STRESS COMPATIBLE BIMATERIAL INTERFACE ELEMENTS WITH APPLICATION TO TRANSIENT DYNAMIC STRESS ANALYSIS

by

Jun Yi

October, 1994

Department of Civil Engineering and Applied Mechanics
McGill University, Montreal, Canada

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfilment of the requirements for the degree of
Master of Engineering

National Library of Canada
Bibliothèque nationale du Canada

Acquisitions and Bibliographic Services Branch
Direction des acquisitions et des services bibliographiques

395 Wellington Street
Ottawa, Ontario
K1A 0N4

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

ISBN 0-612-07991-0

Canada

Abstract

Conventional displacement-based finite element programs do not yield unique values of stress components which ought to be continuous at element interfaces. The errors, being the differences from the correct unique values, become unacceptably large at a bimaterial interface when the moduli of the two materials are significantly different.

This thesis formulates and implements new finite elements for obtaining the correct values of the stress components, both continuous and discontinuous ones, at bimaterial interface points under general dynamic loading, assuming linear, isotropic, elastic material behaviour.

The constructed finite element programs, suitable for analyzing two-dimensional and axisymmetric three-dimensional problems, have been validated by comparing the predicted responses with the exact analytical solutions of some non-trivial impact loading (wave-propagation) problems.

The work provides a necessary tool for analyzing and designing composite structures, for example prosthetic knee and hip joints in the biomechanics field.

Résumé

Les programmes classiques de calculs par la méthode des éléments finis, fondés sur une résolution en terme de déplacements, ne permettent pas d'assurer la continuité des contraintes à l'interface des différents éléments utilisés. L'erreur sur les valeurs des composantes des contraintes, définie comme l'écart entre la valeur correcte et la valeur fournie par le calcul utilisant la méthode des éléments finis, devient inacceptable à l'interface entre deux matériaux lorsque les modules de ces deux matériaux diffèrent de manière significative.

Cette thèse décrit l'élaboration et la mise en application d'éléments finis nouveaux permettant d'obtenir les valeurs correctes des composantes des contraintes en certains points de l'interface entre deux matériaux pour un chargement dynamique quelconque, en supposant un comportement élastique linéaire et isotrope des matériaux.

Les programmes d'éléments finis ainsi élaborés, utilisables pour l'étude de problèmes à deux dimensions ou pour l'analyse de problèmes axisymétriques à trois dimensions, ont été validés en comparant les résultats qu'ils permettent d'obtenir avec les solutions analytiques exactes de problèmes non triviaux de chargements impulsionnels (propagation d'ondes).

Ce travail fournit un outil nécessaire pour l'élaboration et l'analyse de structures composites comme celles, par exemple, d'une prothèse de genou ou de hanche dans le domaine de la biomécanique.

Acknowledgements

I would like to thank my supervisor, Professor S. C. Shrivastava, for his guidance and advice throughout this research.

I also wish to express thanks to my wife, Liu Li, for her support and patience that made my research work successful.

Table of Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

1.1 Bimaterial Interface Problem

Composite structures, in which two materials of different properties are joined together to behave as a unit, present challenging problems in the field of stress analysis. The source of the challenge is the fact that the stress and strain fields are not continuous as transition is made from one material zone to the other at the interface boundaries. While some stress and strain components must necessarily remain continuous, others are allowed to become discontinuous. Moreover, the greater are the differences in the material properties of the two materials, the greater is the degree of discontinuity.

There are many examples of engineering structures which involve use of two materials joined to each other. The most familiar examples are reinforced concrete structures wherein concrete and steel are joined together to form a "perfect bond". The ratio of the Young's moduli in this case ranges from 10 to 25 depending on the types of concrete and steel. Another example, from the field of biomechanics, is a prosthetic joint in which a stainless steel prosthesis is cemented in the knee or hip joint by PMMA bone cement [1]. The ratio of the moduli at the steel/PMMA interface is of the order of 100.

The stress analysis of composite structures of practical importance can only be carried out numerically, usually by the finite element method. However, it is well known that in conventional finite element analysis only the displacement compatibility is satisfied. The requirements of stress compatibility are completely ignored, with the result that the conventional finite element analysis gives non-unique values of even those stress component at inter-element boundaries which ought to be unique. The differences between the non-unique values is exacerbated at bimaterial interfaces when the two materials are highly different in their material properties, for example at an interface of rubber and steel (in bearing pads for bridges to simulate the roller condition). The differences between the non-unique values remain significant, even when a refined mesh is used at the interface.

From a design point of view, the knowledge of correct magnitudes of interface stresses is of critical importance in order to be able to design the interface against bond failure by shear or separation. Hence the interface problem for the present work consists in accurately predicting the magnitudes of all interface stress components, both the continuous ones and the discontinuous ones at bimaterial interfaces.

Solution of this problem requires formulation and implementation of finite elements which are capable of providing the correct values of all stress components at bimaterial interfaces irrespective of the differences in the material properties of the two materials.

Although, as discussed below, this problem has been solved for the static loading case, the present thesis addresses this problem for dynamic loading in general and impact loading in particular. In the dynamic loading case, it is not only the moduli which determine the bimaterial interface stresses, but also the mass densities of the constituent materials.

1.2 Literature Review

The popular finite element programs of today, for example, NASTRAN, SAP, ANSYS, provide no elements which can be used to obtain the compatible stress values at bimaterial interfaces. In fact, the author is not aware of any commercially available finite element program which solves this problem.

When using the conventional finite elements, the stress incompatibility problem exists even for structures which are made from one material of homogeneous properties. The stresses jump from element to element as the inter-element boundaries are crossed. However, these jumps are small, and can be corrected by employing averaging and smoothing techniques [2,3,4,5].

For bimaterial interfaces, Salama and Utku [6] have proposed a method of post-processing the results of conventional (displacement based) finite element analysis to obtain the interface stresses. They claim success of this indirect and approximate procedure of enforcing stress continuity on the basis of some example problems solved by them.

Soh [7] attacked the problem in a slightly different way by introducing a nine-node rectangular element for plane problems, with 3 nodes on each side and one interior node in the centre. The element is used by centering it appropriately at a bimaterial interface boundary. The element is formulated by the usual assumed displacement approach and the compatible stress components at the centre (interface) node are obtained by a second procedure on the calculated nodal displacements of the nine node element.

The most direct approach was taken by Angelides [8], who introduced bimaterial interface elements which enforced the stress compatibility conditions at one point of the bimaterial interface belonging to the element. Such elements were developed for two-dimensional plane-stress, plane-strain, axisymmetric, and three-dimensional analyses. Validation test were performed for the plane and axisymmetric elements by comparing the finite element results with analytical solutions of some simple structures with bimaterial interfaces.

1.3 Objective of Present Work

The elements formulated by Angelides [8], solve the stated interface problem in a general way, but only for static loading. The objective of this thesis is to formulate and implement the interface elements for solving plane and axisymmetric elasticity problems under general dynamic loading. Extension to three-dimensional problems, although straight forward, is not pursued in the interest of restricting the scope of the investigation. The theoretical approach is similar to that used by Angelides [8] in that interface stress compatibility conditions are enforced as part of the formulation and solution procedure.

The numerical procedure adopted for constructing the finite element programs is to be based on the mode superposition technique. Damping effects are to be neglected.

The programs are to be validated by comparing the numerical values obtained by using the constructed programs with the values from exact solutions of some simple but non-trivial problems.

Application of the constructed programs is to be demonstrated by analyzing time variation of stresses at the metal/cement interface of an axisymmetric model of a knee joint fitted with a central stem-type prosthesis, and loaded by a suddenly applied step loading. Both static and dynamic analyses are to be performed with the objective of demonstrating the differing effects on interface stresses when the same load is applied statically and dynamically.

1.4 Organization of Thesis

The thesis is presented in six chapters and three appendices. The first chapter introduces the reader to the interface problem addressed by the thesis, relevant previous work, and the objectives to be achieved.

The second chapter is central to the thesis. First, it presents general theoretical aspects of the interface problem. Following that, interface elements for solving two-dimensional and axisymmetric elasticity problems are formulated. This chapter constitutes the most original part of the work.

The third chapter outlines the construction of finite element programs incorporating the newly formulated elements. A brief exposition of the mode superposition technique of dynamic stress analysis is given from a rather independent point of view.

The fourth chapter is devoted to the validation of the constructed finite element programs against test problems, the exact solutions of which are worked out in Appendices A and B.

The fifth chapter demonstrates the application of the constructed finite element program to the study of the interface stresses in an axisymmetric model of a prosthetic tibia fitted with a metal prosthesis.

The sixth and concluding chapter summarizes the important aspects of the work, and reiterates some of its salient conclusions.

Appendix A is concerned with obtaining the exact solution to the wave propagation problem in a bimaterial composite bar of finite length, subjected to impact type loading. This solution is used to validate the two-dimensional finite element program of the thesis.

Exact solutions, for validating the axisymmetric finite element program, are obtained in Appendix B. The problem tackled is that of a bimaterial composite disc subjected to a step-jump of radial pressure at the outer periphery. The solutions obtained constitute an original contribution, as this problem does not appear to have been solved in full by other researches.

Appendix C provides a listing of the constructed finite element program.

Chapter 2

PROBLEM FORMULATION

Theoretical formulation of bimaterial interface elements is presented in this chapter. Although the scope of applications in the present study is limited to two dimensional and axisymmetric problems, the formulation is presented from a general view point, appropriate for three dimensional problems as well. The materials are considered to be linear elastic isotropic materials, and attention is restricted to small deformations. Hence the mathematical problem is a linear one.

2.1 Continuity and Discontinuity Conditions at a Bimaterial Interface

It may be recalled that the nodal displacement compatibility is always enforced in the conventional displacement-based finite element method. In addition, if possible, suitable element displacement functions are chosen to ensure displacement compatibility not only within the elements but also at the interelement boundaries. Thus, in general (at least in two and three-dimensional elasticity problems), displacement compatibility is enforced throughout the continuum. While such a choice of displacement field ensures continuity of some strain components at the interelement boundary, none of the stress components are required to satisfy any continuity conditions. Thus, even when the material and geometrical properties of the two elements on the two sides of an interface boundary are exactly the same, the conventional finite element method does not enforce the required continuity of the inter-element stress and strain components.

Now, according to Newton's third law (i.e. the principle of equal action and reaction), the force exerted by element *"a"* on element *"b"* across their common boundary is equal and opposite to that exerted by element *"b"* on element *"a"*. Thus, in absence of any applied forces at the interelement boundary, we must have the following equality between the stress vectors:

$$\vec{F}_i^a = -\vec{F}_i^b \tag{2.1}$$

at the interelement boundary. If, as indicated in Fig. 2.1, direction x_1 is normal to the interface and directions x_2 and x_3 are in its tangent plane, then from the definition of stress vectors (e.g. $F^a_i = \sigma^a_{ij} n^a_j$) we must have

$$\sigma^a_{11} = \sigma^b_{11}, \qquad \sigma^a_{12} = \sigma^b_{12}, \qquad \sigma^a_{13} = \sigma^b_{13} \tag{2.2}$$

regardless of the material properties. The remaining stress components at the interelement boundary are allowed to be discontinuous.

Assuming that there always exists a perfect bond (i.e. neither slip nor separation is allowed) between the two materials, the displacement must be continuous at the interface

$$u_1^a = u_1^b, \qquad u_2^a = u_2^b, \qquad u_3^a = u_3^b \tag{2.3}$$

Hence, the displacement gradients in the plane of the interface must also be continuous by virtue of the fact that $dx_2^a = dx_2^b$ and $dx_3^a = dx_3^b$. Thus, one has

$$\begin{aligned} \frac{\partial u_1^a}{\partial x_2^a} &= \frac{\partial u_1^b}{\partial x_2^b}, \qquad \frac{\partial u_2^a}{\partial x_2^a} = \frac{\partial u_2^b}{\partial x_2^b}, \qquad \frac{\partial u_3^a}{\partial x_2^a} = \frac{\partial u_3^b}{\partial x_2^b} \\ \frac{\partial u_1^a}{\partial x_3^a} &= \frac{\partial u_1^b}{\partial x_3^b}, \qquad \frac{\partial u_2^a}{\partial x_3^a} = \frac{\partial u_2^b}{\partial x_3^b}, \qquad \frac{\partial u_3^a}{\partial x_3^a} = \frac{\partial u_3^b}{\partial x_3^b} \end{aligned} \tag{2.4}$$

The above relations when used in the definition of the strain components imply that regardless of the material and geometrical properties of the elements, the following equalities must hold between the strain components at the interface:

$$\epsilon^a_{22} = \epsilon^b_{22}, \qquad \epsilon^a_{33} = \epsilon^b_{33}, \qquad \epsilon^a_{23} = \epsilon^b_{23} \tag{2.5}$$

The remaining strain components are, of course, allowed to be discontinuous.

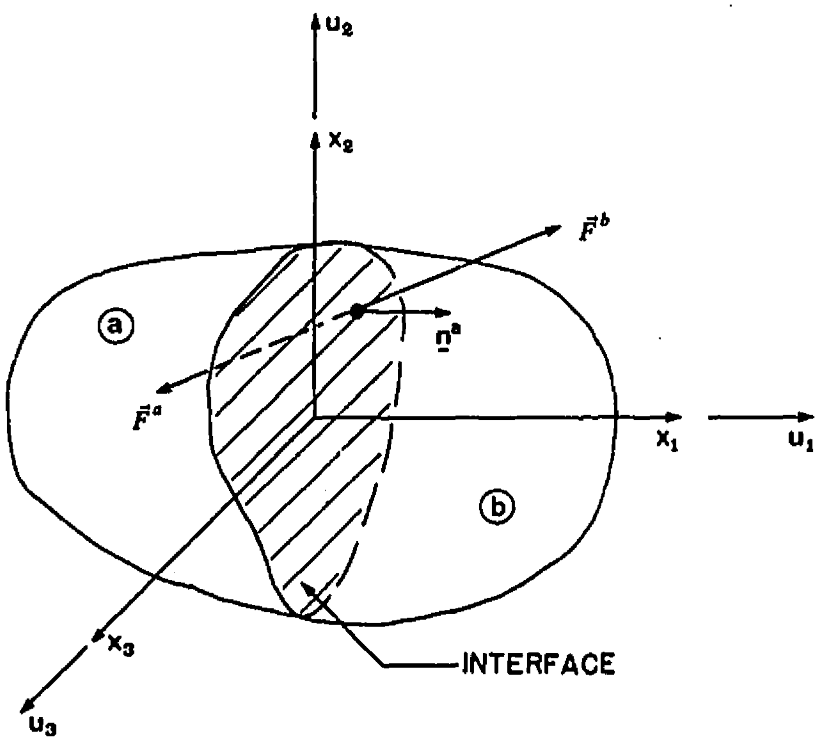


Figure 2.1 Typical bimaterial interface.

Summarizing, we see that according to the above, the stress and strain components at an interface may be written as

$$[\sigma] = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22}^{*} & \sigma_{23}^{*} \\ \sigma_{31} & \sigma_{32}^{*} & \sigma_{33}^{*} \end{bmatrix}, \qquad [\epsilon] = \begin{bmatrix} \epsilon_{11}^{*} & \epsilon_{12}^{*} & \epsilon_{13}^{*} \\ \epsilon_{21}^{*} & \epsilon_{22} & \epsilon_{23} \\ \epsilon_{31}^{*} & \epsilon_{32} & \epsilon_{33} \end{bmatrix} \tag{2.6}$$

wherein the components which must be continuous are those without the asterisks, while those which are allowed to be discontinuous are marked with asterisks. It is interesting to note that the continuity and discontinuity of stress and strain components are complementary in that when a certain stress (or strain) component is required to be continuous, the corresponding strain (or stress) component is allowed to be discontinuous.

2.2 Finite Element Equations of Motion

The equations of motion applicable to a continuum are

$$\frac{\partial \sigma_{ij}}{\partial x_j} + b_i = \rho \ddot{u}_i \qquad \sigma_{ij} = \sigma_{ji} \tag{2.7}$$

in its volume V, and

$$\sigma_{ij} n_j = F_i \tag{2.8}$$

at the boundary S_σ where external tractions F_i per unit area are present; b_i and ρ are respectively the body force components and the mass density per unit volume, n_j are the components of the unit outward normal to S_σ and a dot denotes differentiation with respect to time. The virtual work equation, obtainable from the standard procedure, and applicable for any continuum is

$$-\int_V \sigma_{ij}\delta\epsilon_{ij} dV + \int_V b_i \delta u_i dV + \int_S F_i \delta u_i dS = \int_V \rho \delta u_i \ddot{u}_i dV \qquad (2.9)$$

wherein δu_i denotes a virtual infinitesimal displacement field, and $\delta\epsilon_{ij}$ the corresponding virtual strain field. The interelement boundary terms at an interface vanish on account of the continuity of the assumed displacement field.

For finite element analysis, the integrals in the above equation can be decomposed over the domains of each element, so that the above equation can be written as:

$$\begin{aligned} -\Sigma\int_{V_e} \sigma_{ij}\delta\epsilon_{ij} dV + \Sigma\int_{V_e} b_i \delta u_i dV + \Sigma\int_S F_i \delta u_i dS \\ = \Sigma\int_{V_e} \rho \delta u_i \ddot{u}_i dV \end{aligned} \qquad (2.10)$$

For deriving finite element equations, a switch to matrix notation is convenient. Assume that the displacement field within an element is expressible as

$$\{u\}_{3x1} = [N]_{3xn}\{\Delta\}_{nx1} \qquad (2.11)$$

where $\{u\}^T = \langle u_1, u_2, u_3 \rangle$ denotes the components of the displacement vector and $[N]$ is the matrix of shape functions which are assumed to be functions of the coordinates alone (and not of time). On the other hand $\{\Delta\}$, the vector of nodal displacements, is assumed to be a function of time. Hence

$$\{\ddot{u}\} = [N]\{\ddot{\Delta}\} \qquad (2.12)$$

The strain components, expressed as a row (or column) vector $\{\epsilon\}^T = \langle \epsilon_{11}, \epsilon_{22}, \epsilon_{33}, \epsilon_{12}, \epsilon_{23}, \epsilon_{31} \rangle$, are obtainable by differentiation of Eq. (2.11) according to the definition of strain tensor, and can be expressed as

$$\{\epsilon\} = [B]\{\Delta\} \qquad (2.13)$$

where $[B]$ matrix contains derivatives of the shape functions $[N]$. Invoking the stress-strain law for linear elastic isotropic material we obtain

$$\{\sigma\} = [E]\{\epsilon\} = [E][B]\{\Delta\} \tag{2.14}$$

where $\{\sigma\}$ is the column vector of stress components and $[E]$ is the matrix of elastic moduli. The above constitutive law implies that any damping (i.e. dissipation) is neglected. Substitution of the above constitutive relations into the virtual work equation, Eq. (2.10), then leads to

$$\begin{aligned} -\Sigma\{\delta\Delta\}^T\left(\int_{V_e}[B]^T[E][B]dV\right)\{\Delta\} + \Sigma\{\delta\Delta\}^T\left(\int_{V_e}[N]^T\{b\}dV\right) \\ + \Sigma\{\delta\Delta\}^T\left(\int_{S_e}[N]^T\{F\}dS\right) = \Sigma\{\delta\Delta\}^T\left(\int_{V_e}\rho[N]^T[N]dV\right)\{\ddot{\Delta}\} \end{aligned} \tag{2.15}$$

Since this relation must hold for arbitrary $\{\delta\Delta\}^T$, we obtain the following equations of motion

$$[M]\{\ddot{\Delta}\} + [K]\{\Delta\} = \{R\} \tag{2.16}$$

wherein all element matrices have been augmented to the system size and then superposed so that

$$[M] = \Sigma\int_{V_e}\rho[N]^T[N]dV \tag{2.17}$$

$$[K] = \Sigma\int_{V_e}[B]^T[E][B]dV \tag{2.18}$$

$$\{R\} = \Sigma\int_{V_e}[N]^T\{b\}dV + \Sigma\int_{S_e}[N]^T\{F\}dS \tag{2.19}$$

In the standard terminology $[M]$ is the consistent mass matrix of the system, $[K]$ is the system stiffness matrix, and $\{R\}$ is the vector of equivalent nodal forces applied to the system. Although not mentioned above explicitly, the column vector of equivalent applied forces, $\{R\}$, includes concentrated forces and also reaction forces. Needless to say, in a dynamic problem these forces are time-dependent.

The above equations of motion are a system of coupled ordinary linear differential equations. These can be solved analytically in the usual way by first determining the natural frequencies and associated mode shapes. However, for a large system the frequencies and mode shapes can only be determined by a numerical procedure, which in general is a difficult and expensive procedure in terms of effort and computer time.

The alternative method of solving the above equations is by means of their simultaneous step by step numerical integration with respect to time. This is a relatively easier method in terms of computer implementation, but suffers from the defect of error accumulation when "large-time" response is required.

In this present work, the method of mode superposition has been chosen to integrate the equations of motion in consideration of the fact that it is a semi-analytical method the accuracy of which largely depends on the accuracy with which frequencies and mode shapes are determined, starting from the lowest ones.

2.3 Two Dimensional Interface Element

Figure 2.2(b) shows the interface element. The quadrilateral 1-2-7-5-6 belongs to material *"a"* while the quadrilateral 2-3-4-5-7 pertains to material *"b"*. Node 7 on the interface boundary is taken to be at the mid-point of line 2-5.

With reference to Figs. 2.2(a) and (b), the "isoparametric" mapping which maps a 2 x 2 square in the ξ-η plane into a quadrilateral 1-2-7-5-6 is taken as

$$\begin{aligned} x &= N_1^a x_1 + N_2^a x_2 + N_7^a x_7 + N_5^a x_5 + N_6^a x_6 \\ y &= N_1^a y_1 + N_2^a y_2 + N_7^a y_7 + N_5^a y_5 + N_6^a y_6 \end{aligned} \tag{2.20}$$

where

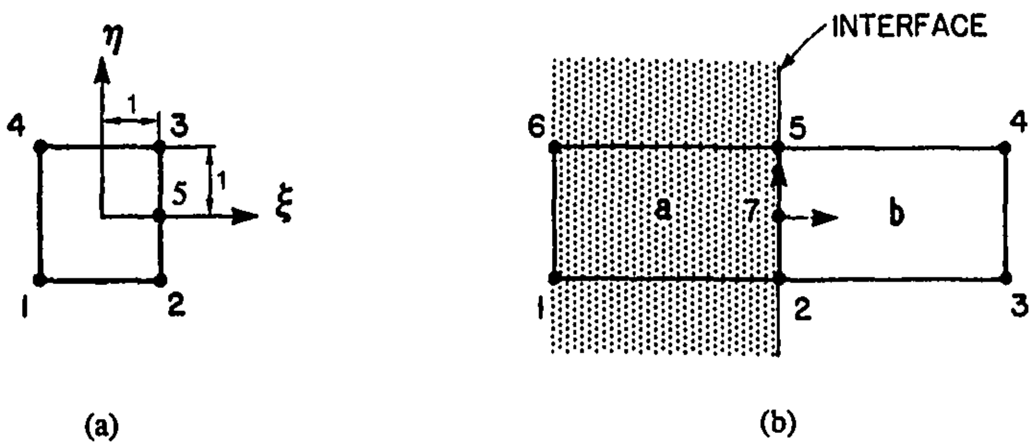


Figure 2.2 (a) Five node quadrilateral isoparametric element, (b) a pair of five node elements sharing an interface.

$$
\begin{aligned}
N_1^a &= (\xi^a-1)(\eta^a-1)/4 \\
N_2^a &= (\xi^a+1)(\eta^a-1)\eta^a/4 \\
N_7^a &= -(\xi^a+1)(\eta^a+1)(\eta^a-1)/2 \\
N_5^a &= (\xi^a+1)(\eta^a+1)\eta^a/4 \\
N_6^a &= -(\xi^a-1)(\eta^a+1)/4
\end{aligned}
\tag{2.21}
$$

Equations (2.21) list the shape functions. Actually, since node 7 is assumed to be at the middle of the side 2-5, one has $x_7=(x_2+x_5)/2$ and $y_7=(y_2+y_5)/2$. This dependence then renders the mapping functions to be subparametric. However, we prefer to use a formulation in which this dependence is not explicitly considered. Node 7 may therefore be placed anywhere on line 2-5.

The normal to the interface is described by the line perpendicular to line 2-5, and is defined by

$$
\vec{n} = \vec{t} \times \vec{k} \tag{2.22}
$$

where $\vec{t}$ is the unit vector along the interface in the direction 2-5, so that one has

$$
\vec{t} = \frac{(x_5-x_2)\vec{i} + (y_5-y_2)\vec{j}}{\sqrt{(x_5-x_2)^2 + (y_5-y_2)^2}} \tag{2.23}
$$

$$
\vec{n} = \frac{-(x_5-x_2)\vec{j} + (y_5-y_2)\vec{i}}{\sqrt{(x_5-x_2)^2 + (y_5-y_2)^2}} \tag{2.24}
$$

Hence, the direction cosines of the normal $\vec{n}$ with respect to the global x,y axes are

$$
c = \cos\theta = \frac{(y_5-y_2)}{\sqrt{(x_5-x_2)^2 + (y_5-y_2)^2}} \tag{2.25}
$$

$$s = \sin\theta = \frac{-(x_5 - x_2)}{\sqrt{(x_5-x_2)^2 + (y_5-y_2)^2}} \tag{2.26}$$

The element displacement functions are taken as

$$\begin{aligned} u(x,y) &= u(\xi^a,\eta^a) = N_1^a u_1 + N_2^a u_2 + N_7^a u_7 + N_5^a u_5 + N_6^a u_6 \\ v(x,y) &= v(\xi^a,\eta^a) = N_1^a v_1 + N_2^a v_2 + N_7^a v_7 + N_5^a v_5 + N_6^a v_6 \end{aligned} \tag{2.27}$$

where according to isoparametric formulation the shape functions $N_1^a, N_2^a \ldots N_6^a$ are exactly those in the mapping functions, i.e. given by Eqs. (2.21).

In a parallel fashion, the mapping and displacement functions for element *"b"* can be written as

$$\begin{aligned} x &= N_2^b x_2 + N_3^b x_3 + N_4^b x_4 + N_5^b x_5 + N_7^b x_7 \\ y &= N_2^b y_2 + N_3^b y_3 + N_4^b y_4 + N_5^b y_5 + N_7^b y_7 \end{aligned} \tag{2.28}$$

$$\begin{aligned} u(x,y) &= u(\xi^b,\eta^b) = N_2^b u_2 + N_3^b u_3 + N_4^b u_4 + N_5^b u_5 + N_7^b u_7 \\ v(x,y) &= v(\xi^b,\eta^b) = N_2^b v_2 + N_3^b v_3 + N_4^b v_4 + N_5^b v_5 + N_7^b v_7 \end{aligned} \tag{2.29}$$

where similar to the element *"a"*,

$$\begin{aligned} N_2^b &= -(\xi^b-1)(\eta^b-1)\eta^b/4 \\ N_3^b &= -(\xi^b+1)(\eta^b-1)/4 \\ N_4^b &= (\xi^b+1)(\eta^b+1)/4 \\ N_5^b &= (\xi^b-1)(\eta^b+1)\eta^b/4 \\ N_7^b &= (\xi^b-1)(\eta^b+1)(\eta^b-1)/2 \end{aligned} \tag{2.30}$$

We note this choice of displacement functions ensures the required displacement compatibility at the interface regardless of the displacement values at nodes 2, 5 and 7. The central idea of the present theory is to select such u_7 and v_7 displacements at the interface that the continuity condition on the stress vector are satisfied at this one point of the interface.

The strains associated with the chosen displacement field can be found as

$$\{\epsilon^a\} = \begin{Bmatrix} \epsilon^a_{xx} \\ \epsilon^a_{yy} \\ 2\epsilon^a_{xy} \end{Bmatrix} = \begin{Bmatrix} \dfrac{\partial u^a}{\partial x} \\ \dfrac{\partial v^a}{\partial y} \\ \dfrac{\partial u^a}{\partial y} + \dfrac{\partial v^a}{\partial x} \end{Bmatrix} = \begin{Bmatrix} \Sigma \dfrac{\partial N^a_i}{\partial x} u^a_i \\ \Sigma \dfrac{\partial N^a_i}{\partial y} v^a_i \\ \Sigma (\dfrac{\partial N^a_i}{\partial y} u^a_i + \dfrac{\partial N^a_i}{\partial x} v^a_i) \end{Bmatrix} = [B^a]\{\Delta^a\} \tag{2.31}$$

where

$$\{\Delta^a\}^T = \langle\, u_1 \; v_1 \; u_2 \; v_2 \; u_7 \; v_7 \; u_5 \; v_5 \; u_6 \; v_6 \,\rangle \tag{2.32}$$

and $[B^a]$ is a 3 x 10 matrix containing derivatives of the shape functions with respect to the global coordinates x and y, i.e.

$$[B^a] = \begin{bmatrix} \dfrac{\partial N^a_1}{\partial x} & 0 & \dfrac{\partial N^a_2}{\partial x} & 0 & \dots \\ 0 & \dfrac{\partial N^a_1}{\partial y} & 0 & \dfrac{\partial N^a_2}{\partial y} & \dots \\ \dfrac{\partial N^a_1}{\partial y} & \dfrac{\partial N^a_1}{\partial x} & \dfrac{\partial N^a_2}{\partial y} & \dfrac{\partial N^a_2}{\partial x} & \dots \end{bmatrix} \tag{2.33}$$

Since the shape functions are defined as functions of ξ-η coordinates, we have

$$\frac{\partial N^a_i}{\partial x} = \frac{\partial N^a_i}{\partial \xi^a}\frac{\partial \xi^a}{\partial x} + \frac{\partial N^a_i}{\partial \eta^a}\frac{\partial \eta^a}{\partial x} \tag{2.34}$$

$$\frac{\partial N_i^a}{\partial y} = \frac{\partial N_i^a}{\partial \xi^a}\frac{\partial \xi^a}{\partial y} + \frac{\partial N_i^a}{\partial \eta^a}\frac{\partial \eta^a}{\partial y} \tag{2.35}$$

Now, $\partial N_i^a/\partial \xi^a$ etc. on the right sides of Eqs. (2.34) and (2.35) are easily computed. However, the computations of $\partial \xi^a/\partial x$ etc. require inversion of the "Jacobian" of the mapping functions. The formulas obtained from such an inversion are as follows:

$$\begin{aligned}
\frac{\partial \xi^a}{\partial x} &= \frac{1}{\det[J^a]}\frac{\partial y}{\partial \eta^a} \\
\frac{\partial \eta^a}{\partial x} &= \frac{-1}{\det[J^a]}\frac{\partial y}{\partial \xi^a} \\
\frac{\partial \xi^a}{\partial y} &= \frac{-1}{\det[J^a]}\frac{\partial x}{\partial \eta^a} \\
\frac{\partial \eta^a}{\partial y} &= \frac{1}{\det[J^a]}\frac{\partial x}{\partial \xi^a}
\end{aligned} \tag{2.36}$$

where

$$\det[J^a] = \det\begin{bmatrix} \dfrac{\partial x}{\partial \xi^a} & \dfrac{\partial y}{\partial \xi^a} \\ \dfrac{\partial x}{\partial \eta^a} & \dfrac{\partial y}{\partial \eta^a} \end{bmatrix} = \frac{\partial x}{\partial \xi^a}\frac{\partial y}{\partial \eta^a} - \frac{\partial x}{\partial \eta^a}\frac{\partial y}{\partial \xi^a} \tag{2.37}$$

Hence, explicitly, the terms of $[B^a]$ are of the following kind

$$\frac{\partial N_i^a}{\partial x} = \frac{1}{\det[J^a]}\left[\frac{\partial N_i^a}{\partial \xi^a}\frac{\partial y}{\partial \eta^a} - \frac{\partial N_i^a}{\partial \eta^a}\frac{\partial y}{\partial \xi^a}\right] \tag{2.38}$$

$$\frac{\partial N_i^a}{\partial y} = \frac{1}{\det[J^a]}\left[-\frac{\partial N_i^a}{\partial \xi^a}\frac{\partial x}{\partial \eta^a} + \frac{\partial N_i^a}{\partial \eta^a}\frac{\partial x}{\partial \xi^a}\right] \tag{2.39}$$

in which right-hand sides can be calculated as functions of ξ,η, the nodal coordinates, and nodal displacement parameters.

The columns of stress components are therefore obtained by invoking the stress-strain relations for the materials

$$\{\sigma^a\} = [E^a]\{\epsilon^a\}, \qquad \{\sigma^b\} = [E^b]\{\epsilon^b\} \tag{2.40}$$

where (omitting the superscript "a" and "b") $\{\sigma\}^T=\langle\sigma_{xx}\ \sigma_{yy}\ \sigma_{xy}\rangle$, and $[E]$ represents the appropriate 3 x 3 matrix of elastic moduli. For plane stress and plane strain problems, these matrices are respectively

$$[E]_{plane\ stress} = \frac{E}{(1-\nu^2)}\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \tag{2.41}$$

$$[E]_{plane\ strain} = \frac{E}{(1+\nu)(1-2\nu)}\begin{bmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix} \tag{2.42}$$

We now recall that the normal and tangential stress components on a plane with outward unit normal making an angle θ with the global x axis are

$$\begin{aligned} \sigma_{nn} &= \sigma_{11}\cos^2\theta + \sigma_{22}\sin^2\theta + 2\sigma_{12}\sin\theta\cos\theta \\ \sigma_{nt} &= (\sigma_{22}-\sigma_{11})\sin\theta\cos\theta + \sigma_{12}(\cos^2\theta-\sin^2\theta) \end{aligned} \tag{2.43}$$

In matrix notation we may write the above relations as

$$\begin{Bmatrix} \sigma_{nn} \\ \sigma_{nt} \end{Bmatrix} = \begin{bmatrix} c^2 & s^2 & 2cs \\ -cs & cs & c^2-s^2 \end{bmatrix} \begin{Bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{Bmatrix} \tag{2.44}$$

or symbolically as

$$\{\sigma\}_{n-t} = [T]\{\sigma\} \tag{2.45}$$

where $[T]$ is the above 2 x 3 transformation matrix, with c = $\cos\theta$ and s = $\sin\theta$.

The continuity of the stress vector at the interface requires that:

$$\{\sigma^a\}_{n-t} = \{\sigma^b\}_{n-t} \tag{2.46}$$

or

$$[T]\{\sigma^a\} = [T]\{\sigma^b\} \tag{2.47}$$

or

$$[T][E^a][\epsilon^a] = [T][E^b][\epsilon^b] \tag{2.48}$$

or

$$[T][E^a][B^a]\{\Delta^a\} = [T][E^b][B^b]\{\Delta^b\} \tag{2.49}$$

or

$$[Q^a]\{\Delta^a\} = [Q^b]\{\Delta^b\} \tag{2.50}$$

where obviously,

$$[Q^a] = [T][E^a][B^a], \quad [Q^b] = [T][E^b][B^b] \tag{2.51}$$

and it may be recalled that

$$\begin{aligned} \{\Delta^a\}^T &= \langle\, u_1\ v_1\ u_2\ v_2\ u_7\ v_7\ u_5\ v_5\ u_6\ v_6 \,\rangle \\ \{\Delta^b\}^T &= \langle\, u_2\ v_2\ u_3\ v_3\ u_4\ v_4\ u_5\ v_5\ u_7\ v_7 \,\rangle \end{aligned} \tag{2.52}$$

Equations (2.50) can be looked upon as two equations to determine two unknowns, u_7 and v_7, in terms of the nodal displacements of the other nodes of elements *"a"* and *"b"*. To solve for u_7 and v_7 we write

$$[Q^a]\{\Delta^a\} = [Q1^a]\{d^e\} + [Q2^a]\{d^i\} \tag{2.53}$$

and

$$[Q^b]\{\Delta^b\} = [Q1^b]\{d^e\} + [Q2^b]\{d^i\} \tag{2.54}$$

where

$$\begin{aligned} \{d^e\}^T &= (\ u_1\ v_1\ u_2\ v_2\ u_3\ v_3\ u_4\ v_4\ u_5\ v_5\ u_6\ v_6\ ) \\ \{d^i\}^T &= \langle\ u_7\ v_7\ \rangle \end{aligned} \tag{2.55}$$

$[Q1^a]$, $[Q1^b]$ are appropriately modified (with additions of zeros) matrices of size 2 x 12, and $[Q2^a]$, $[Q2^b]$ are matrices of size 2 x 2. Explicitly the elements of these matrices, derived from $[Q^a]$ and $[Q^b]$ are

$$[Q1^a] = \begin{bmatrix} Q^a_{1,1} & Q^a_{1,2} & Q^a_{1,3} & Q^a_{1,4} & 0 & 0 & 0 & 0 & Q^a_{1,7} & Q^a_{1,8} & Q^a_{1,9} & Q^a_{1,10} \\ Q^a_{2,1} & Q^a_{2,2} & Q^a_{2,3} & Q^a_{2,4} & 0 & 0 & 0 & 0 & Q^a_{2,7} & Q^a_{2,8} & Q^a_{2,9} & Q^a_{2,10} \end{bmatrix} \tag{2.56}$$

$$[Q2^a] = \begin{bmatrix} Q^a_{1,5} & Q^a_{1,6} \\ Q^a_{2,5} & Q^a_{2,6} \end{bmatrix} \tag{2.57}$$

$$[Q1^b] = \begin{bmatrix} 0 & 0 & Q^b_{1,1} & Q^b_{1,2} & Q^b_{1,3} & Q^b_{1,4} & Q^b_{1,5} & Q^b_{1,6} & Q^b_{1,7} & Q^b_{1,8} & 0 & 0 \\ 0 & 0 & Q^b_{2,1} & Q^b_{2,2} & Q^b_{2,3} & Q^b_{2,4} & Q^b_{2,5} & Q^b_{2,6} & Q^b_{2,7} & Q^b_{2,8} & 0 & 0 \end{bmatrix} \tag{2.58}$$

$$[Q2^b] = \begin{bmatrix} Q^b_{1,9} & Q^b_{1,10} \\ Q^b_{2,9} & Q^b_{2,10} \end{bmatrix} \tag{2.59}$$

The solution for u_7 and v_7 now follows by virtue of the equality of the stress vector at node 7, so that

$$[\, [Q2^b] - [Q2^a] \,]\{d^i\} = [\, [Q1^a] - [Q1^b] \,]\{d^e\} \tag{2.60}$$

or

$$\{d^i\} = [L]\{d^e\} \tag{2.61}$$

where

$$[L] = [\, [Q2^b] - [Q2^a] \,]^{-1}[\, [Q1^a] - [Q1^b] \,] \tag{2.62}$$

is a matrix of size 2 x 12.

The connection between $\{d^e\}$ and $\{\Delta^a\}$, $\{\Delta^b\}$ can be written as

$$\{\Delta^a\} = [R^a]\{d^e\}, \qquad \{\Delta^b\} = [R^b]\{d^e\} \tag{2.63}$$

where $[R^a]$ and $[R^b]$ are 10 x 12 matrices of the following structure:

$$[R^a] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ L_{1,1} & L_{1,2} & L_{1,3} & L_{1,4} & L_{1,5} & L_{1,6} & L_{1,7} & L_{1,8} & L_{1,9} & L_{1,10} & L_{1,11} & L_{1,12} \\ L_{2,1} & L_{2,2} & L_{2,3} & L_{2,4} & L_{2,5} & L_{2,6} & L_{2,7} & L_{2,8} & L_{2,9} & L_{2,10} & L_{2,11} & L_{2,12} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

and

$$
[R^b] = \begin{bmatrix}
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
L_{1,1} & L_{1,2} & L_{1,3} & L_{1,4} & L_{1,5} & L_{1,6} & L_{1,7} & L_{1,8} & L_{1,9} & L_{1,10} & L_{1,11} & L_{1,12} \\
L_{2,1} & L_{2,2} & L_{2,3} & L_{2,4} & L_{2,5} & L_{2,6} & L_{2,7} & L_{2,8} & L_{2,9} & L_{2,10} & L_{2,11} & L_{2,12}
\end{bmatrix}
$$

With the above relations in hand, we can now construct the stiffness and mass matrices of the combined interface element as

$$[k^e] = [R^a]^T[k^a][R^a] + [R^b]^T[k^b][R^b] \tag{2.64}$$

$$[m^e] = [R^a]^T[m^a][R^a] + [R^b]^T[m^b][R^b] \tag{2.65}$$

where $[k^a]$, $[m^a]$ and $[k^b]$,$[m^b]$ are each 10 x 10 matrices, being exactly those which would have been computed for any five-node isoparametric element. Naturally, the combined stiffness and mass matrices $[k^e]$ and $[m^e]$ are of size 12 x 12 with (u_1 v_1 ....u_6 v_6) degrees of freedom.

Hereafter, the method to obtain the assembly of global mass and stiffness matrices and to calculate the nodal displacements of a structure is the same as in the conventional finite element method. The interface nodal degrees of freedom of every node of type 7 are not included in the global nodal displacement field since they are replaced by the stress and displacement continuity at the interface. The global displacements obtained from the solution can be used to calculate the displacements, strain vectors $\{\epsilon^a\}$ and $\{\epsilon^b\}$, and stress vectors $\{\sigma^a\}$ and $\{\sigma^b\}$ for each interface element at the interface node (node 7), which satisfy the interface continuity conditions discussed in Section 2.1.

Summarizing, the formulation of two dimensional interface element has been derived from the perfect continuity conditions by setting up two 5 node elements, one on each side of the interface; the fifth node is taken at the centre of the common interface boundary. The computational procedure to find the displacement, strain, and stress fields for a structure is similar to that used in conventional finite element method. The only difference is that matrices $[L]$ have to be calculated for interface elements.

