transformation matrix as follows:

$$\left\{ \begin{array}{c} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{array} \right\} = \left[J^{-1}\right] \left\{ \begin{array}{c} \frac{\partial}{\partial r} \\ \frac{\partial}{\partial s} \\ \frac{\partial}{\partial t} \end{array} \right\}$$

Here J is called the Jacobian matrix defined by,

$$[J] = \begin{bmatrix} \frac{\partial x}{\partial r} & \frac{\partial x}{\partial s} & \frac{\partial x}{\partial t} \\ \frac{\partial y}{\partial r} & \frac{\partial y}{\partial s} & \frac{\partial y}{\partial t} \\ \frac{\partial z}{\partial r} & \frac{\partial z}{\partial s} & \frac{\partial z}{\partial t} \end{bmatrix}$$

Hence,

$$\left\{ \begin{array}{c} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{array} \right\} = \left\{ \begin{array}{c} \nabla_1 \\ \nabla_2 \\ \nabla_3 \end{array} \right\} \tag{40}$$

where $\nabla_1, \nabla_2, \nabla_3$ are the differential operators defined by,

$$\begin{aligned} \nabla_1 &= J^{-1}(1,1)\frac{\partial}{\partial r} + J^{-1}(1,2)\frac{\partial}{\partial s} + J^{-1}(1,3)\frac{\partial}{\partial t} \\ \nabla_2 &= J^{-1}(2,1)\frac{\partial}{\partial r} + J^{-1}(2,2)\frac{\partial}{\partial s} + J^{-1}(2,3)\frac{\partial}{\partial t} \\ \nabla_3 &= J^{-1}(3,1)\frac{\partial}{\partial r} + J^{-1}(3,2)\frac{\partial}{\partial s} + J^{-1}(3,3)\frac{\partial}{\partial t} \end{aligned}$$

Substituting Equations(39) and (40) into strain-displacement relations, the following strain-displacement matrix B_0 can be obtained:

$$\begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ \varepsilon_{xy} \\ \varepsilon_{xz} \\ \varepsilon_{yx} \\ \varepsilon_{yz} \\ \varepsilon_{zx} \\ \varepsilon_{zy} \end{bmatrix} = B_0 U^e \tag{41}$$

where B_0 is a 9 by 48 matrix for the eight node element.

$$\begin{aligned}
B0(nelem,1,1) = B0(nelem,1,7)....B0(nelem,1,43) &= \nabla_1 N_i \\
B0(nelem,2,2) = B0(nelem,2,8)....B0(nelem,2,44) &= \nabla_2 N_i \\
B0(nelem,3,3) = B0(nelem,3,9)....B0(nelem,3,45) &= \nabla_3 N_i \\
B0(nelem,4,2) = B0(nelem,4,8)....B0(nelem,4,44) &= \nabla_1 N_i \\
B0(nelem,4,6) = B0(nelem,4,12)...B0(nelem,4,48) &= -N_6 \\
B0(nelem,5,3) = B0(nelem,5,9)....B0(nelem,5,45) &= \nabla_1 N_i \\
B0(nelem,5,5) = B0(nelem,5,11)...B0(nelem,5,47) &= +N_5 \\
B0(nelem,6,1) = B0(nelem,6,7)....B0(nelem,6,43) &= \nabla_2 N_i \\
B0(nelem,6,6) = B0(nelem,6,12)...B0(nelem,6,48) &= +N_6 \\
B0(nelem,7,3) = B0(nelem,7,9)....B0(nelem,7,45) &= \nabla_2 N_i \\
B0(nelem,7,4) = B0(nelem,7,10)...B0(nelem,7,46) &= -N_4 \\
B0(nelem,8,1) = B0(nelem,8,7)....B0(nelem,8,43) &= \nabla_3 N_i
\end{aligned}$$

$$B0(nelem,8,5) = B0(nelem,8,11)...B0(nelem,8,47) \;=\; -N_5$$

$$B0(nelem,9,2) = B0(nelem,9,8)....B0(nelem,9,44) \;=\; \nabla_3 N_i$$

$$B0(nelem,9,4) = B0(nelem,9,10)...B0(nelem,9,46) \;=\; +N_4$$

where nelem is the element number.

Similarly for microrotation gradient, the following B_1 matrix can be obtained:

$$\begin{bmatrix} \phi_{xx} \\ \phi_{yy} \\ \phi_{zz} \\ \phi_{xy} \\ \phi_{xz} \\ \phi_{yx} \\ \phi_{yz} \\ \phi_{zx} \\ \phi_{zy} \end{bmatrix} = B_1 U^e \tag{42}$$

where B_1 is a 9 by 48 matrix for the eight-node element.

$$B1(nelem,1,1) = B1(nelem,1,7)....B1(nelem,1,43) \;=\; \nabla_1 N_i$$

$$B1(nelem,2,2) = B1(nelem,2,8)....B1(nelem,2,44) \;=\; \nabla_2 N_i$$

$$B1(nelem,3,3) = B1(nelem,3,9)....B1(nelem,3,45) \;=\; \nabla_3 N_i$$

$$B1(nelem,4,2) = B1(nelem,4,8)....B1(nelem,4,44) \;=\; \nabla_1 N_i$$

$$B1(nelem,5,3) = B1(nelem,5,9)....B1(nelem,5,45) \;=\; \nabla_1 N_i$$

$$B1(nelem,6,1) = B1(nelem,6,7)....B1(nelem,6,43) \;=\; \nabla_2 N_i$$

$$B1(nelem,7,3) = B1(nelem,7,9)....B1(nelem,7,45) \;=\; \nabla_2 N_i$$

$$B1(nelem,8,1) = B1(nelem,8,7)....B1(nelem,8,43) \;=\; \nabla_3 N_i$$

$$B1(nelem,9,2) = B1(nelem,9,8)....B1(nelem,9,44) \quad = \quad \nabla_3 N_i$$

The B_0 and B_1 matrices derived above can be substituted in Equation(35) to obtain the element stiffness matrix k^e. To carry out the volume integral of the Equation(35), numerical integration of Gaussian Quadrature shown in Table(1) is used in the program.

$$k^e \quad = \quad \sum_{i,j}^{q} t_{ij}\alpha_{ij}({B_{0ij}}^T D_0 B_{0ij})DET + \sum_{i,j}^{q} t_{ij}\alpha_{ij}({B_{1ij}}^T D_1 B_{1ij})DET \qquad (43)$$

where α_{ij} are the weighting factors of Gaussian Quadrature.

The element stiffness matrices calculated in the above equations are assembled into a global stiffness matrix in a symmetric banded form with only the upper triangular part stored. Boundary conditions of the prescribed displacements and the microrotations are imposed by modification of the corresponding rows and columns of this global stiffness matrix. Similarly the force vector is generated by superposing element force vectors in Equation(34). The linear algebraic equations can be solved for nodal displacements using the skyline approach described in reference [28].

4.4 Two-Dimensional Four and Eight-Node Elements

In this section, two elements are presented for the two-dimensional cases. First, a simple four-node element is presented and then the number of nodes is extended to eight.

For the four-node element shown in Figure (6), each node has five degrees

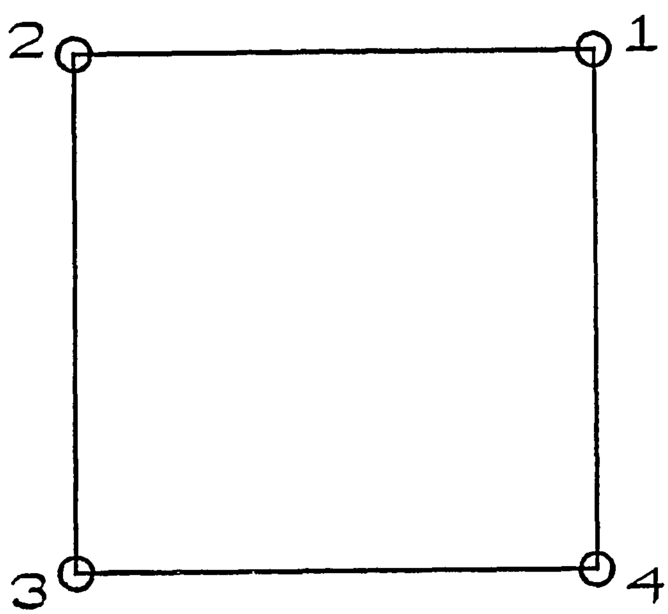


Figure 6: Two-Dimensional Four-Node Element with Five Degrees of Freedom at Each Node.

of freedom: one transverse displacement, two rotations and two microrotations.

The displacements and microrotatons fields inside each element is interpolated as follows:

$$
\begin{aligned}
w &= \sum_{i=1}^{q} N_i w_i \\
\psi_x &= \sum_{i=1}^{q} N_i \psi_{xi} \\
\psi_y &= \sum_{i=1}^{q} N_i \psi_{yi} \\
\phi_x &= \sum_{i=1}^{q} N_i \phi_{xi} \\
\phi_y &= \sum_{i=1}^{q} N_i \phi_{yi}
\end{aligned} \tag{44}
$$

where w is the transverse displacement

ψ_x, ψ_y are slopes

ϕ_x, ϕ_y are the microrotations

q is the total number of nodes in the element

N_i are the shape functions approximating the variables.

These shape functions N_i for the four-node element are given by,

$$
\begin{aligned}
N_1 &= \frac{1}{4}(1+r)(1+s) \\
N_2 &= \frac{1}{4}(1-r)(1+s) \\
N_3 &= \frac{1}{4}(1-r)(1-s) \\
N_4 &= \frac{1}{4}(1+r)(1-s)
\end{aligned} \tag{45}
$$

where r and s are the natural co-ordinates of the element.

Similarly, for the eight-node two-dimensional element, shown in Figure(7), the shape functions can be written as:

$$
\begin{aligned}
N_1 &= \frac{1}{4}(1+r)(1+s)(r+s-1) \\
N_2 &= \frac{1}{4}(1-r)(1+s)(-r+s-1) \\
N_3 &= \frac{1}{4}(1-r)(1-s)(-r-s-1) \\
N_4 &= \frac{1}{4}(1+r)(1-s)(r-s-1) \\
N_5 &= \frac{1}{2}(1+s)(1-r^2) \\
N_6 &= \frac{1}{2}(1-r)(1-s^2) \\
N_7 &= \frac{1}{2}(1-s)(1-r^2) \\
N_8 &= \frac{1}{2}(1+r)(1-s^2)
\end{aligned}
$$

The nodal displacement vector U^e is defined as,

$$U^e = [w_1, \psi_{x1}, \cdots, \phi_{x4}, \phi_{y4}]^T \tag{46}$$

The co-ordinates x, y are also interpolated in the same way as the displacement field:

$$x = \sum_{i=1}^{q} N_i x_i \tag{47}$$

$$y = \sum_{i=1}^{q} N_i y_i \tag{48}$$

Equation(44) can be expressed in a compact form as;

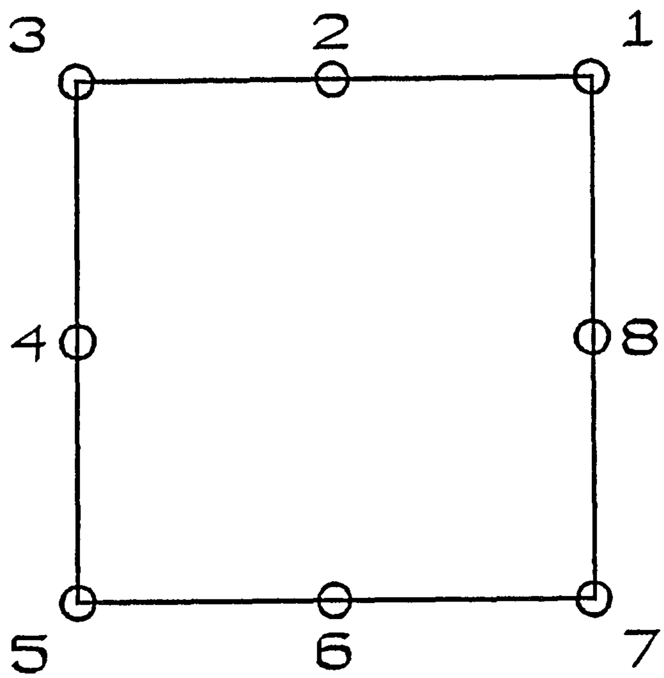


Figure 7: Two-Dimensional Eight-Node Element with Five Degrees of Freedom at Each Node.

$$\left\{ \begin{array}{c} w(r,s) \\ \psi_x(r,s) \\ \psi_y(r,s) \\ \phi_x(r,s) \\ \phi_y(r,s) \end{array} \right\} = \left[\begin{array}{ccccccccccccc} N_1 & 0 & 0 & 0 & 0 & . & . & N_q & 0 & 0 & 0 & 0 \\ 0 & N_1 & 0 & 0 & 0 & . & . & 0 & N_q & 0 & 0 & 0 \\ 0 & 0 & N_1 & 0 & 0 & . & . & 0 & 0 & N_q & 0 & 0 \\ 0 & 0 & 0 & N_1 & 0 & . & . & 0 & 0 & 0 & N_q & 0 \\ 0 & 0 & 0 & 0 & N_1 & . & . & 0 & 0 & 0 & 0 & N_q \end{array} \right] U^e (49)$$

The shape functions in the above equations are in terms of the element natural co-ordinate system. Since the strain-displacement relations involve derivatives of the displacement field in global co-ordinates, a transformation from natural to the global co-ordinate system is required. This is accomplished by the Jacobian matrix shown below:

$$\left\{ \begin{array}{c} \frac{\partial}{\partial r} \\ \frac{\partial}{\partial s} \end{array} \right\} = \left[\begin{array}{cc} \frac{\partial x}{\partial r} & \frac{\partial y}{\partial r} \\ \frac{\partial x}{\partial s} & \frac{\partial y}{\partial s} \end{array} \right] \left\{ \begin{array}{c} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{array} \right\}$$

or,

$$\left\{ \begin{array}{c} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{array} \right\} = \left[J^{-1} \right] \left\{ \begin{array}{c} \frac{\partial}{\partial r} \\ \frac{\partial}{\partial s} \end{array} \right\}$$

where,

$$\left[J^{-1} \right] = \frac{1}{\Delta} \left[\begin{array}{cc} \frac{\partial y}{\partial s} & -\frac{\partial x}{\partial s} \\ -\frac{\partial y}{\partial r} & \frac{\partial x}{\partial r} \end{array} \right]$$

The operator Δ is given by,

$$\Delta = \left| \begin{array}{cc} \frac{\partial x}{\partial r} & \frac{\partial y}{\partial r} \\ \frac{\partial x}{\partial s} & \frac{\partial y}{\partial s} \end{array} \right|$$

For the derivatives of the shape functions, in terms of operators ∇_1 and ∇_2,

$$\begin{aligned}\frac{\partial N_i}{\partial x} &= \nabla_1 N_i \\ \frac{\partial N_i}{\partial y} &= \nabla_2 N_i\end{aligned}$$

where

$$\begin{aligned}\nabla_1 &= \frac{1}{\Delta}\left[\frac{\partial y}{\partial s}\frac{\partial}{\partial r} - \frac{\partial x}{\partial s}\frac{\partial}{\partial s}\right] \\ \nabla_2 &= \frac{1}{\Delta}\left[-\frac{\partial y}{\partial r}\frac{\partial}{\partial r} + \frac{\partial x}{\partial r}\frac{\partial}{\partial s}\right]\end{aligned}$$

Substituting this and Equation(49) into strain-displacement equations, the following strain-displacement matrix B_0 can be obtained:

$$\begin{aligned}\begin{Bmatrix}\varepsilon_{xx}\\ \varepsilon_{yy}\\ \varepsilon_{xy}\\ \varepsilon_{yx}\\ \varepsilon_{xz}\\ \varepsilon_{yz}\end{Bmatrix} &= \begin{bmatrix}0 & \nabla_1 & 0 & 0 & 0\\ 0 & 0 & \nabla_2 & 0 & 0\\ 0 & 0 & \nabla_1 & 0 & 0\\ 0 & \nabla_2 & 0 & 0 & 0\\ \nabla_1 & -1 & 0 & 0 & 1\\ \nabla_2 & 0 & -1 & -1 & 0\end{bmatrix} NU^e \qquad (50) \\ &= B_0 U^e\end{aligned}$$

Similarly the couple-strain components are given by:

$$\begin{aligned}\begin{Bmatrix}\phi_{xx}\\ \phi_{xy}\\ \phi_{yx}\\ \phi_{yy}\end{Bmatrix} &= \begin{bmatrix}0 & 0 & 0 & \nabla_1 & 0\\ 0 & 0 & 0 & \nabla_2 & 0\\ 0 & 0 & 0 & 0 & \nabla_1\\ 0 & 0 & 0 & 0 & \nabla_2\end{bmatrix} NU^e \qquad (51) \\ &= B_1 U^e\end{aligned}$$

A total of ten strain-displacement relations are considered for the two dimensional micropolar bending. The B_0 and B_1 matrices derived above can be substituted into Equation(35) to obtain element stiffness matrix k^e. The D_0 and D_1 matrices relate strains to generalized stresses. For the two-dimensional elements, these are obtained by integrating the moments over the z direction. The material constants used in this study are shown in Table(3). To carry out the volume integral of Equation(35), numerical integration on Gaussian Quadrature is used in the program. The sampling points and the weighting factors for the interval -1 to +1 are given in Table 1.

5 Numerical Examples

5.1 Introductory Comments

The finite elements discussed in the previous chapter are tested for accuracy with various benchmark problems in bending. Four problems are considered for numerical accuracy. They are: (1) Rectangular plate in pure cylindrical bending, (2) Rectangular plate built-in at one end and subjected to a concentrated load at the middle of the free end, (3) Rectangular plate with built-in edges and subjected to uniformly distributed load, (4) Circular plate built-in at the ends and subjected to uniformly distributed load. These are shown in the following pages.

A preliminary study of the elements indicates that efficient results can be obtained with the two-dimensional elements. But to obtain variation of couple stresses along the thickness of the plate, the three-dimensional element is required. The four-node and eight-node elements are used first to obtain numerical results for the four problems. The eight-node three-dimensional element is used next to study the couple stress variation along the thickness of the plate. First, a patch test is performed for the two-dimensional elements as discussed in the next section.

A post-processor with graphic output is developed in this dissertation. The post-processor uses features of HOOPS stationed in a SUN 3/60 workstation. The FORTRAN finite element program outputs the necessary informaton for plotting. This information includes element connectivities, nodal coordinates and constrained nodes, constraint conditions and the deflections of each of the nodes. This data file is input to HOOPS which plots in color, the original element mesh and the deflected configuration.

5.2 Patch Test

A patch test is required for studying the convergence of an element. The patch test requires that atleast one element be completely surrounded by other elements. Then, as the size of that element is gradually reduced, the strains in the element should converge to a constant value. Research has shown that even though some elements are not conformable, they still pass the patch test and as such can be effectively used for implementation.

The patch test is applied to the two-dimensional elements. A rectangular plate is taken with five elements. The plate is loaded by simple moments applied in one direction. The size of the inner element is reduced gradually. The strains and stresses in the inner element converged to a constant value for the four-node and eight-node elements as the size of the inner element is reduced. This indicates that these elements can be used in general for plate bending problems. It can be seen that for two-dimensional elements, it is not possible to obtain complete displacement fields by applying pure bending moments since the ψ terms don't appear in the strain-displacement relations. Hence, the problem of a cantilever

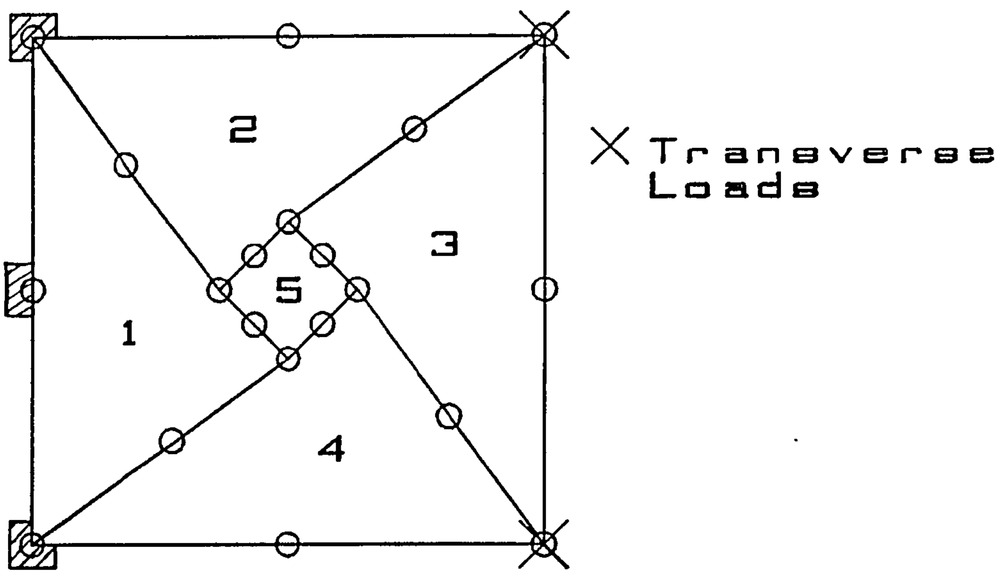


Figure 8: Patch Test for Convergence.

plate with concentrated load is considered for patch test.

5.3 Numerical Examples

5.3.1 Problem 1: Square Plate in Pure Cylindrical Bending.

A square plate is considered first for the pure bending case. The plate has dimensions of 100 inches by 100 inches for the two dimensional elements. A moment of 1 *lb./inch* is used along the x-direction. First, four elements were used and then sixteen elements were used. The pure bending problem has been solved analytically by Gauthier[14] and acts as a reference point for the present study. To test the validity of the present finite element formulation, the value of N and l are taken as 0.0 for the classical case. The boundary conditions can not be adequately defined for the three-dimensional element because the loads are to be applied in the xy plane and the boundary conditions require that these displacements be constrained for simply supported ends. The results from both the elements are shown in Table (2). Application of bending moments is facilitated by modifying the $B0$ matrix to include coupling between the strains and applied moments.

5.3.2 Problem 2: Rectangular Plate with One Built-in Edge

The problem of a rectangular plate built-in at one edge, shown in Figure(9), is considered next. First, a square plate with a concentrated load of 100 *pounds* acting at the middle of the free edge, is used with the eight-node two-dimensional

	Numerical solution	Exact solution[Eqn.16]
Maximum Displacement	2.810 X 10^{-4}	2.481 X 10^{-4}
Stress, t_{xx}	4.9153 psi	3.2723 psi
Stress, t_{yy}	0.9817 psi	1.0700 psi
Strain, ε_{xx}	1.9852 X 10^{-7}	1.8337 X 10^{-7}
Strain, ε_{yy}	8.5080 X 10^{-8}	7.3900 X 10^{-8}

Table 2: Numerical Results from Problem 1 Compared with Analytical Solutions.

element. The convergence for this element and the three-dimensional element is shown in Table(4) and Figure(10). To test the validity of the present finite element scheme, the finite element formulation is reduced to that of Classical Elasticity Theory by making $N = 0.0$ and $l = 0.0$ *inches*. When l is made equal to zero exactly, the finite element solutions were unstable. For this reason, l was reduced gradually from 0.1 *inches* to a low value of 0.001 *inches*. It is observed that the displacements and the moments do converge as the value of l is reduced. Further, there was no noticeable difference between the results when $l = 0.01$ *inches* and when $l = 0.001$ *inches*, indicating convergence. Hence, this value of $l = 0.001$ *inches* is used in the analysis and the results are compared with the Classical Elasticity Theory solutions given by Timoshenko et al[13]. The material constants used for comparison with the Classical Elasticity Theory are based on $N = 0.0$ and $l = 0.001$ *inches* and these values are shown in Table(3).

The maximum displacement is plotted for different values of N and is shown in figure(11). In these figures, the displacements are plotted against distance from the point of application of load along the free edge, calculated analytically. These values are also plotted for a rectangular plate with an aspect ratio of 1.34 and the results are shown in Figure(16). To look in to the effect of the characteristic length on the bending of micropolar plates, a non-zero value for l is considered. Initially, when $l = 0.001$ *inches* , numerical results for variation of N are obtained

N	$\lambda(psi)$	$\mu(psi)$	$\kappa(psi)$
0.00	7.47000 X 10^6	8.72000 X 10^6	0.0
0.25	1.22625 X 10^7	7.63000 X 10^6	1.09000 X 10^6
0.50	9.81000 X 10^6	4.36000 X 10^6	4.36000 X 10^6
0.75	5.72250 X 10^6	-1.0900 X 10^6	9.81000 X 10^6
0.90	2.48550 X 10^6	-5.4060 X 10^6	1.41264 X 10^7

N	$\alpha(pounds)$	$\beta(pounds)$	$\gamma(pounds)$
0.00	0.0	3.49000 X 10^1	3.49000 X 10^1
0.25	0.0	3.27000 X 10^1	3.27000 X 10^1
0.50	0.0	2.61600 X 10^1	2.61600 X 10^1
0.75	0.0	1.52600 X 10^2	1.52600 X 10^2
0.90	0.0	6.62800 X 10^1	6.62800 X 10^1

Table 3: Material Properties Used for Micropolar Plate Bending.

