8613131

**Aminpour, Mohammad Ali**

FINITE ELEMENT ANALYSIS OF PROPAGATING INTERFACE CRACKS IN COMPOSITES

*University of Washington* PH.D. 1986

University
Microfilms
International 300 N. Zeeb Road, Ann Arbor, MI 48106

# FINITE ELEMENT ANALYSIS OF PROPAGATING INTERFACE CRACKS IN COMPOSITES

by

Mohammad Ali Aminpour

A dissertation submitted in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy

University of Washington

1986

Approved by Keith A. Holsapple
(Chairperson of Supervisory Committee)

Program Authorized
to Offer Degree Department of Aeronautics and Astronautics

Date 3/10/86

University of Washington

Abstract

# FINITE ELEMENT ANALYSIS OF PROPAGATING INTERFACE CRACKS IN COMPOSITES

by Mohammad Ali Aminpour

Chairperson of the Supervisory Committee: Prof. Keith A. Holsapple
Department of Aeronautics and Astronautics

A complex variable formulation has been developed to describe the near-field state of stresses and displacements for a propagating crack along the interface of two dissimilar anisotropic materials. It is shown that the formulation is general and can be reduced to all the other subordinate cases without any difficulty. The crack can be propagating or stationary and each of the materials on the sides of the crack can be anisotropic, orthotropic or isotropic. The near-field stresses contain the regular square root singularity and the oscilatory behavior in case of dissimilar materials on the sides of the crack.

Due to the complexity of the problem it was not possible to use the conventional definitions of the stress intensity factors. Therefore a new definition for the stress intensity factors is proposed. It is proportional to the coefficient of the lowest order term of the near-field state of stresses and reduces to all the subordinate cases to within a multiplicative factor.

A detailed description of the development of the near-field state of stresses and displacements is presented. A finite element procedure has been developed to provide solution. The finite element

procedure utilizes a singular element which gives the direct solution of the time-dependent stress intensity factors. The procedure for the finite element formulation including a detailed description of the development of the singular element is presented. The element matrices are derived from a variational principle involving a modified functional for elastodynamic problems.

The resulting discretized dynamic equations of motion are solved by an implicit method of temporal integration using Nemark-$\beta$ formulas. Local asymmetries in the matrices which arise due to crack propagation are dealt with by modifying the finite difference formulation and by the use of an iterative procedure for convergence of the solution. Crack propagation is accomplished by moving the crack-tip inside the singular element according to a prescribed crack-tip position history. A local redefinition of the finite element mesh is required when the crack-tip reaches an extreme position inside the singular element. When the local mesh redefinition takes place, an extra node is created. This is accomplished using a method of double noding technique.

The accuracy of the finite element formulation is evaluated by solving problems for which analytical and numerical solutions are available. The solutions are found to compare well with widely accepted solutions in the literature. The solutions to some problems involving propagating and stationary cracks at the interface of two dissimilar anistropic materials, for which there are no known solutions, are presented. Finally, recommendations are made for further development of the finite element procedure.

TABLE OF CONTENTS

Page

## LIST OF FIGURES

LIST OF TABLES

## NOMENCLATURE

Subscript k corresponds to $k^{th}$ material

Subscript $\ell$ corresponds to $\ell^{th}$ eigenvalue

I. Roman symbols

| | |
|---|---|
| $a$ | Half crack length |
| $\underset{\sim}{a}, \underset{\sim}{a}_k$ | Matrices of elastic constants |
| $a_i$ | Elements of matrix $\underset{\sim}{a}$ |
| $a_{ij}$ | Elastic constants for a state of plane stress |
| $a'_{ij}$ | $a_{ij}$ in a rotated coordinate system |
| $\underset{\sim}{A}$ | $=\underset{\sim}{a}^{-1}$, matrix of elastic constatns |
| $A_i$, $A_{ij}$ | Elements of matrix $\underset{\sim}{A}$ |
| $A_{1k}, A_{2k}$ | Real parameters (Appendix A) |
| $A_2, A_3$ | Complex parameters (Chapter 6) |
| $b$ | A parameter defined by material properties and crack-tip speed (Chapter 3 and Appendix A) |
| $b$ | Half plate width (Chapter 9) |
| $b_{ij}$ | Elements of matrix $\underset{\sim}{a}$ (Chapter 3) |
| $\underset{\sim}{b}$ | Body force vector |
| $b_i$ | Components of body force vector $\underset{\sim}{b}$ |
| $\underset{\sim}{B}, \underset{\sim}{B}_k$ | Matrices relating stresses to $\phi$ (Chapter 3 and Appendix A) |
| $\underset{\sim}{\beta}$ | Matrix relating $\underset{\sim}{\beta}$ to $\underset{\sim}{q}$ (Chapter 5) |
| $c_{ijmn}$ | Elements of compliance tensor |
| $c(t)$ | Crack-tip speed as a function of time |

NOMENCLATURE

(Continued)

| | |
|---|---|
| $C_i$ | Wave amplitudes (Chapter 8) |
| $C_1$-$C_8$ | Complex parameters (Appendix A) |
| $C_{ik},(C_{ik})_\ell$ | Complex coefficients of complex power functions $\Omega_{ik}$ |
| $C_T,C_L$ | Transverse and longitudinal wave speeds of an isotropic material, repsectively |
| cm | Centimeter, unit of length |
| $d_0$ | Determinant of matrix $\underset{\sim}{a}$ |
| $\underset{\sim}{d}$ | Matrix of differential operators |
| $d_{mk\ell}$ | Parameters defining $D_{mk\ell}$ |
| dv | Differential volume element |
| ds | Differential surface element |
| dt | Differential time element |
| $dz,dz_k,dz_{ik}$ | Differential complex plane element |
| $D_{ik},(D_{ik})_\ell$ | Complex coefficients of complex power functions $\Omega_{ik}$ |
| $D_{mk\ell}$ | Functions defining near-field displacements |
| $\underset{\sim}{DIS}_k$ | Matrix |
| $\underset{\sim}{DIS}_{ik}$ | $i^{th}$ row of matrix $\underset{\sim}{DIS}_k$ |
| $(DIS_{ij})_k$ | Elements of matrix $\underset{\sim}{DIS}_k$ |
| dyne | Dyne, unit of force |
| e | Base of natural logrithm |
| $e_i,e_{ij}$ | Strain components |
| $E_1,E_2,E_3$ | Moduli of elasticity |

NOMENCLATURE

(Continued)

| | |
|---|---|
| $E_{ik}$ | $= C_{ik} + D_{ik}$, for real eigenvalues |
| $\underset{\sim}{E}$ | Matrix of derivatives of $\underset{\sim}{U}$ |
| $\underset{\sim}{EPS}_k$ | Matrix |
| $\underset{\sim}{EPS}_{ik}$ | $i^{th}$ row of matrix $\underset{\sim}{EPS}_k$ |
| $(EPS_{ij})_k$ | Elements of matrix $\underset{\sim}{EPS}_k$ |
| f | Parameter |
| f(t) | Crack-tip position as a function of time |
| $F_{ik\ell}$ | Complex parameters defning $C_{ik}$ and $D_{ik}$ |
| $\underset{\sim}{F}$ | Body force integral |
| $\underset{\sim}{F}_R$ | Regular element force vector |
| $\underset{\sim}{F}_s$ | Singular element force vector |
| $\underset{\sim}{\bar{F}}_R$ | Global form of $\underset{\sim}{F}_R$ |
| $\underset{\sim}{\bar{F}}_s$ | Global form of $\underset{\sim}{F}_s$ |
| g | Parameter (Chapter 8) |
| g | Gram, unit of mass |
| G | Lame's constant |
| $G_{12}, G_{13}, G_{23}$ | Elastic shear moduli |
| $\underset{\sim}{G}$ | Boundary integral |
| $H_{ik}$ | Complex parameters |
| $\underset{\sim}{H}$ | Volume integral |

NOMENCLATURE

(Continued)

| | |
|---|---|
| $\underset{\sim}{H}_1$ | Boundary integral |
| $i$ | $\sqrt{-1}$ |
| in. | Inch, unit of length |
| $k$ | Parameter |
| $k_1, k_2$ | Stress intensity factors |
| $K, K_1, K_2$ | Stress intensity factors |
| $K_s$ | $\sigma\sqrt{\pi a}$, static stress intensity factor for an infinite uniform medium |
| $K_{ik}, KK_{ik}$ | Complex parameters |
| $\underset{\sim}{K}$ | Global stiffness matrix |
| $\underset{\sim}{K}_1$ | Matrix |
| $\underset{\sim}{\overline{K}}_1$ | Global form of $\underset{\sim}{K}_1$ |
| $\underset{\sim}{K}_s$ | Singular element stiffness matrix |
| $\underset{\sim}{K}_R$ | Regular element stiffness matrix |
| $\underset{\sim}{\overline{K}}_s$ | Global form of $\underset{\sim}{K}_s$ |
| $\underset{\sim}{\overline{K}}_R$ | Global form of $\underset{\sim}{K}_R$ |
| $\underset{\sim}{K}_{eq}, \underset{\sim}{K}_{eff}, \underset{\sim}{K}'_{eff}$ | Stiffness matrices |
| $\underset{\sim}{K}_{sym}$ | Symmetric part of $\underset{\sim}{K}$ |
| $\underset{\sim}{K}_{asym}$ | Asymmetric part of $\underset{\sim}{K}$ |
| Kg | Kilogram, unit of mass |
| $L$ | Half height of plate |

NOMENCLATURE

(Continued)

| | |
|---|---|
| $\underset{\sim}{L}$ | Matrix of interpolating functions |
| $L_{ij}$ | Elements of matrix $\underset{\sim}{L}$ |
| m | Parameter (Chapter 8) |
| m | Meter, unit of length |
| $\underset{\sim}{M}$ | Global mass matrix |
| $\underset{\sim}{M}_1, \underset{\sim}{M}_2, \underset{\sim}{M}_3, \underset{\sim}{M}_4, \underset{\sim}{M}_5, \underset{\sim}{M}_6$ | Matrices of volume and boundary integrals |
| $\underset{\sim}{M}_1^*$ | Matrix |
| $\underset{\sim}{\overline{M}}_1^*$ | Global form of $\underset{\sim}{M}_1^*$ |
| $\underset{\sim}{M}_S$ | Singular element mass matrix |
| $\underset{\sim}{M}_R$ | Regular element mass matrix |
| $\underset{\sim}{\overline{M}}_S$ | Global form of $\underset{\sim}{M}_S$ |
| $\underset{\sim}{\overline{M}}_R$ | Global form of $\underset{\sim}{M}_R$ |
| N | Newton, unit of force |
| $\underset{\sim}{N}$ | Matrix of bilinear functions |
| $N_i$ | Bilinear functions |
| $n, n_\ell$ | Complex exponent |
| $n_\alpha, n_\beta$ | Number of unknown coefficients $\underset{\sim}{\alpha}$ and $\underset{\sim}{\beta}$, respectively |
| $\underset{\sim}{n}$ | Matrix of unit normal vector components |
| $n_i, n_x, n_y$ | Components of unit normal vector |

NOMENCLATURE

(Continued)

| | |
|---|---|
| $p$ | Number of elements in the finite element mesh (Chapter 5) |
| $p_i$ | Real parameters |
| $\underset{\sim}{P}$ | Matrix relating strains to stress function $\phi$ (Chapter 3) |
| $P_{ij}$ | Elements of $\underset{\sim}{P}$ |
| psi | Pounds per square inch, unit of pressure |
| $q$ | A parameter defining $\varepsilon$ (Chapter 6) |
| $\underset{\sim}{q}(t), \underset{\sim}{q}^*, \underset{\sim}{q}, \underset{\sim}{q}'$ | Vectors of nodal displacements |
| $q_i, q_i'$ | Elements of $\underset{\sim}{q}$ and $\underset{\sim}{q}'$, respectively |
| $Q$ | $\rho v^2$ |
| $Q_o$ | Parameter |
| $Q_{1k}, Q_{2k}, Q_k$ | Matrices |
| $\underset{\sim}{Q}_{eq}, \underset{\sim}{Q}_{eff}, \underset{\sim}{Q}'_{eff}$ | Force matrices |
| $r, \theta, r_k, \theta_k, r_{jk}, \theta_{jk}$ | Polar coordinate variables |
| $R$ | $\rho c^2(t)$ |
| $R$ | Region occupied by cracked body |
| $R_o$ | Parameter |
| $R_n$ | Subregion of R |
| $\underset{\sim}{R}$ | Matrix of boundary tranction functions |
| $\underset{\sim}{R}_1, \underset{\sim}{R}_2$ | Submatrices of $\underset{\sim}{R}$ |
| $R_{ij}$ | Elements of $\underset{\sim}{R}_1$ |

NOMENCLATURE

(Continued)

| | |
|---|---|
| $s$ | Parameter (Chapter 8) |
| $s$ | Parameter along subboundary (Chapter 4) |
| $s_{nk\ell}$ | Parameters defining $S_{nk\ell}$ |
| $S_n$ | Boundary of subregion $R_n$ |
| $S_{in}$ | Interior interface between elements |
| $S_t$ | Part of boundary where tractions are prescribed |
| $S_u$ | Part of boundary where displacements are prescribed |
| $S_{nk\ell}$ | Functions defining near-field stresses |
| $\underset{\sim}{S}$ | Matrix of stress functions |
| $\underset{\sim}{S}_1, \underset{\sim}{S}_2$ | Submatrices of $\underset{\sim}{S}$ |
| $\underset{\sim}{SIG}_k$ | Matrix |
| $\underset{\sim}{SIG}_{ik}$ | $i^{th}$ row of matrix $\underset{\sim}{SIG}_k$ |
| $(SIG_{ij})_k$ | Elements of matrix $\underset{\sim}{SIG}_k$ |
| $t$ | Time |
| $\underset{\sim}{t}$ | Approximating functions for boundary tractions |
| $t_i$ | Components of vector $\underset{\sim}{t}$ |
| $\underset{\sim}{t}^o$ | Prescribed surface tractions |
| $t_i^o$ | Components of $\underset{\sim}{t}^o$ |
| $T$ | Superscript denoting transpose of a matrix |
| $T_1, T_2$ | Constant tractions in X and Y directions, respectively |

NOMENCLATURE

(Continued)

| | |
|---|---|
| $\underset{\sim}{T}$ | Boundary force integral (Chapter 5) |
| $\underset{\sim}{T}$ | Transformation matrix for eliminating double nodes (Chapter 7) |
| $T_{mn}$ | Elements of matrix $\underset{\sim}{T}$ |
| $\underset{\sim}{u}$ | Matrix of displacement components |
| $u_i, u_{ik}$ | Displacements |
| $\underset{\sim}{u}^o$ | Prescribed boundary displacements |
| $u_i^o$ | Elements of $\underset{\sim}{u}^o$ |
| $u_{mk\ell}$ | Displacement eigenfunctions |
| $\underset{\sim}{U}$ | Matrix of displacement functions |
| $\underset{\sim}{U}_1, \underset{\sim}{U}_2$ | Submatrices of $\underset{\sim}{U}$ |
| $U_{ij}$ | Elements of $\underset{\sim}{U}_1$ |
| $v, v_1, v_2$ | Wave propagation speeds (Chapter 8) |
| $\underset{\sim}{v}$ | Boundary displacements |
| $v_i$ | Components of matrix $\underset{\sim}{v}$ |
| $\underset{\sim}{v}^o$ | Prescribed boundary displacements |
| $v_i^o$ | Components of $\underset{\sim}{v}^o$ |
| $V_1, V_2$ | Functions (Chapter 3) |
| $\overline{V}$ | Strain energy density function |
| $\underset{\sim}{V}$ | "Damping" matrix |
| $\underset{\sim}{V}_1$ | Matrix |

NOMENCLATURE

(Continued)

| | |
|---|---|
| $\underset{\sim}{\overline{V}}_1$ | Global form of $\underset{\sim}{V}_1$ |
| $\underset{\sim}{V}_s$ | Singular element "damping" matrix |
| $\underset{\sim}{\overline{V}}_s$ | Global form of $\underset{\sim}{V}_s$ |
| $\underset{\sim}{V}_{sym}$ | Symmetric part of $\underset{\sim}{V}$ |
| $\underset{\sim}{V}_{asym}$ | Asymmetric part of $\underset{\sim}{V}$ |
| $X_i$,X-Y | Global rectilinear coordinate system fixed to the body |
| $x_i$,x-y | Local rectilinear coordinate system moving with crack-tip |
| X′,Y′ | Rotated rectilinear coordinate system |
| $z, z_k, z_{ik}$ | Coordinates of complex plane |

II. Greek symbols

| | |
|---|---|
| $\alpha, \alpha_k$ | Complex paramters |
| $\underset{\sim}{\alpha}$ | Column matrix of unknown coefficients |
| $\underset{\sim}{\alpha}_1, \underset{\sim}{\alpha}_2$ | Submatrices of $\underset{\sim}{\alpha}$ |
| $\alpha_{ij}$ | Elements of $\underset{\sim}{\alpha}$ |
| $\underset{\sim}{\beta}$ | Column matrix of unknown coefficients |
| $\underset{\sim}{\beta}_1, \underset{\sim}{\beta}_2$ | Submatrices of $\underset{\sim}{\beta}$ |
| $\beta_{ij}$ | Elements of $\underset{\sim}{\beta}$ |
| $\beta_{ij}$ | Elastic constants for a state of plane strain (Chpater 2) |
| $\beta_\ell$ | Unknown complex coefficients |

## NOMENCLATURE

(Continued)

| | |
|---|---|
| $(\beta_1)_\ell, (\beta_2)_\ell$ | Real and imaginary parts of $\beta_\ell$ |
| $\gamma_{ij}$ | Shear strain |
| $\Gamma, \Gamma_o$ | Functions (Chapter 3) |
| $\Delta t$ | Time increment |
| $\varepsilon$ | Bielastic parameter |
| $\underset{\sim}{\varepsilon}, \underset{\sim}{\varepsilon}_k$ | Matrices of strains |
| $\varepsilon_{ij}, (\varepsilon_{ij})_k$ | Components of strain tensor |
| $\zeta, \eta$ | Local rectilinear coordinate system for transformation |
| $\eta_k$ | A parameter defined by $\nu_k$ |
| $\theta, r, \theta_k, r_k, \theta_{kj}, r_{kj}$ | Polar coordinate variables |
| $\theta$ | Angle from $X_1$ axis (Chapter 8) |
| $\theta$ | Angle from Z axis (Chapter 9) |
| $\lambda$ | Lame's constant |
| $\mu_k$ | Shear moduli of elasticity (Chapter 6) |
| $\mu, \mu_i, \mu_k, \mu_{ik}$ | Complex paramters |
| $\nu, \nu_k, \nu_{12}, \nu_{13}, \nu_{23}$ | Poisson's ratios |
| $\Pi$ | Functional |
| $\Pi_S$ | Functional for singular element |
| $\Pi_R$ | Functional for regular element |
| $\rho$ | Mass density |

NOMENCLATURE

(Continued)

| | |
|---|---|
| $\underset{\sim}{\sigma}, \underset{\sim}{\sigma}_k$ | Matrices of stresses |
| $\sigma_{ij}, (\sigma_{ij})_\ell$ | Components of stress tensor |
| $\sigma, \sigma_{yy}, \sigma_{xx}, \sigma_{xx_1}, \sigma_{xx_2}$ | Uniformly applied stresses |
| $\sigma_{nk\ell}$ | Stress eigenfunctions |
| $\tau_i$ | Components of matrix of streses |
| $T, T_1, T_2$ | Complex analytic functions |
| $\phi$ | Angle of rotation (Chapter 2) |
| $\phi_o, \phi, \phi_k$ | Stress functions |
| $\underset{\sim}{\Phi}_\sigma, \underset{\sim}{\Phi}_{\sigma k}$ | Matrices of 2nd derivatives of $\phi$ |
| $\psi\ \psi_1, \psi_2$ | Functions (Chapter 3) |
| $\omega$ | Real circular frequency |
| $\Omega_i, \Omega_{ik}$ | Complex analytic functions |
| $\underset{\sim}{\Omega}_\sigma, \underset{\sim}{\Omega}_{\sigma k}, \underset{\sim}{\Omega}_u, \underset{\sim}{\Omega}_{uk}$ | Matrices of derivatives of $\Omega_i$ and $\Omega_{ik}$ |

## ACKNOWLEDGEMENTS

The author wishes to express his sincere gratitude to his advisor, Professor Keith A. Holsapple for his assistance and understanding during this research and in the preparation of this manuscript. The author also wishes to thank Professors R.R. Parmerter, K.Y. Lin, I.M. Fyfe, Z.M. Elias and R. Taggart for serving on the advisory committee, Professor A.S. Kobayashi for his helpful discussions, and Ms. Sandy Horton for typing this dissertation.

Finally, special appreciation is given to my wife, Zohreh, for her constant support, encouragement and forbearance. I must also mention my son, Arash, whom I neglected and did not spend nearly enough time with in the course of this research.

The computing funds and other expenses for this research were provided by the Department of Aeronautics and Astronautics, supplemented by a grant from the Graduate School.

# CHAPTER I

# INTRODUCTION

In recent years there has been much interest in employing anisotropic materials in industry, because of their desirable properties. In contrast to isotropic materials, for which the material properties are independent of directions, anisotropic material properties are directional. For example, materials such as plywood have large differences in their elastic moduli in the principal directions. Composite materials, which are usually formed from several layers of different orientation of the same material bonded together, are being used extensively in the aerospace industry. The problem of debonding or delamination of the layers of composites under various types of loads is of major concern. The study of this problem requires a description of the stress and displacement fields in the vicinity of the crack-tip in the delamination process, at the interface of two dissimilar anisotropic materials.

To the author's knowledge, there has been no attempt to formulate the problem of a propagating interface crack between two dissimilar anisotropic materials. In this dissertation this problem is formulated, and the complete state of stresses, displacements and the stress intensity factors are found. The solution technique employs a hybrid-displacement finite element formulation. In addition, a new and more general definition for stress intensity factors is proposed. It can be reduced, to within a multiplicative factor, to all other subordinate cases.

M.L. Williams [1]* was among the first to derive the form of the stress singularity for interface cracks between two dissimilar isotropic materials for static problems. Later, F. Erdogan [2] and G.C. Sih and J.R. Rice [3-4] successfully formulated the stress state near the crack-tip and derived a formula for the stress intensity factors for the problem considered by Williams [1]. The stress intensity factors are of special interest in fracture mechanics, since they are parameters governing the onset of rapid crack extension and they characterize the near-field stresses and displacements.

K.Y. Lin and J.W. Mar [5] derived the complete expressions for the eigenfunctions of a stationary interface crack between two dissimilar isotropic materials under static loads. Also, employing a finite element formulation, they successfully solved for the state of stresses and displacements and the stress intensity factors (using Sih's definition) for a variety of material combinations and loadings.

P.S. Theocaris [6] and others [21] have questioned the form of the eigenfunctions used by [1-5], in which the exponents "n" of the eigenfunctions are implicitly assumed to be real, since when they are solved for as eigen-values, some of the n's turn out to be complex numbers. Theocaris [6] proposes the use of a more complete form of eigenfunctions involving both n and $\overline{n}$ . (The bar indicates the

*Numbers in brackets indicate references.

complex conjugate). However, the present author has verified that when all the analysis is carried out and the expressions for stresses and displacements are derived, for this particular problem of a stationary interface crack between two dissimilar isotropic materials under static loads, both methods result in the same final expressions. However, there is no reason to believe that this will be the case for the more general problem of an interface crack (stationary or running) between two dissimilar materials (each can be anisotropic, orthotropic or isotropic) under static or dynamic loads, as considered in this dissertation.

It would be a cumbersome and time consuming task to prove or disprove that the two methods will have the same results for the problem considered here. However, after some analysis it is seen that both methods will involve the same number of unknowns in the final equations. In the method used by [1-5], four unknown complex coefficients of the eigenfunctions and their complex conjugates, i.e. a total of eight unknown complex coefficients, will be involved, while in the method proposed by Theocaris, of the eight unknown complex coefficients of the eigenfunctions, either the complex coefficient itself or its complex conjugate will be involved, resulting also in eight unknown complex coefficients. Therefore, the amount of analysis and calculations will be the same for both methods. Here the more complete form of the eigenfunctions proposed by Theocaris will be used, with no further verification of the equivalency of the two methods.

D.B. Bogy [7-8] and also D.N. Fenner [9] calculated the eigenvalues of the exponent n for a more general problem of a stationary crack terminating at the interface of two dissimilar isotropic materials at an angle under static loads, a particular case of which would be the problem considered by [1-6]. Fenner also calculated the near-field stress distribution for that problem.

Recently An-yu Kuo has solved the problem of stationary interface crack in infinite media composed of two dissimilar orthotropic [26] and anisotropic [27] materials under impact loading on the crack surfaces. Singular integral equations and Jacobi polynomials were used to obtain the solution. Furthermore An-yu Kuo and Su-Su Wang [10] have reported the problem of a stationary interface crack between two dissimilar anisotropic materials under dynamic loads by employing a hybrid-stress finite element formulation. This author however, has no knowledge of their results having been published to date.

J. Aboudi [11] has solved for the near-field stresses for a moving interface crack between two dissimilar isotropic materials. An implicit three-level (in time and space) numerical method was used to solve the equations of motion. An arbitrary definition for the stress intensity factors as the ratio of the near-field stresses to a reference stress and a reference dimension was given. This definition of the stress intensity factors is different from that of Sih's derivation. In Aboudi's definition the mode I (the opening mode) stress intensity factor was given in terms of only the normal stresses, while in Sih's derivation, both the mode I and the mode II (the sliding mode) stress intensity factors are functions of both normal and shear stresses.

S.N. Atluri and T. Nishioka [14-15] and C.K. Gunther and K.A. Holsapple [16] have employed a hybrid-displacement finite element formulation for running cracks in a single isotropic material bodies. The finite element formulation in this dissertation is structured after the Gunther and Holsapple [16] formulation. Many modifications and extra subroutines were implemented in the computer program described in [16] to make the program more efficient, and to generalize to the problem of this research.

In the finite element mesh used here, the crack-tip is embedded in a "singular element", and all other elements away from the crack-tip and arround the singular element are regular elements. Different approximating functions for the singular and regular elements are considered in Chapters 3 and 4.

In the literature it is customary to formulate the singular element using two displacement potentials for dynamic problems. However, we will use a totally different and new approach here and formulate the singular element using stress functions. The stress functions developed here are new and resemble the complex variable formulation presented by N.I. Muskhelishvili [12] and used by S.G. Lekhnitskii [13] for anisotropic materials to formulate static problems. Complex power-series eigenfunctions are assumed for Muskhelishvili's functions.

If the crack-tip is not stationary, it will run inside the singular element until it reaches an extreme position, at which time a local remeshing takes place and the position of the singular element is moved forward in the direction of crack propagation. Thus the crack can continue to propagate inside the singular element.

In the process of remeshing as the crack-tip runs, new nodes have to be created in the finite element mesh. A double noding technique proposed by B.M. Liaw, A.S. Kobayashi and A.F. Emery [17] and Santosh K. Arya [18] is employed for the creation of new nodes. In this technique the mesh-points along the crack-line , i.e. the mesh-points on the interface have two nodes, so that when the crack-tip passes through a mesh-point that point becomes two separate nodes. However, before the crack-tip reaches such points, one must to eliminate the extra node at such points by using the equality of displacements of the two nodes so that the final equivalent stiffness matrix is not singular.

In the following chapters a detailed derivation of the near-field functions are presented, and the implementation of these functions into a hybrid-displacement finite element is described. The solution to some problems are compared to the known theoretical and numerical results, and some examples of stationary and running interface cracks between two dissimilar anisotropic materials under static and dynamic loads are presented. A detailed description of the finite element program and the double noding technique are given. Finally a summary and recommendations for further research is presented.

## CHAPTER II

## THE THEORY OF ELASTICITY FOR AN ANISOTROPIC MATERIAL

### 1. The basic assumptions:

For the linearized theory of elasticity for small displacements, the deformations of an elastic body are given by

$$2\varepsilon_{ij}=u_{i,j}+u_{j,i} \qquad i,j=1,2,3 \qquad (2.1)$$

where $\varepsilon_{ij}$ are the elements of the strain tensor and $u_i$ are the components of the displacement vector with

$$u_{i,j}=\partial u_i/\partial X_j \qquad i,j=1,2,3 \qquad (2.2)$$

It is obvious from equations (2.1) that the strain tensor is symmetric, i.e.

$$\varepsilon_{ij}=\varepsilon_{ji} \qquad i,j=1,2,3 \qquad (2.3)$$

The dynamic equilibrium equations are taken to be Navier's equations of motion

$$\sigma_{ij,j}+\rho b_i=\rho\ddot{u}_i \qquad i,j=1,2,3 \qquad (2.4)$$

where $\sigma_{ij}$ are the elements of the stress tensor, $\rho$ is the mass density of the material, $b_i$ are the elements of body force per unit mass vector, and

$$\dot{u}_i=\partial u_i/\partial t$$
$$\ddot{u}_i=\partial^2 u_i/\partial t^2 \qquad i=1,2,3 \qquad (2.5)$$

represent the particle velocity and acceleration. The components of surface tractions $t_i$ are given by

$$t_i=\sigma_{ij}n_j \qquad i,j=1,2,3 \qquad (2.6)$$

where $n_j$ are the components of an outward unit vector normal to the surface. The stress tensor is taken to be symmetric, i.e.

$$\sigma_{ij}=\sigma_{ji} \qquad i,j=1,2,3 \qquad (2.7)$$

For linear elastic materials the constitututive relations are given by

$$\varepsilon_{ij}=c_{ijmn}\sigma_{mn} \qquad i,j,m,n=1,2,3 \qquad (2.8)$$

where $c_{ijmn}$ represent 81 elastic constants. From the considerations of equations (2.3) and (2.7) $c_{ijkl}$ will reduce to 36 independent constants. Define $e_{ij}$ as

$$e_{ij}=\begin{cases} \varepsilon_{ij} & \text{when } i=j \\ \gamma_{ij}=2\varepsilon_{ij} & \text{when } i\neq j \end{cases} \qquad i,j=1,2,3 \qquad (2.9)$$

The existance of a "strain energy density function" $\bar{V}$, is assumed so that

$$\sigma_{ij}=\partial\bar{V}/\partial e_{ij} \qquad i,j=1,2,3 \qquad (2.10)$$

Then the number of independent elastic constants reduces to 21. Furthermore, let

$$[e_1\ e_2\ e_3\ e_4\ e_5\ e_6]=[\varepsilon_{11}\ \varepsilon_{22}\ \varepsilon_{33}\ 2\varepsilon_{23}\ 2\varepsilon_{31}\ 2\varepsilon_{12}]$$

and

$$[\tau_1\ \tau_2\ \tau_3\ \tau_4\ \tau_5\ \tau_6]=[\sigma_{11}\ \sigma_{22}\ \sigma_{33}\ \sigma_{23}\ \sigma_{31}\ \sigma_{12}] \qquad (2.11)$$

so that

$$e_i=a_{ij}\tau_j \qquad i,j=1,2,\ldots 6 \qquad (2.12)$$

with

$$a_{ij}=a_{ji} \qquad i,j=1,2,\ldots 6 \qquad (2.13)$$

Equation (2.12), "Generalized Hooke's Law", represent the constitutive relation for an anisotropic material in 3-dimensions which involves 21 independent elastic constants. If it is assumed that there exist three orthogonal planes of symmetry in the material, then the material is called an "orthotropic material", and the number of independent constants is reduced to 9. Furthermore, for an "isotropic material", where the material is symmetric in all directions, the number of independent elastic constants is reduced to 2. For a detailed treatment of these derivations and reductions in the number of elastic constants refer to [13].

2. The generalized state of plane stress

Consider an elastic uniform anisotropic flat plate of constant thickness, in equilibrium under applied loads along its edges. The following conditions (see S.G. Lekhnitskii [19]) define the generalized state of plane stress.

a) At each point of the plate there is a plane of elastic symmetry parallel to the middle surface.

b) The applied forces along the edges, and the body forces, act on planes parallel to the middle surface, are distributed symmetrically about that surface and suffer only slight changes through the thickness.

c) The strains are small.

Under these conditions, the middle surface remains flat under the applied loads, and equation (2.12) reduces to

$$e_i = a_{ij}\tau_j \qquad i,j=1,2,6 \qquad (2.14)$$

where $\tau_i$ and $e_i$ are now the average stresses and strains over the plate thickness [19]. Also $\tau_3=\sigma_{33}$ will be neglected in comparison with $\tau_1$, $\tau_2$, and $\tau_6$.

3. The state of plane strain

Consider an elastic uniform anisotropic body of cylindrical shape of arbitrary cross-section, in equilibrium under applied loads along its surface. The following conditions (see S.G. Lekhnitskii [19]) define the state of plane strain.

a) The forces act in planes normal to the cylinder and do not vary along it (its length is supposed large by comparision with its cross-sectional dimensions).

b) At each point there is a plane of elastic symmetry perpendicular to the cylinder.

c) The strains are small.

For plane sections far from the ends it is assumed that

$$u_1 = u_1(X,Y),\ u_2 = u_2(X,Y),\ \text{and}\ u_3 = 0 \qquad (2.15)$$

where X-Y is a cross-sectional plane and the Z axis is considered to be parallel to the cylinder. Under these conditions, equations (2.12) reduce to

$$e_i = \beta_{ij}\tau_j \qquad i,j = 1,2,6 \qquad (2.16)$$

where

$$\beta_{ij} = a_{ij}-a_{i3}a_{j3}/a_{33} \qquad i,j = 1,2,6 \qquad (2.17)$$

4. Orthotropic and isotropic materials

For orthotropic materials, it is necessary that

$$a_{16} = a_{26} = 0 \quad \text{and} \quad \beta_{16} = \beta_{26} = 0 \tag{2.18}$$

For an isotropic material, it is necessary that

a) For plane stress

$$\begin{aligned} a_{11} &= a_{22} \\ a_{16} &= a_{26} = 0 \\ a_{66} &= 2(a_{11}-a_{12}) \end{aligned} \tag{2.19a}$$

b) For plane strain

$$\begin{aligned} \beta_{11} &= \beta_{22} \\ \beta_{16} &= \beta_{26} = 0 \\ \beta_{66} &= 2(\beta_{11}-\beta_{12}) \end{aligned} \tag{2.19b}$$

5. Transformation of elastic constants under a transformation of the coordinate system.

In some problems it is necessary to recalculate the elastic constants in a new X′-Y′ coordinate system from the elastic constants in a different X-Y coordinate system. This can be done by considering that the value of the strain energy density function remains the same for any coordinate system. Let $\phi$ be the angle through which the old coordinate system has been rotated to obtain the new coordinate system. Then, when the elastic strain energy is calculated for each coordinate system and equated term by term (see [13 and 19]) the following will result for a plane problem.

$$\begin{aligned} a'_{11} &= a_{11}\cos^4\phi+(2a_{12}+a_{66})\sin^2\phi\cos^2\phi+a_{22}\sin^4\phi+ \\ &\quad (a_{16}\cos^2\phi+a_{26}\sin^2\phi)\sin2\phi \\ a'_{22} &= a_{11}\sin^4\phi+(2a_{12}+a_{66})\sin^2\phi\cos^2\phi+a_{22}\cos^4\phi- \\ &\quad (a_{16}\sin^2\phi+a_{26}\cos^2\phi)\sin2\phi \end{aligned} \tag{2.20}$$

$$a'_{12} = (a_{11}+a_{22}-2a_{12}-a_{66})\sin^2\phi\cos^2\phi+a_{12}+1/2(a_{26}-a_{16})\sin2\phi\cos2\phi$$

$$a'_{66} = 4(a_{11}+a_{22}-2a_{12}-a_{66})\sin^2\phi\cos^2\phi+a_{66}+2(a_{26}-a_{16})\sin2\phi\cos2\phi$$

$$a'_{16} = (a_{22}\sin^2\phi-a_{11}\cos^2\phi+1/2(2a_{12}+a_{66})\cos2\phi)\sin2\phi+ a_{16}(\cos^2\phi-3\sin^2\phi)\cos^2\phi+a_{26}(3\cos^2\phi-\sin^2\phi)\sin^2\phi$$

$$a'_{26} = (a_{22}\cos^2\phi-a_{11}\sin^2\phi-1/2(2a_{12}+a_{66})\cos2\phi)\sin2\phi+ a_{16}(3\cos^2\phi-\sin^2\phi)\sin^2\phi+a_{26}(\cos^2\phi-3\sin^2\phi)\cos^2\phi$$

# CHAPTER III

## A FORMULATION FOR THE STRESSES AND DISPLACEMENTS OF A PROPAGATING CRACK ALONG THE INTERFACE OF TWO DISSIMILAR ANISOTROPIC MATERIALS

In this chapter a detailed derivation of the expressions for the approximating eigenfunctions which govern the near-field solution for a propagating crack at the interface of two dissimilar anisotropic materials is presented. This formulation is new. All previous formulations for interface cracks have been concerned with stationary cracks or for propagating cracks in homogeneous isotropic materials only. Also, in the literature for propagating cracks, traditionally two displacement potentials are used to formulate the problem. Here we derive a single stress function and formulate the near-field solution for propagating cracks using this stress function. Although this formulation is for the more general problem of a propagating interface crack between two dissimilar anisotropic materials, the formulation will reduce to all the subordinate cases, since all the possible solutions are accounted for. Therefore this is a single formulation in which each of the materials on the side of the interface can be isotropic, orthotropic or anisotropic, the crack can be stationary or propagating and the loading can be static or dynamic. In the previous formulations in the literature the formulation for an anisotropic material could not be reduced to that of an isotropic material and the formulation for a propagating crack could not be reduced to that of a stationary crack.

Let X, Y be a reference coordinate system with the X-axis parallel to the crack-line. Also let x, y be a moving coordinate system with its origin at the crack-tip and with the x-axis coinciding with the crack-line. Assuming Fig. 3.1 represents the situation at time $t = t_0$, then

$$\begin{aligned} y &= Y - Y_0 \\ x &= X - f(t) \end{aligned} \tag{3.1}$$

where f(t) represents the position of the crack-tip with

and

$$\begin{aligned} f(t_0) &= X_0 \\ \dot{f}(t) &= c(t) \end{aligned} \tag{3.2}$$

where c(t) is the crack-tip velocity as a function of time. Using the chain rule, one has

$$\begin{aligned} \frac{\partial \Gamma}{\partial Y} &= \frac{\partial \Gamma_0}{\partial y} \\ \frac{\partial \Gamma}{\partial X} &= \frac{\partial \Gamma_0}{\partial x} \\ \frac{\partial \Gamma}{\partial t} &= \frac{\partial \Gamma_0}{\partial x}\frac{\partial x}{\partial t} + \frac{\partial \Gamma_0}{\partial t} = -\frac{\partial \Gamma_0}{\partial x}\dot{f}(t) + \frac{\partial \Gamma_0}{\partial t} \\ \frac{\partial^2 \Gamma}{\partial t^2} &= \frac{\partial^2 \Gamma_0}{\partial x^2}\dot{f}^2(t) - \frac{\partial^2 \Gamma_0}{\partial x \partial t}\dot{f}(t) - \frac{\partial \Gamma_0}{\partial x}\ddot{f}(t) + \frac{\partial^2 \Gamma_0}{\partial t^2} \end{aligned} \tag{3.3}$$

where $\Gamma$ represents functions such as displacements or stresses and $\Gamma = \Gamma(X, Y, t) = \Gamma_0(x,y,t)$.

When an asymptotic expansion is made as $x \to 0$, only the first term in the expansion is taken as an approximation, so that the time derivatives in equations (3.3) simplify as (see Freund [20])

$$\begin{aligned} \frac{\partial \Gamma}{\partial t} &= -\frac{\partial \Gamma_0}{\partial x}\dot{f}(t) = -c(t)\frac{\partial \Gamma_0}{\partial x} \\ \frac{\partial^2 \Gamma}{\partial t^2} &= \frac{\partial^2 \Gamma_0}{\partial x^2}\dot{f}^2(t) = c^2(t)\frac{\partial^2 \Gamma_0}{\partial x^2} \end{aligned} \tag{3.4}$$

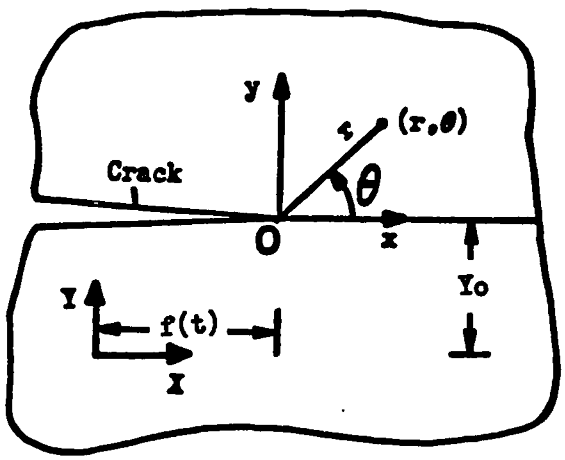



Fig. 3.1 Coordinate System Definition for a Cracked Body at Some Fixed Instant of Time

L. B. Freund [20] using asymptotic expansions and taking only the first term of the expansion as an approximation, shows that it is possible to solve problems of curved cracks with variable crack-tip velocities. Furthermore he shows that the resulting equations are the same as those for straight cracks with respect to a local coordinate system at the crack-tip with the x-axis being tangent to the path of the crack, and with the constant crack-tip speed replaced by the instantaneous crack-tip speed.

Using the equilibrium equations (2.4) and neglecting the body forces, one has

$$\sigma_{ij,j} = \rho \ddot{u}_i \tag{3.5}$$

with respect to the global coordinate system X-Y. Using equations (3.4) for the displacement field in equation (3.5) gives

$$\sigma_{ij,j} = \rho c^2(t) \frac{\partial^2 u_i}{\partial x^2} \tag{3.6}$$

with respect to the local coordinate system x-y. Expanding equation (3.6) for plane problems and letting $R = \rho c^2(t)$ for convenience, yields

$$\sigma_{11,1} + \sigma_{12,2} = R \frac{\partial^2 u_1}{\partial x^2} \tag{3.7a}$$

$$\sigma_{12,1} + \sigma_{22,2} = R \frac{\partial^2 u_2}{\partial x^2} \tag{3.7b}$$

These equations will be solved by introducing a new stress function which governs the near field solution for a propagating crack in an anisotropic material. From that the formulation for a propagating crack at the interface of two dissimiliar anisotropic materials is derived.

The right-side of equations (3.7a) and (3.7b) can be written as

$$\frac{\partial^2 u_1}{\partial x^2} = \frac{\partial}{\partial x}\left(\frac{\partial u_1}{\partial x}\right) = \frac{\partial}{\partial x}(\varepsilon_{11})$$

$$= \frac{\partial}{\partial x}(a_1\sigma_{11} + a_2\sigma_{22} + a_3\sigma_{12})$$

$$\frac{\partial^2 u_2}{\partial x^2} = \frac{\partial}{\partial x}\left(\frac{\partial u_2}{\partial x}\right) = \frac{\partial}{\partial x}\left(\gamma_{12} - \frac{\partial u_1}{\partial y}\right) \tag{3.8}$$

$$= \frac{\partial}{\partial x}(\gamma_{12}) - \frac{\partial}{\partial y}\left(\frac{\partial u_1}{\partial x}\right)$$

$$= \frac{\partial}{\partial x}(a_3\sigma_{11} + a_5\sigma_{22} + a_6\sigma_{12})$$

$$- \frac{\partial}{\partial y}(a_1\sigma_{11} + a_2\sigma_{22} + a_3\sigma_{12})$$

In the above, use has been made of equations (2.1), (2.14), and (2.16) with the following conventions for simplicity:

$$\underset{\sim}{a} = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_2 & a_4 & a_5 \\ a_3 & a_5 & a_6 \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} & b_{16} \\ b_{12} & b_{22} & b_{26} \\ b_{16} & b_{26} & b_{66} \end{bmatrix} \tag{3.9}$$

where $b_{ij} = a_{ij}$ defined by equation (2.14) for plane stress and $b_{ij} = \beta_{ij}$ defined by equation (2.16) for plane strain. In what follows and the rest of this dissertation we will not differentiate between plane stress and plane strain, because only $a_{ij}$ has to change to $\beta_{ij}$ to alternate from plane stress to plane strain.