2.4 Interface Element for Axisymmetric Problems

The interface element for axisymmetric solids of revolution under axisymmetric boundary conditions and loading is mathematically similar in many respects to the two-dimensional interface element. Figure 2.3 shows the cylindrical coordinate system and a typical axisymmetric element. The axisymmetric interface element, similar to the two dimensional interface element discussed in Section 2.3, is composed of two quadrilateral ring elements of materials "a" and "b", in r-z (real) and ξ-η (mapped) radial planes. The mapping and displacement functions for element "a", consistent with isoparametric and axisymmetric assumptions, are taken as

$$\begin{aligned} r &= r^a(\xi,\eta) = N_1^a r_1 + N_2^a r_2 + N_7^a r_7 + N_5^a r_5 + N_6^a r_6 \\ z &= z^a(\xi,\eta) = N_1^a z_1 + N_2^a z_2 + N_7^a z_7 + N_5^a z_5 + N_6^a z_6 \end{aligned} \tag{2.66}$$

and

$$\begin{aligned} u &= u^a(\xi,\eta) = N_1^a u_1 + N_2^a u_2 + N_7^a u_7 + N_5^a u_5 + N_6^a u_6 \\ w &= w^a(\xi,\eta) = N_1^a w_1 + N_2^a w_2 + N_7^a w_7 + N_5^a w_5 + N_6^a w_6 \end{aligned} \tag{2.67}$$

wherein the shape functions are exactly those used for two-dimensional plane element of the previous section, namely

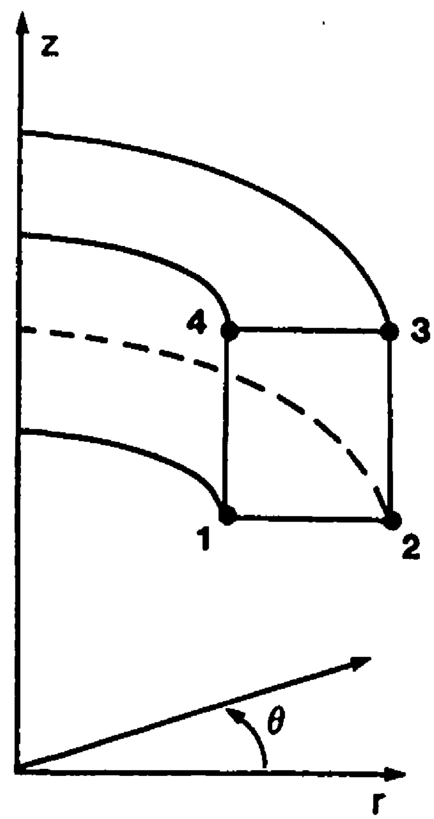


Figure 2.3 Cylindrical coordinate system, and a typical axisymmetric element.

$$
\begin{aligned}
N_2^b &= -(\xi^b-1)(\eta^b-1)\eta^b/4 \\
N_3^b &= -(\xi^b+1)(\eta^b-1)/4 \\
N_4^b &= (\xi^b+1)(\eta^b+1)/4 \\
N_5^b &= (\xi^b-1)(\eta^b+1)\eta^b/4 \\
N_7^b &= (\xi^b-1)(\eta^b+1)(\eta^b-1)/2
\end{aligned}
\tag{2.68}
$$

Evidently, the case of axisymmetric-torsion has been excluded by stipulating that circumferential displacement $v \equiv 0$ throughout.

The non-zero strain components associated with the chosen displacement field may be written as

$$
\begin{Bmatrix} \epsilon_{rr}^a \\ \epsilon_{\theta\theta}^a \\ \epsilon_{zz}^a \\ 2\epsilon_{rz}^a \end{Bmatrix} = \begin{Bmatrix} \dfrac{\partial u^a}{\partial r} \\ \dfrac{u^a}{r} \\ \dfrac{\partial w^a}{\partial z} \\ \dfrac{\partial u^a}{\partial z} + \dfrac{\partial w^a}{\partial r} \end{Bmatrix} = [B^a]\{\Delta^a\}
\tag{2.69}
$$

where $[B]$ is the 4 x 10 matrix involving the shape functions and their derivatives with respect to r and z coordinates. As before, the typical expressions for these derivatives are

$$
\frac{\partial N_i^a}{\partial r} = \frac{1}{\det[J^a]}\left[\frac{\partial N_i^a}{\partial \xi^a}\frac{\partial z}{\partial \eta^a} - \frac{\partial N_i^a}{\partial \eta^a}\frac{\partial z}{\partial \xi^a}\right]
\tag{2.70}
$$

$$
\frac{\partial N_i^a}{\partial z} = \frac{1}{\det[J^a]}\left[-\frac{\partial N_i^a}{\partial \xi^a}\frac{\partial r}{\partial \eta^a} + \frac{\partial N_i^a}{\partial \eta^a}\frac{\partial r}{\partial \xi^a}\right]
\tag{2.71}
$$

where

$$\det[J^a] = \det\begin{bmatrix} \dfrac{\partial r}{\partial \xi^a} & \dfrac{\partial z}{\partial \xi^a} \\ \dfrac{\partial r}{\partial \eta^a} & \dfrac{\partial z}{\partial \eta^a} \end{bmatrix} = \frac{\partial r}{\partial \xi^a}\frac{\partial z}{\partial \eta^a} - \frac{\partial r}{\partial \eta^a}\frac{\partial z}{\partial \xi^a} \tag{2.72}$$

The stress field associated with the assumed displacement (and hence the strain field) is expressible as

$$\{\sigma^a\} = [E^a]\{\epsilon^a\} \tag{2.73}$$

where $\{\sigma^a\} = \langle\, \sigma_{rr}^a \;\; \sigma_{\theta\theta}^a \;\; \sigma_{zz}^a \;\; \sigma_{rz}^a \,\rangle$, and $[E^a]$ = the 4 x 4 matrix of elastic moduli for the (torsionless) axisymmetric behaviour given by

$$[E^a] = \frac{E^a}{(1+\nu^a)(1-2\nu^a)}\begin{bmatrix} 1-\nu^a & \nu^a & \nu^a & 0 \\ \nu^a & 1-\nu^a & \nu^a & 0 \\ \nu^a & \nu^a & 1-\nu^a & 0 \\ 0 & 0 & 0 & \dfrac{1-2\nu^a}{2} \end{bmatrix} \tag{2.74}$$

In the present case, the components of the stress vector on the interface "plane" are

$$\begin{aligned} \sigma^a_{nn} &= \sigma^a_{rr}\cos^2\theta + \sigma^a_{zz}\sin^2\theta + 2\sigma^a_{rz}\cos\theta\sin\theta \\ \sigma^a_{nt} &= (\sigma^a_{zz}-\sigma^a_{rr})\cos\theta\sin\theta + \sigma^a_{rz}(\cos^2\theta - \sin^2\theta) \\ \sigma^a_{n\theta} &\equiv 0 \end{aligned} \tag{2.75}$$

where θ is the inclination of the outward normal to the interface plane belonging to element *"a"* with the r-axis. The direction of the unit tangent vector is defined by line 2-3 in Fig. 2.3. Hence, the expressions for $\cos\theta$ and $\sin\theta$ are similar to those defined previously, Eqs. (2.25) and (2.26) with (x,y) replaced by (r,z). The above relation can be put in the matrix form as

$$\{\sigma^a\}_{n-t} = [T]\{\sigma^a\} \tag{2.76}$$

where $[T]$ is the 2 x 3 transformation matrix defined previously, and $\{\sigma^a\}_{n-t}^T = \langle \sigma_{nn}^a \; \sigma_{nt}^a \rangle$.

A similar procedure applied to element "*b*" yields its respective displacement, strain and stress fields with exactly the same shape functions N_i^b as given by Eq. (2.68). The strain displacement matrix $[B^b]$ is similarly obtained. The components of the stress vector at the interface plane belonging to material "*b*" are then

$$\{\sigma^b\}_{n-t} = [T]\{\sigma^b\} \tag{2.77}$$

where $[T]$ is the same transformation matrix.

The continuity of the stress vector at node 7 is enforced by requiring that

$$\{\sigma^a\}_{n-t} = \{\sigma^b\}_{n-t} \tag{2.78}$$

so that

$$[T][E^a][B^a]\{\Delta^a\} = [T][E^b][B^b]\{\Delta^b\} \tag{2.79}$$

or

$$[Q^a]\{\Delta^a\} = [Q^b]\{\Delta^b\} \tag{2.80}$$

where $[Q^a]$ and $[Q^b]$ are different matrices than for plane problems, peculiar to the axisymmetric problems. Decomposing the $[Q^a]$, $\{\Delta^a\}$ and $[Q^b]$, $\{\Delta^b\}$ matrices in the same fashion, as for the two-dimensional case, we obtain

$$[Q1^a]\{d^e\} + [Q2^a]\{d^i\} = [Q1^b]\{d^e\} + [Q2^b]\{d^i\} \tag{2.81}$$

and

$$\{d^i\} = [L]\{d^e\} \tag{2.82}$$

where, in the same notation as previously used,

$$[L] = [\, [Q2^b] - [Q2^a] \,]^{-1}[\, [Q1^a] - [Q1^b] \,] \tag{2.83}$$

and

$$\{d^e\}^T = (\ u_1\ w_1\ u_2\ w_2\ u_3\ w_3\ u_4\ w_4\ u_5\ w_5\ u_6\ w_6\ ) \qquad (2.84)$$
$$\{d^i\}^T = \langle\ u_7\ w_7\ \rangle$$

The structure of the $[R^a]$ and $[R^b]$ matrices connecting $\{d^c\}$ to $\{\Delta^a\}$ and $\{\Delta^b\}$ is the same as given by Eqs. (2.64) and (2.65) of the previous section. Hence the stiffness $[k]$ and mass $[m]$ matrices of the combined interface element, per radian sweep, are

$$[k^c] = [R^a]^T[k^a][R^a] + [R^b]^T[k^b][R^b] \qquad (2.85)$$

and

$$[m^c] = [R^a]^T[m^a][R^a] + [R^b]^T[m^b][R^b] \qquad (2.86)$$

where

$$[k^a] = \int_{-1}^{1}\int_{-1}^{1}[B^a]^T[E^a][B^a]r\det[J]d\xi d\eta \qquad (2.87)$$

and

$$[m^a] = \int_{-1}^{1}\int_{-1}^{1}\rho[N^a]^T[N^a]r\det[J]d\xi d\eta \qquad (2.88)$$

with similar expressions for $[k^b]$ and $[m^b]$, the stiffness and mass matrices, of the *"b"* element. As before the free degrees of freedom of the interface element are $\langle u_1\ w_1\ u_2\ w_2\ u_3\ w_3\ u_4\ w_4\ u_5\ w_5\ u_6\ w_6\rangle$.

Chapter 3

NUMERICAL METHOD AND COMPUTER IMPLEMENTATION

3.1 Introduction

This chapter presents the numerical method to solve the governing finite element equations, for example Eqs. (2.16), of the previous chapter. The computer programs constructed to implement the interface (and other standard) elements are also outlined, and a flow chart indicating the various stages of the solution process is presented.

As the present work is an extension of a previous research [8], which was concerned solely with analysis under static loading, the computer programs of the present work are constructed by modifying the previous programs and by writing new programs so that the assembled main program is capable of solving problems under general dynamic loadings, with or without the interface elements.

Most of the program development of the present work was done using the microcomputer facilities of the Department of Civil Engineering and Applied Mechanics. These included Intel-486 machines with math-coprocessor, and operating under DOS and OS/2 systems. For the solution of large (real-life) problems, the programs were modified to run on IBM-3090 main-frame computer of McGill University.

The programs have been written in FORTRAN 77, and were compiled using Microsoft Fortran Optimizing Compiler, Version 5.10. The main-frame runs were made after same modifications involving assignment of memory space and input/output statements.

Also worth mentioning is the fact that the program for extracting frequencies needed for the analysis, was an IMSL [9] subroutine named DG2CSP. This was done in the interest of saving time in implementing effectively a crucial step of the numerical procedure. In the analysis procedure adopted in this work, it will be seen that the solution accuracy ultimately hinges upon the accuracy with which frequencies and mode shapes are determined.

3.2 The Mode Superposition Method of Dynamic Analysis

The finite element equations of motion with total m number of degrees of freedom

$$[M]_{mxm}\{\ddot{\Delta}\}_{mx1} + [K]_{mxm}\{\Delta\}_{mx1} = \{R\}_{mx1} \tag{3.1}$$

are looked upon as a system of m coupled second order ordinary differential equations. To integrate them we first apply the specified time-independent boundary conditions of the problem on displacements, and obtain a reduced (n x n) system of equations by deleting the specified degrees of freedom. We then obtain the frequencies of vibration in the process of solving the reduced (n x n) homogeneous system of equations

$$[M]_{nxn}\{\ddot{\Delta}\}_{nx1} + [K]_{nxn}\{\Delta\}_{nx1} = \{0\}_{nx1} \tag{3.2}$$

The solution is assumed as $\{\Delta\} = e^{i\omega t}\{d\}$ which yields the homogeneous system of algebraic equations

$$[\ -\omega^2[M]+[K]\ ]\{d\} = \{0\} \tag{3.3}$$

The frequencies ω_i are the n roots of the characteristic equation

$$\det[\ -\omega^2[M]+[K]\ ] = 0 \tag{3.4}$$

However, as is well known, the determination of the frequencies for a large system can not be pursued in the above direct way. This problem belongs to the general problem of finding eigenvalues λ_i of a system of equations described by

$$[A]\{X\} + \lambda[B]\{X\} = 0 \tag{3.5}$$

There are a number of commercially available subroutines which can solve this problem with efficiency and accuracy on a large or small computer system. As previously mentioned, the subroutine used in this work is the IMSL subroutine called DG2CSP. This subroutine is capable of obtaining all eigenvalues and eigenvectors.

If ω_i and ω_j are any two frequencies of the system and $\{d_i\}$ and $\{d_j\}$ are the associated mode shapes, then it follows from the symmetry of $[M]$ and $[K]$ that:

$$(\omega_i^2-\omega_j^2)\{d_i\}^T[M]\{d_j\} = 0 \tag{3.6}$$

Hence if ω_i and ω_j are distinct, then the associated mode shapes are necessarily orthogonal in the sense that

$$\{d_i\}^T[M]\{d_j\} = \{0\} \tag{3.7}$$

Additionally, we scale the mode shapes $\{d_i\}$ to be of unit "length" by requiring that

$$\{d_i\}^T\{d_i\} = 1 \tag{3.8}$$

We also note that the above kind of orthogonal relation exists also with respect to the stiffness matrix $[K]$ in that

$$\{d_i\}^T[K]\{d_j\} = \{0\} \tag{3.9}$$

Even when the eigenvalues ω_i, ω_j are not distinct (i.e. repeated roots), we can construct distinct orthogonal mode shapes $\{d_i\}$ and $\{d_j\}$ by employing the Gram-Schmidt orthogonalization process. Thus, in any case, we are able to have n distinct and orthonormal mode shapes corresponding to the n degrees of freedom of the system. In the present study, this is achieved by employing the previously mentioned IMSL subroutine DG2CSP.

3.3 Decoupling of the Equations of Motion

We assume that the frequencies have been ordered in increasing magnitudes so that $\omega_1^2 \leq \omega_2^2 \leq \omega_3^2 \ldots\omega_p^2 \ldots \leq \omega_n^2$, and introduce the mode shape matrix $[D]$ as the matrix containing the first p ($p \leq n$) normalized mode shape vectors

$$[D]_{nxp} = [\ \{d_1\}......\{d_p\}\ ] \tag{3.10}$$

Now since $\{d_i\}$ are orthonormal, $[D]$ has the property:

$$[D]^T_{pxn}[D]_{nxp} = [I]_{pxp} \tag{3.11}$$

The choice of p, the number of mode shapes considered, will be dictated in general by the desired accuracy of the solution; large p means accounting of higher frequency modes in the solution. There is no straight-forward way of choosing p, and in general one has to resort to a trial-and-error method.

The orthogonality of $\{d_i\}$, expressed by Eqs. (3.7) and (3.8) can be used to define the following diagonal "mass" and "stiffness" matrices

$$\begin{aligned} [\bar{M}]_{pxp} &= [D]^T[M][D] \\ [\bar{K}]_{pxp} &= [D]^T[K][D] \end{aligned} \tag{3.12}$$

the individual element of which are

$$\begin{aligned} \bar{M}_i &= \{d_i\}^T[M]\{d_i\} \\ \bar{K}_i &= \{d_i\}^T[K]\{d_i\} \end{aligned} \tag{3.13}$$

The definition of the mode shape $\{d_i\}$ as the solution of

$$-\omega_i^2[M]\{d\}_i + [K]\{d\}_i = \{0\} \tag{3.14}$$

then enables us to express the individual frequencies as

$$\omega_i^2 = \frac{\bar{K}_i}{\bar{M}_i} \tag{3.15}$$

and collectively as the diagonal matrix

$$[\omega^2] = [\bar{M}]^{-1}[\bar{K}] \tag{3.16}$$

Introducing now the mode-shape displacement vector $\{Y\}$ by means of the relation

$$\{\Delta\}_{nx1} = [D]_{nxp}\{Y\}_{px1} \tag{3.17}$$

and substituting the above equation into Eq. (3.1), and then premultiplying resulting equation by $[D]^T$, we can write the equations of motion as

$$[D]^T[M][D]\{\ddot{Y}\} + [D]^T[K][D]\{Y\} = [D]^T\{R\} \tag{3.18}$$

or as

$$[\bar{M}]\{\ddot{Y}\} + [\bar{K}]\{Y\} = [D]^T\{R\} \tag{3.19}$$

or as

$$\{\ddot{Y}\}_{px1}+[\omega^2]_{pxp}\{Y\}_{px1} = \{Q\}_{px1} \tag{3.20}$$

where

$$\{Q\}_{px1} = [\bar{M}]^{-1}_{pxp}[D]^T_{pxn}\{R\}_{nx1} \tag{3.21}$$

Eq. (3.20) is the desired system of p decoupled equations, each of the form

$$\ddot{Y}_i+\omega_i^2Y_i = Q_i \tag{3.22}$$

where i = 1, 2, 3, ... p, and p is the number of first frequencies out of the total n.

3.4 Solution of Decoupled Equations

The general solution of the decoupled equations can be written in the matrix form as:

$$\begin{aligned}\{Y\} &= [\sin\omega t]\{A\}+[\cos\omega t]\{B\}+\\ &[\frac{1}{\omega}]\{\int_0^t Q(\tau)\sin\omega(t-\tau)d\tau\}\end{aligned} \tag{3.23}$$

where [sinωt], [cosωt] and [1/ω] are all diagonal matrices of size $p \times p$. The individual equations of the system equation, Eq. (3.22), can be expressed as follows:

$$Y_i = A_i \sin\omega_i t + B_i \cos\omega_i t + \frac{1}{\omega_i}\int_0^t Q_i(\tau)\sin\omega_i(t-\tau)d\tau \tag{3.24}$$

The constants $\{A\}$, $\{B\}$ (or A_i, B_i) can be determined from the initial conditions on Y_i and $\dot{Y}_i$ which are related to the actual initial nodal displacements $\Delta(0)$ and nodal velocities $\dot{\Delta}(0)$ by

$$\begin{aligned} \{Y(0)\} &= \{B\} = [D]^T\{\Delta(0)\} \\ \{\dot{Y}(0)\} &= [\omega]\{A\} = [D]^T\{\dot{\Delta}(0)\} \end{aligned} \tag{3.25}$$

as a consequence of Eqs. (3.17). Hence, the general solution is expressible as

$$\{Y\} = [\frac{\sin\omega t}{\omega}][D]^T\{\dot{\Delta}(0)\}+[\cos\omega t][D]^T\{\Delta(0)\}+ [\frac{1}{\omega}]\{\int_0^t Q(\tau)\sin\omega(t-\tau)d\tau\} \tag{3.26}$$

Invoking now the relation $\{\Delta\}=[D]\{Y\}$ we obtain,

$$\begin{aligned} \{\Delta\} = {} & [D][\frac{\sin\omega t}{\omega}][D]^T\{\dot{\Delta}(0)\}+ \\ & [D][\cos\omega t][D]^T\{\Delta(0)\}+ \\ & [D][\bar{M}]^{-1}\int_0^t[\frac{\sin\omega(t-\tau)}{\omega}][D]^T\{R(\tau)\}d\tau \end{aligned} \tag{3.27}$$

The solution for an individual nodal displacement component is therefore expressible as

$$\begin{aligned} \Delta_j(t) = {} & d_{jl}\frac{\sin\omega_{lk}t}{\omega_{lk}}d_{mk}\dot{\Delta}_m(0)+ \\ & d_{jl}\cos(\omega_{lk}t)d_{mk}\Delta_m(0)+ \\ & d_{je}\frac{1}{\bar{M}_{el}}\int_0^t\frac{\sin\omega_{lk}(t-\tau)}{\omega_{lk}}d_{mk}R_m(\tau)d\tau \end{aligned} \tag{3.28}$$

where (according to the usual convention) a summation is implied on twice-repeated indices in terms on the right hand side, and where $\omega_{ik} = 0$ if $i \neq k$, and $\omega_{ik} = \omega_k$ if $i = k$ and similarly $M_{ei} = 0$ if $e \neq i$ and $M_{ei} = M_i$ is $e = i$. The range of indices is from 1 to p, where as pointed out before p can be chosen less than or equal to n (i.e. $p \leq n$).

3.5 Computer Implementation

For the purpose of computer implementation, we adopt a step-by-step numerical integration procedure by considering the displacements and velocities, $Y_j(t_{k-1})$ and $\dot{Y}_j(t_{k-1})$, at (k-1) *th* time step to be the initial values for determining their values $Y_j(t_k)$ and $\dot{Y}_j(t_k)$ at the k *th* time step. The time step, $\Delta t_k = t_k - t_{k-1}$, is taken to be small enough in that the loading term $Q_j(\tau)$ is approximately constant equal to $Q_j(t_{k-1})$ during the step. Thus, we use the recurrence relations

$$
\begin{aligned}
Y_j(t_k) = {} & Y_j(t_{k-1})\cos(\omega_j \Delta t_k) + \frac{\dot{Y}_j(t_{k-1})}{\omega_j}\sin(\omega_j \Delta t_k) \\
& + \frac{Q_j(t_{k-1})}{\omega_j^2}[1-\cos(\omega_j \Delta t_k)]
\end{aligned}
\tag{3.29}
$$

$$
\begin{aligned}
\dot{Y}_j(t_k) = {} & -\omega_j Y_j(t_{k-1})\sin(\omega_j \Delta t_k) + \\
& \dot{Y}_j(t_{k-1})\cos(\omega_j \Delta t_k) + \frac{Q_j(t_{k-1})}{\omega_j}\sin(\omega_j \Delta t_k)
\end{aligned}
\tag{3.30}
$$

where $k = 0$ (and hence t_0) obviously corresponds to time $t = 0$, the initial time, and $Y_j(t_0)$ and $\dot{Y}_j(t_0)$ are obviously equal to their initial values, i.e.,

$$
Y_j(t_0) = Y_j(0), \qquad \dot{Y}_j(t_0) = \dot{Y}_j(0) \tag{3.31}
$$

The computer program first calculates $\{Y(t_k)\}$ and $\{\dot{Y}(t_k)\}$ for the chosen system frequencies. Then the actual nodal displacements and velocities are calculated by invoking the relationship

$$\Delta(t_k) = [D]\{Y(t_k)\}, \quad \dot{\Delta}(t_k) = [D]\{\dot{Y}(t_k)\} \tag{3.32}$$

Having obtained $\Delta(t_k)$, the steps are retraced to find the corresponding stress and strain fields. The output for each time step is obtained in terms of nodal displacements, components of stress and strain tensor at the Gauss integration points and at the interface nodes.

3.6 Exact Time Integration for Step Loading

As a special case relevant to the biomechanics applications, presented in this work, we note that the loading is considered to be of the step-function type. This means that the load components are also of the same type, for example:

$$Q_j(t) = q_j H(t-0) \tag{3.33}$$

where q_j is the magnitude of the j *th* load component and $H(t\text{-}0)$ is the Heaviside unit step function, defined as

$$\begin{aligned} H(t-0) &= 0 \quad \textit{for} \quad t < 0 \\ H(t-0) &= 1 \quad \textit{for} \quad t > 0 \end{aligned} \tag{3.34}$$

For the above special loading the equation of motion can be integrated exactly as

$$\begin{aligned} Y_j(t) = {} & Y_j(0)\cos(\omega_j t) + \frac{\dot{Y}_j(0)}{\omega_j}\sin(\omega_j t) \\ & + \frac{q_j}{\omega_j^2}[1-\cos(\omega_j t)] \end{aligned} \tag{3.35}$$

Thus for this special type of loading, there is no need to use the step-by-step time integration, which in general will be required for more complex time variations of load.

3.7 Specialization of the Exact Integration to the Static Case

The above modal superposition analysis for dynamic loading can be specialized to static loading case very simply by omitting the time dependent terms in Eq. (3.35), so that

$$Y_j^{statics} = \frac{q_j}{\omega_j^2} \tag{3.36}$$

The rest of the procedure of recovering static displacement, strains, and stress components remains the same as for the dynamic case (with one time step).

In the computer program, the options of dynamic or static analysis are provided by setting the value of a flag parameter to either 1 (dynamic case) or to 0 (static case).

Naturally, the above modal superposition method of static analysis must yield exactly the same results as the direct (conventional) method. However, for correct answers in the static case, it is found that one must use all (and not just first few) computed frequencies and mode shapes in the above procedure.

3.8 Programming Aspects

The computer programs developed in this thesis are based on the theoretical procedure described in Sections 2.2 and 2.3 for two-dimensional and axisymmetric dynamic response using the newly formulated bimaterial interface elements. The flow chart of programs is shown in Fig. 3.1.

The developed programs consist of a main program and a number of subroutines. Besides the interface elements, the usual elements used in other finite element programs are also incorporated in this program. These are bar (i.e. truss) elements for one dimensional problems, constant strain triangular (CST) elements and linear quadrilateral isoparametric (QUAD4) elements which can be used for either plane-stress, plane-strain, or axisymmetric analyses. For QUAD4 and QUAD5 elements, a choice is provided to perform either a 2 x 2 or 3 x 3 Gauss integration to obtain the element mass [*M*] and stiffness [*K*] matrices.

The names of the subroutines used in the program and their functions are described below.

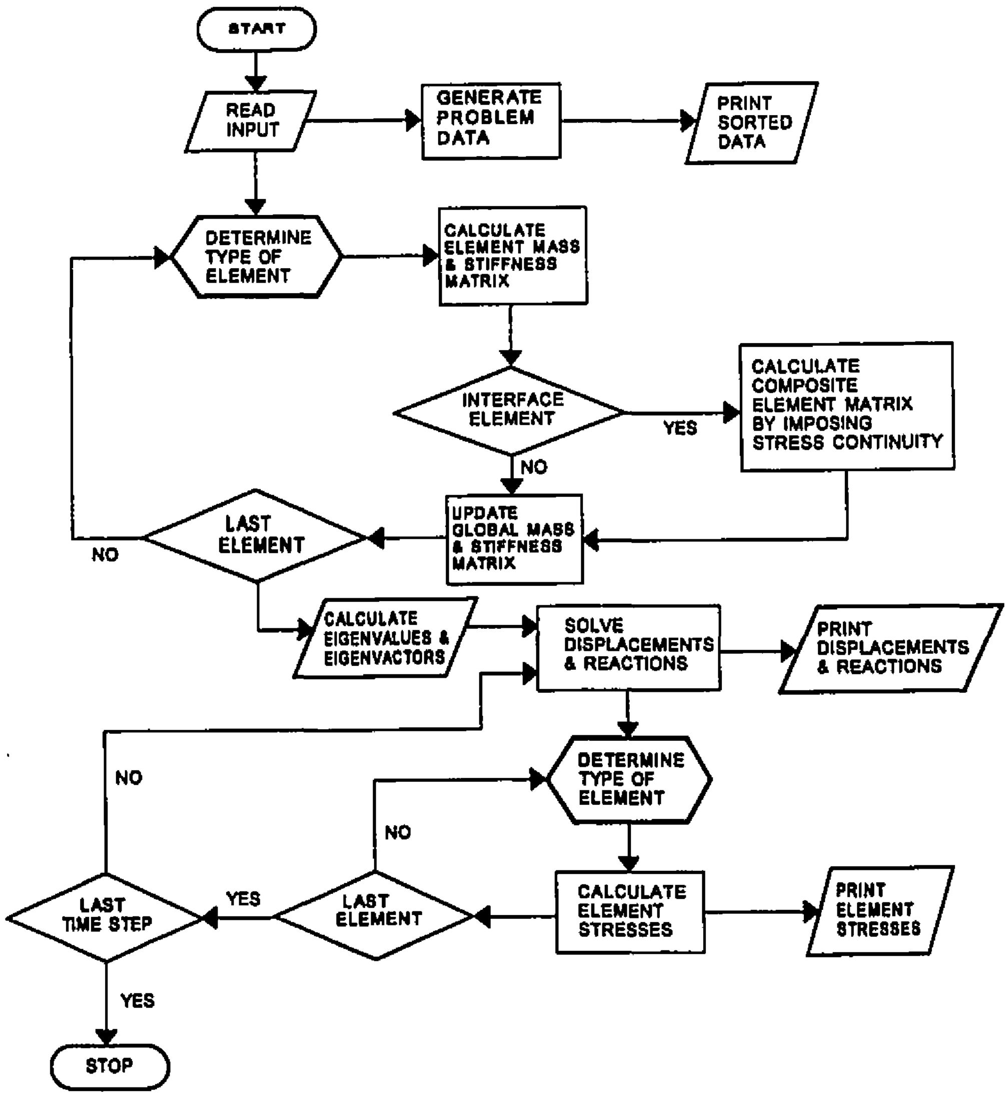


Figure 3.1 Flow chart of the computer program developed in the present work.

GENER : Nodal coordinates and nodal incidence of elements are generated by this subroutine. This requires specification of the node numbers of the first and last elements and the increment values of node and element numbers. The incremented nodal coordinates and element node numbers are then automatically computed.

DATA : The input data is printed in sorted form by this subroutine.

STIFF : This subroutine calculates the individual element stiffness matrices, and assembles global stiffness matrix.

BAR : Calculates stiffness matrix for bar elements.

CST : Calculates the stiffness matrix for the Constant Strain Triangle (CST) element.

QUAD4 : Calculates the stiffness matrix for the 4 node regular and 5 node interface quadrilateral (QUAD4 and QUAD5) elements by using Gaussian integration method [2].

SHAPEF : Calculates strain-displacement matrices for the QUAD4 and QUAD5 elements.

REL : Calculates $[L]$ matrix for enforcing the interface stress and displacement compatibility at node 5 (discussed in Chapter 2).

YOUNG : Calculates the constitutive matrices for the cases of two dimensional (plane-stress and plane-strain) and axisymmetric problems.

DISPL : Gives output of the displacement results, including the interface nodal displacements.

FORCE : Calculates the axial force and elongation in the bar elements.

STR : Calculates stresses in the CST elements.

STRES : Calculates stresses in the QUAD4 and QUAD5 elements at Gauss points.

PRINC : Calculates principal stresses and their directions.

TRANSF : Transforms stresses from global into local interface coordinate system.

MATMAT : Calculates product of two matrices.

MATVEC : Calculates product of a matrix with a vector.

DOT : Calculates the dot product of two vectors.

There are other subroutines which are called TRIAX, QUADAX, SHAPAX, RELAX, STRAX, STRIAX for axisymmetric problems. Their function is similar to the function of subroutines of CST, QUAD4, SHAPEF, REL, STR, STRES for plane- stress and plane-strain problems.

MASS : Subroutine for assembling global consistent mass matrix for a structure.

MBAR : Subroutine for calculating consistent mass matrices of individual bar elements.

MCST : Subroutine for calculating consistent mass matrices of CST elements.

MQUAD4 : Subroutine for formation and numerical integration of consistent mass matrices of QUAD4 or QUAD5 elements.

MSHAPEF : Subroutine for calculating the shape function and the Jacobian determinant of QUAD4 or QUAD5 elements for plane-stress and plane-strain problems.

MTRIAX : Subroutine for calculating mass matrix of triangular axisymmtric elements.

MQUADAX : Subroutine for formation and numerical integration of consistent mass matrix of axisymmetric elements.

MSHAPAX : Subroutine for calculating the shape functions and the Jacobian determinant of axisymmetric QUAD4 or QUAD5 elements.

DG2CSP : An IMSL subroutine for calculating all eigenvalues and eigenvectors of the generalized real symmetric eigenvalue problem $[A]\{x\} = \lambda[B]\{x\}$, with $[B]$ symmetric positive definite [9]. This subroutine is available on the main-frame computer of McGill University.

MODAL : Subroutine to implement the mode superposition method of the computer program. This subroutine is adopted from Rao [10].

Chapter 4

EVALUATION TESTS FOR TWO DIMENSIONAL AND AXISYMMETRIC PROBLEMS

To evaluate the interface elements and the finite element programs developed in this thesis, we carry out three tests on example problems for which analytical solutions have been obtained in Appendix A and Appendix B.

The objective of the first test is to validate the two-dimensional finite element formulation and its program implementation. It consists in computing the dynamic response of a two-materials composite bar supported at one end and loaded by a suddenly applied axial load at the other end. The bar is modelled as an assembly of two dimensional quadrilateral elements, including the 5-node interface element.

The objective of the next two tests is to validate the axisymmetric formulation and program. To this end, dynamic responses of one-material and two-materials discs, loaded by suddenly applied pressure at the outer boundary, are computed and compared with the analytical solutions of Appendix B. In particular, for the two-materials disc, interface stresses obtained by the present interface element are compared with those obtained analytically.

4.1 Two Materials Bar

Figure 4.1 shows the two-materials composite bar, consisting of materials 1 and 2. The right end x = 1, is subjected to a suddenly applied compressive stress $\sigma_0 = 1$ while the left end x = -1 is supported. The interface is taken at x = 0. The element modelling is accomplished by conventional quadrilateral elements and new interface elements. The number of nodes is 103. The position of node 103 is at the middle of interface, chosen to satisfy the stress continuity. The total elements used in this model

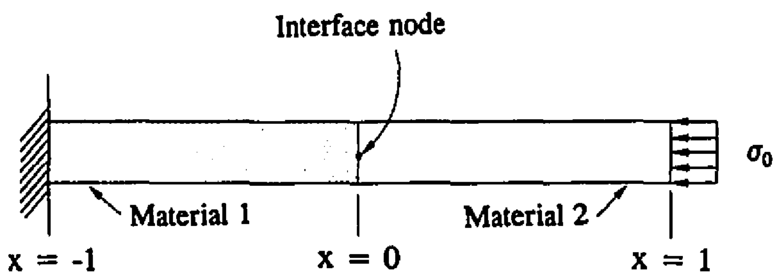


Figure 4.1 Two materials composite bar.

are 50, which include 48 quadrilateral and one pair of interface elements. Bending deformation is prevented by constraining the y-degrees of freedom of nodes, and shear deformation is prevented by keeping the thickness much smaller than the length. Analyses are carried out with and without the interface element.

4.1.1 Hard Material Inside, Soft Material Outside

For this case, we take (dimensionless Young's modulus, density, and cross-sectional area) $E_1 = 200$, $\rho_1 = 8$, and $A_1 = 0.01$ as well as $E_2 = 1$, $\rho_2 = 1$ and $A_2 = 0.01$. The exact solution is obtainable from Appendix A. The frequency comparison is shown in Table 4.1. Clearly the first ten computed frequencies are very close, albeit slightly higher, to the exact ones. The exact frequencies satisfy the relations $\omega_{k+6} = \omega_k + 5\pi$ by virtue of the chosen material parameters.

The computed response in terms of interface displacement closely follows the analytical solution, Fig. 4.2. The interface stress as calculated by the new element also matches closely with the analytical values, Fig. 4.3. On the other hand, although the soft-side interface stress values calculated by using the conventional finite element method are almost equal to those obtained by the interface element, the hard-side stress values are somewhat inaccurate.

4.1.2 Soft Material Inside, Hard Material Outside

Here we switch the positions of the materials. Accordingly, we have $E_1/E_2 = 1/200$ and $\rho_1/\rho_2 = 1/8$. We keep $A_1/A_2 = 1$. Table 4.2 shows that the first ten computed frequencies are quite close to the exact values regardless of whether or not the interface element is used. The exact frequencies again satisfy the relation $\omega_{k+6} = \omega_k + 5\pi$.

The computed displacement response shown in Fig. 4.4 is virtually identical to the exact values. However, as shown in Fig. 4.5, the computed interface stress on the hard-side by using conventional finite elements varies chaotically and bears no relation

Frequency Number i	F.E. with Interface Element ω_i	F.E. without Interface Element ω_i	Analytical Solution ω_i
1	1.5630	1.5630	1.5627
2	4.6853	4.6853	4.6785
3	7.5235	7.5234	7.5079
4	8.2196	8.2195	8.2000
5	11.118	11.118	11.030
6	14.333	14.333	14.145
7	17.614	17.614	17.271
8	20.947	20.947	20.386
9	23.498	23.492	23.216
10	24.576	24.575	23.908

Table 4.1 First ten frequencies (radians/sec) for the two material bar with hard material inside.

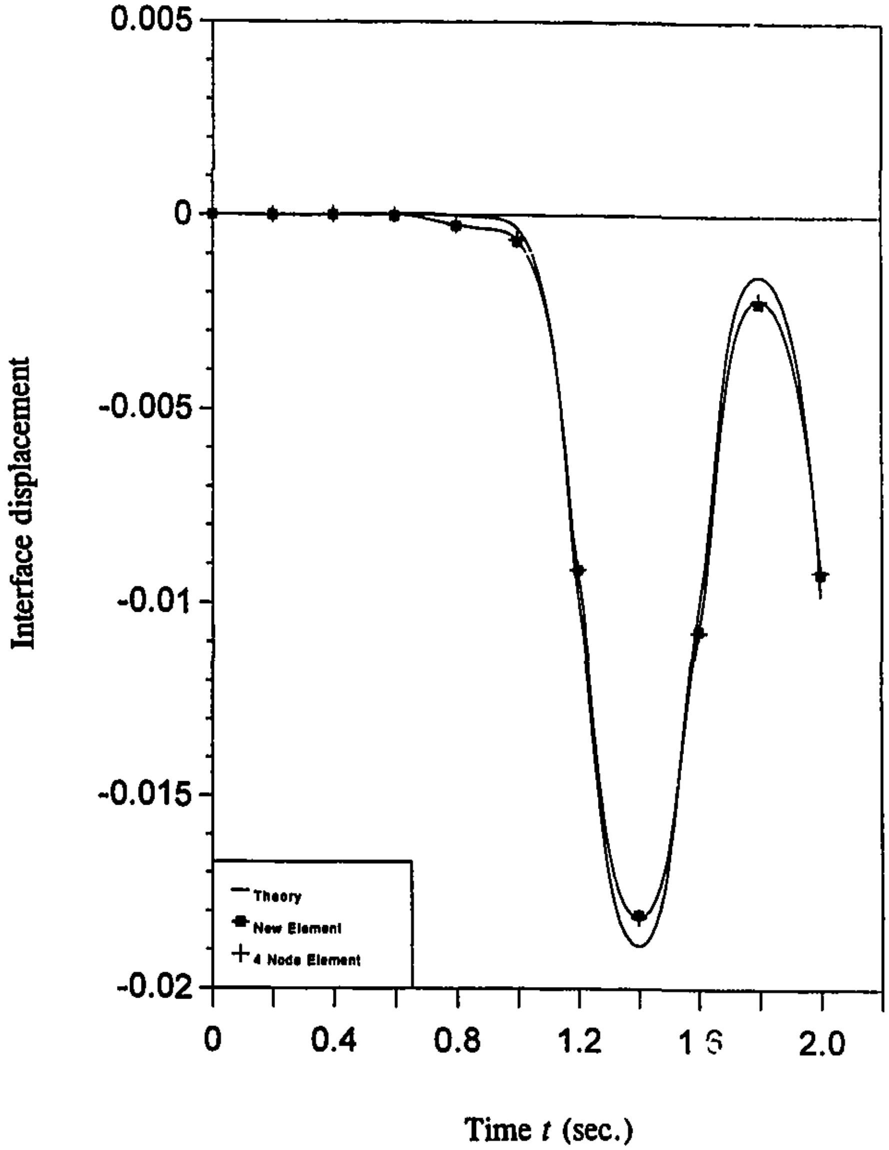


Figure 4.2 Interface displacements for a two materials bar, with hard material inside.

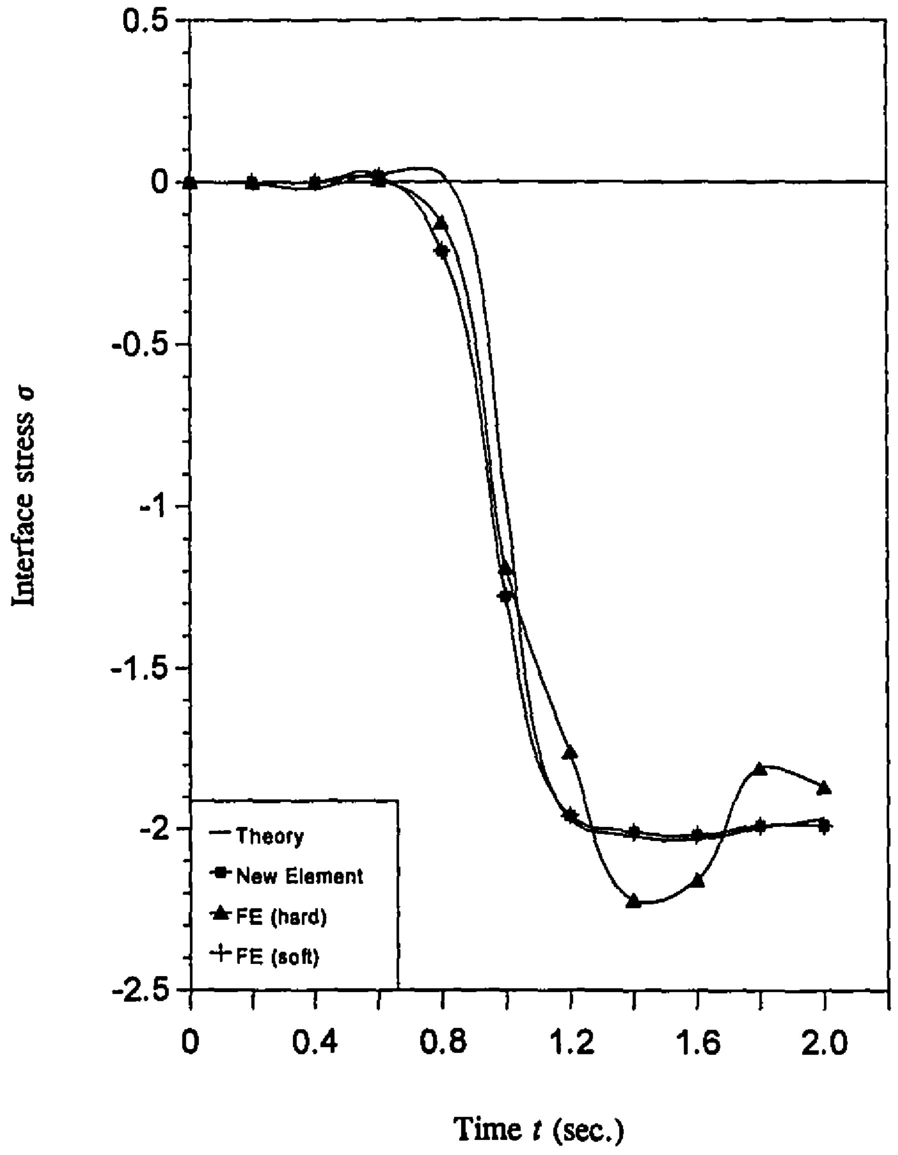


Figure 4.3 Interface stress for a two materials bar, with hard material inside.