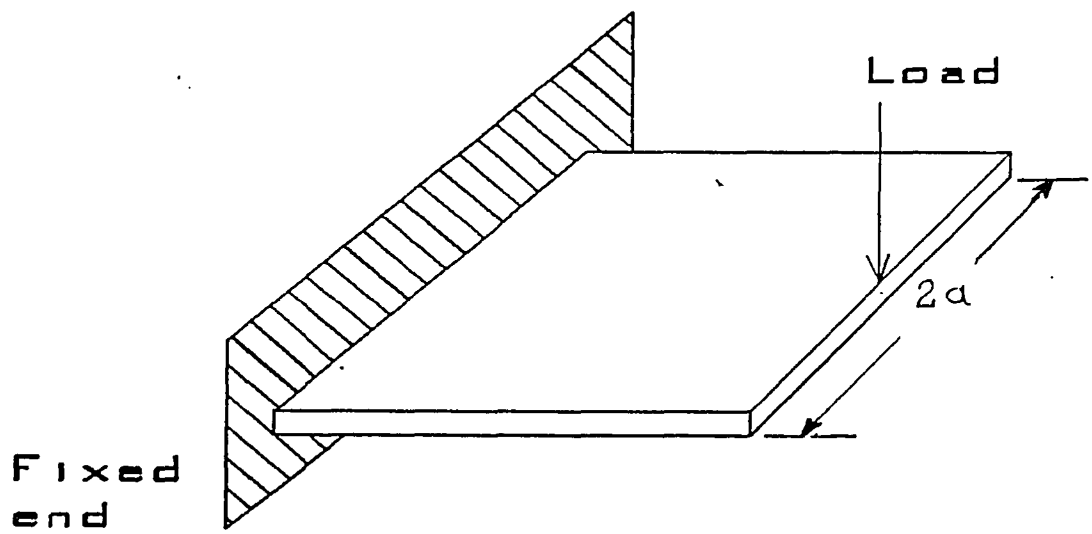


Figure 9: Rectangular Plate with One Built-in Edge with a Concentrated Load Applied at the Middle of the Free Edge.

	$\frac{x}{a}$	Displacement Numerical solution(in.)	Displacement Exact solution(in.)
3D 4 elements	0.0	0.05981050	0.061152
3D 16 elements	0.0	0.06116434	0.061152
	0.5	0.05763541	0.054600
	1.0	0.05190274	0.044044
2D 8-node 4 elements	0.0	0.0587341	0.061152
2D 8-node 16 elements	0.0	0.0610539	0.061152
	0.5	0.0335284	0.054600
	1.0	0.0277030	0.044044
2D 4-node 4 elements	0.0	0.049801334	
2D 4-node 16elements	0.0	0.053721397	0.061152
	0.5	0.029812430	0.054600
	1.0	0.022836324	0.044040

Table 4: Numerical Results for Plate with One Edge Built-in.

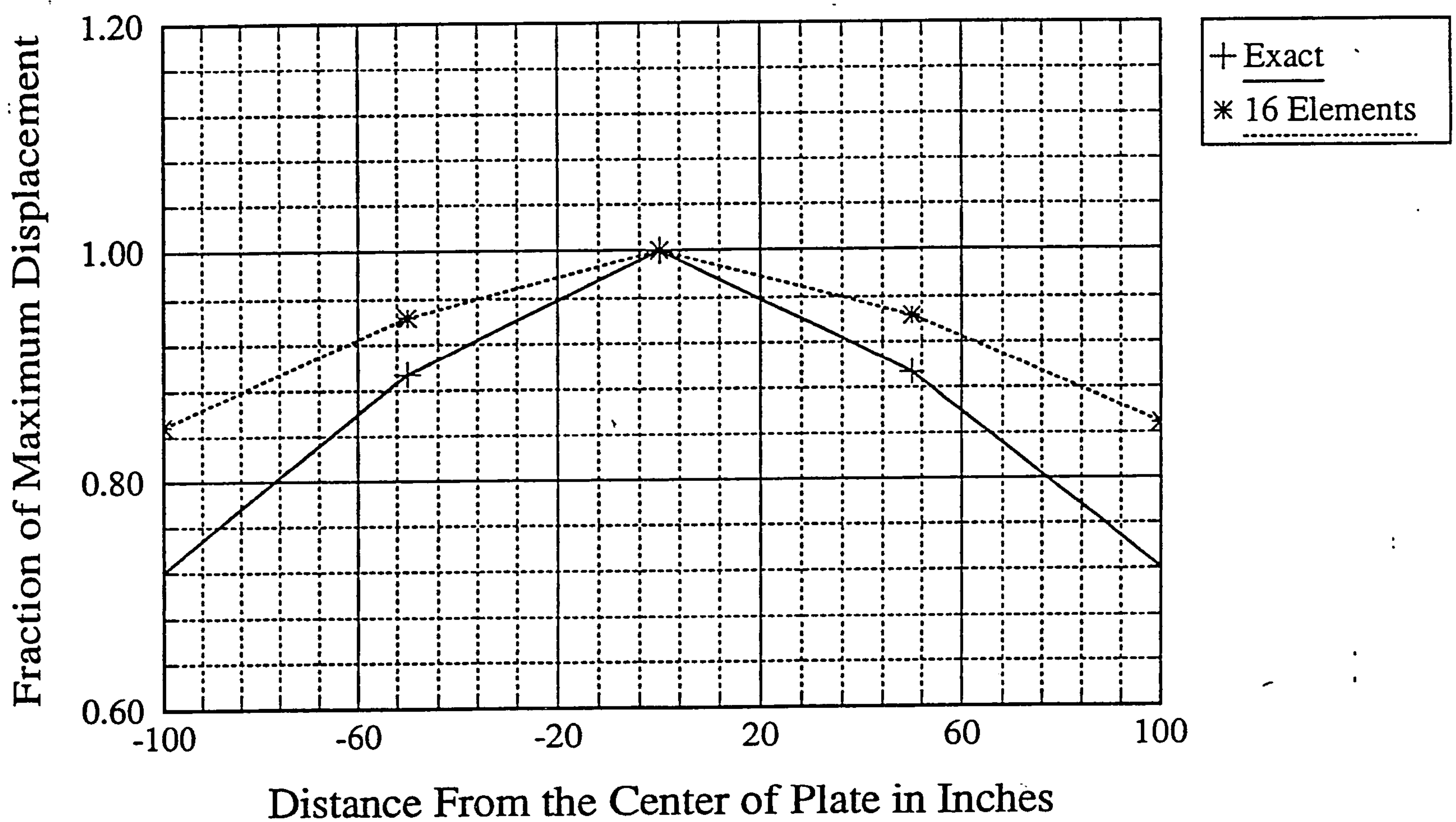


Figure 10: Displacements VS Distance From the point of Application of Loads for Problem 2.

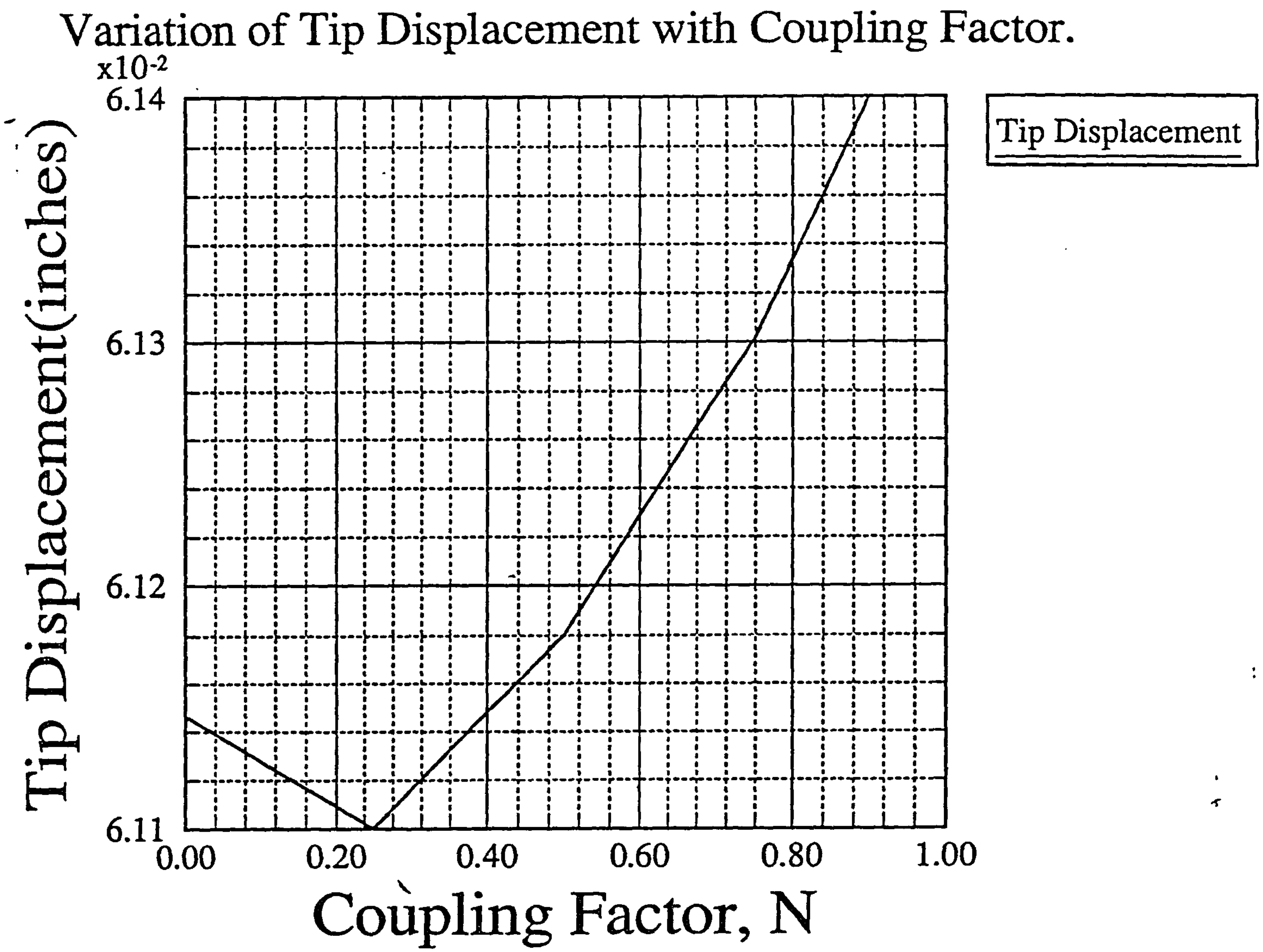


Figure 11: Effect of Variation of N on tip displacement.

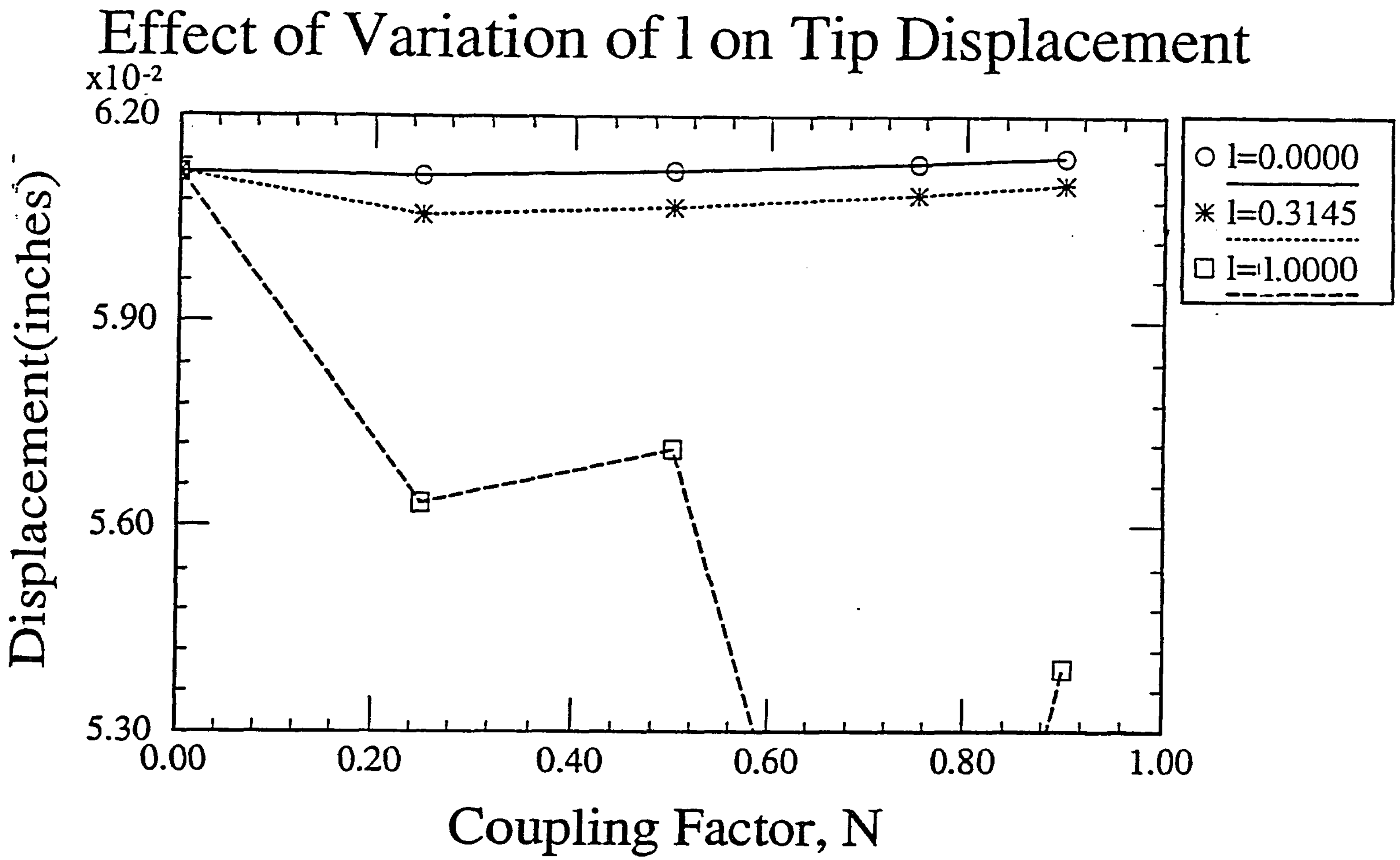


Figure 12: Effect of Variation of l on tip displacement.

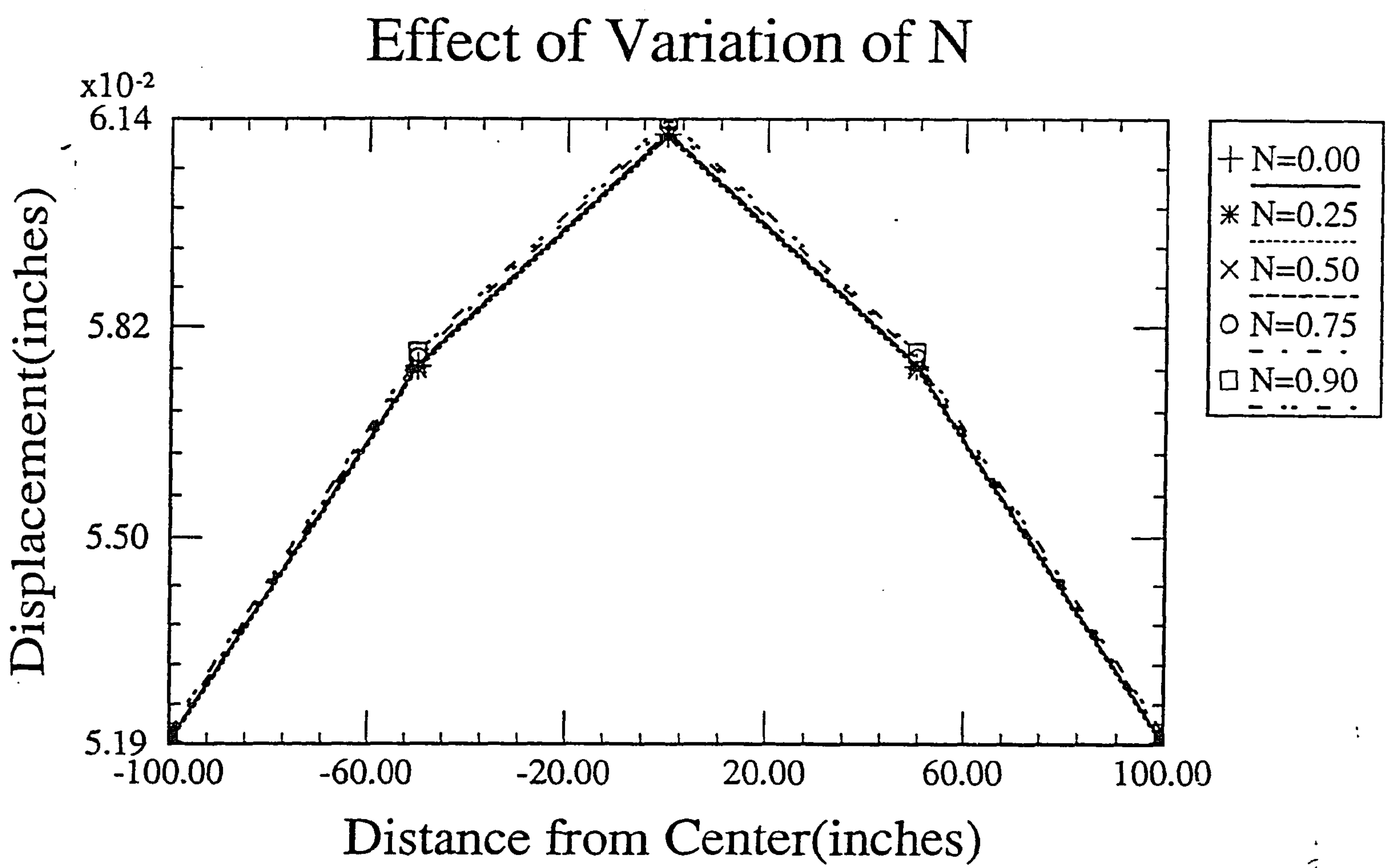


Figure 13: Effect of Variation of N when l is 0.001 *inches*

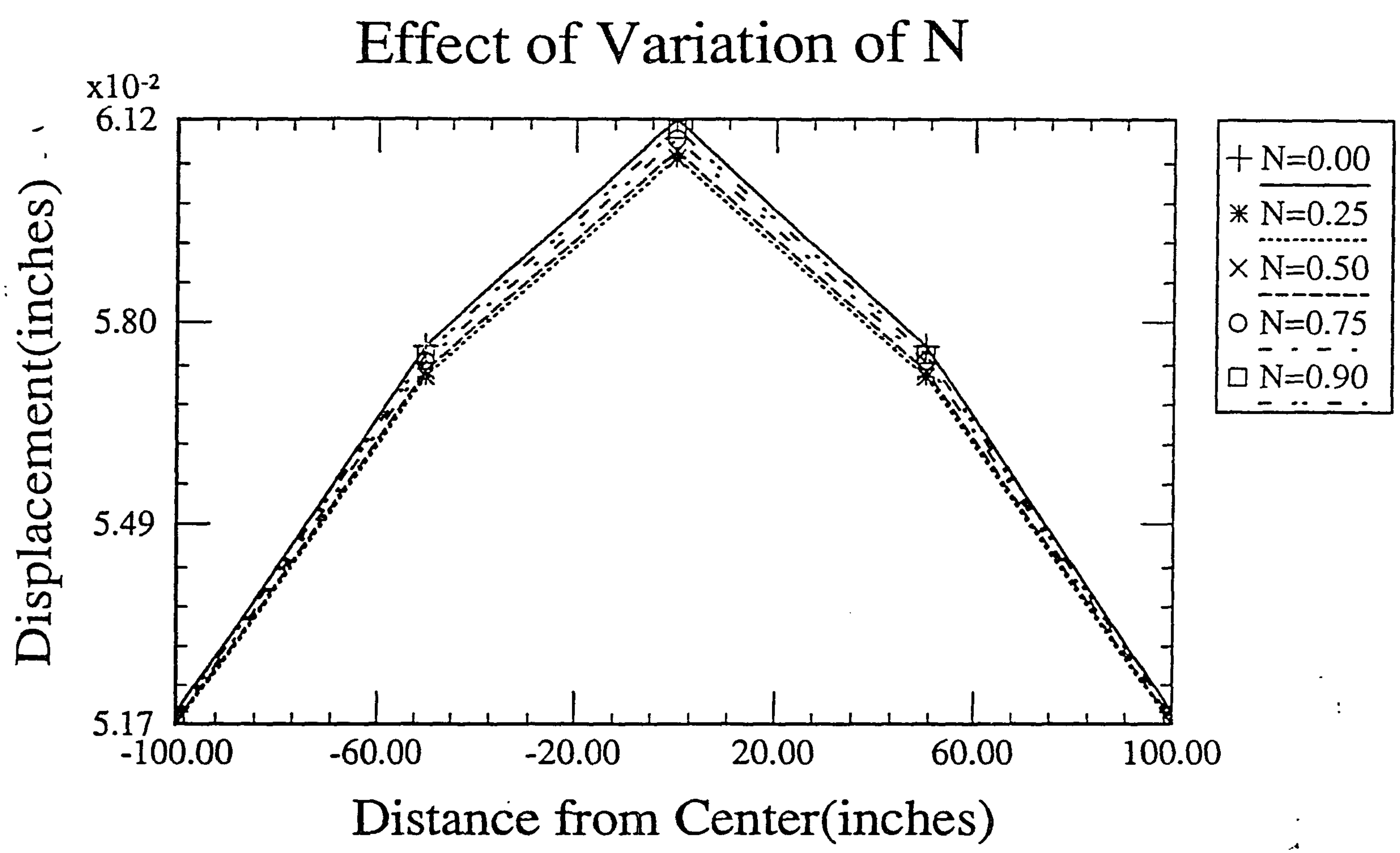


Figure 14: Effect of Variation of N when l is 0.3145 *inches*

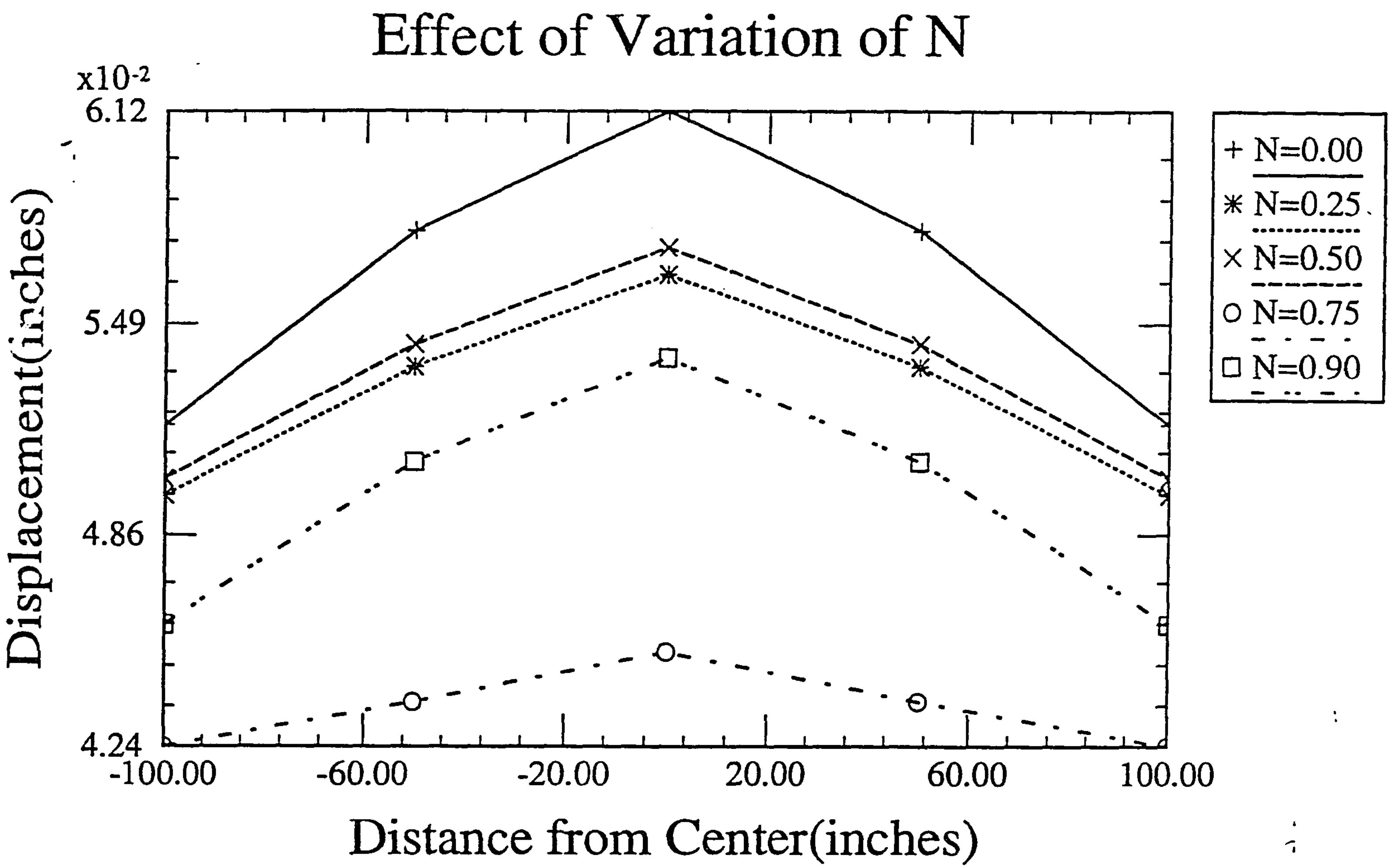


Figure 15: Effect of Variation of N when l is 1.0 *inch*

and are plotted in Figure(13). With the value of l set at 0.3145 *inches* , several runs were made for various values of the coupling factor, N . These results are shown in Figure 14. This is repeated with l set to 1.00 *inches* and the results are shown in Figure(15). It can be seen from these values that for a general micropolar elastic body, the value of characteristic length affects the displacement field. As l is increased, there is a general tendency for the displacements to reduce compared to the displacements when l is set to zero or near zero. Moreover, if l value is small, in the order of 0.1 *inch* or less, the displacements are unaffected. The results are plotted in Figure(12).