Substituting equation (3.8) into equations (3.7) results in

$$\frac{\partial}{\partial x}[(1-a_1R)\sigma_{11} - a_2R\sigma_{22} - a_3R\sigma_{12}] + \frac{\partial}{\partial y}(\sigma_{12}) = 0 \tag{3.9a}$$

$$\frac{\partial}{\partial x}[-a_3R\sigma_{11} - a_5R\sigma_{22} + (1-a_6R)\sigma_{12}] + \frac{\partial}{\partial y}[a_1R\sigma_{11} + (1+a_2R)\sigma_{22} + a_3R\sigma_{12}] = 0 \tag{3.9b}$$

These equations have the general form of

$$\frac{\partial V_1}{\partial x} + \frac{\partial V_2}{\partial y} = 0 \tag{3.10}$$

for appropriate $V_1$ and $V_2$, and can therefore be derived from potential function $\psi(x,y)$ such that

$$V_1 = \frac{\partial \psi}{\partial y} \quad \text{and } V_2 = -\frac{\partial \psi}{\partial x} \tag{3.11}$$

so that equation (3.10) is identically satisfied. Using this idea in equation (3.9), let

$$(1-a_1R)\sigma_{11}-a_2R\sigma_{22}-a_3R\sigma_{12} = \frac{\partial \psi_1}{\partial y} \tag{3.12a}$$

$$\sigma_{12} = -\frac{\partial \psi_1}{\partial x} \tag{3.12b}$$

$$-a_3R\sigma_{11}-a_5R\sigma_{22}+(1-a_6R)\sigma_{12} = -\frac{\partial \psi_2}{\partial y} \tag{3.12c}$$

and

$$a_1R\sigma_{11}+(1+a_2R)\sigma_{22}+a_3R\sigma_{12} = \frac{\partial \psi_2}{\partial x} \tag{3.12d}$$

where $\psi_1$ and $\psi_2$ are two arbitrary potential functions. Using equation (3.12b) in equations (3.12a) and (3.12d) and solving for stresses in terms of the potential $\psi_1$ and $\psi_2$ yields

$$\begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{bmatrix} = \begin{bmatrix} -a_3R & 1+a_2R & a_2R \\ a_3R & -a_1R & 1-a_1R \\ -(1-(a_1-a_2)R) & 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{\partial \psi_1}{\partial x} \\ \frac{\partial \psi_1}{\partial y} \\ \frac{\partial \psi_2}{\partial x} \end{bmatrix} \frac{1}{1-(a_1-a_2)R} \tag{3.13}$$

When equation (3.13) and (3.12b) are substituted into (3.12c), the result is

$$\frac{\partial}{\partial x}\left[d_1\psi_1+d_3\psi_2\right]+\frac{\partial}{\partial y}\left[d_2\psi_1+d_4\psi_2\right] = 0 \qquad (3.14)$$

where

$$d_1 = a_3(a_3-a_5)R^2-(1-a_6R)(1-(a_1-a_2)R)$$

$$d_2 = -a_3R-(a_2a_3-a_1a_5)R^2$$

$$d_3 = -a_5R-(a_2a_3-a_1a_5)R^2$$

$$d_4 = 1-(a_1-a_2)R$$

Now, again let

$$d_1\psi_1+d_3\psi_2 = -\frac{\partial\phi_o}{\partial y}$$

$$\text{and} \quad d_2\psi_1+d_4\psi_2 = \frac{\partial\phi_o}{\partial x} \qquad (3.15)$$

Solving for potentials $\psi_1$ and $\psi_2$ in terms the potential $\phi_o$ gives

$$\begin{bmatrix}\psi_1\\ \psi_2\end{bmatrix} = \begin{bmatrix}-d_3 & -d_4\\ d_1 & d_2\end{bmatrix}\begin{bmatrix}\frac{\partial\phi_o}{\partial x}\\ \frac{\partial\phi_o}{\partial y}\end{bmatrix}\frac{1}{d_1d_4-d_2d_3} \qquad (3.16)$$

Substituting equations (3.16) into equations (3.13) yields

$$\underset{\sim}{\sigma} = \underset{\sim}{B}\underset{\sim}{\phi}_\sigma \qquad (3.17)$$

where

$$\underset{\sim}{\sigma} = [\sigma_{11}\ \sigma_{22}\ \sigma_{12}]^T$$

$$\underset{\sim}{\phi}_\sigma = [\phi_{,11}\ \phi_{,22}\ \phi_{,12}]^T$$

$$\text{and} \quad \underset{\sim}{B} = \begin{bmatrix} a_2R+(a_3a_5-a_2a_6)R^2 & 1+a_2R & -(a_3+a_5)R \\ 1-(a_1+a_6)R+(a_1a_6-a_3^2)R^2 & -a_1R & 2a_3R \\ a_5R+(a_2a_3-a_1a_5)R^2 & 0 & -(1-(a_1-a_2)R) \end{bmatrix} \quad (3.17a)$$

where $\phi = -\dfrac{1}{d_1d_4-d_2d_3}\phi_o$ has been rescaled for convenience.

Let the constitutive equations (2.12) and (2.13) be written as

$$\underset{\sim}{\varepsilon} = \underset{\sim}{a}\underset{\sim}{\sigma} = \underset{\sim}{p}\underset{\sim}{\phi}\sigma \quad (3.18)$$

where

$$\underset{\sim}{p} = \underset{\sim}{a}\underset{\sim}{B} \quad (3.19)$$

$$\text{and} \quad \underset{\sim}{\varepsilon} = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ 2\varepsilon_{12} \end{bmatrix}, \quad \underset{\sim}{\sigma} = \begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{bmatrix}, \quad \underset{\sim}{a} = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_2 & a_4 & a_5 \\ a_3 & a_5 & a_6 \end{bmatrix}$$

From equations (2.1) the usual compatibility equation in 2-D is written as

$$2\varepsilon_{12,12} = \gamma_{12,12} = \varepsilon_{11,22} + \varepsilon_{22,11} \quad (3.20)$$

substituting equation (3.18) into (3.20) yields

$$p_5\,\phi_{,2222} + p_4\,\phi_{,1222} + p_3\,\phi_{,1122} + p_2\,\phi_{,1112} + p_1\,\phi_{,1111} = 0 \quad (3.21)$$

where

$$p_5=p_{12}=a_1$$
$$p_4=p_{13}-p_{32}=-2a_3$$
$$p_3=p_{11}+p_{22}-p_{33}=2a_2+a_6-R[(a_1a_6-a_3^2)+(a_1a_4-a_2^2)]$$
$$p_2=p_{23}-p_{31}=-2\{a_5+R[(a_2a_5-a_3a_4)+(a_2a_3-a_1a_5)]\}$$
$$p_1=p_{21}=a_4-R[(a_4a_6-a_5^2)+(a_1a_4-a_2^2)]+R^2d_0$$

and where, $d_0 = |\underset{\sim}{a}|$, is the determinant of matrix $\underset{\sim}{a}$ and $p_{ij}$ are the elements of matrix $\underset{\sim}{p}$ as defined in equation (3.19).

When c(t), the crack tip velocity, vanishes, so that $R=\rho c^2(t)$ vanishes, equation (3.21) reduces to that of Lekhnitskii [19] for anisotropic materials. Furthermore when the material becomes isotropic, equation (3.21) reduces to the well known equation

$$\nabla^4\phi = \phi_{,2222} + 2\phi_{,2211} + \phi_{,1111} = 0$$

in which $\phi$ assumes the role of Airy stress function. Therefore the function $\phi$ in equation (3.21) can be considered as a new stress function which governs the near field solution for a propagating crack in an anisotropic material.

It is assumed that the function $\phi$ can be represented by an analytic complex function $T$ as

$$\phi = \text{Real } (T) \tag{3.22}$$

where $T = T(z)$

with $z = x + \mu y$

and where $\mu$ is a complex parameter to be determined. When equation (3.22) is substituted in equation (3.21), one obtains

$$p_5\mu^4 + p_4\mu^3 + p_3\mu^2 + p_2\mu + p_1 = 0 \tag{3.23}$$

The characteristic equation (3.23) has, in general, four roots. When the crack-tip speed c(t) vanishes, i.e. for static problems, Lekhnitskii [19] has proven that equation (3.23) has either pairs of complex roots or pairs of purely imaginary roots, with real roots being impossible. Furthermore, if the material becomes isotropic, equation (3.23) will have only two distinct roots $\mu = \pm i$, with $i = \sqrt{-1}$, [19].

However, as the crack-tip speed c(t) increases from zero, there becomes a situation in which, equation (3.23) will have two real roots and one pair of complex roots. When the crack-tip speed is further increased, there becomes a situation in which equation (3.23) will have four real roots. Substituting the isotropic material properties,

$a_3=a_5=0$, $a_4=a_1$ and $2(a_1-a_2)=a_6$ into equation (3.23) one obtains

$$A_1A_6\mu^4+(2A_1A_6-R(A_1+A_6))\mu^2$$
$$+(A_1A_6-R(A_1+A_6)+R^2)=0$$

where $A_i$ are the elements of matrix

$$\underset{\sim}{A}=\underset{\sim}{a}^{-1}=\begin{bmatrix} A_1 & A_2 & A_3 \\ A_2 & A_4 & A_5 \\ A_3 & A_5 & A_6 \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{16} \\ A_{12} & A_{22} & A_{26} \\ A_{16} & A_{26} & A_{66} \end{bmatrix} \qquad (3.24)$$

For isotropic materials in plane strain $A_1=\lambda+2G$ and $A_6=G$ where $\lambda$ and G are the Lame's constants. The above equation gives

$$\mu_1^2 = -1 + \left(\frac{R}{A_6}\right) \qquad \mu_2^2 = -1 + \left(\frac{R}{A_1}\right)$$

or

$$\mu_1^2 = -1 + \left(\frac{c(t)}{C_T}\right)^2 \qquad \mu_2^2 = -1 + \left(\frac{c(t)}{C_L}\right)^2$$

where

$$C_T = \sqrt{G/\rho} \qquad \text{and} \qquad C_L = \sqrt{(\lambda+2G)/\rho}$$

are the transverse (shear) and longitudinal wave speeds of the material respectively. Therefore, for isotropic materials the transitions from complex roots to real roots for the characteristic equation (3.23) take place when the crack-tip speeds are the transverse wave speed and the longitudinal wave speed respectively. However, for a freely propagating crack the crack-tip speed is only a fraction of the shear wave speed of the material. Therefore real roots of the characteristic equation (3.23) do not occur in the usual problems and are not of interest.

Therefore, here it is assumed that the crack-tip speed remains below the first transition, so that equation (3.23) has either two distinct pairs of complex roots or only one distinct pair of complex roots of multiplicity two.

If the equation (3.23) has two distinct pairs of complex roots, $\mu_1$ $\overline{\mu}_1$, and $\mu_2$, $\overline{\mu}_2$, then equation (3.22) can be written as

$$\phi = \text{Real}(T_1 + T_2)$$

where we choose

$$T_1 = \int \Omega_1 dz_1 \ , \quad T_2 = \int \Omega_2 dz_2$$

so that $$\phi = \text{Real}\ (\int \Omega_1 dz_1 + \int \Omega_2 dz_2) \qquad (3.25)$$

where $$\Omega_i = \Omega_i(z_i) \qquad \text{no sum on } i$$

$$z_i = x + \mu_i y \quad i = 1,2$$

If equation (3.23) has only one distinct pair of complex roots, $\mu$ and $\overline{\mu}$, then the equation (3.22) can be written as

$$\phi = \text{Real}(T_1 + T_2)$$

where we choose

$$T_1 = \overline{z}\Omega_1$$

$$T_2 = \int \Omega_2 dz$$

so that $$\phi = \text{Real}\ (\overline{z}\Omega_1 + \int \Omega_2 dz) \qquad (3.26)$$

where $$\Omega_i = \Omega_i(z) \qquad i = 1,2$$

$$z = x + \mu y$$

and the bar over z indicates the complex conjugate. In equations (3.25) and (3.26) we have assumed the above definitions for $T_i$ for convenience so that the displacement and stresses are expressed in terms of $\Omega_i$ and $\Omega_i'$ instead of $\Omega_i'$ and $\Omega_i''$ respectively, see [12,13,19].

Equation (3.26) is in the same form as Muskhelishvili's [12] complex vairable formulation for isotropic materials in static problems, and equation (3.25) is in the same form as that of Lekhintskii's [13,19] for anisotropic materials in static problems.

We shall now turn attention to the problem of a propagating crack at the interface of two dissimiliar anisotropic materials and derive the eigenvalues and eigenfunctions for this problem. Assume that in Fig. 3.1 the material in the positive y axis and the material in the negative y axis are different materials (see Fig. 3.2).

The equations thus far derived in this chapter are valid for either of the two materials, but from now on we shall use an index to differentiate between the two materials. We shall consider the two following cases.

Case (a). Consider the case in which equation (3.23) has two distinct pairs of complex roots $\mu_{1k}$, $\overline{\mu}_{1k}$ and $\mu_{2k}$, $\overline{\mu}_{2k}$. Then let

$$\phi_k = \text{Real}\,(\int \Omega_{1k} dz_{1k} + \int \Omega_{2k} dz_{2k}) \tag{3.27}$$

where $z_{ik} = x + \mu_{ik} y \qquad i, k = 1, 2$

and k identifies the material under consideration. Let $\Omega_{ik}$ have a complex power representation as

$$\Omega_{1k} = C_{1k}\, z_{1k}^{n} + D_{1k}\, z_{1k}^{\overline{n}} \tag{3.28}$$

$$\Omega_{2k} = C_{2k}\, z_{2k}^{n} + D_{2k}\, z_{2k}^{\overline{n}}$$

Case (b). Equation (3.23) has only one distinct pair of complex roots $\mu_k$ and $\overline{\mu}_k$. Let

$$\phi_k = \text{Real}(\overline{z}_k \Omega_{1k} + \int \Omega_{2k} dz_k) \tag{3.29}$$

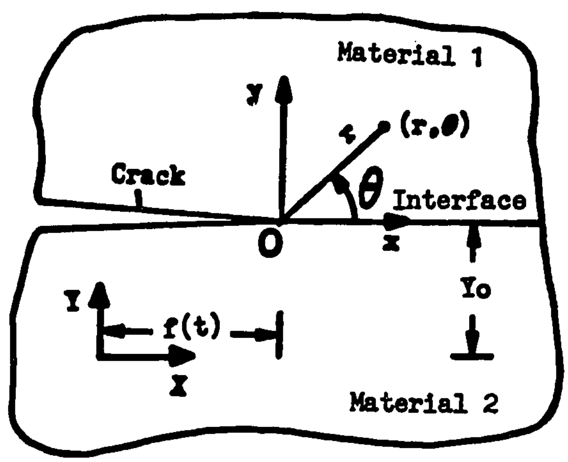



Fig. 3.2 A Propagating Crack Along the Interface of Two Dissimilar Anisotropic Materials

where $z_k = x + \mu_k y \qquad k = 1,2$

Assume that $\Omega_{ik}$ have power representation as

$$\Omega_{1k} = C_{1k} z_k^{n} + D_{1k} z_k^{\bar{n}} \tag{3.30}$$

$$\Omega_{2k} = C_{2k} z_k^{n} + D_{2k} z_k^{\bar{n}}$$

Later it will be shown that there are an infinite number of solutions for the exponent n and the functions $\Omega_{ik}$ actually become complex power series. Since n is a complex parameter, we assume that $\Omega_{ik}$ has the above form of representation which involves both n and $\bar{n}$. As discussed in the introduction chapter, M.L. Williams and others [1-5] have solved the problem of a stationary interface crack between two dissimilar isotropic material under static loads, assuming that $\Omega_{ik} = C_{ik} z_k^{n}$ and implicitly regarding n as a real variable and then recovering its complex values. However P.S. Theocaris [6] and others [21] have questioned this method and Theocaris [6] has suggested the use of a more complete form of function as in equations (3.30) and regarding n as a complex parameter. However, this author has verified that the two methods will result in identical values of n and identical expressions for displacement and stress fields for the problem of a stationary interface crack between two dissimilar isotropic materials under static loads, as considered by [1-6,21]. There is, however, no reason to believe that, for a more general problem such as the one considered here, these two methods will have identical results. It might seem that the first method, assuming implicity n to be real, will result in less calculations, since it involves only four unknown complex coefficients, $C_{ik}$, as compared to eight unknown complex coefficients, $C_{ik}$ and $D_{ik}$ involved in the second method. But this is not the case. When the restrictive conditions of traction-free surfaces and continuity of stresses and displacements ahead of the crack tip are imposed, the first method will involve $C_{ik}$ and $\bar{C}_{ik}$ as eight unknowns with eight equations, while the second method will involve only $C_{ik}$ and $\bar{D}_{ik}$ as eight unknowns with eight equations, as will be shown later (in addition to the solution for the unknown complex exponent n). Thus either method involves the same number of unknowns and the same amount of calculations, and there is no need for further verification to see whether the two methods will truly have identical results for the problem considered here. Thus we have accepted that the functions $\Omega_{ik}$ be presented in the form given by equations (3.28) and (3.30).

To begin the problem, with two different materials there are nine unknowns to be solved for, namely $C_{ik}$, $D_{ik}$, $i,k=1,2$ and the exponent n. However, expressing the conditions of zero tractions on the crack surfaces and the equality of tractions and displacements directly ahead of the crack-tip across the interface will result in eight complex homogeneous equations. To simplify the problem, four of the unknowns $C_{2k}$ and $\bar{D}_{2k}$ are solved for and substituted in the remaining four equations, so that we will have four equations in terms of 5 unknowns $C_{2k}$, $\bar{D}_{2k}$, $k=1,2$ and the exponent n.

The albegra is very involved and the detailed derivations are presented in Appendix A. For each of the cases(a) and (b) the stresses and strains are derived according to equations (3.17) and (3.18) using the definitions of equation (3.27) and (3.29) for the stress function $\phi$. The displacements are derived by integrating the strains. Then the expressions for $\sigma_{22}-i\sigma_{12}$ and $u_1+iu_2$ are formed. Following the conditions of zero tractions on the crack surfaces, i.e. $\sigma_{22}-i\sigma_{12}=0$ on the crack surfaces, we will solve for $C_{2k}$ and $\overline{D}_{2k}$ in terms of $C_{1k}$ and $\overline{D}_{1k}$. Then by substituting $C_{2k}$ and $\overline{D}_{2k}$, the expressions $\sigma_{22}-i\sigma_{12}$ and $u_1+iu_2$ for each material are expressed in terms of $C_{1k}$ and $D_{1k}$. It is shown that these expressions are of the same form for either case (a) or case (b). Then by applying the conditions of equal tractions and displacements directly ahead of the crack-tip across the interface, we will have four homogeneous equations in terms of $C_{2k}$, $\overline{D}_{2k}$ and the exponent n.

Non-trivial solutions for the complex coefficients $C_{ik}$ and $D_{ik}$ are possible only when the determinant of the resulting matrix in equation (A.29) vanishes, which gives

$$(1-x)^2\ (1+2bx+x^2)=0 \tag{3.31}$$

where $x = e^{2in\pi}$ and b is a complex combination of material properties of the materials involved and the crack-tip speed.

From equations (3.31) the acceptable eigenvalues to give finite displacements at the crack-tip are

$$n = 1,\ 2,\ 3,\ \ldots$$

$$n = \frac{1}{2} \pm i\varepsilon,\ \frac{3}{2} \pm i\varepsilon,\ \frac{5}{2} \pm i\varepsilon\ \ \ldots$$

where
$$\varepsilon = \frac{1}{2\pi} \log (b + \sqrt{b^2-1}) \tag{3.32}$$

For static problems the paramter $\varepsilon$ has been called the "bielastic constant" [3-4]. For crack propagation problems however, $\varepsilon$ is a function of the crack-tip speed as well as the material properties, therefore, here we call it the "bielastic parameter".

As discussed in Appendix A the eigenvalues $n = 1/2 + i\varepsilon$ and $n = 1/2 - i\varepsilon$ lead to the same solutions, so that only $n = 1/2 + i\varepsilon$ need to be considered. Thus the eigenvalues are taken to be

$$\begin{aligned} n_\ell &= \frac{1}{2} + i\varepsilon \ , \frac{3}{2} + i\varepsilon \ , \frac{5}{2} + i\varepsilon \ \ldots, \quad && \ell = 1, 3, 5 \ldots \\ n_\ell &= 1, 2, 3 \ldots && \ell = 2, 4, 6 \ldots \end{aligned} \tag{3.33}$$

Then by using each eigenvalue $n_\ell$ we will solve for the corresponding eigenvectors. As is common in any eigenvector problem, the unknowns $C_{ik}$ and $D_{ik}$ are determined to within an unknown scaling complex parameter $\beta_\ell$. The near-field state of stresses and displacements are then expressed in terms of only the complex unknowns $\beta_\ell$. The unknowns $\beta_\ell$ will be determined in the finite element formulation using the external boundary conditions.

For the state of displacements the eigenfunctions corresponding to the eigenvalue $n_2 = 1$ include the appropriate terms for a rigid body rotation. Therefore to complete the equations for the state of displacements, one has to add the terms corresponding to rigid body translations in X and Y directions. These terms involve the eigenfunctions corresponding to $n_0 = 0$ which has not been accounted for.

We have deliberately separated these terms for the rigid body translations in order to be able to carry out the necessary integrations for the finite element formulation in Chapter 5. Therefore, the complete form of the displacement functions are given by

$$\underset{\sim}{u} = \underset{\sim}{U}\,\underset{\sim}{\beta} \tag{3.34}$$

where

$$\underset{\sim}{U} = [\underset{\sim}{U}_1 \; \underset{\sim}{U}_2]$$

$$\underset{\sim}{\beta} = \begin{bmatrix} \underset{\sim}{\beta}_1 \\ \underset{\sim}{\beta}_2 \end{bmatrix}$$

with

$$\underset{\sim}{U}_1 = \begin{bmatrix} U_{11} & U_{12} & U_{13} & \cdots \\ U_{21} & U_{22} & U_{23} & \cdots \end{bmatrix}$$

and

$$\underset{\sim}{\beta}_1 = [\beta_{11} \; \beta_{12} \; \beta_{13} \ldots]^T$$

$$\underset{\sim}{U}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{3.35}$$

$$\underset{\sim}{\beta}_2 = [\beta_{21} \; \beta_{22}]^T$$

where $U_{m\ell}$ are the real and the imaginary parts of the complex eigenfunctions $D_{mk\ell}$ given by equations (A.36) for the $\ell^{th}$ eigenvalue and the $k^{th}$ material. The elements of $\underset{\sim}{U}_2$ correspond to a rigid body transtational motion and $\beta_{ij}$'s (real and imaginary parts of $\beta_\ell$'s renamed) are the real unknowns to be determined from the prescribed boundary conditions from the finite element formulation. Similarly the complete form of the state of stresses derived from the above displacements are given by

$$\underset{\sim}{\sigma} = \underset{\sim}{S}\,\underset{\sim}{\beta} \tag{3.36}$$

where

$$S = [\underset{\sim}{S}_1 \; \underset{\sim}{S}_2]$$

with

$$\underset{\sim}{S}_1 = \begin{bmatrix} (\sigma_{11})_1 & (\sigma_{11})_2 & (\sigma_{11})_3 & \cdots \\ (\sigma_{22})_1 & (\sigma_{22})_2 & (\sigma_{22})_3 & \cdots \\ (\sigma_{12})_1 & (\sigma_{12})_2 & (\sigma_{12})_3 & \cdots \end{bmatrix}$$

and

$$\underset{\sim}{S}_2 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

where $(\sigma_{ij})_\ell$ are the real and the imaginary parts of the complex eigenfunctions $S_{nk\ell}$ given by equations (A.36) for the $\ell^{th}$ eigenvalue and the $k^{th}$ material, and elements of $\underset{\sim}{S}_2$ are the stresses corresponding to the rigid body translations $\underset{\sim}{U}_2$ which are identically zero.

This concludes the derivations of the complete state of near-field stresses and displacements for a propagating crack at the interface of two dissimilar anisotropic materials.

# CHAPTER IV

## DERIVATION OF SINGULAR AND REGULAR ELEMENT FUNCTIONS

For the purpose of finite element formulation, the body containing the crack is divided into subregions. The singular elements are placed at the regions containing the crack tip and regular elements fill the remainder. It is necessary to formulate the internal stresses and displacements. Also, the boundary tractions and boundary displacements for the elements must be formulated independently from the internal stresses and displacements.

For the singular elements the internal displacements and stresses are as expressed in equations (A.36). In the functional equation (Chapter 5) it is also necessary to formulate the boundary displacements and tractions for the singular element, independently from the internal stresses and displacements. To derive the boundary displacement functions, we consider the singular element shown in Fig. 4.1. The element has 10 nodes and the crack-tip is not associated with any node. Although nodes 5 and 6 could be considered as one node, we have deliberately treated this point as two separate nodes. As discussed earlier in Chapter 1, when the tip of a propagating crack passes through this point two separated nodes are needed. But before the crack-tip passes through this point, these two nodes are one and the same. One should note that, since the tip of a running crack always stays inside the singular element by a local re-meshing process, see Chapter 7, nodes 5 and 6 always coincide. Therefore, it is possible to formulate the singular element with 9 nodes instead

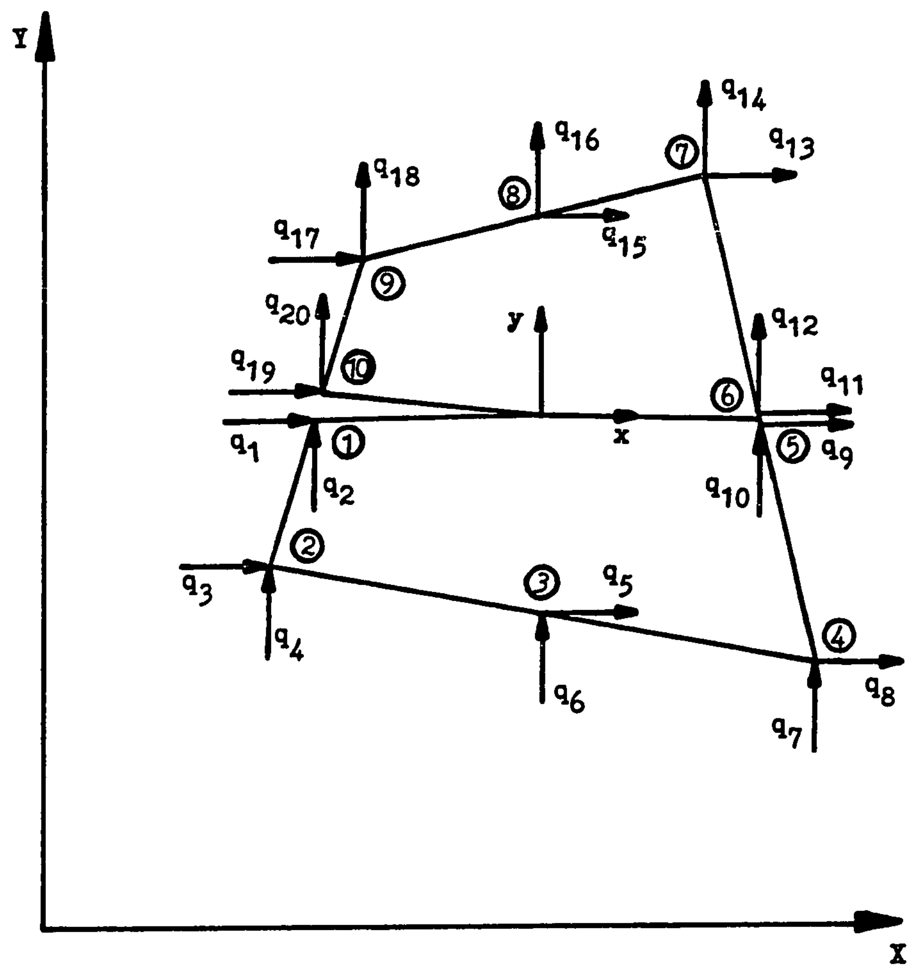



Fig. 4.1 Nodal Displacement Vectors for the Singular Element

of 10 nodes, i.e. taking nodes 5 and 6 to be one single node. However, for the purpose of computer programming it seemed easier to have two nodes at this mesh point.

Letting $q_1$ to $q_{20}$ represent the displacements at the nodes of the singular element, it is assumed that the boundary displacements are given by

$$\underset{\sim}{v} = \underset{\sim}{L}\,\underset{\sim}{q} \tag{4.1}$$

where

$$\underset{\sim}{v} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}, \quad \underset{\sim}{q} = [q_1 \; \ldots \; q_{20}]^T$$

$$\underset{\sim}{L} = \begin{bmatrix} L_{1\,1} & L_{1\,2} & L_{1\,3} & \cdots & L_{1\,20} \\ L_{2\,1} & L_{2\,2} & L_{2\,3} & \cdots & L_{2\,20} \end{bmatrix}$$

and the elements of matrix $\underset{\sim}{L}$ are chosen such that the boundary displacements $v_i$ vary linearly along the boundary and are equal to the nodal displacements at the nodes. For example, for the segment between nodes 7 and 8, one has

$$\begin{aligned} L_{1\,13} &= 1-s \\ L_{1\,15} &= s \\ L_{2\,14} &= 1-s \\ L_{2\,16} &= s \end{aligned} \tag{4.2}$$

with all other $L_{ij}=0$, and s is a parameter on the boundary chosen so that s=0 at node 7 and s=1 at node 8. This gives

$$\begin{aligned} v_1 &= (1-s)q_{13} + sq_{15} \\ v_2 &= (1-s)q_{14} + sq_{16} \end{aligned} \tag{4.3}$$

which is the desired form of the boundary displacement functions described above.

For the formulation of the boundary tractions, it is assumed that

$$\underset{\sim}{t} = \underset{\sim}{R}\,\underset{\sim}{\alpha} \tag{4.4}$$

where

$$\underset{\sim}{t} = \begin{bmatrix} t_1 \\ t_2 \end{bmatrix}, \quad R = \begin{bmatrix} R_{11} & R_{12} & R_{13} & \cdots \\ R_{21} & R_{22} & R_{23} & \cdots \end{bmatrix}$$

$$\alpha = [\alpha_1 \ \alpha_2 \ \alpha_3 \ \ldots]^T$$

where $t_i$ are tractions and $R_{ij}$ are the functions discussed below, and $\alpha_i$ are unknown constants to be determined from the finite element formulation.

The functions $R_{ij}$ can be chosen such that the boundary tractions are distributed linearly along the boundary and are equal to nodal values $\alpha_i$ at the nodes, in the same manner as the boundary displacement functions $L_{ij}$ were chosen. In that case there will be 20 unknown $\alpha_i$'s to be determined.

The functions $R_{ij}$ could also be chosen to give constant boundary tractions along all edges, in which case one has

$$\underset{\sim}{t} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix}$$

with only two unknown $\alpha_i$'s representing the constant tractions along all edges. Other forms of traction distributions are acceptable as well.

However, following the discussion in Chapter 5, it is preferable to choose the number of unknowns $\alpha_i$ to be equal to the number of unknowns $\beta_i$, i.e., the unknowns for the internal displacement functions. One way to do this, is to derive the functions $R_{ij}$ from the displacement functions $U_{ij}$, as

$$\underset{\sim}{R} = \underset{\sim}{n}\,\underset{\sim}{S} = \underset{\sim}{n}\,\underset{\sim}{A}(\underset{\sim}{d}\,\underset{\sim}{U}) \tag{4.5}$$

where

$$\underset{\sim}{n} = \begin{bmatrix} n_x & 0 & n_y \\ 0 & n_y & n_x \end{bmatrix} \tag{4.6}$$

$$\underset{\sim}{d} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \tag{4.7}$$

Here, $n_x$ and $n_y$ are the components of an outward unit vector normal to the sides of the singular element, $\underset{\sim}{S}$, $\underset{\sim}{A}$ and $\underset{\sim}{U}$ are defined by equations (3.36), (3.24) and (3.34) respectively.

In doing so, one should note that the terms corresponding to rigid body displacements do not contribute to these tractions. Thus, if $\underset{\sim}{t}$ is represented as

$$\underset{\sim}{t} = \underset{\sim}{R}\,\underset{\sim}{\alpha} \tag{4.8}$$

where

$$\underset{\sim}{R} = [\underset{\sim}{R}_1 \; \underset{\sim}{R}_2]$$

and

$$\underset{\sim}{\alpha} = \begin{bmatrix} \underset{\sim}{\alpha}_1 \\ \underset{\sim}{\alpha}_2 \end{bmatrix}$$

with

$$\underset{\sim}{R}_1 = \begin{bmatrix} R_{11} & R_{12} & R_{13} & R_{14} & \ldots R_{1n} \\ R_{21} & R_{22} & R_{23} & R_{24} & \ldots R_{2n} \end{bmatrix}$$

$$\underset{\sim}{R}_2 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\underset{\sim}{\alpha}_1 = [\alpha_{11} \; \alpha_{12} \; \alpha_{13} \; \alpha_{14} \; \ldots \alpha_{1n}]^T$$

and

$$\underset{\sim}{\alpha}_2 = [\alpha_{21} \; \alpha_{22}]^T$$

so that the elements of $\underset{\sim}{R}_2$ corresponding to the rigid body translational motion all vanish. One should also note that since the displacement eigenfunctions corresponding to $n_2$=1 contain rigid body rotation terms, the third and the fourth columns of $\underset{\sim}{R}_1$ are not independent, with the possibility that the elements of the fourth column are all zero. This happens when the roots of the characteristic equation (3.23) are pure imaginary, e.g. when isotropic materials are involved.

As discussed in Chapter 5 the elements of matrix $\underset{\sim}{R}_1$ should be chosen so that the matrix $\underset{\sim}{P}$ (Chapter 5) is invertible. Therefore, we shall replace the elements of matrix $\underset{\sim}{R}_2$ with some functions that will make the matrix $\underset{\sim}{P}$ non-singular. One way to do this is to choose the elements of matrix $\underset{\sim}{R}_2$ as if the body had a purely translational motion. i.e.

$$\underset{\sim}{R}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{4.9}$$

and choose the elements of the $4^{th}$ column of matrix $\underset{\sim}{R}_1$ as if the body had a purely rotational motion, i.e.

$$\begin{bmatrix} R_{14} \\ R_{24} \end{bmatrix} = \begin{bmatrix} T_1(X,Y) \\ T_2(X,Y) \end{bmatrix}$$

so that the function $T_1(X,Y)$ and $T_2(X,Y)$ will result in a non-zero resultant moment, e.g., see Fig. 4.2, for a square singular element for which

$$T_1 = \begin{cases} 1 & \text{on side 1} \\ 0 & \text{on sides 2 and 4} \\ -1 & \text{on side 3} \end{cases}$$

and

$$T_2 = \begin{cases} 1 & \text{on side 2} \\ 0 & \text{on sides 1 and 3} \\ -1 & \text{on side 4} \end{cases}$$

When the elements of matrix $\underset{\sim}{R}$ are chosen as described here, the matrix $\underset{\sim}{P}$ (Chapter 5) will be square and invertible.

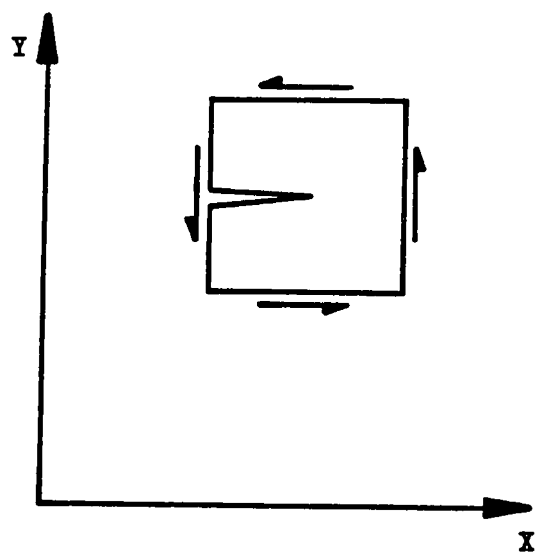



Fig. 4.2 Traction Distribution on a Square Singular Element for a Purely Rotational Motion

This concludes the derivations of all the necessary functions needed for the singular element.

The regular elements are assumed to be four-noded quadrilateral elemnts. These elements are subsequently tranformed into four noded square "parent elements", for which

$$X = N_i X_i \qquad \text{sum on } i \quad i = 1,4 \qquad (4.11)$$
$$Y = N_i Y_i$$

where X-Y is a set of global coordinate system and $X_i$, $Y_i$ represent the positions of the nodes of the element in X-Y coordinate system. $N_i$ are the familiar bilinear shape functions as follows

$$\begin{aligned} N_1 &= 1/4(1-\zeta)(1-\eta) \\ N_2 &= 1/4(1+\zeta)(1-\eta) \\ N_3 &= 1/4(1+\zeta)(1+\eta) \\ N_4 &= 1/4(1-\zeta)(1+\eta) \end{aligned} \qquad (4.12)$$

where $\zeta$-$\eta$ is a set of coordinate systems with its origin at the center of the square parent element and with $\zeta$ and $\eta$ axis parallel to the sides of the square, see Fig. 4.3.

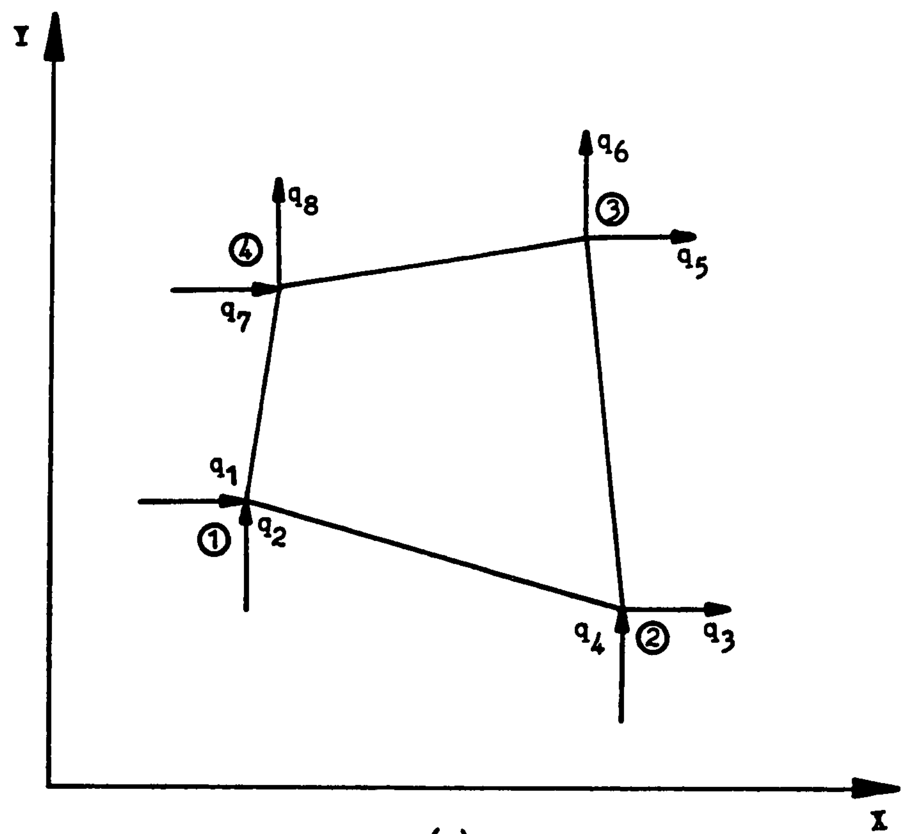
Furthermore, it is assumed that the regular elements are isoparametric, i.e., the field functions are assumed to be in terms of the bilinear functions $N_i$, as follows

$$\underset{\sim}{u} = \underset{\sim}{N}\underset{\sim}{q}$$

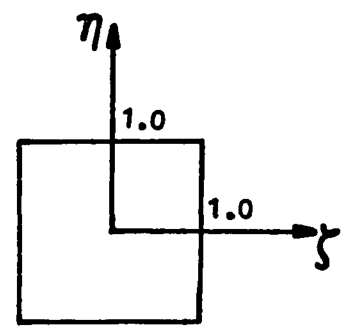
where $\underset{\sim}{u} = [u_1 \ u_2]^T$ is the displacement matrix and $\underset{\sim}{q} = [q_1 \ q_2 \ \ldots \ q_8]^T$ is a matrix containing the eight nodal displacements, and

$$\underset{\sim}{N} = \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 \end{bmatrix} \qquad (4.13)$$

The boundary displacments $\underset{\sim}{v}$ are assumed to be given by the values $\underset{\sim}{u}$ at the element boundary. For two adjacent regular elements the values of




(a)




(b)

Fig. 4.3 (a)-A Four Noded Quadrilateral, (b)-A Square Parent Element nodal displacments at a common boundary are the same, so that the constraint integral in the functional equation (Chapter 5) automatically vanishes. Therefore there is no need to formulate boundary traction functions $\underset{\sim}{t}$ for regular elements.

The stresses are assumed to be derived from the displacement functions as

$$\underset{\sim}{\sigma} = \underset{\sim}{A}\underset{\sim}{\varepsilon} = \underset{\sim}{A}(\underset{\sim}{d}\underset{\sim}{u}) \tag{4.14}$$

where $\underset{\sim}{A}$ is defined by equation (3.24), $\underset{\sim}{\varepsilon}$ is strain matrix, and $\underset{\sim}{d}$ is defined by equation (4.7). Since the $\zeta$-$\eta$ is a material coordinate system, then the time derivatives of the displacements are given by

$$\dot{\underset{\sim}{u}} = \underset{\sim}{N}\dot{\underset{\sim}{q}} \quad \text{and} \quad \ddot{\underset{\sim}{u}} = \underset{\sim}{N}\ddot{\underset{\sim}{q}} \ . \tag{4.15}$$

With the above assumed functions for the singular and regular elements, it is now possible to employ the finite element method to formulate the problem under consideration. In the next chapter a detailed analysis of the finite elment formulation is presented.

# CHAPTER V

## FINITE ELEMENT FORMULATION

Since the finite element technique has proven to be a powerful method for analyzing a wide range of complicated problems, such as elasticity problems, etc., its extension to crack problems seems natural. The finite element methods are generally classified, according to the unknowns to be found at the nodal points, into the force method, the displacement method, and the mixed method [22-24]. For the problem considered here, a type of displacement finite element method called the "hybrid-displacement" finite element method is employed. The body is assumed to occupy a planar region R, with the boundary S which is subdivided into subregions $R_n$ with the boundaries $S_n$ , n=1,2,3...p, where, p is the number of elements or subregions in R. The elements $R_n$ are divided into "singular elements" and "regular elements." The elements containing the crack-tips are called the singular elements and the elements not containing the crack-tips are called the regular elements; see Fig. 5.1. The body is assumed to contain a crack which may grow with time. Also, each crack tip is assumed to be embedded in one of the subregions, say $R_1$ and $R_2$, with boundaries $S_1$ and $S_2$. Subregions $R_n$, n=3,4,...p may border on part of the crack surface. For convenience, it is assumed that the subregions $R_1$ and $R_2$ are squares and that the other subregions are quadrilaterals. The boundary S is assumed to consist of straight line segments.