Frequency Number i	F.E. with Interface Element ω_i	F.E. without Interface Element ω_i	Analytical Solution ω_i
1	0.3461	0.3461	0.3461
2	3.1776	3.1776	3.1755
3	6.3078	6.3078	6.2913
4	9.4722	9.4722	9.4167
5	12.663	12.663	12.532
6	15.447	15.475	15.362
7	16.220	16.220	16.054
8	19.328	19.328	18.884
9	22.710	22.710	21.999
10	26.185	26.184	25.125

Table 4.2 First ten frequencies (radians/sec) for the two material bar with soft material inside.

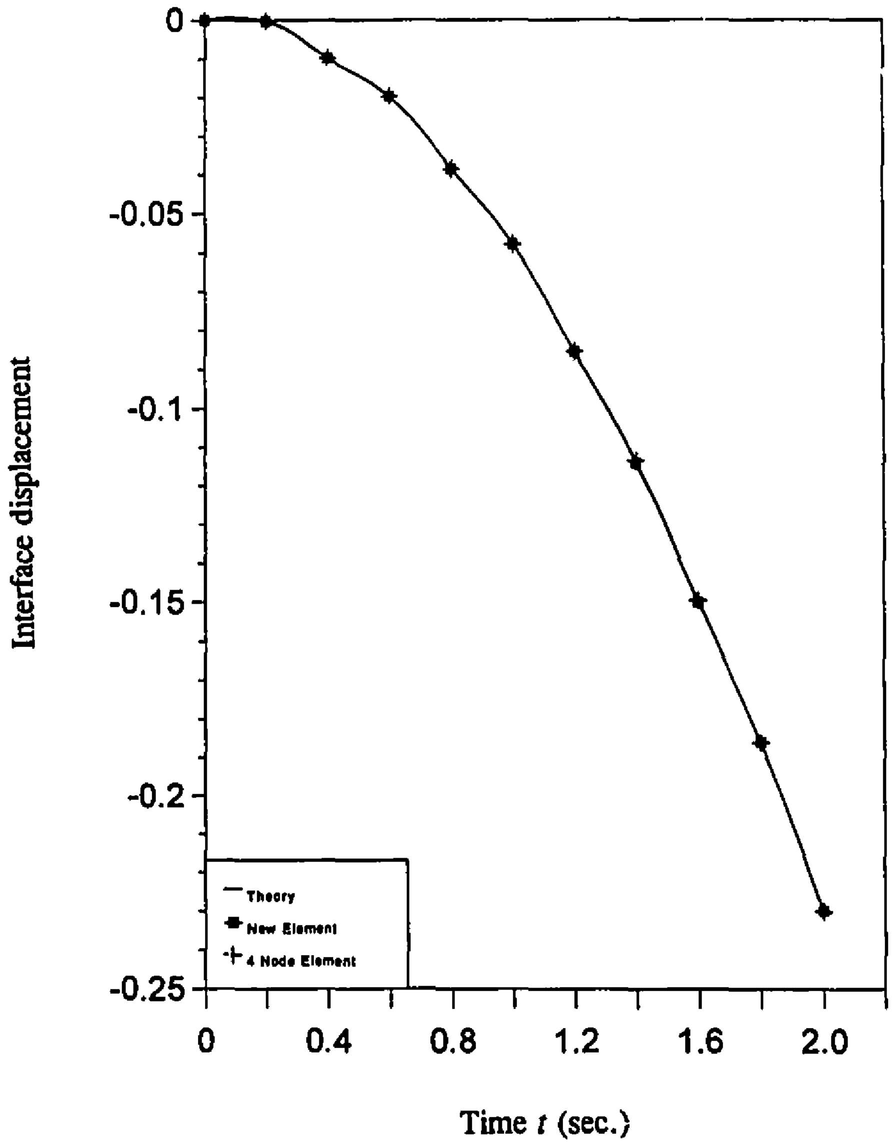


Figure 4.4 Interface displacements for a two materials bar, with soft material inside.

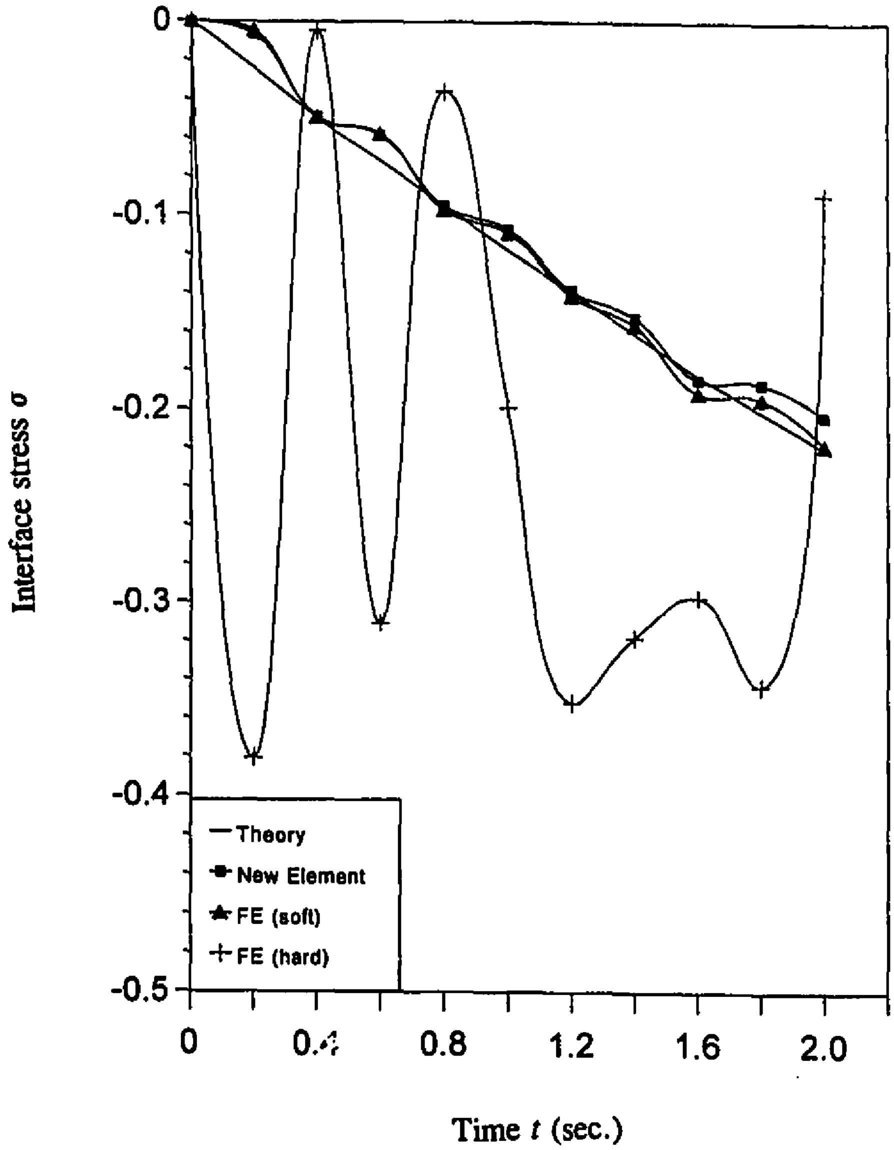


Figure 4.5 Interface stress for a two materials bar, with soft material inside.

to the exact values. The interface stress values by the new element and soft-side values from conventional finite element are quite close to the analytical values.

4.2 One Material Disc

Figure 4.6 shows the type of mesh used for solving this axisymmetric problem. The number of quadrilateral elements is 50. The number of nodes is correspondingly 102. Each node has two (r and z) degrees of freedom. The thickness to radius ratio is taken as 1/10, which ensures that the shear response is inhibited and conditions of the analytical solution are met. The suddenly applied pressure is lumped as equal radial forces at the two outer nodes.

The dimensionless values of Young's modulus (E) and density (ρ) are taken to be unity each. Poisson's ratio is taken equal to zero in accordance with the value used in the analytical solution. The wave velocity $(E/\rho)^{1/2}$ is therefore unity. The response is calculated for a total time t = 2 units, employing an increment $\Delta t = 0.2$. This time is roughly equal to the time for the wave to strike the centre and reach back to the loaded outer boundary.

Table 4.3 shows that the first ten computed frequencies are remarkably close to the analytical ones. Also as required by the theory, the former are always slightly higher than the latter reflecting the somewhat stiffer behaviour of the finite element model.

Displacement response at the outer boundary is shown by values in Table 4.4. Similar to Table 4.3, the agreement between computed and exact values is excellent. Consistent with the observation made with regard to frequency values, the computed displacements are slightly smaller than the exact ones.

Finally, Fig. 4.7 shows comparison of the stress values at the centre (where $\sigma_{rr} = \sigma_{\theta\theta}$). Theoretically, these stresses are zero until the wave arrives at the centre at $t \approx 0.5$, at which time the stresses jump from zero to some finite values. This is reflected both by the present finite-element and exact solutions. The absolute maximum values match quite well, and on the whole the agreement is quite good, considering the fact that

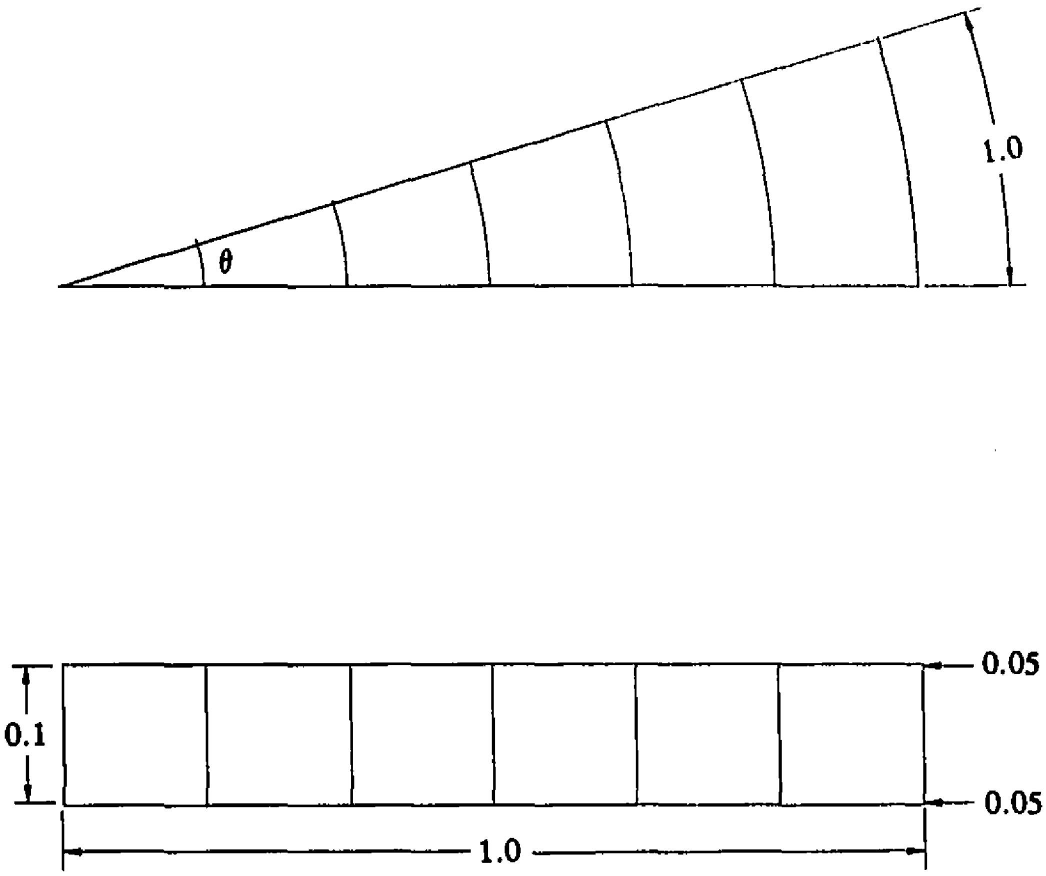


Figure 4.6 Typical mesh for a one material disc.

Frequency Number i	Finite Element Solution ω_i	Analytical Solution ω_i
1	1.8412	1.8412
2	5.3331	5.3314
3	8.5442	8.5363
4	11.728	11.706
5	14.910	14.864
6	18.101	18.016
7	21.306	21.164
8	24.529	24.311
9	27.774	27.457
10	31.046	30.602

Table 4.3 First ten frequencies (radians/sec) for the one material disc.

Time Step ($\Delta t = 0.2$) i	Finite Element solution u_i	Analytical Solution u_i
1	-0.2104	-0.2105
2	-0.4374	-0.4373
3	-0.6818	-0.6818
4	-0.9351	-0.9355
5	-1.1931	-1.1925
6	-1.4420	-1.4427
7	-1.6690	-1.6673
8	-1.8400	-1.8429
9	-1.9166	-1.9230
10	-1.7376	-1.7520

Table 4.4 Displacement along r direction at outer boundary for the one material disc.

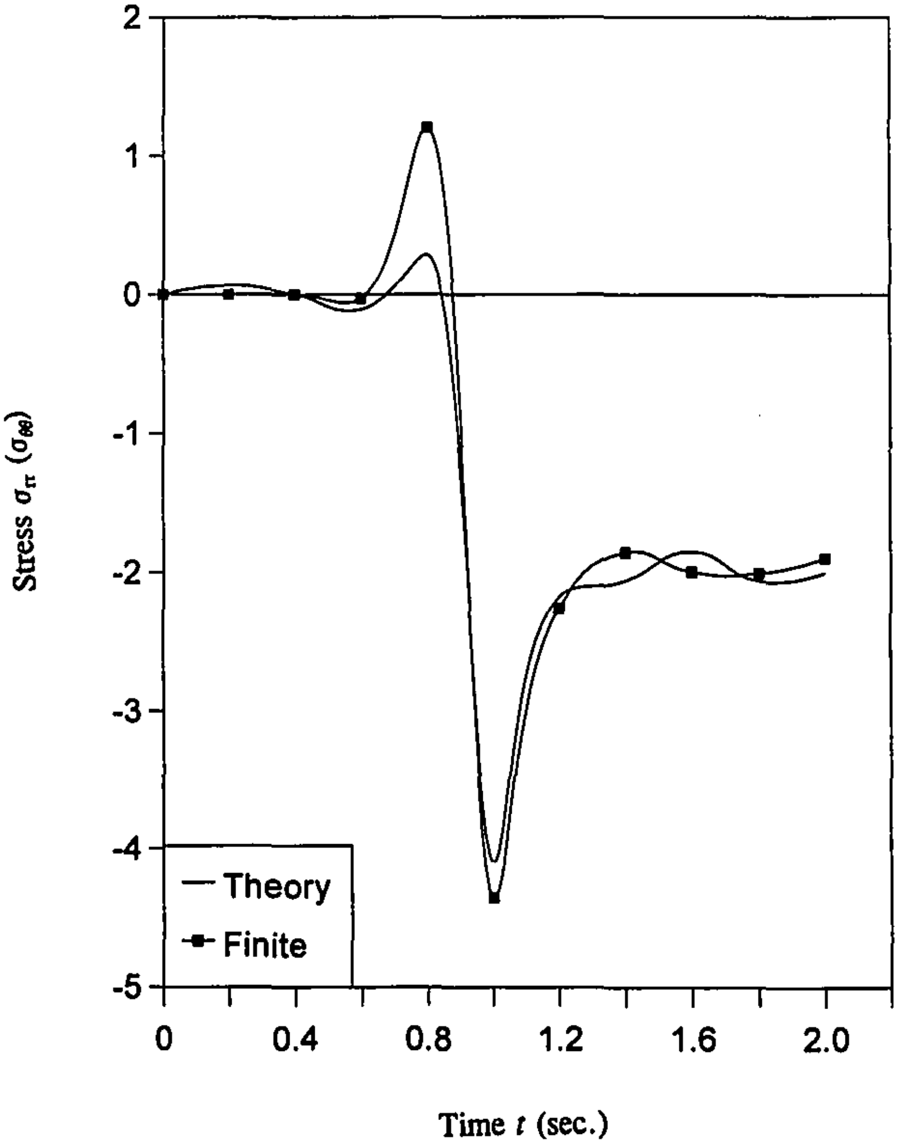


Figure 4.7 Stress σ_{rr} ($\sigma_{\theta\theta}$) at the centre point for one material disc.

the analytical solution is a series solution and generally the finite-element stresses are less accurate than the displacements. The considerable discrepancy at the time value $t = 0.8$ can perhaps be ameliorated by mesh refinement and a finer time increment.

4.3 Two Material Disc

We now evaluate the effectiveness of the new element in predicting correct values of the continuous and discontinuous stress and strain components at the interface under impulse-type dynamic loads.

The composite disc is taken to consist of an inner disc of radius a_1, Young's modulus E_1, density ρ_1, and a surrounding disc of outer radius a_2, E_2, ρ_2. Poisson's ratio for both of inner and outer discs is taken to be zero. Figure 4.8 shows the type of mesh used. The disc is modelled by 48 axisymmetric conventional quadrilateral and one pair of interface elements. The number of nodes is 103. As in other examples, the loading consists of a suddenly applied radial pressure at the outer boundary. The exact analytical expressions for interface displacement, stresses and strains as functions of time are derivable from Appendix B.

4.3.1 Hard Material Inside, Soft Material Outside

We take $a_1 = 1$, $E_1 = 200$ and $\rho_1 = 8$ and $a_2 = 2$, $E_2 = 1$ and $\rho_2 = 1$. Table 4.5 gives a comparison of first ten frequencies obtained analytically and by two finite element analyses, one without any interface elements, and the other with such elements. We note that the computed frequencies are either equal or slightly higher than the analytical values. Moreover, the values are virtually unaffected by the presence of the interface element.

Figure 4.9 shows the time variation of interface displacements. It is remarkable that finite element solution is quite close to the theoretical one, and it does not matter whether or not the interface element is used.

Comparison of the continuous stress component, σ_{rr}, Fig. 4.10, shows that there

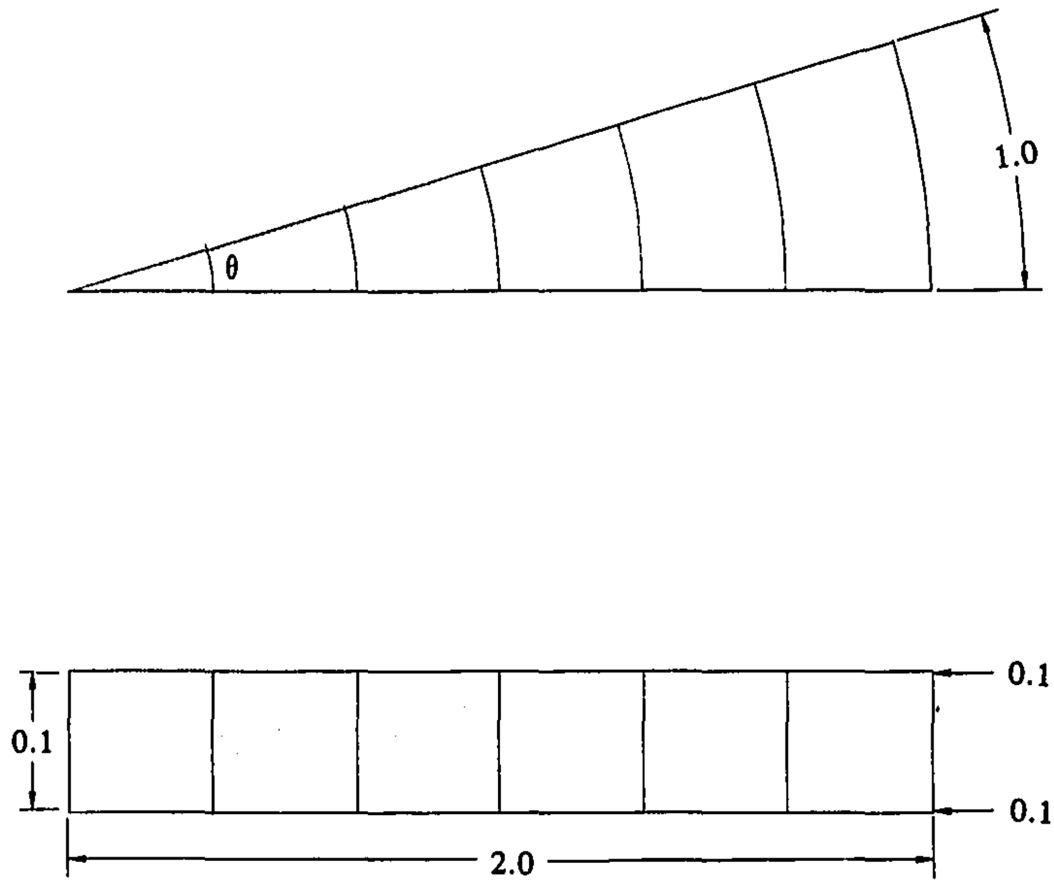


Figure 4.8 Typical mesh for a two materials disc.

Frequency Number i	F.E. with Interface Element ω_i	F.E. without Interface Element ω_i	Analytical Solution ω_i
1	1.4792	1.4792	1.4789
2	4.6744	4.6744	4.6675
3	7.7604	7.7603	7.7311
4	9.2551	9.2549	9.2462
5	11.169	11.169	11.083
6	14.351	14.351	14.162
7	17.629	17.629	17.284
8	20.979	20.979	20.411
9	24.386	24.385	23.526
10	26.650	26.641	26.330

Table 4.5 First ten frequencies (radians/sec) for the two material disc with hard material inside.

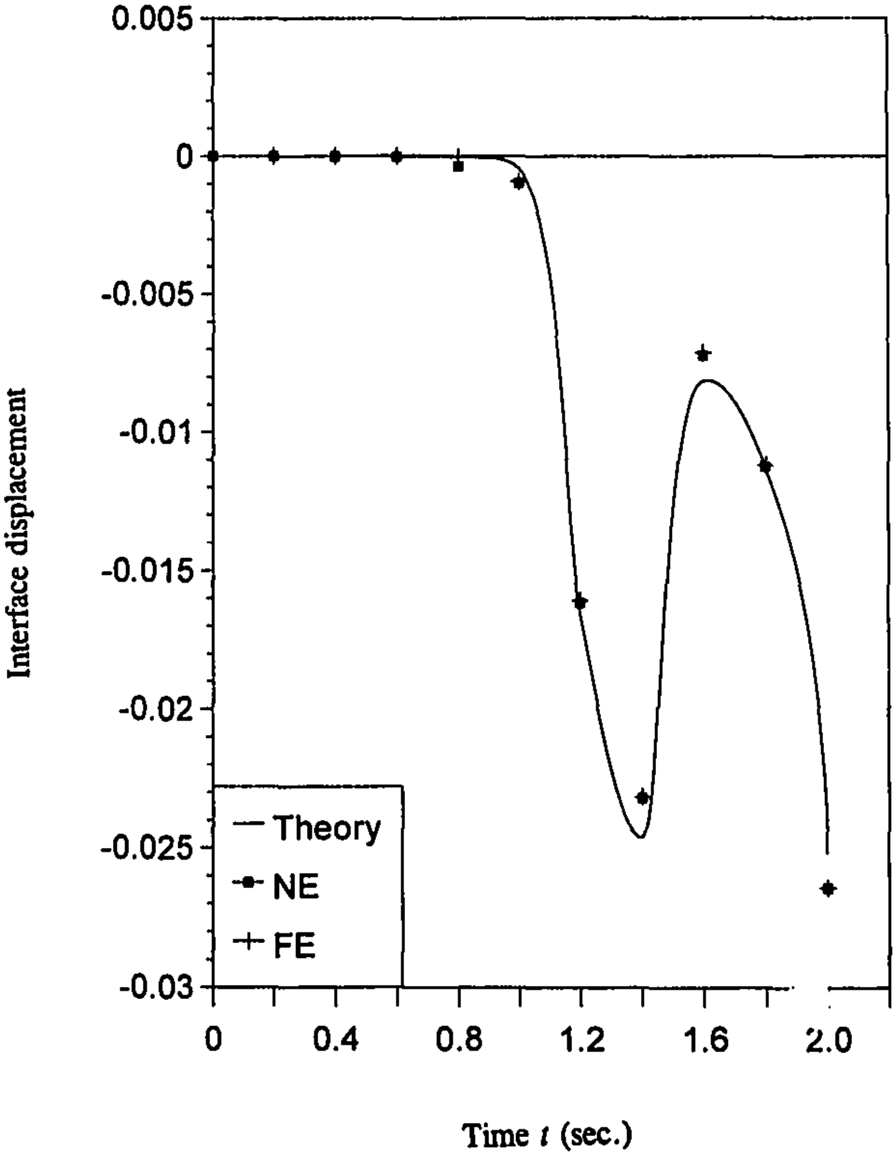


Figure 4.9 Interface displacements for a two materials disc, with hard material inside.

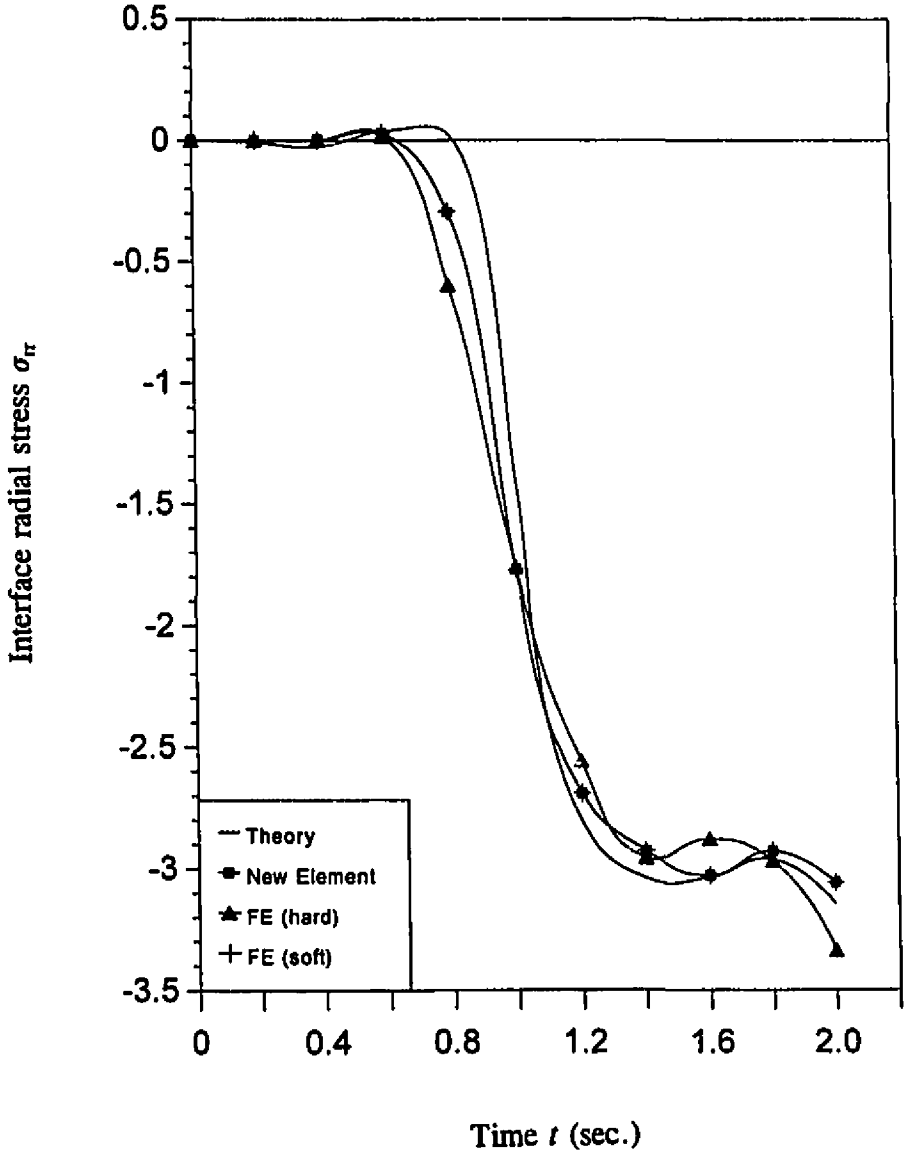


Figure 4.10 Radial stress at the interface of a two materials disc, with hard material inside.

is only a small difference between the finite element results and the analytical ones. However, the interface element yields values which differ the least from the analytical values. Among the two (interpolated) stress values predicted by the usual finite element procedure, the values on soft-side and the hard-side are almost coincident with the values predicted by the new element.

The values of the circumferential stress $\sigma_{\theta\theta}$ are computed and shown in Fig. 4.11 and Fig. 4.12. Since Poisson's ratios are taken to be zero, these stresses are proportional to the interface displacement value and the respective moduli. Hence the comments made with regard to displacements in Fig. 4.9 apply to these stress results as well.

As a conclusion, we can state that although the soft and hard-side finite element results are acceptable if hard material is inside, the unambiguous results obtained by the new interface elements are preferable and more accurate.

4.3.2 Soft Material Inside, Hard Material Outside

Here we switch the positions of the materials. Accordingly, we have E_1/E_2 = 1/200 and ρ_1/ρ_2 = 1/8, we keep a_1/a_2 = 1/2. Table 4.6 shows that the first ten computed frequencies are again close to the analytical exact values. The interface displacement variation, Fig. 4.13, is almost identical for the three types analyses. However, both the radial and circumferential stresses (σ_{rr} and $\sigma_{\theta\theta}$) at the interface are predicted to be rather erratic on the hard-side of the interface, if no interface element is used, Figs. 4.14, 4.15, and 4.16. On the other hand, the values of these stress components, including the discontinuity in $\sigma_{\theta\theta}$, are predicted almost identical to the exact values by the new element. It is clear that the conventional finite element cannot be relied upon to correctly predict the continuous and discontinuous values of σ_{rr} and $\sigma_{\theta\theta}$ at the interface.

4.4 Conclusion from Validation Tests

It has been demonstrated by a comparison of the finite element solutions with

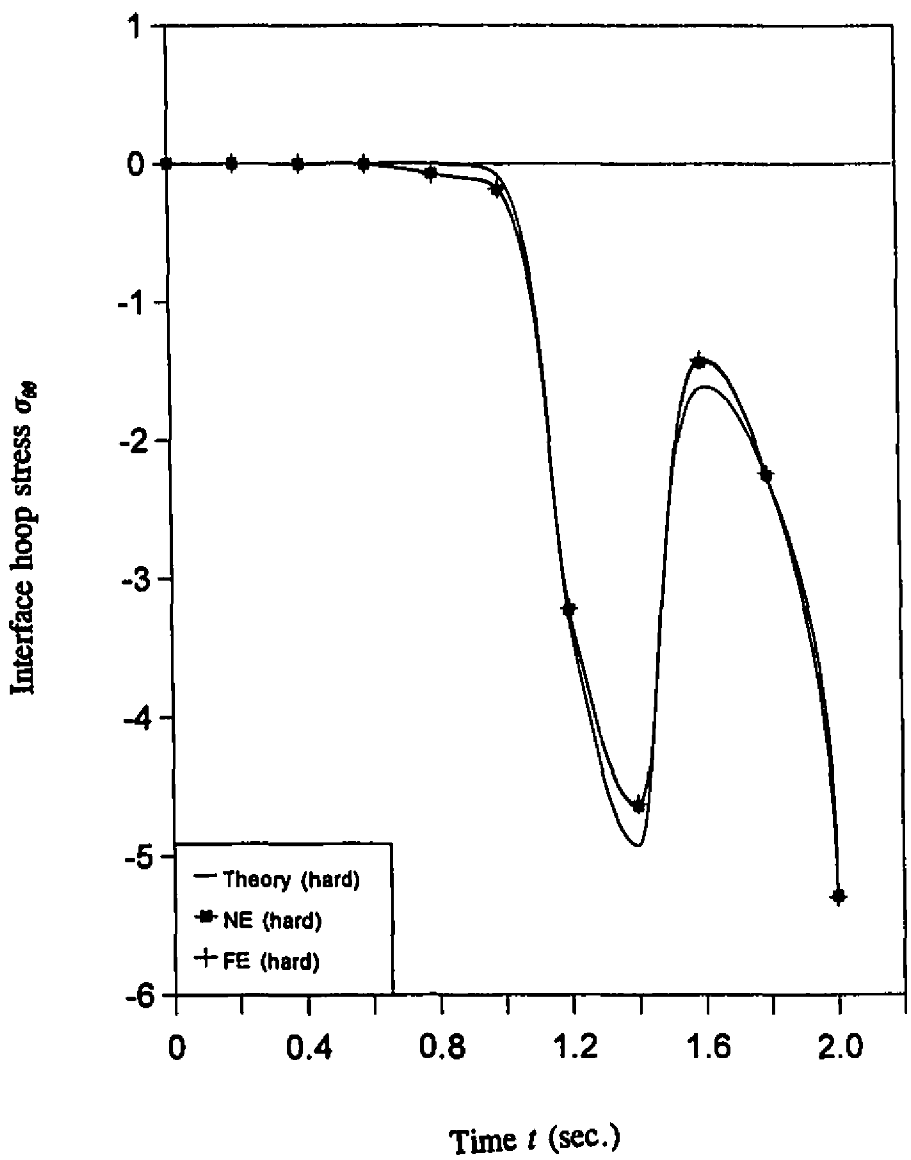


Figure 4.11 Interface hoop stress on the hard side of two material disc, with hard material inside.

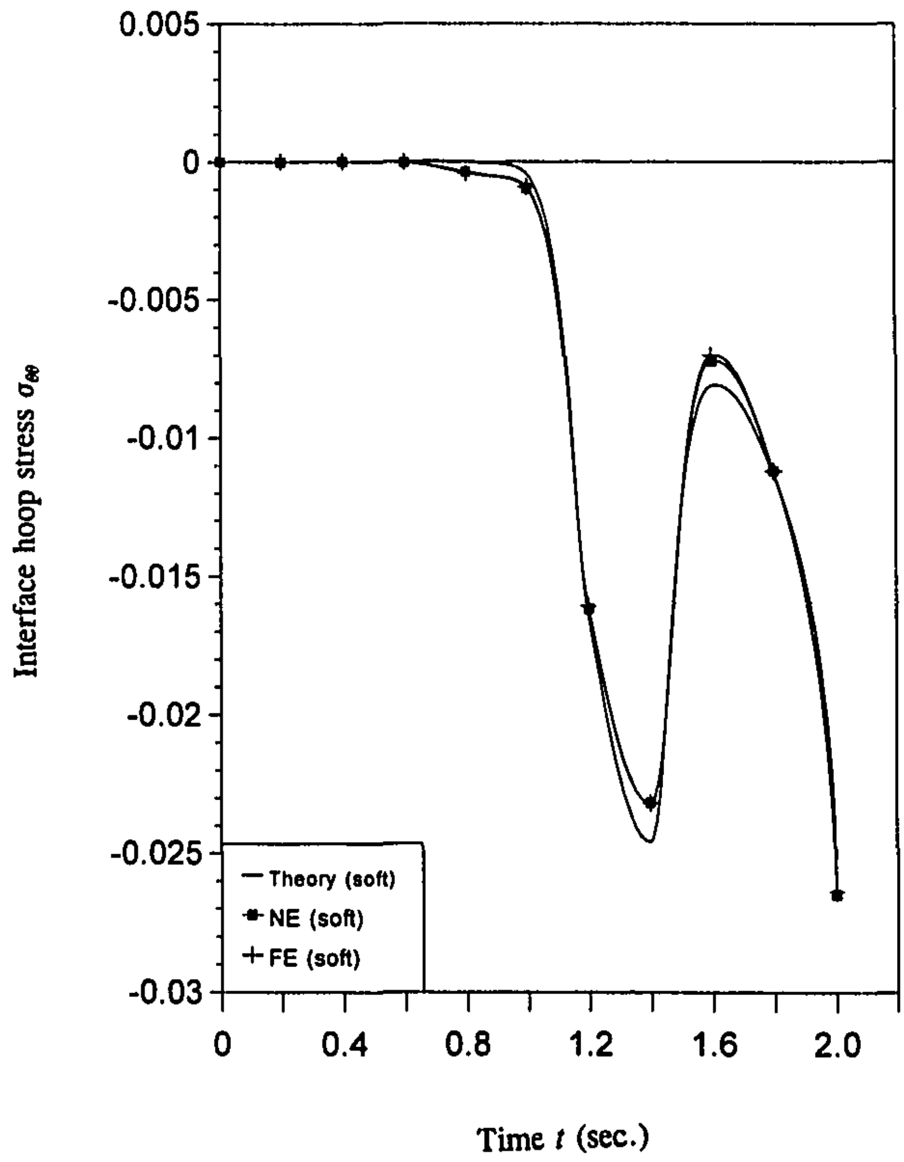


Figure 4.12 Interface hoop stress on the soft side of two materials disc, with hard material inside.

Frequency Number i	F.E. with Interface Element ω_i	F.E. without Interface Element ω_i	Analytical Solution ω_i
1	3.3124	3.3124	3.3123
2	3.9067	3.9067	3.9052
3	7.0334	7.0334	7.0178
4	10.215	10.215	10.161
5	13.414	13.414	13.286
6	16.203	16.202	16.099
7	16.952	16.952	16.780
8	20.085	20.085	19.644
9	23.469	23.469	22.764
10	26.943	26.943	25.891

Table 4.6 First ten frequencies (radians/sec) for the two material disc with soft material inside.

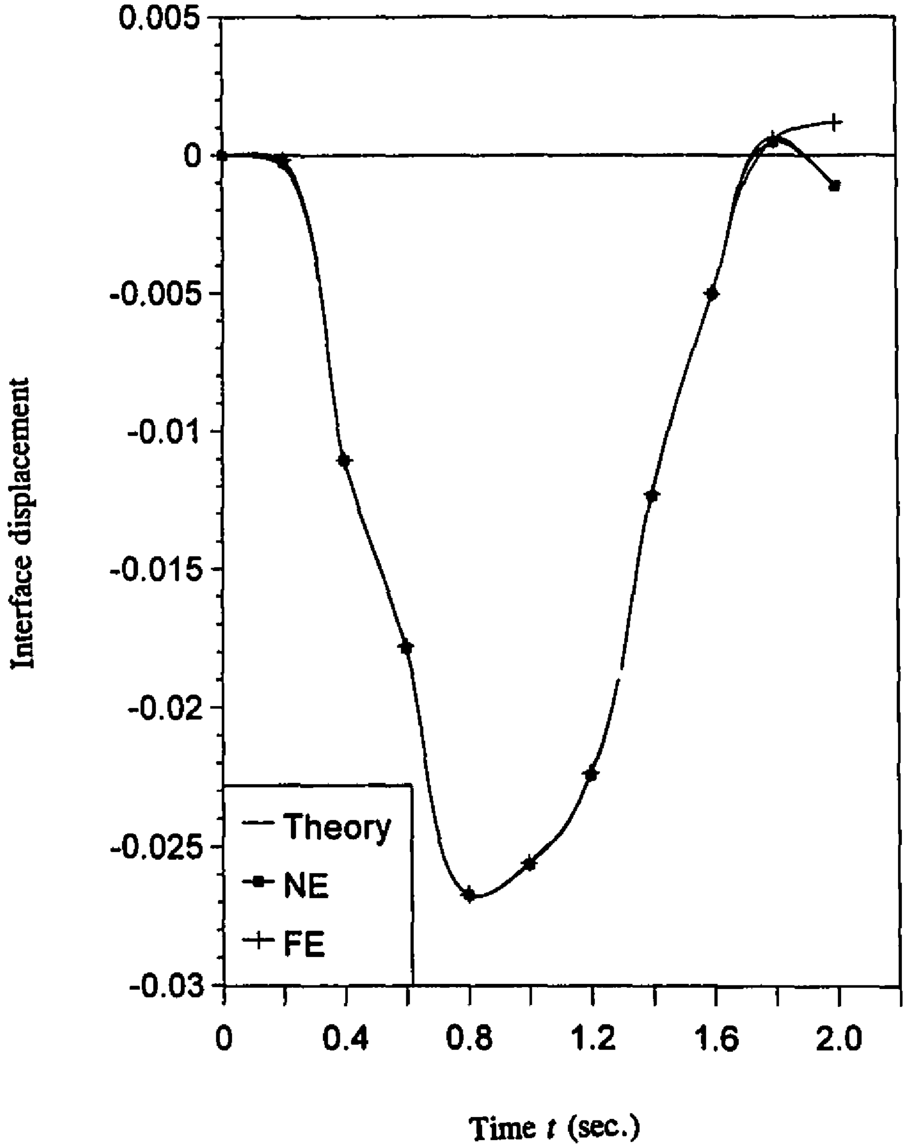


Figure 4.13 Interface displacements for a two materials disc, with soft material inside.