Another set of program runs was obtained in order to study the effect of variation of the ratio $\frac{\beta}{\gamma}$ on the displacements. It is observed that the best suitable value for $\frac{\beta}{\gamma}$ is 1.0. Most previous studies have only considered $\frac{\beta}{\gamma}$ to have a value of 0.0. These results indicate that β affects bending of micropolar plates. Similar to the effect of l , β affects the displacements more if the value of l is taken larger. Post-processor outputs are given in the Appendix.

5.3.3 Problem 3: Square Plate with Built-in Edges.

The deflections for a built-in plate are obtained for uniformly distributed load of 0.0009 $lb./inch^2$ and are shown in Table(5). The exact solution given in the table is from the Classical Elasticity Theory solutions. The square plate is considered

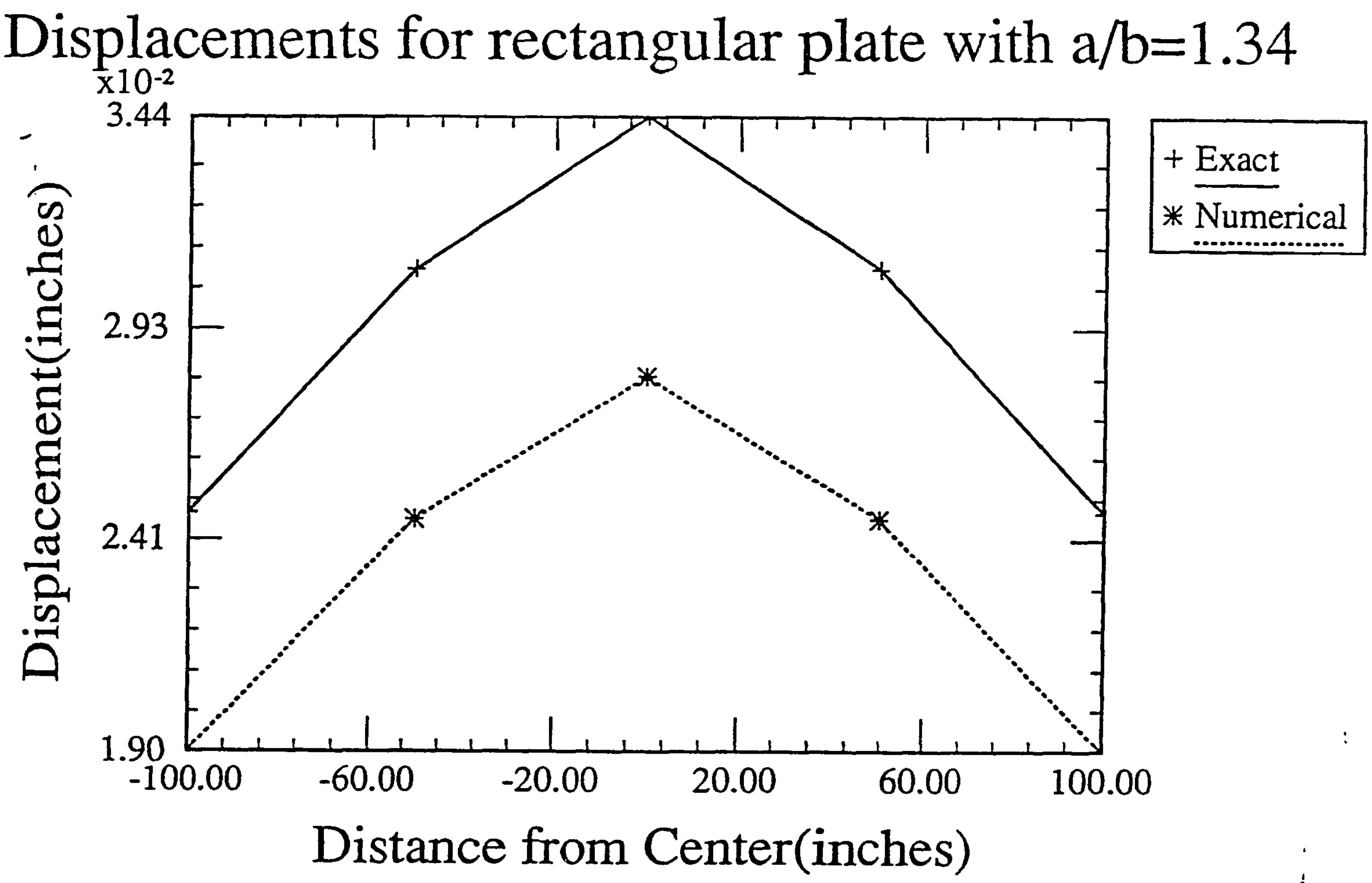


Figure 16: Displacemens VS distance from the point of application of load for a rectangular plate with aspect ratio of 1.34: Probem2

to have the dimensions of 100 in. X 100 in. X 1 in. Four and sixteen elements were used for the study. While it is difficult to approximate uniformly distributed load when the number of elements is small, the results indicate that the eight-node element can be satisfactorily used even with a small number of elements. In all the cases strains and stresses were different from the classical theory, due to the fact that there are additional couples and microrotation strains. These values must also be considered in conjunction with the stress and displacements. The output from the post-processor is shown in the Appendix.

5.3.4 Problem 4: Circular Plate with Built-in Edges.

A circular plate with built-in ends is considered next with a uniform load of 0.0009 $lb/inch^2$. The results from the two dimensional element are shown in Table(6). The exact analytical solution for the maximum displacement from the Classical Elasticity Theory is given by $\frac{q}{64D}(a^2 - r^2)^2$ where a is the radius of the plate and r is measured from the center of the plate. It can be seen from the results that the solutions converge to the exact value.

	$\frac{x}{a}$	Displacement Numerical solution(in.)	Displacement Exact solution(in.)
3D 8-node element	0.0 0.5	5.08002×10^{-5} 2.90143×10^{-5}	4.12776×10^{-5}
2D 8-node element	0.0 0.5	4.24167×10^{-5} 2.59065×10^{-5}	4.12776×10^{-5}
2D 4-node element	0.0 0.5	4.30923×10^{-5} 2.62479×10^{-5}	4.12776×10^{-5}

Table 5: Numerical Results for Square Plate with Built-in Edges.

2D 8-Node Element	radius r	Displacement Numerical solution(in.)	Displacement Exact solution(in.) $= \frac{q}{64D}(a^2 - r^2)^2$
3 Elements	a/2 0.0	0.2374 0.4108	0.3199 0.5687
12 Elements	a/2 0.0	0.2956 0.5593	0.3199 0.5687

Table 6: Numerical results for Circular Clampled Plate.

6 Conclusions

6.1 Concluding Remarks

In this study, isoparametric finite elements for linear isotropic micropolar plate bending analysis were developed. Four-node and eight-node elements were used for the two-dimensional case. Eight-node three-dimensional element was used for the three-dimensional case. Two-dimensional finite element formulation was developed for the case of bending of rectangular plates. A general three-dimensional formulation is used for the plate bending analysis. FORTRAN programs were developed for the above for linear isotropic micropolar materials. A post-processor with graphical output of the displacements was also developed.

The validity of the finite element programs for bending of micropolar plates was established by comparing the numerical results with analytical solutions for the case of pure cylindrical bending of rectangular plates. For other problems, the finite element solutions were tested by reducing the formulation to the classical case and then comparing with available analytical solutions. The newly developed elements were tested for convergence. For the first time, to the best of the author's knowledge, the plots for the influence of N and l on the bending behavior of micropolar plates were obtained. The displacements were not affectted by the variation of N when the characterestic length l was taken very small. However, couple stress effects are noticeable if a different value of l is taken. For the bending of plates, the best value for the ratio of β/γ is found to be 1 based on the current study. It is also found that when N is made equal to zero, the couple stresses and the microrotations vanished-consistant with the theory.

The two dimensional study can be extended to provide C^1 compatibility in the elements. The results from the four-node and the eight-node elements indicates that other higher polynomial interpolations may have to be considered. Numerical results indicate that a considerable CPU time savings can be realized by using the two-dimensional elements. The three-dimensional element can be reduced by integrating in the x and y directions only and explicitly integrating in z direction. For two-dimensional elements, the D_0 matrix can be obtained from the D_0 matrix of the Classical Elasticity Theory solutions. The present study indicates that the finite element solutions are unstable for vanishing D_1 matrix. Hence, based on a value of 0.001 *inches* for l , the elements in D_1 matrix become non-zero and have very small magnitudes.

The four and eight-node two-dimensional elements are used to solve for displacements and generalized stresses for various boundary conditions of the plate. These problems have also been solved by the three-dimensional element. While the two-dimensional elements have proved to be adequate in terms of accuracy of results, a more detailed three-dimensional element could provide some valuable insight into distribution of various stresses and strains. The study of two-dimensioinal elements indicates that, for application of pure moments on the plates, the strain-displacement relations have to be modified to include coupling between the applied moments and the strains. Otherwise, only the displacements corresponding to the loads will be present in the results. The three-dimensional element has no such drawbacks since all the three components of displacements, u, v, w and ϕ_x, ϕ_y, ϕ_z are considered. The plots from the post processor in which the displacements are scaled up, agree with the general expected displacement field.

6.2 Some Future Research Interests

One of the most important works left to be done is the evaluation of material constants κ, α, β and γ. Currently several methods are employed to estimate these constants. The finite element formulation can be extended with addition of new higher order elements as well as higher order polynomials for shape functions. Study of variation of couple stresses along the thickness of the material could be of great interest. Finite element formulations for a general anisotropic micropolar elasticity is an area to be considered for future research.

Appendix

Equilibrium

$$\frac{d}{dx}\begin{bmatrix} t_{xx} \\ t_{xy} \\ t_{xz} \end{bmatrix} + \frac{d}{dy}\begin{bmatrix} t_{yx} \\ t_{yy} \\ t_{yz} \end{bmatrix} + \frac{d}{dz}\begin{bmatrix} t_{zx} \\ t_{zy} \\ t_{zz} \end{bmatrix} = 0$$

$$\frac{d}{dx}\begin{bmatrix} m_{xx} \\ m_{xy} \\ m_{xz} \end{bmatrix} + \frac{d}{dy}\begin{bmatrix} m_{yx} \\ m_{yy} \\ m_{yz} \end{bmatrix} + \frac{d}{dz}\begin{bmatrix} m_{zx} \\ m_{zy} \\ m_{zz} \end{bmatrix} + \begin{bmatrix} t_{yz} - t_{zy} \\ t_{zx} - t_{xz} \\ t_{xy} - t_{yx} \end{bmatrix} = 0$$

Constitutive

$$\begin{bmatrix} t_{xx} \\ t_{yy} \\ t_{zz} \end{bmatrix} = \lambda(\varepsilon_{xx} + \varepsilon_{yy} + \varepsilon_{zz}) + (2\mu + \kappa)\begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \end{bmatrix}$$

$$\begin{bmatrix} t_{xy} \\ t_{xz} \\ t_{yx} \\ t_{yz} \\ t_{zx} \\ t_{zy} \end{bmatrix} = (\mu + \kappa)\begin{bmatrix} \varepsilon_{xy} \\ \varepsilon_{xz} \\ \varepsilon_{yx} \\ \varepsilon_{yz} \\ \varepsilon_{zx} \\ \varepsilon_{zy} \end{bmatrix} + \mu\begin{bmatrix} \varepsilon_{yz} \\ \varepsilon_{zx} \\ \varepsilon_{xy} \\ \varepsilon_{zy} \\ \varepsilon_{xz} \\ \varepsilon_{yz} \end{bmatrix}$$

$$\begin{bmatrix} m_{xx} \\ m_{yy} \\ m_{zz} \end{bmatrix} = \alpha\left[\frac{\partial\phi_x}{\partial x} + \frac{\partial\phi_y}{\partial y} + \frac{\partial\phi_z}{\partial z}\right] + (\beta + \gamma)\begin{bmatrix} \frac{\partial\phi_x}{\partial x} \\ \frac{\partial\phi_y}{\partial y} \\ \frac{\partial\phi_z}{\partial z} \end{bmatrix}$$

$$\begin{bmatrix} m_{xy} \\ m_{xz} \\ m_{yx} \\ m_{yz} \\ m_{zx} \\ m_{zy} \end{bmatrix} = \beta\begin{bmatrix} \frac{\partial\phi_x}{\partial y} \\ \frac{\partial\phi_x}{\partial z} \\ \frac{\partial\phi_y}{\partial x} \\ \frac{\partial\phi_y}{\partial z} \\ \frac{\partial\phi_z}{\partial x} \\ \frac{\partial\phi_z}{\partial y} \end{bmatrix} + \gamma\begin{bmatrix} \frac{\partial\phi_y}{\partial x} \\ \frac{\partial\phi_z}{\partial x} \\ \frac{\partial\phi_x}{\partial y} \\ \frac{\partial\phi_z}{\partial y} \\ \frac{\partial\phi_x}{\partial z} \\ \frac{\partial\phi_y}{\partial z} \end{bmatrix}$$

Strain-Displacement

$$\begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \end{bmatrix} = \begin{bmatrix} \frac{\partial u_x}{\partial x} \\ \frac{\partial u_y}{\partial y} \\ \frac{\partial u_z}{\partial z} \end{bmatrix}$$

$$\begin{bmatrix} \varepsilon_{xy} \\ \varepsilon_{xz} \\ \varepsilon_{yx} \\ \varepsilon_{yz} \\ \varepsilon_{zx} \\ \varepsilon_{zy} \end{bmatrix} = \begin{bmatrix} \frac{\partial u_y}{\partial x} \\ \frac{\partial u_z}{\partial x} \\ \frac{\partial u_x}{\partial y} \\ \frac{\partial u_z}{\partial y} \\ \frac{\partial u_x}{\partial z} \\ \frac{\partial u_y}{\partial z} \end{bmatrix} + \begin{bmatrix} -\phi_z \\ \phi_y \\ \phi_z \\ -\phi_x \\ -\phi_y \\ \phi_x \end{bmatrix}$$

Stress Boundary Conditions

On the boundary surface s of the body,

$$\begin{bmatrix} t_{(n)x} \\ t_{(n)y} \\ t_{(n)z} \end{bmatrix} = n_x \begin{bmatrix} t_{xx} \\ t_{xy} \\ t_{xz} \end{bmatrix} + n_y \begin{bmatrix} t_{yx} \\ t_{yy} \\ t_{yz} \end{bmatrix} + n_z \begin{bmatrix} t_{zx} \\ t_{zy} \\ t_{zz} \end{bmatrix}$$

$$\begin{bmatrix} m_{(n)x} \\ m_{(n)y} \\ m_{(n)z} \end{bmatrix} = n_x \begin{bmatrix} m_{xx} \\ m_{xy} \\ m_{xz} \end{bmatrix} + n_y \begin{bmatrix} m_{yx} \\ m_{yy} \\ m_{yz} \end{bmatrix} + n_z \begin{bmatrix} m_{zx} \\ m_{zy} \\ m_{zz} \end{bmatrix}$$

Compatibility Conditions

$$\frac{\partial \varepsilon_{xx}}{\partial y} - \frac{\partial \varepsilon_{yx}}{\partial x} + \frac{\partial \phi_z}{\partial x} = 0$$

$$\frac{\partial \varepsilon_{xy}}{\partial y} - \frac{\partial \varepsilon_{yy}}{\partial x} + \frac{\partial \phi_z}{\partial y} = 0$$

$$\frac{\partial \varepsilon_{xz}}{\partial y} - \frac{\partial \varepsilon_{yz}}{\partial x} - \frac{\partial \phi_y}{\partial y} - \frac{\partial \phi_x}{\partial x} = 0$$

$$\frac{\partial \varepsilon_{xx}}{\partial z} - \frac{\partial \varepsilon_{zx}}{\partial x} - \frac{\partial \phi_y}{\partial x} = 0$$