Since different approximating functions for displacement fields are considered for the singular and regular elements, discontinuous

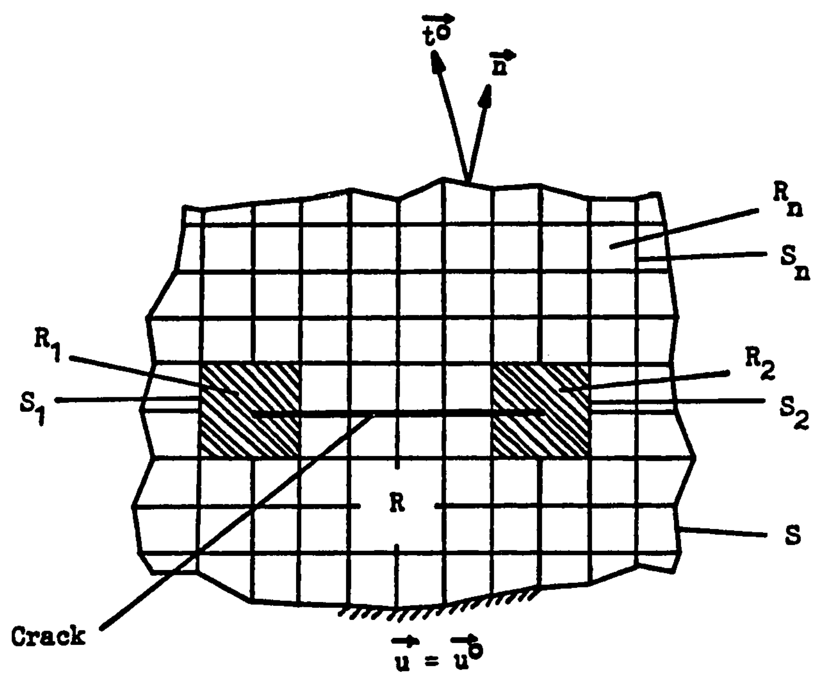



Fig. 5.1 A Cracked Body at Some Fixed Instant of Time

displacement fields across the boundaries of the singular and regular elements will have to be considered. To address this problem, the so-called "hybrid displacement" functional [14-16] is used as follows. Define a functional $\Pi$ as

$$\Pi = \int_{t_0}^{t_1} \sum_{n=1}^{p} \{\int_{R_n}[1/2\sigma_{ij}\varepsilon_{ij}-\rho b_i u_i-1/2\rho\dot{u}_i\dot{u}_i]dv + \int_{S_{in}} t_i(v_i-u_i)ds-\int_{S_{tn}} \overset{o}{t}_i v_i ds\}dt \tag{5.1}$$

Where $S_{in}$ are the boundaries of the subregions and $S_{tn}$ is the part of the boundary where traction boundary conditions $\overset{o}{t}_i$ are prescribed.

The displacements $v_i$ on the boundary of each element are assumed to be linear functions of the nodal displacments. The integral $\int_{S_{in}} t_i(v_i-u_i)ds$ in equation (5.1) is added to the functional to enforce the continuity of displacements between the singular elements and the regular elements, which use different approximating displacement functions. This constraint integral will, in the limit, force the interior displacements $u_i$ to equal the boundary displacements $v_i$ for each element. Since the displacements $v_i$ are the same for two adjacent elements, the contraint integral therefore will insure, in the limit, continuity between all elements. It is assumed that the boundary displacements $v_i$ will satisfy the displacement boundary conditions prescribed on the part of the boundary $S_u$, i.e.

$$v_i = v_i^o \text{ on } S_u \tag{5.2}$$

It follows then that the variation $\delta v_i = 0$ on $Su_i$. Using the constitutive equations (2.8) and the boundary condition (5.2) and also assuming that the variation $\delta u_i$ vanishes at times $t_0$ and $t_1$, one can show that the vanishing of the variation $\delta\Pi$ for arbitrary $\delta u_i$ in R, for arbitrary $\delta t_i$ on $S_{in}$ and for admissible $\delta v_i$ on $S_{in}$, such that $\delta v_i = 0$ on $S_u$, gives the following

$$\sigma_{ij,j} + \rho b_i = \rho \ddot{u}_i \quad \text{in } R$$

$$u_i = v_i \qquad \text{on } S_{in}$$

$$\sigma_{ij} n_j = t_i \qquad \text{on } S_{in}$$

$$t_i = t_i^o \qquad \text{on } S_{tn}$$

The first equation states that the displacement functions $u_i$ in R generate stresses $\sigma_{ij}$, from equations (2.1) and (2.8) that satisfy the local equilibrium equation in R. The second equation states that the values of $u_i$ at the boundary $S_n$ coincide with the inter-element boundary displacement $v_i$, which is treated as an independent unknown, thus enforcing interelement continuity for displacements. The third equation states that the boundary tractions $t_i$, which are treated as independent unknowns in the problem, coincide with the tranctions $\sigma_{ij} n_j$ generated at the boundary $S_n$ by the function $u_i$. The fourth equation states that the tractions generated at the boundary $S_{tn}$ coincides with the tractions specified on this part of the boundary.

Thus, of all the admissible functions $u_i$ in R and $v_i$ and $t_i$ on $S_n$, those functions that render the functional $\Pi$ of equation (5.1) stationary will satisfy all the necessary requirements stated above. Substituting the assumed functions $u_i$, $v_i$, $t_i$ for the singular element derived in Chapter 3 and 4 into equation (5.1) one obtains

$$\Pi_s = \int_{t_o}^{t_1} (1/2 \underset{\sim}{\beta}^T \underset{\approx}{H} \underset{\sim}{\beta} - \underset{\sim}{\beta}^T \underset{\sim}{F} - 1/2 \underset{\sim}{\beta}^T \underset{\approx}{M}_1 \underset{\sim}{\beta} - 1/2 \dot{\underset{\sim}{\beta}}^T \underset{\approx}{M}_2 \dot{\underset{\sim}{\beta}} \tag{5.3}$$

$$- \underset{\sim}{\beta}^T \underset{\approx}{M}_3 \dot{\underset{\sim}{\beta}} + \underset{\sim}{\alpha}^T \underset{\approx}{G} \underset{\sim}{q} - \underset{\sim}{\alpha}^T \underset{\approx}{P} \underset{\sim}{\beta} - \underset{\sim}{q}^T \underset{\sim}{T}) dt$$

where

$$\underset{\sim}{H} = \int_{R_n} \underset{\sim}{E}^T \underset{\sim}{A} \underset{\sim}{E} dv$$

$$\underset{\sim}{F} = \int_{R_n} \underset{\sim}{U}^T \rho \underset{\sim}{b} dv$$

$$\underset{\sim}{M}_1 = \int_{R_n} \dot{\underset{\sim}{U}}^T \rho \dot{\underset{\sim}{U}} dv$$

$$\underset{\sim}{M}_2 = \int_{R_n} \underset{\sim}{U}^T \rho \underset{\sim}{U} dv$$

$$\underset{\sim}{M}_3 = \int_{R_n} \dot{\underset{\sim}{U}}^T \rho \underset{\sim}{U} dv$$

$$\underset{\sim}{G} = \int_{S_{in}} \underset{\sim}{R}^T \underset{\sim}{L} ds$$

$$\underset{\sim}{P} = \int_{S_{in}} \underset{\sim}{R}^T \underset{\sim}{U} ds$$

$$\underset{\sim}{T} = \int_{S_{tn}} \underset{\sim}{L}^T \overset{o}{\underset{\sim}{t}} ds$$

and where $\underset{\sim}{A}$ is the stress-strain constitutive relation matrix defined by equation (3.24), and $\underset{\sim}{E}$ is defined by

$$\underset{\sim}{E} = \underset{\sim}{d}\underset{\sim}{U}$$

with $\underset{\sim}{d}$ as defined by equation (4.7). The dots indicate the time derivatives defined by equations (3.4a) as

$$\dot{\underset{\sim}{U}} = -c(t) \frac{\partial \underset{\sim}{U}}{\partial x} \tag{5.3b}$$

Since the coefficients $\underset{\sim}{\alpha}$ and $\underset{\sim}{\beta}$ of the singular element are assumed to be independent of the nodal displacements $\underset{\sim}{q}$, then the vanishing of the variation of equation (5.3) with respect to the independent coefficients $\underset{\sim}{\alpha}$ and $\underset{\sim}{\beta}$, for each singular element, yields

$$\int_{t_0}^{t_1}[\delta\beta^T H\beta - \delta\beta^T F - \delta\beta^T M_1\beta - \delta\dot{\beta}^T M_2\dot{\beta} - \delta\beta^T M_3\dot{\beta} - \beta^T M_3\delta\dot{\beta}$$

$$- \alpha^T P\delta\beta]dt = 0 \qquad (5.4)$$

and

$$\int_{t_0}^{t_1}\delta\alpha^T[Gq - P\beta]dt = 0 \qquad (5.5)$$

Assuming that the variations of $\beta$ vanish at times $t_0$ and $t_1$, it is possible to integrate equation (5.4) to yield

$$H\beta - F - M_1\beta + \dot{M}_2\beta + M_2\ddot{\beta} - M_3\dot{\beta} + \dot{M}_3^T\beta + M_3^T\dot{\beta} - P^T\alpha = 0 \qquad (5.6)$$

Also, equation (5.5) gives

$$Gq - P\beta = 0 \qquad (5.7)$$

Equation (5.6) is a set of $n_\beta$ equations and equation (5.7) is a set of $n_\alpha$ equations, where, $n_\alpha$ and $n_\beta$ are the number of unknown coefficients $\alpha$ and $\beta$ respectively. Therefore it is possible to solve for the coefficients $\alpha$ and $\beta$ in terms of the nodal displacements $q$, using equations (5.6) and (5.7). From equation (5.7) it is also observed that one should select $n_\beta > n_\alpha$.

It is noted that the simultaneous solution of equations (5.6) and (5.7) is not an easy task. The matrices have to be partitioned, as done in [41] for static problems. In addition, temporal integration numerical techniques have to be employed to solve for the coeffieicents $\alpha$ and $\beta$ in terms of the nodal displacements $q$. However, it is noted that if one selects $n_\alpha = n_\beta$, then the matrix $P$ becomes square and assuming $P$ is invertible, then, from equation (5.7) one obtains

$$\underset{\sim}{\beta} = \underset{\sim}{P}^{-1}\underset{\sim\sim}{Gq} \tag{5.8}$$

Equation (5.8) can be used in equation (5.7) to solve for the coefficients $\underset{\sim}{\alpha}$, but this is not necessary since, when equation (5.8) is substituted in the functional equation (5.3), the coefficients $\underset{\sim}{\alpha}$ are automatically eliminated, and thus the solution for $\underset{\sim}{\alpha}$ is not needed. Therefore the advantage of selecting $n_\alpha = n_\beta$ is obvious. The invertibility of the matrix $\underset{\sim}{P}$ must be assured by proper selection of the assumed eigenfunctions for $\underset{\sim}{U}$ and $\underset{\sim}{R}_4$ as discussed in Chapter 3 and 4.

Introducing

$$\underset{\sim}{B} = \underset{\sim}{P}^{-1}\underset{\sim}{G} \tag{5.9}$$

equation (5.8) becomes

$$\underset{\sim}{\beta} = \underset{\sim\sim}{Bq} \tag{5.10}$$

Introducing equation (5.10) into the function equation (5.3) yields

$$\Pi s = \int_{t_0}^{t_1} [1/2\ \underset{\sim}{q}^T \underset{\sim}{K}_1 \underset{\sim}{q} - 1/2\ \dot{\underset{\sim}{q}}^T \underset{\sim}{M}_1^* \dot{\underset{\sim}{q}} - \dot{\underset{\sim}{q}}^T \underset{\sim}{V}_1 \underset{\sim}{q} - \underset{\sim}{q}^T \underset{\sim}{F}_s]dt \tag{5.11}$$

where

$$\underset{\sim}{K}_1 = \underset{\sim}{B}^T \underset{\sim\sim}{HB} - \underset{\sim}{B}^T \underset{\sim}{M}_1 \underset{\sim}{B} - \dot{\underset{\sim}{B}}^T \underset{\sim}{M}_2 \dot{\underset{\sim}{B}} - 2\underset{\sim}{B}^T \underset{\sim}{M}_3 \dot{\underset{\sim}{B}}$$

$$\underset{\sim}{M}_1 = \underset{\sim}{B}^T \underset{\sim}{M}_2 \underset{\sim}{B}$$

$$\underset{\sim}{V}_1 = \underset{\sim}{B}^T \underset{\sim}{M}_2 \dot{\underset{\sim}{B}} + \underset{\sim}{B}^T \underset{\sim}{M}_3{}^T \underset{\sim}{B} \tag{5.12}$$

$$\underset{\sim}{F}s = \underset{\sim}{B}^T \underset{\sim}{F} + \underset{\sim}{T}$$

For a propagating crack the matrix $\underset{\sim}{M}_1^*$ is symmetric, while the matrices $\underset{\sim}{K}_1$ and $\underset{\sim}{V}_1$ are not. For a stationary crack the matrix $\underset{\sim}{V}_1$ vanishes and the matrix $\underset{\sim}{K}_1$ becomes symmetric. Equation (5.11) is the functional equation valid for the singular elements.

For regular elements, substituting equation of Chapter 4 into the functional equation (5.1) yields

$$\Pi_R = \int_{t_0}^{t_1} [1/2\, \underset{\sim}{q}^T \underset{\sim}{K}_R \underset{\sim}{q} - 1/2\, \underset{\sim}{q}^T \underset{\sim}{M}_R \dot{\underset{\sim}{q}} - \underset{\sim}{q}^T \underset{\sim}{F}_R] dt \tag{5.13}$$

where

$$\underset{\sim}{K}_R = \int_{R_n} (\underset{\sim}{d}\underset{\sim}{N})^T \underset{\sim}{A} (\underset{\sim}{d}\underset{\sim}{N}) dv$$

$$\underset{\sim}{M}_R = \int_{R_n} \underset{\sim}{N}^T \rho \underset{\sim}{N} dv \tag{5.14}$$

$$\underset{\sim}{F}_R = \int_{S_{tn}} \underset{\sim}{N}^T \underset{\sim}{t}^o ds + \int_{R_n} \underset{\sim}{N}^T \rho \underset{\sim}{b} dv$$

Denoting the global nodal displacements by $\underset{\sim}{q}^*$, and combining all the element matrices into corresponding global matrices yields

$$\Pi = \int_{t_0}^{t_1} \{ \sum_{n=1}^{2} (1/2\, \underset{\sim}{q}^{*T} \overline{\underset{\sim}{K}}_1 \underset{\sim}{q}^* - 1/2\, \dot{\underset{\sim}{q}}^{*T} \overline{\underset{\sim}{M}}_1 \dot{\underset{\sim}{q}}^* - \dot{\underset{\sim}{q}}^{*T} \underset{\sim}{V}_1 \underset{\sim}{q}^* - \underset{\sim}{q}^{*T} \overline{\underset{\sim}{F}}_S) \tag{5.15}$$
$$+ \sum_{n=3}^{p} (1/2\, \underset{\sim}{q}^{*T} \overline{\underset{\sim}{K}}_R \underset{\sim}{q}^* - 1/2\, \dot{\underset{\sim}{q}}^{*T} \overline{\underset{\sim}{M}}_R \dot{\underset{\sim}{q}}^* - \underset{\sim}{q}^{*T} \overline{\underset{\sim}{F}}_R) \} dt$$

where the bar over the matrices denote the global matrices corresponding to the element matrices.

The variation of $\Pi$ with respect to $\underset{\sim}{q}^*$, yields

$$\delta\Pi = \int_{t_0}^{t_1} \{ \sum_{n=1}^{2} (1/2 \delta \underset{\sim}{q}^{*T} (\overline{\underset{\sim}{K}}_1 + \overline{\underset{\sim}{K}}_1^T) \underset{\sim}{q}^* - \delta \dot{\underset{\sim}{q}}^{*T} \overline{\underset{\sim}{M}}_1 \dot{\underset{\sim}{q}}^* - \delta \dot{\underset{\sim}{q}}^{*T} \underset{\sim}{V}_1 \underset{\sim}{q}^*$$
$$- \delta \underset{\sim}{q}^{*T} \overline{\underset{\sim}{V}}_1^T \dot{\underset{\sim}{q}}^* - \delta \underset{\sim}{q}^{*T} \overline{\underset{\sim}{F}}_S)$$
$$+ \sum_{n=3}^{p} (\delta \underset{\sim}{q}^{*T} \overline{\underset{\sim}{K}}_R \underset{\sim}{q}^* - \delta \dot{\underset{\sim}{q}}^{*T} \overline{\underset{\sim}{M}}_R \dot{\underset{\sim}{q}}^* - \delta \underset{\sim}{q}^{*T} \overline{\underset{\sim}{F}}_R) \} dt$$

Assuming that the variation of $\delta \underset{\sim}{q}^*$ vanishes at times $t_o$ and $t_1$, integration in time yields

$$\delta\Pi = \int_{t_0}^{t_1}\{\delta \underset{\sim}{q}^{*T}[\sum_{n=1}^{2}(\underset{\sim}{\bar{K}}_s \underset{\sim}{q}^* + \underset{\sim}{\bar{V}}_s \dot{\underset{\sim}{q}}^* + \underset{\sim}{\bar{M}}_s \ddot{\underset{\sim}{q}}^* - \underset{\sim}{\bar{F}}_s)]$$

$$+ \delta \underset{\sim}{q}^{*T}[\sum_{n=3}^{p}(\underset{\sim}{\bar{K}}_R \underset{\sim}{q}^* + \underset{\sim}{\bar{M}}_R \ddot{\underset{\sim}{q}}^* - \underset{\sim}{\bar{F}}_R)]\}dt$$

where

$$\underset{\sim}{\bar{K}}_s = 1/2(\underset{\sim}{\bar{K}}_1 + \underset{\sim}{\bar{K}}_1^T) + \dot{\underset{\sim}{\bar{V}}}_1$$

$$\underset{\sim}{\bar{V}}_s = \dot{\underset{\sim}{\bar{M}}}_1^* + \underset{\sim}{\bar{V}}_1 - \underset{\sim}{\bar{V}}_1^T \tag{5.16}$$

$$\underset{\sim}{\bar{M}}_s = \underset{\sim}{\bar{M}}_1^*$$

The variation $\delta\Pi$ should vanish for arbitrary $\delta\underset{\sim}{q}^*$, yielding

$$\underset{\sim}{M}\ddot{\underset{\sim}{q}}^* + \underset{\sim}{V}\dot{\underset{\sim}{q}}^* + \underset{\sim}{K}\underset{\sim}{q}^* = \underset{\sim}{Q} \tag{5.17}$$

where

$$\underset{\sim}{M} = \sum_{n=1}^{2} \underset{\sim}{\bar{M}}_s + \sum_{n=3}^{p} \underset{\sim}{\bar{M}}_R$$

$$\underset{\sim}{V} = \sum_{n=1}^{2} \underset{\sim}{\bar{V}}_s \tag{5.18}$$

$$\underset{\sim}{K} = \sum_{n=1}^{2} \underset{\sim}{\bar{K}}_s + \sum_{n=3}^{p} \underset{\sim}{\bar{K}}_R$$

$$\underset{\sim}{Q} = \sum_{n=1}^{2} \underset{\sim}{\bar{F}}_s + \sum_{n=3}^{p} \underset{\sim}{\bar{F}}_R$$

The global matrices $\underset{\sim}{M}$, $\underset{\sim}{V}$, $\underset{\sim}{K}$ and $\underset{\sim}{Q}$ are obtained by summing all the element matrices, a process conventially referred to as the merging of the element matrices.

Equation (5.17) represents the governing equation of motion for the elastodynamic problem considered here. It is noted that although there is no damping system present in the problem, the matrix $\underset{\sim}{V}$ called the "pseudo-damping" matrix is present. From equations (5.3a), (5.12) and (5.18) it is also noted that the stiffness matrix $\underset{\sim}{K}$ is not symmetric. Furthermore from equation (5.18) it is deduced that only the singular elements contribute to the pseudo-damping matrix $\underset{\sim}{V}$ and only the singular elements contribute to the non-symmetricity of the stiffness matrix $\underset{\sim}{K}$. These complexities arise due to the fact that the eigen functions for the singular element were derived with respect to a moving local coordinate system x-y at the crack-tip and therefore, matrices $\underset{\sim}{M}_1^*$, $\underset{\sim}{K}_1$ and $\underset{\sim}{V}_1$ are functions of time. An examination of equations (5.3a), (5.3b) and (5.12) reveals that when the instantaneous crack-tip velocity c(t) vanishes, so that the local crack-tip coordinate system x-y becomes stationary, then the pseudo-damping matrix $\underset{\sim}{V}$ vanishes and the stiffness matrix $\underset{\sim}{K}$ becomes symmetric.

From equation (5.16) the corresponding element matrix for the singular elements can be written as

$$\begin{aligned} \underset{\sim}{K}_s &= 1/2(\underset{\sim}{K}_1 + \underset{\sim}{K}_1{}^T) + \dot{\underset{\sim}{V}}_1 \\ \underset{\sim}{V}_s &= \dot{\underset{\sim}{M}}{}_1^* + \underset{\sim}{V}_1 - \underset{\sim}{V}_1^T \\ \underset{\sim}{M}_s &= \underset{\sim}{M}_1^* \end{aligned} \qquad (5.19)$$

Calculating the time derivatives of $\underset{\sim}{V}_1$ and $\underset{\sim}{M}_1^*$ and using the fact that

$$\dot{\underset{\sim}{M}}_2 = \underset{\sim}{M}_3 + \underset{\sim}{M}_3{}^T$$

and

$$\dot{\underset{\sim}{M}}_3 = \underset{\sim}{M}_1 + \underset{\sim}{M}_4$$

with

$$\underset{\sim}{M}_4 = \int_{R_n} \underset{\sim}{\ddot{U}}^T \rho \underset{\sim}{U} dv \tag{5.20}$$

equation (5.19) yields

$$\begin{aligned} \underset{\sim}{K}_s &= \underset{\sim}{B}^T \underset{\sim}{H} \underset{\sim}{B} + \underset{\sim}{B}^T (\underset{\sim}{M}_2 \underset{\sim}{\ddot{B}} + 2\underset{\sim}{M}_3^T \underset{\sim}{\dot{B}} + \underset{\sim}{M}_4^T \underset{\sim}{B}) \\ \underset{\sim}{V}_s &= 2\underset{\sim}{B}^T (\underset{\sim}{M}_2 \underset{\sim}{\dot{B}} + \underset{\sim}{M}_3^T \underset{\sim}{B}) \\ \underset{\sim}{M}_s &= \underset{\sim}{B}^T \underset{\sim}{M}_2 \underset{\sim}{B} \\ \underset{\sim}{F}_s &= \underset{\sim}{B}^T \underset{\sim}{F} + \underset{\sim}{T} \end{aligned} \tag{5.21}$$

The regular element matrices can be evaluated from equation (5.14) by evaluating the integrals using numerical techniques such as the Gaussian integration method without any difficulty. However, in evaluating the singular element matrices of equation (5.20) it is noted that when the various functions for the singular element derived in Chapter 3 and 4 are substituted into equations (5.3a) and (5.20), matrices $\underset{\sim}{H}$, $\underset{\sim}{M}_3$, and $\underset{\sim}{M}_4$ contain various orders of singularities. Therefore, some modifications have to be made before one can perform the intergration task. From equations (5.3a) and (5.20) writing the matrices $\underset{\sim}{H}$ and $\underset{\sim}{M}_4$ in decomposed form one has

$$\underset{\sim}{H} = \int_{R_n} \underset{\sim}{E}^T \underset{\sim}{A} \underset{\sim}{E} dv = \begin{bmatrix} \int_{R_n} (\underset{\sim}{dU}_1)^T \underset{\sim}{A} (\underset{\sim}{dU}_1) dv & 0 \\ 0 & 0 \end{bmatrix}$$

$$\underset{\sim}{M}_4 = \int_{R_n} \underset{\sim}{\ddot{U}}^T \rho \underset{\sim}{U} dv = \begin{bmatrix} \int_{R_n} \underset{\sim}{\ddot{U}}_1^T \rho \underset{\sim}{U}_1 dv & \int_{R_n} \underset{\sim}{\ddot{U}}_1^T \rho \underset{\sim}{U}_2 dv \\ 0 & 0 \end{bmatrix}$$

Integrating by parts, one has

$$\int_{R_n} (\underset{\sim}{d}\underset{\sim}{U}_1)^T \underset{\sim}{A}(\underset{\sim}{d}\underset{\sim}{U}_1)dv = \int_{S_n} (\underset{\sim}{n}\underset{\sim}{U}_1)^T \underset{\sim}{A}(\underset{\sim}{d}\underset{\sim}{U}_1)ds - \int_{R_n} \underset{\sim}{U}_1^T(\underset{\sim}{d}^T\underset{\sim}{A}(\underset{\sim}{d}\underset{\sim}{U}_1))dv$$

Therefore

$$\underset{\sim}{H} + \underset{\sim}{M}_4^T = \underset{\sim}{H}_1 + \underset{\sim}{M}_5^T \tag{5.22}$$

with

$$\underset{\sim}{H}_1 = \begin{bmatrix} \int_{S_n} \underset{\sim}{U}_1^T \underset{\sim}{n}^T (\underset{\sim}{d}\underset{\sim}{U}_1)ds & 0 \\ 0 & 0 \end{bmatrix} \tag{5.23}$$

$$\underset{\sim}{M}_5 = \begin{bmatrix} 0 & \int_{R_n} \ddot{\underset{\sim}{U}}_1^T \rho \underset{\sim}{U}_2 dv \\ 0 & 0 \end{bmatrix}$$

in the above derivations use was made of the fact that $\dot{U}_2 = \ddot{U}_2 = 0$, see equation (3.35), and also

$$\rho\ddot{\underset{\sim}{U}}_1 = \underset{\sim}{d}^T \underset{\sim}{A}(\underset{\sim}{d}\underset{\sim}{U}_1)$$

since $\underset{\sim}{U}_1$ was chosen to satisfy the dynamic linear momentum equations of (3.5).

The matrix $\underset{\sim}{M}_5$ still contains the singularity, but since $\underset{\sim}{U}_2$ is a constant coefficient matrix, one is able to write

$$\int_{R_n} \ddot{\underset{\sim}{U}}_1^T \rho \underset{\sim}{U}_2 dv = \int_{R_n} (-c(t)\dot{\underset{\sim}{U}}_1^T \rho \underset{\sim}{U}_2)_{,x} dv = -c(t)\int_{S_n} \dot{\underset{\sim}{U}}_1^T \rho \underset{\sim}{U}_2 n_1 ds$$

Therefore

$$\underset{\sim}{M}_5 = \begin{bmatrix} 0 & -c(t)\int_{S_n} \dot{\underset{\sim}{U}}_1^T \rho \underset{\sim}{U}_2 ds \\ 0 & 0 \end{bmatrix} \tag{5.24}$$

The integral $\underset{\sim}{M}_3$ can be treated in a similar manner. Writing $\underset{\sim}{M}_3$ in the decomposed form one has

$$\underset{\sim}{M}_3 = \begin{bmatrix} \int_{R_n} \dot{\underset{\sim}{U}}_1^T \rho \underset{\sim}{U}_1 dv & \int_{R_n} \dot{\underset{\sim}{U}}_1^T \rho \underset{\sim}{U}_2 dv \\ 0 & 0 \end{bmatrix}$$

The only integral containing a singularity is $\int_{R_n} \dot{\underset{\sim}{U}}_1^T \rho \underset{\sim}{U}_2 dv$. Again since $\underset{\sim}{U}_2$ is a constant coefficient matrix, $\underset{\sim}{M}_3$ is replaced by

$$\underset{\sim}{M}_6 = \begin{bmatrix} \int_{R_n} \dot{\underset{\sim}{U}}_1^T \rho \underset{\sim}{U}_1 dv & -c(t) \int_{S_n} \underset{\sim}{U}_1^T \rho \underset{\sim}{U}_2 n_1 ds \\ 0 & 0 \end{bmatrix} \tag{5.25}$$

Also since the body forces were ignored, the singular element matrices of equation (5.21) change to

$$\underset{\sim}{K}_s = \underset{\sim}{B}^T \underset{\sim}{H}_1 \underset{\sim}{B} + \underset{\sim}{B}^T \underset{\sim}{M}_2 \ddot{\underset{\sim}{B}} + \underset{\sim}{2B}^T \underset{\sim}{M}_6^T \dot{\underset{\sim}{B}} + \underset{\sim}{B}^T \underset{\sim}{M}_5^T \underset{\sim}{B}$$

$$\underset{\sim}{V}_s = 2\underset{\sim}{B}^T (\underset{\sim}{M}_2 \dot{\underset{\sim}{B}} + \underset{\sim}{M}_6^T \underset{\sim}{B}) \tag{5.26}$$

$$\underset{\sim}{M}_s = \underset{\sim}{B}^T \underset{\sim}{M}_2 \underset{\sim}{B}$$

$$\underset{\sim}{F}_s = \underset{\sim}{T}$$

With the above modifications, it is now possible to perform all the necessary integrations and, by combining all the element matrices, the total mass matrix $\underset{\sim}{M}$, pseudo-damping matrix $\underset{\sim}{V}$ and the stiffness matrix $\underset{\sim}{K}$ of equation (5.17) can be evaluated. The solution procedure of the governing dynamic equation of motion (5.17) is discussed in Chapter 7.

# CHAPTER VI

## STRESS INTENSITY FACTORS

Stress intensity factors are parameters capable of characterizing the near-field displacements and stresses and enables one to make judgements concerning the behavior of crack under different load conditions, since they govern the onset of rapid crack extension. The stress intensity factors in plane problems are usually defined by removing the stress singularities as

$$K_1 = \lim_{r \to 0} \sqrt{2\pi r}\, \sigma_{22}\,(r,0) \tag{6.1}$$

$$K_2 = \lim_{r \to 0} \sqrt{2\pi r}\, \sigma_{12}(r,0)$$

where $K_1$ and $K_2$ are the opening mode and the sliding mode stress intensity factors respectively, and r is a distance directly ahead of the crack-tip with respect to the local coordinate system at the crack-tip.

The above definition is valid for problems with only one material, in which case normal loads will result in only the opening mode, and shear loads will result in only the sliding mode. In the bimaterial case, however, either normal or shear loads will result in both the opening mode and the sliding mode. Therefore the parameters $K_1$ and $K_2$ can no longer be regarded as the crack-tip stress intensity factors for symmetrical and skew-symmetrical stress distributions. The parameters $K_1$ and $K_2$ may in general be considered as the strength of the stress singularities at the crack-tip [4].

For static interface crack problems with two dissimilar isotropic materials, the stress intensity factors are defined [3] by removing the stress singularities as,

$$k_1 - ik_2 = 2\sqrt{2}\ e^{-\pi\varepsilon} \lim_{z_1 \to 0} z_1^{1/2-i\varepsilon}\ \Omega_{11}(z_1) \tag{6.2}$$

with $\Omega_{11}(z)$ and $\varepsilon$ as defined by equations (A.17) and (A.31) respectively. For this case they are given by

$$\Omega_{11}(z_1) = \sum_{\ell=1}^{m} \beta_\ell z_1^{n_\ell}$$

$$\varepsilon = -\frac{1}{2\pi}\log(q) \tag{6.3}$$

$$\text{with } q = \frac{\eta_1 + \mu_1/\mu_2}{(\mu_1/\mu_2)\eta_2 + 1}$$

where

$$\eta_k = \begin{cases} 3-4\nu_k & \text{for plane strain} \\ \dfrac{3-\nu_k}{1+\nu_k} & \text{for plane stress} \end{cases} \qquad k=1,2$$

and where $\mu_k$ and $\nu_k$ are the shear moduli and the Poisson's ratios of the two isotropic materials in the problem, $\beta_\ell$ are the unknown complex coefficients discussed in Appendix A and $n_\ell$ are given by equation (A.32).

Substituting equations (6.3) into equation (6.2), yields

$$k_1 - ik_2 = 2\sqrt{2}\ e^{-\pi\varepsilon} n_1\beta_1 \tag{6.4}$$

For this problem the first term of the stresses directly ahead of the crack-tip is given by

$$\sigma_{22} = \text{Real}\ [(1+q)\ n_1\beta_1\ r^{-\frac{1}{2}+i\varepsilon}] \tag{6.5}$$

$$\sigma_{12} = \text{Real}\ [i(1+q)\ n_1\beta_1 r^{-\frac{1}{2}+i\varepsilon}] = -\text{Im}[(1+q)n_1\beta_1 r^{-\frac{1}{2}+i\varepsilon}]$$

From equation (6.3), one obtains

$$q = e^{-2\pi\varepsilon} \qquad \text{and} \qquad (1+q) = 2\cosh(\pi\varepsilon)e^{-\pi\varepsilon}$$

so that equation (6.2) can be defined in an alternative way as

$$k_1 - ik_2 = \frac{\sqrt{2}}{\cosh(\pi\varepsilon)} \lim_{r\to 0} \left[ r^{\frac{1}{2}-i\varepsilon} (\sigma_{22} - i\sigma_{12}) \right] \tag{6.6}$$

Turning attention to the more general problem of dynamic interface cracks with two dissimilar anisotropic materials considered in this dissertation, it is noted that the first term of $\Omega_{11}(z)$ is given by equations (3.28), (A.17), and (A.33) as

$$\Omega_{11}(z) = z^{n_1}\beta_1 + \overline{F}_{211} z^{\overline{n}_1}\overline{\beta}_1 \tag{6.7}$$

where in general $\overline{F}_{211} \neq 0$.

Therefore, it is seen that the stress singularities can not be removed in the same manner as in equation (6.2), since for the general case both $n_1$ and $\overline{n}_1$ are involved, while in the special case of equation (6.2), $\overline{F}_{211}$ turned out to be zero and only $n_1$ was involved.

Therefore, a new definition is proposed here for the stress intensity factors as follows. Let the first term of the stresses directly ahead of the crack-tip be given by equation (A.36) as

$$\sigma_{22} = \text{Real } (A_2 r^{n_1-1}\beta_1) \tag{6.8}$$

$$\sigma_{12} = \text{Real } (A_3 r^{n_1-1}\beta_1)$$

where $A_2$ and $A_3$ are known complex constants from equations (A.36a) and (A.36b) as follows. For case (a) let $z_{1k}=z_{2k}=r$, $\ell=1$ and k=1 in equation (A.36a) to get

$$A_i=n_1[s_{i11}F_{111}+\bar{s}_{i11}F_{211}+s_{i21}F_{311}+\bar{s}_{i21}F_{411}] \qquad i=2,3$$

and for case (b) let $z_k=r$, $\ell=1$ and $k=1$ in equation (A.36b) to get

$$A_i=n_1[(s_{i11}F_{311}+s_{i21}F_{111})+(\bar{s}_{i11}F_{411}+\bar{s}_{i21}F_{211})$$

$$+s_{i11}(n_1-1)F_{111}+\bar{s}_{i11}(n_1-1)F_{211}] \qquad i=2,3$$

Now, we define

$$\begin{aligned} K_1 &= \sqrt{2\pi}\ \text{Real}\ (A_2\beta_1) \\ K_2 &= \sqrt{2\pi}\ \text{Real}\ (A_3\beta_1) \end{aligned} \tag{6.9}$$

It is readily seen that this definition reduces to the definition of equation (6.1) for the problems with one material, for which $\varepsilon = 0$.

For the static interface crack problem with two dissimilar isotropic materials, from equations (6.5) and (6.8) one has

$$A_2 = -iA_3 = (1+q)n_1 = 2\cosh(\pi\varepsilon)e^{-\pi\varepsilon}n_1 \tag{6.10}$$

Substituting equation (6.10) into (6.9) yields

$$K_1 - iK_2 = \sqrt{2\pi}\ \ 2\cosh(\pi\varepsilon)e^{-\pi\varepsilon}n_1\beta_1 \tag{6.11}$$

Comparing equation (6.11) and (6.4) one obtains

$$k_1 - ik_2 = \frac{K_1 - iK_2}{\sqrt{\pi}\ \cosh(\pi\varepsilon)}$$

so that definition (6.9) reduces to that of (6.2) to within a multiplicative factor of $\sqrt{\pi}\cosh(\pi\varepsilon)$. Thus we adopt the new definition of equation (6.9) for the stress intensity factors, which reduces to the previous definitions, to within a multiplicative factor.

CHAPTER VII

# SOLUTION PROCEDURE

The complete solution to the elastodynamic problem considered here is obtained by solving the discretized governing equation of motion (5.17).

There are two classes of methods for the direct integration of the equation of motion: explicit methods, in which accelerations are found from the equations of motion and then integrated to obtain displacements; and the implicit methods, in which the equations of motion are combined with the time integration operator so that displacements are found directly. Broadly speaking, implicit methods permit larger time steps, whereas explicit methods are restricted to small time steps by numerical stability requirements.

An implicit method of temporal integration which is developed from finite difference formulas and known as the Newmark-$\beta$ formulas [22] is used here. The Newmark-$\beta$ formulas can be written as

$$\dot{q}(t+\Delta t) = \dot{q}(t) + \frac{\Delta t}{2}[\ddot{q}(t) + \ddot{q}(t+\Delta t)] \tag{7.1}$$

$$q(t+\Delta t) = q(t) + \Delta t\,\dot{q}(t) + \Delta t^2[(\frac{1}{2} - \beta)\,\ddot{q}(t) + \beta\ddot{q}(t+\Delta t)]$$

In these formulas a value of $\beta$=1/4 corresponds to the assumption of a constant acceleration equal to the average of the accelerations at the ends of the interval between t and t+$\Delta$t. A value of $\beta$=1/6 corresponds to a piecewise linear acceleration within each time interval, and a value of $\beta$=1/8 corresponds to a step function with a uniform value equal to the initial value for the first half of the time interval and a uniform value equal to the final value for the second half of the time interval. A value of $\beta=0$ corresponds to double pulses of acceleration at the beginning and end of the time interval with each double pulse consisting of parts equal to 1/2 of the acceleration times the time interval, one occuring just before the end of the preceeding interval and the other just after the beginning of the next interval. It can also be shown that the Newmark-$\beta$ formulas of (7.1) with $\beta=0$ are equivalent to the central difference formulas

$$\dot{q}(t+\frac{\Delta t}{2}) = \dot{q}(t-\frac{\Delta t}{2}) + \Delta t\ddot{q}(t)$$

$$q(t+\Delta t) = q(t) + \Delta t\,\dot{q}(t+\frac{\Delta t}{2}) \tag{7.2a}$$

provided that we assume

$$\dot{q}(t+\frac{\Delta t}{2}) = \dot{q}(t) + \frac{\Delta t}{2}\ddot{q}(t) \tag{7.2b}$$

When used in an implicit method, the Newmark-$\beta$ formula of equation (7.1) yields unconditionally stable algorithms in linear problems for $\beta \geq 1/4$. The central difference formulas of equations (7.2), i.e. equations (7.1) with $\beta=0$, are only conditionally stable and give an explicit procedure.

To formulate an implicit method of temporal integration, it is assumed that $\beta>0$. Another aspect of the temporal integration methods which should be considered is their artifical damping, which may be considered as the tendency of the difference formulas to damp certain components of the response. The artificial damping may or may not be desirable depending on the desired solution [23]. The Newmark-$\beta$ formulas have no artifical damping.

Substituting equations (7.1) into equation (5.17) and dropping the "*" for convenience, yields

$$\underset{\sim}{K}_{eq}\underset{\sim}{q}(t+\Delta t) = \underset{\sim}{Q}_{eq} \tag{7.3}$$

where $\underset{\sim}{K}_{eq} = \underset{\sim}{M} + \frac{\Delta t}{2}\underset{\sim}{V} + \beta\Delta t^2\underset{\sim}{K}$

$$\underset{\sim}{Q}_{eq} = \underset{\sim}{M}\,[\underset{\sim}{q}(t) + \Delta t\dot{\underset{\sim}{q}}(t) + (\tfrac{1}{2} - \beta)\Delta t^2\ddot{\underset{\sim}{q}}(t)]$$
$$+ \underset{\sim}{V}[\tfrac{\Delta t}{2}\,\underset{\sim}{q}(t) + (\tfrac{1}{2} - \beta)\Delta t^2\dot{\underset{\sim}{q}}(t) + (\tfrac{1}{4} - \beta)\Delta t^3\ddot{\underset{\sim}{q}}(t)]$$
$$+ \beta\Delta t^2\underset{\sim}{Q}(t+\Delta t)$$

The solution to equation (5.17) is obtained by solving equation (7.3) at each time interval $\Delta t$. However, it is seen that $\underset{\sim}{K}_{eq}$ is not symmetric, since the matrices $\underset{\sim}{V}$ and $\underset{\sim}{K}$ in equation (5.17) are not symmetric as discussed earlier. In order to be able to use a procedure for the solution of a set of algebraic equations with symmetric coefficient matrix, equation (7.3) is modified in the following manner. Matrices $\underset{\sim}{V}$ and $\underset{\sim}{K}$ are divided into symmetric and asymmetric parts as

$$\underset{\sim}{V} = \frac{1}{2}\,(\underset{\sim}{V} + \underset{\sim}{V}^T) + \frac{1}{2}(\underset{\sim}{V} - \underset{\sim}{V}^T)$$

i.e.

$$\underset{\sim}{V} = \underset{\sim}{V}_{sym} + \underset{\sim}{V}_{asym}$$

with similar formulas for $\underset{\sim}{K}$. Substituting these expressions into equation (7.3) yields

$$\underset{\sim}{K}_{eff}\ \underset{\sim}{q}(t+\Delta t) = \underset{\sim}{Q}_{eff} \tag{7.4}$$

where
$$\underset{\sim}{K}_{eff} = \underset{\sim}{M} + \frac{\Delta t}{2}\underset{\sim}{V}_{sym} + \beta\Delta t^2\ \underset{\sim}{K}_{sym}$$

and
$$\underset{\sim}{Q}_{eff} = \underset{\sim}{M}[\underset{\sim}{q}(t) + \Delta t\dot{\underset{\sim}{q}}(t) + (\tfrac{1}{2} - \beta)\Delta t^2\ddot{\underset{\sim}{q}}(t)]$$
$$+ \underset{\sim}{V}_{sym}[\frac{\Delta t}{2}\underset{\sim}{q}(t) + (\tfrac{1}{2} - \beta)\Delta t^2\dot{\underset{\sim}{q}}(t) + (\tfrac{1}{4} - \beta)\Delta t^3\ddot{\underset{\sim}{q}}(t)]$$
$$+ \beta\Delta t^2[\underset{\sim}{Q}(t + \Delta t) - \underset{\sim}{V}_{asym}\dot{\underset{\sim}{q}}(t + \Delta t) - \underset{\sim}{K}_{asym}\underset{\sim}{q}(t + \Delta t)]$$

It is seen that the evaluation of $\underset{\sim}{Q}_{eff}$ at $t+\Delta t$ requires a knowledge of $\underset{\sim}{q}(t+\Delta t)$ and $\dot{\underset{\sim}{q}}(t+\Delta t)$, which we are about to solve. Therefore an iterative procedure is employed as follows. In the the expression for $\underset{\sim}{Q}_{eff}$, change $\dot{\underset{\sim}{q}}(t+\Delta t)$ and $\underset{\sim}{q}(t+\Delta t)$ to $\underset{\sim}{q}(t)$ and $\dot{\underset{\sim}{q}}(t)$ respectively, solve for $\underset{\sim}{q}(t+\Delta t)$ from equation (7.4) and for $\dot{\underset{\sim}{q}}(t+\Delta t)$ from equation (7.1), then use these values of $\underset{\sim}{q}(t+\Delta t)$ and $\dot{\underset{\sim}{q}}(t+\Delta t)$ in the expression for $\underset{\sim}{Q}_{eff}$ to find a new set of $\underset{\sim}{q}(t+\Delta t)$ and $\dot{\underset{\sim}{q}}(t+\Delta t)$. Repeat this process until some convergence criteria has been met. Therefore equation (7.4) can be rewritten as

$$\underset{\sim}{K}_{eff}\ \underset{\sim}{q}^{(p)}_{(t+\Delta t)} = \underset{\sim}{Q}^{(p-1)}_{eff} \tag{7.5}$$

for any $p^{th}$ iteration.

However, since the asymmetry of $\underset{\sim}{K}$ relates to the small part of matrix $\underset{\sim}{K}$ which corresponds to the singular element, and the matrix $\underset{\sim}{V}$ corresponds only to the singular element, therefore, a fast convergence for equation (7.5) is to be expected. In fact experimentation by this author indicated that only one iteration was adequate for convergence and subsequent iterations hardly improved the solution.

It remains now to devise a method of crack-propagation in the finite element mesh for dynamic problems with crack extensions. A method for crack propagation can be described as follows. The singular element is placed in the finite element mesh such that the initial crack-tip position is situated inside the singular element. Then the crack-tip is advanced at each time step according to the prescribed crack-tip position history. When the crack-tip reaches an extreme forward position, i.e. the 3/4 point position, inside the singular element, a local remeshing takes place and the position of the singular element is moved forward, i.e. in the direction of crack propagation, by half the size of the singular element. Thus the crack-tip is located at the 1/4 point position of the singular element and the crack-tip can continue to extend inside the singular element (Fig. 7.1). Therefore, two elements that were regular at the previous time step turn into a part of the new singular element and half of the region of the previous singular element is replaced with two new regular elements.