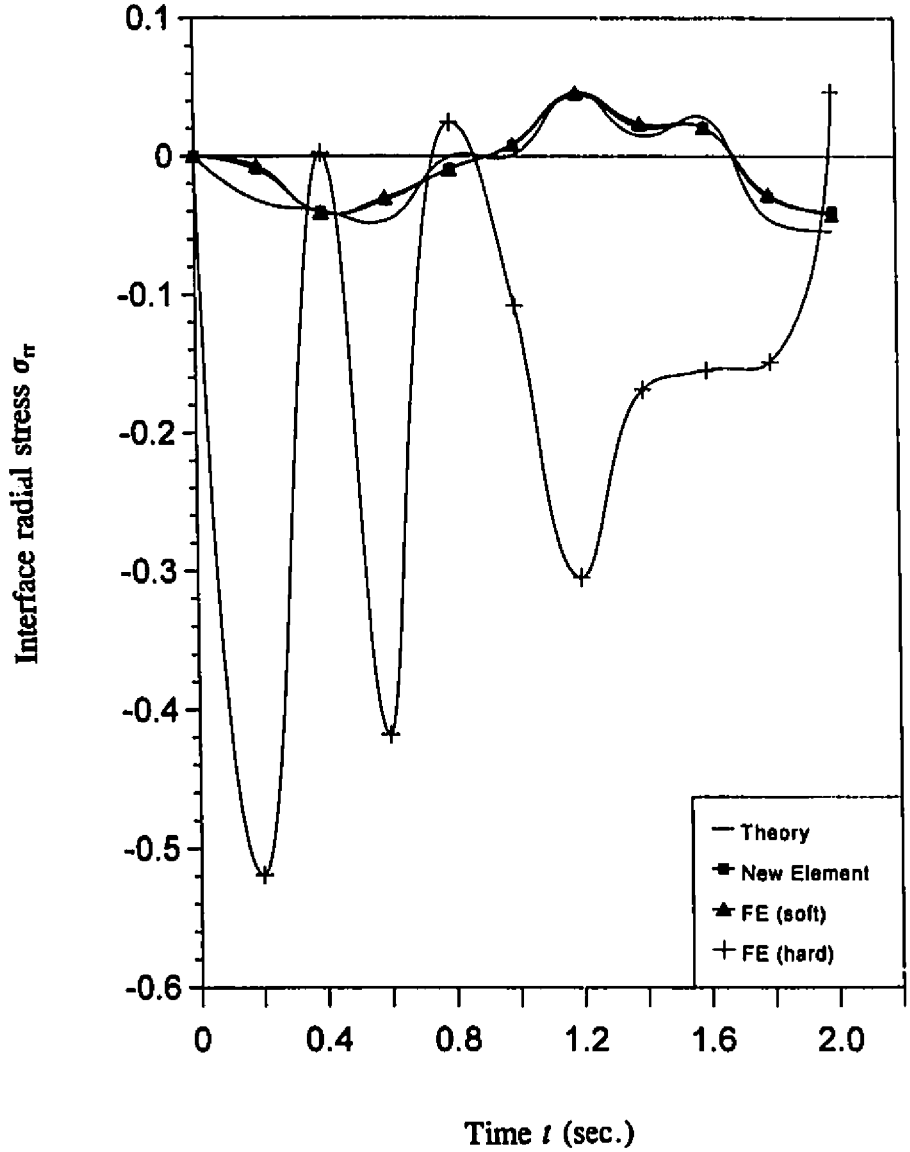


Figure 4.14 Radial stress at the interface of a two materials disc, with soft material inside.

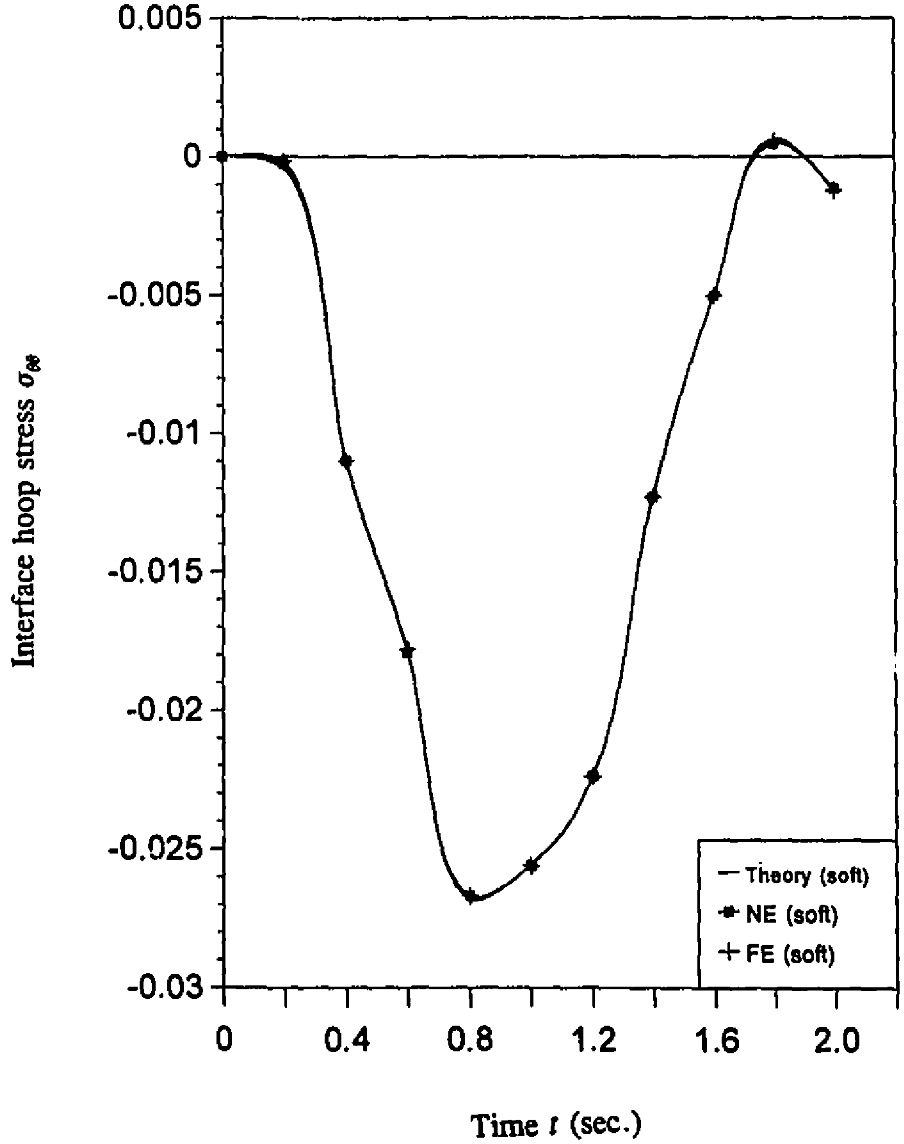


Figure 4.15 Interface hoop stress on the soft side of two material disc, with soft material inside.

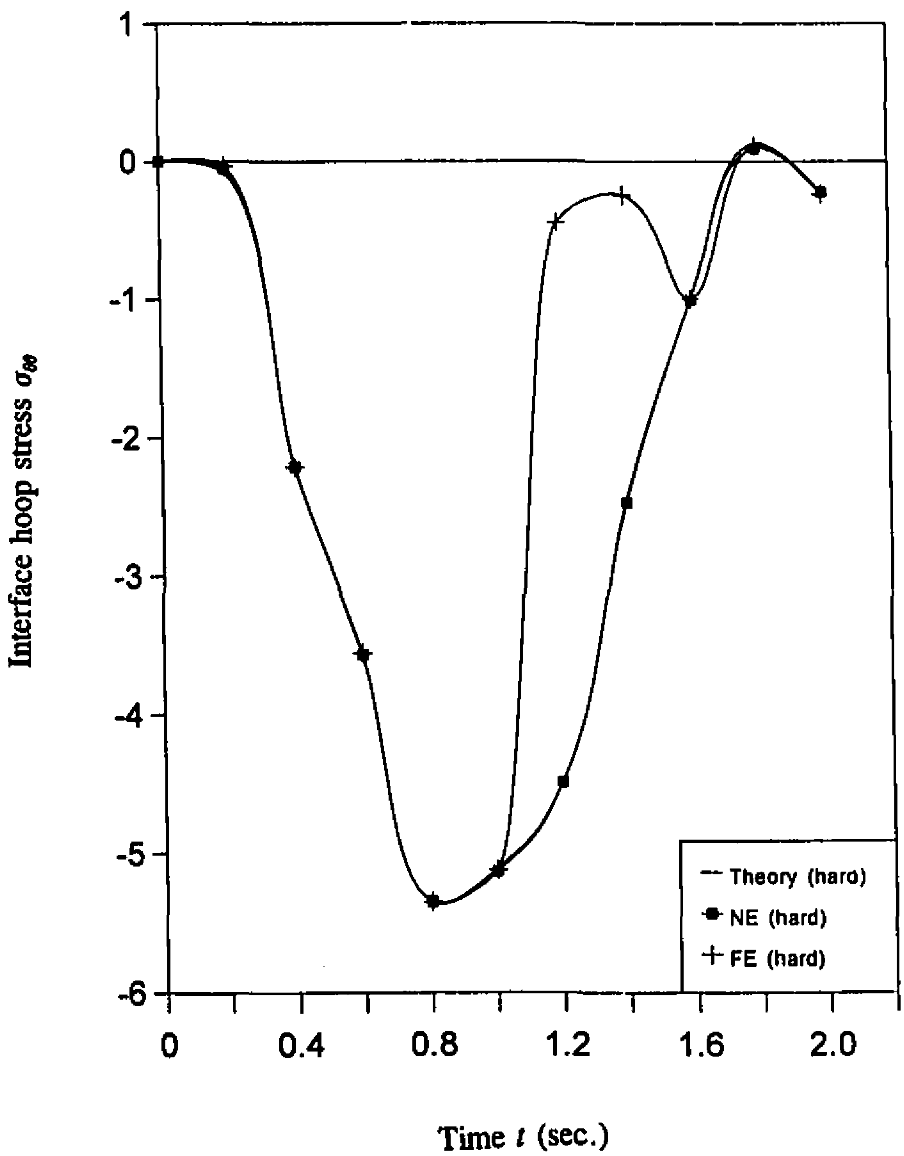


Figure 4.16 Interface hoop stress on the hard side of two material disc, with soft material inside.

the exact ones, that the present dynamic analysis program has been correctly implemented. Moreover, the new interface finite element correctly predicts the continuous and discontinuous interface stress components. Thus, the main objective of the present thesis has been accomplished.

The conventional finite element does give good results insofar as the soft-side of the interface is concerned. However, it cannot be used to obtain the hard-side stresses, particularly the interface stress σ for a two materials bar and σ_{rr} for a two materials disc problems if soft material is on the inside. This, then underscores the necessity of abandoning the conventional finite element, and using in its stead the new "stress-compatible" interface element.

Chapter 5

BIOMECHANICAL APPLICATION: DYNAMIC STRESS RESPONSE OF THE PROSTHESIS/CEMENT INTERFACE OF A KNEE TIBIAL COMPONENT

5.1 Introduction

As an illustration of the useful applications of the computer program developed in this thesis, the present chapter is devoted to an analytical investigation of the possible adverse effects of stress response caused by dynamic loads at the prosthesis/cement interfaces of a prosthetic human knee joint. In recent times, the replacement of fractured or degenerated human joints by artificial joint prostheses has become a common orthopaedic surgical operation. However, for the success of such operations, it is crucial that the fixation system is engineered to assure long-term integrity of the artificial joint. In this context, the present study is desirable because bond failure at prosthesis/cement interfaces of prostheses fixed with PMMA (Polymethylmethacrylate) bone cement, may induce wide-spread fracture of the cement and the eventual loosening of the prothesis.

A study on dynamic stresses at the prosthesis/cement interfaces has been conducted previously [11]. However, this study was conducted on the basis of conventional finite-element method, specifically by using the SAP IV [12] finite element program. Hence the interface stresses obtained in this study were ambiguous, and a choice had to be made to select the cement-side results as approximately correct, and ignore the metal-side stresses as incorrect. This choice therefore left the validity of the analytical results of this study in doubt. The present finite element program gives unique values of the interface stresses, and can now be used to check the validity of the previous results and conclusions.

There exist a variety of prosthesis designs and fixation systems. However, only a few have been investigated to determine their engineering efficiency. Figure 5.1 shows

Figure 5.1 A human knee joint after total knee arthroplasty.

a typical prosthetically reconstructed human knee joint. For the purpose of the present study, the focus is kept on the tibial part of a reconstructed knee joint, and the finite element model is simplified by assuming both the bone and prosthesis geometries as axisymmetric. Accordingly, the prosthesis is considered to consist of a circular metal plate with a central stem which is inserted in the centre of the tibia. The fixation is achieved by an intervening PMMA cement layer.

The loading and boundary conditions are also taken as axisymmetric, which then render the model problem amenable to be analyzed by the axisymmetric option of the present computer program.

5.2 Finite Element Model

The finite element idealization of the prosthetic tibia axisymmetric model is similar to the model employed by Shirazi-Adl and Ahmed [13], in their study of interface stresses under static loads. However, the mesh layout of the present study is slightly different.

5.2.1 Geometry

Dimensions of the tibia and prosthesis used in the analysis are shown in Fig. 5.2. The tibia portion is taken 40 mm long, and the base is considered fixed against vertical (but not horizontal) displacements. Ideally, a longer length representing the total tibia would have been preferable. However, this would have required more powerful computing resources in terms of memory and speed. A shorter length is justified on the ground that the most relevant information pertains to interface stresses at the time of first pass of the stress waves resulting from suddenly applied dynamic loads. In reality the stresses from subsequent passes resulting from reflection of the stress waves at the distal end boundary will be substantially damped out (because of the nature of the intervening cancellous bone) for the dynamic load variation considered in the present analysis.

The thickness of the metal plate is 2 mm and its radius is equal to 32 mm. The

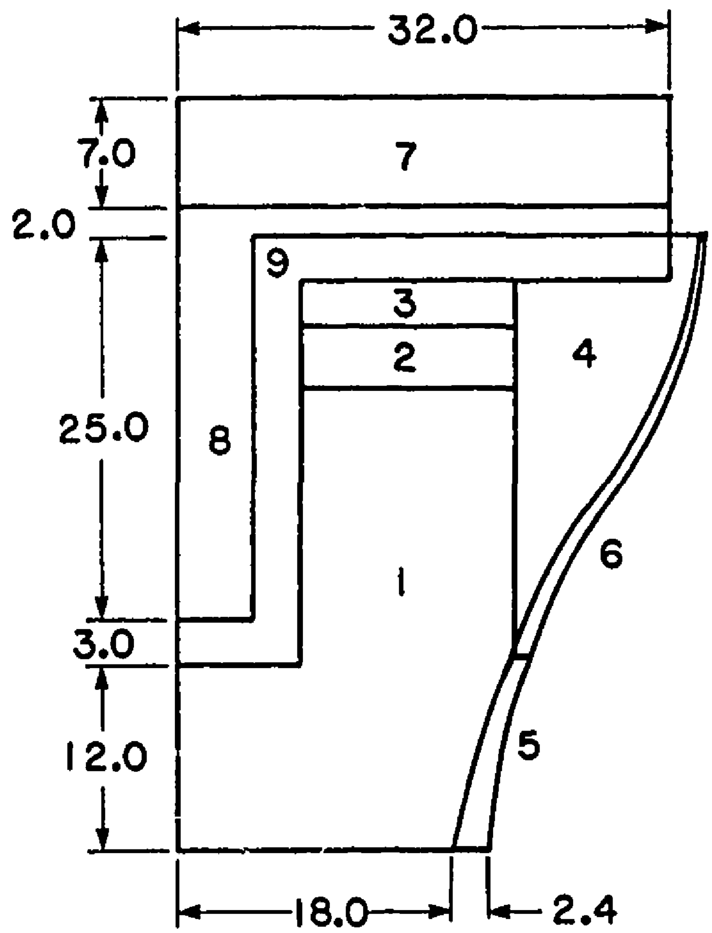


Figure 5.2 Geometrical dimensions (in mm) and the distribution of different material regions of the axisymmetric prosthetic tibia model. The numbers indicate the material regions whose properties are listed in Table 5.1.

metal plate lies entirely on the cancellous bone so that nowhere does it touch the cortical bone. This is a safe design assumption in view of the fact that it is very difficult to achieve and guarantee the contact of the prosthesis with the cortical bone. A prosthesis with long stem is reported to be more effective [19]; for this reason, a 25 mm long metal stem is used in the model. The thickness of the PMMA cement layer is taken to be 3 mm all around the prosthesis and cancellous bone.

5.2.2 Material Properties

Table 5.1 lists the different material properties with respect to Fig. 5.2 in which material regions are indicated by numbers which correspond to the material properties listed in Table 5.1. The distribution of the material properties in the model is in accordance with that reported by Goldstein et al. [14], being exactly the same as employed in [11]. All these materials are assumed to be linear elastic and isotropic.

5.2.3 Finite Element Mesh

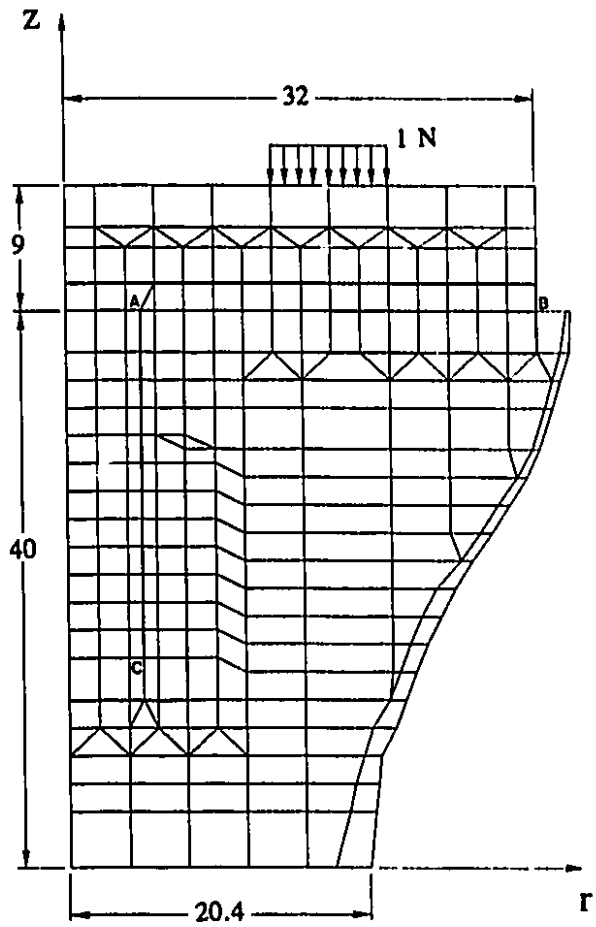
The model of resurfaced tibia is analyzed by the newly developed stress compatible interface elements and computer program. The finite element mesh shown in Fig. 5.3 is used in the present analyses. The mesh comprises mainly of isoparametric quadrilateral elements, and some triangular elements. Elements for which aspect ratio is greater than 3 are avoided wherever possible. The quadrilateral elements have the property of a linear displacement variation along their edges, whereas the triangular elements are based on a constant strain formulation. The new interface elements are used to model the prosthesis/cement interface. These elements have quadrilateral shapes and their formulation has been discussed in Chapter 2.

The model consists of 285 nodes and 280 axisymmetric ring elements of quadrilateral and triangular cross-sections. In the analysis by the newly developed program, 25 pairs of quadrilateral elements to form 25 interface elements are used.

5.2.4 Boundary Conditions and Loading

Number	Material type	Density (kg/m^3)	Young's modulus (MPa)	Poisson's ratio	Source reference
1	cancellous bone	280	50	0.2	[11,14]
2	cancellous bone	280	100	0.2	[11,14]
3	cancellous bone	280	150	0.2	[11,14]
4	cancellous bone	500	300	0.2	[11,14]
5	cortical bone	1800	14000	0.3	[11,15]
6	cortical bone	1800	7000	0.3	[11,15]
7	UHMWP Plastic	900	1000	0.35	[11,16]
8	stainless steel	8380	200000	0.3	[11,17]
9	PMMA bone cement	1100	2000	0.3	[11,18]

Table 5.1 Material properties used in the tibia model. The numbers refer to material regions shown in Fig. 5.2.



Dimensions are in mm

Figure 5.3 Axisymmetric finite element mesh of a tibia model. The dimensions are shown in Fig. 5.2. Horizontal interface, line AB, consists of nodes from 298 to 310. Vertical interface, line AC, consists of nodes from 286 to 297.

The boundary conditions constrain to fix the movement of the distal end of the tibia in the longitudinal (vertical) direction. In accordance with the axisymmetric condition, nodes along the axis of symmetry are fixed against radial movement.

The model simulates the load transmitted from the femur part as a ring load of compressive nature. For ease in interpretation of analysis results, the load is taken to be a compressive load of 1 N distributed uniformly over an annular ring of inside and outside radii of 14 mm and 22 mm respectively. The load is applied through a 7 mm thick UHMWP (Ultra-High Molecular Weight Polyethylene) plastic plate. This plate provides the articulating surface on the top of the horizontal metal plate of the prosthesis [13].

5.3 Dynamic Stress Response by Developed New Interface Element

This section presents some important results obtained by the dynamic analysis of the prosthetic tibia model, Fig. 5.3, under a suddenly applied (and then maintained) resultant unit load, i.e. a Heaviside step load of 1 N magnitude, Fig. 5.4.

As stated earlier, there are 25 pairs of interface elements employed in this study. The interface modelled by such elements is only that consisting of the steel stem on one side and the PMMA cement on the other. The horizontal part of this interface extends from nodes 298 to 310, whereas its vertical part extends from nodes 286 to 297. Since the differences in the densities and elastic moduli are greatest at this interface, the stresses due to dynamic load can be expected to be important at this interface.

As the loading is taken to be a Heaviside step loading, there is no need for step by step integration of the equations of motion (Section 3.6); the solution for any time t after the application of the load can be obtained in a single step. Thus, there is no accumulation of error as would be the case if the loading were a more complex one requiring step by step integration in small steps.

However, in order to obtain the history of deformation and stresses it is necessary to compute these quantities at suitable times t_1, t_2 ... etc. Since this thesis is concerned

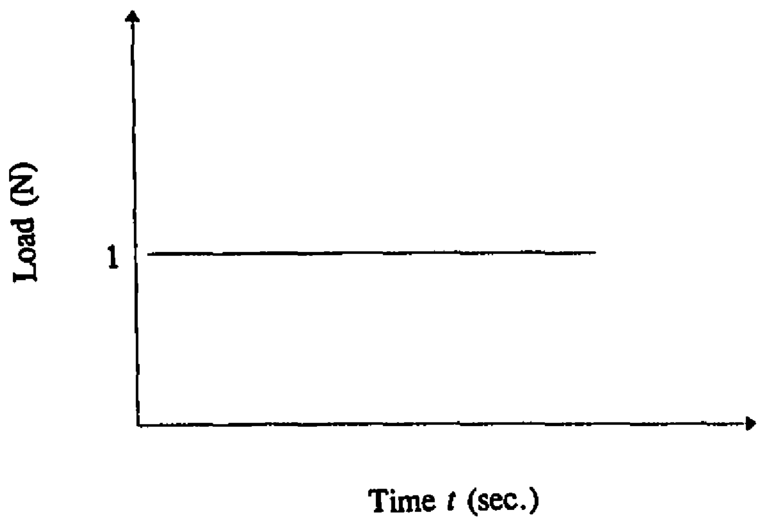


Figure 5.4 The suddenly applied load on the tibia model.

mainly with the theoretical development, and computer implementation of the newly formulated interface elements, we refrain from a comprehensive study of the present biomechanics problem, and instead restrict our attention to demonstrating the effectiveness of the new elements in obtaining useful results. For this reason we choose the history times as

$$t_1 = 3.0 \times 10^{-5} \text{ sec.} \quad \text{and} \quad t_2 = 6.0 \times 10^{-5} \text{ sec.}$$

The time t_1 is the approximate time the stress wave takes in travelling through steel from the top of the stem (near the load application point) to the bottom of the stem. The time $t_2 = 2t_1$ is the time by which the stress wave has arrived back at the top. These time periods are important from a practical point of view because of the following reason. Since the tibia materials, with the exception of steel, are highly dissipative, we will expect as stated earlier, that the effect of the suddenly applied load will be important for only a few (say 1 or 2) cycles of the stress wave travelling up and down the stem. Hence for realistic conclusions we confine ourselves to computing the relevant interface stress values at the above times of the first cycle.

In order to show the time-dependent dynamic stresses against the reference of their time-independent static values for the same magnitude of applied load, we also perform a static stress analysis according to the method and the program developed in this thesis. The present finite element model of the tibia has 540 unconstrained degrees of freedom. Hence the number of frequencies computed by the program is 540. All these frequencies have been used in obtaining the dynamic as well as the static solutions. We note the fact, reiterated earlier, that for a correct static solution, all computed frequencies and mode shapes must be used. Table 5.2 lists the first and last ten of the computed frequencies.

5.3.1 Horizontal Interface Stresses

The horizontal steel/cement interface is modelled by cement-side elements 201 to 213 and steel-side elements 220 to 232. For continuous components of stress it is immaterial which side of the interface the elements are considered. Figure 5.5 shows the

Number of first ten frequencies i	Value of frequencies ω_i	Number of last ten frequencies i	Value of frequencies ω_i
1	0.14210×10^5	531	0.16272×10^8
2	0.39922×10^5	532	0.16421×10^8
3	0.67109×10^5	533	0.16866×10^8
4	0.71073×10^5	534	0.17160×10^8
5	0.86424×10^5	535	0.17335×10^8
6	0.89966×10^5	536	0.17561×10^8
7	0.10308×10^6	537	0.17630×10^8
8	0.10697×10^6	538	0.17692×10^8
9	0.11500×10^6	539	0.18254×10^8
10	0.11637×10^6	540	0.18593×10^8

Table 5.2 The first and last ten of the computed frequencies of a prosthetic tibia model.

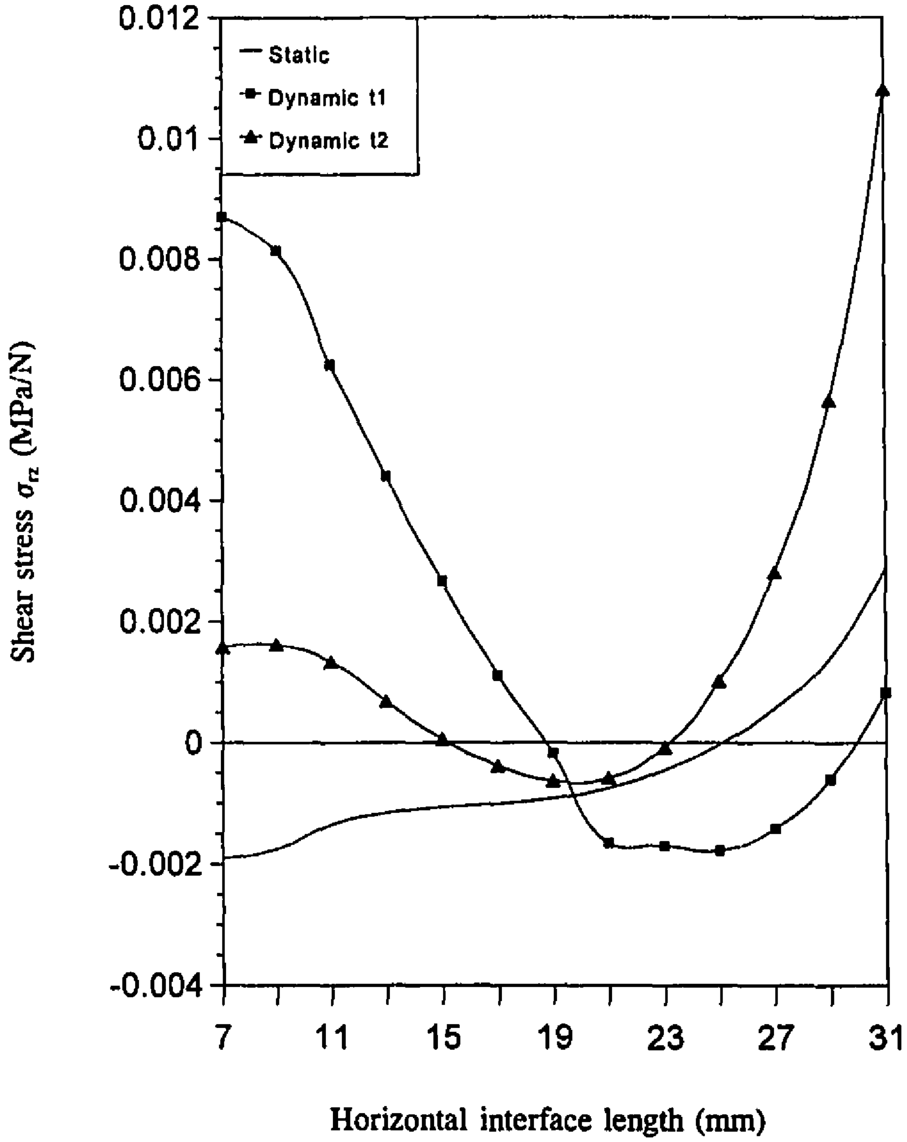


Figure 5.5 Shear stress σ_{rz} on the horizontal interface (line AB, Fig. 5.3).

variation of the interface shear stress component σ_{rz} at the two time values and also their values under static loading.

From this figure it is seen that the shear stress at time t_1 is higher by a factor more than 4 times and is of a different sign than its static value at the near-end (i.e. the point close to the axis of symmetry) of the prosthesis plate. On the other hand at the far-end (near the prosthesis plate edge), this shear stress is about one third of its static value and is of the same sign.

The above situation is reversed, in a sense, at time t_2. The shear stress at the far-end of the prosthesis plate is about 4 times the static value, and is of the same sign. In contrast, the near-end shear stress is nearly of the same magnitude as the static value, but of a different sign.

Figure 5.6 shows the variation of the normal interface stress at the horizontal interface. The results show similar dynamic effects as noted above for the interface shear stress. At time t_1, the stress is higher at the near-end, about two times the static value, and is of the opposite sign. The far-end value is almost equal to the static value both in magnitude and direction. At time t_2, it is the far-end which experiences higher stress, about three times the static value, although remaining of the same sign. At the near-end the stress value is almost equal to the static value.

Thus, at the horizontal interface, both the shear and normal stresses, particularly the former have much higher values than those computed for static loading. We will remark on the importance of these differences after examining the interface stresses at the vertical stem/cement interface.

5.3.2 Vertical Interface Stresses

The vertical interface is modelled by elements numbering 55 (bottom), 66, 77, 88, 99, 111, 123, 135, 150, 163, 176 and 199 (top) at steel side. The corresponding elements on the cement side are 56, 67, 78, 89, 100, 112, 124, 136, 151, 164, 177 and 200.

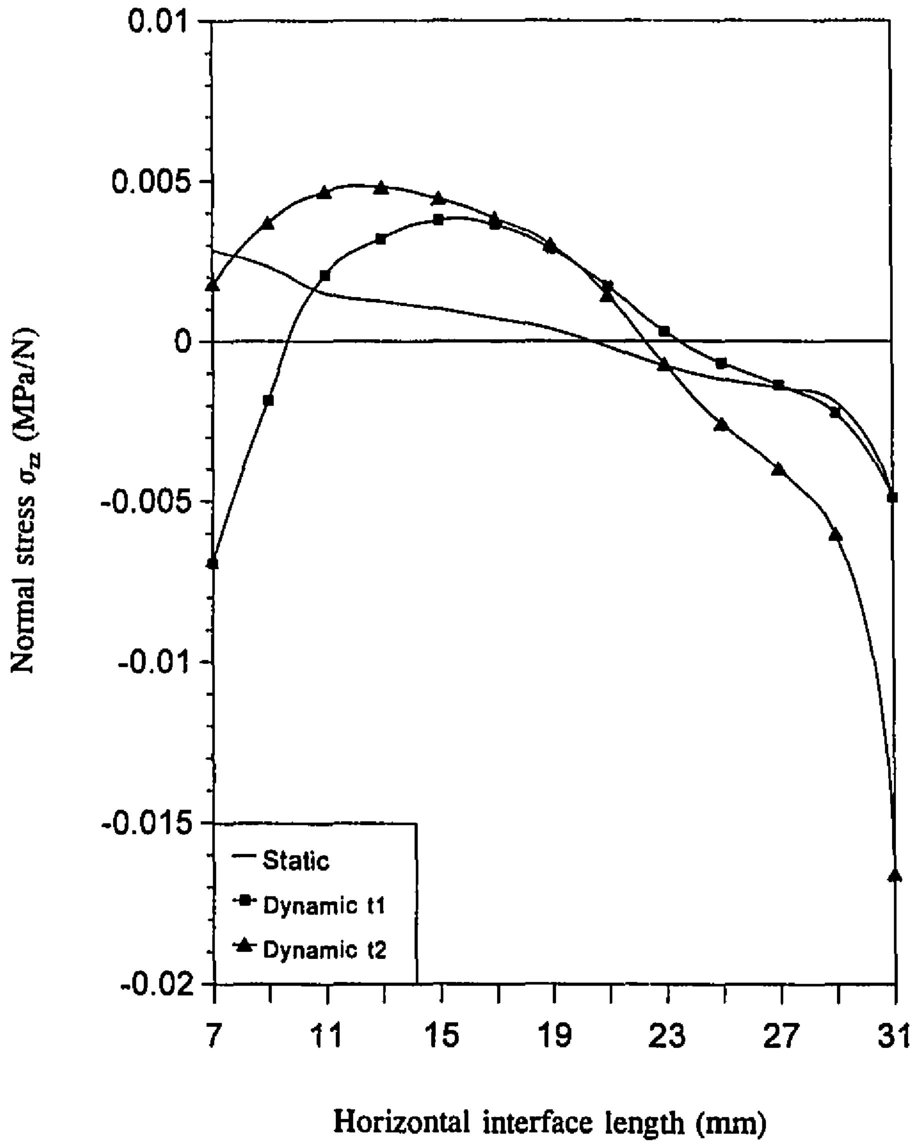


Figure 5.6 Normal stress σ_{zz} on the horizontal interface (line AB, Fig. 5.3).

The shear stress variation along the stem at the two times, along with their static part is shown in Fig. 5.7. We note that at time t_1, the bottom-end experiences a dynamic stress which is almost equal to the static value. However, at the top (near the point of load application), the shear stress is higher in magnitude by about 4 times, and of opposite sign than the static value.

At time t_2, it is the bottom-end which experiences a dynamic stress which is about 3 times higher than the static stress, although of the same sign. In contrast, the shear stress at the top-end experiences a value which is smaller by a factor of 3 than the static value.

Figure 5.8 shows the variation of the normal stress along the vertical interface. We note that these stresses are significantly different from their static values at time t_1, but not at time t_2. An important aspect of their variation at time t_1 is that they are of opposite sign for most of the interface length, and are about 2.5 times higher than the static value at the top-end. It is interesting to note that insofar as the vicinity of the bottom-end is concerned these stress do not show significant variation due to dynamic load.

5.4 Conclusions Resulting From the Interface Stress Analyses

The constraints on computer resources (i.e. the expenses of using the main-frame computer) and the time already spent in developing and verifying the computer program precluded a more thorough and comprehensive study of this biomechanics problem. Nevertheless, the above limited number of results do indicate the overall trend of the general results which may be expected by a more detailed analysis. The significance of these results and their trends can be summarized as follows.

(1) For the present finite element model of a knee joint with a stem-type prosthesis, the dynamic interface shear stress can be expected to be about 3 to 4 times higher than the static ones, both at the horizontal and vertical interfaces.

(2) The most vulnerable points (insofar as high shear stress is concerned) are the

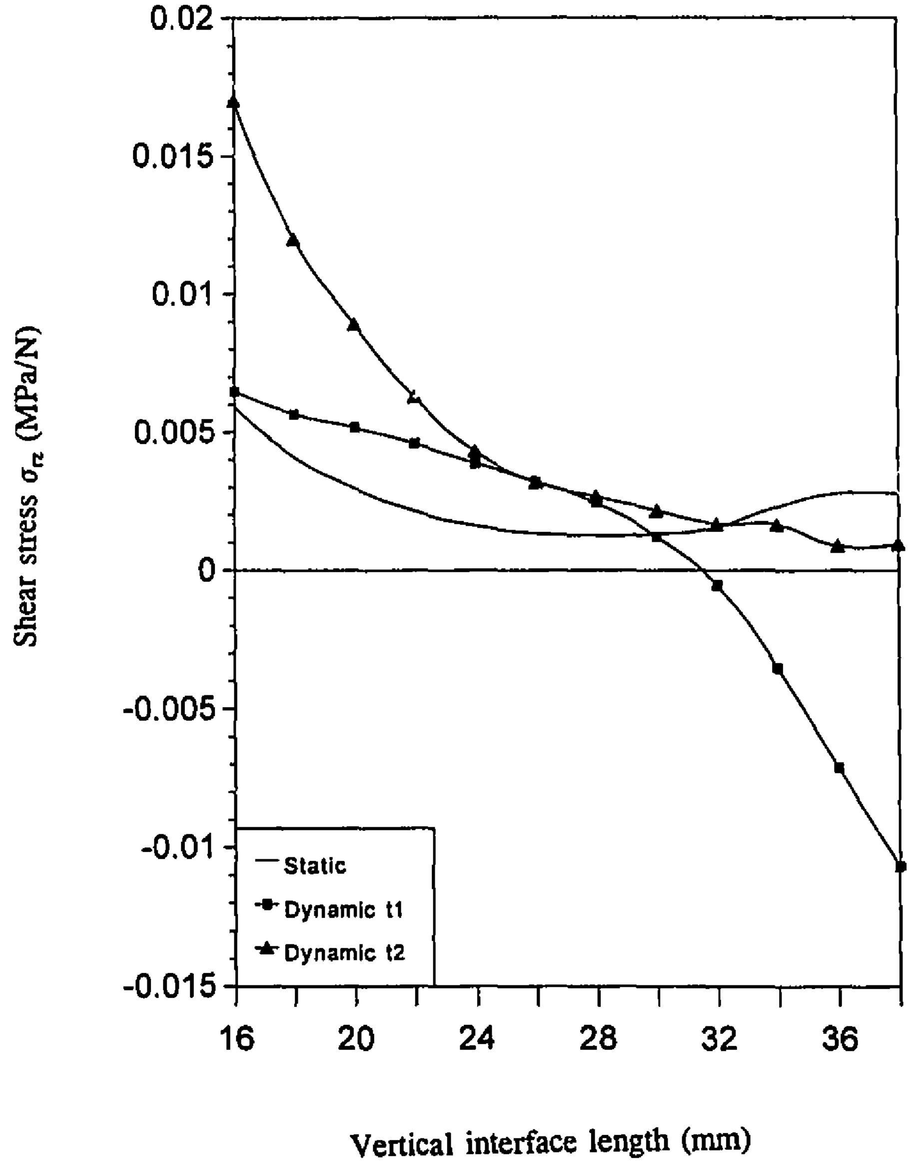


Figure 5.7 Shear stress σ_{rz} on the vertical interface (line AC, Fig. 5.3).

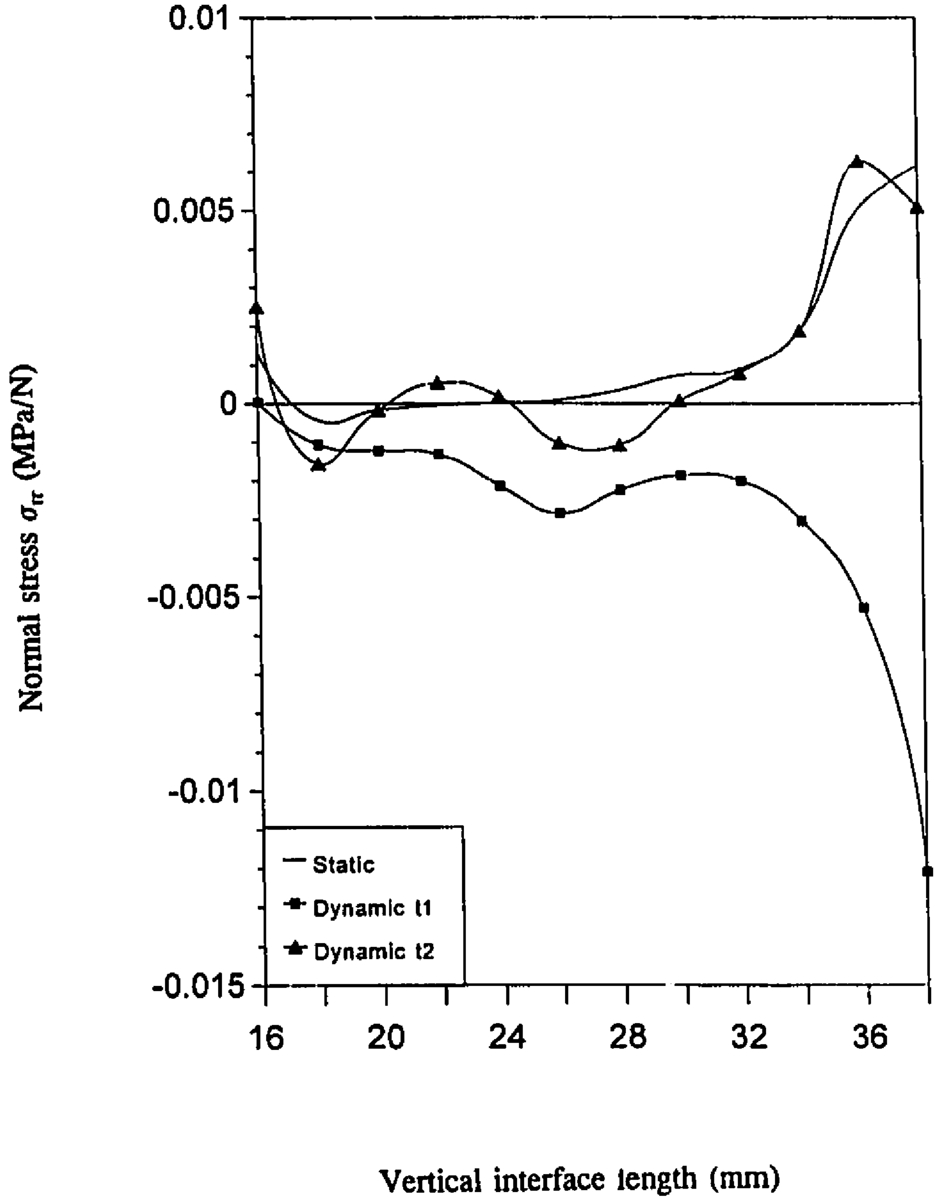


Figure 5.8 Normal stress σ_{rr} on the vertical interface (line AC, Fig. 5.3).

three corner points, viz., the near and far ends of the horizontal prosthesis plate, and the bottom end of the stem.