$$
\begin{aligned}
\frac{\partial \varepsilon_{xy}}{\partial z} - \frac{\partial \varepsilon_{zy}}{\partial x} + \frac{\partial \phi_z}{\partial z} + \frac{\partial \phi_x}{\partial x} &= 0 \\
\frac{\partial \varepsilon_{xz}}{\partial z} - \frac{\partial \varepsilon_{zz}}{\partial x} - \frac{\partial \phi_y}{\partial z} &= 0 \\
\frac{\partial \varepsilon_{yx}}{\partial z} - \frac{\partial \varepsilon_{zx}}{\partial y} - \frac{\partial \phi_z}{\partial z} - \frac{\partial \phi_y}{\partial y} &= 0 \\
\frac{\partial \varepsilon_{yy}}{\partial z} - \frac{\partial \varepsilon_{zy}}{\partial y} + \frac{\partial \phi_x}{\partial y} &= 0 \\
\frac{\partial \varepsilon_{yz}}{\partial z} - \frac{\partial \varepsilon_{zz}}{\partial y} + \frac{\partial \phi_x}{\partial z} &= 0
\end{aligned}
$$

```
C ***************************************************
C *                                                 *
C *          2-D MICROPOLAR FINITE ELEMENT METHOD   *
C *                 FOR PLATE BENDING               *
C *                  SKYLINE=MICRO:2D               *
C *                                                 *
C *                   Developed by                  *
C *                                                 *
C *               Vallanore K. Suresh               *
C *                    May 1989                     *
C *                                                 *
C ***************************************************
C                    MAIN
C                     I
C                     I---STSTIFF---ELSTIF(EK,NE)
C                     I
C                     I---LOADER
C                     I
C                     I---COLSOL
C                     I
C                     I---STRESS
C
C                    ** EXPLANATION OF THE SYMBOLS **
C
C                                 SCALAR
C     AIJ        ....  MATERIAL PROPERTIES FOR FORCE -STRESS
C     BIJ        ....  MATERIAL PROPERTIES FOR COUPLE-STRESS
C     NB         ....  NUMBER OF BAND-WIDTH OF SK(ND,ND)
C     NC         ....  NUMBER OF TOTAL CONSTRAINED NODE
C     ND         ....  NUMBER OF TOTAL DEGREE OF FREEDOM
C     NE         ....  NUMBER OF TOTAL ELEMENT
C     NN         ....  NUMBER OF TOTAL NODAL POINTS
C     TH         ....  THICKNESS OF THE ELEMENT
C
C                                 VECTOR
C     SDISP  (ND)....  STRUCTURAL NODAL DISPLACEMENTS & ROTATIONS
C     EDISP  (40)....  ELEMENT    NODAL DISPLACEMENTS & ROTATIONS
C     KSTRN  (NC)....  CONSTRAINED NODE
C     KSTRT  (NC)....  TYPE OF CONSTRAIN
C                         10000  W-DISPLACEMENT  IS FIXED
C                         01000  X-SLOPE IS FIXED
C                         00100  Y-SLOPE IS FIXED
C                         00010  X-MICROROTATION IS FIXED
C                         00001  Y-MICROROTAION IS FIXED
C     INXY   (NE)....  LOCATION OF INTEREST IN ELEMENT FOR OUTPUT
C                         000  AT CENTROID OF RECTANGULAR ELEMENT
C
C                                 MATRIX
C     E    (NE,6)....  DISPLACEMENT-STRAIN
C     PHIJ(NE,4)....   ROTATION    -STRAIN
C     T    (NE,6)....  FORCE       -STRESS
C     CM   (NE,4)....  COUPLE      -STRESS
C     EK   (40,40).... ELEMENT    STIFFNESS MATRIX
C     SK  (ND,ND)....  STRUCTURAL STIFFNESS MATRIX
C     NNP  (NE,8)....  NODE NUMBER FORMING THE ELEMENT
C
C                                 TENSOR
C     B0(NE,6,40)....  DISP-STRAIN DISPLACEMENT MATRIX B
C     B1(NE,4,40)....  ROTN-STRAIN ROTATION     MATRIX B
C****************************************************************
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/B01/B0(64,6,40),B1(64,4,40),ID(500),INXY(200)
      COMMON/NNNN/NNP(64,8)
      COMMON/XYZ/X(100),Y(100)
      COMMON/CSTRN/KSTRN(500),KSTRT(500)
      COMMON/MAT/D0(6,6),D1(4,4),A1111,A1122,A2222,A1212,A1221,A2121,
```

```
     *                              B1111,B1122,B2222,B1212,B1221,B2121
      COMMON/AIN/NB,NC,ND,NE,NN,TH,NINT,DET,R,S,NELEM
      COMMON/FSK/A(20000),V(500),EK(40,40),MAXA(501)
C DIMENSION OF MAXA() MUST BE ONE GREATER THAN V().
      CALL ASSIGN(5,'p2e16.DAT')
      CALL ASSIGN(6,'p2e16.OUT')
C     SAVE
C
C I-------------------- INPUT & OUTPUT FILE OPEN ---------------------I
C $INSERT SYSCOM>ERRD.F
C $INSERT SYSCOM>KEYS.F
C $INSERT SYSCOM>A$KEYS
C     CALL SRCH$$(K$DELE,'FEMO',INTS(4),INTS(2),TYPE,CODE)
C     CALL OPEN$A(A$WRIT+A$SAMF,'FEMO',INTS(4),INTS(2))
C     CALL SRCH$$(K$READ,'FEMI',INTS(4),INTS(1),TYPE,CODE)
C     CALL SRCH$$(K$WRIT,'FEMO',INTS(4),INTS(2),TYPE,CODE)
C     CT1=CTIM$A(ITIM)
C
C I---------------------------- DATA INPUT ---------------------------I
C NE? READ TOTAL NUMBER OF ELEMENTS
      READ(5,100) NE
 100  FORMAT(I4)
C NP? READ ELEMENT CONNECTION INFORMATION
      DO 10 N=1,NE
      READ(5,200) (NNP(N,I),I=1,8)
 200  FORMAT(4X,8I4,I6)
   10 CONTINUE
C NN? READ TOTAL NUMBER OF NODAL POINTS
      READ(5,100) NN
C XY? READ COORDINATE (X,Y) OF NODAL POINTS
      DO 20 N=1,NN
      READ(5,300)X(N),Y(N)
 300  FORMAT(4X,3F20.10)
 20   CONTINUE
C NC? READ NUMBER OF CONSTRAINED NODES
      READ(5,100) NC
C KS? READ CONSTRAINTS
      DO 30 N=1,NC
      READ(5,400) KSTRN(N),KSTRT(N)
 400  FORMAT(4X,I4,I8)
   30 CONTINUE
C TH? READ THICKNESS OF THE MATERIAL
      TH = 1.0D0
C
C     READ GAUSS-NUMERICAL INTEGRATION ORDER
C
      READ(5,500)NINT
 500  FORMAT(I5)
C
C AB? READ MATERIAL PROPERTIES
      READ(5,600) A1111,A1122,A2222,A1212,A1221,A2121
      READ(5,600) B1111,B1122,B2222,B1212,B1221,B2121
 600  FORMAT(6E14.7)
C
C I--------------------- TOTAL DEGREE OF FREEDOM ---------------------I
C EACH NODE HAS 5-DOF OF (W,THX,THY,MRX,MRY)
      ND=NN*5
C CALCULATION OF BAND-WIDTH OF STRUCTURAL STIFFNESS MATRIX SK(ND,ND)
      NB=0
      DO 40 IB=1,NE
         IMAX=MAX0(NNP(IB,1),NNP(IB,2),NNP(IB,3),NNP(IB,4),NNP(IB,5),
     *        NNP(IB,6),NNP(IB,7),NNP(IB,8))
         IMIN=MIN0(NNP(IB,1),NNP(IB,2),NNP(IB,3),NNP(IB,4),NNP(IB,5),
     *        NNP(IB,6),NNP(IB,7),NNP(IB,8))
         NBCHEK=(IMAX-IMIN+1)*5
         IF(NBCHEK.GT.NB) NB=NBCHEK
```

```
          IF(NB .GT. 300) GO TO 99
   40 CONTINUE
C
      SCALE=1.0D0
      DO 41 I=1,NN
         x(i)=x(i)*SCALE
         Y(I)=Y(I)*SCALE
   41 CONTINUE
C
C I--------------------------- DATA OUTPUT ----------------------------I
C PRINT THE HEAD OF OUTPUT
      WRITE(6,700)
  700 FORMAT(1H1)
      WRITE(6,800)
  800 FORMAT(3X,'******************************************************
     ***************')
      WRITE(6,900)
  900 FORMAT(3X,'*',68X,'*')
      WRITE(6,1000)
 1000 FORMAT(3X,'*',13X,'2-D MICROPOLAR PLATE BENDING ANALYSIS',
     *       18X,'*')
      WRITE(6,1050)
 1050 FORMAT(3X,'*',28X,' SKYLINEMICRO',27X,'*')
      WRITE(6,900)
      WRITE(6,800)
C
C WRITE THE INFORMATIONS
      WRITE(6,1100)
 1100 FORMAT(/20X,'**** DISCRETIZATION NUMBER ****')
      WRITE(6,1200)
 1200 FORMAT(/13X,'ELEMT.#',3X,'NODES.#',3X,'CONSTR.#',3X,'THIKNES',
     *        3X,'BAND-WIDTH',3X,'GAUSS NUMERICAL INTEGRATION ORDER')
      WRITE(6,1300) NE,NN,NC,TH,NB,nint
 1300 FORMAT(13X,I4,7X,I3,7X,I3,5X,E10.3,5X,I5,19X,I2)
C
C WRITE ORTHOTROPIC MATERIAL PROPERTIES
      WRITE(6,1400)
 1400 FORMAT(//22X,'**** MATERIAL PROPERTIES ****',
     *      //6X,'A1111',7X,'A1122',7X,'A2222',7X,'A1212',7X,
     *      'A1221',7X,'A2121')
      WRITE(6,1500) A1111,A1122,A2222,A1212,A1221,A2121
 1500 FORMAT(6E12.3)
      WRITE(6,1600)
 1600 FORMAT(/6X,'B1111',7X,'B1122',7X,'B2222',7X,'B1212',
     *        7X,'B1221',7X,'B2121')
      WRITE(6,1500) B1111,B1122,B2222,B1212,B1221,B2121
C
C WRITE ELEMENT CONNECTION INFORMATIONS
      WRITE(6,1700)
 1700 FORMAT(//21X,'**** ELEMENT-NODE CONNECTION ****')
      WRITE(6,1800)
 1800 FORMAT(/3X,'ELM NP1 NP2 NP3 NP4 NP5 NP6 NP7 NP8 IXY')
      DO 45 I=1,NE
         ID(I)=I
      WRITE(6,1900) ID(I),NNP(I,1),NNP(I,2),NNP(I,3),
     1NNP(I,4),NNP(I,5),NNP(I,6),NNP(I,7),NNP(I,8),INXY(I)
   45 CONTINUE
 1900 FORMAT(2X,I4,8(1X,I3))
C
C WRITE NODAL COORDINATES
   56 WRITE(6,2000)
 2000 FORMAT(//24X,'**** NODAL COORDINATE ****')
      WRITE(6,2100)
 2100 FORMAT(/1X,'NODE',5X,'X',8X,'Y',4X,'NODE',5X,'X',8X,'Y',
     * 4X,'NODE',5X,'X',8X,'Y')
      DO 55 I=1,NN
```

```
         ID(I)=I
   55 CONTINUE
      LINE=NN/3
      IRESID=NN-3*LINE
      DO 60 N=1,LINE
   60 WRITE(6,2200)  (ID(3*(N-1)+I),X(3*(N-1)+I),Y(3*(N-1)+I),I=1,3)
 2200 FORMAT(3(2X,I3,1X,F8.3,1X,F8.3))
      IF(IRESID .EQ. 0) GO TO 57
      WRITE(6,2200)  (ID(3*LINE+I),X(3*LINE+I),Y(3*LINE+I),I=1,IRESID)
C
C WRITE CONSTRAINTS
   57 WRITE(6,2300)
 2300 FORMAT(//27X,'**** CONSTRAINT ****')
      WRITE(6,2400)
 2400 FORMAT(/24X,'CNSTRND-NODE',2X,'CNSTRND-CODE')
      WRITE(6,2500)  (KSTRN(N),KSTRT(N),N=1,NC)
 2500 FORMAT(28X,I3,12X,I5)
C
C I---------------------------- MAIN PROGRAM ----------------------------I
                              CALL STSTIFF
                              CALL LOADER
                              CALL COLSOL
                              CALL STRESS
C
C     CT3=CTIM$A(ITIM)
C     C=CT3-CT1
C     WRITE(6,9999) C
C
   99 WRITE(6,999) NB,IB
  999 FORMAT('*****STOP NB=',I5,'  AT ELEMENT=',I5)
C9999 FORMAT(2X,'COMP. TIME T3=',F10.3)
C
C     CALL SRCH$$(K$CLOS,'FEMI',INTS(4),INTS(1),TYPE,CODE)
C     CALL SRCH$$(K$CLOS,'FEMO',INTS(4),INTS(2),TYPE,CODE)
C
      STOP
                                           END
C ************************************************************
      SUBROUTINE STSTIFF
C ************************************************************
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/CSTRN/KSTRN(500),KSTRT(500)
      COMMON/NNNN/NNP(64,8)
      COMMON/XYZ/X(100),Y(100)
      COMMON/AIN/NB,NC,ND,NE,NN,TH,NINT,DET,R,S,NELEM
      COMMON/FSK/A(20000),V(500),EK(40,40),MAXA(501)
      DIMENSION MHT(2000)
      NBND = NB * ND
      DO 10 I=1,NBND
         A(I) = 0.0D0
 10   CONTINUE
      DO 20 NELEM=1,NE
C
C     COLUMN DETERMINATION
C
      CALL ELSTIFF
      DO 20 INC=1,8
         INOC = NNP(NELEM,INC)
         IBC = (NNP(NELEM,INC)-1) * 5
      DO 20 IDC=1,5
         ICEL = (INC-1)*5 + IDC
         ICST = IBC + IDC
         IDI = 0
         IF (ICST.GT.NB) IDI = ICST - NB
         IVC = (ICST-1) * NB
C
```

```
C         ROW DETERMINATION
C
          DO 18 INR=1,8
             INOR = NNP(NELEM,INR)
             IF (INOC.LT.INOR) GO TO 18
             IBR = (NNP(NELEM,INR)-1) * 5
             IDVC = IDC
             IF (INOC.GT.INOR) IDVC = 5
             DO 15 IDR=1,IDVC
                IREL = (INR-1) * 5 + IDR
                IVV = IVC + IBR + IDR - IDI
                SS = A(IVV) + EK(IREL,ICEL)
                IF (DABS(SS).LT.1.0D-14) SS = 0.0D0
                A(IVV) = SS
 15          CONTINUE
 18       CONTINUE
 20    CONTINUE
C
C      ELIMINATE CONSTRAINT POINTS
C
C      KSTRN(N) ............ N TH CONSTRAINED NODE
C      KSTRT(N) ............ TYPE OF N TH CONSTRAINT
C
C                 10000 - W IS FIXED
C                 01000 - THETA X IS FIXED
C                 00100 - THETA Y IS FIXED
C                 00010 - MIC.ROT. X IS FIXED
C                 00001 - MIC.ROT. Y IS FIXED
C
C
C
       DO 210 N=1,NC
          ICW = KSTRN(N) * 5 - 4
          ICTHX = KSTRN(N) * 5 - 3
          ICTHY = KSTRN(N) * 5 - 2
          ICMRX = KSTRN(N) * 5 - 1
          ICMRY = KSTRN(N) * 5
C
          KCHK = KSTRT(N)
C
C         ELIMINATE DISPLACEMENT W
C
          IF (KCHK.LT.10000) GO TO 60
          ICB = ND - ICW + 1
          IF (ICB.GT.NB) ICB = NB
          DO 50 I=1,ICB
             IDI = 0
             ICWW = ICW + I - 1
             IF (ICWW.GT.NB)IDI = ICWW - NB
             IWV = (ICW-2+I) * NB + ICW - IDI
             IF (I.EQ.1) GO TO 30
             A(IWV) = 0.0D0
             GO TO 50
 30          A(IWV) = 1.0D0
             IF(ICW.EQ.1) GO TO 50
             DO 40 J=1,ICW-IDI-1
                IWWV = IWV - J
                A(IWWV) = 0.0D0
 40          CONTINUE
 50       CONTINUE
C
          KCHK = KCHK - 10000
C
C         ELIMINATE SLOPE, THETA X
C
 60       IF ( KCHK.LT.01000) GO TO 100
```

```
      ICB = ND - ICTHX + 1
      IF(ICB.GT.NB) ICB = NB
      DO 90 I=1,ICB
         IDI = 0
         ICTHXX = ICTHX + I -1
         IF(ICTHXX.GT.NB)IDI = ICTHXX - NB
         ITHXV = (ICTHX-2+I) * NB + ICTHX - IDI
         IF (I.EQ.1) GO TO 70
         A(ITHXV) = 0.0D0
         GO TO 90
 70      A(ITHXV) = 1.0D0
         IF(ICTHX.EQ.1) GO TO 90
         DO 80 J=1,ICTHX-IDI-1
            ITHXXV = ITHXV - J
            A(ITHXXV) = 0.0D0
 80      CONTINUE
 90   CONTINUE
C
      KCHK = KCHK - 01000
C
C     ELIMINATE SLOPE, THETA Y
C
 100  IF (KCHK.LT.00100) GO TO 140
      ICB = ND - ICTHY + 1
      IF ( ICB.GT.NB) ICB = NB
      DO 130 I=1,ICB
         IDI = 0
         ICTHYY = ICTHY + I - 1
         IF(ICTHYY.GT.NB)IDI = ICTHYY - NB
         ITHYV = (ICTHY-2+I) * NB + ICTHY - IDI
         IF(I.EQ.1) GO TO 110
         A(ITHYV) = 0.0D0
         GO TO 130
 110     A(ITHYV) = 1.0D0
         IF(ICTHY.EQ.1) GO TO 130
         DO 120 J=1,ICTHY - IDI - 1
            ITHYYV = ITHYV -J
            A(ITHYYV) = 0.0D0
 120     CONTINUE
 130  CONTINUE
C
      KCHK = KCHK - 00100
C
C     ELIMINATE MICRO-ROTATION, 'MR X'
C
 140  IF(KCHK.LT.00010) GO TO 180
      ICB = ND - ICMRX + 1
      IF(ICB.GT.NB)ICB = NB
      DO 170 I=1,ICB
         IDI = 0
         ICMRXX = ICMRX + I - 1
         IF(ICMRXX.GT.NB) IDI = ICMRXX - NB
         IMRXV = (ICMRX-2+I) * NB + ICMRX - IDI
         IF(I.EQ.1) GO TO 150
         A(IMRXV) = 0.0D0
         GO TO 170
 150     A(IMRXV) = 1.0D0
         IF(ICMRX.EQ.1) GO TO 170
         DO 160 J = 1,ICMRX - IDI - 1
            IMRXXV = IMRXV - J
            A(IMRXXV) = 0.0D0
 160     CONTINUE
 170  CONTINUE
C
      KCHK = KCHK - 00010
C
```

```
C        ELIMINATE Y MICRO-ROTATION, 'MR Y'
C
 180     IF(KCHK.LT.00001) GO TO 210
         ICB = ND - ICMRY + 1
         IF(ICB.GT.NB)ICB = NB
         DO 210 I=1,ICB
            IDI = 0
            ICMRYY = ICMRY + I - 1
            IF(ICMRYY.GT.NB)IDI = ICMRYY - NB
            IMRYV = (ICMRY-2+I) * NB + ICMRY - IDI
            IF(I.EQ.1) GO TO 190
            A(IMRYV) = 0.0D0
            GO TO 210
 190        A(IMRYV) = 1.0D0
            IF(ICMRY.EQ.1) GO TO 210
            DO 200 J=1,ICMRY - IDI - 1
               IMRYYV = IMRYV - J
               A(IMRYYV) = 0.0D0
 200        CONTINUE
 210  CONTINUE
C
C     CALCULATE COLUMNM HEIGHTS
C
      DO 230 I=1,ND
         IDI = 0
         IF(I.GT.NB) IDI = I - NB
         IIV = (I-1) * NB
         DO 220 J=1,I
            IF(A(IIV+J).EQ.0.0D0) GO TO 220
            MHT(I) = I - J - IDI
            GO TO 230
 220     CONTINUE
 230  CONTINUE
C
C     PROGRAM TO CALCULATE ADDRESES OF DIAGONAL ELEMENTS
C     IN BANDED MATRIX WHOSE CLUMN HEIGHTS ARE KNOWN
C
C     MHT = ACTIVE COLUMN HEIGHTS
C     MAXA = ADDRESES OF DIAGONAL ELEMENTS
C
      NM = ND + 1
      DO 240 I=1,NM
         MAXA(I) = 0
 240  CONTINUE
C
      MAXA(1) = 1
      MAXA(2) = 2
      IF(ND.EQ.1) GO TO 260
      DO 250 I=2,ND
         MAXA(I+1) = MAXA(I) + MHT(I) + 1
 250  CONTINUE
 260  NWK = MAXA(ND+1) - MAXA(1)
C
C     TO STORE STIFFNESS MATRIX(NB*ND) IN COMPACTED
C     FORM A(NWK)
C
      IAN = 0
      DO 270 I=1,ND
         ICK = MAXA(I+1) - MAXA(I)
         IDI = 0
         IF(I.GT.NB)IDI = I - NB
         INBB = (I-1) * NB + I - IDI
         DO 270 II=1,ICK
            IAN = IAN + 1
            IAV = INBB - II + 1
            A(IAN) = A(IAV)
```

```
 270       CONTINUE
C
C      CLEAR THE REST OF THE A(NB*ND) ARRAY
C
       DO 280 I=NWK+1,NBND
       A(I) = 0.0D0
 280   CONTINUE
       RETURN
       END
C*********************************************************************
       SUBROUTINE ELSTIFF
C*********************************************************************
       IMPLICIT REAL*8(A-H,O-Z)
       COMMON/FSK/A(20000),V(500),EK(40,40),MAXA(501)
       COMMON/AIN/NB,NC,ND,NE,NN,TH,NINT,DET,R,S,NELEM
       COMMON/NNNN/NNP(64,8)
       COMMON/XYZ/X(100),Y(100)
       COMMON/MAT/D0(6,6),D1(4,4),A1111,A1122,A2222,A1212,A1221,A2121,
      *                    B1111,B1122,B2222,B1212,B1221,B2121
       COMMON/B01/B0(64,6,40),B1(64,4,40),ID(500),INXY(200)
       DIMENSION D0B0(6),D1B1(4),XG(4,4),WGT(4,4)
       DATA  XG/0.0D0,0.0D0,0.0D0,0.0D0,-0.5773502691896D0,
      1 0.5773502691896D0,0.0D0,0.0D0,-0.7745966692415D0,
      2 0.0D0,0.7745966692415D0,0.0D0,-0.8611363115941D0,
      3 -0.3399810435849D0,0.3399810435849D0,
      4 -0.8611363115941D0/
          DATA WGT/2.0D0,0.0D0,0.0D0,0.0D0,1.0D0,1.0D0,0.0D0,
      1  0.0D0,0.5555555555556D0,0.8888888888889D0,
      2  0.5555555555556D0,0.0D0,0.3478548451375D0,
      3  0.6521451548625D0,0.6521451548625D0,
      4  0.3478548451375D0/
C
C
C
       DO 5 I=1,6
       DO 5 J=1,6
          D0(I,J) = 0.0D0
 5     CONTINUE
       DO 10 I=1,4
       DO 10 J=1,4
          D1(I,J) = 0.0D0
 10    CONTINUE
C
       D0(1,1) = A1111
       D0(1,2) = A1122
       D0(2,1) = A1122
       D0(2,2) = A2222
       D0(3,3) = A1212
       D0(3,4) = A1221
       D0(4,3) = A1221
       D0(4,4) = A2121
       D0(5,5) = A1212
       D0(5,6) = A1221
       D0(6,5) = A1221
       D0(6,6) = A2121
C
       D1(1,1) = B1111
       D1(1,2) = B1122
       D1(2,1) = B1122
       D1(2,2) = B2222
       D1(3,3) = B1212
       D1(3,4) = B1221
       D1(4,3) = B1221
       D1(4,4) = B2121
C
C      DETERMINE NODAL POINTS OF ELEMENT N
```

```
C
      DO 30 I=1,40
      DO 30 J=1,40
         EK(I,J) = 0.0D0
 30   CONTINUE
      DO 80 LX = 1,nint
         R = XG(LX,nint)
      DO 80 LY = 1,nint
         S = XG(LY,nint)
C
         CALL STDM
         WT = WGT(LX,NINT) * WGT(LY,NINT) * DET
         DO 70 J=1,40
            DO 40 K=1,6
               D0B0(K) = 0.0D0
               DO 40 L=1,6
               D0B0(K) = D0B0(K) + D0(K,L) * B0(NELEM,L,J)
 40         CONTINUE
            DO 45 K=1,4
               D1B1(K) = 0.0D0
               DO 45 L=1,4
               D1B1(K) = D1B1(K) + D1(K,L) * B1(NELEM,L,J)
 45         CONTINUE
            DO 60 I=J,40
               STIFF = 0.0D0
               DO 50 L=1,6
                  STIFF = STIFF + B0(NELEM,L,I) * D0B0(L)
 50            CONTINUE
               DO 55 L=1,4
                  STIFF = STIFF + B1(NELEM,L,I) * D1B1(L)
 55            CONTINUE
               EK(I,J) = EK(I,J) + STIFF * WT
 60         CONTINUE
 70      CONTINUE
 80   CONTINUE
      DO 90 J=1,20
      DO 90 I=J,20
         EK(J,I) = EK(I,J)
 90   CONTINUE
      IF(INXY(NELEM).EQ.100) GOTO 100
      IF(INXY(NELEM).EQ.010) GOTO 110
      IF(INXY(NELEM).EQ.001) GOTO 120
      IF(INXY(NELEM).EQ.011) GOTO 130
      R = 0.0D0
      S = 0.0D0
      GOTO 170
 100  R = 1.0D0
      S = 1.0D0
      GOTO 170
 110  R = -1.0D0
      S = 1.0D0
      GOTO 170
 120  R = -1.0D0
      S = -1.0D0
      GOTO 170
 130  R = 1.0D0
      S = -1.0D0
 170  CALL STDM
      RETURN
      END
C*******************************************************
      SUBROUTINE STDM
C*******************************************************
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/FSK/A(20000),V(500),EK(40,40),MAXA(501)
      COMMON/XYZ/X(100),Y(100)
```

```
      COMMON/NNNN/NNP(64,8)
      COMMON /B01/B0(64,6,40),B1(64,4,40),ID(500),INXY(200)
      COMMON/AIN/NB,NC,ND,NE,NN,TH,NINT,DET,R,S,NELEM
C
      DIMENSION PP(4,4),HH(4),P(2,8),H(8),XY(2,8),XJ(2,2),XJI(2,2)
C
      II = NNP(NELEM,1)
      JJ = NNP(NELEM,2)
      KK = NNP(NELEM,3)
      LL = NNP(NELEM,4)
      III = NNP(NELEM,5)
      JJJ = NNP(NELEM,6)
      KKK = NNP(NELEM,7)
      LLL = NNP(NELEM,8)
C
      RP = 1.0D0 + R
      RM = 1.0D0 - R
      SP = 1.0D0 + S
      SM = 1.0D0 - S
      RSM = 1.0D0 - (R*R)
      SSM = 1.0D0 - (S*S)
C
C     INTERPOLATION FUNCTIONS
C
      H(1) = 0.25D0 * RP * SP * (R+S-1.0D0)
      H(2) = 0.25D0 * RM * SP * (-R+S-1.0D0)
      H(3) = 0.25D0 * RM * SM * (-R-S-1.0D0)
      H(4) = 0.25D0 * RP * SM * (R-S-1.0D0)
      H(5) = 0.5D0 * RSM * SP
      H(6) = 0.5D0 * SSM * RM
      H(7) = 0.5D0 * RSM * SM
      H(8) = 0.5D0 * SSM * RP
C
C     NATURAL CO-ORDINATE DERIVATIVES
C
      P(1,1) = 0.25D0 * SP * (R+R+S)
      P(1,2) = 0.25D0 * SP * (R+R-S)
      P(1,3) = 0.25D0 * SM * (R+R+S)
      P(1,4) = 0.25D0 * SM * (R+R-S)
      P(1,5) = -R * SP
      P(1,6) = -0.5D0 * SSM
      P(1,7) = -R * SM
      P(1,8) = 0.5D0 * SSM
      P(2,1) = 0.25D0 * RP * (R+S+S)
      P(2,2) = 0.25D0 * RM * (S+S-R)
      P(2,3) = 0.25D0 * RM * (R+S+S)
      P(2,4) = 0.25D0 * RP * (S+S-R)
      P(2,5) = 0.5D0 * RSM
      P(2,6) = -S * RM
      P(2,7) = -0.5D0 * RSM
      P(2,8) = -S * RP
C
      XY(1,1) = X(II)
      XY(1,2) = X(JJ)
      XY(1,3) = X(KK)
      XY(1,4) = X(LL)
      XY(1,5) = X(III)
      XY(1,6) = X(JJJ)
      XY(1,7) = X(KKK)
      XY(1,8) = X(LLL)
      XY(2,1) = Y(II)
      XY(2,2) = Y(JJ)
      XY(2,3) = Y(KK)
      XY(2,4) = Y(LL)
      XY(2,5) = Y(III)
      XY(2,6) = Y(JJJ)
```

```
      XY(2,7)  = Y(KKK)
      XY(2,8)  = Y(LLL)
C
C     EVALUATE JACOBIAN AT (R,S)
C
      DO 30 I= 1,2
         DO 30 J=1,2
         DUM = 0.0D0
         DO 20 K=1,8
            DUM = DUM + P(I,K) * XY(J,K)
 20      CONTINUE
         XJ(I,J) = DUM
 30   CONTINUE
C
C     COMPUTE DETERMINANT AT (R,S)
C
      DET = XJ(1,1) * XJ(2,2) - XJ(2,1) * XJ(1,2)
      IF (DET.GT.1.0D-07) GOTO 40
      WRITE(6,2000) NELEM
 2000 format(3X,'*** ERROR, ZERO OR NEGATIVE JACOBIAN
     1DETERMINANT AT ELEMENT = ',I4)
      STOP
C
C     COMPUTE INVERSE OF JACOBIAN MATRIX
C
 40   CONTINUE
      DUM = 1.0/DET
      XJI(1,1) = XJ(2,2) * DUM
      XJI(1,2) = - XJ(1,2) * DUM
      XJI(2,1) = - XJ(2,1) * DUM
      XJI(2,2) = XJ(1,1) * DUM
C
C     EVALUATE B0 AND B1 MATRICES
C
      DO 50 I=1,6
         DO 50 J=1,40
         B0(NELEM,I,J) = 0.0D0
 50   CONTINUE
C
      DO 55 I=1,4
         DO 55 J=1,40
            B1(NELEM,I,J) = 0.0D0
 55   CONTINUE
C
      B0(NELEM,1,2) = (XJI(1,1)*P(1,1) + XJI(1,2)*P(2,1))
      B0(NELEM,1,7) = (XJI(1,1)*P(1,2) + XJI(1,2)*P(2,2))
      B0(NELEM,1,12)= (XJI(1,1)*P(1,3) + XJI(1,2)*P(2,3))
      B0(NELEM,1,17)= (XJI(1,1)*P(1,4) + XJI(1,2)*P(2,4))
      B0(NELEM,1,22)= (XJI(1,1)*P(1,5) + XJI(1,2)*P(2,5))
      B0(NELEM,1,27)= (XJI(1,1)*P(1,6) + XJI(1,2)*P(2,6))
      B0(NELEM,1,32)= (XJI(1,1)*P(1,7) + XJI(1,2)*P(2,7))
      B0(NELEM,1,37)= (XJI(1,1)*P(1,8) + XJI(1,2)*P(2,8))
C
      B0(NELEM,2,3) = (XJI(2,1)*P(1,1) + XJI(2,2)*P(2,1))
      B0(NELEM,2,8) = (XJI(2,1)*P(1,2) + XJI(2,2)*P(2,2))
      B0(NELEM,2,13)= (XJI(2,1)*P(1,3) + XJI(2,2)*P(2,3))
      B0(NELEM,2,18)= (XJI(2,1)*P(1,4) + XJI(2,2)*P(2,4))
      B0(NELEM,2,23)= (XJI(2,1)*P(1,5) + XJI(2,2)*P(2,5))
      B0(NELEM,2,28)= (XJI(2,1)*P(1,6) + XJI(2,2)*P(2,6))
      B0(NELEM,2,33)= (XJI(2,1)*P(1,7) + XJI(2,2)*P(2,7))
      B0(NELEM,2,38)= (XJI(2,1)*P(1,8) + XJI(2,2)*P(2,8))
C
      B0(NELEM,3,3) = B0(NELEM,1,2)
      B0(NELEM,3,8) = B0(NELEM,1,7)
      B0(NELEM,3,13)= B0(NELEM,1,12)
      B0(NELEM,3,18)= B0(NELEM,1,17)
```

```
      B0(NELEM,3,23)= B0(NELEM,1,22)
      B0(NELEM,3,28)= B0(NELEM,1,27)
      B0(NELEM,3,33)= B0(NELEM,1,32)
      B0(NELEM,3,38)= B0(NELEM,1,37)
C
      B0(NELEM,4,2)  = B0(NELEM,2,3)
      B0(NELEM,4,7)  = B0(NELEM,2,8)
      B0(NELEM,4,12)= B0(NELEM,2,13)
      B0(NELEM,4,17)= B0(NELEM,2,18)
      B0(NELEM,4,22)= B0(NELEM,2,23)
      B0(NELEM,4,27)= B0(NELEM,2,28)
      B0(NELEM,4,32)= B0(NELEM,2,33)
      B0(NELEM,4,37)= B0(NELEM,2,38)
C
      B0(NELEM,5,1)  = (B0(NELEM,1,2))
      B0(NELEM,5,6)  = (B0(NELEM,1,7))
      B0(NELEM,5,11)  = (B0(NELEM,1,12))
      B0(NELEM,5,16)  = (B0(NELEM,1,17))
      B0(NELEM,5,21)  = (B0(NELEM,1,22))
      B0(NELEM,5,26)  = (B0(NELEM,1,27))
      B0(NELEM,5,31)  = (B0(NELEM,1,32))
      B0(NELEM,5,36)  = (B0(NELEM,1,37))
      B0(NELEM,5,5)  = H(1)
      B0(NELEM,5,10)  = H(2)
      B0(NELEM,5,15) .= H(3)
      B0(NELEM,5,20)  = H(4)
      B0(NELEM,5,25)  = H(5)
      B0(NELEM,5,30)  = H(6)
      B0(NELEM,5,35)  = H(7)
      B0(NELEM,5,40)  = H(8)
C
      B0(NELEM,5,2)  = - H(1)
      B0(NELEM,5,7)  = - H(2)
      B0(NELEM,5,12)= - H(3)
      B0(NELEM,5,17)= - H(4)
      B0(NELEM,5,22)= - H(5)
      B0(NELEM,5,27)= - H(6)
      B0(NELEM,5,32)= - H(7)
      B0(NELEM,5,37)= - H(8)
C
      B0(NELEM,6,1)  = (B0(NELEM,4,2))
      B0(NELEM,6,6)  = (B0(NELEM,4,7))
      B0(NELEM,6,11)  = (B0(NELEM,4,12))
      B0(NELEM,6,16)  = (B0(NELEM,4,17))
      B0(NELEM,6,21)  = (B0(NELEM,4,22))
      B0(NELEM,6,26)  = (B0(NELEM,4,27))
      B0(NELEM,6,31)  = (B0(NELEM,4,32))
      B0(NELEM,6,36)  = (B0(NELEM,4,37))
      B0(NELEM,6,4)= -H(1)
      B0(NELEM,6,9)= -H(2)
      B0(NELEM,6,14)= -H(3)
      B0(NELEM,6,19)= -H(4)
      B0(NELEM,6,24)= -H(5)
      B0(NELEM,6,29)= -H(6)
      B0(NELEM,6,34)= -H(7)
      B0(NELEM,6,39)= -H(8)
C
      B0(NELEM,6,3)  = -H(1)
      B0(NELEM,6,8)  = -H(2)
      B0(NELEM,6,13)= -H(3)
      B0(NELEM,6,18)= -H(4)
      B0(NELEM,6,23)= -H(5)
      B0(NELEM,6,28)= -H(6)
      B0(NELEM,6,33)= -H(7)
      B0(NELEM,6,38)= -H(8)
C
```

```
      B1(NELEM,1,4)  = B0(NELEM,1,2)
      B1(NELEM,1,9)  = B0(NELEM,1,7)
      B1(NELEM,1,14)  = B0(NELEM,1,12)
      B1(NELEM,1,19)  = B0(NELEM,1,17)
      B1(NELEM,1,24)  = B0(NELEM,1,22)
      B1(NELEM,1,29)  = B0(NELEM,1,27)
      B1(NELEM,1,34)  = B0(NELEM,1,32)
      B1(NELEM,1,39)  = B0(NELEM,1,37)
C
      B1(NELEM,2,4)  = B0(NELEM,2,3)
      B1(NELEM,2,9)  = B0(NELEM,2,8)
      B1(NELEM,2,14)  = B0(NELEM,2,13)
      B1(NELEM,2,19)  = B0(NELEM,2,18)
      B1(NELEM,2,24)  = B0(NELEM,2,23)
      B1(NELEM,2,29)  = B0(NELEM,2,28)
      B1(NELEM,2,34)  = B0(NELEM,2,33)
      B1(NELEM,2,39)  = B0(NELEM,2,38)
C
      B1(NELEM,3,5)  =  B0(NELEM,3,3)
      B1(NELEM,3,10)  =  B0(NELEM,3,8)
      B1(NELEM,3,15)  =  B0(NELEM,3,13)
      B1(NELEM,3,20)  =  B0(NELEM,3,18)
      B1(NELEM,3,25)  =  B0(NELEM,3,23)
      B1(NELEM,3,30)  =  B0(NELEM,3,28)
      B1(NELEM,3,35)  =  B0(NELEM,3,33)
      B1(NELEM,3,40)  =  B0(NELEM,3,38)
C
      B1(NELEM,4,5)  =  B0(NELEM,4,2)
      B1(NELEM,4,10)  =  B0(NELEM,4,7)
      B1(NELEM,4,15)  =  B0(NELEM,4,12)
      B1(NELEM,4,20)  =  B0(NELEM,4,17)
      B1(NELEM,4,25)  =  B0(NELEM,4,22)
      B1(NELEM,4,30)  =  B0(NELEM,4,27)
      B1(NELEM,4,35)  =  B0(NELEM,4,32)
      B1(NELEM,4,40)  =  B0(NELEM,4,37)
C
C
      RETURN
      END
C
C *******************************************************************
C *                                                                 *
                          SUBROUTINE COLSOL
C *                                                                 *
C *******************************************************************
C********************************************************************
C*                                                                  *
C*    PROGRAM                                                       *
C*       TO SOLVE FINITE ELEMENT STATIC EQUILIBRIUM EQUATIONS IN    *
C*       CORE, USING COMPACTED STORAGE AND COLUMN REDUCTION SCHEME  *
C*                                                                  *
C*    --INPUT VARIABLES--                                           *
C*       A(NWK)       = STIFFNESS MATRIX STORED IN COMPACTED FORM   *
C*       V(ND)        = RIGHT-HAND-SIDE LOAD VECTOR                 *
C*       MAXA(ND+1)   = VECTOR CONTAINING ADDRESSES OF DIAGONAL     *
C*                      ELEMENTS OF STIFFNESS MATRIX IN A           *
C*       ND           = NUMBER OF EQUATIONS                         *
C*       NWK          = NUMBER OF ELEMENTS BELOW SKYLINE OF MATRIX  *
C*                                                                  *
C*    --OUTPUT--                                                    *
C*       A(NWK)       = D AND L - FACTORS OF STIFFNESS MATRIX       *
C*       V(ND)        = DISPLACEMENT VECTOR                         *
C*                                                                  *
C********************************************************************
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/AIN/NB,NC,ND,NE,NN,TH,NINT,DET,R,S,NELEM
```

```
      COMMON/FSK/A(20000),V(500),EK(40,40),MAXA(501)
C
C     PERFORM L*D*L(T) FACTORIZATION OF STIFFNESS MATRIX
C
C
C
      DO 140 N=1,ND
         KN=MAXA(N)
         KL=KN+1
         KU=MAXA(N+1)-1
         KH=KU-KL
         IF(KH)110,90,50
50       K=N-KH
         IC=0
         KLT=KU
         DO 80 J=1,KH
            IC=IC+1
            KLT=KLT-1
            KI=MAXA(K)
            NND=MAXA(K+1)-KI-1
            IF(NND)80,80,60
60          KK=MIN0(IC,NND)
            C=0.0D0
            DO 70 L=1,KK
               C=C+A(KI+L)*A(KLT+L)
 70         CONTINUE
            A(KLT)=A(KLT)-C
            K=K+1
 80      CONTINUE
90       K=N
         B=0.0D0
         DO 100 KK=KL,KU
            K=K-1
            KI=MAXA(K)
            C=A(KK)/A(KI)
            B=B+C*A(KK)
            A(KK)=C
 100     CONTINUE
         A(KN)=A(KN)-B
110      IF(A(KN))120,120,140
120      WRITE(6,2000)N,A(KN),KN
         STOP
140   CONTINUE
C
C
C
C     REDUCE RIGHT-HAND-SIDE LOAD VECTOR
C

      DO 180 N=1,ND
         KL=MAXA(N)+1
         KU=MAXA(N+1)-1
         IF(KU-KL)180,160,160
160      K=N
         C=0.0D0
         DO 170 KK=KL,KU
            K=K-1
            C=C+A(KK)*V(K)
 170     CONTINUE
         V(N)=V(N)-C
180   CONTINUE
C
C     BACK-SUBSTITUTE
C
      DO 200 N=1,ND
         K=MAXA(N)
```

```
         V(N)=V(N)/A(K)
  200   CONTINUE
        IF(ND.EQ.1)RETURN
        N=ND
        DO 230 L=2,ND
           KL=MAXA(N)+1
           KU=MAXA(N+1)-1
           IF(KU-KL)230,210,210
 210       K=N
           DO 220 KK=KL,KU
              K=K-1
              V(K)=(V(K)-A(KK)*V(N))
  220      CONTINUE
           N=N-1
  230   CONTINUE
        WRITE(6,1000)
  1000  FORMAT(//25X,'*** NODAL DISPLACEMENT ***')
        WRITE(6,1500)
  1500  FORMAT(/1X,'NODE',4X,'W-DISP',5X,'X-SLOPE',4X,'Y-SLOPE',
       *        4X,'X-ROTN',5X,'Y-ROTN')
        IND=0
        DO 250 K=1,ND,5
           IND=IND+1
           WRITE(6,*) IND,V(K  ),V(K+1),V(K+2),
       *                  V(K+3),V(K+4)
  250   CONTINUE
  2000  FORMAT(//46H STOP - STIFFNESS MATRIX NOT POSITIVE DEFINITE,//
       *           32H NONPOSITIVE PIVOT FOR EQUATION ,I4,//
       *           10H PIVOT =  ,E20.12,I6)
  3000  FORMAT(2X,I3,5(5X,E7.3))
        RETURN
        END
C
C **********************************************************************
C *                                                                    *
                           SUBROUTINE STRESS
C *                                                                    *
C **********************************************************************
        IMPLICIT REAL*8(A-H,O-Z)
        COMMON/B01/B0(64,6,40),B1(64,4,40),ID(500),INXY(200)
        COMMON/NNNN/NNP(64,8)
        COMMON/XYZ/X(100),Y(100)
        COMMON/MAT/D0(6,6),D1(4,4),A1111,A1122,A2222,A1212,A1221,A2121
       *                          B1111,B1122,B2222,B1212,B1221,B2121
        COMMON/AIN/NB,NC,ND,NE,NN,TH,NINT,DET,R,S,NELEM
        COMMON/FSK/A(20000),V(500),EK(40,40),MAXA(501)
C
        DIMENSION EDISP(20),E(4,6),PHIJ(4,4),T(4,6),CM(4,4),U(500)
C
C PICK UP ELEMENT NODAL DISPLACEMENTS EDISP( 9)
C FROM STRUCTURAL NODAL DISPLACEMENTS SDISP(ND)
C
C FOR EACH ELEMENT "IE"
        DO 300 IE=1,NE
           DO 20 IJM=1,8
              IEB=(IJM-1)*5
              ISB=(NNP(IE,IJM)-1)*5
           DO 20 IDOF=1,5
              EDISP(IEB+IDOF)=V(ISB+IDOF)
  20       CONTINUE
C
C DISPLACEMENT STRAIN : EIJ=B0*UE
           DO 40 IC=1,6
              SUM=0.0D0
              DO 30 K=1,40
                 SUM=SUM+B0(IE,IC,K)*EDISP(K)
```

```
 30          CONTINUE
             E(IE,IC)=SUM
 40       CONTINUE
C
C ROTATION STRAIN : PHI,J=B1*UE
          DO 60 IC=1,4
             SUM=0.0D0
             DO 50 K=1,40
                SUM=SUM+B1(IE,IC,K)*EDISP(K)
 50          CONTINUE
             PHIJ(IE,IC)=SUM
 60       CONTINUE
C
C FORCE STRESS : TIJ=D0*EIJ
          DO 80 IC=1,6
             SUM=0.0D0
             DO 70 K=1,6
                SUM=SUM+D0(IC,K)*E(IE,K)
 70          CONTINUE
             T(IE,IC)=SUM
 80       CONTINUE
C
C COUPLE STRESS : MIJ=D1*PHIJ
          DO 100 IC=1,4
             SUM=0.0D0
             DO 90 K=1,4
                SUM=SUM+D1(IC,K)*PHIJ(IE,K)
 90          CONTINUE
             CM(IE,IC)=SUM
 100      CONTINUE
C STRAIN-ENERGY : U
          SUM=0.0D00
          DO 200 IC=1,6
             SUM=SUM+E(IE,IC)*T(IE,IC)
 200      CONTINUE
          DO 250 IC=1,4
             SUM=SUM+PHIJ(IE,IC)*CM(IE,IC)
 250      CONTINUE
          U(IE)=SUM*0.50D00
  300 CONTINUE
C
C WRITE THE CALCULATED STRAINS AND STRESSES
      WRITE(6,1000)
 1000 FORMAT(//19X,'**** STRESSES & STRAINS CALCULATED ****')
      WRITE(6,2000)
 2000 FORMAT(/,'ELMT',1X,'COMP',1X,'DISP-STRAN',3X,'FORCE-STRS'
     *      ,2X,'COMP',1X,'ROTATN-GRAD',2X,'COUPLE-STRS'
     *      ,2X,'STRN-ENEGY')
      WRITE(6,3000)
 3000 FORMAT(14X,'E',12X,'T',15X,'PHI,J',9X,'M',12X,'U')
      DO 400 IE=1,NE
  999 CONTINUE
      WRITE(6,4000) IE
 4000 FORMAT(I6)
      WRITE(6,5000) E(IE,1),T(IE,1),PHIJ(IE,1),CM(IE,1),U(IE)
 5000 FORMAT(6X,'XX',1X,1PE11.4,2X,1PE11.4,2X,'XX',1X,1PE11.4,
     *        2X,1PE11.4,2X,1PE11.4)
      WRITE(6,6000) E(IE,2),T(IE,2),PHIJ(IE,2),CM(IE,2)
 6000 FORMAT(6X,'YY',1X,1PE11.4,2X,1PE11.4,2X,'XY',1X,1PE11.4,
     *        2X,1PE11.4)
      WRITE(6,7000) E(IE,3),T(IE,3),PHIJ(IE,3),CM(IE,3)
 7000 FORMAT(6X,'XY',1X,1PE11.4,2X,1PE11.4,2X,'YZ',1X,1PE11.4,
     *        2X,1PE11.4)
      WRITE(6,8000) E(IE,4),T(IE,4),PHIJ(IE,4),CM(IE,4)
 8000 FORMAT(6X,'YX',1X,1PE11.4,2X,1PE11.4,2X,'YY',1X,1PE11.4,
     *        2X,1PE11.4)
```

```
      WRITE(6,8100)E(IE,5),T(IE,5)
 8100 FORMAT(6X,'XZ',1X,1PE11.4,2X,1PE11.4)
      WRITE(6,8200)E(IE,6),T(IE,6)
 8200 FORMAT(6X,'ZX',1X,1PE11.4,2X,1PE11.4)
  400 CONTINUE
      RETURN
                                        END
C**********************************************************************
C*                                                                   *
                        SUBROUTINE LOADER
C*                                                                   *
C**********************************************************************
C
         IMPLICIT REAL*8(A-H,O-Z)
         COMMON/AIN/NB,NC,ND,NE,NN,TH,NINT,DET,R,S,NELEM
         COMMON/FSK/A(20000),V(500),EK(40,40),MAXA(501)
         DO 10 LOAD = 1,ND
            v(LOAD) = 0.0D0
 10      CONTINUE
         SCALE=1.0D0
         V(11)=1.0D0 * SCALE
         V(31)=1.0D0 * SCALE
         V(51)=1.0D0 * SCALE
         V(36)=1.0D0 * SCALE
         V(26)=1.0D0 .* SCALE
         V(56)=1.0D0 * SCALE
         V(71)=1.0D0 * SCALE
         V(66)=1.0D0 * SCALE
         V(61) = 1.0D0 * SCALE
         WRITE(6,1000)
 1000    FORMAT(//34X,'**** EXTERNAL LOAD ****')
         WRITE(6,2000)
 2000    FORMAT(//X,'NODE',8X,'W-FORCE',7X,'X-MOMENT',7X,'Y-MOMENT',
     1   7X,'X-MICROROT',5X,'Y-MICROROT')
         IND = 0
         DO 20 N = 1,ND,5
            IND = IND+1
            CHECK = ABS(V(N)) + ABS(V(N+1)) + ABS(V(N+2))
     1      +ABS(V(N+3)) + ABS(V(N+4))
            IF(CHECK.EQ.0.0D0) GOTO 20
            WRITE(6,3000)IND,V(N),V(N+1),V(N+2),V(N+3),V(N+4)
 20      CONTINUE
 3000    FORMAT(2X,I3,5(5X,E10.3))
         RETURN
         END
```

```
C                        3-D MICROPOLAR FINITE ELEMENT METHOD
C  Modified version of general three dimensional program developed
C  by Jen and S. Nakamura. This program includes effects of microrotation
C  in isotropic linear micropolar elasticity. Output to post-processing
C  is also included.
C                            ** PROGRAMMING ORGANIZATION **
C
C                           MAIN
C                            I
C                            I---STSTIFF---ELSTIF(EK,NE)
C                            I
C                            I---LOADER
C                            I
C                            I---COLSOL
C                            I
C                            I---STRESS
C
C                           ** EXPLANATION OF THE SYMBOLS **
C
C                                        SCALAR
C     AIJ         ....   MATERIAL PROPERTIES FOR FORCE -STRESS
C     BIJ         ....   MATERIAL PROPERTIES FOR COUPLE-STRESS
C     NB          ....   NUMBER OF BAND-WIDTH OF A(NB*ND)
C     NC          ....   NUMBER OF TOTAL CONSTRAINED NODE
C     ND          ....   NUMBER OF TOTAL DEGREE OF FREEDOM
C     NE          ....   NUMBER OF TOTAL ELEMENT
C     NN          ....   NUMBER OF TOTAL NODAL POINTS
C     TH          ....   THICKNESS OF THE ELEMENT
C
C                                        VECTOR
C     A   (NB*ND)....   STRUCTURAL STIFFNESS VECTOR
C     A      (NWK)....   D AND L - FACTORS OF STIFFNESS MATRIX
C                        FOR OUTPUT
C     V       (ND)....   STRUCTURAL NODAL DISPLACEMENTS & ROTATIONS
C                        FOR OUTPUT
C     EDISP (48)....   ELEMENT    NODAL DISPLACEMENTS & ROTATIONS
C     KSTRN (NC)....   CONSTRAINED NODE
C     KSTRT (NC)....   TYPE OF CONSTRAIN
C                           100000  X-DISPLACEMENT  IS FIXED
C                           010000  Y-DISPLACEMENT  IS FIXED
C                           001000  Z-DISPLACEMENT  IS FIXED
C                           000100  X-ROTATION      IS FIXED
C                           000010  Y-ROTATION      IS FIXED
C                           000001  Z-ROTATION      IS FIXED
C                           100100  X-DIS. & X-ROT. IS FIXED
C                           100001  X-DIS. & Z-ROT. IS FIXED
C     INXY    (NE)....   LOCATION OF INTEREST IN ELEMENT FOR OUTPUT
C                           00001  AT NODAL POINT  1
C                           00010  AT NODAL POINT  2
C                           00011  AT NODAL POINT  3
C                           00100  AT NODAL POINT  4
C                           00101  AT NODAL POINT  5
C                           00110  AT NODAL POINT  6
C                           00111  AT NODAL POINT  7
C                           01000  AT NODAL POINT  8
C                           01001  AT NODAL POINT  9
C                           01010  AT NODAL POINT 10
C                           01011  AT NODAL POINT 11
C                           01100  AT NODAL POINT 12
C                           01101  AT NODAL POINT 13
C                           01110  AT NODAL POINT 14
C                           01111  AT NODAL POINT 15
C                           10000  AT NODAL POINT 16
C                           10001  AT NODAL POINT 17
C                           10010  AT NODAL POINT 18
C                           10011  AT NODAL POINT 19
```

```
C                         10100  AT NODAL POINT 20
C                         00000  AT CENTROID OF PARALLELEPIPED ELEMNT
C
C                                      MATRIX
C     E    (NE,9)....  DISPLACEMENT-STRAIN
C     PHIJ(NE,9)....  ROTATION    -STRAIN
C     T    (NE,9)....  FORCE       -STRESS
C     CM   (NE,9)....  COUPLE      -STRESS
C     EK  (48,48)....  ELEMENT    STIFFNESS MATRIX
C     NNP  (NE,8)....  NODE NUMBER FORMING THE ELEMENT
C
C                                      TENSOR
C     B0(NE,9,48)....  DISP-STRAIN DISPLACEMENT MATRIX B
C     B1(NE,9,48)....  ROTN-STRAIN ROTATION     MATRIX B
C
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER*4 ITIM
      INTEGER*2 TYPE,CODE
      COMMON/B01/B0(80,9,48),B1(80,9,48),INXY(80),ID(300)
      COMMON/XYZ/NNP(80,8),X(300),Y(300),Z(300)
      COMMON/CSTRN/KSTRN(74),KSTRT(74)
      COMMON/MAT/D0(9,9),D1(9,9)
      COMMON/AIN/NB,NC,ND,NE,NN,TH,NINT,R,S,T,DET
      COMMON/FSK/A(84530),V(1800),EK(48,48),MAXA(1801),NEIRE
      CALL ASSIGN(5,'p1e48.DAT')
      CALL ASSIGN(6,'p1e48.OUT')
      CALL ASSIGN(60,'p2e16_3d')
C     SAVE
C
C I------------------- INPUT & OUTPUT FILE OPEN ---------------------I
C $INSERT SYSCOM>ERRD.F
C $INSERT SYSCOM>KEYS.F
C $INSERT SYSCOM>A$KEYS
C     CALL SRCH$$(K$DELE,'FEMO',INTS(4),INTS(2),TYPE,CODE)
C     CALL OPEN$A(A$WRIT+A$SAMF,'FEMO',INTS(4),INTS(2))
C     CALL SRCH$$(K$READ,'FEMI',INTS(4),INTS(1),TYPE,CODE)
C     CALL SRCH$$(K$WRIT,'FEMO',INTS(4),INTS(2),TYPE,CODE)
C     CT1=CTIM$A(ITIM)
C
C I-------------------------- DATA INPUT ---------------------------I
C NE? READ TOTAL ELEMENT NUMBER (NE=6)
      READ(5,100) NE
  100 FORMAT(I4)
C NP? READ ELEMENT CONNECTION INFORMATION
      DO 10 N=1,NE
      READ(5,200)  (NNP(N,I),I=1,8),INXY(N)
  200 FORMAT(4X,8I4,I6)
   10 CONTINUE
C NN? READ TOTAL NUMBER OF NODAL POINTS (NN=8)
      READ(5,100) NN
C XY? READ COORDINATE (X,Y,Z) OF NODAL POINTS
      DO 20 N=1,NN
      READ(5,300) X(N),Y(N),Z(N)
   20 CONTINUE
  300 FORMAT(4X,3F20.10)
C NC? READ NUMBER OF CONSTRAINED NODES (NC=3)
      READ(5,100) NC
C KS? READ CONSTRAINTS
      DO 30 N=1,NC
      READ(5,400) KSTRN(N),KSTRT(N)
   30 CONTINUE
  400 FORMAT(4X,I4,I8)
C TH? READ THICKNESS OF THE MATERIAL
      READ(5,500) TH
  500 FORMAT(E11.4)
C     READ GAUSS NUMERICAL INTEGRATION ORDER
```

```
      READ(5,550)NINT
  550 FORMAT(I5)
C AB? READ MATERIAL PROPERTIES
      DO 35 I=1,9
   35 READ(5,600)(D0(I,J),J=I,9)
      DO 36 I=1,9
   36 READ(5,600)(D1(I,J),J=I,9)
      DO 38 I=1,9
      DO 38 J=I,9
      D0(J,I)=D0(I,J)
   38 D1(J,I)=D1(I,J)
  600 FORMAT(9E14.7)
C
C I-------------------- TOTAL DEGREE OF FREEDOM ---------------------I
C EACH NODE HAS 3-DOF OF (UX,UY,PHIZ)
      ND=NN*6
C CALCULATION OF BAND-WIDTH OF STRUCTURAL STIFFNESS MATRIX A(NB*ND)
      NB=0
      DO 40 IB=1,NE
      IMAX=MAX0(NNP(IB,1),NNP(IB,2),NNP(IB,3),NNP(IB,4),NNP(IB,5),
     *NNP(IB,6),NNP(IB,7),NNP(IB,8))
      IMIN=MIN0(NNP(IB,1),NNP(IB,2),NNP(IB,3),NNP(IB,4),NNP(IB,5),
     *NNP(IB,6),NNP(IB,7),NNP(IB,8))
      NBCHEK=(IMAX-IMIN+1)*6
      IF(NBCHEK.GT.NB) NB=NBCHEK
      IF(NB .GT. 1000) GO TO 99
   40 CONTINUE
C
      SCALE=1.0D0
      DO 41 I=1,NN
      X(I)=X(I)*SCALE
      Y(I)=Y(I)*SCALE
   41 Z(I)=Z(I)*SCALE
C
C I-------------------------- DATA OUTPUT ---------------------------I
C PRINT THE HEAD OF OUTPUT
      WRITE(6,700)
  700 FORMAT(1H1)
      WRITE(6,800)
  800 FORMAT(3X,'*****************************************************
     ***************')
      WRITE(6,900)
  900 FORMAT(3X,'*',68X,'*')
      WRITE(6,1000)
 1000 FORMAT(3X,'*',13X,'3-D ORTHOTROPIC MICROPOLAR STRESS ANALYSIS',
     *       13X,'*')
      WRITE(6,1050)
 1050 FORMAT(3X,'*',28X,' SKYLINEMICRO',27X,'*')
      WRITE(6,900)
      WRITE(6,800)
C
C WRITE THE INFORMATIONS
      WRITE(6,1100)
 1100 FORMAT(/20X,'**** DISCRETIZATION NUMBER ****')
      WRITE(6,1200)
 1200 FORMAT(/13X,'ELEMT.#',3X,'NODES.#',3X,'CONSTR.#',3X,'THIKNES',
     *       3X,'BAND-WIDTH',3X,'GAUSS NUMERICAL INTEGRATION ORDER')
      WRITE(6,1300) NE,NN,NC,TH,NB,NINT
 1300 FORMAT(13X,I4,7X,I3,7X,I3,5X,E10.3,5X,I5,19X,I2)
C
C WRITE ORTHOTROPIC MATERIAL PROPERTIES
      WRITE(6,1400)
 1400 FORMAT(//22X,'**** MATERIAL PROPERTIES ****',
     *   //6X,'***D0-MATRIX:'//10X,'1',11X,'2',11X,'3',11X,'4',
     *11X,'5',11X,'6',11X,'7',11X,'8',11X,'9'/)
      WRITE(6,1500)(I,(D0(I,J),J=1,9),I=1,9)
```

```
 1500 FORMAT(I2,2X,9E12.3)
      WRITE(6,1600)
 1600 FORMAT(/6X,'***D1-MATRIX:'/)
      WRITE(6,1500)(I,(D1(I,J),J=1,9),I=1,9)
C
C WRITE ELEMENT CONNECTION INFORMATIONS
      WRITE(60,*)NE
      DO 1650 I=1,NE
         WRITE(60,1750)NNP(I,1),NNP(I,2),NNP(I,3),NNP(I,4)
     *    ,NNP(I,5),NNP(I,6),NNP(I,7),NNP(I,8)
 1650 CONTINUE
 1750 FORMAT(2X,I4,8(1X,I3))
      WRITE(6,1700)
 1700 FORMAT(//21X,'**** ELEMENT-NODE CONNECTION ****')
      WRITE(6,1800)
 1800 FORMAT(/3X,'ELM NP1 NP2 NP3 NP4 NP5 NP6 NP7 NP8    IXY   ELM
     * NP1 NP2 NP3 NP4 NP5 NP6 NP7 NP8    IXY')
      DO 45 I=1,NE
   45 ID(I)=I
      LINE=NE/2
      IRESID=NE-2*LINE
      DO 50 N=1,LINE
   50 WRITE(6,1900)  (ID(2*(N-1)+I),NNP(2*(N-1)+I,1),NNP(2*(N-1)+I,2),
     *NNP(2*(N-1)+I,3),NNP(2*(N-1)+I,4),NNP(2*(N-1)+I,5),
     *NNP(2*(N-1)+I,6),NNP(2*(N-1)+I,7),NNP(2*(N-1)+I,8),
     *INXY(2*(N-1)+I),I=1,2)
 1900 FORMAT(2(2X,I4,1X,I3,1X,I3,1X,I3,1X,I3,1X,I3,1X,I3,1X,I3,1X,I3,
     *1X,I6))
      IF(IRESID .EQ. 0) GO TO 56
      WRITE(6,1900)  (ID(2*LINE+I),NNP(2*LINE+I,1),NNP(2*LINE+I,2),
     *NNP(2*LINE+I,3),NNP(2*LINE+I,4),NNP(2*LINE+I,5),NNP(2*LINE+I,6),
     *NNP(2*LINE+I,7),NNP(2*LINE+I,8),INXY(2*LINE+I),I=1,IRESID)
C
C WRITE NODAL COORDINATES
   56 WRITE(6,2000)
 2000 FORMAT(//24X,'**** NODAL COORDINATE ****')
      WRITE(6,2100)
 2100 FORMAT(/1X,'NODE',5X,'X',8X,'Y',8X,'Z',4X,'NODE',5X,'X',8X,'Y',
     * 8X,'Z',4X,'NODE',5X,'X',8X,'Y',8X,'Z',4X,'NODE',5X,'X',
     * 8X,'Y',8X,'Z')
      DO 55 I=1,NN
   55 ID(I)=I
      LINE=NN/4
      IRESID=NN-4*LINE
      DO 60 N=1,LINE
   60 WRITE(6,2200)  (ID(4*(N-1)+I),X(4*(N-1)+I),Y(4*(N-1)+I),
     * Z(4*(N-1)+I),I=1,4)
 2200 FORMAT(4(2X,I3,1X,F8.3,1X,F8.3,1X,F8.3))
      IF(IRESID .EQ. 0) GO TO 57
      WRITE(6,2200)  (ID(4*LINE+I),X(4*LINE+I),Y(4*LINE+I),
     * Z(4*LINE+I),I=1,IRESID)
      WRITE(60,*)NN
      DO 2250 I=1,NN
         WRITE(60,2260)ID(I),X(I),Y(I),Z(I)
 2250 CONTINUE
 2260 FORMAT(2x,I3,3(1X,F8.3))
C
C WRITE CONSTRAINTS
   57 WRITE(6,2300)
 2300 FORMAT(//27X,'**** CONSTRAINT ****')
      WRITE(6,2400)
 2400 FORMAT(/24X,'CNSTRND-NODE',2X,'CNSTRND-CODE')
      WRITE(6,2500)  (KSTRN(N),KSTRT(N),N=1,NC)
 2500 FORMAT(28X,I3,10X,I6)
      WRITE(60,*)NC
      DO 2550 I=1,NC
```

```
      WRITE(60,2560)KSTRN(I),KSTRT(I)
 2550 CONTINUE
 2560 FORMAT(28X,I3,10X,I6)
C
C I-------------------------- MAIN PROGRAM --------------------------I
                         CALL STSTIF
                         CALL LOADER
                         CALL COLSOL
                         CALL STRESS
C
C     CT3=CTIM$A(ITIM)
C     C=CT3-CT1
C     WRITE(6,9999) C
C
   99 WRITE(6,999) NB,IB
  999 FORMAT('*****STOP NB=',I5,'  AT ELEMENT=',I5)
C9999 FORMAT(2X,'COMP. TIME T3=',F10.3)
C
C     CALL SRCH$$(K$CLOS,'FEMI',INTS(4),INTS(1),TYPE,CODE)
C     CALL SRCH$$(K$CLOS,'FEMO',INTS(4),INTS(2),TYPE,CODE)
C
      STOP
                                  END
C
C ********************************************************************
C *                                                                  *
                       SUBROUTINE STSTIF
C *                                                                  *
C ********************************************************************
C TO CALCULATE STRUCTURAL STIFFNESS MATRIX A(NB*ND)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/CSTRN/KSTRN(74),KSTRT(74)
      COMMON/XYZ/NNP(80,8),X(300),Y(300),Z(300)
      COMMON/AIN/NB,NC,ND,NE,NN,TH,NINT,R,S,T,DET
      COMMON/FSK/A(84530),V(1800),EK(48,48),MAXA(1801),NEIRE
C
      DIMENSION MHT(1800)
C
C FORM THE ARRAY A(NB*ND) FOR STRUCTURAL MATRIX
      NBND=NB*ND
      DO 10 I=1,NBND
10    A(I)=0.0D0
C
C SUPERPOSE THE ELEMENT STIFFNESS MATRIX EK(48,48) OF THE
C ELEMENT "NEIRE" TO THE STRUCTURAL STIFFNESS MATRIX A(NB*ND)
      DO 20 NEIRE=1,NE
C
C   COLUMN DETERMINATION
                              CALL ELSTIF
      DO 20 INC=1,8
      INOC=NNP(NEIRE,INC)
      IBC=(NNP(NEIRE,INC)-1)*6
      DO 20 IDC=1,6
      ICEL=(INC-1)*6+IDC
      ICST=IBC+IDC
      IDI=0
      IF(ICST.GT.NB)IDI=ICST-NB
      IVC=(ICST-1)*NB
C
C   ROW DETERMINATION
      DO 18 INR=1,8
      INOR=NNP(NEIRE,INR)
      IF(INOC.LT.INOR)GO TO 18
      IBR=(NNP(NEIRE,INR)-1)*6
      IDVC=IDC
```

```
      IF(INOC.GT.INOR)IDVC=6
      DO 15 IDR=1,IDVC
      IREL=(INR-1)*6+IDR
      IVV=IVC+IBR+IDR-IDI
      SS=A(IVV)+EK(IREL,ICEL)
C     IF(DABS(SS).LT.1.0D-14)SS=0.0D0
15    A(IVV)=SS
18    CONTINUE
20    CONTINUE
C
C ELIMINATE THE CONSTRAINT POINTS
C       KSTRN(N)....  N-TH CONSTRAINED NODE
C       KSTRT(N)....  TYPE OF N-TH CONSTRAIN
C                             100000  X-DISPLACEMENT  IS FIXED
C                             010000  Y-DISPLACEMENT  IS FIXED
C                             001000  Z-DISPLACEMENT  IS FIXED
C                             000100  X-ROTATION      IS FIXED
C                             000010  Y-ROTATION      IS FIXED
C                             000001  Z-ROTATION      IS FIXED
C                             100100  X-DIS. & X-ROT. IS FIXED
C                             100001  X-DIS. & Z-ROT. IS FIXED
      DO 140 N=1,NC
      IRCX=KSTRN(N)*6-5
      IRCY=KSTRN(N)*6-4
      IRCZ=KSTRN(N)*6-3
      IRCRX=KSTRN(N)*6-2
      IRCRY=KSTRN(N)*6-1
      IRCRZ=KSTRN(N)*6
      KCHK=KSTRT(N)
C
C  ELIMINATE THE X-DISPLACEMENT
      IF(KCHK.LT.100000) GO TO 40
      ICB=ND-IRCX+1
      IF(ICB.GT.NB)ICB=NB
      DO 35 I=1,ICB
      IDI=0
      IRCXX=IRCX+I-1
      IF(IRCXX.GT.NB)IDI=IRCXX-NB
      IXV=(IRCX-2+I)*NB+IRCX-IDI
      IF(I.EQ.1)GO TO 25
      A(IXV)=0.0D0
      GO TO 35
25    A(IXV)=1.0D0
      IF(IRCX.EQ.1)GO TO 35
      DO 30 J=1,IRCX-IDI-1
      IXXV=IXV-J
30    A(IXXV)=0.0D0
35    CONTINUE
      KCHK=KCHK-100000
C
C  ELIMINATE THE Y-DISPLACEMENT
40    IF(KCHK.LT.010000) GO TO 60
      ICB=ND-IRCY+1
      IF(ICB.GT.NB)ICB=NB
      DO 55 I=1,ICB
      IDI=0
      IRCYY=IRCY+I-1
      IF(IRCYY.GT.NB)IDI=IRCYY-NB
      IYV=(IRCY-2+I)*NB+IRCY-IDI
      IF(I.EQ.1)GO TO 45
      A(IYV)=0.0D0
      GO TO 55
45    A(IYV)=1.0D0
      IF(IRCY.EQ.1)GO TO 55
      DO 50 J=1,IRCY-IDI-1
      IYYV=IYV-J
```

```
50     A(IYYV)=0.0D0
55     CONTINUE
       KCHK=KCHK-10000
C
C ELIMINATE THE Z-DISPLACEMENT
60     IF(KCHK.LT.001000) GO TO 80
       ICB=ND-IRCZ+1
       IF(ICB.GT.NB)ICB=NB
       DO 75 I=1,ICB
       IDI=0
       IRCZZ=IRCZ+I-1
       IF(IRCZZ.GT.NB)IDI=IRCZZ-NB
       IZV=(IRCZ-2+I)*NB+IRCZ-IDI
       IF(I.EQ.1)GO TO 65
       A(IZV)=0.0D0
       GO TO 75
65     A(IZV)=1.0D0
       IF(IRCZ.EQ.1)GO TO 75
       DO 70 J=1,IRCZ-IDI-1
       IZZV=IZV-J
70     A(IZZV)=0.0D0
75     CONTINUE
       KCHK=KCHK-1000
C
C  ELIMINATE THE X-ROTATION
80     IF(KCHK.LT.000100) GO TO 100
       ICB=ND-IRCRX+1
       IF(ICB.GT.NB)ICB=NB
       DO 95 I=1,ICB
       IDI=0
       IRCRXX=IRCRX+I-1
       IF(IRCRXX.GT.NB)IDI=IRCRXX-NB
       IRXV=(IRCRX-2+I)*NB+IRCRX-IDI
       IF(I.EQ.1)GO TO 85
       A(IRXV)=0.0D0
       GO TO 95
85     A(IRXV)=1.0D0
       IF(IRCRX.EQ.1)GO TO 95
       DO 90 J=1,IRCRX-IDI-1
       IRXXV=IRXV-J
90     A(IRXXV)=0.0D0
95     CONTINUE
       KCHK=KCHK-100
C
C  ELIMINATE THE Y-ROTATION
100    IF(KCHK.LT.000010) GO TO 120
       ICB=ND-IRCRY+1
       IF(ICB.GT.NB)ICB=NB
       DO 115 I=1,ICB
       IDI=0
       IRCRYY=IRCRY+I-1
       IF(IRCRYY.GT.NB)IDI=IRCRYY-NB
       IRYV=(IRCRY-2+I)*NB+IRCRY-IDI
       IF(I.EQ.1)GO TO 105
       A(IRYV)=0.0D0
       GO TO 115
105    A(IRYV)=1.0D0
       IF(IRCRY.EQ.1)GO TO 115
       DO 110 J=1,IRCRY-IDI-1
       IRYYV=IRYV-J
110    A(IRYYV)=0.0D0
115     CONTINUE
       KCHK=KCHK-10
C
C ELIMINATE THE Z-ROTATION
120    IF(KCHK.LT.000001) GO TO 140
```

```
      ICB=ND-IRCRZ+1
      IF(ICB.GT.NB)ICB=NB
      DO 135 I=1,ICB
      IDI=0
      IRCRZZ=IRCRZ+I-1
      IF(IRCRZZ.GT.NB)IDI=IRCRZZ-NB
      IRZV=(IRCRZ-2+I)*NB+IRCRZ-IDI
      IF(I.EQ.1)GO TO 125
      A(IRZV)=0.0D0
      GO TO 135
125   A(IRZV)=1.0D0
      IF(IRCRZ.EQ.1)GO TO 135
      DO 130 J=1,IRCRZ-IDI-1
      IRZZV=IRZV-J
130   A(IRZZV)=0.0D0
135   CONTINUE
140   CONTINUE
C
C    TO CALCULATE COLUMN HEIGHTS
C
      DO 160 I=1,ND
      IDI=0
      IF(I.GT.NB)IDI=I-NB
      IIV=(I-1)*NB
      DO 150 J=1,I
      IF(A(IIV+J).EQ.0.0D0)GO TO 150
      MHT(I)=I-J-IDI
      GO TO 160
150   CONTINUE
160   CONTINUE
C
C     PROGRAM
C        TO CALCULATE ADDRESSES OF DIAGONAL ELEMENTS IN
C        BANDED MATRIX WHOSE COLUMN HEIGHTS ARE KNOWN
C
C        MHT  = ACTIVE COLUMN HEIGHTS
C        MAXA = ADDRESSES OF DIAGONAL ELEMENTS
C
C
C
C     CLEAR ARRAY MAXA
C
      NM=ND+1
      DO 170 I=1,NM
170   MAXA(I)=0
C
      MAXA(1)=1
      MAXA(2)=2
      IF(ND.EQ.1)GO TO 190
      DO 180 I=2,ND
180   MAXA(I+1)=MAXA(I)+MHT(I)+1
190   NWK=MAXA(ND+1)-MAXA(1)
C
C    TO STORE STIFFNESS MATRIX A(NB*ND) IN COMPACTED FORM A(NWK)
C
      IAN=0
      DO 200 I=1,ND
      ICK=MAXA(I+1)-MAXA(I)
      IDI=0
      IF(I.GT.NB)IDI=I-NB
      INBB=(I-1)*NB+I-IDI
      DO 200 II=1,ICK
      IAN=IAN+1
      IAV=INBB-II+1
      A(IAN)=A(IAV)
200   CONTINUE
```

```
C
C   CLEAR THE REST OF A(NB*ND) ARRAY
C
      DO 210 I=NWK+1,NBND
210   A(I)=0.0D0
      RETURN
                                        END
C
C ***********************************************************************
C *                                                                     *
                        SUBROUTINE ELSTIF
C *                                                                     *
C ***********************************************************************
C TO CALCULATE ELEMENT STIFFNESS MATRIX
C        EK(48,48) ... ELEMENT STIFFNES MATRIX FOR
C        NEIRE     ... ELEMENT NUMBER
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/FSK/A(84530),V(1800),EK(48,48),MAXA(1801),NEIRE
      COMMON/AIN/NB,NC,ND,NE,NN,TH,NINT,R,S,T,DET
      COMMON/B01/B0(80,9,48),B1(80,9,48),INXY(80),ID(300)
      COMMON/XYZ/NNP(80,8),X(300),Y(300),Z(300)
      COMMON/MAT/D0(9,9),D1(9,9)
C
      DIMENSION XG(4,4),WGT(4,4),D0B0(9),D1B1(9)
C
C     GAUSS-LEGENDRE SAMPLING POINT
C
      DATA XG/0.0D0,0.0D0,0.0D0,0.0D0,-.5773502691896D0,
     *.5773502691896D0,0.0D0,0.0D0,-.7745966692415D0,
     *0.0D0,.7745966692415D0,0.0D0,-.8611363115941D0,
     *-.3399810435849D0,.3399810435849D0,.8611363115941D0/
C
C     GAUSS-LEGENDRE WEIGHTING FACTORS
C
      DATA WGT/2.0D0,0.0D0,0.0D0,0.0D0,1.0D0,1.0D0,0.0D0,
     *0.0D0,.5555555555556D0,.8888888888889D0,
     *.5555555555556D0,0.0D0,.3478548451375D0,
     *.6521451548625D0,.6521451548625D0,
     *.3478548451375D0/
C
C     CALCULATION OF ELEMENT STIFFNESS MATRIX
C
      DO 30 I=1,48
      DO 30 J=1,48
30    EK(I,J)=0.0D0
      DO 80 LX=1,NINT
      R=XG(LX,NINT)
      DO 80 LY=1,NINT
      S=XG(LY,NINT)
      DO 80 LZ=1,NINT
      T=XG(LZ,NINT)
C
C     CALCULATION OF DERIVATIVE OPERATOR B0, B1
C     AND THE JACOBIAN DETERMINANT DET
C
      CALL STDM
C
      WT=WGT(LX,NINT)*WGT(LY,NINT)*WGT(LZ,NINT)*DET
      DO 70 J=1,48
      DO 40 K=1,9
      D0B0(K)=0.0D0
      D1B1(K)=0.0D0
      DO 40 L=1,9
      D0B0(K)=D0B0(K)+D0(K,L)*B0(NEIRE,L,J)
40    D1B1(K)=D1B1(K)+D1(K,L)*B1(NEIRE,L,J)
```

```
      DO 60 I=J,48
      STIFF=0.0D0
      DO 50 L=1,9
50    STIFF=STIFF+B0(NEIRE,L,I)*D0B0(L)+B1(NEIRE,L,I)*D1B1(L)
60    EK(I,J)=EK(I,J)+STIFF*WT
70    CONTINUE
80    CONTINUE
C
      DO 90 J=1,48
      DO 90 I=J,48
90    EK(J,I)=EK(I,J)
C
C CONSTRUCT B0 AND B1-MATRIX FOR LATER USE IN SUBROUTINE STRESS
C
C
C CHECK THE LOCATION OF INTEREST FOR OUTPUT
      IF(INXY(NEIRE).EQ.00001)GO TO 100
      IF(INXY(NEIRE).EQ.00010)GO TO 110
      IF(INXY(NEIRE).EQ.00011)GO TO 120
      IF(INXY(NEIRE).EQ.00100)GO TO 130
      IF(INXY(NEIRE).EQ.00101)GO TO 140
      IF(INXY(NEIRE).EQ.00110)GO TO 150
      IF(INXY(NEIRE).EQ.00111)GO TO 160
      IF(INXY(NEIRE).EQ.01000)GO TO 170
      IF(INXY(NEIRE).EQ.01001)GO TO 180
      IF(INXY(NEIRE).EQ.01010)GO TO 190
      IF(INXY(NEIRE).EQ.01011)GO TO 200
      IF(INXY(NEIRE).EQ.01100)GO TO 210
      IF(INXY(NEIRE).EQ.01101)GO TO 220
      IF(INXY(NEIRE).EQ.01110)GO TO 230
      IF(INXY(NEIRE).EQ.01111)GO TO 240
      IF(INXY(NEIRE).EQ.10000)GO TO 250
      IF(INXY(NEIRE).EQ.10001)GO TO 260
      IF(INXY(NEIRE).EQ.10010)GO TO 270
      IF(INXY(NEIRE).EQ.10011)GO TO 280
      IF(INXY(NEIRE).EQ.10100)GO TO 290
      R= 0.0D0
      S= 0.0D0
      T= 0.0D0
      GO TO 300
100   R= 1.0D0
      S= 1.0D0
      T= 1.0D0
      GO TO 300
110   R=-1.0D0
      S= 1.0D0
      T= 1.0D0
      GO TO 300
120   R=-1.0D0
      S=-1.0D0
      T= 1.0D0
      GO TO 300
130   R= 1.0D0
      S=-1.0D0
      T= 1.0D0
      GO TO 300
140   R= 1.0D0
      S= 1.0D0
      T=-1.0D0
      GO TO 300
150   R=-1.0D0
      S= 1.0D0
      T=-1.0D0
      GO TO 300
160   R=-1.0D0
      S=-1.0D0
```

```
      T=-1.0D0
      GO TO 300
170   R= 1.0D0
      S=-1.0D0
      T=-1.0D0
      GO TO 300
180   R= 0.0D0
      S= 1.0D0
      T= 1.0D0
      GO TO 300
190   R=-1.0D0
      S= 0.0D0
      T= 1.0D0
      GO TO 300
200   R= 0.0D0
      S=-1.0D0
      T= 1.0D0
      GO TO 300
210   R= 1.0D0
      S= 0.0D0
      T= 1.0D0
      GO TO 300
220   R= 0.0D0
      S= 1.0D0
      T=-1.0D0
      GO TO 300
230   R=-1.0D0
      S= 0.0D0
      T=-1.0D0
      GO TO 300
240   R= 0.0D0
      S=-1.0D0
      T=-1.0D0
      GO TO 300
250   R= 1.0D0
      S= 0.0D0
      T=-1.0D0
      GO TO 300
260   R= 1.0D0
      S= 1.0D0
      T= 0.0D0
      GO TO 300
270   R=-1.0D0
      S= 1.0D0
      T= 0.0D0
      GO TO 300
280   R=-1.0D0
      S=-1.0D0
      T= 0.0D0
      GO TO 300
290   R= 1.0D0
      S=-1.0D0
      T= 0.0D0
C
300   CALL STDM
C
      RETURN
C
      END
C
C ************************************************************************
C *                                                                      *
                         SUBROUTINE STDM
C *                                                                      *
C ************************************************************************
C TO EVALUATE THE STRAIN-DISPLACEMENT TRANSFORMATION MATRIX
```

```
C B0 & B1 AT POINT (R,S,T) FOR A PARALLELEPIPED ELEMENT
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/FSK/A(84530),V(1800),EK(48,48),MAXA(1801),NEIRE
      COMMON/AIN/NB,NC,ND,NE,NN,TH,NINT,R,S,T,DET
      COMMON/B01/B0(80,9,48),B1(80,9,48),INXY(80),ID(300)
      COMMON/XYZ/NNP(80,8),X(300),Y(300),Z(300)
C
      DIMENSION H(8),P(8,3),XYZ(8,3),XJ(3,3),XJI(3,3)
C
C     DETERMINE THE NODAL POINTS OF THE ELEMENT NEIRE
C
       I=NNP(NEIRE,1)
       J=NNP(NEIRE,2)
       K=NNP(NEIRE,3)
       L=NNP(NEIRE,4)
      IJ=NNP(NEIRE,5)
      JK=NNP(NEIRE,6)
      KL=NNP(NEIRE,7)
      LI=NNP(NEIRE,8)
C
      XYZ(1,1)=X( I)
      XYZ(2,1)=X( J)
      XYZ(3,1)=X( K)
      XYZ(4,1)=X( L)
      XYZ(5,1)=X(IJ)
      XYZ(6,1)=X(JK)
      XYZ(7,1)=X(KL)
      XYZ(8,1)=X(LI)
C
      XYZ(1,2)=Y( I)
      XYZ(2,2)=Y( J)
      XYZ(3,2)=Y( K)
      XYZ(4,2)=Y( L)
      XYZ(5,2)=Y(IJ)
      XYZ(6,2)=Y(JK)
      XYZ(7,2)=Y(KL)
      XYZ(8,2)=Y(LI)
C
      XYZ(1,3)=Z( I)
      XYZ(2,3)=Z( J)
      XYZ(3,3)=Z( K)
      XYZ(4,3)=Z( L)
      XYZ(5,3)=Z(IJ)
      XYZ(6,3)=Z(JK)
      XYZ(7,3)=Z(KL)
      XYZ(8,3)=Z(LI)
C
C
C     INTERPOLATION FUNCTION
C
C
      RP=1.0D0+R
      RM=1.0D0-R
      SP=1.0D0+S
      SM=1.0D0-S
      TP=1.0D0+T
      TM=1.0D0-T
C
C
      H(1)=  0.125D0*RP*SP*TP
      H(2)=  0.125D0*RM*SP*TP
      H(3)=  0.125D0*RM*SM*TP
      H(4)=  0.125D0*RP*SM*TP
      H(5)=  0.125D0*RP*SP*TM
      H(6)=  0.125D0*RM*SP*TM
      H(7)=  0.125D0*RM*SM*TM
```

```
      H(8)=  0.125D0*RP*SM*TM
C
C
C
C     NATURAL COORDINATE DERIVATIVES OF THE INTERPOLATION
C     FUNCTIONS
C
C
C     1. WITH RESPECT TO R
C
      P( 1,1)=  0.125D0*SP*TP
      P( 2,1)= -0.125D0*SP*TP
      P( 3,1)= -0.125D0*SM*TP
      P( 4,1)=  0.125D0*SM*TP
      P( 5,1)=  0.125D0*SP*TM
      P( 6,1)= -0.125D0*SP*TM
      P( 7,1)= -0.125D0*SM*TM
      P( 8,1)=  0.125D0*SM*TM
C
C
C     2.WITH RESPECT TO  S
C
C
      P( 1,2)  =  0.125D0*RP*TP
      P( 2,2)  =  0.125D0*RM*TP
      P( 3,2)  = -0.125D0*RM*TP
      P( 4,2)  = -0.125D0*RP*TP
      P( 5,2)  =  0.125D0*RP*TM
      P( 6,2)  =  0.125D0*RM*TM
      P( 7,2)  = -0.125D0*RM*TM
      P( 8,2)  = -0.125D0*RP*TM
C
C
C     3.WITH RESPECT TO  T
C
C
      P( 1,3)  =  0.125D0*RP*SP
      P( 2,3)  =  0.125D0*RM*SP
      P( 3,3)  =  0.125D0*RM*SM
      P( 4,3)  =  0.125D0*RP*SM
      P( 5,3)  = -0.125D0*RP*SP
      P( 6,3)  = -0.125D0*RM*SP
      P( 7,3)  = -0.125D0*RM*SM
      P( 8,3)  = -0.125D0*RP*SM
C
C
C     EVALUATE THE JACOBIAN MATRIX AT POINT (R,S,T)
C
C
      DO 30 I=1,3
      DO 30 J=1,3
      DUM= 0.0D0
      DO 20 K=1,8
20    DUM=DUM+P(K,I)*XYZ(K,J)
30    XJ (I,J)=DUM
C
C
C     COMPUTE THE DETERMINAT OF THE JACOBIAN MATRIX
C
C
      DET=XJ(1,1)*XJ(2,2)*XJ(3,3)+XJ(1,3)*XJ(2,1)*XJ(3,2)
     *+XJ(1,2)*XJ(2,3)*XJ(3,1)
     *-XJ(1,3)*XJ(2,2)*XJ(3,1)-XJ(1,2)*XJ(2,1)*XJ(3,3)
     *-XJ(1,1)*XJ(2,3)*XJ(3,2)
      IF(DET.GT.1.0D-14) GO TO 40
      WRITE(6,1000)DET
```

```
1000  FORMAT(5X,'DET= ',E14.7//)
      WRITE (6,2000) NEIRE
2000  FORMAT(3X,'*** ERROR, ZERO OR NEGATIVE JACOBIAN
     * DETERMINANT AT ELEMENT=',I4)
      STOP
C
C
C     COMPUTE INVERSE OF THE JACOBIAN MATRIX
C
C
40    DUM=1.0D0/DET
      XJI(1,1)= (XJ(2,2)*XJ(3,3)-XJ(2,3)*XJ(3,2))*DUM
      XJI(2,1)=-(XJ(2,1)*XJ(3,3)-XJ(2,3)*XJ(3,1))*DUM
      XJI(3,1)= (XJ(2,1)*XJ(3,2)-XJ(2,2)*XJ(3,1))*DUM
      XJI(1,2)=-(XJ(1,2)*XJ(3,3)-XJ(1,3)*XJ(3,2))*DUM
      XJI(2,2)= (XJ(1,1)*XJ(3,3)-XJ(1,3)*XJ(3,1))*DUM
      XJI(3,2)=-(XJ(1,1)*XJ(3,2)-XJ(1,2)*XJ(3,1))*DUM
      XJI(1,3)= (XJ(1,2)*XJ(2,3)-XJ(1,3)*XJ(2,2))*DUM
      XJI(2,3)=-(XJ(1,1)*XJ(2,3)-XJ(1,3)*XJ(2,1))*DUM
      XJI(3,3)= (XJ(1,1)*XJ(2,2)-XJ(1,2)*XJ(2,1))*DUM
C
C
C     EVALUATE GLOBAL DERIVATIVE OPERATOR B0 & B1
C
C
      DO 50 I=1,9
      DO 50 J=1,48
      B0(NEIRE,I,J)=0.0D0
50    B1(NEIRE,I,J)=0.0D0
      K6=0
      DO 60 K=1,8
      K6=K6+6
      B0(NEIRE,4,K6  )=-H(K)
      B0(NEIRE,5,K6  )= H(K)
      B0(NEIRE,6,K6-1)= H(K)
      B0(NEIRE,7,K6-1)=-H(K)
      B0(NEIRE,8,K6-2)=-H(K)
      B0(NEIRE,9,K6-2)= H(K)
      DO 60 I=1,3
      B0(NEIRE,1,K6-5)=B0(NEIRE,1,K6-5)+XJI(1,I)*P(K,I)
      B0(NEIRE,2,K6-4)=B0(NEIRE,2,K6-4)+XJI(2,I)*P(K,I)
      B0(NEIRE,3,K6-3)=B0(NEIRE,3,K6-3)+XJI(3,I)*P(K,I)
      B0(NEIRE,4,K6-4)=B0(NEIRE,4,K6-4)+XJI(1,I)*P(K,I)
      B0(NEIRE,5,K6-5)=B0(NEIRE,5,K6-5)+XJI(2,I)*P(K,I)
      B0(NEIRE,6,K6-3)=B0(NEIRE,6,K6-3)+XJI(1,I)*P(K,I)
      B0(NEIRE,7,K6-5)=B0(NEIRE,7,K6-5)+XJI(3,I)*P(K,I)
      B0(NEIRE,8,K6-3)=B0(NEIRE,8,K6-3)+XJI(2,I)*P(K,I)
      B0(NEIRE,9,K6-4)=B0(NEIRE,9,K6-4)+XJI(3,I)*P(K,I)
      B1(NEIRE,1,K6-2)=B1(NEIRE,1,K6-2)+XJI(1,I)*P(K,I)
      B1(NEIRE,2,K6-1)=B1(NEIRE,2,K6-1)+XJI(2,I)*P(K,I)
      B1(NEIRE,3,K6  )=B1(NEIRE,3,K6  )+XJI(3,I)*P(K,I)
      B1(NEIRE,4,K6-2)=B1(NEIRE,4,K6-2)+XJI(2,I)*P(K,I)
      B1(NEIRE,5,K6-1)=B1(NEIRE,5,K6-1)+XJI(1,I)*P(K,I)
      B1(NEIRE,6,K6-2)=B1(NEIRE,6,K6-2)+XJI(3,I)*P(K,I)
      B1(NEIRE,7,K6  )=B1(NEIRE,7,K6  )+XJI(1,I)*P(K,I)
      B1(NEIRE,8,K6-1)=B1(NEIRE,8,K6-1)+XJI(3,I)*P(K,I)
60    B1(NEIRE,9,K6  )=B1(NEIRE,9,K6  )+XJI(2,I)*P(K,I)
C
      RETURN
C
      END
C
C **********************************************************************
C *                                                                    *
                      SUBROUTINE LOADER
C *                                                                    *
```

```
C ********************************************************************
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/AIN/NB,NC,ND,NE,NN,TH,NINT,R,S,T,DET
      COMMON/FSK/A(84530),V(1800),EK(48,48),MAXA(1801),NEIRE
C
C SET THE EXTERNAL LOAD TO BE ZERO FOR EACH DEGREE OF FREEDOM
      DO 10 LO=1,ND
   10 V(LO)=0.0D0
C
C LOAD THE STRUCTURE
      SCALE=1.0D0
       V(51)=-100.0d0*SCALE
C PRINT THE EXTERNAL LOAD
      WRITE(6,1000)
 1000 FORMAT(//34X,'**** EXTERNAL LOAD ****')
      WRITE(6,2000)
 2000 FORMAT(/1X,'NODE',8X,'X-FORCE',8X,'Y-FORCE',8X,'Z-FORCE',
     *          8X,'X-MOMNT',8X,'Y-MOMNT',8X,'Z-MOMNT')
      IND=0
      DO 20 N=1,ND,6
      IND=IND+1
      CHECK= DABS(V(N))+DABS(V(N+1))+DABS(V(N+2))
     1       +DABS(V(N+3))+DABS(V(N+4))+DABS(V(N+5))
      IF(CHECK .EQ. 0.0D0) GO TO 20
      WRITE(6,3000) (IND,V(N),V(N+1),V(N+2),
     *                V(N+3),V(N+4),V(N+5))
   20 CONTINUE
 3000 FORMAT(2X,I3,6(5X,E10.3))
C
      RETURN
                                        END
C
C ********************************************************************
C *                                                                  *
                              SUBROUTINE COLSOL
C *                                                                  *
C ********************************************************************
C*********************************************************************
C*                                                                   *
C*    PROGRAM                                                        *
C*       TO SOLVE FINITE ELEMENT STATIC EQUILIBRIUM EQUATIONS IN     *
C*       CORE, USING COMPACTED STORAGE AND COLUMN REDUCTION SCHEME   *
C*                                                                   *
C*    --INPUT VARIABLES--                                            *
C*       A(NWK)      = STIFFNESS MATRIX STORED IN COMPACTED FORM     *
C*       V(ND)       = RIGHT-HAND-SIDE LOAD VECTOR                   *
C*       MAXA(ND+1)  = VECTOR CONTAINING ADDRESSES OF DIAGONAL       *
C*                     ELEMENTS OF STIFFNESS MATRIX IN A             *
C*       ND          = NUMBER OF EQUATIONS                           *
C*       NWK         = NUMBER OF ELEMENTS BELOW SKYLINE OF MATRIX    *
C*                                                                   *
C*    --OUTPUT--                                                     *
C*       A(NWK)      = D AND L - FACTORS OF STIFFNESS MATRIX         *
C*       V(ND)       = DISPLACEMENT VECTOR                           *
C*                                                                   *
C*********************************************************************
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/AIN/NB,NC,ND,NE,NN,TH,NINT,R,S,T,DET
      COMMON/FSK/A(84530),V(1800),EK(48,48),MAXA(1801),NEIRE
C
C     PERFORM L*D*L(T) FACTORIZATION OF STIFFNESS MATRIX
C
      DO 140 N=1,ND
      KN=MAXA(N)
      KL=KN+1
      KU=MAXA(N+1)-1
```

```
      KH=KU-KL
      IF(KH)110,90,50
50    K=N-KH
      IC=0
      KLT=KU
      DO 80 J=1,KH
      IC=IC+1
      KLT=KLT-1
      KI=MAXA(K)
      NND=MAXA(K+1)-KI-1
      IF(NND)80,80,60
60    KK=MIN0(IC,NND)
      C=0.0D0
      DO 70 L=1,KK
70    C=C+A(KI+L)*A(KLT+L)
      A(KLT)=A(KLT)-C
80    K=K+1
90    K=N
      B=0.0D0
      DO 100 KK=KL,KU
      K=K-1
      KI=MAXA(K)
      C=A(KK)/A(KI)
      B=B+C*A(KK)
100   A(KK)=C
      A(KN)=A(KN)-B
110   IF(A(KN))120,120,140
120   WRITE(6,2000)N,A(KN)
      STOP
140   CONTINUE
C
C     REDUCE RIGHT-HAND-SIDE LOAD VECTOR
C
      DO 180 N=1,ND
      KL=MAXA(N)+1
      KU=MAXA(N+1)-1
      IF(KU-KL)180,160,160
160   K=N
      C=0.0D0
      DO 170 KK=KL,KU
      K=K-1
170   C=C+A(KK)*V(K)
      V(N)=V(N)-C
180   CONTINUE
C
C     BACK-SUBSTITUTE
C
      DO 200 N=1,ND
      K=MAXA(N)
200   V(N)=V(N)/A(K)
      IF(ND.EQ.1)RETURN
      N=ND
      DO 230 L=2,ND
      KL=MAXA(N)+1
      KU=MAXA(N+1)-1
      IF(KU-KL)230,210,210
210   K=N
      DO 220 KK=KL,KU
      K=K-1
220   V(K)=V(K)-A(KK)*V(N)
230   N=N-1
      DO 240 I=1,ND
 240  CONTINUE
      WRITE(6,1000)
 1000 FORMAT(//33X,'*** NODAL DISPLACEMENT ***')
      WRITE(6,1500)
```