As shown in Fig. 7.1, this procedure requires a certain degree of regularity in the finite element mesh. In fact, the finite element program requires the singular element to be a square composed of four square regular elements and for the propagating cracks it is required that the elements adjacent to the interface be squares for the distance required for remeshing as long as the crack is extending. Therefore as long as remeshing takes place, the singular element is a square composed of four square regular elements. These restrictions

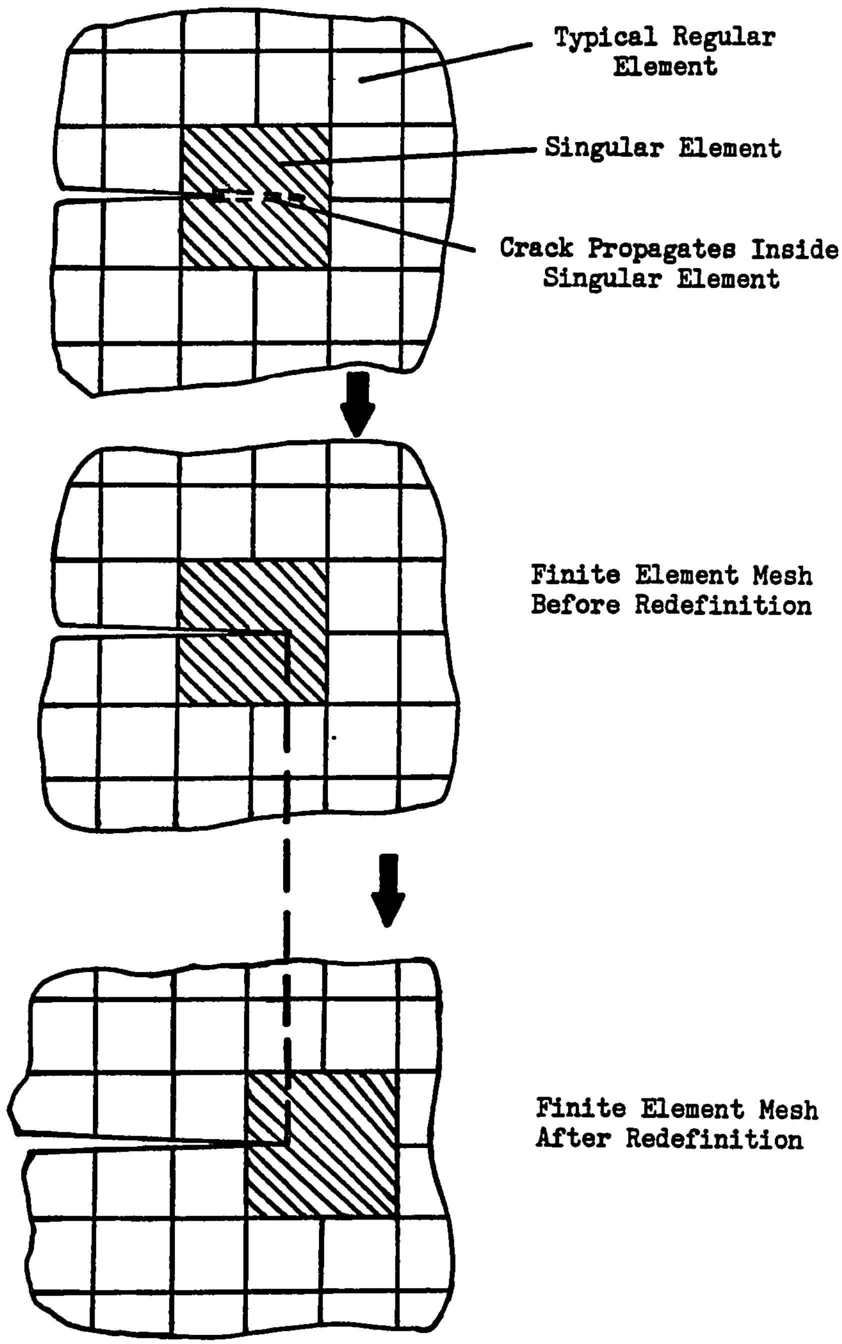



Fig. 7.1 Finite Element Mesh Redefinition Procedure

are not the results of the procedure developed here, but are imposed for programming ease and economic reasons.

As it is seen from Fig. 7.1, in the process of remeshing new nodes have to be created in the finite element mesh. A method of double noding proposed by B.M. Liaw, A.S. Kobayashi and A.F. Emery [17] and also Santosh K. Arya [18] was employed to create new nodes. In this technique the mesh-points along the crack-tip, i.e. the mesh points on the interface, have two nodes, so that when remeshing takes place the necessary extra node is available. Although the center of the singular element is not associated with any node, nevertheless, there have to be two imaginary nodes at that point. These nodes are, of course eliminated in the calculation process in order to make $\underset{\sim}{K}_{eff}$ non-singular. Also one of the nodes on all the mesh points ahead of the crack-tip which are associated with two nodes are eliminated by using the equality of the displacements of the two nodes at such points, to ensure that the final $\underset{\sim}{K}_{eff}$ is not singular and thus invertible.

This procedure is implemented as follows. Assume that $\underset{\sim}{K}_{eff}$ and $\underset{\sim}{Q}_{eff}$ in equation (7.5) are formed. Then for the mesh points ahead of the crack-tip which are assiciated with two nodes, the horizontal and vertical degrees of freedom of these two nodes have to be equal. Now, assuming that the $i^{th}$ and $j^{th}$ degrees of freedom are constrained to be equal, i.e.

$$q_i = q_j \tag{7.6}$$

define the relative displacement vector $\underset{\sim}{q}'$ as

$$\underset{\sim}{q} = \underset{\sim}{T}\,\underset{\sim}{q}' \tag{7.7}$$

and

$$\underset{\sim}{Q}'_{eff} = \underset{\sim}{T}^T \underset{\sim}{Q}_{eff} \tag{7.8}$$

where

$$T_{mn} = \begin{cases} 1 & \text{when } m=n \\ 1 & \text{when } m=i \text{ and } n=j \\ 0 & \text{for all other } m \text{ and } n \end{cases}$$

and

$$q'_k = \begin{cases} q_i - q_j = 0 & \text{when } k=i \\ q_k & \text{when } k \neq i \end{cases}$$

Substituting equations (7.7) and (7.8) into equation (7.5) yields

$$\begin{aligned} \underset{\sim}{K}'_{eff}\, \underset{\sim}{q}' &= \underset{\sim}{Q}'_{eff} \\ &= \underset{\sim}{T}^T \underset{\sim}{Q}_{eff} \\ &= \underset{\sim}{T}^T \underset{\sim}{K}_{eff} \underset{\sim}{q} \\ &= \underset{\sim}{T}^T \underset{\sim}{K}_{eff} \underset{\sim}{T} \underset{\sim}{q}' \end{aligned} \tag{7.9}$$

The above equation requires that

$$\underset{\sim}{K}'_{eff} = \underset{\sim}{T}^T \underset{\sim}{K}_{eff} \underset{\sim}{T} \tag{7.10}$$

Therefore equation (7.9) is solved instead of equation (7.5) and then the displacements $\underset{\sim}{q}$ are found from equation (7.7).

A computer program was developed that carries out the procedure described in this dissertation and solves for the displacements $\underset{\sim}{q}$. The velocities and accelerations $\dot{\underset{\sim}{q}}$ and $\ddot{\underset{\sim}{q}}$ are found from the difference equations (7.1). The stresses at the center of regular elements are found according to equation (4.14), and the stresses at four points of the singular element corresponding to the center points of the regular elements which form the singular element are found according to equation (3.34). The finite element mesh definition, initial conditions, boundary conditions, external load history, crack-tip velocity history, crack-tip position history and the material properties are provided as input. The program obtains the eigenvalues and eigenfunctions for the singular element and evaluates all the regular and singular element matrices and stores them into global matrices $\underset{\sim}{K}$, $\underset{\sim}{V}$ and $\underset{\sim}{M}$ of equations (5.17). Then, $\underset{\sim}{K}_{eff}$ and $\underset{\sim}{Q}_{eff}$ are formed from which $\underset{\sim}{K}'_{eff}$ and $\underset{\sim}{Q}'_{eff}$ are obtained. Then using the iterative procedure described earlier, the solution for $\underset{\sim}{q}$, and hence the complete solution, including the stress intensity factors as defined by equation (6.9) are obtained. A more detailed description of the computer program and its complete listing are presented in the Appendices.

In the next chapter solutions to some simple problems are presented. These results are compared with known theoretical and numerical results. Finally the solutions to some stationary and propagating interface cracks between two dissimilar anisotropic materials under static and dynamic loads, for which there are no known solutions, are presented.

# CHAPTER VIII

## WAVE PROPAGATION IN ANISOTROPIC MEDIUM

When a structure is subjected to dynamic loads such as impacts, generated waves travel through it. It is necessary to understand the response characteristics of the material body and the way the generated waves propagate through the material.

It is well known that for isotropic materials there are two characteristic wave speeds [24], $C_T$ and $C_L$, where $C_T = \sqrt{G/\rho}$ is the transverse (also called shear or distortional) wave speed, and $C_L = \sqrt{(\lambda+2G)/\rho}$ is the longitudinal or dilatational wave speed. In the above $\lambda$ and G are Lame's constants and $\rho$ is the mass density of the material. For isotropic materials, the waves propagate through the medium with the same speeds in all directions. This is obviously due to the fact that isotropic material properties are independent of direction.

For anisotropic materials, which exhibit different properties in different directions, the wave propagation is directional, i.e., the wave speeds are a function of direction as well as material properties.

The displacement form for a plane harmonic wave can be written as [25]

$$u_n = C_n \exp[i\omega(\frac{n_m X_m}{v} - t)] \qquad n,m = 1,2,3 \qquad (8.1)$$

where $\omega$ is the real circular frequency, $n_m$ are the elements of a unit vector representing the direction of propagation, $C_n$ are the wave amplitudes, and v is the velocity of wave propagation.

For a plane problem substituting equation (8.1) into the strain-displacement equations (2.1), the constitutive equations (2.12) and the equilibrium equations (2.4) and ignoring the body forces, one obtains

$$\begin{bmatrix} A_{11}n_1^2+2A_{16}n_1n_2+A_{66}n_2^2-Q & A_{16}n_1^2+(A_{12}+A_{66})n_1n_2+A_{26}n_2^2 \\ A_{16}n_1^2+(A_{12}+A_{66})n_1n_2+A_{26}n_2^2 & A_{66}n_1^2+2A_{26}n_1n_2+A_{22}n_2^2-Q \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = 0 \tag{8.2}$$

where $Q=\rho v^2$, $A_{ij}$ are the elements of the matrix $\underset{\sim}{A}$ in the constitutive equation $\underset{\sim}{\sigma}=\underset{\approx}{A}\underset{\sim}{\varepsilon}$ and $C_1$ and $C_2$ are the wave amplitudes for $u_1$ and $u_2$ in equation (8.1).

For non-trivial solutions of $C_1$ and $C_2$, the determinant of the matrix in equation (8.2) should vanish, which gives

$$\begin{aligned} &Q^2-Q[A_{11}n_1^2+2(A_{16}+A_{26})n_1n_2+A_{22}n_2^2+A_{66}] \\ &\quad +[(A_{11}A_{66}-A_{16}^2)n_1^4+2(A_{11}A_{26}-A_{16}A_{12})n_1^3n_2+(A_{11}A_{22}+2A_{16}A_{26}- \\ &\quad A_{12}^2-2A_{12}A_{66})n_1^2n_2^2+2(A_{16}A_{22}-A_{26}A_{12})n_1n_2^3+(A_{22}A_{66}-A_{26}^2)n_2^4]=0 \end{aligned} \tag{8.3}$$

For any given direction $n_1$ and $n_2$ equation (8.3) gives two roots for Q which in turn gives two values for the wave speed v. These two values correspond to the shear wave and longitudinal wave speeds in the specified direction for the material. For example taking the $X_1$ direction one obtains

$$v_1^2 = \frac{A_{11}+A_{66}+\sqrt{(A_{11}-A_{66})^2+4A_{16}^2}}{2\rho}$$

$$v_2^2 = \frac{A_{11}+A_{66}-\sqrt{(A_{11}-A_{66})^2+4A_{16}^2}}{2\rho}$$

and in the $X_2$ direction one obtains

$$v_1^2 = \frac{A_{22} + A_{66} + \sqrt{(A_{22} - A_{66})^2 + 4A_{26}^2}}{2\rho}$$

$$v_2^2 = \frac{A_{22} + A_{66} - \sqrt{(A_{22} - A_{66})^2 + 4A_{26}^2}}{2\rho}$$

where $v_1$ and $v_2$ are the longitudinal and shear wave speeds in the indicated directions, respectively.

To find the directions in which the wave speeds are minimum or maximum, equation (8.2) should be differentiated with respect to the angle $\theta$ from $X_1$ direction and $Q' = \frac{dQ}{d\theta}$ be set to zero. Recalling that

$$n_1 = \cos\theta$$
$$n_2 = \sin\theta$$

one obtains

$$n_1' = -n_2$$
$$n_2' = n_1$$

The differentiation of equation (8.2) yields

$$\begin{aligned} &Q[A_{11}n_1n_2-(A_{16}+A_{26})(n_1^2-n_2^2)] \\ &+[-2(A_{11}A_{66}-A_{12}^2)n_1^3n_2+(A_{11}A_{26}-A_{16}A_{12})(n_1^4-3n_1^2n_2^2) \\ &+(A_{11}A_{22}+2A_{16}A_{26}-A_{12}^2-2A_{12}A_{66})(n_1^3n_2-n_1n_2^3) \\ &+(A_{16}A_{22}-A_{26}A_{12})(3n_1^2n_2^2-n_2^4)+2(A_{22}A_{66}-A_{26}^2)n_2^3n_1]=0 \end{aligned} \quad (8.4)$$

The simultaneous solution of equations (8.3) and (8.4) gives the minimum or maximum values of Q and the corresponding directions. Obviously the solution has to be obtained by numerical means. To see whether a value of the wave speed found corresponds to a minimum or a maximum, one could twice differentiate equation (8.3) and then set $Q'=0$ and check for the sign of $Q''$, or more simply the values of $Q$ can be evaluated from equation (8.3) for slightly different directions to determine whether the wave speed corresponds to a minimum or a maximum in that direction.

It is noted that if the material is orthotropic, i.e. $A_{16}=A_{26}=0$, equations (8.3) and (8.4) are considerably simplified. Equation (8.3) gives

$$Q^2-Q[A_{11}-A_{22})n_1^2+(A_{22}+A_{66})] \quad (8.5)$$
$$+[(A_{11}A_{66}n_1^4)+(A_{11}A_{22}-A_{12}^2-2A_{12}A_{66})(n_1^2-n_1^4)+(A_{22}A_{66})(1-n_1^2)^2]=0$$

and equation (8.4) gives

$$Q[A_{11}]+[-2(A_{11}A_{66}-A_{12}^2)n_1^2+(A_{11}A_{22}-A_{12}^2-2A_{12}A_{66})(2n_1^2-1) \quad (8.6)$$
$$+2(A_{22}A_{66}-A_{26}^2)(1-n_1^2)]=0$$

and

$$n_1n_2=0 \quad (8.7)$$

Here, use was made of the fact that $n_1^2+n_2^2=1$.

Equation (8.7) states that for orthotropic materials the wave propagation speeds in $X_1$ and $X_2$ directions correspond to minima or maxima. The results from equation (8.5) are

$$v_1 = \sqrt{\frac{A_{11}}{\rho}}$$

$$v_2 = \sqrt{\frac{A_{66}}{\rho}}$$

in the $X_1$ direction and

$$v_1 = \sqrt{\frac{A_{22}}{\rho}}$$

$$v_2 = \sqrt{\frac{A_{66}}{\rho}}$$

in the $X_2$ direction.

The $v_1$'s correspond to longitudinal wave speeds and $v_2$'s correspond to transverse wave speeds in the indicated directions.

For convenience define

$$f=A_{11}-A_{22}$$

$$g=A_{22}+A_{66}$$

$$k=-A_{11}A_{22}+A_{12}^2+(2A_{12}+A_{11}+A_{22})A_{66}$$

$$m=A_{11}A_{22}-A_{12}^2-2(A_{12}+A_{22})A_{66}$$

$$s=A_{22}A_{66}$$

Then, substituting the above into equations (8.5) and (8.6) gives

$$Q^2-Q[fn_1^2+g]+(kn_1^4 + mn_1^2+s)=0 \tag{8.8a}$$

and

$$Qf=2kn_1^2+m \tag{8.8b}$$

Substituting equation (8.8b) into equation (8.8a) yields

$$n_1^4[k(4k-f^2)]+2n_1^2[k(2m-gf)]+[m^2-mfg+sf^2]=0 \tag{8.9}$$

For a given set of material properties the solution to equation (8.9) is easily obtained. This solution gives the directions in which the wave speeds are minimum or maximum. Substituting these directions into equation (8.8b) the magnitude of the wave speeds in the respective directions are obtained.

In the case of isotropic materials, one has

$$A_{22}=A_{11}$$
$$A_{12}=A_{11}-2A_{66} \tag{8.10}$$

Substituting the above into equation (8.5) yields

$$Q^2-Q(A_{11}+A_{66})+A_{11}A_{66}=0 \tag{8.11}$$

and equation (8.8b) is identically satisfied as expected.

The solution to equation (8.11) gives

$$v_1 = \sqrt{\frac{A_{11}}{\rho}}$$
$$v_2 = \sqrt{\frac{A_{66}}{\rho}} \tag{8.12}$$

As it is seen equation (8.11) does not contain the directional parameters $n_1$ and $n_2$, which indicates the fact that wave propagation in isotropic materials is independent of direction. Equations (8.12) represent the solution to equation (8.11), in which $v_1$ is the longitudinal wave speed and $v_2$ is the transverse wave speeds. These equations are the same as the ones discussed at the beginning of this chapter with $A_{11}=\lambda+2G$ and $A_{66}=G$.

# CHAPTER IX

# RESULTS

The correctness of the singular element stiffness, psuedo-damping, and mass matrices were evaluated by calculating the corresponding forces due to a unit rigid body displacement, velocity, and acceleration of the singular element respectively. This was done by addition of all the elements of the matrices corresponding to horizontal and vertical forces to produce the above unit rigid body motion in the horizontal and vertical directions. From Newton's law of motion the sum of the forces applied to the nodes of the element should be equal to zero in each direction with a zero net moment about the center of mass for rigid body displacements and velocities. Furthermore the sum of the forces applied to the nodes of the element in each direction should be equal to the mass of the element with a zero net moment about the center of mass for a unit rigid body acceleration in the respective direction.

All the above conditions were satisfied with at least six digits of accuracy. The above test was conducted for several combinations of different materials and crack-tip speeds.

## 9.1 Static problems

As a first test, the problem of a stationary center crack in a homogeneous isotropic plate is solved. Fig. 9.1 shows the finite element mesh for one half of the plate.

Six Gaussian integration points and the first fifteen terms of the singular element eigenfunctions were used. Computed results of stress intensity factor $K_1$ for various crack length to width ratios a/b are plotted in Fig. 9.2. These results are compared with a solution by Isida [28] using a boundary collocation method. It is seen that the results of the present finite element solution are in good agreement with Isida's results with a maximum difference of less than 2% for small values of a/b. The mesh shown in Fig. 9.1 is for a/b = .1. For other values of a/b the singular element should be moved to proper positions to incorporate the crack-tip. It was also noted here that the results for the same crack-length but different location inside the singular element differed with a maximum of 1.5% for extreme positions. This is also shown in Fig. 9.2. A state of plane strain was assumed and the material properties $G=2.94\ 10^{10} N/m^2$, $\nu$ = .292 were used.

As a second test, the problem of a center crack in a homogeneous orthotropic plate is solved. It is assumed that the elastic axes for $E_1$ and $E_2$ coincide with X and Y axes, respectively. Fig. 9.3 shows the finite element mesh for one half of the plate. Five Gaussian integration points and the first thirteen terms of the eigenfunctions were employed. The results for the stress intensity factor $K_1$ for various crack-length to width ratios a/b are plotted in Fig. 9.4. These results are compared with a solution by Bowie [29] obtained through a mapping-collocation method and also with a solution by K.Y. Lin and Pin Tong [30] using a finite element method with a 17 noded singular element. It is seen that the present results are in good

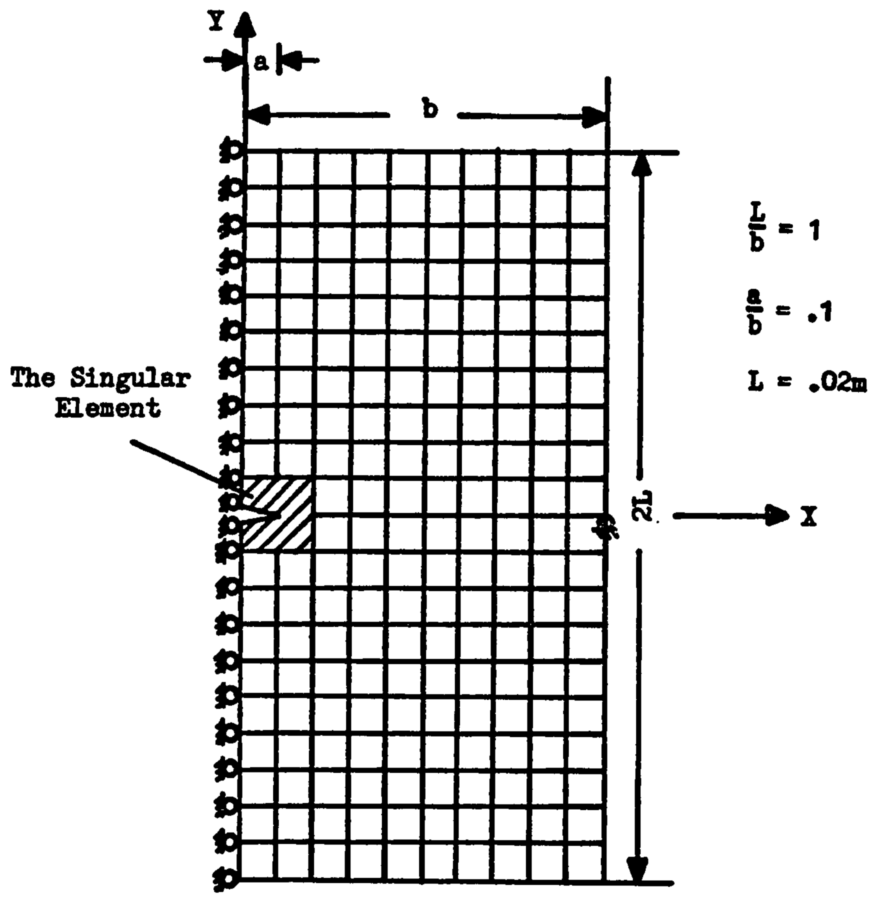



Fig. 9.1 Finite Element Mesh for Center Crack in Isotropic Medium

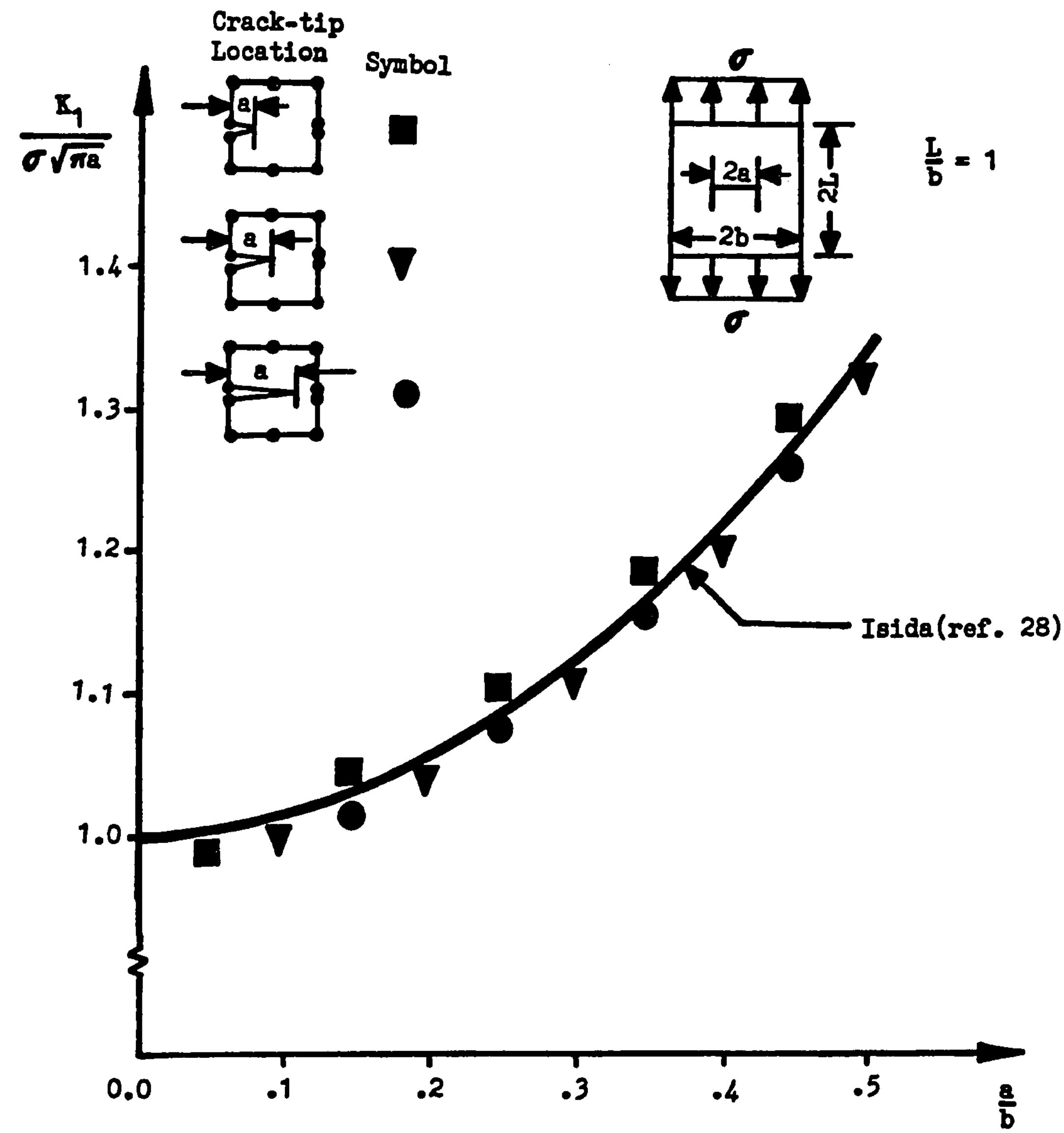



Fig. 9.2 Stress Intensity Factor for Center Crack Tension Plate in Isotropic Medium

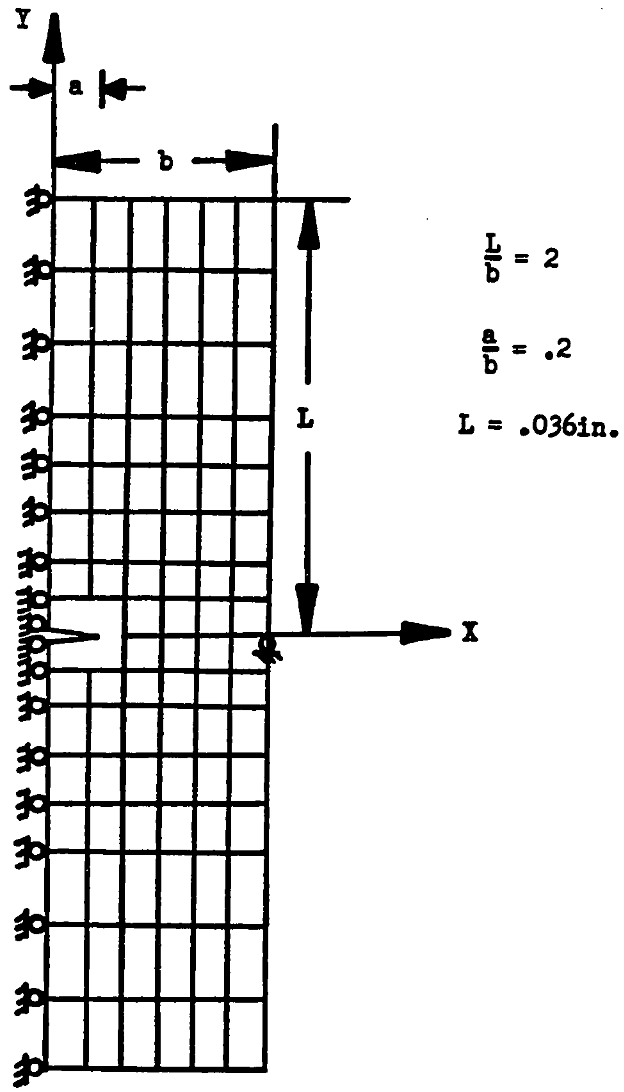



Fig. 9.3 Finite Element Mesh for Center Crack in Orthotropic Medium

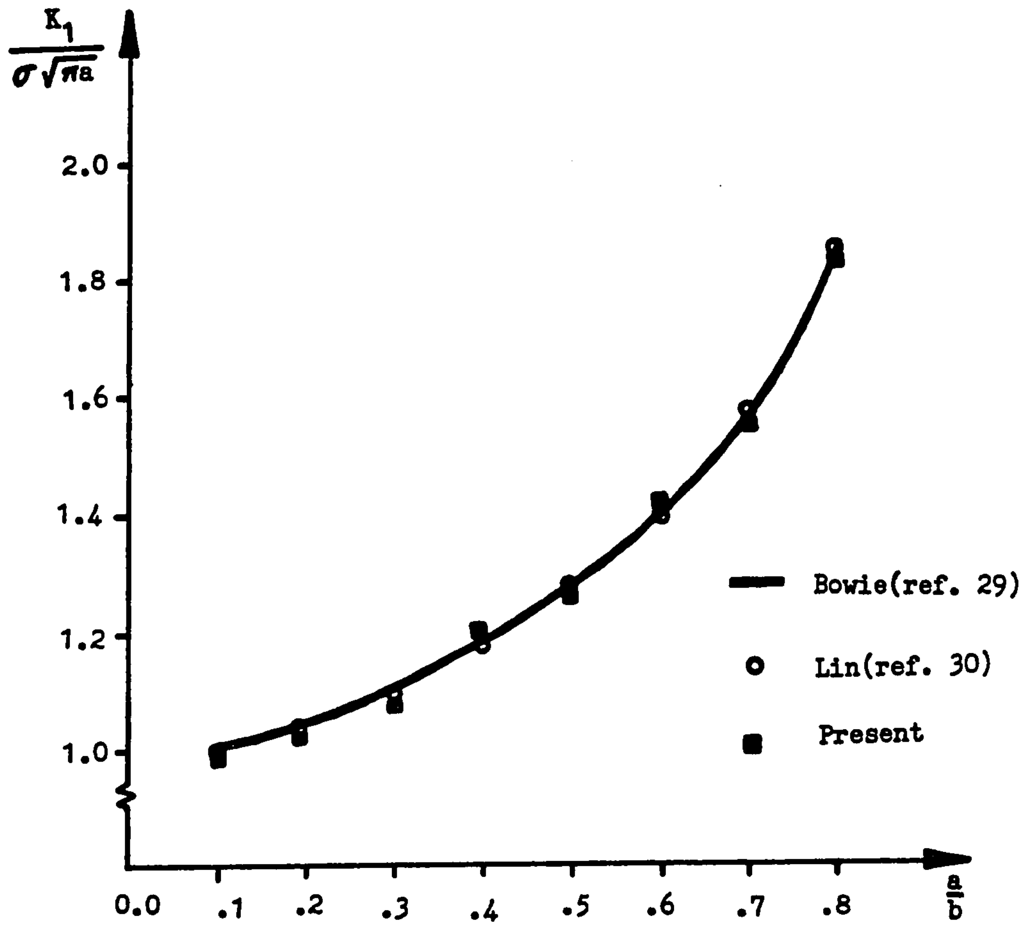



Fig. 9.4 Stress Intensity Factor for Center Crack in Orthotropic Medium

agreement with both solutions with a maximum difference of less than 2%. A state of plane stress was assumed and the following material properties $E_1 = 1$, $E_2 = 10$, $\nu_{21} = .21$, $G_{12} = .8757$ were used.

As another example of a static problem, the problem of a centrally cracked plate along the interface of two dissimilar isotropic materials is solved. A state of plane stress is assumed. Fig. 9.5 shows the finite element mesh for one half of the plate. Six Gaussian integration points and eighteen terms of the singular element eigenfunctions were used. Computed results of stress intensity factors $K_1$ and $K_2$ are listed in Table 9.1. These results are compared with classical exact solutions by Sih [4] and also with the K.Y. Lin and J.W. Mar [5] finite element method with a 17 noded singular element. As discussed in Chapter 6, the results shown in Table 9.1 are scaled by a factor of $\sqrt{\pi}\cosh(\pi\varepsilon)$ for comparison with the results of Sih [4] and Lin [5].

| $\frac{\mu_1}{\mu_2}$ | $\nu_2$ | $\sigma_{xx2}$ | $k_1$ | | | $k_2$ | | |
|---|---|---|---|---|---|---|---|---|
| | | | Lin | exact | present | Lin | exact | present |
| 1 | .3 | 1.00 | 1.01 | 1.0 | .99 | 0.0 | 0.0 | 0.0 |
| 3 | .3 | .53 | 1.0 | .99 | .98 | .073 | .072 | .078 |
| 10 | .3 | .37 | .98 | .97 | .96 | .118 | .117 | .126 |
| 23.1 | .35 | .38 | .97 | .97 | .95 | .121 | .121 | .132 |
| 100 | .3 | .31 | .96 | .95 | .94 | .140 | .139 | .151 |
| 144.2 | .35 | .36 | .97 | .96 | .95 | .131 | .130 | .143 |
| 1000 | .3 | .30 | .96 | .95 | .94 | .142 | .142 | .153 |
| $\infty$ | .3 | .30 | -- | .95 | .94 | -- | .142 | .153 |

Table 9.1 Stress intensity factors $k_1$ and $k_2$ for a crack along the interface of two dissimilar isotropic materials. $E_1$ = 1psi, $\nu_1$ = .3, $\sigma_{yy}$= 1psi, $\sigma_{xx_1}$= 1psi, 20"x20" plate, crack-length = 2", plane-stress.

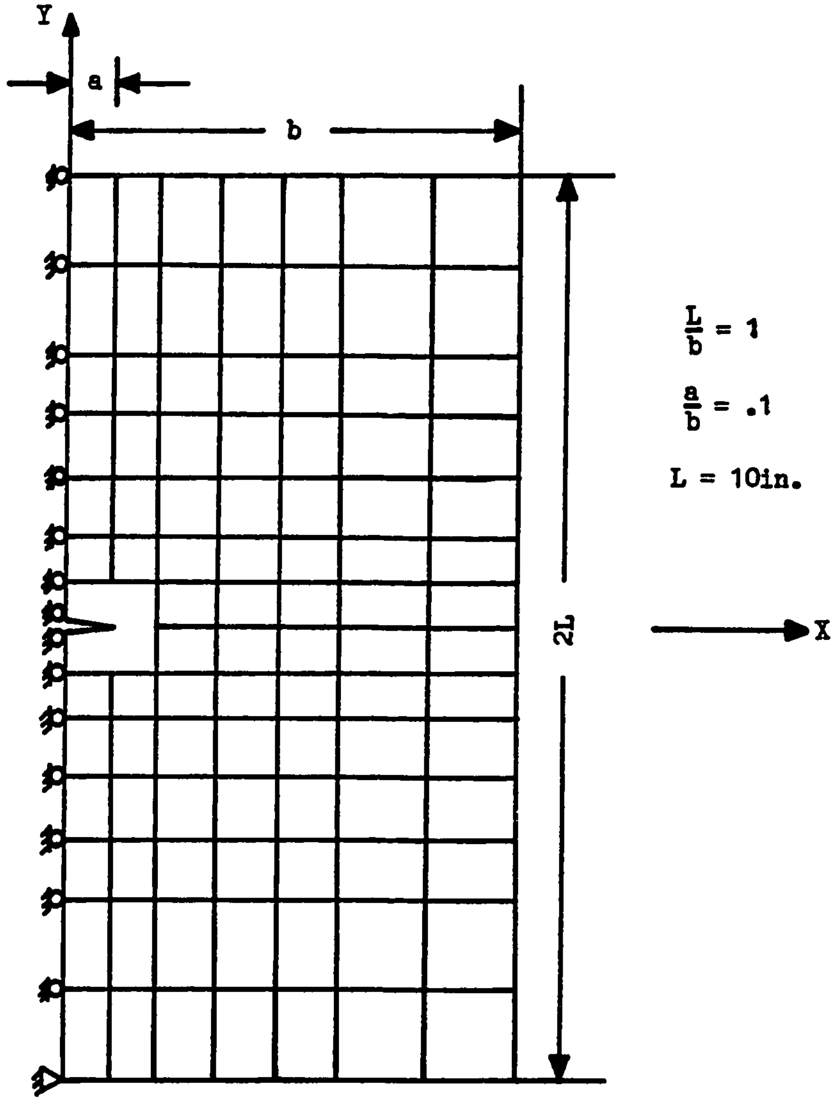
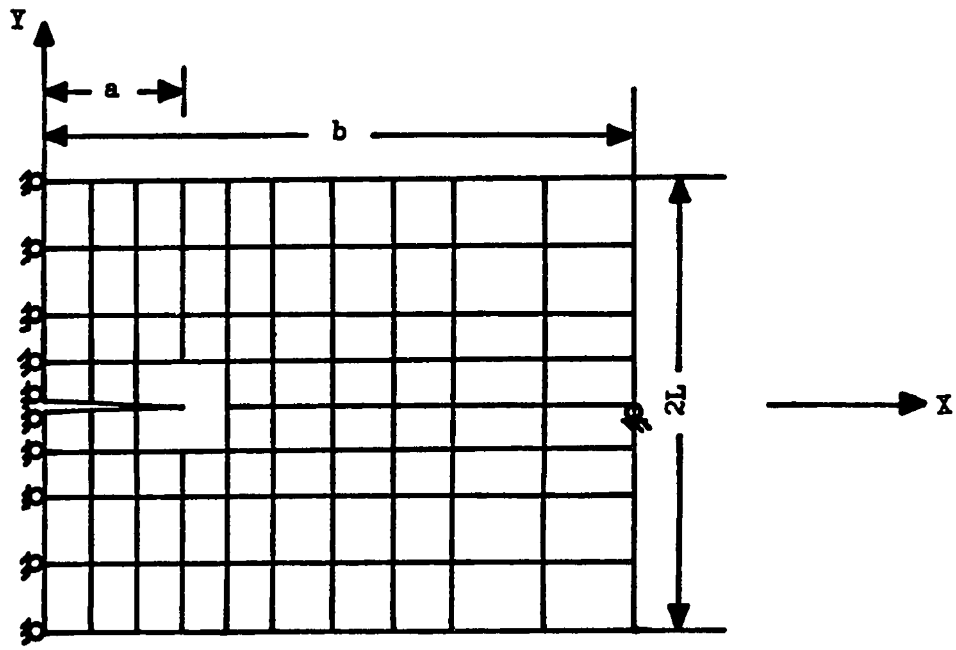



Fig. 9.5 Finite Element Mesh for Center Crack at the Interface of Two Dissimilar Isotropic Materials

From table 9.1 is is seen that the results for $k_1$ are in good agreement, but the error in the results for $k_2$ are somewhat greater. This is due to the fact that the singular element used by [5] is a 17 noded element, while the singular element used presantly is in fact a 9 noded element. A 17 noded singular element was not used here because it would have required a large core storage and been uneconomical for this general problem. However, one should note that the values of $k_2$ are small compared to the values of $k_1$. Therefore, this inaccuracy is not important.

## 9.2 Stationary cracks under impact

As a first test of the accuracy of the procedure for dynamic problems, the problem of a rectangular plate with a centrally located crack in a homogeneous isotropic plate is solved. The material properties are taken as: $G = 2.94 \times 10^{11}$ dyne/cm$^2$; $\nu = .286$; and $\rho = 2.45$ g/cm$^3$. A state of plane strain is assumed. Uniformly distributed uniaxial tensile stresses, with a Heaviside step-function time dependence were assumed to act at the edges of the plate parallel to the crack-axis. The crack is assumed to be stationary under the action of the applied load. Fig. 9.6 shows the finite element mesh for one half of the plate. For this problem a value of $\beta = 1/4$ and a value of time step $\Delta t = 1.0 \times 10^{-6}$ sec. were used. This time step allows the longitudinal waves to travel 63.3% of the smallest dimension in the mesh. Five Gaussian integration points and the first eighteen terms of the singular element eigenfunctions were used. The computed stress intensity factor $K_1$ is plotted in Fig. 9.7.




L = 5cm

b = 13cm

a = 3cm

Fig. 9.6 Finite Element Mesh for Impact Loaded Homogeneous Isotropic Plate With Stationary Center Crack and for Impact Loaded Plate With Stationary Center Crack Along the Interface of Two Dissimilar Orthotropic Materials

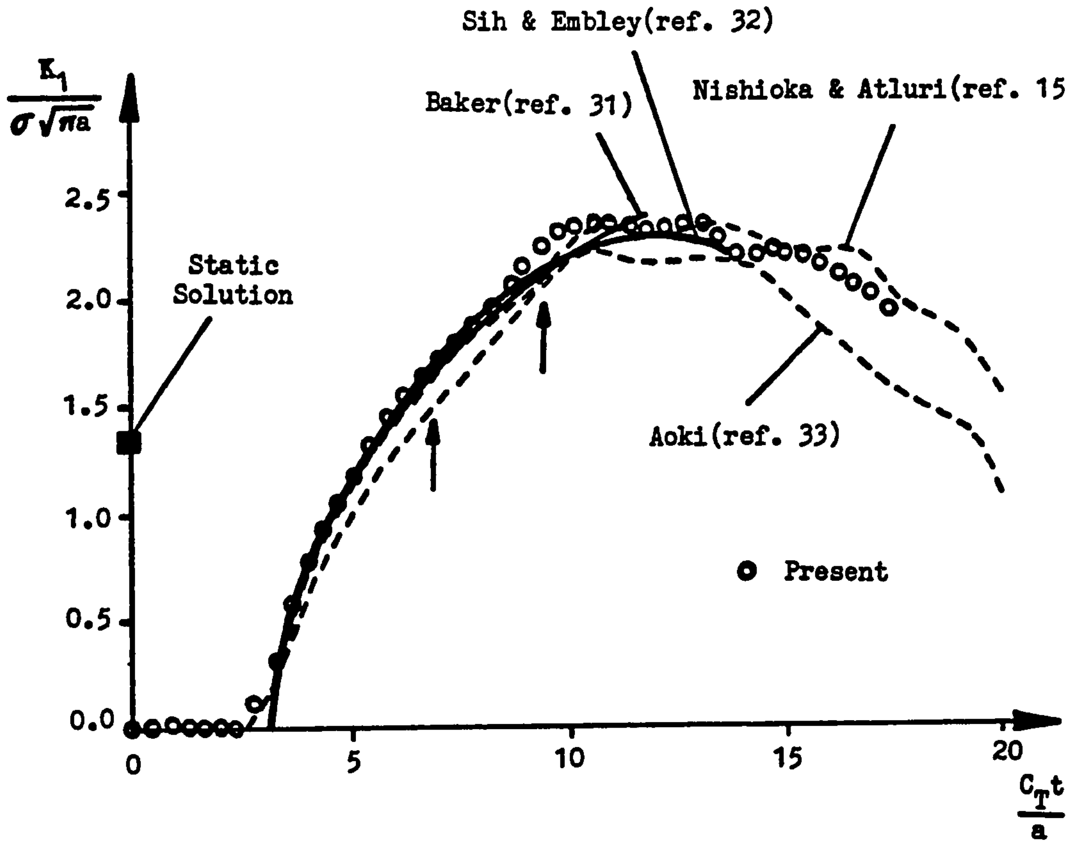



Fig. 9.7 Stress Intensity Factor for Impact Loaded Homogeneous Isotropic Plate With Stationary Center Crack

The problem of a semi-infinite crack subjected to a sudden impact at the crack surface was solved by B.R. Baker [31], and the problem of a finite size crack in an infinite medium subjected to a sudden impact loading at the crack surface was solved by G.C. Sih and G.T. Embley [32]. Transform methods, such as the Wiener-Hopf and Cagniard methods were used to obtain the solutions.