(3) The dynamic interface normal stress can be expected to be about 2 to 3 times higher than the static values. However these high stresses occur mostly at the far and near ends of the horizontal prosthesis plate. The bottom end of the stem appears to be not affected in a significant way by these stresses.

(4) Although in an actual situation of a patient, the loading cannot be expected to be as severe as the step loading considered here, the high values of the stresses indicate that they would be relatively high even for more gradually applied dynamic loads.

(5) The high values of the interface stresses (particularly for the shear stresses) above their static values, suggest a fatigue-type of failure of the steel/PMMA interface bond. The site of these failures is predicted by the present analysis to be the three corner points.

We close this chapter by recalling the conclusion reached in the previously referred study [11]. In that study, which was conducted on the basis of conventional finite elements, cement-side stresses were considered to be the correct interface stresses. They indicated that for a ramp-type loading of rise-time 2×10^{-3} sec, the interface stresses differed from their static part by about $\pm$ 30%. In contrast, the conclusions of the present study, reached on the basis of correct interface stresses, make it clear that dynamic stresses can in general be much higher, as much as four times the static values for the suddenly applied dynamic loading.

Chapter 6

SUMMARY AND CONCLUSION

This thesis formulates and implements new finite elements for obtaining correct values of the continuous and discontinuous components of stress and strain tensor at points of bimaterial interfaces of highly dissimilar materials under dynamic loadings.

Two types of elements have been formulated, one for two-dimensional (plane stress and plane strain) problems, and the other for three-dimensional axisymmetric problems. The elements have been incorporated in a full-fledged dynamic analysis finite element programs, constructed on the basis of programs previously constructed by Angelides [8] for static analysis. The constructed programs are capable of solving stress analysis problems of the above categories under general dynamic loading. The interface elements are 6 node composite elements, consisting of two quadrilateral element belonging to the two different materials. Node 7 is taken at the centre of the interface boundary. However displacements of node 7 are not independent; they are chosen so as to satisfy the continuity of the stress vector at this one point. Thus, displacements of node 7 are not included as independent parameters. The equations of motion are solved by employing the mode superposition technique. This technique has the advantage of providing natural frequencies of the system, and can be used to obtain a series of increasingly accurate solution depending on the number of frequencies used. Damping is neglected entirely. Although the present elements and programs are specially designed for dynamic stress analysis, they can also be easily used for static analysis as has been discussed and shown in Chapter 3.

Performance of both plane and axisymmetric elements has been validated by obtaining exact solutions of some simple wave-propagation problems (Appendices A and B) and comparing the finite element solutions with them. The agreement is excellent first between the values of the natural frequencies, and second between the values of the continuous and discontinuous stress and strain components at the interface points. Thus, the new elements not only fulfil their expected usefulness, but also do not affect the analysis results away from the interface in an adverse way. They can therefore be used with confidence in analyzing general dynamic problems. To the author's knowledge there are no elements of this type available either in the literature or in any commercial finite element codes.

The fact that the stress continuity is enforced by appropriately choosing the displacements of the interface node, means that the new interface elements are similar to the conventional elements in their formulation and implementation. Therefore, they can be implemented in any displacement based finite element code rather easily. The only significant difference consists in computing the [L] matrices of Chapter 2.

The usefulness of the new elements and programs developed in this work is demonstrated by analyzing an axisymmetric model of a prosthetic tibia under a step-jump loading. The prosthesis consists of a circular metal plate with a central stem. The prosthesis is fixed in the bone of the joint via an intervening layer of PMMA bone cement. The analysis is aimed at calculating the correct interface stresses at the cement/metal interface under the above type of loading. It has been shown that in general the interface stresses under dynamic loads are considerable different, to the tune of three to four times their static values. The sites at which the differences are the highest have been identified to be the prosthesis-edge, prosthesis-stem junction, and the bottom end of the prosthesis stem. Fatigue fracture of the PMMA may therefore occur at these points.

An obvious suggestion for further research in this topic is to extend the present formulation and implementation to three-dimensional solid interface elements. This will undoubtedly increase the scope of the interface problems which can then be analyzed.

References

[1] Shrivastava, S.C., Ahmed, A.M., Shirazi-Adl, A., and Burke, D.L., "Effect of a Cement-Bone Composite Layer and Prosthesis Geometry on Stresses in a Prosthetically Resurfaced Tibia," *Journal of Biomedical Materials Research*, Vol. 16, pp.929-949, 1982.

[2] Cook, R.D., *Concepts and Applications of Finite Element Analysis,* 3rd Edition, John Wiley & Sons, New York, 1989.

[3] Herrmann, L.R., "Improved Stress Calculation for Simple Quadrilateral Elements," *Computers and Structures*, Vol. 6, No. 2, pp.141-148, 1976.

[4] Hinton, E. and Campbell, J.S., "Local and Global Smoothing of Discontinuous Finite Element Functions Using a least Squares Method," *International Journal of Numerical Methods in Engineering,* Vol. 8, No. 3, pp. 461-480, 1974.

[5] Loubignac, G., Cantin, G., and Touzot, G., "Continuous Stress Fields in Finite Element Analysis," Technical Notes, *AIAA Journal,* Vol. 15, No. 11, pp. 1645-1647, 1977.

[6] Salama, M. and Utku, S., "Stress Computation in Displacement Methods for Two-Material Elastic Media," *Computer Methods in Applied Mechanics and Engineering,* Vol. 10, pp. 325-338, 1977.

[7] Soh, A.K., "A Modified Finite Element Technique for the Determination of Interfacial Stresses in a Composite Material," *Proceedings of International Conference on Finite Elements in Computational Mechanics,* Pergamon Press Ltd., Bombay, India, pp.249-257, 1985.

[8] Angelides, Michael, "*Stress Compatible Finite Elements for Bimaterial Interface Problems,*" M.Eng. Thesis, McGill University, Montreal, Canada, 1987.

[9] IMSL MATH/LIBRARY, *Fortran Subroutines for Mathematical Applications,* Vol. 1, pp. 394-396, IMSL Inc., 1989.

[10] Rao, S.S., *The Finite Element Method in Engineering,* 2nd Edition, 1989.

[11] Ahmed, A.M., Tissakht, M., Shrivastava, S.C., Chan, K., "Dynamic Stress Response of the Implant/Cement Interface: An Axisymmetric Analysis of a Knee Tibial Component," *Journal of Orthopaedic Research 8,* pp. 435-447, 1990.

[12] Bathe KJ, Wilson EL, Peterson FE: SAP IV: *A Structural Analysis Program for Static and Dynamic Response of Linear Systems.* Earthquake Engineering Research Center, EERC 73-11, University of California, Berkeley, 1973.

[13] Shirazi-Adl, A., Ahmed, A.M., "Micro-Motions at the Bone/Prosthesis Interface in Porous-Surface Metal Tibial Implants - An Axisymmetric Finite Element Study," *Transactions of the 32nd Annual Meeting of the Orthopaedic Research Society,* ORS, Chicago, Vol. 11, p. 262, 1986.

[14] Goldstein, S.A., Wilson, D.L., Sonstegard, D.A., Mathews, L.S., "The Mechanical Properties of Human Tibial Trabecula Bone as a Function of Metaphyseal Location," *Journal of Biomechanics,* Vol. 16, pp. 965-969, 1983.

[15] Murray, R.P., Hayes, W.C., Edwards, W.T., Harry, J.D., "Mechanical Properties of the Subchondral Plate and the Metaphyseal Shell," *Transactions of the 30th Annual Meeting of the Orthopaedic Research Society,* ORS, Chicago, Vol. 9, p. 197, 1984

[16] Parker, E.B., *Materials Data Book,* McGraw-Hill, New York, 1967.

[17] Popov, E.P., *Mechanics of Materials,* 2nd Edition, Prentice Hall Inc., Englewood Cliffs, NJ, 1976.

[18] Haas, S.S., Brauer, G.M., Dickson, M.A., " A Characterization of Polymethyl methacrylate Bone Cement," *Journal of Bone and Joints Surgery,* Vol. 57-A, pp. 380-391, 1975

[19] Murase, K., Crowninshield, R.D., Pederson, D.R., Chang, T-S., "An analysis of Tibial Component Design in Total Knee Arthroplasty," *Journal of Biomechanics,* Vol. 16, pp. 13-22, 1982.

[20] Timoshenko, S.P., and Goodier, J.N., *Theory of Elasticity,* 3rd Edition, McGraw-Hill, New York, 1970.

[21] McLachlan, N.W., *Complex Variable Theory and Transform Calculus,* Cambridge University Press, 1963.

APPENDIX A

Wave Propagation in a Composite Bar

For the purpose of validating the constructed computer program, we solve, exactly and analytically to a limited extent, some simple problems of wave propagation in composite structures.

The objective of this appendix is to present a solution of wave propagation problem in a composite bar. For convenience of analysis, the origin is taken at the interface. The subscripts or superscripts 1 and 2 refer to the materials at the left and right of the interface. Assuming only the axial behaviour, and no damping, the equations of motion can be written, in non-dimensional forms, as:

$$\frac{\partial^2 \bar{u}_1}{\partial \xi^2} = \frac{\partial^2 \bar{u}_1}{\partial \tau^2}, \qquad \frac{\partial^2 \bar{u}_2}{\partial \xi^2} = \kappa^2 \frac{\partial^2 \bar{u}_2}{\partial \tau^2} \tag{A.1}$$

where the variables are rendered non-dimensional as follows:

$$\begin{gathered} x = a\xi, \quad t = (a/c_1)\tau \\ u_1(x,t) = a\bar{u}_1(\xi,\tau), \quad u_2(x,t) = a\bar{u}_2(\xi,\tau) \end{gathered} \tag{A.2}$$

Obviously, $u_1(x,t)$ and $u_2(x,t)$ denote the axial displacement in the left and the right materials, and

$$\kappa = c_1/c_2 = (E_1\rho_2/E_2\rho_1)^{1/2} \tag{A.3}$$

is the ratio of the velocities of one-dimensional wave propagation in the two materials. The displacement and force continuities at the interface require that

$$\bar{u}_1(0,\tau) = \bar{u}_2(0,\tau), \quad \frac{\partial \bar{u}_1}{\partial \xi}(0,\tau) = \gamma \frac{\partial \bar{u}_2}{\partial \xi}(0,\tau) \tag{A.4}$$

where $\gamma = E_2A_2/E_1A_1$. The right end $x = a$ (or $\xi = 1$), is considered free and is subjected to say a compressive step loading, while the left end $x = -b$ (or $\xi = b/a = \alpha$), is considered supported. The boundary conditions are therefore:

$$\frac{\partial \bar{u}_2}{\partial \xi}(1,\tau) = -\bar{\sigma}_0 H(\tau - 0), \quad \bar{u}_1(\alpha,\tau) = 0 \tag{A.5}$$

where

$$\bar{\sigma}_0 = \frac{\sigma_0}{E_2} \tag{A.6}$$

and $H(\tau - 0)$ denotes the Heaviside step function having the property that $H(\tau - 0) = 1$ for $t > 0$ and $H(\tau - 0) = 0$ for $t < 0$.

Taking the Laplace Transform with respect to time τ, and solving the resulting ordinary differential equations, the transforms of the displacements can be written as:

$$\bar{u}_1^* = M_1\cosh(\xi s) + N_1\sinh(\xi s), \quad \bar{u}_2^* = M_2\cosh(\kappa\xi s) + N_2\sinh(\kappa\xi s) \tag{A.7}$$

where s is the transform variable, and a superscript asterisk signifies the Laplace Transform. The four constants introduced above can be determined by the following four transformed boundary conditions:

$$\bar{u}_1^* = 0 \tag{A.8}$$

at the supported end $\xi = -\alpha$,

$$\frac{\partial \bar{u}_2^*}{\partial \xi} = -\frac{\bar{\sigma}_0}{s} \tag{A.9}$$

at the free end $\xi = 1$, and

$$\bar{u}_1^* = \bar{u}_2^*, \qquad \frac{\partial \bar{u}_1}{\partial \xi} = \gamma \frac{\partial \bar{u}_2}{\partial \xi} \tag{A.10}$$

at the interface $\xi = 0$. The transformed solutions for displacements and stresses are therefore expressible as:

$$\bar{u}_1^* = \frac{-\gamma \bar{\sigma}_0}{s^2} \left[\frac{\sinh[(\xi + \alpha)s]}{\gamma\kappa \sinh(s\kappa\alpha)\sinh(s\alpha) + \cosh(s\kappa\alpha)\cosh(s\alpha)} \right] \tag{A.11}$$

$$\bar{u}_2^* = \frac{-\bar{\sigma}_0}{\kappa s^2} \left[\frac{\gamma\kappa \cosh(\xi s\kappa)\sinh(s\alpha) + \sinh(\xi s\kappa)\cosh(s\alpha)}{\gamma\kappa \sinh(s\kappa\alpha)\sinh(s\alpha) + \cosh(s\kappa\alpha)\cosh(s\alpha)} \right] \tag{A.12}$$

$$\bar{\sigma}_1^* = \frac{-A_2 \sigma_0}{s A_1} \left[\frac{\cosh[(\xi + \alpha)s]}{\gamma\kappa \sinh(s\kappa\alpha)\sinh(s\alpha) + \cosh(s\kappa\alpha)\cosh(s\alpha)} \right] \tag{A.13}$$

$$\bar{\sigma}_2^* = \frac{-\sigma_0}{s} \left[\frac{\gamma\kappa \sinh(\xi s\kappa)\sinh(s\alpha) + \cosh(\xi s\kappa)\cosh(s\alpha)}{\gamma\kappa \sinh(s\kappa\alpha)\sinh(s\alpha) + \cosh(s\kappa\alpha)\cosh(s\alpha)} \right] \tag{A.14}$$

The solutions with respect to time now requires finding the inverse Laplace transform, which can be obtained from the residues at the poles which are zeros of the denominator:

$$s[\gamma\kappa \sinh(s\kappa\alpha)\sinh(s\alpha) + \cosh(s\kappa\alpha)\cosh(s\alpha)] = 0 \tag{A.15}$$

The corresponding equation giving the natural frequencies, ω, of free vibration of a composite bar of length L_1 and L_2, is obtained by putting $s = i\omega$ (where $i = (-1)^{1/2}$) in the above equation:

$$-\gamma\kappa \sin(\omega\kappa\alpha)\sin(\omega\alpha) + \cos(\omega\kappa\alpha)\cos(\omega\alpha) = 0 \tag{A.16}$$

Although we may obtain solution for all points ($-\alpha \leq \xi \leq 1$), we restrict our attention to the interface ($\xi = 0$) point, and obtain the following solutions for time variations of interface displacement and stresses, upon using the inversion theorem and the calculus of residues:

$$\bar{u}|_{\xi=0} = -\gamma\bar{\sigma}_0\left[\alpha-\Sigma\frac{1}{\omega_i^2}\frac{2\sin(\alpha\omega_i)\cos(\omega_i\tau)}{(\gamma\kappa^2+\alpha)\cos(\alpha\kappa\omega_i)\sin(\alpha\omega_i)+(\gamma\kappa\alpha+\kappa)\sin(\alpha\kappa\omega_i)\cos(\alpha\omega_i)}\right] \tag{A.17}$$

$$\bar{\sigma}_2|_{\xi=0} = -\bar{\sigma}_0\left[1-\Sigma\frac{1}{\omega_i}\frac{2\cos(\alpha\omega_i)\cos(\omega_i\tau)}{(\gamma\kappa^2+\alpha)\cos(\alpha\kappa\omega_i)\sin(\alpha\omega_i)+(\gamma\kappa\alpha+\kappa)\sin(\alpha\kappa\omega_i)\cos(\alpha\omega_i)}\right] \tag{A.18}$$

$$\bar{\sigma}_1 = \frac{A_2}{A_1}\bar{\sigma}_2 \tag{A.19}$$

where we recall that ω_i are the natural frequencies of the composite bar, and $\tau = (a/c_1)t$.

As a special case, we may also obtain solution for a one material bar (material 1 = material 2) by putting $\alpha = 0$, $\gamma = 1$, and $\kappa = 1$ in Eqs. (A.16) to (A.18). In this case, the frequency equation specializes to

$$\omega_i = \frac{(2i-1)\pi}{2} \tag{A.20}$$

The time variations of free end ($\xi = 1$) non-dimensional displacement, and the supported end ($\xi = 0$) non-dimensional stress are found to be

$$\bar{u}_2|_{\xi=1} = -\bar{\sigma}_0\left[1+\Sigma\frac{4\cos[(2i-1)\tau/\pi]}{(2i-1)^2\pi^2}\right] \tag{A.21}$$

$$\bar{\sigma}_2|_{\xi=0} = -\bar{\sigma}_0\left[1-\Sigma(-1)^i\frac{4\cos[(2i-1)\tau/\pi]}{(2i-1)\pi}\right] \tag{A.22}$$

Results given by Eq. (A.22) agree with those in Timoshenko and Goodier [20].

APPENDIX B

Wave Propagation in a Composite Disc

As mentioned previously, the purpose of this Appendix is to obtain exact (series) solutions to simple axisymmetric problems to validate the axisymmetric interface elements. These solutions, although relatively simple, are non-trivial enough to test the validity of finite element solutions. The problem of a composite disc worked out here appears to be unavailable in the literature.

B.1 One Material Disc Under a Step Loading

We first solve the simpler problem of axisymmetric dynamic response of a plane, one material circular disc under a step-rise of radial pressure at the outer periphery. The differential equation governing the motion is

$$\frac{\partial \sigma_{rr}}{\partial r} + \frac{\sigma_{rr} - \sigma_{\theta\theta}}{r} = \rho \frac{\partial^2 u}{\partial t^2} \tag{B.1}$$

Assuming a small-strain, linear elastic, isotropic material behaviour the stress-strain-displacement relations are

$$\begin{aligned} \sigma_{rr} &= E\epsilon_{rr} = E\frac{\partial u}{\partial r} \\ \sigma_{\theta\theta} &= E\epsilon_{\theta\theta} = \frac{Eu}{r} \end{aligned} \tag{B.2}$$

where to simplify the analysis we have assumed that Poisson's ratio is zero. Hence the governing differential equation, in terms of the radial displacement u, is

$$\frac{\partial^2 \bar{u}}{\partial \xi^2} + \frac{1}{\xi}\frac{\partial \bar{u}}{\partial \xi} - \frac{\bar{u}}{\xi^2} = \frac{\partial^2 \bar{u}}{\partial \tau^2} \tag{B.3}$$

where we have made the following change of variables to render the differential equation non-dimensional:

$$u = a\bar{u}, \quad r = a\xi, \quad t = \frac{a}{c}\tau \tag{B.4}$$

in which a is the outer radius, and

$$c = \sqrt{\frac{E}{\rho}} \tag{B.5}$$

is the velocity of wave propagation (in a one-dimensional bar under axial loading). The boundary conditions of the problem are

$$\begin{aligned} \bar{u}(0,t) &= 0 \\ \frac{\partial \bar{u}}{\partial \xi}(1,t) &= -\frac{p_0}{E}H(t-0) \end{aligned} \tag{B.6}$$

where p is the applied pressure, and $H(t\text{-}0)$ is the Heaviside unit-step function defined by

$$\begin{aligned} H(t-0) &= 0 \quad for \quad t < 0 \\ H(t-0) &= 1 \quad for \quad t > 0 \end{aligned} \tag{B.7}$$

The disc is thus loaded by a suddenly pressure applied at the boundary $r = a$. The initial conditions are:

$$\bar{u}(\xi,0) = \frac{\partial \bar{u}}{\partial \tau}(\xi,0) = 0 \tag{B.8}$$

As before, we use the method of Laplace Transform to solve this problem of step-jump dynamic loading. The differential equations and boundary conditions become

$$\frac{d^2\bar{u}^*}{d\xi^2} + \frac{1}{\xi}\frac{d\bar{u}^*}{d\xi} - \left(\frac{1}{\xi^2} + s^2\right)\bar{u}^* = 0 \tag{B.9}$$

$$\bar{u}^* = 0$$
$$\frac{\partial \bar{u}^*}{\partial \xi}(1) = -\frac{p}{Es} \tag{B.10}$$

where s is the transform variable (in lieu of τ) and a superscript asterisk signifies the Laplace Transform of displacement or stresses.

The above differential equation is a Bessel equation, the solution of which, taking into account the boundary conditions, is

$$\bar{u}^* = -\frac{pI_1(s\xi)}{Es^2 I_1'(s)}$$
$$\bar{\sigma}_{rr}^* = -\frac{p}{Es}\frac{I_1'(s\xi)}{I_1'(s)} \tag{B.11}$$

where I_1 is one of the modified Bessel functions of first order, and $I_1'(s\xi)$ denotes the derivative of $I_1(s)$ evaluated at $s\xi$. Accordingly, the transforms of the stresses at the centre ($\xi = 0$), and the displacement at the periphery ($\xi = 1$) are

$$\bar{\sigma}_{rr}^*(0) = \bar{\sigma}_{\theta\theta}(0)^* = -\frac{p}{2Es}\frac{1}{I_1'(s)} \tag{B.12}$$

$$\bar{u}^*(1) = -\frac{p}{Es^2}\frac{I_1(s)}{I_1'(s)} \tag{B.13}$$

The transforms can be inverted using the Residue Theorem [21]. We find (with the help of Mathematica program of Wolfram Research Inc., Champaign, Ill.) that the denominator becomes zero at

$$s_0 = 0$$
$$s_1 = \pm\ 1.84118\ i$$
$$s_2 = \pm\ 5.33144\ i$$
$$s_3 = \pm\ 8.53632\ i$$

$$s_4 = \pm\ 11.7060\ i$$
$$s_5 = \pm\ 14.8636\ i$$
$$s_6 = \pm\ 18.0155\ i$$
$$s_7 = \pm\ 21.1644\ i$$
$$s_8 = \pm\ 24.3113\ i$$
$$s_9 = \pm\ 27.4571\ i$$
$$s_{10} = \pm\ 30.6019\ i$$

It can be seen that these points are simple poles. The solution for the stress components at the origin is then obtained as

$$\bar{\sigma}_{rr}(0) = \bar{\sigma}_{\theta\theta}(0) = -\frac{p}{2E}[\ 2-2.64798\cos(1.84118\tau)+1.2332\cos(5.33144\tau) - \ .....\ ] \tag{B.14}$$

where

$$\sigma_{rr} = E\bar{\sigma}_{rr}, \qquad \sigma_{\theta\theta} = E\bar{\sigma}_{\theta\theta} \tag{B.15}$$

Similarly, the radial displacement of the periphery is obtained as

$$\bar{u}(1) = -\frac{p}{E}[\ 1-0.836835\cos(1.84118\tau)-0.07299281\cos(5.33144\tau) - \ .....\ ] \tag{B.16}$$

where

$$u = a\bar{u} \tag{B.17}$$

In Chapter 4, the finite element solutions are compared with these solutions to ascertain the former's validity.

We also note that the natural frequencies (in radians per second) of the disc under the boundary conditions, $\sigma_{rr}(1) = u(0) = 0$, are

$$\omega_k^2 = -\frac{c^2}{a^2} s_k^2 \tag{B.18}$$

where s_k are the roots given above. These are also used in Chapter 4 for comparing the frequencies obtained by the finite element method.

B.2 Two-Materials Composite Disc Under Suddenly Applied Pressure

We now obtain an exact solution for a composite disc, consisting of a core disc of material 1 (E_1, ρ_1), size $0 \le r \le a_1$ and an annular disc of material 2 (E_2, ρ_2), size $a_1 \le r \le a_2$, subjected to a step-rise of pressure at the outer boundary. For simplicity we assume the Poisson's ratios $\nu_1 = \nu_2 = 0$.

The governing differential equations are:

$$\begin{aligned} \frac{\partial^2 \bar{u}_1}{\partial \xi^2} + \frac{1}{\xi}\frac{\partial \bar{u}_1}{\partial \xi} - \frac{\bar{u}_1}{\xi^2} &= \frac{\partial^2 \bar{u}_1}{\partial \tau^2} \\ \frac{\partial^2 \bar{u}_2}{\partial \xi^2} + \frac{1}{\xi}\frac{\partial \bar{u}_2}{\partial \xi} - \frac{\bar{u}_2}{\partial \xi^2} &= \kappa^2 \frac{\partial^2 \bar{u}_2}{\partial \tau^2} \end{aligned} \tag{B.19}$$

where

$$\kappa^2 = \frac{\rho_2 E_1}{\rho_1 E_2} = \frac{c_1^2}{c_2^2} \; ; \quad u_1 = a_1 \bar{u}_1 \; ; \quad u_2 = a_1 \bar{u}_2 \; ; \quad t = \frac{a_1}{c_1}\tau \tag{B.20}$$

The boundary conditions at the interface $r = a_1$ are

$$\begin{aligned} \bar{u}_1(1,\tau) &= \bar{u}_2(1,\tau) \\ \frac{\partial \bar{u}_1}{\partial \xi} &= \frac{E_2}{E_1}\frac{\partial \bar{u}_2}{\partial \xi} \end{aligned} \tag{B.21}$$

wherein the first equation states the compatibility of displacements (perfect-bond condition), and the second one that of the radial stress. The other boundary conditions are

$$\frac{\partial \bar{u}_2}{\partial \xi}(\alpha,\tau) = -\bar{p}\cdot H(\tau - 0) \quad at \ \xi = \frac{a_2}{a_1} = \alpha$$
$$\bar{u}_1(0,\tau) = 0 \quad at \ \xi = 0 \tag{B.22}$$

where $H(\tau - 0)$ is again the Heaviside unit-step function, and

$$\bar{p} = \frac{p}{E_2} \tag{B.23}$$

Application of the Laplace Transform with respect to the variable τ then yields

$$\frac{d^2\bar{u}_2^*}{\partial \xi^2} + \frac{1}{\xi}\frac{d\bar{u}_2^*}{d\xi} - (\frac{1}{\xi^2} + \kappa^2 s^2)\bar{u}_2^* = 0 \tag{B.24}$$

$$\frac{d^2\bar{u}_1^*}{\partial \xi^2} + \frac{1}{\xi}\frac{d\bar{u}_1^*}{d\xi} - (\frac{1}{\xi^2} + s^2)\bar{u}_1^* = 0 \tag{B.25}$$

$$\bar{u}_1^* = \bar{u}_2^* \quad at \ \xi = 1 \tag{B.26}$$

$$\frac{\partial \bar{u}_1^*}{\partial \xi} = \frac{E_2}{E_1}\frac{\partial \bar{u}_2^*}{\partial \xi} \quad at \ \xi = 1 \tag{B.27}$$

$$\bar{u}_1^* = 0 \quad at \ \xi = 0 \tag{B.28}$$

$$\frac{\partial \bar{u}_2^*}{\partial \xi} = -\frac{\bar{p}}{s} \quad at \ \xi = \alpha \tag{B.29}$$

The transformed solution are

$$\bar{u}_1^* = A_1 I_1(\xi s)$$
$$\bar{u}_2^* = A_2 I_1(\kappa\xi s) + B_2 K_1(\kappa\xi s) \qquad \text{(B.30)}$$

where I_1 and K_1 are the standard notations for the modified Bessel functions. The interface conditions (at $\xi = 1$) can now be used to express A_2 and B_2 in terms of A_1 as follows:

$$\begin{bmatrix} I_1(\kappa s) & K_1(\kappa s) \\ \frac{E_2\kappa}{E_1} I_1'(\kappa s) & \frac{E_2\kappa}{E_1} K_1'(\kappa s) \end{bmatrix} \begin{Bmatrix} A_2 \\ B_2 \end{Bmatrix} = A_1 \begin{Bmatrix} I_1(s) \\ I_1'(s) \end{Bmatrix} \qquad \text{(B.31)}$$

and hence

$$A_2 = \frac{1}{\Delta}[\beta I_1(s)K_1'(\kappa s) - I_1'(s)K_1(\kappa s)]A_1 = -\frac{E_1}{E_2} sF_1(s)A_1 \qquad \text{(B.32)}$$

$$B_2 = \frac{1}{\Delta}[I_1'(s)I_1(\kappa s) - \beta I_1(s)I_1'(\kappa s)]A_1 = -\frac{E_1}{E_2} sF_2(s)A_1 \qquad \text{(B.33)}$$

where we have used the well-known property

$$\Delta = \beta[I_1(\kappa s)K_1'(\kappa s) - I_1'(\kappa s)K_1(\kappa s)] = -\frac{E_2}{E_1 s} \qquad \text{(B.34)}$$

and have introduced

$$\beta = \sqrt{\frac{E_2\rho_2}{E_1\rho_1}} \qquad \text{(B.35)}$$

$$F_1(s) = \beta I_1(s)K_1'(\kappa s) - I_1'(s)K_1(\kappa s) \qquad \text{(B.36)}$$

$$F_2(s) = I_1'(s)I_1(\kappa s) - \beta I_1(s)I_1'(\kappa s) \tag{B.37}$$

Now, the boundary condition at outer periphery $\xi = \alpha$ gives

$$A_1 = \frac{\bar{p}E_2}{E_1\kappa s^3}\frac{1}{F_1(s)I_1'(\kappa\alpha s) + F_2(s)K_1'(\kappa\alpha s)} \tag{B.38}$$

This enables the transforms of the solutions to be expressed as

$$\bar{u}_1^* = A_1 I_1(\xi s) \tag{B.39}$$

$$\bar{u}_2^* = -A_1\frac{E_1}{E_2}s[F_1(s)I_1(\kappa\xi s) + F_2(s)K_1(\kappa\xi s)] \tag{B.40}$$

$$\bar{\sigma}_{rr}^{1*} = \frac{E_1}{E_2}\frac{\partial\bar{u}_1^*}{\partial\xi} = A_1\frac{E_1}{E_2}sI_1'(\xi s) \tag{B.41}$$

$$\bar{\sigma}_{rr}^{2*} = \frac{\partial\bar{u}_2^*}{\partial\xi} = -A_1\frac{E_1}{E_2}\kappa s^2[F_1(s)I_1'(\kappa\xi s) + F_2(s)K_1'(\kappa\xi s)] \tag{B.42}$$

$$\bar{\sigma}_{\theta\theta}^{1*} = \frac{E_1}{E_2}\frac{\bar{u}_1^*}{\xi} = A_1\frac{E_1}{E_2}\frac{I_1(\xi s)}{\xi} \tag{B.43}$$

$$\bar{\sigma}_{\theta\theta}^{2*} = \frac{\bar{u}_2^*}{\xi} = -A_1\frac{E_1}{E_2}\frac{s}{\xi}[F_1(s)I_1(\kappa\xi s) + F_2(s)K_1(\kappa\xi s)] \tag{B.44}$$

Then, the interface ($\xi = 1$) displacement and stresses have the following transforms.

$$\bar{u}_1^* = \bar{u}_2^* = A_1 I_1(s) \tag{B.45}$$

$$\bar{\sigma}_{rr}^{1*} = \bar{\sigma}_{rr}^{2*} = A_1 \frac{E_1}{E_2} s I_1'(s) \tag{B.46}$$

$$\bar{\sigma}_{\theta\theta}^{1*} = A_1 \frac{E_1}{E_2} I_1(s) \tag{B.47}$$

$$\bar{\sigma}_{\theta\theta}^{2*} = -A_1 \frac{E_1}{E_2} s[F_1(s) I_1(\kappa s) + F_2(s) K_1(\kappa s)] \tag{B.48}$$

The inversion of the transforms Eq. (B.43) to Eq. (B.46) can be accomplished by computing residues at the poles of the integrands in the inversion formula. We can show that $s = 0$ is a simple pole. The other poles are also simple poles, being the zeros of the denominator in the expression for A_1 expressed above

$$F_1(s) I_1'(\kappa\alpha s) + F_2(s) K_1'(\kappa\alpha s) = 0 \tag{B.49}$$

These zeros depend on the specific values of κ and α in a given problem. For the case $\alpha = 2$, $E_1/E_2 = 200$, and $\rho_1/\rho_2 = 8$, we have $\kappa = 5$ and the first twenty roots are

$$s_1 = \pm\, 0.29578\, i$$
$$s_2 = \pm\, 0.93350\, i$$
$$s_3 = \pm\, 1.54623\, i$$
$$s_4 = \pm\, 1.84924\, i$$
$$s_5 = \pm\, 2.21656\, i$$
$$s_6 = \pm\, 2.83241\, i$$
$$s_7 = \pm\, 3.45687\, i$$
$$s_8 = \pm\, 4.08213\, i$$
$$s_9 = \pm\, 4.70514\, i$$
$$s_{10} = \pm\, 5.26591\, i$$
$$s_{11} = \pm\, 5.40722\, i$$
$$s_{12} = \pm\, 5.97597\, i$$
$$s_{13} = \pm\, 6.59910\, i$$

$$s_{14} = \pm\ 7.22424\ i$$

$$s_{15} = \pm\ 7.84779\ i$$

$$s_{16} = \pm\ 8.43402\ i$$

$$s_{17} = \pm\ 8.58269\ i$$

$$s_{18} = \pm\ 9.11831\ i$$

$$s_{19} = \pm\ 9.74093\ i$$

$$s_{20} = \pm\ 10.3660\ i$$

Again, as remarked before, the natural frequencies are given by

$$\omega_k^2 = -\frac{c_1^2}{a_1^2} s_k^2 \qquad \text{(B.50)}$$

The expressions for the interface displacement and stresses are found as:

$$u(1) = -0.007976 + 0.010122\cos(0.295784\tau) - 0.003847\cos(0.9335050\tau) + 0.005016\cos(1.54623\tau)\ ......$$

(B.51)

$$\sigma_{rr}(1) = -1.59521 + 1.97989\cos(0.295784\tau) - 0.59539\cos(0.933505\tau) + 0.33325\cos(1.54623\tau)\ ....$$

(B.52)

$$\sigma_{\theta\theta}^1(1) = -1.59521 + 2.02433\cos(0.295784\tau) - 0.769462\cos(0.933505\tau) + 1.00329\cos(1.54623\tau)\ ....$$

(B.53)

$$\sigma_{\theta\theta}^2(1) = -0.007976 + 0.010122\cos(0.295784\tau) - 0.003847\cos(0.9335050\tau) + 0.005016\cos(1.54623\tau)\ ......$$

(B.54)

where τ is the non-dimensional time defined in Eq. (B.20).

When the material positions are switched we have $E_1/E_2 = 1/200$ and $\rho_1/\rho_2 = 1/8$, $\kappa = 1/5$. The corresponding roots in such a case are

$$s_1 = \pm\, 3.31225\, i$$
$$s_2 = \pm\, 3.90521\, i$$
$$s_3 = \pm\, 7.01779\, i$$
$$s_4 = \pm\, 10.1065\, i$$
$$s_5 = \pm\, 13.2862\, i$$
$$s_6 = \pm\, 16.0985\, i$$
$$s_7 = \pm\, 16.7796\, i$$
$$s_8 = \pm\, 19.6440\, i$$
$$s_9 = \pm\, 22.7638\, i$$
$$s_{10} = \pm\, 25.8912\, i$$
$$s_{11} = \pm\, 29.0045\, i$$
$$s_{12} = \pm\, 31.5845\, i$$
$$s_{13} = \pm\, 32.3841\, i$$
$$s_{14} = \pm\, 35.3587\, i$$
$$s_{15} = \pm\, 37.4786\, i$$
$$s_{16} = \pm\, 41.6046\, i$$
$$s_{17} = \pm\, 44.7153\, i$$
$$s_{18} = \pm\, 47.2059\, i$$
$$s_{19} = \pm\, 48.0713\, i$$
$$s_{20} = \pm\, 51.0694\, i\ .$$

The interface displacement and stresses, obtained by using the Residue Theorem for the inversion of Laplace Transform, are of the following form:

$$u(1) = -2.64463 + 2.5563\cos(3.31225\tau) + 0.289828\cos(3.90521\tau) + 0.000705\cos(7.01779\tau) \ \ldots\ldots \tag{B.55}$$

$$\sigma_{rr}(1) = -0.013223 - 0.080905\cos(3.31225\tau) + 0.076137\cos(3.90521\tau) + 0.011204\cos(7.01779\tau) \ \ldots\ldots. \tag{B.56}$$

$$\sigma_{\theta\theta}^{1}(1) = -0.013223 - 0.012782\cos(3.31225\tau) + 0.001449\cos(3.90521\tau) + ..$$

(B.57)

$$\sigma_{\theta\theta}^{2}(1) = -2.64463 + 2.5563\cos(3.31225\tau) + 0.289828\cos(3.90521\tau) + 0.000705\cos(7.01779\tau) ......$$

(B.58)

where again τ is the non-dimensional time of Eq. (B.20).