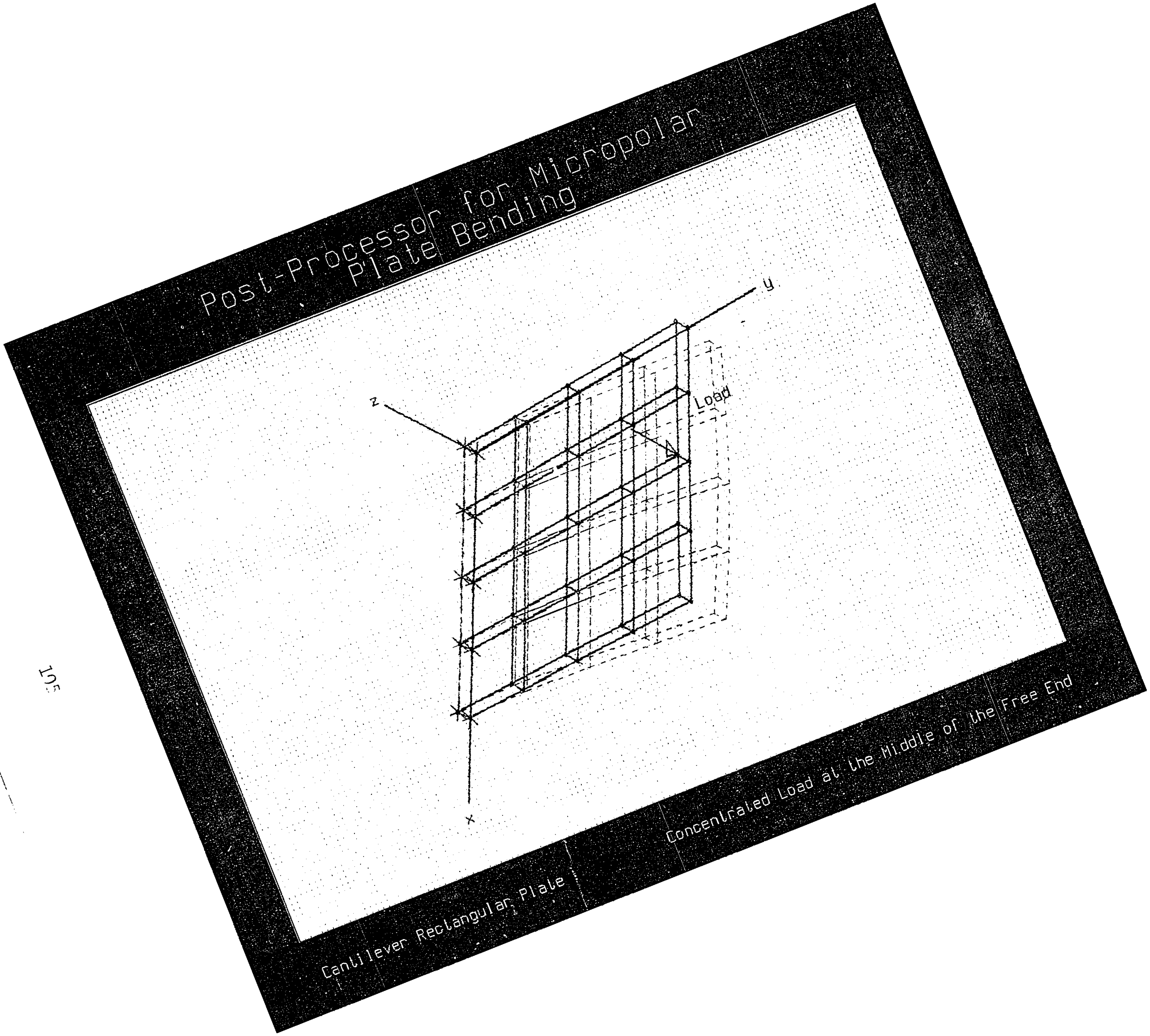
```
 1500 FORMAT(/1X,'NODE',8X,'X-DISP',13X,'Y-DISP',13X,'Z-DISP',
     *          13X,'X-ROTN',13X,'Y-ROTN',13X,'Z-ROTN')
      IND=0
      DO 250 K=1,ND,6
      IND=IND+1
      WRITE(6,3000) IND,V(K),V(K+1),V(K+2),
     *                 V(K+3),V(K+4),V(K+5)
      WRITE(60,3010)IND,V(K),V(K+1),V(K+2)
250   CONTINUE
      RETURN
2000  FORMAT(//48H STOP - STIFFNESS MATRIX NOT POSITIVE DEFINITE  ,//
     *            32H NONPOSITIVE PIVOT FOR EQUATION ,I4,//
     *            10H PIVOT =  ,E20.12)
3000  FORMAT(2X,I3,6(5X,E14.7))
3010  FORMAT(2X,I3,3(5X,E10.3))
      END
C
C ********************************************************************
C *                                                                  *
                        SUBROUTINE STRESS
C *                                                                  *
C ********************************************************************
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/B01/B0(80,9,48),B1(80,9,48),INXY(80),ID(300)
      COMMON/XYZ/NNP(80,8),X(300),Y(300),Z(300)
      COMMON/MAT/D0(9,9),D1(9,9)
      COMMON/AIN/NB,NC,ND,NE,NN,TH,NINT,R,S,T,DET
      COMMON/FSK/A(84530),V(1800),EK(48,48),MAXA(1801),NEIRE
C
      DIMENSION EDISP(48),E(80,9),PHIJ(80,9),T(80,9),CM(80,9),U(80)
C
C PICK UP ELEMENT NODAL DISPLACEMENTS EDISP(48)
C FROM STRUCTURAL NODAL DISPLACEMENTS V(ND)
C
C FOR EACH ELEMENT "IE"
      DO 300 IE=1,NE
       DO 20 IJM=1,8
       IEB=(IJM-1)*6
       ISB=(NNP(IE,IJM)-1)*6
        DO 20 IDOF=1,6
   20   EDISP(IEB+IDOF)=V(ISB+IDOF)
C
C DISPLACEMENT STRAIN : EIJ=B0*UE
       DO 40 IC=1,9
       SUM=0.0D0
        DO 30 K=1,48
   30   SUM=SUM+B0(IE,IC,K)*EDISP(K)
   40  E(IE,IC)=SUM
C
C ROTATION STRAIN : PHI,J=B1*UE
       DO 60 IC=1,9
       SUM=0.0D0
        DO 50 K=1,48
   50   SUM=SUM+B1(IE,IC,K)*EDISP(K)
   60  PHIJ(IE,IC)=SUM
C
C FORCE STRESS : TIJ=D0*EIJ
       DO 80 IC=1,9
       SUM=0.0D0
        DO 70 K=1,9
   70   SUM=SUM+D0(IC,K)*E(IE,K)
   80  T(IE,IC)=SUM
C
C COUPLE STRESS : MIJ=D1*PHIJ
       DO 100 IC=1,9
       SUM=0.0D0
```