The solution given by Baker is valid from the time that the longitudinal wave created by the impact reaches the crack-tip to the time that the longitudinal wave created at one crack-tip reaches the other. The solution by Sih is valid up to the time that the longitudinal wave travels the length of the plate and reflects from the loaded edges of the plate and reaches the crack axis. The results of Baker and Sih are also plotted in Fig. 9.7 and it is seen that the present results are in good agreement with both solutions for valid times. These times are shown by vertical arrows in Fig. 9.7. Also shown in fig. 9.7 are the numerical results by T. Nishioka and S.N. Atluri [15], and S. Aoki [33]. It is seen that the present results are closer to those of Nishioka [15], and higher than the results by Aoki [33]. However, the solution by Aoki, et. al., appears to be lower than the theoretical results by Baker [31] and Sih [32]. It is of interest to note that the numerical results plotted in Fig. 9.7 show a non-zero stress intensity factor even before the time that the longitudinal waves arrive from the loaded boundary to the crack axis, as computed from the material wave speeds. This is attributed partly to the fact that in the finite element formulation the crack-tip is not associated with a node, and when the first longitudinal wave reaches the nodes of the singular element, the formulation will give rise to some value for the stress intensity factor. In addition, Atluri [15] suggests that the use of the consistent mass matrix may contribute to this problem.

Next, the problem of a rectangular plate with a centrally located crack at the interface of two dissimilar orthotropic materials under impact loading is solved. The composite media is assumed to be made of two dissimilar unidirectional fiber reinforced graphite-epoxy composite materials. The fiber directions of the upper and lower media are and parallel to Z and X axes, i.e., $\theta = 0°$, and $\theta = 90°$, respectively, see Fig. 9.8. The following material properties of graphite fiber-epoxy composites [27] are used:

$$\rho = 7.44 \text{ g/cm}^3$$

$$E_1 = 1.378 \times 10^{12} \text{ dyne/cm}^2$$
$$E_2 = E_3 = .14469 \times 10^{12} \text{ dyne/cm}^2$$
$$G_{12} = G_{13} = G_{23} = .058565 \times 10^{12} \text{ dyne/cm}^2$$
$$\nu_{12} = \nu_{13} = \nu_{23} = .21$$

After respective rotations the material properties for the composite media becomes:

$$a_{11}^{(1)} = 6.9113 \times 10^{-12} \text{ cm}^2/\text{dyne}$$
$$a_{12}^{(1)} = -1.4510 \times 10^{-12} \text{ cm}^2/\text{dyne}$$
$$a_{22}^{(1)} = 6.9113 \times 10^{-12} \text{ cm}^2/\text{dyne}$$
$$a_{66}^{(1)} = 17.075 \times 10^{-12} \text{ cm}^2/\text{dyne}$$
$$a_{16}^{(1)} = a_{26}^{(1)} = 0$$
$$\rho^{(1)} = 7.44 \text{ g/cm}^3$$

for the upper medium and

$$a_{11}^{(2)} = .72569 \times 10^{-12} \text{ cm}^2/\text{dyne}$$
$$a_{12}^{(2)} = -.15240 \times 10^{-12} \text{ cm}^2/\text{dyne}$$
$$a_{22}^{(2)} = 6.9113 \times 10^{-12} \text{ cm}^2/\text{dyne}$$
$$a_{66}^{(2)} = 17.075 \times 10^{-12} \text{ cm}^2/\text{dyne}$$
$$a_{16}^{(2)} = a_{26}^{(2)} = 0$$
$$\rho^{(2)} = 7.44 \text{ g/cm}^3$$

for the lower medium.

The finite element mesh used for this problem is the same as the one shown in Fig. 9.6, expect that the half crack length $a$ is taken to be 1cm and the singular element is relocated accordingly. Five Gaussian integration points and the first eighteen terms of the singular element approximating eigenfunctions were used. Uniformly distributed uniaxial tensile stresses, with a Heaviside step-function time dependence were assumed to act at the edges of the plate parallel to the crack axis. The crack is assumed to be stationary under the action of the applied loads.

For this problem values of $\beta = 1/4$ and $\Delta t = 1.5 \times 10^{-6}$ sec. were used. This time step allows the fastest wave to propagate a distance of 64.7% of the smallest dimension in the mesh. The results of $K = \sqrt{K_1^2 + K_2^2}$ are plotted in Fig. 9.9. It is seen that similar behavior to the previous problem are exhibited. For the given material properties the first wave to reach the crack tip is the longitudinal wave in the second material at an angle of $\theta \approx 46°$, from $X_1$ axis, from

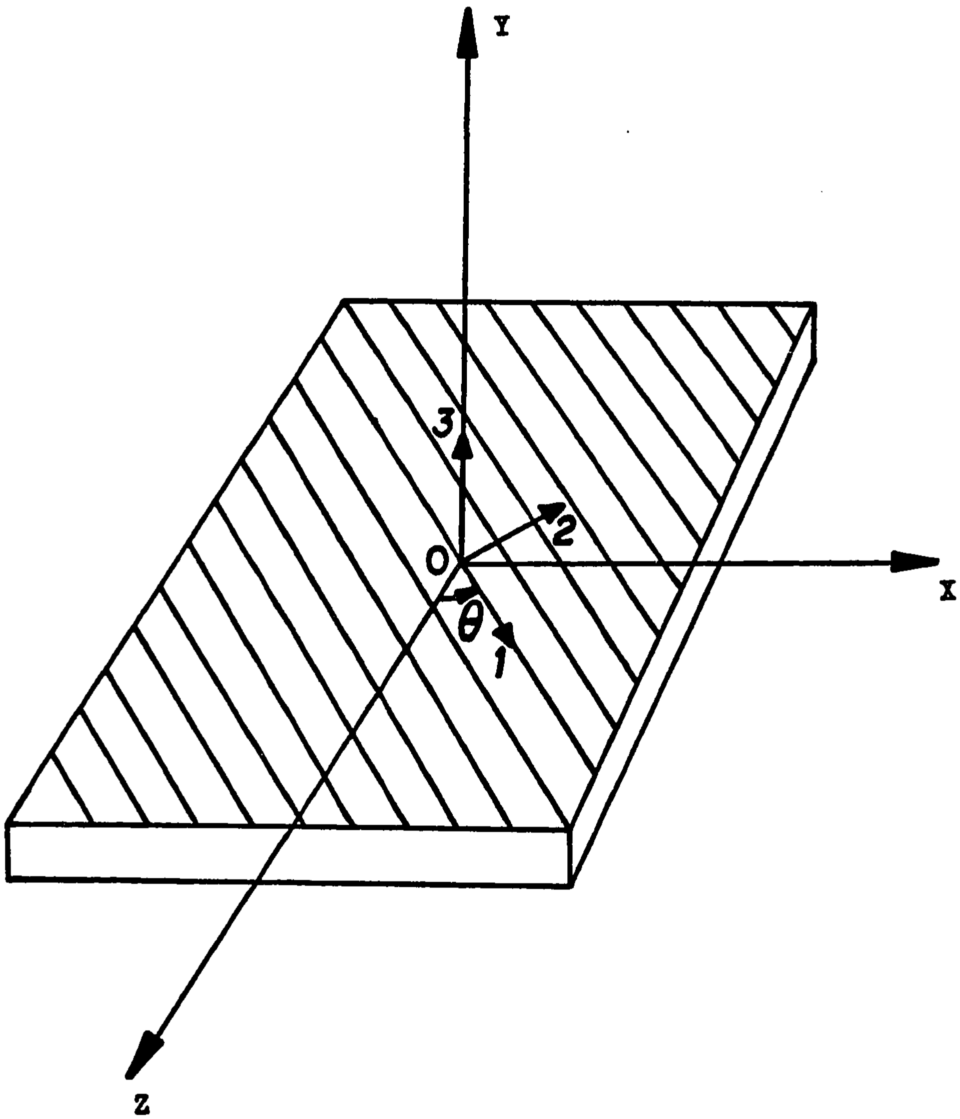



Fig. 9.8 Unidirectional Graphite-Epoxy Composites

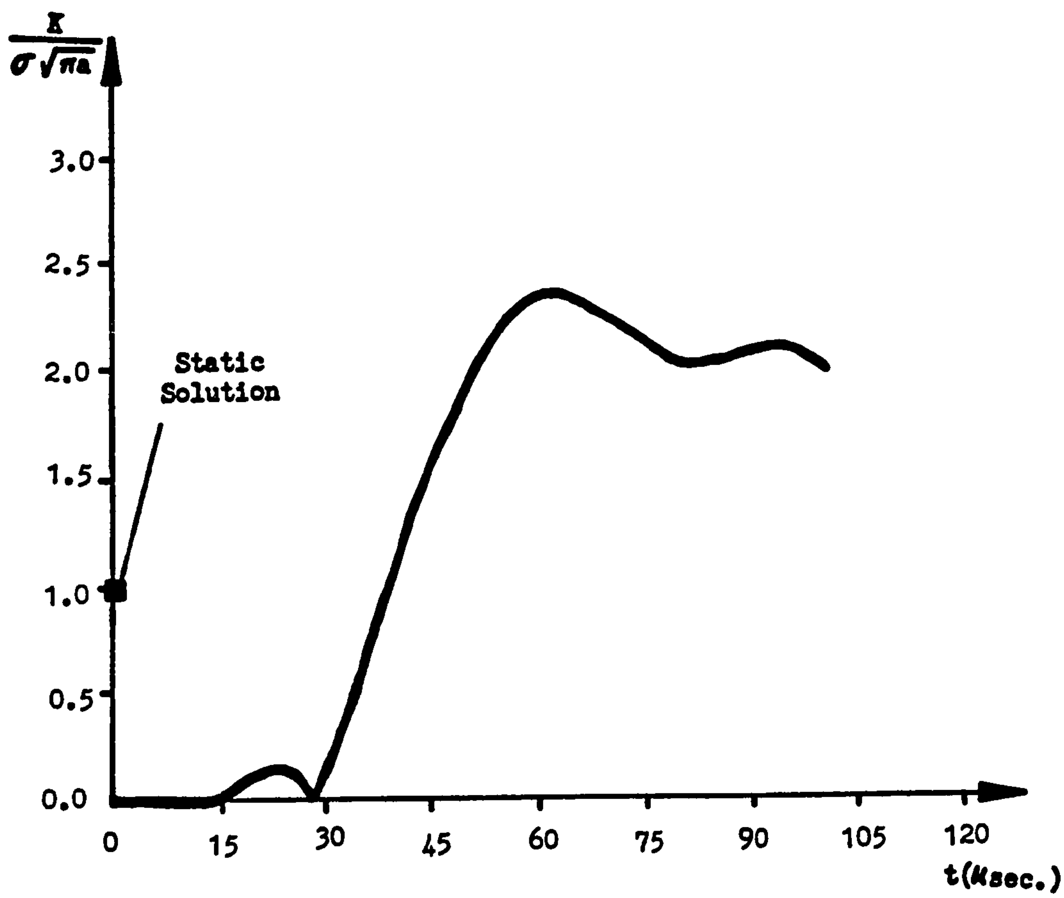



Fig. 9.9 Results of Stress Intensity Factors for a Center Crack at the Interface of Two Dissimilar Orthotropic Media Under Impact

the edges parallel to the crack axis (see Chapter 8). The wave reaches the crack-tip approximately 23 $\mu$sec after the application of the load. Therefore, the stress intensity factors should be zero for times $t<23$ $\mu$sec, but for the reasons explained earlier, non-zero values arise even before the waves reach the crack-tip.

## 9.3 Propagating crack problems

As a test of the procedure for dynamic crack propagation, the problem of the sudden appearance of a crack in a homogeneous, isotropic plate in a uniform tension field is solved. The finite element mesh used for this problem is shown in Fig. 9.10. Five Gaussian integration points and the first eighteen terms of the singular element eigenfunctions were used. Broberg [34] has investigated the problem of a suddenly appearing crack, which propagates at constant velocity in a uniform tension field. Although Broberg's solution is for an infinite medium, the results apply to a finite size plate for the times until the waves created at the crack-tip by the sudden appearance of the crack reflect from the boundaries of the plate and reach the crack-tip. Broberg's solution reduces to $K_1/K_s = .505$ for steel, for which $G = 2.94 \times 10^{11}$ dyne/cm$^2$, $\nu = .292$, $\rho = 2.45$ g/cm$^3$ for a crack speed of $c(t) = .6C_T$. Here $K_s = \sigma\sqrt{\pi a}$ is the static stress intensity factor for an infinite uniform medium with a half crack length of a and applied stresses of $\sigma$.

The finite element solution starts with a zero crack length with the static solution as the initial condition. Therefore the solution for the stress intensity factor starts with a value of $K_1/K_S = 1.0$ and approaches the Broberg's solution of $K_1/K_S = .505$ as the steady state solution. The results are plotted in Fig. 9.11 and it is seen that the finite element solution approaches Broberg's steady state solution in about thirty time steps. Also shown in Fig. 9.11 are the results of Aoki [33] and Gunther [16]. For this problem, values of $\beta = 1/4$ and $\Delta t = .481125 \times 10^{-6}$ sec. were used. This time step allows the crack to propagate a distance of .05cm or 1/20 of the singular element dimension. Also the time step corresponds to the longitudinal wave travelling a distance of 30.8 % of the smallest dimension in mesh.

Next the problem of a centrally located crack at the interface of two dissimilar orthotropic media is solved. It is assumed that the crack is at rest under the action of uniformly applied loads parallel to the crack axis, then suddenly starts to propagate at a constant velocity. The composite media is the same as the one described for the problem of a stationary crack at the interface of two dissimilar orthotropic materials under impact.

The finite element mesh used for this problem is the same as the one shown in Fig. 9.10. Five Gaussian integration points and the first eighteen terms of the singular element eigenfunctions are used.

It is assumed that the inital crack-length is 2cm and the crack suddenly starts to propagate at a constant velocity of $.5(C_T)_{min}$, where $(C_T)_{min}$ is the minimum shear wave speed of the media, i.e.

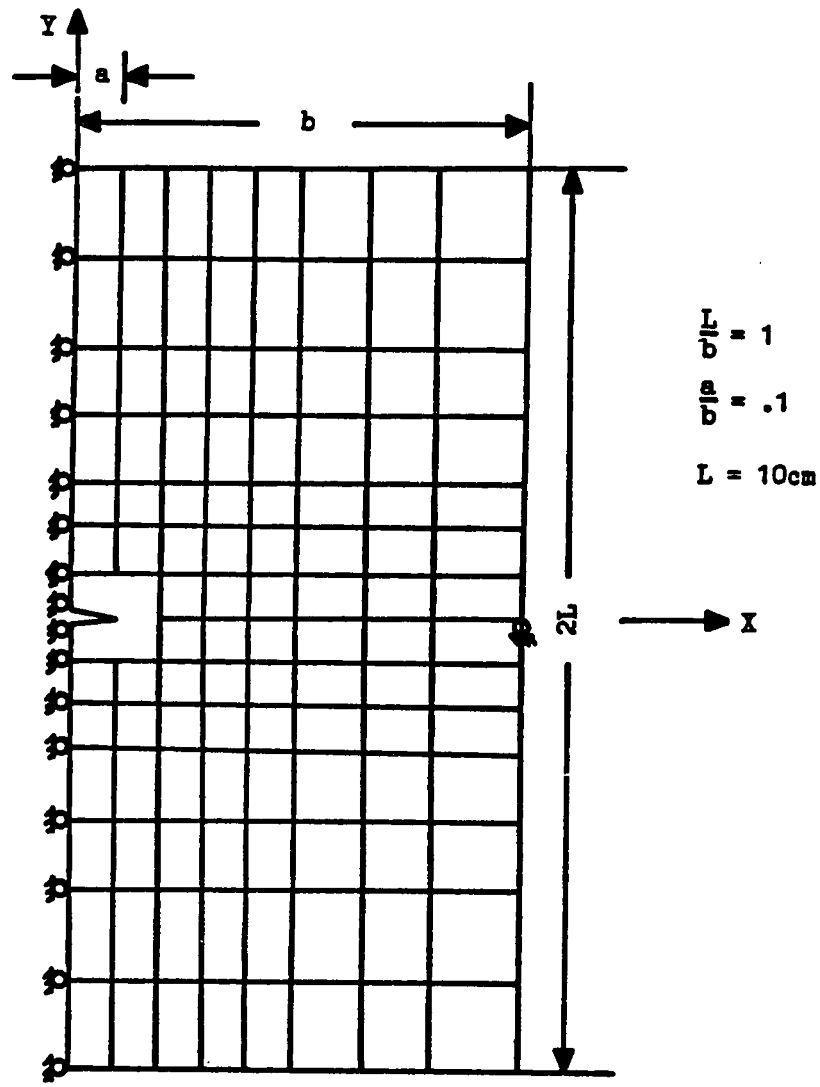



Fig. 9.10 Finite Element Mesh for Propagating Center Crack in a Homogeneous Isotropic Plate and for Propagating Center Crack Along the Interface of Two Dissimilar Orthotropic Materials

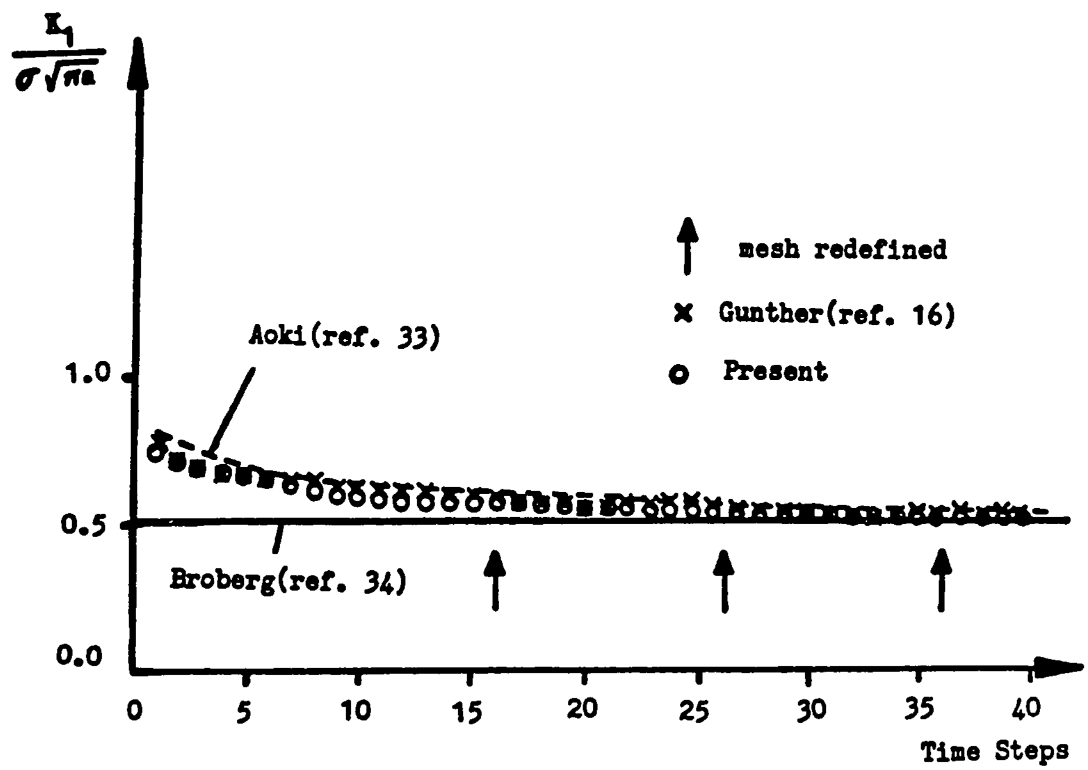



Fig. 9.11 Stress Intensity Factor for a Propagating Center Crack in a Homogeneous Isotropic Medium

$(C_T)_{min} = \sqrt{\frac{A_{66}}{\rho}} = .08872 \times 10^6$ cm/sec. This is the speed of the shear wave propagation in $X_1$ and $X_2$ directions.

The time step is chosen so that the crack will propagate a distance equal to 1/30 of the length of the singular element, i.e. $\Delta t = 1.50282 \times 10^{-6}$ sec. This time step allows the fastest longitudinal wave, i.e. $(C_L)_{max} = \sqrt{\frac{A_{11}^{(2)}}{\rho}} = .43137 \times 10^6$ cm/sec., to travel a distance of 64.8 % of the smallest dimension in the mesh. Again a value of $\beta = 1/4$ was used. The computed results for $K = \sqrt{K_1^2 + K_2^2}$ are plotted in Fig. 9.12. It is seen, as in the previous problem, that when the crack starts to propagate the stress intensity factor starts to decrease from the static value and finally assumes a constant value as the steady state solution. For this problem the steady state solution, Fig. 9.12, is $K_1 = .635K$ , where $K_s = \sigma\sqrt{\pi a}$ is the static stress intensity factor for an infinite uniform medium with a half crack length of a and applied stresses of $\sigma$.

It is seen that the rate of convergence to the steady state solution is much faster than in the previous problem. This is at least partly due to the fact that in the previous problem the initial crack length was taken to be zero, which causes the numerical algorithm to produce rather poor results at the start of the problem. In any case it is seen that the results do converge to a steady state solution.

## 9.4 Crack arrest problems

For safe design of structures, it is important to provide mechanisms for crack arrests. Therefore, it is of interest to know

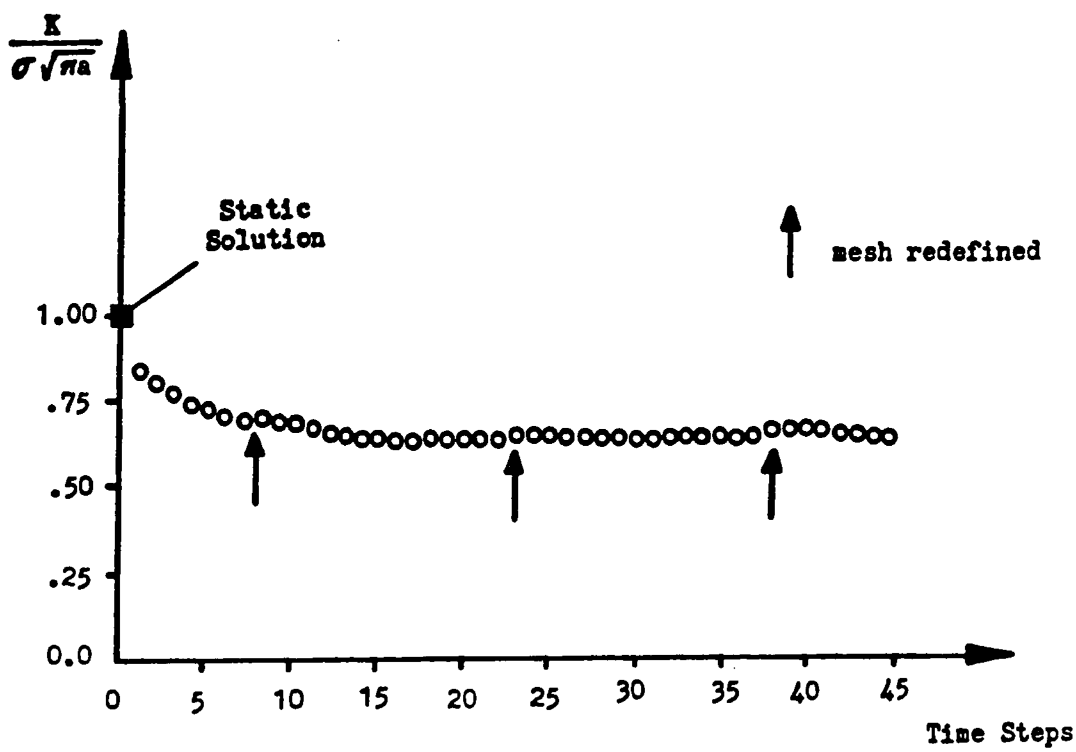



Fig. 9.12 Results of Stress Intensity Factors for a Propagating Center Crack at the Interface of Two Dissimilar Orthotropic Media

the values of stress intensity factors after a propagating crack is suddenly stopped.

The propagating cracks in the previous section were suddenly stopped at a half crack-length of 4cm. The computed results of the stress intensity factor $K_1$ for the first problem are plotted in Fig. 9.13. It is seen that the stress intensity factor $K_1$ rises sharply after the crack has been suddenly stopped. The stress intensity factor then reaches a peak value of 1.46 times the static value and then starts to decrease. For long times however, the stress intensity factor should assume the static value, see [35-36].

The peak value of the stress intensity factor occurs after the longitudinal waves created at the crack-tip by the sudden stoping of the crack reflect from the loaded boundary and reach the crack axis. Also plotted in Fig. 9.13 are the results of C.K. Gunther [16].

The computed results for the second problem of the previous section which involves the suddenly stopping of an interface crack between two dissimilar orthotropic materials are plotted in Fig. 9.14. It is seen that the same behavior as in the previous problem are exhibited. The peak value for the stress intensity factor in this problem is 1.48 times the static value.

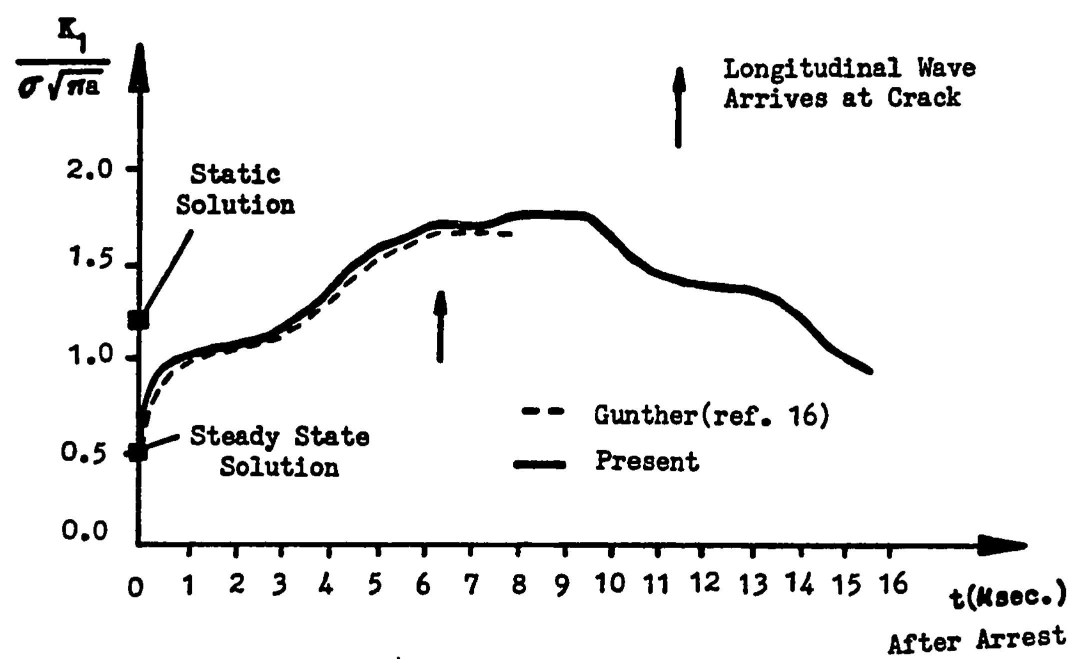



Fig. 9.13 Stress Intensity Factor for the Propagating Center Crack in a Homogeneous Isotropic Medium After Arrest

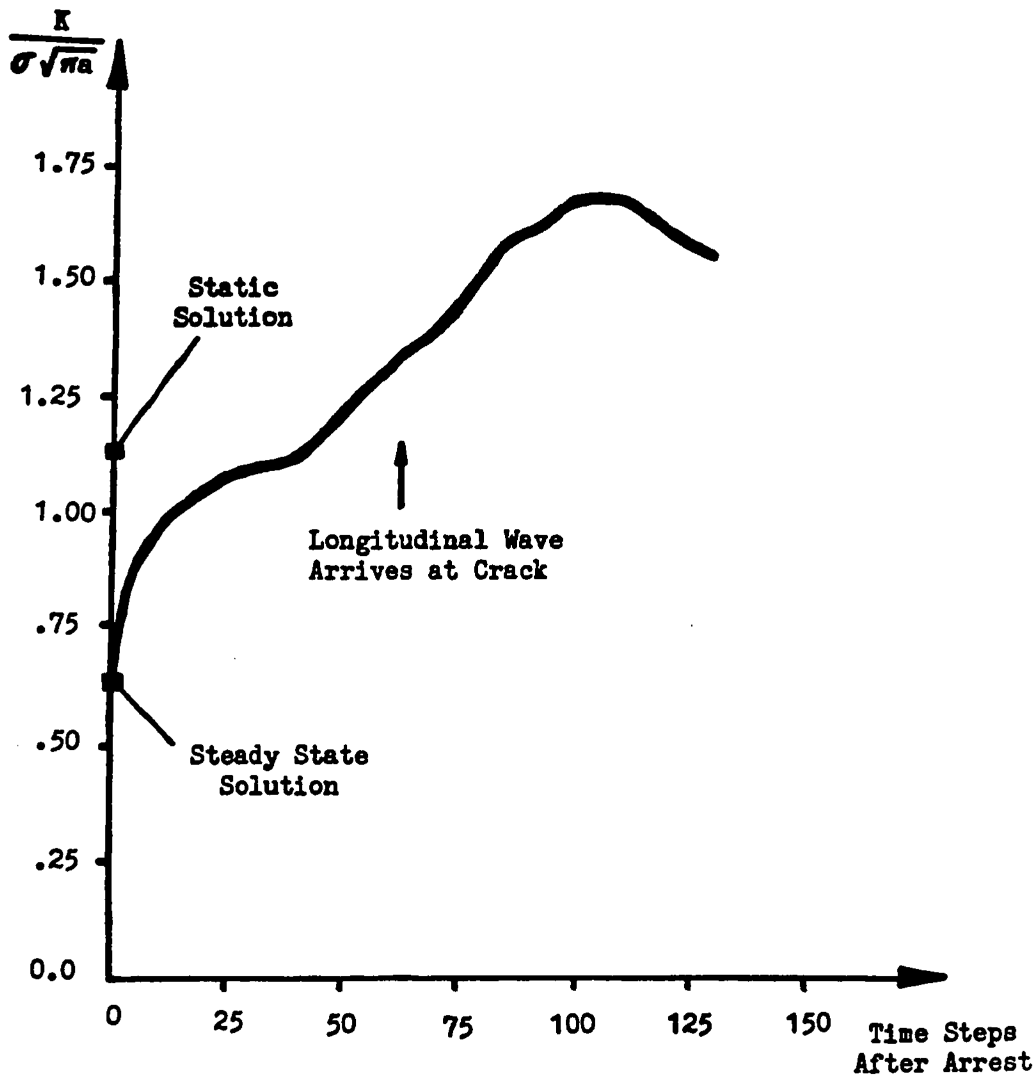



Fig. 9.14 Results of Stress Intensity Factors for the Propagating Center Crack at the Interface of Two Dissimilar Orthotropic Media After Arrest

# CHAPTER X

## SUMMARY AND CONCLUSION

A complete procedure for the elastodynamic analysis of interface cracks between two dissimilar anisotropic media has been formulated. It was shown that for all cases the stresses near the crack-tip exhibit the familiar oscillatory singularity of the type $r^{-1/2+i\varepsilon}$, where -1/2 represents the conventional square root singularity and the imaginary part represents the oscillatory behavior. When the two materials embedding the crack-tip become identical then $\varepsilon = 0$ and there is no oscillatory behavior for the stresses and displacements near the crack-tip.

The above derivations were successfully implemented into a hybrid-displacement finite element formulation. The resulting discretzed equations of motion were solved using Newmark-$\beta$ formulas. A Fortran computer code was then developed which is capable of solving a wide range of problems. Cracks can be either stationary or propagating at a prescribed rate and the materials can be isotropic, orthotropic or fully anisotropic. Due to the complexity of the problem analyzed here it was not possible to define the stress intensity factors by removing the stress singularity near the crack-tip by conventional definitions. Therefore a new definition for stress intensity factors is proposed which reduces to all the previous definitions to within a multiplicative factor.

In the finite element mesh, the crack-tip is embeded in a relatively large singular element, and all other elements away from the crack-tip are regular elements. If the crack-tip is not stationary, it will propagate inside the singular element until it reaches an extreme position, at which time a local remeshing takes place, the position of the singular element is moved forward, so that the crack-tip is relocated inside the singular element and can continue to propagate inside the singular element.

In the process of re-meshing, as the crack-tip propagates, new nodes have to be created in the finite element mesh. This requirement was satisfied by use of a method which was referred to as the double noding technique.

The results obtained on the basis of the developed procedure are very satisfying and further development is encouraged. In the next chapter some recommendations concerning further developments are presented.

# CHAPTER XI

## RECOMMENDATIONS FOR FURTHER DEVELOPMENT

The procedure described in this dissertation represents the development of a very complex numerical algorithm. Recommendations are made concerning the accuracy of the algorithm, the economics of computer code and increasing the capability of the procedure.

As it was mentioned the results for the same crack-length but different position of the crack-tip inside the singular element were slightly different. In order to modify this discrepancy it is recommended to study the effects of

(1) The number of Gaussian points for the integration procedure.

(2) The number of terms of the approximating eigenfunctions for the singular element.

(3) The use of singular elements with higher number of nodes such as the one used by Lin [5 and 30]. Also see [37].

(4) For dynamic problems the effect of the size of the time step and the use of difference formulas with artificial damping should be studied, see [22, 23, 38].

To reduce the cost of executing the computer code it is recommended to study the following

(1) Every effort has been made to keep the core storage requirement to a minimum. However, since the program is designed to handle dynamic problems involving propagating cracks, the core requirement is considerable. It might be worthwhile to

modify this program and make three separate programs, one for static problems, one for stationary cracks with dynamic loads and one for propagating cracks.

(2) The effect of the number of Gaussian points for the integration procedure and to see how many are needed for sufficient accuracy.

(3) The effect of the number of terms of the singular element eigenfunctions and to see how many terms are required for sufficient accuracy.

(4) The feasibility of a lumped mass matrix instead of a consistent mass matrix for both the regular elements and singular elements should be studied. This would allow the use of an explicit time integration procedure instead of an implicit one, which causes the elimination of the requirement of solving a large number of simultaneous equations. Also, if a lumped mass matrix for the singular element proves to be inaccurate, it is recommended to study the feasibility of a mixture of implicit and explicit time integration procedure. The subregion around the crack-tip could be treated implicitly while the region away from the crack-tip could be treated explicitly, thus resulting in considerable reduction in the number of simultaneous equations.

(5) Atluri, et al. [14] have used a different crack propagation method. Their procedure uses a fixed crack-tip position inside the singular element and crack propagation is

accomplished by moving the singular element which requires deforming the adjacent regular elements. In this procedure, for a given crack-tip speed, the singular element matrices have to be calculated only once. This saving in computing time is partially offset by the fact that the deforming regular element matrices have to be computed at each time step. However for a constant crack-tip speed the savings are expected to be considerable and it is recommended that this method of crack propagation be studied for the problem considered here.

As it was mentioned in the previous chapter the crack-tip velocity and position histories are prescribed a priori in the present analysis. However, in practical problems this information is not available. Thus it is recommended that the equation of motion for the crack-tip for interface cracks be implemented in the present code. This would allow the crack-tip velocity and position to be determined from the equation of motion according to a prescribed fracture criterion, see [39-40].

## REFERENCES


1. Williams, M.L., "The Stresses Around a Fault or Crack in Dissimilar Media", Bulletin of Seismological Society of America, Vol. 49, No. 2, April 1959, pp. 199-204.
2. Erdogan, F., "Stress Distribution in a Non-Homogeneous Elastic Plane with Cracks", Journal of Applied Mechanics, Transactions of the ASME, June 1963, pp. 232-236.
3. Sih, G.C. and Rice, J.R., "The Bending of Plates of Dissimilar Materials with Cracks", Journal of Applied Mechanics, Transactions of the ASME, September 1964, pp. 477-482.
4. Rice, J.R. and Sih, G.C., "Plane Problems of Cracks in Dissilimar Media", Journal of Applied Mechanics, Transactions of the ASME, June 1965, pp. 418-423.
5. Lin, K.Y. and Mar, J.W., "Finite Element Analysis of Stress Intensity Factors for Cracks at a Bi-material Interface", International Journal of Fracture, 1976, pp. 521-531.
6. Theocaris, P.S., "The Order of Singularity at a Multi-wedge Corner of a Composite Plate", International Journal of Engineering and Science, Pergamon Press, printed in Great Britain, Vol. 12, 1974, pp. 107-120.
7. Bogy, D.B., "Two Edge-Bonded Elastic Wedges of Different Materials and Wedge Angles under Surface Tractions", Journal of Applied Mechanics, Transactions of the ASME, June 1971, pp. 377-386.

8. Bogy, D.B., "On the Plane Elastostatic Problem of a Loaded Crack Terminating at a Material Interface", Journal of Applied Mechanics, Transactions of the ASME, December 1971, pp. 911-918.

9. Fenner, D.N., "Stress Singularity in Composite Materials with an Arbitrary Oriented Crack Meeting an Interface", International Journal of Fracture, Vol. 12, 1976, pp. 705-721.

10. An-Yu-Kuo* and Su-Su Wang**, "A Dynamic Hybrid Finite Element Analysis of Delamination in Composites", *Nutech Engineers, Inc., San Jose, CA, USA, **University of Illinois, Urbana, IL, USA.

11. Aboudi, J., "The Dynamic Stresses Induced by Moving Interfacial Cracks", Computer Methods in Applied Mechanics and Engineering, North-Holland Publishing Co., Vol. 10, 1977, pp. 303-323.

12. Muskhelishvili, N.I., Some Basic Problems of the Mathematical Theory of Elasticity, 2nd English Translation (1962) by J.R.M. Radok from the Fourth Russian edition of 1954, P. Noordhoff Ltd, Groningen-The Netherlands, 1963.

13. Lekhnitskii, S.G., "Theory of Elasticity of an Anistropic Body", Translated by P. Fern, edited by Julius J. Brandstatter, Holden-Day Series in Mathematical Physics, Holden-Day Inc., San Francisco, 1963.

14. Atluri, S.N., Nishioka, T. and Nakagaki, M., "Numerical Modeling of Dynamic and Non-Linear Crack Propagation in Finite Bodies by Moving Singular Elements", Nonlinear and Dynamic Fracture

Mechanics, edited by N. Perrone and S.N. Atluri, AMD, Vol. 35, ASME, 1979, pp. 37-66.

15. Nishioka, T. and Atluri, S.N., "Numerical Modeling of Dynamic Crack Propagation in Finite Bodies, by Moving Singular Elements", Journal of Applied Mechanics, Transactions of the ASME, Vol. 47, Sept. 1980, Part 1: Formulation, pp. 570-576, Part 2: Results, pp. 577-582.

16. Gunther, C.K., Holsapple, K.A. and Kobayashi, A.S., "Finite Element Analysis of Cracking Bodies", AIAA Journal, Vol. 19, No. 6, June 1981, pp. 789-795.

17. Liaw, B.M., Kobayashi, A.S. and Emery, A.F., "Double Noding Technique for Mixed Mode Crack Propagation Studies", University of Washington, Department of Mechanical Engineering, Seattle, WA.

18. Santosh, K. Arya, and Hegemier, Gilbert A., "Finite Element Method for Interface Problems", Journal of the Structural Division, Proceedings of the ASCE, Vol. 108, No. ST2, Feb. 1982, pp. 327-342.

19. Lekhnitskii, S.G., "Anisotropic Plates", Contributions to the Metallurgy of Steel, No. 50, American Iron and Steel Institute, translated from the Russian by Elbridge Z. Stowell, June 1956.

20. Freund, L.B. and Clifton, R.J., "On the Uniqueness of Plane Elastodynamic Solutions for Running Cracks", Journal of Elasticity, Vol. 4, No. 4, Dec. 1974, Noordhoff International Publishing-Leyden, printed in The Netherlands.

21. Huth, J.H., "The Complex Variable Approach to Stress Singularities", Journal of Applied Mechanics, Transactions of the ASME, Vol. 75, 1953, pp. 561-562.

22. Newmark, N.M., "A Method of Computation for Structural Dynamics", Journal of the Engineering Mechanics Division, proceedings of the ASCE, 1959, pp. 67-94.

23. Belytshko, T., "A survey of Numerical Methods and Computer Programs for Dynamic Structural Analysis", Nuclear Engineering and Design, North-Holland Publishing Company, Vol. 37, 1976, pp. 23-34.

24. Fung, Y.C., Foundations of Solid Mechanics, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1965, pp. 176-232.

25. Christensen, R.M., Mechanics of Compositie Materials, John Wiley and Sons, Inc., New York, 1979, pp. 225-273.

26. Kuo, A.Y., "Transient Stress Intensity Factors of an Interfacial Crack Between Two Dissimilar Anisotropic Half-spaces", Part 1; Orthotropic Materials, Journal of Applied Mechanics, Transactions of the ASME, Vol. 51, March 1984, pp. 71-76.

27. Kuo, A.Y., "Transient Stress Intensity Factors of an Interfacial Crack Between Two Dissimilar Anisotropic Half-spaces", Part 2: Fully Orthotropic Materials, Journal of Applied Mechanics, Transactions of the ASME, Vol. 51, Dec. 1984, pp. 780-786.

28. Isida, M., "Effect of With and Length on Stress Intensity Factors of Internally Cracked Plates Under Various Boundary

Conditions", International Journal of Fracture, Vol. 7, No. 3, 1971, pp. 301-306.

29. Bowie, O.L. and Freese, C.E., "Central Crack in Plane Orthotropic Rectangular Sheet", International Journal of Fracture, Vol. 8, No. 1, 1972, pp. 49-58.

30. Lin, K.Y. and Tong, P., "A Hybrid Crack Element for the Fracture Mechanics Analysis of Composite Materials", paper presented at the International Conference on Numerical Methods in Fracture Mechanics, Swansea, U.K., Jan. 9-13, 1978, pp. 733-746.

31. Baker, B.R., "Dynamic Stresses Created by a Moving Crack:, Journal of Applied Mechanics, ASME, Vol. 29, 1962, pp. 449-545.

32. Sih, G.C., Embley, G.T., and Ravera, R.S., "Impact Response of a Finite Crack in Plane Extension", International Journal of Solids and Structures, Vol. 8, 1972, pp. 977-993.

33. Aoki, S., et al., "Elastodynamic Analysis of Crack by Finite Element Method Using Singular Element", International Journal of Fracture, Vol. 14, 1978, pp. 59-68.

34. Broberg, K.B., "The Propagation of a Brittle Crack", Arkiv for Fysik, Band 18, No. 10, 1960, pp. 159-192.

35. Freund, L.B., "The Analysis of Elastodynamic Crack Tip Stress Fields", Mechanics Today, Vol. 3, 1976, edited by S. Nemat-Nasser, Pergamon Press, New York, pp. 55-91.

36. Erdogan, F., "Crack Propagation Theories", Fracture, Vol. 2, edited by H. Liebowitz, Academic Press, New York, 1968, pp. 498-586.

37. Nishioka, T., Stonesifer, R.B., and Atluri, S.N., "An evaluation of several Moving Singularity Finite Element Models for Fast Fracture Analysis", Engineering Fracture Mechanics, Vol. 15, No. 1-2, 1981, pp. 205-218.

38. Wood, W.L., Bossak, M., and Zienkiewicz, O.C., "An Alpha Modification of Newmark's Method", International Journal for Numerical Methods in Engineering, Vol. 15, 1980, pp. 1562-1566.

39. Achenbach, J.D. and Brock, L.M., "On Quasistatic and Dynamic Fracture", Proceedings of an International Conference on Dynamic Crack Propagation, G.C. Sih, Editor, Noordhoff International Publising, Leyden, The Netherlands, 1973, pp. 529-541.

40. Achenbach, J.D., "Dynamic Effects in Brittle Fracture", Mechanics Today, edited by S. Nemat-Nasser, Vol. 1, Pergamon Press, Oxford, 1974, pp. 1-54.