APPENDIX C

Listing of Finite Element Program

```
/INFO MVS  CL(92)  CODE(CY99) TI(9999)  PA(9999) SUB(000)
/INFO      R(CENTRAL) MSGCLASS(Q)
//   EXEC VSFTCLG,PARM.FORT='DC(GLOB)',PARM.FORT='DC(GLOBE)'
//FORT.SYSIN DD *
C-----------------------------------------------------------------------
C
C FINITE ELEMENT PROGRAM TO ANALYZE A TWO DIMENSIONAL
C OR AXISYMMETRIC TRANSIT DYNAMIC PROBLEM USING THE FOLLOWING ELEMENTS :
C  - BAR ELEMENTS (PLANE STRESS)
C  - CONSTANT STRAIN TRIANGULAR ELEMENTS
C  - ISOPARAMETRIC LINEAR QUADRILATERAL ELEMENTS (TWO DIMENSIONAL AND
C    AXISYMMETRIC CASES)
C  - ISOPARAMETRIC 5-NODE QUADRILATERAL ELEMENTS WITH ENFORCED
C    NORMAL AND SHEAR STRESS CONTINUITY AT THE 5TH NODE (TWO DIMENSIONAL
C    AND AXISYMMETRIC CASES)
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
C  DECLARATION OF VARIABLES
C-----------------------------------------------------------------------
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER N,NEL,NDOF,NLOAD,DOF,NGEN,INTER,MEQNS,KEQNS,REDOF,NCOUNT
      INTEGER NID(640),BCX(640),BCY(640),ELID(640),NTYPE(640),ID(2,640)
      INTEGER N1(640),N2(640),N3(640),N4(640),N5(640),NODRED(640)
      INTEGER ELIDB,N11,N22,N33,N44,N55,NELA(640),NELB(640),NRES
      INTEGER NODFOR(640),KK1,KK2,IFPRE(640),MM,NGAUS1,NGAUS2,INCOMP
      INTEGER NOUT
      DOUBLE PRECISION X(640),Y(640)
      DOUBLE PRECISION A(640),E(640),NU(640),T(640),FX(640),FY(640)
      DOUBLE PRECISION KGLOB(640,640),ASLOD(640)
      DOUBLE PRECISION LOAD(640),FIXED(640),REACT(640),XDISP(640)
      DOUBLE PRECISION D1(640),PE1,PE2,PE,KEL(20,20)
      DOUBLE PRECISION D(640),MGLOB(640,640),MEL(20,20)
C      DOUBLE PRECISION AA(640,640),BB(640,640),HH(640,640),OMEG(640),
      DOUBLE PRECISION AA(640,640),OMEG(640),
     +             T1(640,640),ZETA(640),TT(640,640),XO(640),
     +             XDO(640),YO(640),YDO(640),WN(640,13),F(640,13),
     +             TGM(640,640),V(640,13),XD(640,13),TIME(13),
     +             DT(13),TGMT(640,640),U(640,13),EPS,TPS,FACT(13),
     +             Q(640,13),C(291600),R(291600),WK(540)
      INTEGER ITMAX
      CHARACTER*80 TITLE
C
      COMMON/GLOB/X,Y,A,E,NU,T,D
      COMMON/GLOBE/AA,T1,TT,KGLOB,MGLOB,TGM,TGMT,C,R
C-----------------------------------------------------------------------
C  DATA INPUT
C-----------------------------------------------------------------------
C      OPEN(1,FILE='FI.IN')
C      OPEN(2,FILE='FI.OUT')
C      READ(1,*) TITLE
      READ(1,*) NCASE,N,NEL,NDOF,NLOAD,NGEN,INTER,NGAUS1,NGAUS2,
     +       INCOMP
```

```
      READ(1,*) NFLAG,NFREQ,ITMAX,EPS,TPS,NSTEP
      MEQNS=640
      KEQNS=640
      DOF=N*NDOF
      IF(NGEN .EQ. 1) THEN
        CALL GENER(N,NDOF,BCX,BCY,NEL,NID,X,Y,ELID,
     +            NTYPE,N1,N2,N3,N4,N5,A,E,NU,T,D,KEQNS,MEQNS)
      ELSE
        READ(1,*) (NID(I),X(I),Y(I),BCX(I),BCY(I),I=1,N)
        READ(1,*) (ELID(I),NTYPE(I),N1(I),N2(I),N3(I),N4(I),N5(I),A(I),
     +      E(I),NU(I),T(I),D(I),I=1,NEL)
      END IF
      IF(INTER .NE. 0) THEN
        READ(1,*) (NELA(I),NELB(I),I=1,INTER)
      END IF
      IF(NLOAD .NE. 0) THEN
        READ(1,*) (NODFOR(I),FX(I),FY(I),I=1,NLOAD)
      END IF
      DOF=N*NDOF
C-----------------------------------------------------------------------
C  RENUMBER NODES NEGLECTING INTERFACE NODES
C-----------------------------------------------------------------------
      NNN=N-INTER
      DO 50 I=1,NNN
        ID(1,I)=BCX(I)
        ID(2,I)=BCY(I)
   50 CONTINUE
      NEQ=0
      DO 70 I=1,NNN
        DO 60 J=1,NDOF
          IF(ID(J,I).EQ.1) THEN
            ID(J,I)=0
          ELSE
            NEQ=NEQ+1
            ID(J,I)=NEQ
          END IF
   60   CONTINUE
   70 CONTINUE
C-----------------------------------------------------------------------
C  DATA ECHO
C-----------------------------------------------------------------------
      CALL DATA(N,NDOF,DOF,BCX,BCY,NEL,NID,X,Y,ELID,NTYPE,N1,
     +          N2,N3,N4,N5,A,E,NU,T,D,NODFOR,FX,FY,KEQNS,MEQNS,NLOAD,
     +          INTER,NGAUS1,NGAUS2,TITLE,INCOMP,NCASE,NRES)
C-----------------------------------------------------------------------
C  CALCULATION OF ELEMENT AND GLOBAL STIFFNESS/MASS MATRICES
C-----------------------------------------------------------------------
      REDOF=DOF-NDOF*INTER-NRES
      DO 900 I=1,REDOF
        DO 900 J=1,REDOF
          KGLOB(I,J)=0.0D0
          MGLOB(I,J)=0.0D0
  900 CONTINUE
      DO 1005 I=1,NEL
        MM=2*NTYPE(I)
        NCOND=1
        IF(NTYPE(I) .EQ. 5) THEN
          NCOND=0
          DO 1000 J=1,INTER
            IF(ELID(I) .EQ. NELA(J)) THEN
              ELIDB=NELB(J)
              N11=N1(ELIDB)
```

```
            N22=N2(ELIDB)
            N33=N3(ELIDB)
            N44=N4(ELIDB)
            N55=N5(ELIDB)
            MM=12
            CALL STIFF(ELID(I),ELIDB,NTYPE(I),N1(I),N2(I),N3(I),
     +             N4(I),N5(I),N11,N22,N33,N44,N55,KGLOB,KEL,
     +             MEQNS,NEL,MM,NGAUS1,NGAUS2,ID,INCOMP,
     +             NCASE)
            CALL MASS (ELID(I),ELIDB,NTYPE(I),N1(I),N2(I),N3(I),
     +             N4(I),N5(I),N11,N22,N33,N44,N55,MGLOB,MEL,
     +             MEQNS,NEL,MM,NGAUS1,NGAUS2,ID,INCOMP,
     +             NCASE)
           END IF
 1000    CONTINUE
        END IF
        IF(NCOND .EQ. 1) THEN
        CALL STIFF(ELID(I),ELIDB,NTYPE(I),N1(I),N2(I),N3(I),N4(I),N5(I)
     +        ,N11,N22,N33,N44,N55,KGLOB,KEL,MEQNS,NEL,MM,NGAUS1,
     +         NGAUS2,ID,INCOMP,NCASE)
        CALL MASS (ELID(I),ELIDB,NTYPE(I),N1(I),N2(I),N3(I),N4(I),N5(I)
     +        ,N11,N22,N33,N44,N55,MGLOB,MEL,MEQNS,NEL,MM,NGAUS1,
     +         NGAUS2,ID,INCOMP,NCASE)
        END IF
 1005 CONTINUE
C---------------------------------------------------------------------------
C  INITIALIZATION OF FORCE AND B.C. VECTORS
C---------------------------------------------------------------------------
      DO 1020 I=1,NNN*NDOF
        ASLOD(I)=0.0D0
 1020 CONTINUE
      DO 1030 I=1,NLOAD
C        KK1=2*NODRED(NODFOR(I))-1
        KK1=2*NODFOR(I)-1
        KK2=KK1+1
        ASLOD(KK1)=FX(I)
        ASLOD(KK2)=FY(I)
 1030 CONTINUE
      DO 1040 I=1,NNN
        II=ID(1,I)
        IF(II.GT.0) THEN
          ASLOD(II)=ASLOD(I*2-1)
        END IF
        JJ=ID(2,I)
        IF(JJ.GT.0) THEN
          ASLOD(JJ)=ASLOD(I*2)
        END IF
 1040 CONTINUE
      NMODE=REDOF
C
      DO 1100 I=1,N
        ZETA(I)=0.0
        XO(I)=0.0
        XDO(I)=0.0D0
 1100 CONTINUE
C
      DO 1101 I=1,NSTEP
        TIME(I)=I
        FACT(I)=1.0
 1101 CONTINUE
C
      DO 1120 I=1,NSTEP
```

```
      TIME(I)=TIME(I)*TPS
 1120 CONTINUE
      DO 1130 I=1,REDOF
        DO 1130 J=1,NSTEP
          F(I,J)=ASLOD(I)*FACT(J)
 1130 CONTINUE
C
      DO 1141 I=1,REDOF
        IF ( KGLOB(I,I) .LT. 0.0 ) THEN
          WRITE(2,2501) I, KGLOB(I,I)
        END IF
 1141 CONTINUE
C
      CALL DG2CSP (REDOF,KGLOB,MEQNS,MGLOB,MEQNS,OMEG,AA,MEQNS,C,R,WK)
      WRITE(2,2505)
C
      DO 1150 I=1,REDOF
        OMEG(I)=SQRT(OMEG(I))
 1150 CONTINUE
C
      DO 1160 J=1,REDOF
C        I=REDOF+1-J
        WRITE(2,2501) J,OMEG(J)
 1160 CONTINUE
C
      DO 1165 I=1,REDOF
        DO 1165 J=1,REDOF
          T1(I,J)=AA(I,J)
 1165 CONTINUE
      DO 1170 I=1,REDOF
        DO 1170 J=1,REDOF
          TT(I,J)=T1(J,I)
 1170 CONTINUE
C
      IF ( NFLAG. EQ. 1 ) THEN
        CALL MODAL (MGLOB,OMEG,T1,ZETA,XO,XDO,YO,YDO,WN,F,U,V,XD,TIME,
     +              DT,TT,M,NSTEP,REDOF,NMODE,TGMT,TGM,MEQNS)
        WRITE(2,2506)
      END IF
C
      IF ( NFLAG. EQ. 0 ) THEN
        CALL STATIC (OMEG,T1,TT,U,F,XD,Q,REDOF,MEQNS,NSTEP,NFREQ)
      END IF
C
      DO 1180 I=1,REDOF
        WRITE(2,2507) I, (XD(I,J),J=1,NSTEP)
 1180 CONTINUE
C
      DO 1400 JY=1,NSTEP
        WRITE(2,2402) JY
        DO 1200 IY=1,NNN
          IF(IY.EQ.1) THEN
            KKY=0
          END IF
          IIY=ID(1,IY)
          JJY=ID(2,IY)
          IF(IIY.EQ.0) THEN
            XDISP(2*IY-1)=0.0D0
          ELSE
            KKY=KKY+1
            XDISP(2*IY-1)=XD(KKY,JY)
          END IF
```

```
        IF(JJY.EQ.0) THEN
          XDISP(2*IY)=0.0D0
        ELSE
          KKY=KKY+1
          XDISP(2*IY)=XD(KKY,JY)
        END IF
 1200   CONTINUE
      CALL DISPL(N,DOF,NID,NODRED,N1,N2,N3,N4,N5,NELA,NELB,INTER,
     +          XDISP,REACT,MEQNS,NCASE)
C
      DO 1251 I=1,N
        WRITE(2,2403)  XDISP(2*I-1), XDISP(2*I)
 1251   CONTINUE
C
      DO 1300 I=1,NEL
        IF(NTYPE(I) .EQ. 2)THEN
          CALL FORCE(ELID(I),N1(I),N2(I),X(N1(I)),Y(N1(I)),X(N2(I)),
     +              Y(N2(I)),A(I),E(I),MEQNS,XDISP)
        ELSE
          IF(NTYPE(I) .EQ. 3) THEN
            IF(NCASE .EQ. 4) THEN
              CALL STRIAX(ELID(I),N1(I),N2(I),N3(I),X(N1(I)),Y(N1(I)
     +                   ),X(N2(I)),Y(N2(I)),X(N3(I)),Y(N3(I)),E(I)
     +                   ,NU(I),XDISP,MEQNS)
            ELSE
              CALL STR(ELID(I),N1(I),N2(I),N3(I),X(N1(I)),Y(N1(I)),
     +             X(N2(I)),Y(N2(I)),X(N3(I)),Y(N3(I)),E(I),NU(I),
     +             T(I),XDISP,MEQNS)
            END IF
          ELSE
            IF(NTYPE(I) .EQ. 4) THEN
              X5=0.D0
              Y5=0.D0
            ELSE
              X5=X(N5(I))
              Y5=Y(N5(I))
            END IF
            IF(NCASE .EQ. 4) THEN
              CALL STRAX(ELID(I),N1(I),N2(I),N3(I),N4(I),N5(I),
     +                   X(N1(I)),Y(N1(I)),X(N2(I)),Y(N2(I)),X(N3(I)
     +                   ),Y(N3(I)),X(N4(I)),Y(N4(I)),X5,Y5,E(I),
     +                   NU(I),XDISP,MEQNS,NTYPE(I))
            ELSE
              CALL STRES(ELID(I),N1(I),N2(I),N3(I),N4(I),N5(I),
     +                   X(N1(I)),Y(N1(I)),X(N2(I)),Y(N2(I)),X(N3(I)
     +                   ),Y(N3(I)),X(N4(I)),Y(N4(I)),X5,Y5,E(I),
     +                   NU(I),T(I),XDISP,MEQNS,NTYPE(I),INCOMP)
            END IF
          END IF
        END IF
 1300   CONTINUE
 1400 CONTINUE
C
C-----------------------------------------------------------------------
C  END OF PROGRAM
C-----------------------------------------------------------------------
C2401 FORMAT(/,12D10.4)
 2402 FORMAT(///,3X,'TIME STEP =',2X,I2,/)
 2403 FORMAT(3X,D14.8,6X,D14.8)
 2507 FORMAT(/,11HCOORDINATES,I5,/,1X,5D14.8,/,1X,5D14.8,/,1X,5D14.8)
 2500 FORMAT(/,2X, 'I=', I5, 2X, 'J=', I5, 3X, D12.5, 3X, D12.5)
 2501 FORMAT(/,2X,I5,3X,D12.5)
```

```
2502 FORMAT(/,2X,12I5)
C
     STOP
     END
C
     SUBROUTINE GENER(N,NDOF,BCX,BCY,NEL,NID,X,Y,ELID,
    +                NTYPE,N1,N2,N3,N4,N5,A,E,NU,T,D,KEQNS,MEQNS)
C-----------------------------------------------------------------------
C  THIS SUBROUTINE READS AND GENERATES INPUT DATA
C-----------------------------------------------------------------------
     IMPLICIT REAL*8 (A-H,O-Z)
     DIMENSION BCX(KEQNS),BCY(KEQNS),NID(KEQNS),X(KEQNS),Y(KEQNS)
     DIMENSION ELID(MEQNS),N1(MEQNS)
     DIMENSION N2(MEQNS),N3(MEQNS),E(MEQNS),NU(MEQNS),T(MEQNS)
     DIMENSION N4(MEQNS),N5(MEQNS),NTYPE(MEQNS),A(MEQNS),D(MEQNS)
     INTEGER BCX,BCY,ELID,BCX1,BCY1
     REAL*8 NU,NU1
C-----------------------------------------------------------------------
C  GENERATION OF NODAL POINTS
C-----------------------------------------------------------------------
     LL=1
C    DO 40 WHILE(LL .LE. N)
     DO WHILE(LL .LE. N)
        READ(1,*) NID1,X1,Y1,BCX1,BCY1,KGEN
        IF(KGEN .NE. 0) THEN
          READ(1,*) NID2,X2,Y2,KK
          KFACT=NID2-NID1
          KFACT1=KFACT/KK + 1
          DO 20 MM=1,KFACT1
            NIDGEN = NID1 + (MM-1)*KK
            NID(NIDGEN) = NIDGEN
            X(NIDGEN) = X1 + (X2-X1)*(MM-1)/(KFACT/KK)
            Y(NIDGEN) = Y1 + (Y2-Y1)*(MM-1)/(KFACT/KK)
            BCX(NIDGEN) = BCX1
            BCY(NIDGEN) = BCY1
  20      CONTINUE
          LM = KFACT1
        ELSE
          NID(NID1) = NID1
          X(NID1) = X1
          Y(NID1) = Y1
          BCX(NID1) = BCX1
          BCY(NID1) = BCY1
          LM = 1
        END IF
        LL = LL + LM
C  40 CONTINUE
     ENDDO
C-----------------------------------------------------------------------
C  GENERATION OF ELEMENT CONNECTIONS
C-----------------------------------------------------------------------
     LL = 1
C    DO 80 WHILE(LL .LE. NEL)
     DO  WHILE(LL .LE. NEL)
        READ(1,*) NELID1,NNTYPE,NN1,NN2,NN3,NN4,NN5,A1,E1,NU1,T1,D1,
    +             KGEN
        IF(KGEN .NE. 0) THEN
          READ(1,*) NELID2,NNTYPE,NNN1,NNN2,NNN3,NNN4,NNN5,KK
          KELFAC = NELID2-NELID1
          KEL1 = KELFAC/KK
          KELF1 = KEL1 + 1
          DO 60 MM = 1,KELF1
```

```
          NELID = NELID1 + (MM-1)*KK
          ELID(NELID) = NELID
          NTYPE(NELID) = NNTYPE
          N1(NELID) = NN1 + (MM-1)*(NNN1-NN1)/KEL1
          N2(NELID) = NN2 + (MM-1)*(NNN2-NN2)/KEL1
          N3(NELID) = NN3 + (MM-1)*(NNN3-NN3)/KEL1
          N4(NELID) = NN4 + (MM-1)*(NNN4-NN4)/KEL1
          N5(NELID) = NN5 + (MM-1)*(NNN5-NN5)/KEL1
          A(NELID) = A1
          E(NELID) = E1
          NU(NELID) = NU1
          T(NELID) = T1
          D(NELID) = D1
 60      CONTINUE
        LM = KELF1
       ELSE
        ELID(NELID1) = NELID1
        NTYPE(NELID1) = NNTYPE
        N1(NELID1) = NN1
        N2(NELID1) = NN2
        N3(NELID1) = NN3
        N4(NELID1) = NN4
        N5(NELID1) = NN5
        A(NELID1) = A1
        E(NELID1) = E1
        NU(NELID1) = NU1
        T(NELID1) = T1
        D(NELID1) = D1
        LM = 1
       END IF
       LL = LL + LM
C   80 CONTINUE
      ENDDO
C
      RETURN
      END
C
      SUBROUTINE DATA(N,NDOF,DOF,BCX,BCY,NEL,NID,X,Y,
     +          ELID,NTYPE,N1,N2,N3,N4,N5,A,E,NU,T,D,NODFOR,FX,FY,
     +          KEQNS,MEQNS,NLOAD,INTER,NGAUS1,NGAUS2,TITLE,
     +          INCOMP,NCASE,NRES)
C-----------------------------------------------------------------------
C THIS SUBROUTINE ECHOES THE INPUT DATA FOR CHECKING
C AND REFERENCE PURPOSES
C-----------------------------------------------------------------------
      IMPLICIT REAL*8 (A-H,O-Z)
      DOUBLE PRECISION NU
      DIMENSION BCX(KEQNS),BCY(KEQNS),NID(KEQNS),X(KEQNS),Y(KEQNS)
      DIMENSION ELID(MEQNS),N1(MEQNS),FY(KEQNS)
      DIMENSION N2(MEQNS),A(MEQNS),E(MEQNS),NODFOR(KEQNS),FX(KEQNS)
      DIMENSION N3(MEQNS),N4(MEQNS),N5(MEQNS),NU(MEQNS),T(MEQNS)
      DIMENSION NTYPE(MEQNS),D(MEQNS)
      CHARACTER*5 TYPE(400)
      CHARACTER*80 TITLE
      CHARACTER*3 MODES
      INTEGER DOF,BCX,BCY,ELID,INCOMP
      WRITE(2,110) TITLE
  100 FORMAT('- PLANE STRESS ANALYSIS  -  DATA  ECHO')
  102 FORMAT('- PLANE STRAIN ANALYSIS  -  DATA  ECHO')
  105 FORMAT('- AXISYMMETRIC ANALYSIS  -  DATA  ECHO')
      IF(NCASE .EQ. 2) THEN
       WRITE(2,100)
```

```
    ELSE
     IF(NCASE .EQ. 3) THEN
      WRITE(2,102)
     ELSE
      IF(NCASE .EQ. 4) THEN
       WRITE(2,105)
      END IF
     END IF
    END IF
110 FORMAT('1',A80)
    WRITE(2,120) N
120 FORMAT('- NUMBER OF JOINTS',T32,':',I3)
    WRITE(2,130) NDOF
130 FORMAT('- NUMBER OF D.O.F. PER JOINT',T32,':',I3)
    WRITE(2,140) DOF
140 FORMAT('- TOTAL NUMBER OF D.O.F.',T32,':',I3)
    NRES=0
    DO 160 I=1,N
      NRES=NRES+BCX(I)+BCY(I)
160 CONTINUE
    WRITE(2,170) NRES
170 FORMAT('- NUMBER OF RESTRAINED D.O.F.',T32,':',I3)
    WRITE(2,180) DOF-NRES
180 FORMAT('- NUMBER OF UNRESTRAINED D.O.F.',T32,':',I3)
    NEL1=0
    NEL2=0
    NEL3=0
    NEL4=0
    DO 185 I=1,NEL
      IF(NTYPE(I) .EQ. 2) THEN
       NEL1=NEL1+1
       TYPE(I)=' BAR'
      ELSE
       IF(NTYPE(I) .EQ. 3) THEN
        NEL2=NEL2+1
        IF(NCASE .EQ. 4) THEN
         TYPE(I)='TRIAX'
        ELSE
         TYPE(I)='C S T'
        END IF
       ELSE
        IF(NTYPE(I) .EQ. 4) THEN
         NEL3=NEL3+1
         TYPE(I)='QUAD4'
        ELSE
         NEL4=NEL4+1
         TYPE(I)='QUAD5'
        END IF
       END IF
      END IF
185 CONTINUE
    WRITE(2,190) NEL1
190 FORMAT('- NUMBER OF BAR ELEMENTS',T32,':',I3)
    WRITE(2,195) NEL2
195 FORMAT('- NUMBER OF TRIANG ELEMENTS',T32,':',I3)
    WRITE(2,196) NEL3
196 FORMAT('- NUMBER OF QUAD4 ELEMENTS',T32,':',I3)
    WRITE(2,197) NEL4
197 FORMAT('- NUMBER OF INTERFACE ELEMENTS',T32,':',I3)
    WRITE(2,198) NGAUS1,NGAUS1
198 FORMAT('- QUAD4 INTEGRATION ORDER',T32,':',I2,' BY',I2)
    WRITE(2,199) NGAUS2,NGAUS2
```

```
 199 FORMAT('- QUADS INTEGRATION ORDER',T32,':',I2,' BY',I2)
     IF(INCOMP .EQ. 1) THEN
      MODES = 'YES'
     ELSE
      MODES = 'NO'
     END IF
     WRITE(2,200) MODES
 200 FORMAT('- INCOMPATIBLE BENDING MODES',T32,': ',A3)
     WRITE(2,209)
 209 FORMAT('1 NODE COORDINATES')
     WRITE(2,210)
 210 FORMAT('- NODE',T15,'X',T25,'Y',T32,'X-BC',T42,'Y-BC')
     WRITE(2,220) (NID(I),X(I),Y(I),BCX(I),BCY(I),
    +            I=1,N)
 220 FORMAT('-',I3,T10,F7.3,T20,F7.3,T32,I3,T42,I3)
     WRITE(2,230)
 230 FORMAT('1 ELEMENT INCIDENCES')
     WRITE(2,240)
 240 FORMAT('- ELEMENT',T11,'TYPE',T17,'NODE-1',T24,'NODE-2',T31,
    +      'NODE-3',T38,'NODE-4',T45,'NODE-5',T56,'AREA',T68,'E',
    +       T77,'NU',T87,'T',T97,'D')
     WRITE(2,250) (ELID(I),TYPE(I),N1(I),N2(I),N3(I),N4(I),N5(I),A(I),
    +            E(I),NU(I),T(I),D(I),I=1,NEL)
 250 FORMAT('-',T4,I3,T11,A5,T19,I3,T26,I3,T33,I3,T40,I3,T47,I3,T53,
    +        E8.2,T63,E10.3,T76,F4.2,T82,E10.3,T96,E10.3)
     WRITE(2,260)
 260 FORMAT('1 APPLIED LOADS')
     IF(NLOAD .NE. 0) THEN
      WRITE(2,270)
 270  FORMAT('- NODE',T13,'X-FORCE',T28,'Y-FORCE')
      WRITE(2,280) (NODFOR(I),FX(I),FY(I),I=1,NLOAD)
 280  FORMAT('-',I4,T11,E10.3,T26,E10.3)
     ELSE
      WRITE(2,290)
 290  FORMAT('- NO CONCENTRATED LOADS APPLIED')
     END IF
C
     RETURN
     END
C
     SUBROUTINE STIFF(ELID,ELIDB,NTYPE,N1,N2,N3,N4,N5,N11,N22,N33,N44,
    +              N55,KGLOB,KEL,MEQNS,NEL,MM,NGAUS1,NGAUS2,ID,
    +              INCOMP,NCASE)
C-----------------------------------------------------------------------
C  THIS SUBROUTINE CALCULATES THE ELEMENT STIFFNESS MATRIX
C  AND ASSEMBLES THE GLOBAL STIFFNESS MATRIX
C-----------------------------------------------------------------------
     IMPLICIT REAL*8 (A-H,O-Z)
     DOUBLE PRECISION KPRIM(4,4),KEL(MM,MM),NU
     DOUBLE PRECISION KELA(10,10),KELB(10,10),KELARA(10,12)
     DOUBLE PRECISION KELBRB(10,12),KEL1(12,12),KEL2(12,12)
     DOUBLE PRECISION KGLOB(MEQNS,MEQNS)
     DIMENSION TR(4,4),B(3,10)
     DIMENSION Q(2,12),RA(10,12),RB(10,12),RAT(12,10)
     DIMENSION RBT(12,10)
     INTEGER N1,N2,N3,N4,N5,N11,N22,N33,N44,N55,ELID,ELIDB,NGLOB(12)
     INTEGER NOD(6),INCOMP,ID(2,MEQNS),KKK(12)
C
     COMMON/GLOB/X(640),Y(640),A(640),E(640),NU(640),T(640),D(640)
C
     NNTYPE=NTYPE
     NDIM=NTYPE*2
```

```
      NOD(1)=N1
      NOD(2)=N2
      NOD(3)=N3
      NOD(4)=N4
      KKK(1)=ID(1,NOD(1))
      KKK(2)=ID(2,NOD(1))
      KKK(3)=ID(1,NOD(2))
      KKK(4)=ID(2,NOD(2))
      KKK(5)=ID(1,NOD(3))
      KKK(6)=ID(2,NOD(3))
      KKK(7)=ID(1,NOD(4))
      KKK(8)=ID(2,NOD(4))
C
   IF(NTYPE .EQ. 2) THEN
    CALL BAR(ELID,N1,N2,X(N1),Y(N1),X(N2),Y(N2),A(ELID),E(ELID),
  +         KEL,TR,KPRIM)
   ELSE
    IF(NTYPE .EQ. 3) THEN
     IF(NCASE .EQ. 4) THEN
      CALL TRIAX(ELID,N1,N2,N3,X(N1),Y(N1),X(N2),Y(N2),X(N3),
  +            Y(N3),E(ELID),NU(ELID),KEL)
     ELSE
      CALL CST(ELID,N1,N2,N3,X(N1),Y(N1),X(N2),Y(N2),X(N3),Y(N3),
  +           E(ELID),NU(ELID),T(ELID),B,KEL)
     END IF
    ELSE
     IF(NTYPE .EQ. 4) THEN
      IF(NCASE .EQ. 4) THEN
       CALL QUADAX(NGAUS1,ELID,NTYPE,N1,N2,N3,N4,N5,KEL,NDIM)
      ELSE
       CALL QUAD4(NGAUS1,ELID,NTYPE,N1,N2,N3,N4,N5,KEL,NDIM,
  +            INCOMP)
      END IF
     ELSE
      IF(NCASE .EQ. 4) THEN
       CALL RELAX(ELID,ELIDB,N1,N2,N3,N4,N5,N11,N22,N33,N44,N55,
  +            Q)
       CALL QUADAX(NGAUS2,ELID,NTYPE,N1,N2,N3,N4,N5,KELA,NDIM)
       CALL QUADAX(NGAUS2,ELIDB,NTYPE,N11,N22,N33,N44,N55,KELB,
  +            NDIM)
      ELSE
       CALL REL(ELID,ELIDB,N1,N2,N3,N4,N5,N11,N22,N33,N44,N55,Q)
       CALL QUAD4(NGAUS2,ELID,NTYPE,N1,N2,N3,N4,N5,KELA,NDIM,
  +            INCOMP)
       CALL QUAD4(NGAUS2,ELIDB,NTYPE,N11,N22,N33,N44,N55,KELB,
  +            NDIM,INCOMP)
      END IF
C-----------------------------------------------------------------------
C  FORM R MATRICES
C-----------------------------------------------------------------------
       DO 50 LI=1,8
        DO 50 LJ=1,12
         RA(LI,LJ)=0.D0
         RB(LI,LJ)=0.D0
  50    CONTINUE
       RA(7,1)=1.D0
       RA(8,2)=1.D0
       RA(1,3)=1.D0
       RA(2,4)=1.D0
       DO 55 LK=3,6
        LM=LK+6
        RA(LK,LM)=1.D0
```

```
 55    CONTINUE
       DO 58 KK=3,8
         RB(KK,KK)=1.D0
 58    CONTINUE
       RB(1,9)=1.D0
       RB(2,10)=1.D0
       DO 60 K=9,10
         DO 60 L=1,12
           MMM=K-8
           RA(K,L)=Q(MMM,L)
           RB(K,L)=Q(MMM,L)
 60    CONTINUE
C-------------------------------------------------------------------------
C  MINIMIZE P.E. AND FORM STIFFNESS MATRIX
C-------------------------------------------------------------------------
       DO 80 M=1,10
         DO 80 N=1,12
           RAT(N,M)=RA(M,N)
           RBT(N,M)=RB(M,N)
 80    CONTINUE
       CALL MATMAT(10,10,12,KELA,RA,KELARA)
       CALL MATMAT(12,10,12,RAT,KELARA,KEL1)
       CALL MATMAT(10,10,12,KELB,RB,KELBRB)
       CALL MATMAT(12,10,12,RBT,KELBRB,KEL2)
       DO 90 K1=1,12
         DO 90 K2=1,12
           KEL(K1,K2)=KEL1(K1,K2)+KEL2(K1,K2)
 90    CONTINUE
      END IF
    END IF
   END IF
C-------------------------------------------------------------------------
C  ASSEMBLE GLOBAL STIFFNESS MATRIX
C-------------------------------------------------------------------------
   IF(NTYPE .EQ. 5) THEN
     NOD(1)=N4
     NOD(2)=N22
     NOD(3)=N33
     NOD(4)=N44
     NOD(5)=N2
     NOD(6)=N3
     NNTYPE=6
     KKK(1) =ID(1,NOD(1))
     KKK(2) =ID(2,NOD(1))
     KKK(3) =ID(1,NOD(2))
     KKK(4) =ID(2,NOD(2))
     KKK(5) =ID(1,NOD(3))
     KKK(6) =ID(2,NOD(3))
     KKK(7) =ID(1,NOD(4))
     KKK(8) =ID(2,NOD(4))
     KKK(9) =ID(1,NOD(5))
     KKK(10)=ID(2,NOD(5))
     KKK(11)=ID(1,NOD(6))
     KKK(12)=ID(2,NOD(6))
   END IF
   DO 1300 I=1,MM
     IF(KKK(I).GT.0) THEN
       K=KKK(I)
       DO 1250 J=1,MM
         IF(KKK(J).GT.0) THEN
           L=KKK(J)
```

```
            KGLOB(K,L)=KGLOB(K,L)+KEL(I,J)
          END IF
1250      CONTINUE
       END IF
1300 CONTINUE
3000 FORMAT(//,'  NTYPE =', I3)
C
      RETURN
      END
C
      SUBROUTINE BAR(ELID,N1,N2,X1,Y1,X2,Y2,A,E,KEL,TR,KPRIM)
C-----------------------------------------------------------------------
C  THIS SUBROUTINE CALCULATES THE ELEMENT STIFFNESS MATRIX
C  AND TRANSFORMATION MATRIX OF A BAR ELEMENT
C-----------------------------------------------------------------------
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER N1,N2,ELID,NGLOB(4)
      REAL*8 L,L2,COS,SIN,C2,S2,XDIF,YDIF,CS,KEL(4,4),K,LSMALL,TR(4,4)
      REAL*8 KPRIM(4,4)
C
      XDIF=X2-X1
      YDIF=Y2-Y1
      L2=XDIF*XDIF + YDIF*YDIF
      L=DSQRT(L2)
      COS=XDIF/L
      SIN=YDIF/L
      C2=COS*COS
      S2=SIN*SIN
      CS=COS*SIN
      K = A*E/L
      KEL(1,1)=C2*K
      KEL(1,2)=CS*K
      KEL(1,3)=-KEL(1,1)
      KEL(1,4)=-KEL(1,2)
      KEL(2,1)=KEL(1,2)
      KEL(3,1)=KEL(1,3)
      KEL(4,1)=KEL(1,4)
      KEL(2,2)=S2*K
      KEL(2,3)=-KEL(1,2)
      KEL(2,4)=-KEL(2,2)
      KEL(3,2)=KEL(2,3)
      KEL(4,2)=KEL(2,4)
      KEL(3,3)=KEL(1,1)
      KEL(3,4)=KEL(1,2)
      KEL(4,4)=KEL(2,2)
      KEL(4,3)=KEL(3,4)
C
      DO 1200 I=1,4
        DO 1200 J=1,4
          TR(I,J)=0.0
          KPRIM(I,J)=0.0
 1200 CONTINUE
C
      TR(1,1)=COS
      TR(1,2)=SIN
      TR(2,1)=-SIN
      TR(2,2)=COS
      TR(3,3)=COS
      TR(4,3)=-SIN
      TR(4,4)=COS
C
      KPRIM(1,1)=K
```

```
      KPRIM(1,3)=-K
      KPRIM(3,1)=-K
      KPRIM(3,3)=K
C
      RETURN
      END
C
      SUBROUTINE CST(ELID,N1,N2,N3,X1,Y1,X2,Y2,X3,Y3,E,NU,T,B,KEL)
C---------------------------------------------------------------------------
C  THIS SUBROUTINES CALCULATES THE ELEMENT STIFFNESS MATRIX OF A
C  CONSTANT STRAIN TRIANGULAR ELEMENT
C---------------------------------------------------------------------------
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER ELID
      REAL*8 X1,Y1,X2,Y2,X3,Y3,NU,T,EM(3,3),B(3,6),KEL(6,6)
      REAL*8 BT(6,3),BTE(6,3),K1(6,6),A
C
      B(1,1)=Y2-Y3
      B(1,2)=0.0
      B(1,3)=Y3-Y1
      B(1,4)=0.0
      B(1,5)=Y1-Y2
      B(1,6)=0.0
      B(2,1)=0.0
      B(2,2)=X3-X2
      B(2,3)=0.0
      B(2,4)=X1-X3
      B(2,5)=0.0
      B(2,6)=X2-X1
      B(3,1)=B(2,2)
      B(3,2)=B(1,1)
      B(3,3)=B(2,4)
      B(3,4)=B(1,3)
      B(3,5)=B(2,6)
      B(3,6)=B(1,5)
C
      A=(X2*Y3-Y2*X3-X1*Y3+Y1*X3+X1*Y2-Y1*X2)/2.
C
      DO 20 I=1,3
        DO 20 J=1,6
          B(I,J)=B(I,J)/(2.*A)
   20 CONTINUE
      DO 50 I=1,6
        DO 50 J=1,3
          BT(I,J)=B(J,I)
   50 CONTINUE
C
      CALL YOUNG(EM,E,NU,3,0)
      CALL MATMAT(6,3,3,BT,EM,BTE)
      CALL MATMAT(6,3,6,BTE,B,K1)
      DO 100 I=1,6
        DO 100 J=1,6
          KEL(I,J)=K1(I,J)*T*A
  100 CONTINUE
C
      RETURN
      END
C
      SUBROUTINE QUAD4(NGAUSS,NELEM,NORDER,N1,N2,N3,N4,N5,STIFEL,NDIM,
     +                 INCOMP)
C---------------------------------------------------------------------------
C  SUBROUTINE FOR FORMATION AND NUMERICAL INTEGRATION OF ISOPARAMETRIC
```

```
C  ELEMENT STIFFNESS MATRIX
C-----------------------------------------------------------------------------------------------------------
      IMPLICIT REAL*8 (A-H,O-Z)
      DOUBLE PRECISION JACOB,NU,KEL(14,14)
      DIMENSION XX(5),YY(5),PLACE(3,3),WGT(3,3),B(3,14),BTE(14,3)
      DIMENSION EM(3,3),STIFEL(NDIM,NDIM)
C
      COMMON/Q4/EN(5),JACOB(2,2)
      COMMON/GLOB/X(640),Y(640),A(640),E(640),NU(640),T(640),D(640)
C
      DATA PLACE(1,1),PLACE(2,1),PLACE(2,3),PLACE(3,1),PLACE(3,2)/
     +     5*0.000000000000000D0/
      DATA PLACE(1,2)/-0.577350269189626D0/
      DATA PLACE(2,2)/ 0.577350269189626D0/
      DATA PLACE(1,3)/-0.774596669241483D0/
      DATA PLACE(3,3)/ 0.774596669241483D0/
      DATA WGT(1,1)/2.000000000000000D0/,WGT(2,3)/0.888888888888889D0/
      DATA WGT(1,2),WGT(2,2)/2*1.000000000000000D0/
      DATA WGT(2,1),WGT(3,1),WGT(3,2)/3*0.000000000000000D0/
      DATA WGT(1,3),WGT(3,3)/2*0.555555555555556D0/
C
      CALL YOUNG(EM,E(NELEM),NU(NELEM),3,1)
      MORDER=2*NORDER
      NSIZE = NDIM + 2*INCOMP
C
      XX(1)=X(N1)
      XX(2)=X(N2)
      XX(3)=X(N3)
      XX(4)=X(N4)
      YY(1)=Y(N1)
      YY(2)=Y(N2)
      YY(3)=Y(N3)
      YY(4)=Y(N4)
      IF(NORDER .NE. 4) THEN
        XX(5)=X(N5)
        YY(5)=Y(N5)
      END IF
C
C-----------------------------------------------------------------------------------------------------------
C  CLEAR UPPER TRIANGLE OF ELEMENT STIFFNESS MATRIX
C-----------------------------------------------------------------------------------------------------------
      DO 40 K=1,NSIZE
        DO 40 L=K,NSIZE
          KEL(K,L)=0.D0
   40 CONTINUE
C-----------------------------------------------------------------------------------------------------------
C  START QUADRATURE LOOP
C-----------------------------------------------------------------------------------------------------------
      DO 180 NA=1,NGAUSS
        XI=PLACE(NA,NGAUSS)
        DO 160 NB=1,NGAUSS
          ET=PLACE(NB,NGAUSS)
          CALL SHAPEF(XI,ET,XX,YY,DETJAC,B,NORDER,INCOMP)
          DV=WGT(NA,NGAUSS)*WGT(NB,NGAUSS)*T(NELEM)*DETJAC
C-----------------------------------------------------------------------------------------------------------
C  FORM PRODUCT B-TRANSPOSE * E = BTE
C-----------------------------------------------------------------------------------------------------------
          KORDER = NSIZE/2
          DO 80 J=1,KORDER
            L=2*J
            K=L-1
C-----------------------------------------------------------------------------------------------------------
```

```
C CARRY OUT ONLY NON ZERO MULTIPLICATIONS
C---------------------------------------------------------------------------------------------------
          DO 60 N=1,3
            BTE(K,N)=B(1,K)*EM(1,N) + B(3,K)*EM(3,N)
            BTE(L,N)=B(2,L)*EM(2,N) + B(3,L)*EM(3,N)
 60       CONTINUE
 80     CONTINUE
C---------------------------------------------------------------------------------------------------
C LOOP ON THE ROWS
C---------------------------------------------------------------------------------------------------
        DO 140 NROW=1,NSIZE
          DO 120 NCOL=NROW,NSIZE
            DUM=0.D0
C---------------------------------------------------------------------------------------------------
C LOOP FOR PRODUCT BTE
C---------------------------------------------------------------------------------------------------
            DO 100 J=1,3
              DUM=DUM + BTE(NROW,J)*B(J,NCOL)
 100        CONTINUE
            KEL(NROW,NCOL)=KEL(NROW,NCOL) + DUM*DV
 120      CONTINUE
 140    CONTINUE
 160  CONTINUE
 180 CONTINUE
C---------------------------------------------------------------------------------------------------
C FILL IN LOWER TRIANGLE OF KEL BY SYMMETRY
C---------------------------------------------------------------------------------------------------
    DO 200 K=1,NSIZE
      DO 200 L=K,NSIZE
        KEL(L,K)=KEL(K,L)
 200 CONTINUE
C---------------------------------------------------------------------------------------------------
C CONDENSATION OF STIFEL
C---------------------------------------------------------------------------------------------------
    IF(INCOMP .EQ. 1) THEN
      DO 340 K=1,4
        LL = NSIZE - K
        KK = LL + 1
        DO 320 L=1,LL
          IF(KEL(KK,L) .EQ. 0.) GO TO 320
            DUM = KEL(KK,L)/KEL(KK,KK)
            DO 300 M=1,L
              KEL(L,M) = KEL(L,M) - KEL(KK,M)*DUM
 300        CONTINUE
 320    CONTINUE
 340  CONTINUE
      LL=NSIZE-4
      DO 350 K=1,LL
        DO 350 L=1,K
          KEL(L,K) = KEL(K,L)
 350  CONTINUE
    END IF
C---------------------------------------------------------------------------------------------------
C FORM ACTUAL STIFFNESS MATRIX
C---------------------------------------------------------------------------------------------------
    DO 400 I = 1,MORDER
      DO 400 J = 1,MORDER
        STIFEL(I,J)=KEL(I,J)
 400 CONTINUE
C
    RETURN