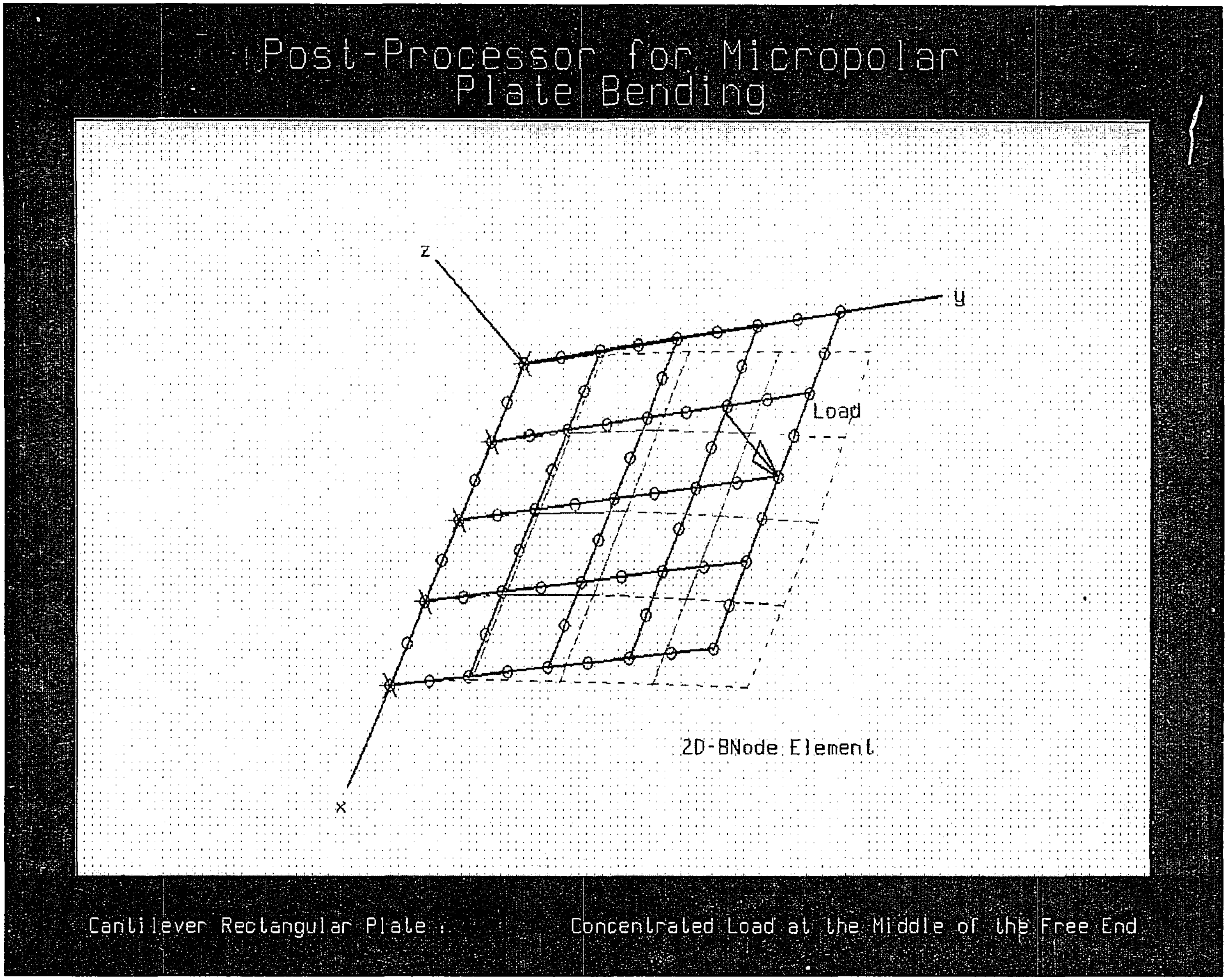
```
          DO 90 K=1,9
    90    SUM=SUM+D1(IC,K)*PHIJ(IE,K)
   100   CM(IE,IC)=SUM
C STRAIN-ENERGY : U
          SUM=0.0D00
          DO 200 IC=1,9
   200    SUM=SUM+E(IE,IC)*T(IE,IC)+PHIJ(IE,IC)*CM(IE,IC)
         U(IE)=SUM*0.50D00
   300  CONTINUE
C
C WRITE THE CALCULATED STRAINS AND STRESSES
        WRITE(6,1000)
  1000  FORMAT(//19X,'**** STRESSES & STRAINS CALCULATED ****')
        WRITE(6,2000)
  2000  FORMAT(/1X,'ELMT',1X,'COMP',1X,'DISP-STRAN',3X,'FORCE-STRS'
     *        ,2X,'COMP',1X,'ROTATN-GRAD',2X,'COUPLE-STRS'
     *        ,2X,'STRN-ENEGY')
        WRITE(6,3000)
  3000  FORMAT(15X,'E',12X,'T',15X,'PHI,J',9X,'M',12X,'U')
        DO 400 IE=1,NE
        WRITE(6,4000) IE
  4000  FORMAT(I5)
        WRITE(6,5000) E(IE,1),T(IE,1),PHIJ(IE,1),CM(IE,1),U(IE)
  5000  FORMAT(7X,'XX',1X,1PE11.4,2X,1PE11.4,2X,'X,X',2X,1PE11.4,
     *        2X,1PE11.4,1X,1PE11.4)
        WRITE(6,6000) E(IE,2),T(IE,2),PHIJ(IE,2),CM(IE,2)
  6000  FORMAT(7X,'YY',1X,1PE11.4,2X,1PE11.4,2X,'Y,Y',2X,1PE11.4,
     *        2X,1PE11.4)
        WRITE(6,7000) E(IE,3),T(IE,3),PHIJ(IE,3),CM(IE,3)
  7000  FORMAT(7X,'ZZ',1X,1PE11.4,2X,1PE11.4,2X,'Z,Z',2X,1PE11.4,
     *        2X,1PE11.4)
        WRITE(6,8000) E(IE,4),T(IE,4),PHIJ(IE,4),CM(IE,4)
  8000  FORMAT(7X,'XY',1X,1PE11.4,2X,1PE11.4,2X,'X,Y',2X,1PE11.4,
     *        2X,1PE11.4)
        WRITE(6,9000) E(IE,5),T(IE,5),PHIJ(IE,5),CM(IE,5)
  9000  FORMAT(7X,'YX',1X,1PE11.4,2X,1PE11.4,2X,'Y,X',2X,1PE11.4,
     *        2X,1PE11.4)
        WRITE(6,10000) E(IE,6),T(IE,6),PHIJ(IE,6),CM(IE,6)
 10000  FORMAT(7X,'XZ',1X,1PE11.4,2X,1PE11.4,2X,'X,Z',2X,1PE11.4,
     *        2X,1PE11.4)
        WRITE(6,11000) E(IE,7),T(IE,7),PHIJ(IE,7),CM(IE,7)
 11000  FORMAT(7X,'ZX',1X,1PE11.4,2X,1PE11.4,2X,'Z,X',2X,1PE11.4,
     *        2X,1PE11.4)
        WRITE(6,12000) E(IE,8),T(IE,8),PHIJ(IE,8),CM(IE,8)
 12000  FORMAT(7X,'YZ',1X,1PE11.4,2X,1PE11.4,2X,'Y,Z',2X,1PE11.4,
     *        2X,1PE11.4)
        WRITE(6,13000) E(IE,9),T(IE,9),PHIJ(IE,9),CM(IE,9)
 13000  FORMAT(7X,'ZY',1X,1PE11.4,2X,1PE11.4,2X,'Z,Y',2X,1PE11.4,
     *        2X,1PE11.4)
   400  CONTINUE
        RETURN
                                        END
```