41. Tong, P., "New Displacement Hybrid Finite Element Models for Solid Continua", International Journal for Numerical Methods in Engineering, Vol. 2, 1970, pp. 73-83.

## APPENDIX A

## THE DERIVATION OF THE EIGENVALUES AND EIGENFUNCTIONS FOR THE STRESSES AND DISPLACEMENTS OF A PROPAGATING CRACK AT THE INTERFACE OF TWO DISSIMILAR ANISOTROPIC MATERIALS

As discussed in Chapter 3, the characteristic equation (3.23) has either two distinct pairs of complex roots or one pair of complex root of multiplicity two. Both cases are considered here. For each case we will show that using the conditions of zero tractions on the crack surfaces and the equality of the stresses and displacements ahead of the crack-tip, we shall obtain similar equations. Then we will solve for the eigenvalues and the corresponding eigenvectors and eigenfunctions. The two cases of interest are now considered.

Case (a). Consider the case in which equation (3.23) has two distinct pairs of complex roots $\mu_{1k}$, $\bar{\mu}_{1k}$ and $\mu_{2k}$, $\bar{\mu}_{2k}$. Then let

$$\phi_k = \text{Real}\left(\int \Omega_{1k} dz_{1k} + \int \Omega_{2k} dz_{2k}\right) \tag{A.1}$$

where $z_{ik} = x + \mu_{ik} y \qquad i, k = 1, 2$

and k identifies the material under consideration. Let $\Omega_{ik}$ have a complex power representation as

$$\Omega_{1k} = C_{1k}\, z_{1k}^{n} + D_{1k}\, z_{1k}^{\bar{n}} \tag{A.2}$$

$$\Omega_{2k} = C_{2k}\, z_{2k}^{n} + D_{2k}\, z_{2k}^{\bar{n}}$$

Later it will be shown that there are an infinite number of solutions for the exponent n and the functions $\Omega_{ik}$ actually become complex power series.

Introducing equation (A.1) into equation (3.17) gives

$$\underset{\sim}{\sigma}_k = \underset{\sim}{B}_k \, \underset{\sim}{\phi}_{\sigma k} = \underset{\sim}{B}_k \ \text{Real} \ (\underset{\sim}{Q}_{1k}\underset{\sim}{\Omega}_{\sigma k}) = \text{Real} \ (\underset{\sim}{SIG}_k \, \underset{\sim}{\Omega}_{\sigma k}) \quad \text{(A.3)}$$

where

$$\underset{\sim}{Q}_{1k} = \begin{bmatrix} 1 & 1 \\ \mu_{1k}^2 & \mu_{2k}^2 \\ \mu_{1k} & \mu_{2k} \end{bmatrix} , \quad \underset{\sim}{\Omega}_{\sigma k} = \begin{bmatrix} \Omega'_{1k} \\ \Omega'_{2k} \end{bmatrix}$$

$$\underset{\sim}{\phi}_{\sigma k} = \begin{bmatrix} \phi_{k,11} \\ \phi_{k,22} \\ \phi_{k,12} \end{bmatrix} , \quad \underset{\sim}{\sigma}_k = \begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{bmatrix}_k \quad \text{(A.3a)}$$

and $\underset{\sim}{SIG}_k = \underset{\sim}{B}_k \underset{\sim}{Q}_{1k}$

with $\underset{\sim}{B}_k$ being the matrix $\underset{\sim}{B}$ given by equation (3.17a) for material k.

Introducing equation (A.3) into the constitutive equation (3.18) yields

$$\underset{\sim}{\varepsilon}_k = \underset{\sim}{a}_k \underset{\sim}{\sigma}_k = \underset{\sim}{a}_k \text{Real}(\underset{\sim}{SIG}_k \underset{\sim}{\Omega}_{\sigma k}) = \text{Real}(\underset{\sim}{EPS}_k \underset{\sim}{\Omega}_{\sigma k}) \quad \text{(A.4)}$$

where $\underset{\sim}{EPS}_k = \underset{\sim}{a}_k \underset{\sim}{SIG}_k$

and

$$\underset{\sim}{\varepsilon}_k = \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ 2\varepsilon_{12} \end{bmatrix}_k \qquad \underset{\sim}{a}_k = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_2 & a_4 & a_5 \\ a_3 & a_5 & a_6 \end{bmatrix}_k \quad \text{(A.4a)}$$

Using the displacement - strain equations (2.1) one has

$$u_{1k} = \int (\varepsilon_{11})_k dx \ , \quad u_{2k} = \int (\varepsilon_{22})_k dy$$

Using equation (A.4) in the above gives

$$u_{ik} = \text{Real} \ (\underset{\sim}{DIS}_{ik} \underset{\sim}{\Omega}_{uk}) \qquad i = 1,2 \quad \text{(A.5)}$$

where

$$\underset{\sim}{\Omega}_{uk} = \begin{bmatrix} \Omega_{1k} \\ \Omega_{2k} \end{bmatrix} \quad \text{(A.5a)}$$

$$\underset{\sim}{DIS}_{1k} = \underset{\sim}{EPS}_{1k}$$

$$\underset{\sim}{DIS}_{2k} = \underset{\sim}{EPS}_{2k}\underset{\sim}{Q}_{2k}$$

where $\underset{\sim}{EPS}_{ik}$ is the $i^{th}$ row of matrix $\underset{\sim}{EPS}_k$ defined in equation (A.4)

and
$$\underset{\sim}{Q}_{2k} = \begin{bmatrix} 1/\mu_{1k} & 0 \\ 0 & 1/\mu_{2k} \end{bmatrix}$$

Equations (A.5) can be written as

$$\underset{\sim}{u}_k = \text{Real}(\underset{\sim}{DIS}_k \underset{\sim}{\Omega}_{uk}) \tag{A.6}$$

with
$$\underset{\sim}{u}_k = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}_k \quad \text{and} \quad \underset{\sim}{DIS}_k = \begin{bmatrix} \underset{\sim}{DIS}_{1k} \\ \underset{\sim}{DIS}_{2k} \end{bmatrix} \tag{A.6a}$$

Now let us form the expressions for $\sigma_{22}-i\sigma_{12}$ and $u_1+iu_2$ in order to express the conditions of zero tractions on the crack surfaces and the conditions of equality of stresses and displacements ahead of the crack-tip across the interface.

$$\sigma_{22} - i\sigma_{12} = \text{Real}[\underset{\sim}{SIG}_{2k}\underset{\sim}{\Omega}_{\sigma k}] - i\text{Real}[\underset{\sim}{SIG}_{3k}\underset{\sim}{\Omega}_{\sigma k}]$$

$$= \frac{\underset{\sim}{SIG}_{2k} - i\underset{\sim}{SIG}_{3k}}{2}\underset{\sim}{\Omega}_{\sigma k} + \frac{\overline{\underset{\sim}{SIG}}_{2k} - i\overline{\underset{\sim}{SIG}}_{3k}}{2}\overline{\underset{\sim}{\Omega}}_{\sigma k}$$

or

$$\sigma_{22} - i\sigma_{12} = K_{1k}\Omega'_{1k} + K_{2k}\overline{\Omega}'_{1k} + K_{3k}\Omega'_{2k} + K_{4k}\overline{\Omega}'_{2k} \tag{A.7}$$

where $\underset{\sim}{SIG}_{ik}$ is the $i^{th}$ rows of matrix $\underset{\sim}{SIG}_k$ and

$$K_{1k} = \frac{(SIG_{21})_k - i(SIG_{31})_k}{2} \tag{A.7a}$$

$$K_{2k} = \frac{\overline{(SIG_{21})_k} - i\overline{(SIG_{31})_k}}{2}$$

$$K_{3k} = \frac{(SIG_{22})_k - i(SIG_{32})_k}{2}$$

$$K_{4k} = \frac{\overline{(SIG_{22})_k} - i\overline{(SIG_{32})_k}}{2}$$

where $(SIG_{ij})_k$ are the elements of $\underset{\sim}{SIG}_k$.

$$u_{1k} + iu_{2k} = \text{Real}[\underset{\sim}{DIS}_{1k}\underset{\sim}{\Omega}_{uk}] + i\text{Real}[\underset{\sim}{DIS}_{2k}\underset{\sim}{\Omega}_{uk}]$$

$$= \frac{\underset{\sim}{DIS}_{1k} + i\underset{\sim}{DIS}_{2k}}{2}\underset{\sim}{\Omega}_{uk} + \frac{\overline{\underset{\sim}{DIS}}_{1k} + i\overline{\underset{\sim}{DIS}}_{2k}}{2}\overline{\underset{\sim}{\Omega}}_{uk}$$

or

$$u_{1k} + iu_{2k} = K_{5k}\Omega_{1k} + K_{6k}\overline{\Omega}_{1k} + K_{7k}\Omega_{2k} + K_{8k}\overline{\Omega}_{2k} \tag{A.8}$$

where

$$K_{5k} = \frac{(DIS_{11})_k + i(DIS_{21})_k}{2}$$

$$K_{6k} = \frac{\overline{(DIS_{11})_k} + i\overline{(DIS_{21})_k}}{2}$$

$$K_{7k} = \frac{(DIS_{12})_k + i(DIS_{22})_k}{2}$$

$$K_{8k} = \frac{\overline{(DIS_{12})_k} + i\overline{(DIS_{22})_k}}{2}$$

with $(DIS_{ij})_k$ being the elements of $\underset{\sim}{DIS}_k$.

The condition of zero tractions on the crack surfaces can be expressed as

$$\sigma_{22} - i\sigma_{12} = 0 \tag{A.9}$$

on the negative x-axis.

Let us express $z_{ik}$ in terms of polar coordinate variables r, $\theta$ as

$$z_{jk} = r_{jk}e^{i\theta_{jk}} \tag{A.10}$$

However, on the crack surface

$$z_{1k} = z_{2k} = re^{\frac{\alpha_k}{2n}} \tag{A.10a}$$

with $\alpha$ defined as $\alpha_k = \begin{cases} 2in\pi & \text{for } k=1 \\ -2in\pi & \text{for } k=2 \end{cases}$ (A.10b)

Then, introducing the above into equation (A.3a) and using equation (A.2) gives

$$\underset{\sim}{\Omega}_{\sigma k} = \begin{bmatrix} \Omega'_{1k} \\ \Omega'_{2k} \end{bmatrix} = \begin{bmatrix} C_{1k}ne^{\frac{\alpha_k}{2}} & D_{1k}\bar{n}e^{-\frac{\bar{\alpha}_k}{2}} \\ C_{2k}ne^{\frac{\alpha_k}{2}} & D_{2k}\bar{n}e^{-\frac{\bar{\alpha}_k}{2}} \end{bmatrix} \begin{bmatrix} r^{n-1} \\ r^{\bar{n}-1} \end{bmatrix} \tag{A.11}$$

substituting equations (A.11) into equations (A.8) and, using equation (A.7) yields

$$[K_{1k}C_{1k}e^{\alpha_k} + K_{2k}\bar{D}_{1k} + K_{3k}C_{2k}e^{\alpha_k} + K_{4k}\bar{D}_{2k}]r^{n-1}ne^{-\frac{\alpha_k}{2}}$$

$$+ [K_{1k}D_{1k} + K_{2k}\bar{C}_{1k}e^{\bar{\alpha}_k} + K_{3k}D_{2k} + K_{4k}\bar{C}_{2k}e^{\bar{\alpha}_k}]r^{\bar{n}-1}\bar{n}e^{-\frac{\bar{\alpha}_k}{2}} = 0$$

This equation has to hold for all values of r. Since $r^n$ and $r^{\bar{n}}$ are independent, each of the expressions in the brackets must vanish, which gives

$$K_{3k}C_{2k}e^{\alpha_k} + K_{4k}\bar{D}_{2k} + K_{1k}C_{1k}e^{\alpha_k} + K_{2k}\bar{D}_{1k} = 0$$

and $$K_{4k}\bar{C}_{2k}e^{\bar{\alpha}_k} + K_{3k}D_{2k} + K_{2k}\bar{C}_{1k}e^{\bar{\alpha}_k} + K_{1k}D_{1k} = 0$$

Forming the complex conjugate of the second equation and solving for $C_{2k}$ and $\overline{D}_{2k}$ in terms of $C_{1k}$ and $\overline{D}_{1k}$ gives

$$\begin{bmatrix} C_{2k} \\ \overline{D}_{2k} \end{bmatrix} = \begin{bmatrix} KK_{1k} & KK_{2k} e^{-\alpha_k} \\ \overline{KK}_{2k}\, e^{\alpha_k} & \overline{KK}_{1k} \end{bmatrix} \begin{bmatrix} C_{1k} \\ \overline{D}_{1k} \end{bmatrix} \tag{A.12}$$

where

$$KK_{1k} = (K_{1k}\overline{K}_{3k} - K_{4k}\overline{K}_{2k})/R_o$$

$$KK_{2k} = (K_{2k}\overline{K}_{3k} - K_{4k}\overline{K}_{1k})/R_o$$

with $\quad R_o = K_{4k}\overline{K}_{4k} - K_{3k}\overline{K}_{3k}$

The conditions of equality of tractions and displacements across the interface ahead of the crack-tip requires

$$(\sigma_{22} - i\sigma_{12})_1 = (\sigma_{22} - i\sigma_{12})_2 \tag{A.13}$$

and

$$(u_1 + iu_2)_1 = (u_1 + iu_2)_2$$

on the positive x-axis, where, equation (A.10) becomes

$$z_{1k} = z_{2k} = r \tag{A.14}$$

Introducing equation (A.14) into equations (A.3a) and (A.5a), and using equations (A.12) and substituting the results into equations (A.13) gives

$$(\sigma_{22} - i\sigma_{12})_k = \{(H_{1k} + H_{2k}e^{\alpha_k})C_{1k} + (H_{3k} + H_{4k}e^{-\alpha_k})\overline{D}_{1k}\}(-n)r^{n-1}$$

$$+ \{(H_{3k} + H_{4k}e^{\overline{\alpha}_k})\overline{C}_{1k} + (H_{1k} + H_{2k}e^{-\overline{\alpha}_k})D_{1k}\}(-\overline{n})r^{\overline{n}-1} \tag{A.15a}$$

and

$$(u_1 + iu_2)_k = \{(H_{5k} + H_{6k}e^{\alpha_k})C_{1k} + (H_{7k} + H_{8k}e^{-\alpha_k})\overline{D}_{1k}\}(-n)r^{n-1}$$

$$+ \{(H_{7k} + H_{8k}e^{\overline{\alpha}_k})\overline{C}_{1k} + (H_{5k} + H_{6k}e^{-\overline{\alpha}_k})D_{1k}\}(-\overline{n})r^{\overline{n}-1} \quad \text{(A.15b)}$$

where

$$H_{1k} = K_{1k} + K_{3k}KK_{1k} = -K_{4k}\overline{KK}_{2k}$$

$$H_{2k} = K_{4k}\overline{KK}_{2k}$$

$$H_{3k} = K_{2k} + K_{4k}\overline{KK}_{1k} = -K_{3k}KK_{2k}$$

$$H_{4k} = K_{3k}KK_{2k}$$

$$H_{5k} = K_{5k} + K_{7k}KK_{1k}$$

$$H_{6k} = K_{8k}\overline{KK}_{2k}$$

$$H_{7k} = K_{6k} + K_{8k}\overline{KK}_{1k}$$

$$H_{8k} = K_{7k}KK_{2k}$$

It should be noted that the parameters $H_{ik}$ are functions of material properties and crack-tip speed and are known quantities.

We will come back to this point in our analysis later, but for now, let us leave this case here, and let us do the same treatment for the case (b) where equation (3.23) has only one distinct pair of complex roots. It will be shown that case (b) will also result in similar set of equations as in (A.15) for case (a).

Case (b). Equation (3.23) has only one distinct pair of complex roots $\mu_k$ and $\overline{\mu}_k$. Let

$$\phi_k = \text{Real}(\overline{z}_k\Omega_{1k} + \int \Omega_{2k}dz_k) \quad \text{(A.16)}$$

where $z_k = x + \mu_k y \qquad k = 1,2$

Assume that $\Omega_{ik}$ have power series representation as

$$\Omega_{1k} = C_{1k} z_k^{n} + D_{1k} z_k^{\bar{n}} \tag{A.17}$$

$$\Omega_{2k} = C_{2k} z_k^{n} + D_{2k} z_k^{\bar{n}}$$

Introducing equations (A.17) into equations (3.17) gives

$$\underset{\sim}{\sigma}_k = \underset{\sim}{B}_k \underset{\sim}{\phi}_{\sigma k} = \underset{\sim}{B}_k \text{ Real}(\underset{\sim}{Q}_{1k}\underset{\sim}{\Omega}_{\sigma k}) = \text{Real}(\underset{\sim}{SIG}_k \underset{\sim}{\Omega}_{\sigma k}) \tag{A.18}$$

Where

$$\underset{\sim}{Q}_{1k} = \begin{bmatrix} 1 & 2 & 1 \\ \mu_k^2 & -2i\mu_k & \mu_k^2 \\ \mu_k & \mu_k - i & \mu_k \end{bmatrix} \qquad \underset{\sim}{\Omega}_{\sigma k} = \begin{bmatrix} \Omega'_{2k} \\ \Omega'_{1k} \\ \bar{z}_k \Omega''_{1k} \end{bmatrix} \tag{A.18a}$$

$$\underset{\sim}{SIG}_k = \underset{\sim}{B}_k \underset{\sim}{Q}_{1k}$$

and $\underset{\sim}{\sigma}_k$, $\underset{\sim}{\phi}_{\sigma k}$ and $\underset{\sim}{B}_k$ are as defined in equation (A.3a).

Introducing equation (A.18) into the constitutive equation (3.18) yields

$$\underset{\sim}{\varepsilon}_k = \underset{\sim}{a}_k \underset{\sim}{\sigma}_k = \underset{\sim}{a}_k \text{Real}(\underset{\sim}{SIG}_k \underset{\sim}{\Omega}_{\sigma k}) = \text{Real}(\underset{\sim}{EPS}_k \underset{\sim}{\Omega}_{\sigma k}) \tag{A.19}$$

where $\underset{\sim}{EPS}_k = \underset{\sim}{a}_k \underset{\sim}{SIG}_k$ (A.19a)

and $\underset{\sim}{\varepsilon}_k$ and $\underset{\sim}{\sigma}_k$ are as defined in (A.4a).

Substituting equation (A.19) into the displacement-strain relations of (2.1) and (A.5) gives

$$u_{1k} = \int (\varepsilon_{11})_k dx = \int \underset{\sim}{\text{Real}}[\underset{\sim}{EPS}_{1k} \underset{\sim}{\phi}_{\sigma k}] dx = \underset{\sim}{\text{Real}}[\underset{\sim}{EPS}_{1k} \underset{\sim}{Q}_{2k} \underset{\sim}{\Omega}_{uk}]$$

or $u_{1k} = \text{Real}[\underset{\sim}{DIS}_{1k} \underset{\sim}{\Omega}_{uk}]$ (A.20a) and $u_{2k} = \int(\varepsilon_{22})_k dy = \int Real[\underset{\sim}{EPS}_{2k}\underset{\sim}{\Omega}_{\sigma k}]dy = Real[\underset{\sim}{EPS}_{2k}\underset{\sim}{Q}_{3k}\underset{\sim}{\Omega}_{uk}]$

or $u_{2k} = Real[\underset{\sim}{DIS}_{2k}\underset{\sim}{\Omega}_{uk}]$ (A.20b)

where

$$\underset{\sim}{\Omega}_{uk} = \begin{bmatrix} \Omega_{2k} \\ \Omega_{1k} \\ \bar{z}_k\Omega'_{1k} \end{bmatrix}, \quad \underset{\sim}{Q}_{2k} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix}$$

$$\underset{\sim}{Q}_{3k} = \begin{bmatrix} 1/\mu_k & 0 & 0 \\ 0 & 1/\mu_k & 0 \\ 0 & i/\mu_k^2 & 1/\mu_k \end{bmatrix} \qquad (A.20c)$$

$$\underset{\sim}{DIS}_{1k} = \underset{\sim}{EPS}_{1k}\underset{\sim}{Q}_{2k}$$
$$\underset{\sim}{DIS}_{2k} = \underset{\sim}{EPS}_{2k}\underset{\sim}{Q}_{3k}$$

with $\underset{\sim}{EPS}_{ik}$ being the $i^{th}$ row of matrix $\underset{\sim}{EPS}_k$ defined in (A.4).

Equations (A.20a) and (A.20b) can be written as

$$\underset{\sim}{u}_k = Real(\underset{\sim}{DIS}_k\underset{\sim}{\Omega}_{uk}) \qquad (A.21)$$

with $\underset{\sim}{u}_k$ and $\underset{\sim}{DIS}_k$ as defined in equation (A.6a).

Now let us form the expressions for $\sigma_{22}-i\sigma_{12}$ and $u_1+iu_2$

$$\sigma_{22} - i\sigma_{12} = Real[\underset{\sim}{SIG}_{2k}\underset{\sim}{\Omega}_{\sigma k}] - iReal[\underset{\sim}{SIG}_{3k}\underset{\sim}{\Omega}_{\sigma k}]$$

$$= K_{1k}\Omega'_{2k} + K_{2k}\bar{\Omega}'_{2k} + K_{3k}\Omega'_{1k} + K_{4k}\bar{\Omega}'_{1k} + K_{1k}\bar{z}_k\Omega''_{1k} + K_{2k}z_k\bar{\Omega}''_{1k} \qquad (A.22)$$

where $K_{1k}$, $K_{2k}$, $K_{3k}$, $K_{4k}$ are defined in the same manner as in equations (A.7a), but with $(\underset{\sim}{SIG}_{ij})_k$ being the elements of matrix $\underset{\sim}{SIG}_k$ as defined in equation (A.18a).

Note that the first column and the third column of the matrix $\underset{\sim}{Q}_{1k}$ in equation (A.18a) are identical, which causes the matrix $\underset{\sim}{SIG}_k$

defined in equation (A.18a); $EPS_k$ defined in equation (A.19a) and $\underset{\sim}{DIS}_{1k}$ and $\underset{\sim}{DIS}_{2k}$ defined in equation (A.20c), with definitions of $\underset{\sim}{Q}_{2k}$ and $\underset{\sim}{Q}_{3k}$ in equation (A.20c), have the same property of identical first and third columns. Therefore, in equation (A.22), the coefficients of $\Omega'_{2k}$ and $\overline{\Omega}'_{2k}$ are identical to the coeffieicents of $\overline{z}_k\Omega''_{1k}$ and $z_k\overline{\Omega}''_{1k}$ respectively.

Now
$$u_{1k}+iu_{2k}=\text{Real}[\underset{\sim}{DIS}_{1k}\underset{\sim}{\Omega}_{uk}]+i\text{Real}[\underset{\sim}{DIS}_{2k}\underset{\sim}{\Omega}_{uk}]$$
$$=K_{5k}\Omega_{2k}+K_{6k}\overline{\Omega}_{2k}+K_{7k}\Omega_{1k}+K_{8k}\overline{\Omega}_{1k}+K_{5k}z_k\overline{\Omega}'_{1k}+K_{6k}z_k\overline{\Omega}'_{1k} \tag{A.23}$$

with $K_{5k}$, $K_{6k}$, $K_{7k}$ and $K_{8k}$ as defined in equations (A.8a); but, with the elements of $(DIS_{ij})_k$ being the elements of matrix $\underset{\sim}{DIS}_k$ as defined in equation (A.20c). Again, in equation (A.23) in coefficients of $\Omega_{2k}$ and $\overline{\Omega}_{2k}$ are identical to those of $\overline{z}_k\Omega'_{1k}$ and $z_k\overline{\Omega}'_{1k}$ respectively, because of the above reasoning.

The condition of traction free surfaces is expressed by equation (A.9). Let us express $z_k$ in terms of the polar coordinates in the same manner as in equations (A.10) as

$$z_k = r_k e^{i\theta_k} \tag{A.24}$$

However, on the crack surfaces

$$z_k = re^{\frac{\alpha_k}{2n}} \tag{A.25}$$

with $\alpha_k$ defined as in equation (A.10b).

Introducing equation (A.24a) into equations (A.18a), and using equation (A.17), gives

$$\underset{\sim}{\Omega}_{\sigma k} = \begin{bmatrix} \Omega'_{2k} \\ \Omega'_{1k} \\ \bar{z}_k \Omega''_{1k} \end{bmatrix} = - \begin{bmatrix} C_{2k} n e^{\frac{\alpha_k}{2}} & D_{2k} \bar{n} e^{-\frac{\bar{\alpha}_k}{2}} \\ C_{1k} n e^{\frac{\alpha_k}{2}} & D_{1k} \bar{n} e^{-\frac{\bar{\alpha}_k}{2}} \\ C_{1k} n(n-1) e^{\frac{\alpha_k}{2}} & D_{1k} \bar{n}(\bar{n}-1) e^{-\frac{\bar{\alpha}_k}{2}} \end{bmatrix} \begin{bmatrix} r^{n-1} \\ r^{\bar{n}-1} \end{bmatrix} \tag{A.26}$$

Substituting equation (A.26) into equation (A.9) and carrying out the same process and the same reasoning as for case (a) one obtains

$$\begin{bmatrix} C_{2k} \\ \bar{D}_{2k} \end{bmatrix} = \begin{bmatrix} KK_{1k} - (n-1) & KK_{2k} e^{-\alpha_k} \\ \overline{KK}_{2k} \, e^{\alpha_k} & \overline{KK}_{1k} - (n-1) \end{bmatrix} \begin{bmatrix} C_{1k} \\ \bar{D}_{1k} \end{bmatrix} \tag{A.27}$$

where

$$KK_{1k} = (K_{3k}\bar{K}_{1k} - K_{2k}\bar{K}_{4k})/R_o$$

$$KK_{2k} = (K_{4k}\bar{K}_{1k} - K_{2k}\bar{K}_{3k})/R_o$$

with

$$R_o = K_{2k}\bar{K}_{2k} - K_{1k}\bar{K}_{1k}$$

where $K_{1k}$, $K_{2k}$, $K_{3k}$, $K_{4k}$ are as defined in equation (A.22) for case(b).

The conditions of equality of tractions and displacements ahead of the crack-tip are as expressed by equations (A.13). Also expressing $z_k$ along the positive x-axis as $z_k=r$ and using equations (A.27) yields

$$(\sigma_{22} - i\sigma_{12})_k = \{(H_{1k} + H_{2k}e^{\alpha_k})C_{1k} + (H_{3k} + H_{4k}e^{-\alpha_k})\bar{D}_{1k}\}(-n)r^{n-1}$$
$$+ \{(H_{3k} + H_{4k}e^{\bar{\alpha}_k})\bar{C}_{1k} + (H_{1k} + H_{2k}e^{-\bar{\alpha}_k})D_{1k}\}(-\bar{n})r^{\bar{n}-1} \tag{A.28a}$$

and

$$(u_1+iu_2)_k=\{(H_{5k}+H_{6k}e^{\alpha_k})C_{1k}+(H_{7k}+H_{8k}e^{-\alpha_k})\overline{D}_{1k}\}(-n)r^{n-1} \quad (A.28b)$$

$$+\{(H_{7k}+H_{8k}e^{\overline{\alpha}_k})\overline{C}_{1k}+(H_{5k}+H_{6k}e^{-\overline{\alpha}_k})D_{1k}\}(-\overline{n})r^{\overline{n}-1}$$

where

$$H_{1k}= K_{1k}KK_{1k}+K_{3k}= - K_{2k}\overline{KK}_{2k}$$

$$H_{2k}= K_{2k}\overline{KK}_{2k}$$

$$H_{3k}= K_{2k}\overline{KK}_{1k}+K_{4k}= - K_{1k}KK_{2k}$$

$$H_{4k}= K_{1k}KK_{2k}$$

$$H_{5k}= K_{5k}[KK_{1k}+1]+K_{7k}$$

$$H_{6k}= K_{6k}\overline{KK}_{2k}$$

$$H_{7k}= K_{6k}[\overline{KK}_{1k}+1]+K_{8k}$$

$$H_{8k}= K_{5k}KK_{2k}$$

It is observed that equations (A.15) of case (a) and (A.28) of case (b) are in the same form. Therefore, no matter whether the characteristic equation of (3.23) has one, or two distinct pairs of complex roots for either of the two materials involved, the restrictions expressed by equations (A.9) and (A.13) result in similar sets of equations.

In the literature for the formulation of anisotropic materials and also the formulation of propagating cracks only case (a) has been considered. Therefore the formulations could not have been reduced to isotropic materials and stationary cracks, since for problems involving isotropic materials and stationary cracks case (b) has to be considered. Since the nature of the solution for case (b) is different than for case (a) this reduction has not been possible. However, we have shown here that for both cases the formulations reduce to the same form and therefore it is possible to have one single formualtion, as presented here, for all the possible cases, i.e. each one of the materials on the sides of the interface can be isotropic, orthrotorpic, or anisotropic and the crack can be stationary or propagating.

Now using the earlier argument of independency of $r^{n}$ and $r^{\overline{n}}$, the following sets of equations will result from conditions of zero tractions on the crack surfaces and the continuity of stresses and displacements across the interface ahead of the crack-tip expressed by equations (A.9) and (A.13)

$$\begin{bmatrix} H_{11}(1-e^{\alpha}) & H_{31}(1-e^{-\alpha}) & -H_{12}(1-e^{-\alpha}) & -H_{32}(1-e^{\alpha}) \\ \overline{H}_{31}(1-e^{\alpha}) & \overline{H}_{11}(1-e^{-\alpha}) & -\overline{H}_{32}(1-e^{-\alpha}) & -\overline{H}_{12}(1-e^{\alpha}) \\ H_{51}+H_{61}e^{\alpha} & H_{71}+H_{81}e^{-\alpha} & -(H_{52}+H_{62}e^{-\alpha}) & -(H_{72}+H_{82}e^{\alpha}) \\ \overline{H}_{71}+\overline{H}_{81}e^{\alpha} & \overline{H}_{51}+\overline{H}_{61}e^{-\alpha} & -(\overline{H}_{72}+\overline{H}_{82}e^{-\alpha}) & -(\overline{H}_{52}+\overline{H}_{62}e^{\alpha}) \end{bmatrix} \begin{bmatrix} C_{11} \\ \overline{D}_{11} \\ C_{12} \\ \overline{D}_{12} \end{bmatrix} = 0 \qquad (A.29)$$

where $\alpha=2in\pi$

Non-trivial solutions for the complex coefficients $C_{ik}$ and $D_{ik}$ are possible only when the determinant of the matrix in equation (A.29) vanishes, which gives

$$(1-x)^{2}\ (1+2bx+x^{2})=0 \qquad (A.30)$$

where

$$x = e^{\alpha}$$

$$b = \frac{R_o}{2Q_o}$$

where

$$R_o = -A_{11}A_{22}-A_{12}A_{21}+2\text{Real}(C_5C_7-C_1C_2)$$

$$Q_o = -A_{11}B_2-A_{12}B_1+(-C_1C_3+C_5C_8+\bar{C}_5\bar{C}_6-\bar{C}_1\bar{C}_4)$$

where

$$A_{1k} = H_{1k}\bar{H}_{1k}-H_{3k}\bar{H}_{3k} \qquad \text{no sum} \quad k=1,2$$

$$A_{2k} = H_{5k}\bar{H}_{5k}+H_{6k}\bar{H}_{6k}-H_{7k}\bar{H}_{7k}-H_{8k}\bar{H}_{8k}$$

$$B_1 = H_{51}\bar{H}_{61}-H_{81}\bar{H}_{71}$$

$$B_2 = H_{62}\bar{H}_{52}-H_{72}\bar{H}_{82}$$

$$C_1 = H_{11}\bar{H}_{32}-H_{12}\bar{H}_{31}$$

$$C_2 = H_{72}\bar{H}_{51}+H_{82}\bar{H}_{61}-H_{71}\bar{H}_{52}-H_{81}\bar{H}_{62}$$

$$C_3 = H_{72}\bar{H}_{61}-H_{81}\bar{H}_{52}$$

$$C_4 = H_{82}\bar{H}_{51}-H_{71}\bar{H}_{62}$$

$$C_5 = H_{32}\bar{H}_{31}-H_{11}\bar{H}_{12}$$

$$C_6 = H_{52}\bar{H}_{51}-H_{71}\bar{H}_{72}$$

$$C_7 = H_{62}\bar{H}_{51}+H_{52}\bar{H}_{61}-H_{71}\bar{H}_{82}-H_{81}\bar{H}_{72}$$

$$C_8 = H_{62}\bar{H}_{61}-H_{81}\bar{H}_{82}$$

It is noted that the parameters $A_{ik}$, $i,k=1,2$ are real so that the parameter $R_o$ will be real. It can be shown, after some algebra, that the parameters $B_1$ and $B_2$ and also the expression $(-C_1C_3+C_5C_8+\overline{C}_5\overline{C}_6-\overline{C}_1\overline{C}_4)$ are all real so that the parameter $Q_0$ is also real. Furthermore it can be shown that $b = \frac{R_0}{2Q_0} \geq 1$, with the equality holding when the two materials became identical.

From equations (A.30) the acceptable eigenvalues to give finite displacements at the crack-tip are

$$n = 1, 2, 3, \ldots$$

$$n = \frac{1}{2} \pm i\varepsilon, \frac{3}{2} \pm i\varepsilon, \frac{5}{2} \pm i\varepsilon \ \ldots$$

where

$$\varepsilon = \frac{1}{2\pi} \log (b + \sqrt{b^2-1}) \qquad (A.31)$$

For static problems the paramter $\varepsilon$ has been called the "bielastic constant" [3-4]. For crack propagation problems however, $\varepsilon$ is a function of the crack-tip speed as well as the material properties, therefore, here we call it the "bielastic parameter".

It will be shown later that $n = 1/2 + i\varepsilon$ and $n = 1/2 - i\varepsilon$ lead to the same solutions, so that only $n = 1/2 + i\varepsilon$ needs to be considered. Thus the eigenvalues are taken to be

$$\begin{aligned} n_\ell &= \frac{1}{2} + i\varepsilon, \frac{3}{2} + i\varepsilon, \frac{5}{2} + i\varepsilon \ \ldots, \quad \ell = 1, 3, 5 \ldots \\ n_\ell &= 1, 2, 3 \ldots \qquad\qquad\qquad \ell = 2, 4, 6 \ldots \end{aligned} \qquad (A.32)$$

Now, for each eigenvalue $n_\ell$, we can use equations (A.29) along with equation (A.27) or equation (A.12), whichever appropriate, to solve for all the unknown coefficients in terms of only one set of as yet unknown coefficients, say $\beta_\ell$'s, as follows. If the eigenvalue $n_\ell$ is complex eliminate one row, e.g. the third row, of equations (A.29) and from the remaining three equations solve for $\bar{D}_{11}$, $C_{12}$, $\bar{D}_{12}$ in terms of $C_{11}$ and then use equations (A.27) or (A.12), whichever is appropriate, to solve for the other unknown coefficients, so that one has

$$\begin{bmatrix} C_{11} \\ \bar{D}_{11} \\ C_{21} \\ \bar{D}_{21} \\ C_{12} \\ \bar{D}_{12} \\ C_{22} \\ \bar{D}_{22} \end{bmatrix}_{\ell} = \begin{bmatrix} F_{11\ell} \\ F_{21\ell} \\ F_{31\ell} \\ F_{41\ell} \\ F_{12\ell} \\ F_{22\ell} \\ F_{32\ell} \\ F_{42\ell} \end{bmatrix} \beta_{\ell} \quad \text{or} \quad \begin{bmatrix} C_{11} & D_{11} \\ C_{21} & D_{21} \\ C_{12} & D_{12} \\ C_{22} & D_{22} \end{bmatrix}_{\ell} = \begin{bmatrix} F_{11\ell} & \bar{F}_{21\ell} \\ F_{31\ell} & \bar{F}_{41\ell} \\ F_{12\ell} & \bar{F}_{22\ell} \\ F_{32\ell} & \bar{F}_{42\ell} \end{bmatrix} \begin{bmatrix} \beta_{\ell} & 0 \\ 0 & \bar{\beta}_{\ell} \end{bmatrix} \qquad (A.33)$$

with $F_{11\ell}=1$ and where we have renamed the unknown coefficients $C_{11\ell}$ as $\beta_{\ell}$. All the parameters $F_{ik\ell}$'s are the result of calculations and are known quantities.

However, if the eigenvalue $n_{\ell}$ is real, all the unknown coefficients can not be determined independently and only the following partial sums can be determined

$$\begin{aligned} E_{11} &= C_{11} + D_{11} \\ E_{21} &= C_{21} + D_{21} \\ E_{12} &= C_{12} + D_{12} \\ E_{22} &= C_{22} + D_{22} \end{aligned} \qquad (A.34)$$

To solve for $E_{ik}$'s eliminate the first two rows in equation (A.29) and solve for $C_{12}$ and $\bar{D}_{12}$ in terms of $C_{11}$ and $\bar{D}_{11}$ from the last two equations and then solve for $E_{12}$ in terms of $E_{11}$ and $\bar{E}_{11}$. Then use equations (A.27) or (A.12) to find $E_{21}$ and $E_{22}$ in terms of $E_{11}$ and $\bar{E}_{11}$. Therefore,

$$\begin{bmatrix} E_{11} \\ E_{21} \\ E_{12} \\ E_{22} \end{bmatrix}_{\ell} = \begin{bmatrix} F_{11\ell} & \bar{F}_{21\ell} \\ F_{31\ell} & \bar{F}_{41\ell} \\ F_{12\ell} & \bar{F}_{22\ell} \\ F_{32\ell} & \bar{F}_{42\ell} \end{bmatrix} \begin{bmatrix} \beta_{\ell} \\ \bar{\beta}_{\ell} \end{bmatrix} \tag{A.35}$$

with $F_{11\ell}=1$, $F_{21\ell}=0$ and the unknown coefficient $E_{11\ell}$ is renamed as $\beta_{\ell}$. Again, all the parameters $F_{ik\ell}$'s are the result of calculations and are known quantities.

It is proper now to show that $n=1/2+i\varepsilon$ and $n=1/2-i\varepsilon$ actually lead to the same solution, so that one needs to consider only one of them, namely, $n=1/2+i\varepsilon$.

For $n_{\ell}=1/2+i\varepsilon$ the result is as in equations (A.33). A careful study of equation (A.29) and (A.27) and (A.12) reveals that $n_m = 1/2-i\varepsilon$ will have the result

$$\begin{bmatrix} C_{11} & D_{11} \\ C_{21} & D_{21} \\ C_{12} & D_{12} \\ C_{22} & D_{22} \end{bmatrix}_{m} = \begin{bmatrix} \bar{F}_{21m} & F_{11m} \\ \bar{F}_{41m} & F_{31m} \\ \bar{F}_{22m} & F_{12m} \\ \bar{F}_{42m} & F_{32m} \end{bmatrix} \begin{bmatrix} \beta_m & 0 \\ 0 & \bar{\beta}_m \end{bmatrix}$$

Substituting a linear combination of the above and equations (A.33) into equations (A.2) and substituting only the equations (A.33) into equations (A.2) one obtains similiar expressions with only the unknown complex coefficient $\beta_\ell$'s looking different. Therefore the two eigenvalues of $n=1/2 + i\varepsilon$ will lead to the same solution and only one of the eigenvalues needs to be considered.

Consider now the limiting case when one of the materials become rigid. It should be noted that for a rigid material all of the displacements and strains are identically zero by definition. Therefore, in the finite element formulation a rigid material does not contribute to the functional equation (Chapter 5, equation 5.1).

Assuming that material k is the rigid material, in which case $H_{5k}$,$H_{6k}$, $H_{7k}$ and $H_{8k}$ in equation (A.29) vanish. Then, it is seen that for the eigenvalues $n_\ell$=1, 2, 3... The complex coefficients corresponding to the non-rigid material vanish, while the complex coefficients corresponding to the rigid material remain to be determined. In other words, for these eigenvalues the non-rigid material does not contribute to the functional equation, and since the rigid material as discussed does not contribute to the functional equation either, these terms should not be accounted for.

However, for the eigenvalues $n_\ell = \ell/2 + i\varepsilon$, $\ell$ = 1, 3, 5,..., the complex coefficients $C_{ik}$'s and $D_{ik}$'s do not vanish for either of the two materials. Although the rigid material does not contribute to the functional equation, these terms have to be accounted for since the terms corresponding to the non-rigid material do contribute to the functional equation.

Now let us derive the expressions for the state of stresses and displacements. Substituting equations (A.33) and (A.35) into equations (A.3) and (A.6) or (A.18) and (A.21) one obtains

$$\sigma_{pk\ell} = \mathrm{Real}(S_{pk\ell}\beta_\ell) = [\mathrm{Real}(S_{pk\ell}) \;\; -\mathrm{Im}(S_{pk\ell})] \begin{bmatrix}(\beta_1)_\ell \\ (\beta_2)_\ell\end{bmatrix} \tag{A.36}$$

$$u_{mk\ell} = \mathrm{Real}(D_{mk\ell}\beta_\ell) = [\mathrm{Real}(D_{mk\ell}) \;\; -\mathrm{Im}(D_{mk\ell})] \begin{bmatrix}(\beta_1)_\ell \\ (\beta_2)_\ell\end{bmatrix}$$

where $(\beta_1)_\ell$ and $(\beta_2)_\ell$ are two real unknown coefficients to be determined from the prescribed boundary conditions, and

$$S_{pk\ell} = n_\ell[s_{p1k}F_{1k\ell}z_{1k}^{n_\ell-1} + \bar{s}_{p1k}F_{2k\ell}\bar{z}_{1k}^{n_\ell-1} + s_{p2k}F_{3k\ell}z_{2k}^{n_\ell-1} + \bar{s}_{p2k}F_{4k\ell}\bar{z}_{2k}^{n_\ell-1}] \tag{A.36a}$$

$$D_{mk\ell} = d_{m1k}F_{1k\ell}z_{1k}^{n_\ell} + \bar{d}_{m1k}F_{2k\ell}\bar{z}_{1k}^{n_\ell} + d_{m2k}F_{3k\ell}z_{2k}^{n_\ell} + \bar{d}_{m2k}F_{4k\ell}\bar{z}_{2k}^{n_\ell}$$

for case (a) and

$$S_{pk\ell} = n_\ell[(s_{p1k}F_{3k\ell} + s_{p2k}F_{1k\ell})z_k^{n_\ell-1} + (\bar{s}_{p1k}F_{4k\ell} + \bar{s}_{p2k}F_{2k\ell})\bar{z}_k^{n_\ell-1}$$
$$+ (s_{p1k}(n_\ell-1)F_{1k\ell})\bar{z}_k z_k^{n_\ell-2} + (\bar{s}_{p1k}(n_\ell-1)F_{2k\ell})z_k\bar{z}_k^{n_\ell-2}] \tag{A.36b}$$

$$D_{mk\ell} = [(d_{m1k}F_{3k\ell} + d_{m2k}F_{1k\ell})z_k^{n_\ell} + (\bar{d}_{m1k}F_{4k\ell} + \bar{d}_{m2k}F_{2k\ell})\bar{z}_k^{n_\ell}$$
$$+ (d_{m1k}n_\ell F_{1k\ell})\bar{z}_k z_k^{n_\ell-1} + (\bar{d}_{m1k}n_\ell F_{2k\ell})z_k\bar{z}_k^{n_\ell-1}]$$

for case (b), with p=1,2,3 for $\sigma_{pk\ell}$ representing $\begin{bmatrix}\sigma_{1k\ell}\\ \sigma_{2k\ell}\\ \sigma_{3k\ell}\end{bmatrix} = \begin{bmatrix}\sigma_{11}\\ \sigma_{22}\\ \sigma_{12}\end{bmatrix}_{k\ell}$

for the $k^{th}$ material and the $\ell^{th}$ eigenvalue $n_\ell$ and with m=1,2 for

$u_{mk\ell}$ representing $\begin{bmatrix} u_{1k\ell} \\ u_{2k\ell} \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}_{k\ell}$ with $k$ and $\ell$ as above. Also $s_{pjk}$ and $d_{mjk}$ with j=1,2 represent the elements of matrices $\underset{\sim}{SIG}_k$ and $\underset{\sim}{DIS}_k$ in equations (A.3), (A.6), (A.13) and (A.21).