    END
```

```
C
      SUBROUTINE SHAPEF(XI,ET,XX,YY,DETJAC,B,NORDER,INCOMP)
C----------------------------------------------------------------------------------
C  SUBROUTINE FOR CALCULATION OF THE SHAPE FUNCTIONS AND THE
C  STRAIN-DISPLACEMENT MATRIX OF AN ISOPARAMETRIC QUAD ELEMENT
C----------------------------------------------------------------------------------
      IMPLICIT REAL*8 (A-H,O-Z)
      DOUBLE PRECISION JACOB,NU
      DIMENSION RXI(5),RET(5),RK(5),RL(5)
      DIMENSION XX(5),YY(5),B(3,14),EM(3,3),ENXI(7),ENET(7)
C
      COMMON/Q4/EN(5),JACOB(2,2)
      COMMON/GLOB/X(640),Y(640),A(640),E(640),NU(640),T(640),D(640)
C
      DATA RXI/-1.,1.,1.,-1.,1./,RET/-1.,-1.,1.,1.,1./
      DATA RK/1.,1.,1.,1.,0./
C
      RL(1)=1.
      RL(2)=1.
      RL(3)=0.
      RL(4)=0.
      RL(5)=-2.
      IF(NORDER .EQ. 4) THEN
        RL(1)=0.
        RL(2)=0.
      END IF
C----------------------------------------------------------------------------------
C  CALCULATE THE SHAPE FUNCTIONS AND THEIR DERIVATIVES
C----------------------------------------------------------------------------------
      DO 20 L=1,NORDER
        F1=(1. + RXI(L)*XI)
        F2=(1. + RET(L)*ET)
        F3=(1. - XI*XI)
        F4=(1. - ET)
        EN(L)=RK(L)*F1*F2/4. - RL(L)*F3*F4/4.
        ENXI(L)=RK(L)*RXI(L)*F2/4. + RL(L)*XI*F4/2.
        ENET(L)=RK(L)*RET(L)*F1/4. + RL(L)*F3/4.
   20 CONTINUE
C----------------------------------------------------------------------------------
C  COMPUTE DERIVATIVES OF INCOMPATIBLE MODES
C----------------------------------------------------------------------------------
      MORDER = NORDER
      IF(INCOMP .EQ. 1) THEN
        LMIN = NORDER + 1
        LMAX = NORDER + 2
        ENXI(LMIN) = -2.0D0*XI
        ENET(LMIN) = 0.0D0
        ENXI(LMAX) = 0.0D0
        ENET(LMAX) = -2.0D0*ET
        MORDER = LMAX
      END IF
C
      NSIZE = 2*MORDER
      DO 40 I=1,3
        DO 40 J=1,NSIZE
          B(I,J)=0.0D0
   40 CONTINUE
      JACOB(1,1)=0.0D0
      JACOB(1,2)=0.0D0
      JACOB(2,1)=0.0D0
      JACOB(2,2)=0.0D0
C----------------------------------------------------------------------------------
```

```
C  FIND JACOBIAN AND REPLACE IT BY ITS DETERMINANT
C-----------------------------------------------------------------------------------
      DO 60 L=1,NORDER
        JACOB(1,1)=JACOB(1,1)+ENXI(L)*XX(L)
        JACOB(1,2)=JACOB(1,2)+ENXI(L)*YY(L)
        JACOB(2,1)=JACOB(2,1)+ENET(L)*XX(L)
        JACOB(2,2)=JACOB(2,2)+ENET(L)*YY(L)
   60 CONTINUE
      DETJAC=JACOB(1,1)*JACOB(2,2)-JACOB(1,2)*JACOB(2,1)
      F5=JACOB(1,1)/DETJAC
      JACOB(1,1)=JACOB(2,2)/DETJAC
      JACOB(1,2)=-JACOB(1,2)/DETJAC
      JACOB(2,1)=-JACOB(2,1)/DETJAC
      JACOB(2,2)=F5
C-----------------------------------------------------------------------------------
C  FORM STRAIN-DISPLACEMENT MATRIX B
C-----------------------------------------------------------------------------------
      DO 80 J=1,MORDER
        L=2*J
        K=L-1
        B(1,K)=JACOB(1,1)*ENXI(J) + JACOB(1,2)*ENET(J)
        B(2,L)=JACOB(2,1)*ENXI(J) + JACOB(2,2)*ENET(J)
        B(3,K)=B(2,L)
        B(3,L)=B(1,K)
   80 CONTINUE
C
      RETURN
      END
C
      SUBROUTINE REL(NELEM1,NELEM2,N1,N2,N3,N4,N5,N11,N22,N33,N44,N55,Q)
C-----------------------------------------------------------------------------------
C  SUBROUTINE FOR ESTABLISHING THE LINEAR RELATIONSHIP BETWEEN
C  MID-SIDE NODES OF ADJACENT QUAD5 ELEMENTS SO THAT NORMAL AND SHEAR
C  STRESSES WILL BE CONTINUOUS AT THAT INTERFACE
C-----------------------------------------------------------------------------------
      IMPLICIT REAL*8 (A-H,O-Z)
      DOUBLE PRECISION NUA,NUB,NU
      DIMENSION XA(5),YA(5),XB(5),YB(5),TRAN1(2,3),BA(3,14),BB(3,14)
      DIMENSION EA(3,3),EB(3,3),PROD1(3,10),TRAN2(2,3),QA(2,10),QB(2,10)
      DIMENSION Q1(2,2),Q2(2,12),Q(2,12),XX1(5),XX2(5),YY1(5),YY2(5)
      INTEGER NELEM1,NELEM2,N1,N2,N3,N4,N5,N11,N22,N33,N44,N55
C
      COMMON/GLOB/X(640),Y(640),A(640),E(640),NU(640),T(640),D(640)
C
      PI=DACOS(-1.0D0)
      DETJAC=0.0D0
      XX1(1)=X(N1)
      XX1(2)=X(N2)
      XX1(3)=X(N3)
      XX1(4)=X(N4)
      XX1(5)=X(N5)
      YY1(1)=Y(N1)
      YY1(2)=Y(N2)
      YY1(3)=Y(N3)
      YY1(4)=Y(N4)
      YY1(5)=Y(N5)
      XX2(1)=X(N11)
      XX2(2)=X(N22)
      XX2(3)=X(N33)
      XX2(4)=X(N44)
      XX2(5)=X(N55)
      YY2(1)=Y(N11)
```

```
      YY2(2)=Y(N22)
      YY2(3)=Y(N33)
      YY2(4)=Y(N44)
      YY2(5)=Y(N55)
C
      CALL TRANSF(NELEM1,XX1(1),XX1(2),YY1(1),YY1(2),TRAN1,2,3)
      CALL TRANSF(NELEM2,XX2(1),XX2(2),YY2(1),YY2(2),TRAN2,2,3)
C
      XI=0.0D0
      ET=-1.0D0
      CALL SHAPEF(XI,ET,XX1,YY1,DETJAC,BA,5,0)
      CALL YOUNG(EA,E(NELEM1),NU(NELEM1),3,0)
      CALL SHAPEF(XI,ET,XX2,YY2,DETJAC,BB,5,0)
      CALL YOUNG(EB,E(NELEM2),NU(NELEM2),3,0)
C-----------------------------------------------------------------------
C  CALCULATE QA AND QB MATRICES
C-----------------------------------------------------------------------
      CALL MATMAT(3,3,10,EA,BA,PROD1)
      CALL MATMAT(2,3,10,TRAN1,PROD1,QA)
      CALL MATMAT(3,3,10,EB,BB,PROD1)
      CALL MATMAT(2,3,10,TRAN2,PROD1,QB)
C-----------------------------------------------------------------------
C  CALCULATE Q1 AND Q2 MATRICES
C-----------------------------------------------------------------------
      F1=(QB(1,9)-QA(1,9))*(QB(2,10)-QA(2,10))
      F2=(QB(1,10)-QA(1,10))*(QB(2,9)-QA(2,9))
      DETQ=(F1-F2)
      Q1(1,1)=(QB(2,10)-QA(2,10))/DETQ
      Q1(1,2)=-(QB(1,10)-QA(1,10))/DETQ
      Q1(2,1)=-(QB(2,9)-QA(2,9))/DETQ
      Q1(2,2)=(QB(1,9)-QA(1,9))/DETQ
C
      DO 80 L=1,2
        Q2(L,1)=QA(L,7)
        Q2(L,2)=QA(L,8)
        Q2(L,3)=QA(L,1)-QB(L,3)
        Q2(L,4)=QA(L,2)-QB(L,4)
        Q2(L,5)=-QB(L,5)
        Q2(L,6)=-QB(L,6)
        Q2(L,7)=-QB(L,7)
        Q2(L,8)=-QB(L,8)
        Q2(L,9)=QA(L,3)-QB(L,1)
        Q2(L,10)=QA(L,4)-QB(L,2)
        Q2(L,11)=QA(L,5)
        Q2(L,12)=QA(L,6)
   80 CONTINUE
C-----------------------------------------------------------------------
C  FORM PRODUCT Q1*Q2 = Q
C-----------------------------------------------------------------------
      CALL MATMAT(2,2,12,Q1,Q2,Q)
C
      RETURN
      END
C
      SUBROUTINE YOUNG(EM,E,NU,NDIM,NPLANE)
C-----------------------------------------------------------------------
C  SUBROUTINE FOR CALCULATION OF THE ELASTICITY MATRIX
C-----------------------------------------------------------------------
      REAL*8 E,NU,EM(NDIM,NDIM),COEF1,COEF2
C
      DO 20 I=1,NDIM
        DO 20 J=1,NDIM
```

```
          EM(I,J)=0.D0
   20 CONTINUE
C
      IF(NDIM .EQ. 3) THEN
        IF(NPLANE .EQ. 0) THEN
          COEF1=E/(1.D0-NU*NU)
          EM(1,1)=COEF1
          EM(1,2)=COEF1*NU
          EM(2,1)=EM(1,2)
          EM(2,2)=COEF1
          EM(3,3)=COEF1*(1.D0-NU)/2.D0
        ELSE
          COEF2=E/((1.D0+NU)*(1.D0-2.D0*NU))
          EM(1,1)=COEF2*(1.D0-NU)
          EM(1,2)=COEF2*NU
          EM(2,1)=EM(1,2)
          EM(2,2)=EM(1,1)
          EM(3,3)=COEF2*(1.D0-2.D0*NU)/2.D0
        END IF
      ELSE
        COEF2=E/((1.D0+NU)*(1.D0-2.D0*NU))
        EM(1,1)=COEF2*(1.D0-NU)
        EM(1,2)=COEF2*NU
        EM(1,3)=EM(1,2)
        EM(2,1)=EM(1,2)
        EM(2,2)=EM(1,1)
        EM(2,3)=EM(1,2)
        EM(3,1)=EM(1,2)
        EM(3,2)=EM(1,2)
        EM(3,3)=EM(1,1)
        EM(4,4)=COEF2*(1.D0-2.D0*NU)/2.D0
        IF(NDIM .EQ. 6) THEN
          EM(5,5)=EM(1,1)
          EM(6,6)=EM(1,1)
        END IF
      END IF
C
      RETURN
      END
C
      SUBROUTINE MATMAT(M,N,K,A,B,C)
C-----------------------------------------------------------------------
C SUBROUTINE FOR CALCULATION OF A MATRIX-MATRIX PRODUCT
C-----------------------------------------------------------------------
      INTEGER M,N,K,R,S,I
      DOUBLE PRECISION A(M,N),B(N,K),C(M,K),SUM
C
      R=1
C      DO 80 WHILE (R .LE. M)
      DO WHILE (R .LE. M)
          S=1
C          DO 60 WHILE(S .LE. K)
          DO WHILE(S .LE. K)
              SUM=0.0
              I=1
C              DO 40 WHILE(I .LE. N)
              DO WHILE(I .LE. N)
                  SUM=SUM+A(R,I)*B(I,S)
                  I=I+1
C 40              CONTINUE
              ENDDO
              C(R,S)=SUM
```

```
          S=S+1
C 60      CONTINUE
        ENDDO
        R=R+1
C 80 CONTINUE
     ENDDO
C
     RETURN
     END
C
     SUBROUTINE MATVEC(N,M,A,Z,V,NEQNS,MEQNS)
C-----------------------------------------------------------------------
C  SUBROUTINE FOR CALCULATION OF A MATRIX-VECTOR PRODUCT
C-----------------------------------------------------------------------
     DIMENSION A(NEQNS,MEQNS),Z(MEQNS)
     REAL*8 SUM,A,V(N),Z
C
     DO 40 I=1,N
       SUM=0.0
       DO 20 J=1,M
         SUM=SUM+A(I,J)*Z(J)
  20   CONTINUE
       V(I)=SUM
  40 CONTINUE
     RETURN
     END
C
     SUBROUTINE DOT(N,A,B,PRODUC,MEQNS)
C-----------------------------------------------------------------------
C  SUBROUTINE FOR CALCULATION OF A VECTOR-VECTOR DOT PRODUCT
C-----------------------------------------------------------------------
     DIMENSION A(MEQNS),B(MEQNS)
     REAL*8 A,B,PRODUC,SUM
     SUM=0.0
     DO 20 I=1,N
       SUM=SUM + A(I)*B(I)
  20 CONTINUE
     PRODUC=SUM
     RETURN
     END
C
     SUBROUTINE DISPL(N,DOF,NID,NODRED,N1,N2,N3,N4,N5,NELA,NELB,INTER,
    +          XDISP,REACT,MEQNS,NCASE)
C-----------------------------------------------------------------------
C  THIS SUBROUTINE OUTPUTS THE GLOBAL DISPLACEMENTS AND REACTIONS
C  IN THE ORIGINAL USER-SUPPLIED NODE NUMBERING ORDER
C-----------------------------------------------------------------------
     IMPLICIT REAL*8 (A-H,O-Z)
     INTEGER N,DOF,NID(MEQNS),NODRED(MEQNS),N1(MEQNS),N2(MEQNS)
     INTEGER N3(MEQNS),N4(MEQNS),N5(MEQNS),NELA(MEQNS),NELB(MEQNS)
     INTEGER INTER,ELID,ELIDB,NOD1,NOD2,NOD3,NOD4,NOD5
     INTEGER NOD11,NOD22,NOD33,NOD44,NOD55,NN(6)
     DOUBLE PRECISION NU
     DIMENSION XDISP(MEQNS),REACT(MEQNS),Q(2,12),XR(12),XX(2)
     DIMENSION XRDISP(640),RREACT(640)
C
     COMMON/GLOB/X(640),Y(640),A(640),E(640),NU(640),T(640),D(640)
C
     IF(INTER .NE. 0) THEN
      DO 60 I=1,INTER
        ELID=NELA(I)
        ELIDB=NELB(I)
```

```
      NOD1=N1(ELID)
      NOD2=N2(ELID)
      NOD3=N3(ELID)
      NOD4=N4(ELID)
      NOD5=N5(ELID)
      NOD11=N1(ELIDB)
      NOD22=N2(ELIDB)
      NOD33=N3(ELIDB)
      NOD44=N4(ELIDB)
      NOD55=N5(ELIDB)
C-----------------------------------------------------------------------
C CALCULATE THE DISPLACEMENTS OF THE INTERFACE NODES
C-----------------------------------------------------------------------
      IF(NCASE .EQ. 4) THEN
        CALL RELAX(ELID,ELIDB,NOD1,NOD2,NOD3,NOD4,NOD5,NOD11,NOD22
     +             ,NOD33,NOD44,NOD55,Q)
      ELSE
        CALL REL(ELID,ELIDB,NOD1,NOD2,NOD3,NOD4,NOD5,NOD1., .OD22
     +           ,NOD33,NOD44,NOD55,Q)
      END IF
      NN(1)=NOD4
      NN(2)=NOD22
      NN(3)=NOD33
      NN(4)=NOD44
      NN(5)=NOD2
      NN(6)=NOD3
      DO 40 L=1,6
        LK=2*L-1
        LL=2*L
        XR(LK)=XDISP(2*NN(L)-1)
        XR(LL)=XDISP(2*NN(L))
 40     CONTINUE
C
      CALL MATVEC(2,12,Q,XR,XX,2,12)
      XDISP(2*N5(ELID)-1)=XX(1)
      XDISP(2*N5(ELID))=XX(2)
 60   CONTINUE
    END IF
C
    RETURN
    END
C
    SUBROUTINE TRIAX(ELID,N1,N2,N3,R1,Z1,R2,Z2,R3,Z3,E,NU,KEL)
C-----------------------------------------------------------------------
C THIS SUBROUTINE CALCULATES THE ELEMENT STIFFNESS MATRIX OF A
C TRIANGULAR AXISYMMETRIC ELEMENT
C-----------------------------------------------------------------------
    IMPLICIT REAL*8 (A-H,O-Z)
    INTEGER ELID
    REAL*8 NU,KEL(6,6),EM(4,4),H(6,6),F(6,6),HT(6,6),PROD(6,6)
C
C-----------------------------------------------------------------------
C CALCULATE ELASTICITY MATRIX , INITIALIZE (ZERO) ALL MATRICES
C-----------------------------------------------------------------------
    CALL YOUNG(EM,E,NU,4,0)
    RL = R2*(Z3-Z1) + R1*(Z2-Z3) + R3*(Z1-Z2)
    DO 20 I=1,6
      DO 20 J=1,6
        H(I,J)=0.D0
        KEL(I,J)=0.D0
        F(I,J)=0.D0
  20 CONTINUE
```

```
C-----------------------------------------------------------------------------------------------
C  CALCULATE H-MATRIX
C-----------------------------------------------------------------------------------------------
      H(1,1) = (R2*Z3-R3*Z2)/RL
      H(1,3) = (R3*Z1-R1*Z3)/RL
      H(1,5) = (R1*Z2-R2*Z1)/RL
      H(2,2) = H(1,1)
      H(2,4) = H(1,3)
      H(2,6) = H(1,5)
      H(3,1) = (Z2-Z3)/RL
      H(3,3) = (Z3-Z1)/RL
      H(3,5) = (Z1-Z2)/RL
      H(4,2) = H(3,1)
      H(4,4) = H(3,3)
      H(4,6) = H(3,5)
      H(5,1) = (R3-R2)/RL
      H(5,3) = (R1-R3)/RL
      H(5,5) = (R2-R1)/RL
      H(6,2) = H(5,1)
      H(6,4) = H(5,3)
      H(6,6) = H(5,5)
C-----------------------------------------------------------------------------------------------
C  CALCULATE HT-MATRIX : TRANSPOSE OF H
C-----------------------------------------------------------------------------------------------
      DO 40 I=1,6
        DO 40 J=1,6
          HT(I,J)=H(J,I)
   40 CONTINUE
      A = ((R2-R1)*(Z3-Z1) - (R3-R1)*(Z2-Z1))/2.D0
      RC=(R1+R2+R3)/3.D0
      ZC=(Z1+Z2+Z3)/3.D0
C-----------------------------------------------------------------------------------------------
C  CALCULATE F-MATRIX BY NUMERICAL INTEGRATION
C-----------------------------------------------------------------------------------------------
      F(1,1) = EM(2,2)*(1.D0/RC)*A
      F(1,3) =(EM(2,1) + EM(2,2))*A
      F(1,5) = EM(2,2)*(ZC/RC)*A
      F(1,6) = EM(2,3)*A
      F(3,3) =(EM(1,1)+EM(1,2)+EM(2,1)+EM(2,2))*(RC)*A
      F(3,5) =(EM(1,2)+EM(2,2))*ZC*A
      F(3,6) =(EM(1,3)+EM(2,3))*RC*A
      F(4,4) = EM(4,4)*RC*A
      F(4,5) = F(4,4)
      F(5,5) = EM(2,2)*ZC*ZC/RC +
     +         EM(4,4)*RC*A
      F(5,6)= EM(2,3)*ZC*A
      F(6,6) = EM(3,3)*RC*A
C
      DO 60 I=1,5
        DO 60 J=1,6
          F(J,I)=F(I,J)
   60 CONTINUE
C-----------------------------------------------------------------------------------------------
C  FORM PRODUCT HT-F-H : STIFFNESS MATRIX
C-----------------------------------------------------------------------------------------------
      CALL MATMAT(6,6,6,F,H,PROD)
      CALL MATMAT(6,6,6,HT,PROD,KEL)
C
      RETURN
      END
C
      SUBROUTINE QUADAX(NGAUSS,NELEM,NORDER,N1,N2,N3,N4,N5,STIFEL,NDIM)
```

```
C----------------------------------------------------------------------------------------
C  SUBROUTINE FOR FORMATION AND NUMERICAL INTEGRATION OF ISOPARAMETRIC
C  ELEMENT STIFFNESS MATRIX FOR AXISYMMETRIC ANALYSIS
C----------------------------------------------------------------------------------------
      IMPLICIT REAL*8 (A-H,O-Z)
      DOUBLE PRECISION JACOB,NU,KEL(10,10)
      DIMENSION XX(5),YY(5),PLACE(3,3),WGT(3,3),B(4,10),BTE(10,4)
      DIMENSION EM(4,4),STIFEL(NDIM,NDIM)
C
      COMMON/Q4/EN(5),JACOB(2,2)
      COMMON/GLOB/X(640),Y(640),A(640),E(640),NU(640),T(640),D(640)
C
      DATA PLACE(1,1),PLACE(2,1),PLACE(2,3),PLACE(3,1),PLACE(3,2)/
     +     5*0.000000000000000D0/
      DATA PLACE(1,2)/-0.577350269189626D0/
      DATA PLACE(2,2)/ 0.577350269189626D0/
      DATA PLACE(1,3)/-0.774596669241483D0/
      DATA PLACE(3,3)/ 0.774596669241483D0/
      DATA WGT(1,1)/2.000000000000000D0/,WGT(2,3)/0.888888888888889D0/
      DATA WGT(1,2),WGT(2,2)/2*1.000000000000000D0/
      DATA WGT(2,1),WGT(3,1),WGT(3,2)/3*0.000000000000000D0/
      DATA WGT(1,3),WGT(3,3)/2*0.555555555555556D0/
C
      PI=DACOS(-1.D0)
      CALL YOUNG(EM,E(NELEM),NU(NELEM),4,0)
      MORDER=2*NORDER
      NSIZE = NDIM
C
      XX(1)=X(N1)
      XX(2)=X(N2)
      XX(3)=X(N3)
      XX(4)=X(N4)
      YY(1)=Y(N1)
      YY(2)=Y(N2)
      YY(3)=Y(N3)
      YY(4)=Y(N4)
      IF(NORDER .NE. 4) THEN
        XX(5)=X(N5)
        YY(5)=Y(N5)
      END IF
C
C----------------------------------------------------------------------------------------
C  CLEAR UPPER TRIANGLE OF ELEMENT STIFFNESS MATRIX
C----------------------------------------------------------------------------------------
      DO 40 K=1,NSIZE
        DO 40 L=K,NSIZE
          KEL(K,L)=0.D0
   40 CONTINUE
      DO 180 NA=1,NGAUSS
        XI=PLACE(NA,NGAUSS)
        DO 160 NB=1,NGAUSS
          ET=PLACE(NB,NGAUSS)
          CALL SHAPAX(XI,ET,XX,YY,DETJAC,B,NORDER)
          R=0.D0
          DO 50 NR=1,NORDER
            R=R+EN(NR)*XX(NR)
   50     CONTINUE
          DV=WGT(NA,NGAUSS)*WGT(NB,NGAUSS)*DETJAC
C
          KORDER = NSIZE/2
          DO 80 J=1,KORDER
            L=2*J
```

```
          K=L-1
          DO 60 N=1,3
            BTE(K,N)=B(1,K)*EM(1,N) + B(2,K)*EM(2,N)
            BTE(L,N)=B(3,L)*EM(3,N)
 60       CONTINUE
          BTE(L,4)=B(4,L)*EM(4,4)
          BTE(K,4)=B(4,K)*EM(4,4)
 80     CONTINUE
C
        DO 140 NROW=1,NSIZE
          DO 120 NCOL=NROW,NSIZE
            DUM=0.D0
C
            DO 100 J=1,4
              DUM=DUM + BTE(NROW,J)*B(J,NCOL)
 100        CONTINUE
            KEL(NROW,NCOL)=KEL(NROW,NCOL) + DUM*DV*R
 120      CONTINUE
 140    CONTINUE
 160  CONTINUE
 180 CONTINUE
C
    DO 200 K=1,NSIZE
      DO 200 L=K,NSIZE
        KEL(L,K)=KEL(K,L)
 200 CONTINUE
C-----------------------------------------------------------------------
C FORM ACTUAL STIFFNESS MATRIX
C-----------------------------------------------------------------------
    DO 400 I = 1,MORDER
      DO 400 J = 1,MORDER
        STIFEL(I,J)=KEL(I,J)
 400 CONTINUE
C
    RETURN
    END
C
    SUBROUTINE SHAPAX(XI,ET,XX,YY,DETJAC,B,NORDER)
C-----------------------------------------------------------------------
C SUBROUTINE FOR CALCULATION OF THE SHAPE FUNCTIONS AND THE
C STRAIN-DISPLACEMENT MATRIX OF AN ISOPARAMETRIC QUAD ELEMENT
C FOR AXISYMMETRIC ANALYSIS
C-----------------------------------------------------------------------
    IMPLICIT REAL*8 (A-H,O-Z)
    DOUBLE PRECISION JACOB,NU
    DIMENSION RXI(5),RET(5),RK(5),RL(5)
    DIMENSION XX(5),YY(5),B(4,10),EM(4,4),ENXI(5),ENET(5)
C
    COMMON/Q4/EN(5),JACOB(2,2)
    COMMON/GLOB/X(640),Y(640),A(640),E(640),NU(640),T(640),D(640)
C
    DATA RXI/-1.,1.,1.,-1.,1./,RET/-1.,-1.,1.,1.,1./
    DATA RK/1.,1.,1.,1.,0./
C
    RL(1)=1.
    RL(2)=1.
    RL(3)=0.
    RL(4)=0.
    RL(5)=-2.
    IF(NORDER .EQ. 4) THEN
      RL(1)=0.
      RL(2)=0.
```

```
      END IF
C-----------------------------------------------------------------------
C  CALCULATE THE SHAPE FUNCTIONS AND THEIR DERIVATIVES
C-----------------------------------------------------------------------
      DO 20 L=1,NORDER
        F1=(1. + RXI(L)*XI)
        F2=(1. + RET(L)*ET)
        F3=(1. - XI*XI)
        F4=(1. - ET)
        EN(L)=RK(L)*F1*F2/4. - RL(L)*F3*F4/4.
        ENXI(L)=RK(L)*RXI(L)*F2/4. + RL(L)*XI*F4/2.
        ENET(L)=RK(L)*RET(L)*F1/4. + RL(L)*F3/4.
   20 CONTINUE
C
      MORDER=NORDER
      NSIZE = 2*MORDER
      DO 40 I=1,4
        DO 40 J=1,NSIZE
          B(I,J)=0.D0
   40 CONTINUE
      JACOB(1,1)=0.D0
      JACOB(1,2)=0.D0
      JACOB(2,1)=0.D0
      JACOB(2,2)=0.D0
C-----------------------------------------------------------------------
C  FIND JACOBIAN AND REPLACE IT BY ITS DETERMINANT
C-----------------------------------------------------------------------
      DO 60 L=1,NORDER
        JACOB(1,1)=JACOB(1,1)+ENXI(L)*XX(L)
        JACOB(1,2)=JACOB(1,2)+ENXI(L)*YY(L)
        JACOB(2,1)=JACOB(2,1)+ENET(L)*XX(L)
        JACOB(2,2)=JACOB(2,2)+ENET(L)*YY(L)
   60 CONTINUE
      DETJAC=JACOB(1,1)*JACOB(2,2)-JACOB(1,2)*JACOB(2,1)
      F5=JACOB(1,1)/DETJAC
      JACOB(1,1)=JACOB(2,2)/DETJAC
      JACOB(1,2)=-JACOB(1,2)/DETJAC
      JACOB(2,1)=-JACOB(2,1)/DETJAC
      JACOB(2,2)=F5
C-----------------------------------------------------------------------
C  FORM STRAIN-DISPLACEMENT MATRIX B
C-----------------------------------------------------------------------
      DO 80 J=1,MORDER
        L=2*J
        K=L-1
        B(1,K)=JACOB(1,1)*ENXI(J) + JACOB(1,2)*ENET(J)
        B(3,L)=JACOB(2,1)*ENXI(J) + JACOB(2,2)*ENET(J)
        B(4,K)=B(3,L)
        B(4,L)=B(1,K)
   80 CONTINUE
      R=0.D0
      DO 100 I=1,MORDER
        IF(XX(I) .EQ. 0.D0) THEN
          XX(I) = 1.D-6
        END IF
        R=R+EN(I)*XX(I)
  100 CONTINUE
      DO 120 I=1,MORDER
        L=2*I - 1
        B(2,L)=EN(I)/R
  120 CONTINUE
C
```

```
      RETURN
      END
C
      SUBROUTINE RELAX(NELEM1,NELEM2,N1,N2,N3,N4,N5,N11,N22,N33,N44,N55,
     +               Q)
C------------------------------------------------------------------------------
C  SUBROUTINE FOR ESTABLISHING THE LINEAR RELATIONSHIP BETWEEN
C  MID-SIDE NODES OF ADJACENT QUAD5 ELEMENTS SO THAT NORMAL AND SHEAR
C  STRESSES WILL BE CONTINUOUS AT THAT INTERFACE
C  FOR AXISYMMETRIC ANALYSIS
C------------------------------------------------------------------------------
      IMPLICIT REAL*8 (A-H,O-Z)
      DOUBLE PRECISION NUA,NUB,NU
      DIMENSION XA(5),YA(5),XB(5),YB(5),TRAN1(2,4),BA(4,10),BB(4,10)
      DIMENSION EA(4,4),EB(4,4),PROD1(4,10),TRAN2(2,4),QA(2,10),QB(2,10)
      DIMENSION Q1(2,2),Q2(2,12),Q(2,12),XX1(5),XX2(5),YY1(5),YY2(5)
C
      COMMON/GLOB/X(640),Y(640),A(640),E(640),NU(640),T(640),D(640)
C
      PI=DACOS(-1.D0)
      DETJAC=0.D0
      XX1(1)=X(N1)
      XX1(2)=X(N2)
      XX1(3)=X(N3)
      XX1(4)=X(N4)
      XX1(5)=X(N5)
      YY1(1)=Y(N1)
      YY1(2)=Y(N2)
      YY1(3)=Y(N3)
      YY1(4)=Y(N4)
      YY1(5)=Y(N5)
      XX2(1)=X(N11)
      XX2(2)=X(N22)
      XX2(3)=X(N33)
      XX2(4)=X(N44)
      XX2(5)=X(N55)
      YY2(1)=Y(N11)
      YY2(2)=Y(N22)
      YY2(3)=Y(N33)
      YY2(4)=Y(N44)
      YY2(5)=Y(N55)
C
      CALL TRANSF(NELEM1,XX1(1),XX1(2),YY1(1),YY1(2),TRAN1,2,4)
      CALL TRANSF(NELEM2,XX2(1),XX2(2),YY2(1),YY2(2),TRAN2,2,4)
C
      XI=0.D0
      ET=-1.D0
      CALL SHAPAX(XI,ET,XX1,YY1,DETJAC,BA,5)
      CALL YOUNG(EA,E(NELEM1),NU(NELEM1),4,0)
      CALL SHAPAX(XI,ET,XX2,YY2,DETJAC,BB,5)
      CALL YOUNG(EB,E(NELEM2),NU(NELEM2),4,0)
C------------------------------------------------------------------------------
C  CALCULATE QA AND QB MATRICES
C------------------------------------------------------------------------------
      CALL MATMAT(4,4,10,EA,BA,PROD1)
      CALL MATMAT(2,4,10,TRAN1,PROD1,QA)
      CALL MATMAT(4,4,10,EB,BB,PROD1)
      CALL MATMAT(2,4,10,TRAN2,PROD1,QB)
C------------------------------------------------------------------------------
C  CALCULATE Q1 AND Q2 MATRICES
C------------------------------------------------------------------------------
      F1=(QB(1,9)-QA(1,9))*(QB(2,10)-QA(2,10))
```

```
      F2=(QB(1,10)-QA(1,10))*(QB(2,9)-QA(2,9))
      DETQ=(F1-F2)
      Q1(1,1)=(QB(2,10)-QA(2,10))/DETQ
      Q1(1,2)=-(QB(1,10)-QA(1,10))/DETQ
      Q1(2,1)=-(QB(2,9)-QA(2,9))/DETQ
      Q1(2,2)=(QB(1,9)-QA(1,9))/DETQ
C
      DO 80 L=1,2
        Q2(L,1)=QA(L,7)
        Q2(L,2)=QA(L,8)
        Q2(L,3)=QA(L,1)-QB(L,3)
        Q2(L,4)=QA(L,2)-QB(L,4)
        Q2(L,5)=-QB(L,5)
        Q2(L,6)=-QB(L,6)
        Q2(L,7)=-QB(L,7)
        Q2(L,8)=-QB(L,8)
        Q2(L,9)=QA(L,3)-QB(L,1)
        Q2(L,10)=QA(L,4)-QB(L,2)
        Q2(L,11)=QA(L,5)
        Q2(L,12)=QA(L,6)
   80 CONTINUE
C-----------------------------------------------------------------------
C  FORM PRODUCT Q1*Q2 = Q
C-----------------------------------------------------------------------
      CALL MATMAT(2,2,12,Q1,Q2,Q)
C
      RETURN
      END
C
      SUBROUTINE MASS (ELID,ELIDB,NTYPE,N1,N2,N3,N4,N5,N11,N22,N33,N44,
     +                 N55,MGLOB,MEL,MEQNS,NEL,MM,NGAUS1,NGAUS2,ID,
     +                 INCOMP,NCASE)
C-----------------------------------------------------------------------
C  THIS SUBROUTINE CALCULATES THE ELEMENT MASS MATRIX
C  AND ASSEMBLES THE GLOBAL MASS MATRIX
C-----------------------------------------------------------------------
      IMPLICIT REAL*8 (A-H,O-Z)
      DOUBLE PRECISION KPRIM(4,4),MEL(MM,MM),NU
      DOUBLE PRECISION MELA(10,10),MELB(10,10),MELARA(10,12)
      DOUBLE PRECISION MELBRB(10,12),MEL1(12,12),MEL2(12,12)
      DOUBLE PRECISION MGLOB(MEQNS,MEQNS)
      DIMENSION TR(4,4),B(3,10)
      DIMENSION Q(2,12),RA(10,12),RB(10,12),RAT(12,10)
      DIMENSION RBT(12,10)
      INTEGER N1,N2,N3,N4,N5,N11,N22,N33,N44,N55,ELID,ELIDB,NGLOB(12)
      INTEGER NOD(6),INCOMP,ID(2,MEQNS),KKK(12)
C
      COMMON/GLOB/X(640),Y(640),A(640),E(640),NU(640),T(640),D(640)
C
      NNTYPE=NTYPE
      NDIM=NTYPE*2
C
        NOD(1)=N1
        NOD(2)=N2
        NOD(3)=N3
        NOD(4)=N4
        KKK(1)=ID(1,NOD(1))
        KKK(2)=ID(2,NOD(1))
        KKK(3)=ID(1,NOD(2))
        KKK(4)=ID(2,NOD(2))
        KKK(5)=ID(1,NOD(3))
        KKK(6)=ID(2,NOD(3))
```

```
      KKK(7)=ID(1,NOD(4))
      KKK(8)=ID(2,NOD(4))
C
      IF(NTYPE .EQ. 2) THEN
       CALL MBAR(ELID,N1,N2,MEL,NDIM)
      ELSE
       IF(NTYPE .EQ. 3) THEN
        IF(NCASE .EQ. 4) THEN
         CALL MTRIAX(ELID,X(N1),Y(N1),X(N2),Y(N2),X(N3),
     +             Y(N3),T(ELID),D(ELID),MEL)
        ELSE
         CALL MCST(ELID,X(N1),Y(N1),X(N2),Y(N2),X(N3),Y(N3),
     +           T(ELID),D(ELID),MEL)
        END IF
       ELSE
        IF(NTYPE .EQ. 4) THEN
         IF(NCASE .EQ. 4) THEN
          CALL MQUADAX(NGAUS1,ELID,NTYPE,N1,N2,N3,N4,N5,MEL,NDIM)
         ELSE
          CALL MQUAD4(NGAUS2,ELID,NTYPE,N1,N2,N3,N4,N5,MEL,NDIM,
     +              INCOMP)
         END IF
        ELSE
         IF(NCASE .EQ. 4) THEN
          CALL RELAX(ELID,ELIDB,N1,N2,N3,N4,N5,N11,N22,N33,N44,N55,
     +             Q)
          CALL MQUADAX(NGAUS2,ELID,NTYPE,N1,N2,N3,N4,N5,MELA,NDIM)
          CALL MQUADAX(NGAUS2,ELIDB,NTYPE,N11,N22,N33,N44,N55,MELB,
     +              NDIM)
         ELSE
          CALL REL(ELID,ELIDB,N1,N2,N3,N4,N5,N11,N22,N33,N44,N55,Q)
          CALL MQUAD4(NGAUS2,ELID,NTYPE,N1,N2,N3,N4,N5,MELA,NDIM,
     +              INCOMP)
          CALL MQUAD4(NGAUS2,ELIDB,NTYPE,N11,N22,N33,N44,N55,MELB,
     +              NDIM,INCOMP)
         END IF
C
         DO 50 LI=1,8
           DO 50 LJ=1,12
             RA(LI,LJ)=0.D0
             RB(LI,LJ)=0.D0
  50     CONTINUE
         RA(7,1)=1.D0
         RA(8,2)=1.D0
         RA(1,3)=1.D0
         RA(2,4)=1.D0
         DO 55 LK=3,6
           LM=LK+6
           RA(LK,LM)=1.D0
  55     CONTINUE
         DO 58 KK=3,8
           RB(KK,KK)=1.D0
  58     CONTINUE
         RB(1,9)=1.D0
         RB(2,10)=1.D0
         DO 60 K=9,10
           DO 60 L=1,12
             MMM=K-8
             RA(K,L)=Q(MMM,L)
             RB(K,L)=Q(MMM,L)
  60     CONTINUE
C-----------------------------------------------------------------------
```

```
C  FORM MASS MATRIX
C-----------------------------------------------------------------------
        DO 80 M=1,10
          DO 80 N=1,12
            RAT(N,M)=RA(M,N)
            RBT(N,M)=RB(M,N)
 80       CONTINUE
        CALL MATMAT(10,10,12,MELA,RA,MELARA)
        CALL MATMAT(12,10,12,RAT,MELARA,MEL1)
        CALL MATMAT(10,10,12,MELB,RB,MELBRB)
        CALL MATMAT(12,10,12,RBT,MELBRB,MEL2)
        DO 90 K1=1,12
          DO 90 K2=1,12
            MEL(K1,K2)=MEL1(K1,K2)+MEL2(K1,K2)
 90       CONTINUE
       END IF
      END IF
     END IF
C-----------------------------------------------------------------------
C  ASSEMBLE GLOBAL MASS MATRIX
C-----------------------------------------------------------------------
     IF(NTYPE .EQ. 5) THEN
      NOD(1)=N4
      NOD(2)=N22
      NOD(3)=N33
      NOD(4)=N44
      NOD(5)=N2
      NOD(6)=N3
      NNTYPE=6
      KKK(1) =ID(1,NOD(1))
      KKK(2) =ID(2,NOD(1))
      KKK(3) =ID(1,NOD(2))
      KKK(4) =ID(2,NOD(2))
      KKK(5) =ID(1,NOD(3))
      KKK(6) =ID(2,NOD(3))
      KKK(7) =ID(1,NOD(4))
      KKK(8) =ID(2,NOD(4))
      KKK(9) =ID(1,NOD(5))
      KKK(10)=ID(2,NOD(5))
      KKK(11)=ID(1,NOD(6))
      KKK(12)=ID(2,NOD(6))
     END IF
C
     DO 1300 I=1,MM
       IF(KKK(I).GT.0) THEN
         K=KKK(I)
         DO 1250 J=1,MM
           IF(KKK(J).GT.0) THEN
             L=KKK(J)
             MGLOB(K,L)=MGLOB(K,L)+MEL(I,J)
           END IF
 1250    CONTINUE
       END IF
 1300 CONTINUE
C
     RETURN
     END
C
     SUBROUTINE MQUAD4(NGAUSS,NELEM,NORDER,N1,N2,N3,N4,N5,MASSEL,NDIM,
    +                 INCOMP)
C-----------------------------------------------------------------------
C  SUBROUTINE FOR FORMATION AND NUMERICAL INTEGRATION OF ISOPARAMETRIC
```

```
C  ELEMENT MASS MATRIX
C-----------------------------------------------------------------------------------------------------------
      IMPLICIT REAL*8 (A-H,O-Z)
      DOUBLE PRECISION JACOB,NU,MEL(10,10),MASSEL(NDIM,NDIM)
      DIMENSION XX(5),YY(5),PLACE(3,3),WGT(3,3),SHAE(2,10),SHAET(10,2)
      DIMENSION EM(3,3)
C
      COMMON/Q4/EN(5),JACOB(2,2)
      COMMON/GLOB/X(640),Y(640),A(640),E(640),NU(640),T(640),D(640)
C
      DATA PLACE(1,1),PLACE(2,1),PLACE(2,3),PLACE(3,1),PLACE(3,2)/
     +     5*0.000000000000000D0/
      DATA PLACE(1,2)/-0.577350269189626D0/
      DATA PLACE(2,2)/ 0.577350269189626D0/
      DATA PLACE(1,3)/-0.774596669241483D0/
      DATA PLACE(3,3)/ 0.774596669241483D0/
      DATA WGT(1,1)/2.000000000000000D0/,WGT(2,3)/0.888888888888889D0/
      DATA WGT(1,2),WGT(2,2)/2*1.000000000000000D0/
      DATA WGT(2,1),WGT(3,1),WGT(3,2)/3*0.000000000000000D0/
      DATA WGT(1,3),WGT(3,3)/2*0.555555555555556D0/
C
C      PI=DACOS(-1.D0)
      CALL YOUNG(EM,E(NELEM),NU(NELEM),3,0)
      MORDER=2*NORDER
      NSIZE = NDIM + 2*INCOMP
C
      XX(1)=X(N1)
      XX(2)=X(N2)
      XX(3)=X(N3)
      XX(4)=X(N4)
      YY(1)=Y(N1)
      YY(2)=Y(N2)
      YY(3)=Y(N3)
      YY(4)=Y(N4)
      IF(NORDER .NE. 4) THEN
        XX(5)=X(N5)
        YY(5)=Y(N5)
      END IF
C
      DO 40 K=1,NSIZE
        DO 40 L=K,NSIZE
          MEL(K,L)=0.D0
   40 CONTINUE
C
      DO 180 NA=1,NGAUSS
        XI=PLACE(NA,NGAUSS)
        DO 160 NB=1,NGAUSS
          ET=PLACE(NB,NGAUSS)
          CALL MSHAPEF(XI,ET,XX,YY,DETJAC,SHAE,NORDER,INCOMP)
C
          DV=D(NELEM)*WGT(NA,NGAUSS)*WGT(NB,NGAUSS)*T(NELEM)*DETJAC
C
          DO 80 I=1,2
            DO 80 J=1,NSIZE
              SHAET(J,I)=SHAE(I,J)
   80     CONTINUE
C
          DO 140 NROW=1,NSIZE
            DO 120 NCOL=NROW,NSIZE
              DUM=0.D0
              DO 100 J=1,2
                DUM=DUM + SHAET(NROW,J)*SHAE(J,NCOL)
```

```
 100          CONTINUE