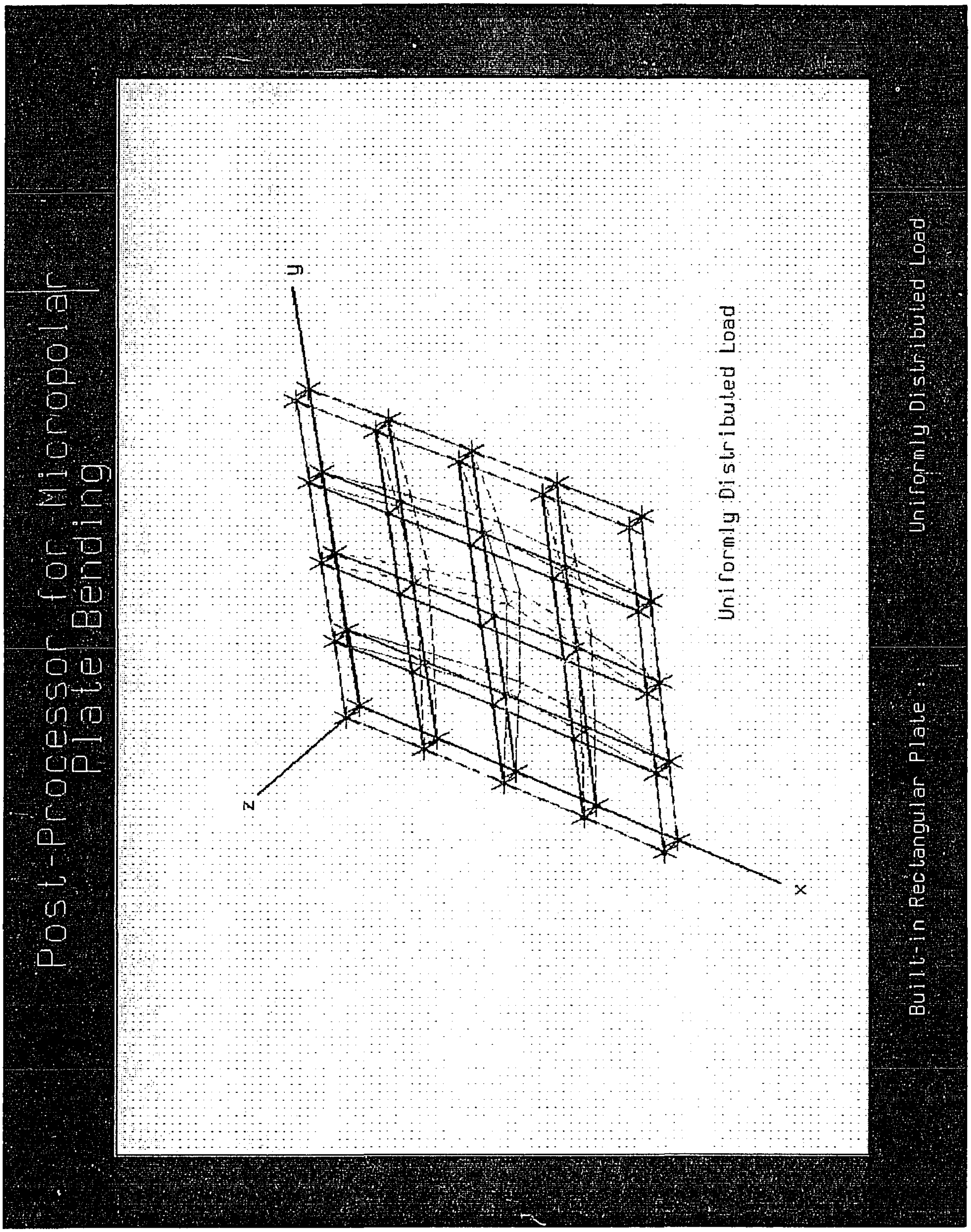
```
#include<hf.h>
C ****************************************************
C *                                                  *
C *  Post-Processor for Micropolar Plate Bending     *
C *            Finite Element Analysis               *
C *                                                  *
C *                      by                          *
C *               Vallanore K. Suresh                *
C *                                                  *
C ****************************************************
C
C
C This program produces the original finite element mesh
C and the displaced configuration for the bending analysis
C of micropolar plates. It uses HOOPS subroutines and
C the graphical output is on a SUN 3/60 workstation.
C The hard copy is produced by a TEKTRONIX 4693D printer.
C
C
      REAL XD(100),YD(100),ZD(100),X(100),Y(100),Z(100)
      DIMENSION NNP(16,8),KSTRN(100),KSTRT(100)
      OPEN(UNIT=50,FILE='ple16_3d_pp',FORM='FORMATTED',STATUS='OLD')
      CALL HF_OPEN_SEGMENT("?PICTURE")
      CALL HF_SET_WINDOW(-1.0,1.0,-1.0,1.0)
      CALL HF_OPEN_SEGMENT("top")
      CALL HF_SET_WINDOW(-1.0,1.0,0.8,1.0)
      CALL HF_SET_COLOR("WINDOWS=blue,LINES=white,TEXT=white")
      CALL HF_INSERT_TEXT(0.0,0.4,0.0,"Post-Processor for Micropolar")
      CALL HF_INSERT_TEXT(0.0,-0.4,0.0,"Plate Bending")
      CALL HF_CLOSE_SEGMENT()
      CALL HF_OPEN_SEGMENT("bottom")
      CALL HF_SET_WINDOW(-1.0,1.0,-1.0,-0.8)
      CALL HF_SET_COLOR("WINDOWS=blue,LINES=white,TEXT=white")
      CALL HF_INSERT_TEXT(0.0,0.0,0.0,"Developed by Vallanore K. Suresh")
      CALL HF_CLOSE_SEGMENT()
      CALL HF_OPEN_SEGMENT("left")
      CALL HF_SET_WINDOW(-1.0,-0.9,-0.8,0.8)
      CALL HF_SET_COLOR("WINDOWS=violet")
      CALL HF_CLOSE_SEGMENT()
      CALL HF_OPEN_SEGMENT("right")
      CALL HF_SET_WINDOW(0.9,1.0,-0.8,0.8)
      CALL HF_SET_COLOR("WINDOWS=violet")
      CALL HF_CLOSE_SEGMENT()
      CALL HF_OPEN_SEGMENT ("main")
      CALL HF_SET_WINDOW(-0.9,0.9,-0.8,0.8)
      CALL HF_SET_COLOR("LINES=blue, MARKERS=black,WINDOWS=spring green,
     *                  FACES=tan,EDGES=white")
      CALL HF_OPEN_SEGMENT ("POINTSET1")
      CALL HF_SET_LINE_WEIGHT(2.0)
      CALL HF_SET_MARKER_SIZE (.3)
      CALL HF_SET_HANDEDNESS("right")
      CALL HF_SET_CAMERA_POSITION (700.0,800.0,700.0)
      CALL HF_SET_CAMERA_TARGET (50.0,50.0,0.5)
      CALL HF_SET_CAMERA_FIELD (180.0,180.0)
      CALL HF_SET_FACE_PATTERN("\\")
      CALL HF_OPEN_SEGMENT("supports")
      CALL HF_SET_MARKER_SYMBOL("*")
      CALL HF_SET_MARKER_SIZE(1.0)
      CALL HF_SET_COLOR("LINES=black")
      CALL HF_INSERT_LINE(0.0,0.0,0.0,130.0,0.0,0.0)
      CALL HF_INSERT_LINE(50.0,100.0,5.0,50.0,100.0,25.0)
      CALL HF_INSERT_LINE(50.0,100.0,5.0,53.0,100.0,10.0)
      CALL HF_INSERT_LINE(50.0,100.0,5.0,47.0,100.0,10.0)
      CALL HF_INSERT_LINE(53.0,100.0,10.0,47.0,100.0,10.0)
      CALL HF_INSERT_TEXT(55.0,135.0,25.0,"Load")
      CALL HF_INSERT_LINE(0.0,0.0,0.0,0.0,130.0,0.0)
```

```
      CALL HF_INSERT_LINE(0.0,0.0,0.0,0.0,0.0,40.0)
      CALL HF_SET_TEXT_SIZE(0.5)
      CALL HF_INSERT_TEXT(135.0,0.0,0.0,"x")
      CALL HF_INSERT_TEXT(0.0,135.0,0.0,"y")
      CALL HF_INSERT_TEXT(0.0,0.0,45.0,"z")
      XX=0.0
      READ(50,*)NE
      DO 10 I = 1,NE
         READ(50,15) NELEM, NNP(I,1),NNP(I,2),NNP(I,3),NNP(I,4)
     *   ,NNP(I,5),NNP(I,6),NNP(I,7),NNP(I,8)
 10   CONTINUE
 15   FORMAT(2X,I4,8(1X,I3))
      READ(50,*)NON
      DO 25 N=1,NON
      READ(50,100) III,X(N),Y(N),Z(N)
      Z(N)=5.0D0 * Z(N)
 25   CONTINUE
      READ(50,*)NOC
      DO 35 I=1,NOC
      READ(50,106)KSTRN(I),KSTRT(I)
      IJK = KSTRN(I)
      CALL HF_INSERT_MARKER(X(IJK),Y(IJK),Z(IJK))
106   FORMAT(28X,I3,10X,I6)
 35   CONTINUE
      CALL HF_CLOSE_SEGMENT()
      DO 20 I = 1,NON
         READ(50,105) NN,XD(I),YD(I),ZD(I)
 20   CONTINUE
 105  FORMAT(2X,I3,3(5X,E10.3))
      DO  40 I = 1,NON
      CALL HF_INSERT_MARKER(X(I),Y(I),Z(I))
 40   CONTINUE
100   FORMAT(2x,I3,3(1X,F8.3))
      DO 110 I = 1,NE
         I1 = NNP(I,1)
         I2 = NNP(I,2)
         I3 = NNP(I,3)
         I4 = NNP(I,4)
         I5 = NNP(I,5)
         I6 = NNP(I,6)
         I7 = NNP(I,7)
         I8 = NNP(I,8)
         CALL HF_INSERT_LINE(X(I1),Y(I1),Z(I1),X(I2),Y(I2),Z(I2))
         CALL HF_INSERT_LINE(X(I2),Y(I2),Z(I2),X(I3),Y(I3),Z(I3))
         CALL HF_INSERT_LINE(X(I3),Y(I3),Z(I3),X(I4),Y(I4),Z(I4))
         CALL HF_INSERT_LINE(X(I4),Y(I4),Z(I4),X(I1),Y(I1),Z(I1))
         CALL HF_INSERT_LINE(X(I5),Y(I5),Z(I5),X(I6),Y(I6),Z(I6))
         CALL HF_INSERT_LINE(X(I6),Y(I6),Z(I6),X(I7),Y(I7),Z(I7))
         CALL HF_INSERT_LINE(X(I7),Y(I7),Z(I7),X(I8),Y(I8),Z(I8))
         CALL HF_INSERT_LINE(X(I8),Y(I8),Z(I8),X(I5),Y(I5),Z(I5))
         CALL HF_INSERT_LINE(X(I1),Y(I1),Z(I1),X(I5),Y(I5),Z(I5))
         CALL HF_INSERT_LINE(X(I2),Y(I2),Z(I2),X(I6),Y(I6),Z(I6))
         CALL HF_INSERT_LINE(X(I3),Y(I3),Z(I3),X(I7),Y(I7),Z(I7))
         CALL HF_INSERT_LINE(X(I4),Y(I4),Z(I4),X(I8),Y(I8),Z(I8))
110   CONTINUE
      DO 120 I=1,50
         X(I) = X(I) - 300.0*XD(I)
         Y(I) = Y(I) - 300.0*YD(I)
         Z(I) = Z(I) - 300.0*ZD(I)
120   CONTINUE
      CALL HF_OPEN_SEGMENT("DISPLACE")
      CALL HF_SET_LINE_PATTERN ("- - ")
      CALL HF_SET_COLOR("LINES=red")
      DO 130 I=1,NE
         I1 = NNP(I,1)
         I2 = NNP(I,2)
```

```
          I3 = NNP(I,3)
          I4 = NNP(I,4)
          I5 = NNP(I,5)
          I6 = NNP(I,6)
          I7 = NNP(I,7)
          I8 = NNP(I,8)
          CALL HF_INSERT_LINE(X(I1),Y(I1),Z(I1),X(I2),Y(I2),Z(I2))
          CALL HF_INSERT_LINE(X(I2),Y(I2),Z(I2),X(I3),Y(I3),Z(I3))
          CALL HF_INSERT_LINE(X(I3),Y(I3),Z(I3),X(I4),Y(I4),Z(I4))
          CALL HF_INSERT_LINE(X(I4),Y(I4),Z(I4),X(I1),Y(I1),Z(I1))
          CALL HF_INSERT_LINE(X(I5),Y(I5),Z(I5),X(I6),Y(I6),Z(I6))
          CALL HF_INSERT_LINE(X(I6),Y(I6),Z(I6),X(I7),Y(I7),Z(I7))
          CALL HF_INSERT_LINE(X(I7),Y(I7),Z(I7),X(I8),Y(I8),Z(I8))
          CALL HF_INSERT_LINE(X(I8),Y(I8),Z(I8),X(I5),Y(I5),Z(I5))
          CALL HF_INSERT_LINE(X(I1),Y(I1),Z(I1),X(I5),Y(I5),Z(I5))
          CALL HF_INSERT_LINE(X(I2),Y(I2),Z(I2),X(I6),Y(I6),Z(I6))
          CALL HF_INSERT_LINE(X(I3),Y(I3),Z(I3),X(I7),Y(I7),Z(I7))
          CALL HF_INSERT_LINE(X(I4),Y(I4),Z(I4),X(I8),Y(I8),Z(I8))
130   CONTINUE
      CALL HF_CLOSE_SEGMENT()
      CALL HF_PAUSE()
      CALL HF_CLOSE_SEGMENT()
      CALL HF_CLOSE_SEGMENT()
      CALL HF_CLOSE_SEGMENT()
      STOP
      END
```