In the finite element formulation of the problem the first and second time derivatives of the stresses and displacements are needed, but their derivations from equations (A.36) and (3.4) are straightforward and will not be produced here.

This concludes the derivation of the eigenvalues and the corresponding eigenfunctions for the near-field stresses and displacements of a propagating crack at the interface of two dissimilar anisotropic materials.

# APPENDIX B

## COMPUTER PROGRAM DESCRIPTION

The computer program, Finite Element Analysis of Propagating Interface Cracks in Composites (FEAPICC), is written to be compatible with both FORTRAN IV and FORTRAN V compilers. The program is operable on the CDC (Cyber) 180-855. Many subroutines were newly developed, some others were taken from [16] with the necessary modifications implemented. The program is lengthy and complicated. Part of the complication is due to the efforts to make the program as efficient as possible in terms of both memory management and execution time. In the following pragraphs a brief description of the program subroutines is presented.

FEAPICC

This is the main program. It reads, checks and writes the problem title and a number of control paramters. It calculates for $\underset{\sim}{A}=\underset{\sim}{a}^{-1}$ of equation (3.24) for each material. It calls several subroutines to initiate and to proceed with solution to the problem.

GAUSSPT

This subroutine is called by the main program FEAPICC and sets the values of the necessary parameters for Gaussian quadrature formulas to perform the necessary integrations. Up to 10 points for the Gaussian quadrature can be selected.

DATAIN

This subroutine is called by the main program FEAPICC and reads the bulk of the input data and checks for errors. In case of a restart problem, this subroutine calls subroutine TAPIN to read the restart values from unit TAPE16.

<u>SOLVE</u>

This subroutine is called by the main program FEAPICC. The entire solution process is directed from this subroutine. It contains a loop on timesteps which is executed until a final specified time is reached. During the execution of each loop it calls other subroutines which assemble the global matrices and solve the system of equations.

<u>MATPROD</u>

This subroutine is called by subroutine SOLVE and performs matrix multiplication $\underset{\sim}{A}\underset{\sim}{b}=\underset{\sim}{C}$ for a given banded symmetric matrix $\underset{\sim}{A}$, which is stored in upper triangular form and a given column matrix $\underset{\sim}{b}$.

<u>FORMK</u>

This subroutine is called by subroutine SOLVE and stores the element mass and stiffness matrices in upper triangular form into the global matrices. Storage is performed in blocks with dimensions of one bandwidths by one bandwidths. The core requirement for program execution is thus kept to a minimum. The blocks are written onto disk files after the completion of each block storage operation. Subroutines STIFEL and MASSEL are called to obtain the regular element stiffness and mass matrices, respectively. The singular element stiffness, damping and mass matrices are provided by subroutine SINGEL.

<u>ESOLVE</u>

This subroutine is called by subroutine SOLVE at each time step.

It accounts for the displacement boundary conditions and solves the system of equations

$$\underset{\sim}{K}\underset{\sim}{q} = \underset{\sim}{Q}$$

for $\underset{\sim}{q}$ (the nodal displacements). Matrix $\underset{\sim}{K}$ is banded symmetric and stored in upper triangular form. The solution process is by Gaussian elimination which is carried out block by block compatible with storage procedure in FORMK. This subroutine calculates the nodal velocities and accelerations using the difference formulas (7.1). It calculates the displacements, velocities and accelerations of the internal nodes of the singular element using equations (3.34) and (5.10) and their derivatives. The stress intensity factors are also calculated in this subroutine.

<u>MASSEL</u>

It is called by subroutine FORMK and calculates the regular element mass matrices.

<u>FUNCTR</u>

It is called by subroutine MASSEL and calculates values of shape functions for regular isoparametric elements.

<u>STIFEL</u>

It is called by subroutine FORMK and calculates regular element stiffness matrices.

<u>DIFFB</u>

It is called by subroutines STIFEL and DATOUT and calculates values of derivatives of shape functions for regular isoparametric elements.

POSIT

It is called by subroutines DATAIN and SOLVE. It determines the crack-tip position from a given crack-tip position history. It also redefines the finite element mesh when necessary.

VELOC

It is called by subroutines DATAIN and SOLVE. It determines the crack-tip velocity from a given crack-tip velocity history. It also determines the time step for the next cycle in conjunction with subroutine LOAD.

LOAD

It is called by subroutines DATAIN and SOLVE. It stores the external element loads into the global external force vector from the given load history. It also determines the time step for the next cycle in conjunction with subroutine VELOC.

DATOUT

This subroutine is called by subroutine SOLVE. It calculates the stresses at the center of specified elements and writes the nodal displacmeents, velocities, accelerations, and center-element stresses on the output at specified times.

TAPOUT

It is called by subroutine SOLVE. It writes nodal displacements, velocities, accelerations and all the other necessary matrices and parameters to unit TAPE16, for a restart.

TAPIN

It is called by subroutine DATAIN. It reads nodal displacements, velocities, accelerations, and all the other necessary matrices and parameters from unit TAPE16 for a restart.

SINGEL

This subroutine is called by subroutine SOLVE. This subroutine is another executive routine calling a number of subroutines which perform the necessary integrations to assemble the singular element mass, damping and stiffness matrices. It also calculates the symmetric and the assymmetric parts of the singular element matrices. This subroutine also calls the subroutine "LINV2F" of the IMSL library of the University of Washington to invert the generally non-symmetric matrix $\underset{\sim}{P}$ of equation (5.3).

LINEI

It is called by subroutine SINGEL. It performs all the necessary integrations along the boundary of the singular element.

STIFK

It is called by subroutine SINGEL. It mainly assembles the singular element stiffness matrix. It also calls the subroutine "LINV2F" of the IMSL Library described above (see description for subroutine SINGEL).

MAREAI

It is called by subroutines SINGEL. It perform the necessary integrations over the surface of the singular element.

MASSM

It is called by subroutine SINGEL. It assembles singular element mass and damping matrices.

INPOL

It is called by subroutine LINEI. It calculates values of interpolating coefficients of boundary displacement functions for the singular element.

TRANS

It is called by subroutines ESOLVE, DATOUT, LINEI and MAREAI. It performs coordinate transformation from $\zeta$, $\eta$ into physical coordinates X,Y and calculates values of differentials for the singular element.

NORMAL

It is called by subroutine LINEI. It calculates the components of a unit vector normal to the sides of the singular element.

PAIR

It is called by subroutine ESOLVE. It eliminates the redundant degrees of freedom for the double nodes in the finite element mesh.

FUNCTS

Is called by subroutines ESOLVE, DATOUT, LINEI and MAREAI. It calculates values of singular element approximating functions and their derivatives.

PRECRCK

Is called by subroutine SOLVE. This subroutine is another executive routine calling a number of subroutines to calculate the complex roots of the characteristic equation for each material and to calculate the complex exponent eigen-values for the assumed eigen-functions of the singular element.

ROOTS

Is called by subroutine PRECRCK. It calculates the complex roots of the characteristic equaiton for a given set of material properties and a crack-tip speed. If the roots can not be found analytically, this subroutine calls the subroutine "PROOT" of the BMATH library of the Boeing Company to find the distinct complex roots by numerical methods.

MULT1

Is called by subroutine PRECRCK. It calculates the values of parameters $K_{ik}$(i=1,8) of equations (A.7a) and (A.7b) for material k whose characteristic equation has two pairs of distinct complex roots.

MULT2

Same as MULT1, except that the characteristic equation for the material has one pair of complex roots of multiplicity 2.

EIGEN

Is called by subroutine PRECRCK. It calls a number of other subroutines and calculates the complex exponent eigen-values for the assumed eigen-functions of the singular element.

INV22

Is called by subroutine EIGEN. It calculates the inverse of a 2x2 complex coefficient matrix to solve for the complex paramters $E_{ik}$ of equation (A.35) for real values of the exponent n.

INV33

Is called by subroutine EIGEN. It calculates the inverse of a

3x3 complex coefficent matrix to solve for the complex parameters $C_{ik}$ and $D_{ik}$ of equation (A.33) for complex values of the exponent n.

MAT

Is called by subroutine EIGEN. It calculates the elements of the 4x4 complex coefficient matrix of equation (A.29).

COEFF

Is called by subroutine FUNCTS. It calls other subroutines to calculate the complex parameters $S_{nk\ell}$ and $D_{mk\ell}$ of equations (A.36a) and (A.36b).

CONS1

Is called by subroutine COEFF. It calculates the values of the complex paramters $F_{ik\ell}$ of equations (A.36a).

CONS2

Same as CONS1 for equation (A.36b).

MULT

Is called by subroutine COEFF. It calculates the complex paramters $S_{nk\ell}$ and $D_{mk\ell}$ of equations (A.36a) and (A.36b) for the assumed singular element eigen-functions. It also calculates similar complex paramters for the derivatives of the assumed singular element eigen-functions to be used by subroutine FUNCTS.

EQUATE

Is called by subroutine PRECRCK. It equates the elements of certain matrices in order to prevent unnecessary calculations in case the singular element is composed of only one material.

# APPENDIX C

## INPUT INSTRUCTIONS

<u>CARD SET 1</u>  <u>Title Card (8A10)</u>

80 column problem identification - Any BCD information. For a restart this card should be the same as the original, otherwise the program will not run.

<u>Note:</u> For a restart CARD SETS 3 through 8 must be eliminated.

<u>CARD SET 2</u>  <u>Control Parameters (8I5, 2E10.0)</u>

NBT — Number of terms for eigen-functions.
If NBT $\leq$ 0, the code sets NBT = 15.
If 0 < NBT < 12, the code sets NBT = 12.
Must have, NBT $\leq$ 18.

NINT — Number of Gaussian points for evaluation of integrals of singular element.
If NINT $\leq$ 0, the code sets NINT = 6.
If 0 < NINT < 4, the code sets NINT = 4.
Must have, NINT $\leq$ 10.

NITER — Number of iterations per cycle for solution convergence.
If NITER < 0, the code sets NITER = 0.
If NITER > 2, the code sets NITER = 2.
Default, NITER = 1.
This input is ignored for stationary cracks, where no iterations are needed.

ICOND — = 0, initial nodal displacements, velocities and accelerations are set to zero by the code.

= 1, static solution is desired (as initial condition or not).

=2, restart run, read initial nodal displacements, velocities and accelerations and also mesh definition cards from unit TAPE16.

NUMPC Number of pressure boundary condition cards. Must have, $1 \leq NUMPC \leq 100$.

NUMLP Number of pressure time-history points. Must have, $2 \leq NUMLP \leq 20$.

NUMCV Number of velocity time-history points. If NUMCV = 0, the code assumes a stationary crack. Otherwise, must have, $2 \leq NUMCV \leq 20$.

NUMPS Number of position time-hisotry points. Must have, $2 \leq NUMPS \leq 20$.

NOTE: For parameters NUMLP, NUMCV (if not zero), NUMPS the minimum value must be 2 for interpolation purposes.

$\beta$ Newmark-$\beta$, if $\beta \leq 0$, the code sets $\beta = .25$.

DT3 Time Step.

If $DT3 \leq 0$, the code calculates a time step from material properties.

NOTE: If ICOND=2, go to card set 9. Do not input card sets 3 through 8.

<u>CARD SET 3</u> <u>Control Parameters (10I5, E10.0)</u>

NUMMAT Number of materials in the mesh. Must have, $NUMMAT \leq 6$.

NELTYP Number of different elements in the mesh (two elements with same geometric shapes but two different materials are considered as two different elements, not one).

NUMNP Number of nodal points (including the double nodes and the two nodes at the center of the singular element). Must have, NUMLP < 300.

Remember, as long as remeshing takes place, the singular element must be a square composed of four square regular elements and the center mesh point on the right side of the singular lement must be double noded.

NUMEL Number of elements (including the four square elements which form the singular element, i.e. the singular element is counted as four element and not one).

Must have, NUMEL < 250.

IALL > 0 all elements are squares.

< 0 non-square elements are present.

ISTAT > 0 only the static solution is desired.

< 0 dynamic solution is also desired.

This input is ignored for ICOND = 0.

NBAND Bandwidth for the singular element. Must have, NBAND < 96. If NBAND < 0, NBAND is calculated by the code.

NBRED Bandwidth for regular elements. Must have, NBRED < 96.

Remember, for the elements adjacent to the interface (i.e. the double noded line) the node on one side of the interface is considered as the node for an element on the other side of the interface.

NBRED must be < NBAND.

If NBRED < 0, NBRED is calculated by the code.

NOTE: NBAND and NBRED are calculated internally and checked against the input values, so it is good practice to input these values to make sure that you and the code understand each other.

IPLANE > 0, plane strain problem.

< 0, plane stress problem.

ISYMT > 1, the problem is symmetric about a vertical axis coinciding with the left side of the mesh. In this case the conditions of symmetry is set by the code and the input values of the array ICODE (see nodal input cards) for nodes on the left side of the mesh are ignored. This parameter is intended to help reduce the number of input cards required in case of a symmetry.

< 0, read the values of ICODE for all nodes from input cards and do not impose any conditions of symmetry.

TT Starting time (i.e. actual starting problem time).

<u>CARD SET 4</u> <u>Material Property Card(s) (8E10.0)</u>

$\rho$ Material mass density.

$a_{11}, a_{12}, a_{16}$, $a_{22}, a_{26}, a_{66}$ Material elastic constants in $\underset{\sim}{\varepsilon} = \underset{\approx}{a}\underset{\sim}{\sigma}$ for a given X′-Y′ coordinate system.

ANG The angle through which the X′, Y′ coordinate system has to rotate to become parallel to the global X-Y coordinate system of the problem.

NOTE: Card set 4 should be repeated NUMMAT times.

CARD SET 5 Material Property Card(s) (4E10.0)

NOTE: This card set must be eleminated for plane stress problems, i.e. for IPLANE < 0.

$a_{13}, a_{23}$, $a_{33}, a_{36}$ Material elastic constants in $\underset{\sim}{\varepsilon} = \underset{\sim}{a}\underset{\sim}{\sigma}$. These are needed for transformation to a plane strain problem.

NOTE: If card set 5 is present, it must be repeated NUMMAT times.

CARD SET 6 Nodal Point Cards (I5, F5.0, 2E10.0, I5)

N Node number.

CODE(N) = 0, the node is not restrained in either X or Y directions.

= 1, the node is restrained in X direction but is free in Y direction.

= 2, the node is restrained in Y direction but is free in X direction.

= 3, the node is restrained in both X and Y directions.

R(N) X-coordinate value.

Z(N) Y-coordinate value.

ND Interval for generation of nodal points, if zero, the interval is set to one by the code.

NOTE: Cards must be in increasing nodal number sequence. When nodal points are skipped in the data file, the program fills the missing data by linear interpolation. First and last nodal cards must be supplied. Repeat these cards until finished.

<u>CARD SET 7</u> <u>Element Cards (8I5)</u>

M Element number.

IX(J,M) J=1,4, nodal point numbers i,j,k,l of element M, starting counterclockwise from bottom left corner.

IX(5,M) Element type number.

IX(6,M) Material type (i.e. element material number).

ND Interval for generation of element cards, if zero, the interval is set to one by the code.

NOTE: Cards must be in increasing element number sequence. When elements are skipped in the data file, the program fills in the missing data by linear interpolation. First and last element cards must be supplied. Repeat these cards until finished.

<u>CARD Set 8</u> <u>Singular Element Card (3I5)</u>

NCR1, NCR2 Element numbers for top and bottom regular elements forming the left half of the singular element.

NELX Increment of element numbers in X-direction, so that one has

NCR3=NCR1 + NELX

NCR4=NCR2 + NELX

where NCR3 and NCR4 are the top and bottom regular elements forming the right half of the singular element.

<u>CARD SET 9</u> <u>Pressure Card(s) (2I5, 3E10.0)</u>

| | |
|---|---|
| INI(K) | Nodal point 1 } Counterclockwise around the elements. |
| JNJ(K) | Nodal point 2 } Counterclockwise around the elements. |
| PI(K) | Pressure multiplier $p_1$ for node 1. |
| PJ(K) | Pressure multiplier $p_2$ for node 2. |
| T(K) | Arrival time of pressure at the center of the element surface. |
| NOTE: | Card set 9 must be repeated NUMPC times. |

<u>CARD SET 10</u> <u>Pressure Time-Hisotry (2E10.0)</u>

| | |
|---|---|
| P(1,M) | Time t. |
| P(2,M) | Pressure value p(t). |
| NOTE: | Repeat NUMLP times. |

<u>CARD SET 11</u> <u>Velocity Time-History (2E10.0)</u>

| | |
|---|---|
| CVH(1,M) | Time t. |
| CVH(2,M) | Velocity c(t). |
| NOTE: | Repeat NUMCV times. |

<u>CARD SET 12</u> <u>Position Time-Hisotry (2E10.0)</u>

| | |
|---|---|
| POST(1,M) | Time t. |
| POST(2,M) | Position f(t). |
| NOTE: | Repeat NUMPS times. |

| CARD Set 13 | Output Parameters (5E10.0/16I5/16I5) |
|---|---|
| TI01 | First time of printed output. |
| TI02 | Last time of printed output. |
| TI03 | Time interval of printed output. |
| SIOD | Time interval for restart tape output. |
| TMACH | Machine STOP time (i.e. computer (cpu) stop time). |
| NDSOUT(I) | I=1,16, nodal numbers for which output (displacements, velocities and accelerations) is desired.<br>If NDSOUT(I)=0, for all I, then all nodes are printed. |
| NSTOUT(I) | I=1,16 element numbers for which output (stresses) is desired.<br>If NSTOUT(I)=0, for all I, then all elements are printed. |