             MEL(NROW,NCOL)=MEL(NROW,NCOL) + DUM*DV
 120        CONTINUE
 140      CONTINUE
 160    CONTINUE
 180 CONTINUE
C
      DO 200 K=1,NSIZE
        DO 200 L=K,NSIZE
          MEL(L,K)=MEL(K,L)
 200 CONTINUE
C------------------------------------------------------------------------
C  FORM ACTUAL MASS MATRIX
C------------------------------------------------------------------------
      DO 400 I = 1,MORDER
        DO 400 J = 1,MORDER
          MASSEL(I,J)=MEL(I,J)
 400 CONTINUE
C
      RETURN
      END
C
      SUBROUTINE MSHAPEF(XI,ET,XX,YY,DETJAC,SHAE,NORDER,INCOMP)
C------------------------------------------------------------------------
C  SUBROUTINE FOR CALCULATION OF THE SHAPE FUNCTIONS AND THE
C  STRAIN-DISPLACEMENT MATRIX OF AN ISOPARAMETRIC QUAD ELEMENT
C  FOR AXISYMMETRIC ANALYSIS
C------------------------------------------------------------------------
      IMPLICIT REAL*8 (A-H,O-Z)
      DOUBLE PRECISION JACOB,NU
      DIMENSION RXI(5),RET(5),RK(5),RL(5)
      DIMENSION XX(5),YY(5),SHAE(2,10),EM(3,3),ENXI(7),ENET(7)
C
      COMMON/Q4/EN(5),JACOB(2,2)
      COMMON/GLOB/X(640),Y(640),A(640),E(640),NU(640),T(640),D(640)
C
      DATA RXI/-1.,1.,1.,-1.,1./,RET/-1.,-1.,1.,1.,1./
      DATA RK/1.,1.,1.,1.,0./
C
      RL(1)=1.
      RL(2)=1.
      RL(3)=0.
      RL(4)=0.
      RL(5)=-2.
      IF(NORDER .EQ. 4) THEN
       RL(1)=0.
       RL(2)=0.
      END IF
C------------------------------------------------------------------------
C  CALCULATE THE SHAPE FUNCTIONS AND THEIR DERIVATIVES
C------------------------------------------------------------------------
      DO 20 L=1,NORDER
        F1=(1. + RXI(L)*XI)
        F2=(1. + RET(L)*ET)
        F3=(1. - XI*XI)
        F4=(1. - ET)
        EN(L)=RK(L)*F1*F2/4. - RL(L)*F3*F4/4.
        ENXI(L)=RK(L)*RXI(L)*F2/4. + RL(L)*XI*F4/2.
        ENET(L)=RK(L)*RET(L)*F1/4. + RL(L)*F3/4.
   20 CONTINUE
C------------------------------------------------------------------------
C  COMPUTE DERIVATIVES OF INCOMPATIBLE MODES
```

```
C-----------------------------------------------------------------------------------
      MORDER = NORDER
      IF(INCOMP .EQ. 1) THEN
        LMIN = NORDER + 1
        LMAX = NORDER + 2
        ENXI(LMIN) = -2.0D0*XI
        ENET(LMIN) = 0.0D0
        ENXI(LMAX) = 0.0D0
        ENET(LMAX) = -2.0D0*ET
        MORDER = LMAX
      END IF
C-----------------------------------------------------------------------------------
C  CLEAR ARRAYS JACOB AND SHAE
C-----------------------------------------------------------------------------------
C       MORDER=NORDER
      NSIZE = 2*MORDER
      DO 40 I=1,2
        DO 40 J=1,NSIZE
          SHAE(I,J)=0.D0
   40 CONTINUE
      JACOB(1,1)=0.D0
      JACOB(1,2)=0.D0
      JACOB(2,1)=0.D0
      JACOB(2,2)=0.D0
C-----------------------------------------------------------------------------------
C  FIND JACOBIAN AND REPLACE IT BY ITS DETERMINANT
C-----------------------------------------------------------------------------------
      DO 60 L=1,NORDER
        JACOB(1,1)=JACOB(1,1)+ENXI(L)*XX(L)
        JACOB(1,2)=JACOB(1,2)+ENXI(L)*YY(L)
        JACOB(2,1)=JACOB(2,1)+ENET(L)*XX(L)
        JACOB(2,2)=JACOB(2,2)+ENET(L)*YY(L)
   60 CONTINUE
      DETJAC=JACOB(1,1)*JACOB(2,2)-JACOB(1,2)*JACOB(2,1)
C-----------------------------------------------------------------------------------
C  FORM SHAPE FUNCTION MATRIX SHAE
C-----------------------------------------------------------------------------------
      DO 80 J=1,MORDER
        L=2*J
        K=L-1
        SHAE(1,K)=EN(J)
        SHAE(2,L)=EN(J)
   80 CONTINUE

C
      RETURN
      END
C

C
      SUBROUTINE MQUADAX(NGAUSS,NELEM,NORDER,N1,N2,N3,N4,N5,MASSEL,NDIM)
C-----------------------------------------------------------------------------------
C  SUBROUTINE FOR FORMATION AND NUMERICAL INTEGRATION OF ISOPARAMETRIC
C  ELEMENT MASS MATRIX FOR AXISYMMETRIC ANALYSIS
C-----------------------------------------------------------------------------------
      IMPLICIT REAL*8 (A-H,O-Z)
      DOUBLE PRECISION JACOB,NU,MEL(10,10),MASSEL(NDIM,NDIM)
      DIMENSION XX(5),YY(5),PLACE(3,3),WGT(3,3),SHAE(2,10),SHAET(10,2)
      DIMENSION EM(4,4)
C
      COMMON/Q4/EN(5),JACOB(2,2)
      COMMON/GLOB/X(640),Y(640),A(640),E(640),NU(640),T(640),D(640)
```

```
C
      DATA PLACE(1,1),PLACE(2,1),PLACE(2,3),PLACE(3,1),PLACE(3,2)/
     +     5*0.000000000000000D0/
      DATA PLACE(1,2)/-0.577350269189626D0/
      DATA PLACE(2,2)/ 0.577350269189626D0/
      DATA PLACE(1,3)/-0.774596669241483D0/
      DATA PLACE(3,3)/ 0.774596669241483D0/
      DATA WGT(1,1)/2.000000000000000D0/,WGT(2,3)/0.888888888888889D0/
      DATA WGT(1,2),WGT(2,2)/2*1.000000000000000D0/
      DATA WGT(2,1),WGT(3,1),WGT(3,2)/3*0.000000000000000D0/
      DATA WGT(1,3),WGT(3,3)/2*0.555555555555556D0/
C
      PI=DACOS(-1.D0)
      CALL YOUNG(EM,E(NELEM),NU(NELEM),4,0)
      MORDER=2*NORDER
      NSIZE = NDIM
C
      XX(1)=X(N1)
      XX(2)=X(N2)
      XX(3)=X(N3)
      XX(4)=X(N4)
      YY(1)=Y(N1)
      YY(2)=Y(N2)
      YY(3)=Y(N3)
      YY(4)=Y(N4)
      IF(NORDER .NE. 4) THEN
       XX(5)=X(N5)
       YY(5)=Y(N5)
      END IF
C
      DO 40 K=1,NSIZE
        DO 40 L=K,NSIZE
          MEL(K,L)=0.D0
   40 CONTINUE
C
      DO 180 NA=1,NGAUSS
        XI=PLACE(NA,NGAUSS)
        DO 160 NB=1,NGAUSS
          ET=PLACE(NB,NGAUSS)
          CALL MSHAPAX(XI,ET,XX,YY,DETJAC,SHAE,NORDER)
          R=0.D0
          DO 50 NR=1,NORDER
            R=R+EN(NR)*XX(NR)
   50     CONTINUE
C
          DV=D(NELEM)*WGT(NA,NGAUSS)*WGT(NB,NGAUSS)*DETJAC
C
          DO 80 I=1,2
            DO 80 J=1,NSIZE
              SHAET(J,I)=SHAE(I,J)
   80     CONTINUE
C
          DO 140 NROW=1,NSIZE
            DO 120 NCOL=NROW,NSIZE
              DUM=0.D0
              DO 100 J=1,2
                DUM=DUM + SHAET(NROW,J)*SHAE(J,NCOL)
  100         CONTINUE
              MEL(NROW,NCOL)=MEL(NROW,NCOL) + DUM*DV*R
  120       CONTINUE
  140     CONTINUE
  160   CONTINUE
```

```
 180 CONTINUE
C
     DO 200 K=1,NSIZE
       DO 200 L=K,NSIZE
         MEL(L,K)=MEL(K,L)
 200 CONTINUE
C------------------------------------------------------------------------
C  FORM ACTUAL MASS MATRIX
C------------------------------------------------------------------------
     DO 400 I = 1,MORDER
       DO 400 J = 1,MORDER
         MASSEL(I,J)=MEL(I,J)
 400 CONTINUE
C
     RETURN
     END
C
     SUBROUTINE MSHAPAX(XI,ET,XX,YY,DETJAC,SHAE,NORDER)
C------------------------------------------------------------------------
C  SUBROUTINE FOR CALCULATION OF THE SHAPE FUNCTIONS AND THE
C  STRAIN-DISPLACEMENT MATRIX OF AN ISOPARAMETRIC QUAD ELEMENT
C  FOR AXISYMMETRIC ANALYSIS
C------------------------------------------------------------------------
     IMPLICIT REAL*8 (A-H,O-Z)
     DOUBLE PRECISION JACOB,NU
     DIMENSION RXI(5),RET(5),RK(5),RL(5)
     DIMENSION XX(5),YY(5),SHAE(2,10),EM(4,4),ENXI(5),ENET(5)
C
     COMMON/Q4/EN(5),JACOB(2,2)
     COMMON/GLOB/X(640),Y(640),A(640),E(640),NU(640),T(640),D(640)
C
     DATA RXI/-1.,1.,1.,-1.,1./,RET/-1.,-1.,1.,1.,1./
     DATA RK/1.,1.,1.,1.,0./
C
     RL(1)=1.
     RL(2)=1.
     RL(3)=0.
     RL(4)=0.
     RL(5)=-2.
     IF(NORDER .EQ. 4) THEN
       RL(1)=0.
       RL(2)=0.
     END IF
C------------------------------------------------------------------------
C  CALCULATE THE SHAPE FUNCTIONS AND THEIR DERIVATIVES
C------------------------------------------------------------------------
     DO 20 L=1,NORDER
       F1=(1. + RXI(L)*XI)
       F2=(1. + RET(L)*ET)
       F3=(1. - XI*XI)
       F4=(1. - ET)
       EN(L)=RK(L)*F1*F2/4. - RL(L)*F3*F4/4.
       ENXI(L)=RK(L)*RXI(L)*F2/4. + RL(L)*XI*F4/2.
       ENET(L)=RK(L)*RET(L)*F1/4. + RL(L)*F3/4.
  20 CONTINUE
C
     MORDER=NORDER
     NSIZE = 2*MORDER
     DO 40 I=1,2
       DO 40 J=1,NSIZE
         SHAE(I,J)=0.D0
  40 CONTINUE
```

```
      JACOB(1,1)=0.D0
      JACOB(1,2)=0.D0
      JACOB(2,1)=0.D0
      JACOB(2,2)=0.D0
C------------------------------------------------------------------------
C  FIND JACOBIAN AND REPLACE IT BY ITS DETERMINANT
C------------------------------------------------------------------------
      DO 60 L=1,NORDER
        JACOB(1,1)=JACOB(1,1)+ENXI(L)*XX(L)
        JACOB(1,2)=JACOB(1,2)+ENXI(L)*YY(L)
        JACOB(2,1)=JACOB(2,1)+ENET(L)*XX(L)
        JACOB(2,2)=JACOB(2,2)+ENET(L)*YY(L)
   60 CONTINUE
      DETJAC=JACOB(1,1)*JACOB(2,2)-JACOB(1,2)*JACOB(2,1)
C
      DO 80 J=1,MORDER
        L=2*J
        K=L-1
        SHAE(1,K)=EN(J)
        SHAE(2,L)=EN(J)
   80 CONTINUE
C
      RETURN
      END
C
      SUBROUTINE MBAR (ELID,N1,N2,MELL,NDIM)
C
      IMPLICIT REAL*8(A-H,O-Z)
      DOUBLE PRECISION MELL(NDIM,NDIM),NU
      INTEGER ELID,N1,N2
      COMMON/GLOB/X(640),Y(640),A(640),E(640),NU(640),T(640),D(640)
C
      DO 10 I=1,NDIM
        DO 10 J=1,NDIM
          MELL(I,J)=0.0D0
   10 CONTINUE
      A1=ABS(X(N2)-X(N1))
      A2=ABS(Y(N2)-Y(N1))
      CL=SQRT(A1**2+A2**2)
      CC=D(ELID)*A(ELID)*CL/6.0
C
      DO 20 I=1,NDIM
        MELL(I,I)=2.0
   20 CONTINUE
      MELL(1,3)=1.0
      MELL(3,1)=MELL(1,3)
      MELL(2,4)=1.0
      MELL(4,2)=MELL(2,4)
C
      DO 30 I=1,NDIM
        DO 30 J=1,NDIM
          MELL(I,J)=CC*MELL(I,J)
   30 CONTINUE
C
      RETURN
      END
C
      SUBROUTINE MCST(ELID,X1,Y1,X2,Y2,X3,Y3,T,D,MEL)
C
C------------------------------------------------------------------------
C  THIS SUBROUTINES CALCULATES THE ELEMENT MASS MATRIX OF A
C  CONSTANT STRAIN TRIANGULAR ELEMENT
```

```
C-------------------------------------------------------------------------------------------------------
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER ELID
      REAL*8 X1,Y1,X2,Y2,X3,Y3,T,D,MEL(6,6)
      REAL*8 M1(6,6),A
C
      M1(1,1) = 2.0
      M1(1,2) = 0.0
      M1(1,3) = 1.0
      M1(1,4) = 0.0
      M1(1,5) = 1.0
      M1(1,6) = 0.0
      M1(2,1) = 0.0
      M1(2,2) = 2.0
      M1(2,3) = 0.0
      M1(2,4) = 1.0
      M1(2,5) = 0.0
      M1(2,6) = 1.0
      M1(3,1) = 1.0
      M1(3,2) = 0.0
      M1(3,3) = 2.0
      M1(3,4) = 0.0
      M1(3,5) = 1.0
      M1(3,6) = 0.0
      M1(4,1) = 0.0
      M1(4,2) = 1.0
      M1(4,3) = 0.0
      M1(4,4) = 2.0
      M1(4,5) = 0.0
      M1(4,6) = 1.0
      M1(5,1) = 1.0
      M1(5,2) = 0.0
      M1(5,3) = 1.0
      M1(5,4) = 0.0
      M1(5,5) = 2.0
      M1(5,6) = 0.0
      M1(6,1) = 0.0
      M1(6,2) = 1.0
      M1(6,3) = 0.0
      M1(6,4) = 1.0
      M1(6,5) = 0.0
      M1(6,6) = 2.0
C
      A=(X2*Y3-Y2*X3-X1*Y3+Y1*X3+X1*Y2-Y1*X2)/2.
C
      DO 20 I=1,6
        DO 10 J=1,6
          MEL(I,J) = M1(I,J)*D*A*T/12.0
   10   CONTINUE
   20 CONTINUE
C
      RETURN
      END
C
      SUBROUTINE MTRIAX(ELID,X1,Y1,X2,Y2,X3,Y3,T,D,MEL)
C
C-------------------------------------------------------------------------------------------------------
C  THIS SUBROUTINES CALCULATES THE ELEMENT MASS MATRIX OF A
C  TRIANGULAR AXISYMMETRIC ELEMENT
C-------------------------------------------------------------------------------------------------------
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER ELID
```

```
      REAL*8 X1,Y1,X2,Y2,X3,Y3,T,D,MEL(6,6)
      REAL*8 M1(6,6),A
C
      M1(1,1) = 6.0*X1 + 2.0*X2 + 2.0*X3
      M1(1,2) = 0.0
      M1(1,3) = 2.0*X1 + 2.0*X2 + X3
      M1(1,4) = 0.0
      M1(1,5) = 2.0*X1 + X2 + 2.0*X3
      M1(1,6) = 0.0
      M1(2,1) = 0.0
      M1(2,2) = M1(1,1)
      M1(2,3) = 0.0
      M1(2,4) = M1(1,3)
      M1(2,5) = 0.0
      M1(2,6) = M1(1,5)
      M1(3,1) = M1(1,3)
      M1(3,2) = 0.0
      M1(3,3) = 2.0*X1 + 6.0*X2 + 2.0*X3
      M1(3,4) = 0.0
      M1(3,5) = X1 + 2.0*X2 + 2.0*X3
      M1(3,6) = 0.0
      M1(4,1) = 0.0
      M1(4,2) = M1(2,4)
      M1(4,3) = 0.0
      M1(4,4) = M1(3,3)
      M1(4,5) = 0.0
      M1(4,6) = M1(3,5)
      M1(5,1) = M1(1,5)
      M1(5,2) = 0.0
      M1(5,3) = M1(3,5)
      M1(5,4) = 0.0
      M1(5,5) = 2.0*X1 + 2.0*X2 + 6.0*X3
      M1(5,6) = 0.0
      M1(6,1) = 0.0
      M1(6,2) = M1(2,6)
      M1(6,3) = 0.0
      M1(6,4) = M1(4,6)
      M1(6,5) = 0.0
      M1(6,6) = M1(5,5)
C
      A=(X2*Y3-Y2*X3-X1*Y3+Y1*X3+X1*Y2-Y1*X2)/2.
C
      DO 20 I=1,6
        DO 10 J=1,6
          MEL(I,J) = M1(I,J)*D*A*T/60.0
 10     CONTINUE
 20   CONTINUE
C
      RETURN
      END
C
      SUBROUTINE STATIC (OMEG,T,TT,U,F,X,Q,N,MEQNS,NSTEP,NFREQ)
C--------------------------------------------------------------------------
C     THIS SUBROUTINE CALCULATES THE DISPLACEMENTS OF STATIC PROBLEM
C--------------------------------------------------------------------------
      IMPLICIT REAL*8 (A-H,O-Z)
C
      DOUBLE PRECISION OMEG(MEQNS),T(MEQNS,MEQNS),U(MEQNS,NSTEP),
     +F(MEQNS,NSTEP),X(MEQNS,NSTEP),TT(MEQNS,MEQNS),Q(MEQNS,NSTEP)
C
      IF (NFREQ. NE. 0) THEN
        DO 10 I=1,N
```

```
      DO 10 J=NFREQ+1, N
        T(I,J) = 0.0
        TT(J,I) = 0.0
10    CONTINUE
      END IF
C
      CALL MATMUL (TT,F,Q,MEQNS,MEQNS,NSTEP,N,N,NSTEP)
C
      IF (NFREQ. NE. 0) THEN
        DO 100 I=1,NFREQ
          DO 100 J=1,NSTEP
            U(I,J) = Q(I,J) / ( OMEG(I)**2 )
100   CONTINUE
      END IF
C
      IF (NFREQ. EQ. 0) THEN
        DO 200 I=1,N
          DO 200 J=1,NSTEP
            U(I,J) = Q(I,J) / ( OMEG(I)**2 )
200   CONTINUE
      END IF
C
C-----------------------------------------------------------------------
C  FINDING THE DISPLACEMENTS OF ORIGINAL COORDINATES
C-----------------------------------------------------------------------
      CALL MATMUL (T,U,X,MEQNS,MEQNS,NSTEP,N,N,NSTEP)
      RETURN
      END
C
      SUBROUTINE MODAL (GM,OMEG,T,ZETA,XO,XDO,YO,YDO,WN,F,U,V,X,TIME,
     +DT,TT,M,NSTEP,N,NMODE,TGMT,TGM,MEQNS)
C
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER MEQNS
      DOUBLE PRECISION GM(MEQNS,MEQNS),OMEG(MEQNS),T(MEQNS,MEQNS),
     +ZETA(MEQNS),
     +TT(MEQNS,MEQNS),TGM(MEQNS,MEQNS),XO(MEQNS),XDO(MEQNS),YO(MEQNS),
     +YDO(MEQNS),WN(MEQNS,NSTEP),F(MEQNS,NSTEP),U(MEQNS,NSTEP),
     +V(MEQNS,NSTEP),X(MEQNS,NSTEP),TIME(NSTEP),DT(NSTEP),
     +TGMT(MEQNS,MEQNS)
C
      DT(1)=TIME(1)
      DO 10 I=2,NSTEP
   10 DT(I)=TIME(I)-TIME(I-1)
C-----------------------------------------------------------------------
C   NORMALIZATION OF MODAL MATRIX (M-ORTHOGONALIZATION)
C-----------------------------------------------------------------------
      CALL MATMUL (TT,GM,TGM,MEQNS,MEQNS,MEQNS,N,N,N)
      CALL MATMUL (TGM,T,TGMT,MEQNS,MEQNS,MEQNS,N,N,N)
      DO 50 I=1,NMODE
      DO 40 J=1,N
   40 T(J,I)=T(J,I)/SQRT(TGMT(I,I))
   50 CONTINUE
C-----------------------------------------------------------------------
C   TRANSFORMATION OF INFORMATION TO NORMAL COORDINATES
C-----------------------------------------------------------------------
      DO 60 I=1,NMODE
      YO(I)=0.0
   60 YDO(I)=0.0
      DO 80 I=1,NMODE
      DO 70 J=1,N
      YO(I)=YO(I)+T(J,I)*XO(J)
```

```
   70 YDO(I)=YDO(I)+T(J,I)*XDO(J)
   80 CONTINUE
C
      DO 90 I=1,NMODE
        DO 90 J=1,NSTEP
          WN(I,J)=0.0D0
          DO 90 K=1,NMODE
            WN(I,J)=WN(I,J)+TT(I,K)*F(K,J)
   90 CONTINUE
C
      DO 120 I=1,NMODE
      U1=YO(I)
      V1=YDO(I)
      ZET=ZETA(I)
      OM=OMEG(I)
      OMD=OM*SQRT(1.0-ZET**2)
      DO 110 J=1,NSTEP
      IF (J .EQ. 1) GO TO 100
      U1=U(I,J-1)
      V1=V(I,J-1)
  100 C1=EXP(-ZET*OM*DT(J))
      C2=COS(OMD*DT(J))
      C3=SIN(OMD*DT(J))
      C4=(V1+OM*ZET*U1)/OMD
      C5=OM*ZET/OMD
      C6=WN(I,J)/(OM**2)
      U(I,J)=C1*(U1*C2+C4*C3)+C6*(1.0-C1*(C2+C5*C3))
      V(I,J)=OMD*C1*(-U1*C3+C4*C2-C5*(U1*C2+C4*C3))+C6*OMD*C1*(1.0+C5*
     *C5)*C3
  110 CONTINUE
  120 CONTINUE
C-----------------------------------------------------------------------
C     FINDING THE SOLUTION OF ORIGINAL PROBLEM
C-----------------------------------------------------------------------
      CALL MATMUL (T,U,X,MEQNS,MEQNS,NSTEP,N,N,NSTEP)
C
      RETURN
      END
C
      SUBROUTINE MATMUL(A,B,C,L,M,N,LA,MA,NA)
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(L,M),B(M,N),C(L,N)
C
      DO 10 I=1,LA
        DO 10 J=1,NA
          C(I,J)=0.0
          DO 10 K=1,MA
            C(I,J)=C(I,J)+A(I,K)*B(K,J)
   10 CONTINUE
C
      RETURN
      END
C
      SUBROUTINE FORCE(ELID,N1,N2,X1,Y1,X2,Y2,A,E,MEQNS,XDISP)
C-----------------------------------------------------------------------
C  THIS SUBROUTINE COMPUTES THE LOCAL ELEMENT FORCES AND EXTENSIONS
C  FOR BAR ELEMENTS
C-----------------------------------------------------------------------
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION XDISP(MEQNS)
      INTEGER ELID
```

```
      REAL*8 DELTA(4),XDIF,YDIF,L,C,S,K,TR(4,4),KPRIM(4,4),UPRIM(4)
      REAL*8 FPRIM(4),KEL(4,4)
C
      DELTA(1)=XDISP(2*N1-1)
      DELTA(2)=XDISP(2*N1)
      DELTA(3)=XDISP(2*N2-1)
      DELTA(4)=XDISP(2*N2)
C
      WRITE(2,50) ELID,N1,N2
   50 FORMAT('-  ELEMENT NO. : ',I3,2X,'BAR',' -  ',I3,' TO ',I3)
C
      DX=X2-X1
      DY=Y2-Y1
      RL=DSQRT(DX*DX + DY*DY)
      COSB=DX/RL
      SINB=DY/RL
      EL=(DELTA(3)-DELTA(1))*COSB + (DELTA(4)-DELTA(2))*SINB
      FL=(EL/RL)*E*A
      WRITE(2,200) RL
  200 FORMAT('- MEMBER LENGTH ',T22,': ',E10.3)
      WRITE(2,220) EL
  220 FORMAT('- MEMBER ELONGATION ',T22,': ',E10.3)
      WRITE(2,240) FL
  240 FORMAT('- MEMBER FORCE ',T22,': ',E10.3)
C
      RETURN
      END
C
      SUBROUTINE STR(ELID,N1,N2,N3,X1,Y1,X2,Y2,X3,Y3,E,NU,T,XDISP,MEQNS)
C-----------------------------------------------------------------------------------
C THIS SUBROUTINE CALCULATES THE LOCAL STRESSES FOR C.S.T. ELEMENTS
C-----------------------------------------------------------------------------------
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION XDISP(MEQNS)
      INTEGER ELID
      REAL*8 KEL(6,6),U(6),D1(6),ENERGY,STRESS(3),STRAIN(3),B(3,6)
      REAL*8 EM(3,3),SMAX,SMIN,S1,S2,S3,S4,ANGLE,NU,XDISP
      REAL*8 E1,E2,E3,EMAX,EMIN,PI
C
      PI=DACOS(-1.0D0)
      CALL YOUNG(EM,E,NU,3,0)
      CALL CST(ELID,N1,N2,N3,X1,Y1,X2,Y2,X3,Y3,E,NU,T,B,KEL)
C
      U(1)=XDISP(2*N1-1)
      U(2)=XDISP(2*N1)
      U(3)=XDISP(2*N2-1)
      U(4)=XDISP(2*N2)
      U(5)=XDISP(2*N3-1)
      U(6)=XDISP(2*N3)
C
      CALL MATVEC(3,6,B,U,STRAIN,3,6)
      CALL MATVEC(3,3,EM,STRAIN,STRESS,3,3)
      S1=(STRESS(1)+STRESS(2))/2.
      E1=(STRAIN(1)+STRAIN(2))/2.
      S2=(STRESS(1)-STRESS(2))/2.
      E2=(STRAIN(1)-STRAIN(2))/2.
      S3=DSQRT(S2*S2 + STRESS(3)*STRESS(3))
      E3=DSQRT(E2*E2 + STRAIN(3)*STRAIN(3))
      SMAX=S1+S3
      EMAX=E1+E3
      SMIN=S1-S3
      EMIN=E1-E3
```

```
      IF(S2 .EQ. 0.0) THEN
        ANGLE=PI/2.
      ELSE
        ANGLE=(DATAN(STRESS(3)/S2))/2.
      END IF
      CALL MATVEC(6,6,KEL,U,D1,6,6)
      ENERGY=0.0
      DO 20 I=1,6
         ENERGY = ENERGY+D1(I)*U(I)
   20 CONTINUE
      ENERGY=ENERGY/2.
C
      WRITE(2,50) ELID
   50 FORMAT('- ELEMENT NO. :',I3,' C.S.T.')
      WRITE(2,60) STRESS(1),STRAIN(1)
   60 FORMAT('- S 1',T10,':',E10.3,T30,'E 1',T37,':',E10.3)
      WRITE(2,70) STRESS(2),STRAIN(2)
   70 FORMAT('- S 2',T10,':',E10.3,T30,'E 2',T37,':',E10.3)
      WRITE(2,80) STRESS(3),STRAIN(3)
   80 FORMAT('- T XY',T10,':',E10.3,T30,'E XY',T37,':',E10.3)
      WRITE(2,90) SMAX,EMAX
   90 FORMAT('- S MAX',T10,':',E10.3,T30,'E MAX',T37,':',E10.3)
      WRITE(2,100) SMIN,EMIN
  100 FORMAT('- S MIN',T10,':',E10.3,T30,'E MIN',T37,':',E10.3)
      WRITE(2,110) ANGLE,ENERGY
  110 FORMAT('- ANGLE',T10,':',E10.3,T30,'STRAIN ENERGY ',' : ',E10.3)
C
      RETURN
      END
C
      SUBROUTINE STRES(ELID,N1,N2,N3,N4,N5,X1,Y1,X2,Y2,X3,Y3,X4,Y4,X5,
     +                 Y5,E,NU,T,XDISP,MEQNS,NTYPE,INCOMP)
C-----------------------------------------------------------------------
C  SUBROUTINE FOR CALCULATION OF THE LOCAL STRESSES OF QUAD4 AND
C  QUAD5 ELEMENTS
C-----------------------------------------------------------------------
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER ELID
      DOUBLE PRECISION NU
      DIMENSION XDISP(MEQNS),XI(6),ET(6),EM(3,3),NOD(6),XX(5),YY(5)
      DIMENSION XLOC(10),B(3,14),PROD(3,10),S(3),TRAN(3,3),SLOC(3)
      CHARACTER*5 TYPE
C
      DATA XI/-1.,1.,1.,-1.,0.,0./
C
      ET(1)=-1.
      ET(2)=-1.
      ET(3)=1.
      ET(4)=1.
      ET(5)=-1.
      ET(6)=0.
      IF(NTYPE .EQ. 4) THEN
        ET(5)=0.
        TYPE='QUAD4'
      ELSE
        TYPE='QUAD5'
      END IF
      WRITE(2,20) ELID,TYPE
   20 FORMAT('- ELEMENT NO. :',I3,2X,A5)
      MSIZE=2*NTYPE
      NSIZE=NTYPE+1
      CALL YOUNG(EM,E,NU,3,0)
```

```
C
      NOD(1)=N1
      NOD(2)=N2
      NOD(3)=N3
      NOD(4)=N4
      NOD(5)=N5
      NOD(6)=0
      XX(1)=X1
      XX(2)=X2
      XX(3)=X3
      XX(4)=X4
      XX(5)=X5
      YY(1)=Y1
      YY(2)=Y2
      YY(3)=Y3
      YY(4)=Y4
      YY(5)=Y5
C
      DO 40 I=1,NTYPE
        XLOC(2*I-1)=XDISP(2*NOD(I)-1)
        XLOC(2*I)=XDISP(2*NOD(I))
   40 CONTINUE
      WRITE(2,50)
   50 FORMAT('-',T2,'NODE',T12,'S 11',T24,'S 22',T36,'T XY',T48,'S MAX',
     +       T60,'S MIN',T72,'ANGLE')
C
      DO 80 I=1,NSIZE
        CALL SHAPEF(XI(I),ET(I),XX,YY,DETJAC,B,NTYPE,INCOMP)
        CALL MATMAT(3,3,MSIZE,EM,B,PROD)
        CALL MATVEC(3,MSIZE,PROD,XLOC,S,3,MSIZE)
        IF(NTYPE .EQ. 5) THEN
          IF(I .EQ. 5) THEN
            CALL TRANSF(ELID,X1,X2,Y1,Y2,TRAN,3,3)
            CALL MATVEC(3,3,TRAN,S,SLOC,3,3)
            S(1)=SLOC(1)
            S(2)=SLOC(2)
            S(3)=SLOC(3)
          END IF
        END IF
        CALL PRINC(NOD(I),S)
   80 CONTINUE
      WRITE(2,100)
  100 FORMAT('-')
C
      RETURN
      END
C
      SUBROUTINE PRINC(NODE,S)
C-----------------------------------------------------------------------
C SUBROUTINE FOR CALCULATION OF THE PRINCIPAL STRESSES
C-----------------------------------------------------------------------
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER NODE
      DIMENSION S(3)
C
      PI=DACOS(-1.D0)
      S1=(S(1)+S(2))/2.
      S2=(S(1)-S(2))/2.
      S3=DSQRT(S2*S2 + S(3)*S(3))
      SMAX=S1+S3
      SMIN=S1-S3
      IF(S2 .EQ. 0.D0) THEN
```

```
      ANGLE=PI/2.
     ELSE
      ANGLE=(DATAN(S(3)/S2))/2.
     END IF
     ANGLE=(ANGLE*180.D0)/PI
C
     WRITE(2,40) NODE,S(1),S(2),S(3),SMAX,SMIN,ANGLE
  40 FORMAT('-',I3,T9,E10.3,T21,E10.3,T33,E10.3,T45,E10.3,T57,E10.3,
    +       T69,E10.3)
C
     RETURN
     END
C
     SUBROUTINE TRANSF(ELID,X1,X2,Y1,Y2,TRAN,NROW,NCOL)
C-----------------------------------------------------------------------
C SUBROUTINE FOR CALCULATION OF THE STRESS TRANSFORMATION MATRIX
C FOR PLANE STRESS AND AXISYMMETRIC ANALYSES
C-----------------------------------------------------------------------
     IMPLICIT REAL*8(A-H,O-Z)
     INTEGER ELID
     DIMENSION TRAN(NROW,NCOL)
C
     XA=X1-X2
     YA=Y2-Y1
     RL = DSQRT(XA*XA + YA*YA)
     C = YA/RL
     S = XA/RL
C
     IF(NROW .EQ. 2) THEN
      IF(NCOL .EQ. 3) THEN
        TRAN(1,1) = C*C
        TRAN(1,2) = S*S
        TRAN(1,3) = 2.D0*C*S
        TRAN(2,1) = -C*S
        TRAN(2,2) = C*S
        TRAN(2,3) = C*C - S*S
      ELSE
        TRAN(1,1) = C*C
        TRAN(1,2) = 0.D0
        TRAN(1,3) = S*S
        TRAN(1,4) = 2.D0*C*S
        TRAN(2,1) = -C*S
        TRAN(2,2) = 0.D0
        TRAN(2,3) = C*S
        TRAN(2,4) = C*C - S*S
      END IF
     END IF
     IF(NROW .EQ. 3) THEN
      TRAN(1,1) = C*C
      TRAN(1,2) = S*S
      TRAN(1,3) = 2.D0*C*S
      TRAN(2,1) = TRAN(1,2)
      TRAN(2,2) = TRAN(1,1)
      TRAN(2,3) = -TRAN(1,3)
      TRAN(3,1) = -C*S
      TRAN(3,2) = -TRAN(3,1)
      TRAN(3,3) = C*C - S*S
     END IF
     IF(NROW .EQ. 4) THEN
      TRAN(1,1) = C*C
      TRAN(1,2) = 0.D0
      TRAN(1,3) = S*S
```

```
      TRAN(1,4) = 2.D0*C*S
      TRAN(2,1) = 0.D0
      TRAN(2,2) = 1.D0
      TRAN(2,3) = 0.D0
      TRAN(2,4) = 0.D0
      TRAN(3,1) = TRAN(1,3)
      TRAN(3,2) = 0.D0
      TRAN(3,3) = TRAN(1,1)
      TRAN(3,4) = -TRAN(1,4)
      TRAN(4,1) = -C*S
      TRAN(4,2) = 0.D0
      TRAN(4,3) = -TRAN(4,1)
      TRAN(4,4) = C*C -S*S
     END IF
C
     RETURN
     END
C
     SUBROUTINE STRIAX(ELID,N1,N2,N3,R1,Z1,R2,Z2,R3,Z3,E,NU,XDISP,
    +                  MEQNS)
C-----------------------------------------------------------------------------
C  SUBROUTINE FOR CALCULATION OF THE STRESSES FOR AN
C  AXISYMMETRIC TRIANGULAR ELEMENT
C-----------------------------------------------------------------------------
     IMPLICIT REAL*8 (A-H,O-Z)
     INTEGER ELID
     REAL*8 NU
     DIMENSION EM(4,4),H(6,6),G(4,6),S(4)
     DIMENSION XDISP(MEQNS),XLOC(6),PROD1(4,6),PROD2(4,6)
C
     DO 20 I=1,6
       DO 20 J=1,6
         H(I,J)=0.D0
  20 CONTINUE
     DO 40 I=1,4
       DO 40 J=1,6
         G(I,J)=0.D0
  40 CONTINUE
C
     RL=R2*(Z3-Z1) + R1*(Z2-Z3) + R3*(Z1-Z2)
     CALL YOUNG(EM,E,NU,4,0)
C-----------------------------------------------------------------------------
C  CALCULATE H-MATRIX
C-----------------------------------------------------------------------------
     H(1,1)=(R2*Z3-R3*Z2)/RL
     H(1,3)=(R3*Z1-R1*Z3)/RL
     H(1,5)=(R1*Z2-R2*Z1)/RL
     H(2,2)=H(1,1)
     H(2,4)=H(1,3)
     H(2,6)=H(1,5)
     H(3,1)=(Z2-Z3)/RL
     H(3,3)=(Z3-Z1)/RL
     H(3,5)=(Z1-Z2)/RL
     H(4,2)=H(3,1)
     H(4,4)=H(3,3)
     H(4,6)=H(3,5)
     H(5,1)=(R3-R2)/RL
     H(5,3)=(R1-R3)/RL
     H(5,5)=(R2-R1)/RL
     H(6,2)=H(5,1)
     H(6,4)=H(5,3)
     H(6,6)=H(5,5)
```

```
C------------------------------------------------------------------------------
C  CALCULATE G-MATRIX
C------------------------------------------------------------------------------
      R=(R1+R2+R3)/3.D0
      Z=(Z1+Z2+Z3)/3.D0
      G(1,3)=1.D0
      G(2,1)=1.D0/R
      G(2,3)=1.D0
      G(2,5)=Z/R
      G(3,6)=1.D0
      G(4,4)=1.D0
      G(4,5)=1.D0
C------------------------------------------------------------------------------
C  CALCULATE STRESSES : S = EM*G*H*XLOC
C------------------------------------------------------------------------------
      XLOC(1)=XDISP(2*N1-1)
      XLOC(2)=XDISP(2*N1)
      XLOC(3)=XDISP(2*N2-1)
      XLOC(4)=XDISP(2*N2)
      XLOC(5)=XDISP(2*N3-1)
      XLOC(6)=XDISP(2*N3)
      CALL MATMAT(4,4,6,EM,G,PROD1)
      CALL MATMAT(4,6,6,PROD1,H,PROD2)
      CALL MATVEC(4,6,PROD2,XLOC,S,4,6)
C------------------------------------------------------------------------------
C  CALCULATE THE PRINCIPAL STRESSES
C------------------------------------------------------------------------------
      PI=DACOS(-1.D0)
      S1=(S(1) + S(3))/2.D0
      S2=(S(1) - S(3))/2.D0
      S3=DSQRT(S2*S2 + S(4)*S(4))
      SMAX=S1 + S3
      SMIN=S1 - S3
      IF(S2 .EQ. 0.0) THEN
        ANGLE=PI/2.D0
      ELSE
        ANGLE=(DATAN(S(4)/S2))/2.D0
      END IF
C------------------------------------------------------------------------------
C  OUTPUT ELEMENT GLOBAL AND PRINCIPAL STRESSES
C------------------------------------------------------------------------------
      WRITE(2,100) ELID
  100 FORMAT('- ELEMENT NO. : ',I3,' TRIAX')
      WRITE(2,120) S(1),SMAX
  120 FORMAT('- S R',T10,': ',E10.3,T30,'S MAX',T37,': ',E10.3)
      WRITE(2,140) S(2),SMIN
  140 FORMAT('- S TH',T10,': ',E10.3,T30,'S MIN',T37,': ',E10.3)
      WRITE(2,160) S(3),ANGLE
  160 FORMAT('- S Z',T10,': ',E10.3,T30,'ANGLE',T37,': ',E10.3)
      WRITE(2,180) S(4)
  180 FORMAT('- T RZ',T10,': ',E10.3)
C
      RETURN
      END
C
      SUBROUTINE STRAX(ELID,N1,N2,N3,N4,N5,X1,Y1,X2,Y2,X3,Y3,X4,Y4,X5,
     +                 Y5,E,NU,XDISP,MEQNS,NTYPE)
C------------------------------------------------------------------------------
C  SUBROUTINE FOR CALCULATION OF THE LOCAL STRESSES OF QUAD4 AND QUAD5
C  ELEMENTS FOR AXISYMMETRIC ANALYSIS
C------------------------------------------------------------------------------
      IMPLICIT REAL*8 (A-H,O-Z)
```

```
      INTEGER ELID
      DOUBLE PRECISION NU
      DIMENSION XDISP(MEQNS),XI(6),ET(6),EM(4,4),NOD(6),XX(5),YY(5)
      DIMENSION XLOC(10),B(4,10),PROD(4,10),S(4),TRAN(4,4),SLOC(4)
      CHARACTER*5 TYPE
C
      DATA XI/-1.,1.,1.,-1.,0.,0./
C
      ET(1)=-1.
      ET(2)=-1.
      ET(3)=1.
      ET(4)=1.
      ET(5)=-1.
      ET(6)=0.
      IF(NTYPE .EQ. 4) THEN
        ET(5)=0.
        TYPE='QUAD4'
      ELSE
        TYPE='QUAD5'
      END IF
      WRITE(2,20) ELID,TYPE
   20 FORMAT('- ELEMENT NO. :',I3,2X,A5)
      MSIZE=2*NTYPE
      NSIZE=NTYPE+1
      CALL YOUNG(EM,E,NU,4,0)
C
      NOD(1)=N1
      NOD(2)=N2
      NOD(3)=N3
      NOD(4)=N4
      NOD(5)=N5
      NOD(6)=0
      XX(1)=X1
      XX(2)=X2
      XX(3)=X3
      XX(4)=X4
      XX(5)=X5
      YY(1)=Y1
      YY(2)=Y2
      YY(3)=Y3
      YY(4)=Y4
      YY(5)=Y5
C
      DO 40 I=1,NTYPE
        XLOC(2*I-1)=XDISP(2*NOD(I)-1)
        XLOC(2*I)=XDISP(2*NOD(I))
C
   40 CONTINUE
      WRITE(2,50)
   50 FORMAT('-',T2,'NODE',T12,'S  R',T24,'S TH',T36,'S  Z',T48,'T ZR',
     +        T60,'S MAX',T72,'S MIN',T84,'ANGLE')
C
      DO 80 I=1,NSIZE
        CALL SHAPAX(XI(I),ET(I),XX,YY,DETJAC,B,NTYPE)
        CALL MATMAT(4,4,MSIZE,EM,B,PROD)
        CALL MATVEC(4,MSIZE,PROD,XLOC,S,4,MSIZE)
C
        IF(NTYPE .EQ.5) THEN
          IF(I .EQ. 5) THEN
            CALL TRANSF(ELID,X1,X2,Y1,Y2,TRAN,4,4)
            CALL MATVEC(4,4,TRAN,S,SLOC,4,4)
            S(1)=SLOC(1)
```

```
          S(2)=SLOC(2)
          S(3)=SLOC(3)
          S(4)=SLOC(4)
        END IF
      END IF
      CALL PRINAX(NOD(I),S)
   80 CONTINUE
      WRITE(2,100)
  100 FORMAT('-')
C
      RETURN
      END
C
      SUBROUTINE PRINAX(NODE,S)
C--------------------------------------------------------------------------------
C  SUBROUTINE FOR CALCULATION OF THE PRINCIPAL STRESSES
C--------------------------------------------------------------------------------
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER NODE
      DIMENSION S(4)
C
      PI=DACOS(-1.D0)
      S1=(S(1)+S(3))/2.
      S2=(S(1)-S(3))/2.
      S3=DSQRT(S2*S2 + S(4)*S(4))
      SMAX=S1+S3
      SMIN=S1-S3
      IF(S2 .EQ. 0.D0) THEN
        ANGLE=PI/2.
      ELSE
        ANGLE=(DATAN(S(4)/S2))/2.
      END IF
      ANGLE=(ANGLE*180.D0)/PI
C
      WRITE(2,40) NODE,S(1),S(2),S(3),S(4),SMAX,SMIN,ANGLE
   40 FORMAT('-',I3,T9,E10.3,T21,E10.3,T33,E10.3,T45,E10.3,T57,E10.3,
     +        T69,E10.3,T81,E10.3)
C
      RETURN
      END
//GO.SYSIN DD *
//GO.FT01F001 DD *
/INCLUDE FI.IN
//GO.FT02F001 DD SYSOUT=*
//
```