Post-Processor for Micropolar
Plate Bending
y
z
x
Load
Cantilever Rectangular Plate
Concentrated Load at the Middle of the Free End


Post-Processor for Micropolar
Plate Bending
z
y
x
Load
2D-8Node Element
Cantilever Rectangular Plate :
Concentrated Load at the Middle of the Free End


Post-Processor for Micropolar
Plate Bending
y
z
x
Uniformly Distributed Load
Built-in Rectangular Plate :
Uniformly Distributed Load

References

[1] Cauchy, A., *Recherches sur l'équilibre et le mouvement intérieur des corps solides ou fluides, élastiques ou non élastiques* , Bull. Soc. Philomethique, Paris, 1823, pp. 9-13.

[2] Thomson, W. [Lord Kelvin] and Tait, P. G., *Treatise on Natural Philosophy*, Vol. I, Part II, London, Cambridge University Press, 1883.

[3] Voight, W., *Theoretische Studien Über die Elasticitsverhältnisse der Krystalle* , Abhandlungen der Königlichen Gesselschaft der Wissenschaften Zu Gottingen, Vol. 24, Göttingen, 1887.

[4] Cosserat, E. and F., *Théorie des Corps Déformables*, Paris, A. Hermann and Sons, 1909.

[5] Mindlin, R. D. and Tiersten, H. F., *Effects of Couple-Stresses in Linear Elasticity*, Arch. Rat. Mech. Anal., Vol 11, 1962, pp. 415-448.

[6] Toupin, R. A., *Theories of Elasticity with Couple-Stress*, Arch. Rat. Mech. Anal., Vol. 17, 1964, pp. 85-112.

[7] Truesdell, C. and Toupin, R., *The Classical Field Theories*, Handbuch der Physik, S. Flügge ed., Vol III/1, Berlin, Springer-Verlag, 1960, pp. 226-793.

[8] Eringen, A. C. and Suhubi, E. S., *Non-Linear Theory of Simple Micro-Elastic Solids - I*, Int. J. Engrg. Sci., Vol 2, 1964, pp. 189-203.

[9] Suhubi, E. S. and Eringen, A. C., *Non-Linear Theory of Micro-Elastic Solids - II*, Int. J. Engrg. Sci., Vol. 2, 1964, pp. 389-404.

[10] Mindlin, R. D., *Microstructure in Linear Elasticity*, Arch. Rat. Mech. Anal., Vol. 16, 1964, pp. 51-78.

[11] Yang, J. F. C. and Lakes, R. S., *Evidence of Micropolar Elasticity in bone.*

[12] Eringen, A. C., *Theory of Micropolar Elasticity*, FRACTURE - An Advanced Treatise, Liebowitz - Editor, Vol. II, 1968, pp. 621-729.

[13] Timoshenko, S. and Kreiger, S. W., *Theory of Plates and Shells*, McGraw-Hill Book Co., 1959.

[14] Gauthier, R. D., *Analytical and Experimental Investigations in Linear Isotropic Micropolar Elasticity*, Ph.D. Dissertation, Univ. of Colorado, 1974.

[15] Gauthier, R. D. and Jahsman, W. E., *A Quest for Micropolar Elastic Constants*, J. App. Mech., June 1975, pp. 369-374.

[16] Nakamura, S., Benedict, R. and Lakes, R., *Finite Element Method for Orthotropic Micropolar Elasticity*, Int. J. Engrg. Sci., Vol. 22, No.3, 1984, pp. 319-330.

[17] Nakamura, S., *Variational Formulation of Micropolar Elasticity*, Internal Report I, 1982.

[18] Nakamura, S., *Finite Element Formulation of Micropolar Elasticity*, Internal Report II, 1982.

[19] Nakamura, S., *Micropolar Finite Element Analysis of Stress-Concentration Around a Hole*, Internal Report III-V, 1982.

[20] Kardestuncer, H. - Editor-in Chief, *Finite Element Handbook*, McGraw-Hill Book Company, 1987.

[21] Dym, C. L. and Shames, I. H., *Solid Mechanics: A Variational Approach*, McGraw-Hill Inc., 1973.

[22] Hughes, T. J. R., Taylor, R. L. and Kanoknukulchai, W., *A Simple and Efficient Element for Plate Bending*, Int. J. Nuum. Meth. Eng., Vol. 11, 1977, pp. 1529-1543.

[23] MacNeal, R. H., *A Simple Quadrilateral Shell Element*, Computers and Structures, Vol. 8, 1978, pp 175-188.

[24] Hrabok, M. M. and Hrudey, T. M., *A Review and Catalog of Plate Bending Finite Elements*, Computers and Structures, Vol. 19, 1984, pp 479-495.

[25] Belytschko, T. and Chen-Shyh Tsay, *A Stabilization Procedure for the Quadrilateral Plate Element with One Point Quadrature*, Int. J. Num. Meth. Eng.,

Vol. 19, 1983, pp. 405-419.

[26] Lamport, L., LaTeX:*A Document Preparation System*, Version 2.09, Addison-Wesley Publishing Company, 1986.

[27] Cowin, B. C., *An Incorrect Iniquality in Micropolar Elasticity Theory*, ZAMP, Vol. 21, 1970, pp. 494-497.

[28] Bathe, Klaus Jürgen and Wilson, Edward, *Numerical Methods in Finite Element Analysis*, Prentice-Hall of India Pvt. Ltd., 1987.

[29] Huang, Fuang Yuan, *Two and Three-dimensional Isoparametric Finite Elements for Axisymmetric Micropolar Elasticity*, Doctoral Dissertation, New Jersey Institute of Technology, 1986.