APPENDIX D

COMPUTER PROGRAM LISTING

```
      PROGRAM FEAPROC(INPUT,OUTPUT,TAPE11,TAPE12,TAPE14,TAPE15,
     1 TAPE21,TAPE24,TAPE5=INPUT,TAPE6=OUTPUT,TAPE16)
      COMMON/BK1/NUMMAT,NUMNP,NUMEL,NUMPC,NUMLP, IHED(8),NTAPE,NEQ,ICOND
     1,ISTAT,TMACH, IHED1(8),ISYMT
      COMMON/VEL/CV,NUMCV,CVH(2,20)
      COMMON/POS/NUMPS,POST(2,20)
      COMMON/BK3/NDSOUT(16),NSTOUT(16),TIO1,TIO2,TIOD,SIOD,NUMDS,NUMST
      COMMON/BK10/ R(300),Z(300),CODE(300),IX(6,250)
      COMMON/BK11/DELT,DT1,DT3,BETA,BET1,BET2,BET3,BET4,BET5,NBAND,NBD2
      COMMON/SHIFT/NSBL,NDEL,LASTB,NF,NBRED,NELTYP,ISK,RCODE
      COMMON/RHHO/RHO(6),RODUM
      COMMON/SUMAN/A1(3,3,6),A1I(3,3,6),ADET(6),ASIZE(6),CL
      COMMON/INTGR/PT(10),WG(10),PT2(2),WG2(2),PEI
      COMMON/DIM/NA,NAA,NBT,NB,NQ,NR,NINT,NINT2,IALL,NITER,SINCOD
      COMMON/RIG/IRIG1(6),IRIG2
      COMMON/TOLR/TOLER1,TOLER2
      COMMON/IPLN/IPLN(5),IPLN1(2)
C
C MAX NO. OF MATERIALS ALLOWED IS 6
C TO INCREASE THE NUMBER OF MATERIALS INCREASE THE DIMENSION OF THE ARRAYS
C IN COMMON BLOCKS 'SUMAN,RIG,RHHO' TO APPROPRIATE NUMBER OF DESIRED MATERIALS
C
C MAX NO. OF ELEMENT TYPES ALLOWED IS 50
C TO INCREASE THE NUMBER OF ELEMENT TYPES INCREASE THE DIMENSION OF THE ARRAYS
C IN COMMON BLOCK 'ELM' TO APPROPRIATE NUMBER OF DESIRED ELEMENT TYPES
C
C MAX BANDWIDTH IS 96
C TO INCREASE THE BANDWIDTH INCREASE THE DIMENSION OF THE ARRAY
C IN COMMON BLOCK 'STORE' AND ALSO THE DIMENSIONS OF THE ARRAY 'MS'
C IN SUBROUTINE 'SOLVE'  TO APPROPRIATE DESIRED BANDWIDTH, ALSO CHANGE
C THE VARIABLE 'NBD1' BELOW TO THE DESIRED BANDWIDTH
C
      NBD1=96
      NAA=36
      NQ=20
      NR=2
      IERROR=1
      NTAPE=5
      NINT2=2
      ISK=0
      TOLER1=1.E-6
      TOLER2=1.E-1
      PEI=ACOS(-1.)
      SINCOD=0.
      RCODE=0.
C
C          READ AND WRITE TITLE
C
      READ (NTAPE,110) IHED1
      WRITE (6,160) IHED1
C
C          READ AND WRITE CONTROL PARAMETERS
C
      READ(NTAPE,120) NBT,NINT,NITER,ICOND,NUMPC,NUMLP,NUMCV,NUMPS
     1,BETA,DT3
      IF(NBT.GT.18) GO TO 481
```

```
      IF(NINT.GT.10) GO TO 482
      IF(ICOND.LT.0 .OR. ICOND.GT.2) GO TO 483
      IF(NUMPC.GT.100) GO TO 484
      IF(NUMLP.GT.20) GO TO 485
      IF(NUMCV.GT.20) GO TO 486
      IF(NUMPS.GT.20) GO TO 487
      IF(ICOND.NE.2) READ(NTAPE,125) NUMMAT,NELTYP,NUMNP,NUMEL,IALL
     1,ISTAT,NBAND,NBRED,IPLANE ,ISYMT,TT
      IF(ICOND.EQ.2) GO TO 2
      IF(NUMMAT.GT.6) GO TO 488
      IF(NELTYP.GT.50) GO TO 489
      IF(NUMNP.GT.300) GO TO 490
      IF(NUMEL.GT.250) GO TO 491
      IF(NBAND.GT.NBD1) GO TO 492
  2   IF(NINT.LE.0)   NINT=6
      IF(NBT.LE.0) NBT=15
      IF(NITER.EQ.0) NITER=1
      IF(NITER.LT.0) NITER=0
      IF(BETA.LE.0.)  BETA=.25
      IF(ISTAT.LT.0) ISTAT=0
      IF(ISTAT.GT.0) ISTAT=1
C THE FOLLOWING THREE STATEMENTS ARE ADDED FOR SAFETY, ACCURACY AND SPEED
C THESE STATEMENTS CAN BE REMOVED OR CHANGED FOR EXPERIMENTATIONAL PURPOSES
C NINT=1 WILL MAKE THE MATRIX 'P', WHICH IS NAMED AS 'S2' IN
C SUBROUTINE SINGEL, SINGULAR AND THUS NOT INVERTIBLE
C HOWEVER FOR NINT.LT.4 THE ACCURACY IS LOST
C ALSO FOR NBT.LT.12 THE ACCURACY IS LOST
C NITER.GT.2 DOES NOT IMPROVE THE SOLUTION AND ONLY WASTES COMPUTER TIME
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      IF(NINT.LT.4)   NINT=4
      IF(NBT.LT.12) NBT=12
      IF(NITER.GT.2) NITER=2
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      IF(ICOND.EQ.2) CALL TAPIN(TT)
      NEQ=2*NUMNP
      IF(ICOND.EQ.2) GO TO 22
      IRIG=0
      CL=0.
      DO 20 K=1,NUMMAT
      READ(NTAPE,130) RHO(K),A11,A12,A16,A22,A26,A66,ANG
      WRITE(6,310) K
      WRITE(6,311) RHO(K),ANG
      WRITE(6,312) A11,A12,A16,A12,A22,A26,A16,A26,A66
      IRIG1(K)=0
      IF(ABS(A11)+ABS(A22)+ABS(A66)+ABS(A12)+ABS(A16)+ABS(A26).EQ.0.)
     1IRIG1(K)=1
      IRIG=IRIG+IRIG1(K)
      IF(IRIG1(K).EQ.1) GO TO 18
      IF(IPLANE.LE.0) GO TO 10
      READ(NTAPE,130) A13,A23,A33,A36
      WRITE(6,313) A13,A23,A33,A36
      B11=A11-A13*A13/A33
      B12=A12-A13*A23/A33
      B16=A16-A13*A36/A33
      B22=A22-A23*A23/A33
      B26=A26-A23*A36/A33
      B66=A66-A36*A36/A33
      A11=B11
      A12=B12
      A16=B16
      A22=B22
      A26=B26
      A66=B66
10    IF(ANG.EQ.0.) GO TO 12
```

```
      ANG=ANG*ACOS(-1.)/180.
      C=COS(ANG)
      C2=C*C
      C4=C2*C2
      S=SIN(ANG)
      S2=S*S
      S4=S2*S2
      SC=S*C
      B11=A11*C4+(2.*A12+A66)*S2*C2+A22*S4+2.*(A16*C2+A26*S2)*SC
      B22=A11*S4+(2.*A12+A66)*S2*C2+A22*C4-2.*(A16*S2+A26*C2)*SC
      B12=(A11+A22-2.*A12-A66)*S2*C2+A12+(A26-A16)*(C2-S2)*SC
      B66=4.*(A11+A22-2.*A12-A66)*S2*C2+A66+4.*(A26-A16)*(C2-S2)*SC
      B16=(-2.*A11*C2+2.*A22*S2+(2.*A12+A66)*(C2-S2))*SC+
     1         A16*C2*(C2-3.*S2)+A26*S2*(3.*C2-S2)
      B26=(-2.*A11*S2+2.*A22*C2-(2.*A12+A66)*(C2-S2))*SC+
     1         A16*S2*(3.*C2-S2)+A26*C2*(C2-3.*S2)
      A11=B11
      A12=B12
      A16=B16
      A22=B22
      A26=B26
      A66=B66
   12 A1(1,1,K)=A11
      A1(2,1,K)=A12
      A1(3,1,K)=A16
      A1(1,2,K)=A12
      A1(2,2,K)=A22
      A1(3,2,K)=A26
      A1(1,3,K)=A16
      A1(2,3,K)=A26
      A1(3,3,K)=A66
      ASIZE(K)=(A11+A22+A66)/3.
      A1I(1,1,K)=A22*A66-A26*A26
      A1I(2,1,K)=A16*A26-A12*A66
      A1I(3,1,K)=A12*A26-A16*A22
      ADET(K)=A11*A1I(1,1,K)+A12*A1I(2,1,K)+A16*A1I(3,1,K)
      IF(ADET(K).EQ.0.) WRITE(6,340) K
      IF(ADET(K).EQ.0.) STOP 0
      A1I(1,1,K)=A1I(1,1,K)/ADET(K)
      A1I(2,1,K)=A1I(2,1,K)/ADET(K)
      A1I(3,1,K)=A1I(3,1,K)/ADET(K)
      A1I(2,2,K)=(A11*A66-A16*A16)/ADET(K)
      A1I(3,2,K)=(A16*A12-A11*A26)/ADET(K)
      A1I(3,3,K)=(A11*A22-A12*A12)/ADET(K)
      A1I(1,2,K)=A1I(2,1,K)
      A1I(1,3,K)=A1I(3,1,K)
      A1I(2,3,K)=A1I(3,2,K)
      IF(ISTAT.EQ.1) GO TO 20
      AR1=A1I(1,1,K)
      AR3=A1I(1,3,K)
      AR4=A1I(2,2,K)
      AR5=A1I(2,3,K)
      AR6=A1I(3,3,K)
      BR1=((AR1+AR6)+SQRT((AR1-AR6)**2+4.*AR3**2))/2.
      BR2=((AR4+AR6)+SQRT((AR4-AR6)**2+4.*AR5**2))/2.
      RR1=BR1
      IF(RR1.LT.BR2) RR1=BR2
      RR1=SQRT(RR1/RHO(K))
      IF(CL.LT.RR1) CL=RR1
      GO TO 20
   18 A1(1,1,K)=0.
      A1(2,1,K)=0.
      A1(3,1,K)=0.
      A1(1,2,K)=0.
      A1(2,2,K)=0.
```

```
      A1(3,2,K)=0.
      A1(1,3,K)=0.
      A1(2,3,K)=0.
      A1(3,3,K)=0.
      ASIZE(K)=0.
      ADET(K)=0.
      SA=0.
      A1I(1,1,K)=SA
      A1I(2,1,K)=SA
      A1I(3,1,K)=SA
      A1I(1,2,K)=SA
      A1I(2,2,K)=SA
      A1I(3,2,K)=SA
      A1I(1,3,K)=SA
      A1I(2,3,K)=SA
      A1I(3,3,K)=SA
   20 CONTINUE
      IF(IRIG.EQ.NUMMAT) GO TO 500
      IPLN(1)=4HPLAI
      IPLN(2)=4HN ST
      IPLN(3)=4HRESS
      IPLN(4)=4H PRO
      IPLN(5)=4HBLEM
      IF(IPLANE.GT.0) IPLN(3)=4HRAIN
   22 WRITE(6,301) IPLN
      IF(ICOND.EQ.2) GO TO 24
      WRITE(6,302) ISYMT
   24 WRITE(6,140) NINT,NBT,NITER
      IF(ICOND.EQ.2) GO TO 25
      IPLN1(1)=4H  NO
      IF(IALL.LT.0) IALL=0
      IF(IALL.GT.0) IALL=1
      IF(IALL.GT.0) IPLN1(1)=4H YES
      IF(ICOND.NE.1) GO TO 25
      IPLN1(2)=4H  NO
      IF(ISTAT.GT.0) IPLN1(2)=4H YES
      WRITE (6,260) NUMMAT,NELTYP,NUMNP,NUMEL,IPLN1(1),NUMPC,NUMLP,NUMCV
     1,NUMPS,ICOND,IPLN1(2)
      GO TO 27
   25 ISTAT=0
      WRITE(6,270)NUMMAT,NELTYP,NUMNP,NUMEL,IPLN1(1),NUMPC,NUMLP,NUMCV
     1,NUMPS,ICOND
C
   27 CONTINUE

C     SELECT GAUSSIAN POINTS
      CALL GAUSSPT(NINT2)
      DO 30 I=1,NINT2
      PT2(I)=PT(I)
      WG2(I)=WG(I)
   30 CONTINUE
      CALL GAUSSPT(NINT)
C
C         READ AND WRITE DATA
C
      CALL DATAIN(IERROR,TT,NBD1)
C
C         CHECK FOR DATA ERROR
C
C
      IF(IERROR.EQ.0) GO TO 100
      IF(ICOND.EQ.1) GO TO 65
      IF(TT.GE.TIO2) GO TO 98
      IF(ICOND.EQ.0) GO TO 65
      DO 62 I=1,20
```

```
      IF(IHED(I).NE.IHED1(I)) GO TO 63
   62 CONTINUE
      GO TO 65
   63 WRITE(6,380)
      GO TO 98
C
C          SET CONSTANTS
C
   65 CONTINUE
      WRITE(6,250)
C
      CALL SOLVE(TT)
C
   98 STOP 1
  100 CONTINUE
      STOP 2

  481 WRITE(6,401) NBT
      GO TO 501
  482 WRITE(6,402) NINT
      GO TO 501
  483 WRITE(6,403) ICOND
      GO TO 501
  484 WRITE(6,404) NUMPC
      GO TO 501
  485 WRITE(6,405) NUMLP
      GO TO 501
  486 WRITE(6,406) NUMCV
      GO TO 501
  487 WRITE(6,407) NUMPS
      GO TO 501
  488 WRITE(6,408) NUMMAT
      GO TO 501
  489 WRITE(6,409) NELTYP
      GO TO 501
  490 WRITE(6,410) NUMNP
      GO TO 501
  491 WRITE(6,411) NUMEL
      GO TO 501
  492 WRITE(6,412) NBAND
      GO TO 501
  500 WRITE(6,330)
  501 STOP 3
110   FORMAT (20A4)
120   FORMAT(8I5,2E10.2)
125   FORMAT(10I5,2E10.2)
130   FORMAT(8E10.2)
140   FORMAT(1H0,29X,37HNUMBER OF GAUSIAN POINTS-------------,I4,
     1/1H0,29X,37HNUMBER OF TERMS----------------------,I4,
     2/1H0,29X,37HNUMBER OF ITERATIONS PER TIME STEP---,I4)
160   FORMAT (1H1,20X,20A4,/)
180   FORMAT(//,10X,3HTT=,1PE15.7,5X,3HCV=,1PE15.7,5X,5HCTPX=,
     11PE15.7,5X,5HCTPY=,1PE15.7,5X,5HLOAD=,1PE15.7,//
     2,50X,3HK1=,1PE15.7,10X,3HK2=,1PE15.7/)
250   FORMAT (1H1)
260   FORMAT (/,40X  ,27HNUMBER OF MATERIALS--------,I4/1H0,39X,27HNUMBE
     1R OF ELEMENT TYPES----,I4/1H0,39X,                        27HNUMBE
     2R OF NODAL POINTS-----,I4/1H0,39X,27HNUMBER OF ELEMENTS---------,I
     34/1H0,39X,27HALL ELEMENTS ARE SQUARES---,A4,
     4 /1H0,39X,27HNUMBER OF PRESSURE CARDS---,I4/1H0,39X,27HNUMBER OF L
     5OAD POINTS------,I4/1H0,39X,27HNUMBER OF VELOCITY CARDS---,I4/1H0,
     639X,27HNUMBER OF POSITION CARDS---,I4,/,1H0,39X,27HINITIAL CONDITI
     7ON CODE-----,I4/,1H0,39X,27HSTATIC SOLUTION ONLY-------,A4/)
270   FORMAT (/,40X  ,27HNUMBER OF MATERIALS--------,I4/1H0,39X,27HNUMBE
     1R OF ELEMENT TYPES----,I4/1H0,39X,                        27HNUMBE
```

```
     2R OF NODAL POINTS-----,I4/1H0,39X,27HNUMBER OF ELEMENTS---------,I
     34/1H0,39X,27HALL ELEMENTS ARE SQUARES---,A4,
     4 /1H0,39X,27HNUMBER OF PRESSURE CARDS---,I4/1H0,39X,27HNUMBER OF L
     5OAD POINTS------,I4/1H0,39X,27HNUMBER OF VELOCITY CARDS---,I4/1H0,
     639X,27HNUMBER OF POSITION CARDS---,I4,/,1H0,39X,27HINITIAL CONDITI
     7ON CODE-----,I4/)
 280  FORMAT (
     1 /1H0,39X,27HNUMBER OF PRESSURE CARDS---,I4/1H0,39X,27HNUMBER OF L
     2OAD POINTS------,I4/1H0,39X,27HNUMBER OF VELOCITY CARDS---,I4/1H0,
     339X,27HNUMBER OF POSITION CARDS---,I4,/,1H0,39X,27HINITIAL CONDITI
     4ON CODE-----,I4/)
  301 FORMAT(//,50X,5A4)
  302 FORMAT(/1H0,29X,37HY-AXIS SYMMETRIC CODE-----------------,I4)
  310 FORMAT(/,50X,11HMATERIAL # ,I2)
  311 FORMAT(/,40X,25H DENSITY-----------------,1PE12.4,
     1       /,40X,25H ORTHOTROPIC ANGLE--------,1PE12.4)
  312 FORMAT(/,41X,35HSTRESS-STRAIN CONSTITUTIVE RELATION,
     1/3(40X,3(1PE12.4)/))
  313 FORMAT(26X,4HA13=,1PE12.4,7H    A23=,1PE12.4,7H    A33=,1PE12.4,
     1 7H    A36=,1PE12.4/)
  330 FORMAT(///,40X,42H***** ALL MATERIALS CAN NOT BE RIGID *****)
  340 FORMAT(1H1,///,40X,15H*****ERROR*****,//
     1,40X,49HTHE STRAIN-STRESS CONSTITUTIVE LAW FOR MATERIAL #,I3,//,
     240X,58HGIVES A ZERO DETERMINANT. CHANGE THESE MATERIAL CONSTANTS.)
  380 FORMAT(1H1,////,40X,15H*****ERROR*****,//
     1,40X,38HTHIS IS THE RESTART OF A WRONG PROBLEM,////
     2,40X,37HTHE HEADER CARD FOR THIS PROBLEM WAS:,//,30X,20A4)
C
  401 FORMAT(1H1,///,20X,17H*****THE VALUE OF,I5,63H  FOR 'NBT    ' IS MO
     1OR THAN MAX. ALLOWABLE OF 'NBT    = 18'*****)
  402 FORMAT(1H1,///,20X,17H*****THE VALUE OF,I5,63H  FOR 'NINT   ' IS MO
     1OR THAN MAX. ALLOWABLE OF 'NINT   = 10'*****)
  403 FORMAT(1H1,///,20X,17H*****THE VALUE OF,I5,72H  FOR 'ICOND ' IS OU
     1SIDE THE ALLOWABLE RANGE OF ' 0.LE.ICOND.LE.2 '*****)
  404 FORMAT(1H1,///,20X,17H*****THE VALUE OF,I5,63H  FOR 'NUMPC ' IS MO
     1OR THAN MAX. ALLOWABLE OF 'NUMPC =100'*****)
  405 FORMAT(1H1,///,20X,17H*****THE VALUE OF,I5,63H  FOR 'NUMLP ' IS MO
     1OR THAN MAX. ALLOWABLE OF 'NUMLP = 20'*****)
  406 FORMAT(1H1,///,20X,17H*****THE VALUE OF,I5,63H  FOR 'NUMCV ' IS MO
     1OR THAN MAX. ALLOWABLE OF 'NUMCV = 20'*****)
  407 FORMAT(1H1,///,20X,17H*****THE VALUE OF,I5,63H  FOR 'NUMPS ' IS MO
     1OR THAN MAX. ALLOWABLE OF 'NUMPS = 20'*****)
  408 FORMAT(1H1,///,20X,17H*****THE VALUE OF,I5,63H  FOR 'NUMMAT' IS MO
     1OR THAN MAX. ALLOWABLE OF 'NUMMAT=  6'*****)
  409 FORMAT(1H1,///,20X,17H*****THE VALUE OF,I5,63H  FOR 'NELTYP' IS MO
     1OR THAN MAX. ALLOWABLE OF 'NELTYP= 50'*****)
  410 FORMAT(1H1,///,20X,17H*****THE VALUE OF,I5,63H  FOR 'NUMNP ' IS MO
     1OR THAN MAX. ALLOWABLE OF 'NUMNP =300'*****)
  411 FORMAT(1H1,///,20X,17H*****THE VALUE OF,I5,63H  FOR 'NUMEL ' IS MO
     1OR THAN MAX. ALLOWABLE OF 'NUMEL =250'*****)
  412 FORMAT(1H1,///,20X,17H*****THE VALUE OF,I5,63H  FOR 'NBAND ' IS MO
     1OR THAN MAX. ALLOWABLE OF 'NBAND = 98'*****)
      END
```

```
      SUBROUTINE GAUSSPT(NINT)
      COMMON/INTGR/PT(10),WG(10),PT2(2),WG2(2),PEI
      GO TO(1,2,3,4,5,6,7,8,9,10)NINT
  1   PT(1)=0.
      WG(1)=2.
      GO TO 11
  2   PT(2)=.5773502692
      PT(1)=-PT(2)
      WG(2)=1.
      WG(1)=WG(2)
      GO TO 11
  3   PT(3)=.7745966692
      PT(2)=0.
      PT(1)=-PT(3)
      WG(3)=.5555555556
      WG(2)=.8888888889
      WG(1)=WG(3)
      GO TO 11
  4   PT(4)=.8611363116
      PT(3)=.3399810436
      PT(2)=-PT(3)
      PT(1)=-PT(4)
      WG(4)=.3478548451
      WG(3)=.6521451549
      WG(2)=WG(3)
      WG(1)=WG(4)
      GO TO 11
  5   PT(5)=.9061798459
      PT(4)=.5384693101
      PT(3)=0.
      PT(2)=-PT(4)
      PT(1)=-PT(5)
      WG(5)=.2369268850
      WG(4)=.4786286705
      WG(3)=.5688888889
      WG(2)=WG(4)
      WG(1)=WG(5)
      GO TO 11
  6   PT(6)=.9324695142
      PT(5)=.6612093865
      PT(4)=.2386191861
      PT(3)=-PT(4)
      PT(2)=-PT(5)
      PT(1)=-PT(6)
      WG(6)=.1713244924
      WG(5)=.3607615730
      WG(4)=.4679139346
      WG(3)=WG(4)
      WG(2)=WG(5)
      WG(1)=WG(6)
      GO TO 11
  7   PT(7)=.9491079123
      PT(6)=.7415311856
      PT(5)=.4058451514
      PT(4)=0.
      PT(3)=-PT(5)
      PT(2)=-PT(6)
```

```
      PT(1)=-PT(7)
      WG(7)=.1294849662
      WG(6)=.2797053915
      WG(5)=.3818300505
      WG(4)=.4179591837
      WG(3)=WG(5)
      WG(2)=WG(6)
      WG(1)=WG(7)
      GO TO 11

    8 PT(8)=.9602898565
      PT(7)=.7966664774
      PT(6)=.5255324099
      PT(5)=.1834346425
      PT(4)=-PT(5)
      PT(3)=-PT(6)
      PT(2)=-PT(7)
      PT(1)=-PT(8)
      WG(8)=.1012285363
      WG(7)=.2223810345
      WG(6)=.3137066459
      WG(5)=.3626837834
      WG(4)=WG(5)
      WG(3)=WG(6)
      WG(2)=WG(7)
      WG(1)=WG(8)
      GO TO 11

    9 PT(9)=.9681602395
      PT(8)=.8360311073
      PT(7)=.6133714327
      PT(6)=.3242534234
      PT(5)=0.
      PT(4)=-PT(6)
      PT(3)=-PT(7)
      PT(2)=-PT(8)
      PT(1)=-PT(9)
      WG(9)=.0812743884
      WG(8)=.1806481607
      WG(7)=.2606106964
      WG(6)=.3123470770
      WG(5)=.3302393550
      WG(4)=WG(6)
      WG(3)=WG(7)
      WG(2)=WG(8)
      WG(1)=WG(9)
      GO TO 11

   10 PT(10)=.9739065285
      PT(9)=.8650633667
      PT(8)=.6794095683
      PT(7)=.4333953941
      PT(6)=.1488743390
      PT(5)=-PT(6)
      PT(4)=-PT(7)
      PT(3)=-PT(8)
      PT(2)=-PT(9)
      PT(1)=-PT(10)
      WG(10)=.0666713443
      WG(9)=.1494513492
      WG(8)=.2190863625
      WG(7)=.2692667193
      WG(6)=.2955242247
      WG(5)=WG(6)
      WG(4)=WG(7)
```

```
      WG(3)=WG(8)
      WG(2)=WG(9)
      WG(1)=WG(1Ø)
11    RETURN
      END
```

```
      SUBROUTINE DATAIN(IERROR,TT,NBD1)
C
      COMMON/BK1/NUMMAT,NUMNP,NUMEL,NUMPC,NUMLP, IHED(8),NTAPE,NEQ,ICOND
     1,ISTAT,TMACH, IHED1(8),ISYMT
      COMMON/BK3/NDSOUT(16),NSTOUT(16),TI01,TI02,TIOD,SIOD,NUMDS,NUMST
      COMMON/BK11/DELT,DT1,DT3,BETA,BET1,BET2,BET3,BET4,BET5,NBAND,NBD2
      COMMON/DIM/NA,NAA,NBT,NB,NQ,NR,NINT,NINT2,IALL,NITER,SINCOD
      COMMON/PAIR1/NPAIR,LPAIR(3,40)
      COMMON/BK10/ R(300),Z(300),CODE(300),IX(6,250)
      COMMON/SHIFT/NSBL,NDEL,LASTB,NF,NBRED,NELTYP,ISK,RCODE
      COMMON/OLDDISP/U(600),V(600),A(600)
      COMMON/PRESS/INI(100),JNJ(100),PI(100),PJ(100),T(100),P(2,20),PF
      COMMON/RHHO/RHO(6),RODUM
      COMMON/SUMAN/A1(3,3,6),A1I(3,3,6),ADET(6),ASIZE(6),CL
      COMMON/TIP/NCR1,NCR2,NCR3,NCR4,NELX,CTPX,CTPY,SIF1,SIF2
      COMMON/VEL/CV,NUMCV,CVH(2,20)
      COMMON/POS/NUMPS,POST(2,20)
      COMMON/RIG/IRIG1(6),IRIG2
      COMMON/TOLR/TOLER1,TOLER2
      COMMON/MISL/IF(20)
      COMMON/MAIN/CORD(10,2)
C
      IF(ICOND.EQ.2) GO TO 232
C
C     READ OF NODAL POINT DATA
C
      RMIN=1.E100
      I=0
      ID=1
   20 CONTINUE
      READ (NTAPE,640) N,CODE(N),R(N),Z(N),ND
      IF(R(N) .LT. RMIN) RMIN=R(N)
C
      IF (ND) 60,70,60
   60 ID=ND
   70 IF (I) 80,140,80
   80 NL=N-I
      IF (NL-1) 140,130,90
   90 NL=NL/ID
      IF (I+NL*ID-N) 520,100,520
  100 IF(CODE(I).NE.CODE(N)) GO TO 522
      IF (NL-1) 130,130,110
  110 ANL=NL
      DR=(R(N)-R(I))/ANL
      DZ=(Z(N)-Z(I))/ANL
      NL=N-2*ID
      DO 120 J=I,NL,ID
      I1=J+ID
      R(I1)=R(J)+DR
      Z(I1)=Z(J)+DZ
  120 CODE(I1)=CODE(J)
  130 IF (NUMNP-N) 530,150,140
  140 I=N
      GO TO 20
  150 CONTINUE
C
C     READ OF ELEMENT DATA
C
      NUMAT1=0
      NELTP1=0
      ID=1
      I=0
  190 READ(NTAPE,650)M,(IX(J,M),J=1,6),ND
      IF(NELTP1.LT.IX(5,M)) NELTP1=IX(5,M)
      IF(NUMAT1.LT.IX(6,M)) NUMAT1=IX(6,M)
```

```
      IF(ND) 195,198,195
 195  ID=ND
 198  I=I+1
      IF(M-I) 540,220,210
 210  IF(IX(5,M).NE.IX(5,M1)) GO TO 545
      IF(IX(6,M).NE.IX(6,M1)) GO TO 545
      IF((IX(1,M)-ID1)/(M-M1) .NE. ID) GO TO 540
      IF((IX(2,M)-ID2)/(M-M1) .NE. ID) GO TO 540
      IF((IX(3,M)-ID3)/(M-M1) .NE. ID) GO TO 540
      IF((IX(4,M)-ID4)/(M-M1) .NE. ID) GO TO 540
 215  IX(1,I)=IX(1,I-1)+ID
      IX(2,I)=IX(2,I-1)+ID
      IX(3,I)=IX(3,I-1)+ID
      IX(4,I)=IX(4,I-1)+ID
      IX(5,I)=IX(5,I-1)
      IX(6,I)=IX(6,I-1)
      I=I+1
      IF (M-I) 220,220,215
 220  M1=M
      ID1=IX(1,M1)
      ID2=IX(2,M1)
      ID3=IX(3,M1)
      ID4=IX(4,M1)
      IF (NUMEL-M) 550,230,190
 230  CONTINUE
      IF(NELTYP.LE.0) NELTYP=NELTP1
      IF(NELTYP.NE.NELTP1) GO TO 572
      IF(NUMMAT.NE.NUMAT1) GO TO 571
      IF(NELTYP.LT.NUMMAT) GO TO 575
C
C   READ INITIAL SPECIAL ELEMENT NUMBERS
      READ(NTAPE,650) NCR1,NCR2,NELX
      NCR3=NCR1+NELX
      NCR4=NCR2+NELX
C CHECK FOR THE SINGULAR ELEMENT TO BE INSIDE THE MESH
      IF(NELX.EQ.0) GO TO 586
      IF(NCR1.LE.0 .OR. NCR1 .GT. NUMEL) GO TO 586
      IF(NCR2.LE.0 .OR. NCR2 .GT. NUMEL) GO TO 586
      IF(NCR3.LE.0 .OR. NCR3 .GT. NUMEL) GO TO 586
      IF(NCR4.LE.0 .OR. NCR4 .GT. NUMEL) GO TO 586
C CHECK TO SEE IF MATERIALS IN SINGULAR ELEMENT IS CORRECT
      IF(IX(6,NCR1).NE.IX(6,NCR3)) GO TO 579
      IF(IX(6,NCR2).NE.IX(6,NCR4)) GO TO 579
C
      CTPY=Z(IX(4,NCR2))
C
      IF(2)=IX(4,NCR2)*2
      IF(4)=IX(1,NCR2)*2
      IF(6)=IX(2,NCR2)*2
      IF(8)=IX(2,NCR4)*2
      IF(10)=IX(3,NCR4)*2
      IF(12)=IX(2,NCR3)*2
      IF(14)=IX(3,NCR3)*2
      IF(16)=IX(4,NCR3)*2
      IF(18)=IX(4,NCR1)*2
      IF(20)=IX(1,NCR1)*2
      IF(1)=IF(2)-1
      IF(3)=IF(4)-1
      IF(5)=IF(6)-1
      IF(7)=IF(8)-1
      IF(9)=IF(10)-1
      IF(11)=IF(12)-1
      IF(13)=IF(14)-1
      IF(15)=IF(16)-1
      IF(17)=IF(18)-1
```

```
      IF(19)=IF(20)-1
      MAX=IF(2)
      MIN=IF(2)
      DO 2323 I=4,NQ,2
      IF(IF(I).GT.MAX) MAX=IF(I)
      IF(IF(I).LT.MIN) MIN=IF(I)
 2323 CONTINUE
      NBAND1=MAX-MIN+2
      IF(NBAND.LE.0) NBAND=NBAND1
      IF(NBAND1.NE.NBAND) GO TO 573
      IF(NBAND.GT.NBD1) GO TO 583
      NBD2=2*NBAND
      NBRED1=IABS(IF(4)-2*IX(2,NCR1))+2
      NBRED2=IABS(IF(20)-IF(6))+2
      IF(NBRED1.LT.NBRED2) NBRED1=NBRED2
      IF(NBRED.LE.0) NBRED=NBRED1
      IF(NBRED1.NE.NBRED) GO TO 574
      IF(NBRED.GT.NBAND) STOP 13
      WRITE(6,947) NBAND,NBRED
      NSBL=(MAX-1)/NBAND+1
      NDEL=(NSBL-1)*NBAND+2-MIN
      LASTB=(NEQ+NDEL-1)/NBAND+1
      NF=LASTB*NBAND
      DO 2325 I=2,NQ,2
      K=IF(I)/2
      II=I/2
      CORD(II,1)=R(K)
      CORD(II,2)=Z(K)
 2325 CONTINUE
      IF(IRIG1(IX(6,NCR1)).EQ.1 .AND. IRIG1(IX(6,NCR2)).EQ.1) GO TO 578
      IRIG2=0
      IF(IRIG1(IX(6,NCR1)).EQ.1 .OR. IRIG1(IX(6,NCR2)).EQ.1) IRIG2=1
 232  CONTINUE
      IF(IRIG2.EQ.1) NBT=(NBT+1)/2
      NA=NBT*2
      NB=NA
      IF(ISTAT.EQ.1) DT1=DT3
      IF(ISTAT.EQ.1) GO TO 2329
      CLENGT=ABS(R(IF(10)/2)-R(IF(2)/2))/2.
      DT2=(CLENGT/2.)/CL
 2329 CONTINUE
C
C  READ OF PRESSURE B.C. DATA
      IF (NUMPC.LT.1) GO TO 592
      DO 2408 K=1,NUMPC
      READ (NTAPE,660) INI(K),JNJ(K),PI(K),PJ(K),T(K)
      KFF=0
      DO 2407 J=1,NUMEL
      IF(J.EQ.NCR1) GO TO 2403
      IF(J.EQ.NCR3) GO TO 2402
      IF((INI(K).EQ.IX(1,J) .AND. JNJ(K).EQ.IX(2,J)) .OR.
     1(INI(K).EQ.IX(2,J) .AND. JNJ(K).EQ.IX(1,J))) KFF=1
      IF(J.EQ.NCR2) GO TO 2404
 2402 IF((INI(K).EQ.IX(2,J) .AND. JNJ(K).EQ.IX(3,J)) .OR.
     1(INI(K).EQ.IX(3,J) .AND. JNJ(K).EQ.IX(2,J))) KFF=1
      IF(J.EQ.NCR4) GO TO 2407
 2403 IF((INI(K).EQ.IX(3,J) .AND. JNJ(K).EQ.IX(4,J)) .OR.
     1(INI(K).EQ.IX(4,J) .AND. JNJ(K).EQ.IX(3,J))) KFF=1
      IF(J.EQ.NCR3) GO TO 2407
 2404 IF((INI(K).EQ.IX(4,J) .AND. JNJ(K).EQ.IX(1,J)) .OR.
     1(INI(K).EQ.IX(1,J) .AND. JNJ(K).EQ.IX(4,J))) KFF=1
 2407 CONTINUE
      IF(KFF.EQ.0) GO TO 564
 2408 CONTINUE
```

```
C  READ OF PRESSURE LOAD HISTORY
      IF (NUMLP.LT.2) GO TO 592
      M=0
      NUMLP1=NUMLP
      DO 235 N=1,NUMLP1
      M=M+1
      READ (NTAPE,670) (P(K,M),K=1,2)
      IF(M.EQ.1) GO TO 235
      IF(P(1,M).LT.P(1,M-1)) GO TO 593
      IF(P(1,M).NE.P(1,M-1) .OR. P(2,M).NE.P(2,M-1)) GO TO 235
      M=M-1
      NUMLP=NUMLP-1
  235 CONTINUE
      IF (NUMLP.LT.2) GO TO 592
C
C  READ OF CRACK TIP VELOCITY HISTORY
      IF(NUMCV.EQ.0) GO TO 2361
      IF(NUMCV.LT.2) GO TO 597
      M=0
      NUMCV1=NUMCV
      DO 2355 N=1,NUMCV1
      M=M+1
      READ(NTAPE,670) (CVH(K,M),K=1,2)
      IF(CVH(2,M).LT.0.) GO TO 598
      IF(M.EQ.1) GO TO 2355
      IF(CVH(1,M).LT.CVH(1,M-1)) GO TO 598
      IF(CVH(1,M).NE.CVH(1,M-1) .OR. CVH(2,M).NE.CVH(2,M-1)) GO TO 2355
      M=M-1
      NUMCV=NUMCV-1
 2355 CONTINUE
      IF(NUMCV.LT.2) GO TO 597
      IF(ISTAT.EQ.1) GO TO 239
      VMAX=CVH(2,1)
      DO 236 I=2,NUMCV
      IF(CVH(2,I) .GT. VMAX) VMAX=CVH(2,I)
  236 CONTINUE
      IF(VMAX.EQ.0.) GO TO 2361
      DT4=(CLENGT/10.)/VMAX
      DT6=DT4
      DT7=DT4
      IF(DT6.GT.DT2) DT6=DT2
      IF(DT7.LT.DT2) DT7=DT2
      GO TO 2362
 2361 IF(ISTAT.EQ.1) GO TO 239
      DT6=DT2
      DT7=DT2
 2362 IF(DT3.LE.0.) DT1=DT6
      IF(DT3.LE.0.) GO TO 239
      IF(DT3.LT.TOLER2*DT6) WRITE(6,988) DT3
      IF(DT3.GT.2.*DT7) WRITE(6,989) DT3
      DT1=DT3
  239 CONTINUE
      WRITE(6,980) BETA,DT1
C  READ OF CRACK TIP POSITION HISTORY
      IF (NUMPS.LT.2) GO TO 594
      M=0
      NUMPS1=NUMPS
      DO 240 N=1,NUMPS1
      M=M+1
      READ (NTAPE,670) (POST(K,M),K=1,2)
      IF(M.EQ.1) GO TO 240
      IF(POST(1,M).LT.POST(1,M-1)) GO TO 596
      IF(POST(2,M).LT.POST(2,M-1)) GO TO 596
      IF(POST(1,M).NE.POST(1,M-1).OR.POST(2,M).NE.POST(2,M-1))GO TO 240
      M=M-1
```

```
      NUMPS=NUMPS-1
  240 CONTINUE
      IF (NUMPS.LT.2) GO TO 594
C
      IF(ICOND.EQ.2) GO TO 299
      XL=R(IX(2,NCR1))-R(IX(1,NCR1))
      IF(XL.EQ.0.) GO TO 589
C
C  CORRECT THE VALUE OF "CODE" FOR NODES ON THE AXIS OF SYMMETRY FOR
C  SYMMETRIC PROBLEMS
      IF(ISYMT.LE.0) GO TO 246
      DO 245 I=1,NUMNP
      RR=(R(I)-RMIN)/XL
      IF (ABS(RR).LT.TOLER1 .AND. CODE(I).EQ.0.) CODE(I)=1.
      IF (ABS(RR).LT.TOLER1 .AND. CODE(I).EQ.2.) CODE(I)=3.
  245 CONTINUE
  246 CONTINUE
C
C   FIND CRACK-TIP POSITION
      CALL POSIT(TT,ILP,1)
C CHECK FOR THE CRACK-TIP TO BE INSIDE THE SINGULAR ELEMENT
      IF(R(IX(4,NCR2)) .GT. CTPX) GO TO 587
      IF(R(IX(3,NCR4)) .LT. CTPX) GO TO 587
C CHECK TO SEE IF ALL THE ELEMENTS ARE SQURE AND EQUAL DIMENSION
      DO 2461 I=1,NUMEL
      M1=IX(1,I)
      M2=IX(2,I)
      M3=IX(3,I)
      M4=IX(4,I)
      R1=R(M1)
      R2=R(M2)
      R3=R(M3)
      R4=R(M4)
      ZZ1=Z(M1)
      ZZ2=Z(M2)
      ZZ3=Z(M3)
      ZZ4=Z(M4)
      IF(ABS(ABS((R2-R1)/XL)-1.) .GT. TOLER1) GO TO 2462
      IF(ABS((R3-R2)/XL) .GT. TOLER1) GO TO 2462
      IF(ABS(ABS((R3-R4)/XL)-1.) .GT. TOLER1) GO TO 2462
      IF(ABS((R4-R1)/XL) .GT. TOLER1) GO TO 2462
      IF(ABS((ZZ2-ZZ1)/XL) .GT. TOLER1) GO TO 2462
      IF(ABS(ABS((ZZ3-ZZ2)/XL)-1.) .GT. TOLER1) GO TO 2462
      IF(ABS((ZZ3-ZZ4)/XL) .GT. TOLER1) GO TO 2462
      IF(ABS(ABS((ZZ4-ZZ1)/XL)-1.) .GT. TOLER1) GO TO 2462
 2461 CONTINUE
      IALL1=1
      GO TO 2463
 2462 CONTINUE
      IALL1=0
 2463 CONTINUE
      IF(IALL1.NE.IALL) GO TO 591
      IF(IALL.EQ.1) GO TO 2466
C CHECK FOR THE ELEMENTS OF THE SINGULAR ELEMENT TO BE SQUARES
      DO 2465 M=1,4
      IF(M.EQ.1) I=NCR1
      IF(M.EQ.2) I=NCR2
      IF(M.EQ.3) I=NCR3
      IF(M.EQ.4) I=NCR4
      M1=IX(1,I)
      M2=IX(2,I)
      M3=IX(3,I)
      M4=IX(4,I)
      R1=R(M1)
      R2=R(M2)
```

```
      R3=R(M3)
      R4=R(M4)
      ZZ1=Z(M1)
      ZZ2=Z(M2)
      ZZ3=Z(M3)
      ZZ4=Z(M4)
      IF(ABS(ABS((R2-R1)/XL)-1.) .GT. TOLER1) GO TO 589
      IF(ABS((R3-R2)/XL) .GT. TOLER1) GO TO 589
      IF(ABS(ABS((R3-R4)/XL)-1.) .GT. TOLER1) GO TO 589
      IF(ABS((R4-R1)/XL) .GT. TOLER1) GO TO 589
      IF(ABS((ZZ2-ZZ1)/XL) .GT. TOLER1) GO TO 589
      IF(ABS(ABS((ZZ3-ZZ2)/XL)-1.) .GT. TOLER1) GO TO 589
      IF(ABS((ZZ3-ZZ4)/XL) .GT. TOLER1) GO TO 589
      IF(ABS(ABS((ZZ4-ZZ1)/XL)-1.) .GT. TOLER1) GO TO 589
 2465 CONTINUE
 2466 CONTINUE
C CHECK FOR THE SINGULAR ELEMENT TO BE A SQUARE(I.E. CHECK TO SEE IF THE
C ELEMENTS OF THE SINGULAR ELEMENT ARE NUMBERD CORRECTLY)
      M1=IX(1,NCR2)
      M2=IX(2,NCR4)
      M3=IX(3,NCR3)
      M4=IX(4,NCR1)
      R1=R(M1)
      R2=R(M2)
      R3=R(M3)
      R4=R(M4)
      ZZ1=Z(M1)
      ZZ2=Z(M2)
      ZZ3=Z(M3)
      ZZ4=Z(M4)
      XL2=2.*XL
      IF(ABS(ABS((R2-R1)/XL2)-1.) .GT. TOLER1) GO TO 589
      IF(ABS((R3-R2)/XL2) .GT. TOLER1) GO TO 589
      IF(ABS(ABS((R3-R4)/XL2)-1.) .GT. TOLER1) GO TO 589
      IF(ABS((R4-R1)/XL2) .GT. TOLER1) GO TO 589
      IF(ABS((ZZ2-ZZ1)/XL2) .GT. TOLER1) GO TO 589
      IF(ABS(ABS((ZZ3-ZZ2)/XL2)-1.) .GT. TOLER1) GO TO 589
      IF(ABS((ZZ3-ZZ4)/XL2) .GT. TOLER1) GO TO 589
      IF(ABS(ABS((ZZ4-ZZ1)/XL2)-1.) .GT. TOLER1) GO TO 589
C
C   CORRECT THE VALUES OF "CODE" FOR NODES OF THE RIGID MATERIAL
C
      DO 270 I=1,NUMEL
      IIS=IX(6,I)
      IF(IRIG1(IIS).EQ.0) GO TO 270
      DO 265 J=1,4
      CODE(IX(J,I))=3.
 265  CONTINUE
 270  CONTINUE
C
C   WRITE OF NODAL POINT DATA
      MPRINT=0
      J=0
      DO 286 N=1,NUMNP
      J=J+1
      IF (MPRINT.NE.0) GO TO 284
      IF (NUMNP.LT.J+50.AND.J.GT.1) GO TO 288
      IF (NUMNP.GT.J+49.AND.J.GT.1) J=J+50
      WRITE (6,730)
      MPRINT=50
 284  MPRINT=MPRINT-1
      NN=J+50
      IF (NUMNP.LT.NN) NN=NUMNP
      IF (J.GT.NUMNP) GO TO 288
 286  WRITE (6,740) (I,CODE(I),R(I),Z(I),I=J,NN,50)
```

```
  288  CONTINUE
 C
 C   WRITE OF ELEMENT DATA
       MPRINT=0
       J=0
       DO 295 N=1,NUMEL
       J=J+1
       IF (MPRINT.NE.0) GO TO 292
       IF (NUMEL.LT.J+50.AND.J.GT.1) GO TO 298
       IF (NUMEL.GT.J+49.AND.J.GT.1) J=J+50
       WRITE (6,750)
       MPRINT=50
  292  MPRINT=MPRINT-1
       NN=J+50
       IF (NUMEL.LT.NN) NN=NUMEL
       IF (J.GT.NUMEL) GO TO 298
  295  WRITE (6,760) (I,(IX(K,I),K=1,6),I=J,NN,50)
  298  CONTINUE
 C
  299 CONTINUE
 C  WRITE SPECIAL ELEMENT NUMBERS
       WRITE(6,765) NCR1,NCR2,NCR3,NCR4
 C
 C  WRITE OF PRESSURE B.C. DATA AND PRESSURE LOAD HISTORY
       IF (NUMPC.EQ.0) GO TO 325
       WRITE (6,770)
       NN=NUMPC
       IF (NUMLP.GT.NN) NN=NUMLP
       K=1
  300  WRITE(6,600)
       IF (K.GT.NUMPC) GO TO 310
       WRITE (6,780) INI(K),JNJ(K),PI(K),PJ(K),T(K)
  310  IF (K.GT.NUMLP) GO TO 320
       WRITE (6,790) P(1,K),P(2,K)
  320  K=K+1
       IF (K.LE.NN) GO TO 300

 C  WRITE OF CRACK TIP VELOCITY HISTORY
  325  IF(NUMCV.EQ.0) GO TO 328
       WRITE(6,775)
       WRITE(6,776) ((CVH(K,M),K=1,2),M=1,NUMCV)
 C  WRITE OF CRACK TIP POSITION HISTORY
  328  WRITE(6,777)
       DO 329 M=1,NUMPS
       WRITE(6,778) (POST(K,M),K=1,2),CTPY
  329  CONTINUE
  330  CONTINUE
       IF(ICOND.EQ.2) GO TO 444
 C
 C   CORRECT THE VALUE OF "CODE" FOR THE INTERNAL NODES OF
 C   THE SINGULAR ELEMENT
       CODE(IX(2,NCR1))=3.
       CODE(IX(3,NCR2))=3.
 C
 C  FIND THE DOUBEL NODES
       MPAIR=0
       NUMNP1=NUMNP-1
       DO 255 I=1,NUMNP1
       R1=R(I)
       Z1=Z(I)
       II=I+1
       DO 254 J=II,NUMNP
       R2=R(J)
       Z2=Z(J)
       SR=ABS((R1-R2)/XL)
```

```
      SZ=ABS((Z1-Z2)/XL)
      IF(SR.GT.TOLER1 .OR. SZ.GT.TOLER1) GO TO 254
      IF(ABS((Z1-CTPY)/XL).GT.TOLER1) GO TO 577
      IF(R1.LT. R(IX(3,NCR4))) GO TO 254
      MPAIR=MPAIR+1
      NPAIR=MPAIR+MPAIR
      LPAIR(1,NPAIR-1)=I+I-1
      LPAIR(2,NPAIR-1)=J+J-1
      LPAIR(1,NPAIR)=I+I
      LPAIR(2,NPAIR)=J+J
      LPAIR(3,NPAIR-1)=1
      LPAIR(3,NPAIR)=1
      IF(CODE(I).EQ.3) CODE(J)=3
      IF(CODE(J).EQ.3) CODE(I)=3
      IF(CODE(I).NE.CODE(J)) GO TO 512
      CODE(I)=3.
  254 CONTINUE
  255 CONTINUE
      IF(MPAIR.LT.1) GO TO 581
C
C  WRITE OF DOUBEL NODES
      WRITE(6,766)NPAIR/2
      WRITE(6,767) (LPAIR(1,J)/2,J=2,NPAIR,2)
      WRITE(6,768) (LPAIR(2,J)/2,J=2,NPAIR,2)
C CHECK TO SEE IF THE NODAL POINTS OF THE ELEMENTS ARE NUMBERED CORRECTLY
        NUMEL1=NUMEL-1
        DO 259 I=1,NUMEL1
        II=I+1
        DO 258 J=1,4
        IJS=IX(J,I)
        R1=R(IJS)
        Z1=Z(IJS)
        DO 257 K=II,NUMEL
        DO 256 L=1,4
        KL1=IX(L,K)
        KL2=IX(L+1,K)
        IF(L.EQ.4) KL2=IX(1,K)
        R2=R(KL1)
        Z2=Z(KL1)
        R3=R(KL2)
        Z3=Z(KL2)
        R12=(R1-R2)/XL
        Z12=(Z1-Z2)/XL
        R13=(R1-R3)/XL
        Z13=(Z1-Z3)/XL
        ZR=ABS(Z13*R12-Z12*R13)
        IF(ZR.GT.TOLER1) GO TO 256
        IF(ABS(R12).LT.TOLER1) GO TO 256
        IF(ABS(R13).LT.TOLER1) GO TO 256
        IF(R1.LT.R2 .AND. R1.GT.R3) GO TO 576
        IF(R1.LT.R3 .AND. R1.GT.R2) GO TO 576
 256    CONTINUE
 257    CONTINUE
 258    CONTINUE
 259    CONTINUE
C
C   FIND CRACK-TIP POSITION AND RE-MESH IF NECESSARY
      CALL POSIT(TT,ILP,0)
      IF(ILP.EQ.1) GO TO 588
C
      IF(ICOND.EQ.1) GO TO 442
C
C  INITIAL CONDITIONS FOR ICOND=0
 332  DO 335 I=1,NEQ
      U(I)=0.
```

```
      V(I)=0.
      A(I)=0.
  335 CONTINUE
      SIF1=0.
      SIF2=0.
  442 CONTINUE
      CALL LOAD(TT,TTP,2,0)
      CALL VELOC(TT,TTV,1,0)
      IF(CV.NE.0.) GO TO 582
      IF(ICOND.EQ.1) GO TO 444
      IF(PF.NE.0.) GO TO 584
  444 CONTINUE
C
C     READ AND WRITE OF PRINTED OUTPUT PARAMETERS
C
      READ(NTAPE,690)TIO1,TIO2,TIOD,SIOD,TMACH,NDSOUT,NSTOUT
      IF(SIOD.EQ.0.) SIOD=TIOD
      IF(TMACH.EQ.0.) TMACH=3600.
      NUMDS=0
      NUMST=0
      DO 500 I=1,16
      IF (NDSOUT(I).NE.0) NUMDS=NUMDS+1
      IF (NSTOUT(I).NE.0) NUMST=NUMST+1
  500 CONTINUE
      IF (NUMDS.EQ.0) NUMDS=NUMNP
      IF (NUMST.EQ.0) NUMST=NUMEL
      WRITE (6,825) TT
      WRITE (6,830)
      WRITE (6,840) TIO1,TIO2,TIOD,SIOD,TMACH
      WRITE (6,850) NUMDS,NUMST
      WRITE (6,700)
      IF(ICOND.NE.2) GO TO 501
      CALL PRECRCK(CV,NUMMAT,TT,IL)
      IF(IL.EQ.1) STOP 14
  501 IF(ICOND.NE.1) CALL DATOUT(0,TT)
C
  510 RETURN
C
C     INPUT DATA ERROR EXITS
C
  512 WRITE (6,990) I,J
      GO TO 595
  520 WRITE (6,870) I,ND,N
      GO TO 595
  522 WRITE (6,872) I,N
      GO TO 595
  530 WRITE (6,880) N,NUMNP
      GO TO 595
  540 WRITE (6,890) I,M
      GO TO 595
  545 WRITE (6,895) M,M1
      GO TO 595
  550 WRITE (6,900) M,NUMEL
      GO TO 595
  564 WRITE (6,912) INI(K),JNJ(K)
      GO TO 595
  571 WRITE (6,922)
      GO TO 595
  572 WRITE (6,924)
      GO TO 595
  573 WRITE (6,926)
      GO TO 595
  574 WRITE (6,928)
      GO TO 595
  575 WRITE (6,927)
```

```
      GO TO 595
 576  WRITE (6,933) I,K
      GO TO 595
 577  WRITE (6,932) I,J
      GO TO 595
 578  WRITE (6,929)
      GO TO 595
 579  WRITE (6,935)
      GO TO 595
 581  WRITE (6,937)
      GO TO 595
 582  WRITE (6,940)
      GO TO 595
 583  WRITE (6,945) NBAND,NBD1
      GO TO 595
 584  WRITE (6,950)
      GO TO 595
 586  WRITE (6,960)
      GO TO 595
 587  WRITE (6,965)
      GO TO 595
 588  WRITE (6,970)
      GO TO 595
 589  WRITE (6,975)
      GO TO 595
 591  IF(IALL1.EQ.0) WRITE (6,978) I
      IF(IALL1.EQ.1) WRITE (6,979)
      GO TO 595
 592  WRITE (6,981)
      GO TO 595
 593  WRITE (6,982)
      GO TO 595
 594  WRITE (6,983)
      GO TO 595
 596  WRITE (6,984)
      GO TO 595
 597  WRITE (6,985)
      GO TO 595
 598  WRITE (6,986)
 595  IERROR=0
      GO TO 510
C
 600  FORMAT (1H )
 640  FORMAT (I5,F5.0,2E10.0,I5)
 650  FORMAT (8I5)
 660  FORMAT (2I5,3E10.0)
 670  FORMAT (2E10.0)
 671  FORMAT (2E10.0,2I10)
 680  FORMAT (I10,6E10.0,I10)
 690  FORMAT (5E10.0,/16I5/16I5)
 700  FORMAT (1H1)
 730  FORMAT (1H1//44X,23HNODAL POINT COORDINATES///8X,2HNP,4X,4HCODE,8X
     1,5HR-ORD,11X,5HZ-ORD,14X,2HNP,4X,4HCODE,8X,5HR-ORD,11X,5HZ-ORD/)
 740  FORMAT (1X,2(4X,I5,F8.1,2E16.4,4X))
 750  FORMAT (1H1//46X,19HELEMENT DEFINITIONS///6X,7HELEMENT,5X,1HI,5X,1
     11HJ,5X,1HK,5X,1HL,3X,7HELMTYPE,3X,7HMATTYPE,10X,7HELEMENT,5X
     2,1HI,5X,1HJ,5X,1HK,5X,1HL,3X,7HELMTYPE,3X,7HMATTYPE)
 760  FORMAT (2(5X,I5,3X,4I6,I7,I10,7X))
 765  FORMAT(//25X,17HELEMENTS NUMBERED,I5,I5,I5,6H   AND,I5,23H   ARE S
     1PECIAL ELEMENTS)
 766  FORMAT(1H1,//,20X,9HTHERE ARE,I6,13H DOUBEL NODES)
 767  FORMAT(//1X,19HNODES OF FIRST  ROW,20I5)
 768  FORMAT(//1X,19HNODES OF SECOND ROW,20I5)
 770  FORMAT (1H1//11X,28HPRESSURE BOUNDARY CONDITIONS,30X,28HPRESSURE H
     1ISTORY DESCRIPTION///5X,1HI,5X,1HJ,7X,4HPI/P,8X,4HPJ/P,6X,10HSTART
```

```
     2 TIME,23X,4HTIME,9X,10HPRESSURE P/)
 775  FORMAT(1H1//40X,26HCRACK TIP VELOCITY HISTORY,///37X,
     14HTIME,10X,18HCRACK TIP VELOCITY/)
 776  FORMAT(31X,1PE15.7,8X,1PE15.7)
 777  FORMAT(1H1//40X,26HCRACK TIP POSITION HISTORY,///20X,
     14HTIME,10X,20HCRACK-TIP POSITION-R,10X,20HCRACK-TIP POSITION-Z,/)
 778  FORMAT(14X,1PE15.7,8X,1PE15.7,15X,1PE15.7)
 780  FORMAT (1H+,I5,I6,2F12.3,E12.4)
 790  FORMAT (1H+,65X,1P2E15.7)
 825  FORMAT(1H1,//,45X,14HCURRENT TIME =,1PE15.7,//)
 830  FORMAT (//,40X,25HPRINTED OUTPUT PARAMETERS)
 840  FORMAT (/,40X,22HSTART OUTPUT AT-------,1PE15.7//,40X,22HSTOP OUTP
     1UT AT--------,1PE15.7//,40X,22HSTEP OUTPUT AT--------,1PE15.7,////
     2,40X,22HSTEP TAPE OUTPUT AT---,1PE15.7,////
     3,40X,22HMACHINE STOP TIME-----,1PE15.7,////)
 850  FORMAT (/,40X,35HNODAL POINTS TO BE PRINTED---------,I5//,40X,
     135HELEMENT STRESSES TO BE PRINTED-----,I5)
 870  FORMAT (23H INCREMENTING FROM N.P.,I4,3H BY,I3,22H,S WILL NOT REAC
     1H N.P.,I4)
 872  FORMAT (22H CODE FOR NODAL POINTS,I5,5H  AND,I5,
     1              37H  ARE NOT THE SAME,CANNOT INTERPOLATE)
 880  FORMAT (5H N.P.,I4,23H IS GREATER THAN NUMNP=,I4)
 890  FORMAT (25H ELEMENT DEFINITION CARDS,2I4,13H OUT OF ORDER)
 895  FORMAT (54H ELEMENT TYPE AND/OR MATERIAL TYPE ON DEFINITION CARDS,
     1 I4,5H  AND,I4,37H  ARE NOT THE SAME,CANNOT INTERPOLATE)
 900  FORMAT (19H ELEMENT DEFINITION,I4,23H IS GREATER THAN NUMEL=,I4)
 912  FORMAT (25H *PRESSURE CARD FOR NODES,I4,5H  AND,I4,11H  IS WRONG*)
 920  FORMAT (23H INCREMENTING FROM N.P.,I4,3H BY,I3,22HS WILL NOT REACH
     1  N.P.,I4)
 922  FORMAT(1H1,//,30X,44H***INPUT FOR NUMBER OF MATERIALS IS WRONG***)
 924  FORMAT(1H1,//,30X,44H*INPUT FOR NUMBER OF ELEMENT TYPES IS WRONG*)
 926  FORMAT(1H1,//,30X,54H**INPUT FOR BANDWIDTH OF SINGULAR ELEMENTS IS
     1 WRONG***)
 927  FORMAT(1H1,//,30X,86H***INPUT FOR "NUMMAT AND/OR NELTYP AND/OR IX(
     15,I),IX(6,I) ON ELEMENT CARDS IS WRONG***)
 928  FORMAT(1H1,//,30X,54H***INPUT FOR BANDWIDTH OF REGULAR ELEMENTS IS
     1 WRONG***)
 929  FORMAT(///,40X,65H*****BOTH MATERIALS OF THE SINGULAR ELEMENT CAN
     1NOT BE RIGID*****)
 930  FORMAT (5H N.P.,I4,23H IS GREATER THAN NUMNP=,I4)
 932  FORMAT (1H1,///,20X,28H****NODAL CARDS WRONG, NODES,I5,I5,60H  NOT
     1 ON THE CRACK-LINE HAVE THE IDENTICALL COORDINATES*****)
 933  FORMAT (1H1,///,20X,55H****ELEMENT CARDS WRONG, THE NODAL NUMBERS
     1FOR ELEMENTS,I5,I5,15H  ARE WRONG****)
 935  FORMAT (1H1,//,30X,39H*****THE SINGULAR ELEMENT IS WRONG*****,//
     1,10X,54HTHE MATERIAL TYPES IN SINGULAR ELEMENT ARE NOT CORRECT)
 937  FORMAT (1H1,//,30X,39H*****THE SINGULAR ELEMENT IS WRONG*****,//
     1,10X,75HAT THE LEAST, NODES 5 AND 6 OF THE SINGULAR ELEMENT HAVE T
     2O BE DOUBLE NODED)
 940  FORMAT (1H1,//,20X,74H*****FOR ICOND=0 OR ICOND=1 INITIAL CRACK-TI
     1P VELOCITY HAS TO BE ZERO*****)
 945  FORMAT(/1H1,//,20X,62H*****THE BAND WIDTH OF THE PROBLEM FOR THE S
     1INGULAR ELEMENT IS,I5,/,20X,46HWHICH IS MORE THAN THE ALLOWABLE BA
     2ND WIDTH OF,I5,18H  IN THIS PROGRAM.,/10X,122HYOU NEED TO INCREASE
     3 THE SIZE OF  MATRIX "SK" IN COMMON BLOCK "STORE" AND ALSO THE SIZ
     4E OF MATRIX "SM" IN SUBROUTINE FORMK)
 947  FORMAT(30X,
     137HBAND WIDTH FOR SINGULAR ELEMENT------,I4,/1H0,29X,37HBAND WIDTH
     2 FOR REGULAR ELEMENTS------,I4)
 950  FORMAT (1H1,//,20X,49H*****FOR ICOND=0 INITIAL LOAD HAS TO BE ZERO
     1*****)
 960  FORMAT (1H1,//,30X,39H*****THE SINGULAR ELEMENT IS WRONG*****,//
     1,10X,5HNELX=,I3,5X,5HNCR1=,I3,5X,5HNCR2=,I3,5X,5HNCR3=,I3
     2,5X,5HNCR4=,I3,5X,6HNUMEL=,I3)
 965  FORMAT (1H1,//,30X,39H*****THE SINGULAR ELEMENT IS WRONG*****,//
```

```
      1,30X,58H*****THE CRACK-TIP IS NOT INSIDE THE SINGULAR ELEMENT*****
      2)
  970 FORMAT (1H1,//,20X,91H*****THE CRACK-TIP LOCATION HAS PAST THE CEN
     1TER POINT OF THE LAST ELEMENT ON THE CRACK-LINE,/,30X,
     247HIN THE MESH AND RE-MESHING IS NOT POSSIBLE*****)
  975 FORMAT (1H1,//,30X,39H*****THE SINGULAR ELEMENT IS WRONG*****,//
     1,30X,44H*****THE SINGULAR ELEMENT IS NOT SQUARE*****)
  978 FORMAT (1H1,//,15X,44H*****INPUT FOR "IALL" IS "1" BUT ELEMENT NO.
     1,I5,20H  IS NOT SQUARE*****,//
     2,15X,41HIF THIS IS THE CASE SET "IALL=0" ON INPUT)
  979 FORMAT (1H1,//,15X,66H*****INPUT FOR "IALL" IS "0" BUT ALL THE ELE
     1MENTS ARE SQUARES*****,//
     2,15X,41HIF THIS IS THE CASE SET "IALL=1" ON INPUT)
  980 FORMAT(/20X,6HBETA =,1PE15.7,7X,11HTIME STEP =,1PE15.7)
  981 FORMAT(1H1,////,20X,94H*****PRESSURE CARDS MUST BE PRESENT WITH "N
     1UMPC.GE.1"  AND "NUMLP.GE.2" FOR INTERPOLATION*****)
  982 FORMAT(1H1,////,20X,69H*****PRESSURE CARDS P(1,K),K=1,NUMLP MUST B
     1E IN INCREASING ORDER*****)
  983 FORMAT(1H1,////,20X,76H*****POSITION CARDS MUST BE PRESENT WITH "N
     1UMPS.GE.2" FOR INTERPOLATION*****)
  984 FORMAT(1H1,////,5X,119H*****POSITION CARDS POST(1,K),K=1,NUMPS MUS
     1T BE IN INCREASING ORDER AND CRACK-TIP POSITION MUST INCREASE WITH
     2 TIME*****)
  985 FORMAT(1H1,////,20X,88H*****IF VELOCITY CARDS ARE PRESENT, THEN WE
     1MUST HAVE "NUMCV.GE.2" FOR INTERPOLATION*****)
  986 FORMAT(1H1,////,5X,115H*****VELOCITY CARDS CVH(1,K),K=1,NUMCV MUST
     1 BE IN INCREASING ORDER AND CRACK-TIP VELOCITY MUST BE NON-NEGATIV
     2E*****)
  988 FORMAT(1H1,////,5X,19H*****A TIME STEP OF,1PE15.5,70H     MAYBE TOO
     1 SMALL AND WILL PROBABLY CREATE NUMERICAL DIFFICULTY*****)
  989 FORMAT(1H1,////,5X,19H*****A TIME STEP OF,1PE15.5,66H     MAYBE TOO
     1 LARGE AND WILL PROBABLY CREATE UNRIABLE RESULTS*****)
  990 FORMAT(1H1,////,5X,10H*****NODES,I5,I5,71H  ARE DOUBLE NODES BUT T
     1HEY HAVE DIFFERENT VALUES FOR THEIR 'CODE'*****)
C
      END
```

```
      SUBROUTINE SOLVE(TT)
C
      COMMON/BK1/NUMMAT,NUMNP,NUMEL,NUMPC,NUMLP, IHED(8),NTAPE,NEQ,ICOND
     1,ISTAT,TMACH, IHED1(8),ISYMT
      COMMON/BK3/NDSOUT(16),NSTOUT(16),TIO1,TIO2,TIOD,SIOD,NUMDS,NUMST
      COMMON/BK10/ R(300),Z(300),CODE(300),IX(6,250)
      COMMON/OLDDISP/U(600),V(600),A(600)
      COMMON/RHHO/RHO(6),RODUM
      COMMON/SUMAN/A1(3,3,6),A1I(3,3,6),ADET(6),ASIZE(6),CL
      COMMON/BK11/DELT,DT1,DT3,BETA,BET1,BET2,BET3,BET4,BET5,NBAND,NBD2
      COMMON/TIP/NCR1,NCR2,NCR3,NCR4,NELX,CTPX,CTPY,SIF1,SIF2
      COMMON/VEL/CV,NUMCV,CVH(2,20)
      COMMON/POS/NUMPS,POST(2,20)
      COMMON/PRESS/INI(100),JNJ(100),PI(100),PJ(100),T(100),P(2,20),PF
      COMMON/ELM/ESTIFM(8,8,50),ELMASS(8,8,50),VSS(20,20)
      COMMON/MMAT/SMS(20,20),V1S(20,20),AK4S(20,20)
      COMMON/BOUND/NFIX(200),NBC(200),NBB
      COMMON/SHIFT/NSBL,NDEL,LASTB,NF,NBRED,NELTYP,ISK,RCODE
      COMMON/DIM/NA,NAA,NBT,NB,NQ,NR,NINT,NINT2,IALL,NITER,SINCOD
      COMMON/STORE/SK(192,96)
      COMMON/MISL/IF(20)
      COMMON/TOLR/TOLER1,TOLER2
      REAL MS(192,96)
      EQUIVALENCE(MS(1,1),SK(1,1))
C
      COMMON/SSS/S1S(36,20),S1SD(36,20),S1SDD(36,20)
      COMMON/MASS/AMRR(36,36),AMRDR(36,36),AM22(2,2),AMR2(36,2)
     1,AMRD2(36,2),AMRDD2(36,2)
      COMMON/WK/WKAREA(1404)
      COMMON/LON/S2(36,36),S4(36,36),S2D(36,36),S2DD(36,36)
      DIMENSION B(800),ELOAD(800),EM(800),EMAT(800)
      DIMENSION US(600),VS(600),AS(600)
      DIMENSION EX1(2160),EX3(1404),EX5(5184)
      EQUIVALENCE(B(1),EX1(1))
      EQUIVALENCE(ELOAD(1),EX1(1001))
      EQUIVALENCE(EMAT(1),EX3(1))
      EQUIVALENCE(US(1),EX5(1))
      EQUIVALENCE(VS(1),EX5(1001))
      EQUIVALENCE(AS(1),EX5(2001))
      EQUIVALENCE(EM(1),EX5(3001))
      EQUIVALENCE(S1S(1,1),EX1(1))
      EQUIVALENCE(WKAREA(1),EX3(1))
      EQUIVALENCE(S2(1,1),EX5(1))
C
      TIOT=TIO1
      IF(TIOT.LT.TT) TIOT=TT
      IF(ICOND.NE.1) TIOT=TIOT+TIOD
      SIOT=TT
      IF(ICOND.NE.1) SIOT=SIOT+SIOD
      NBG=NBAND
      I=0
      DO 10 N=1,NUMNP
      IPHI=IFIX(CODE(N))
      IF(IPHI.EQ.0) GO TO 10
      I=I+1
      NBC(I)=N
      IF(IPHI.EQ.1) NFIX(I)=10
      IF(IPHI.EQ.2) NFIX(I)=01
      IF(IPHI.EQ.3) NFIX(I)=11
10    CONTINUE
      NBB=I
      RODUM=1.
      KJR=0
      IF(ICOND.EQ.1 .OR. (ICOND.EQ.2 .AND. SINCOD.EQ.0.)) KJR=1
```

```
C
C          OBTAIN STATIC SOLUTION
C
      IF(ICOND.NE.1) GO TO 50
      WRITE(6,2000)
      CV=0.
      IF(ISTAT.GT.0) RODUM=0.
      CALL PRECRCK(CV,NUMMAT,TT,IL)
      IF(IL.EQ.1) RETURN
      CALL SINGEL
      CALL FORMK(4)
      RODUM=1.
      CALL LOAD(TT,TTP,3,LL)
      DO 33 I=1,NF
      R(I)=RLOAD(I)
   33 CONTINUE
      CALL PAIR(0,0)
      CALL RSOLVE(0,0)
      DO 40 N=1,NEQ
      U(N)=US(N)
      V(N)=0.
      A(N)=0.
   40 CONTINUE
      CALL SECOND(TMACH1)
      IF(ISTAT.LE.0) GO TO 42
      CALL DATOUT(0,TT)
      RETURN
   42 IF(TT.LT.TIO2 .AND. TMACH1.LT.TMACH) GO TO 45
      CALL DATOUT(0,TT)
      CALL TAPOUT(TT)
      RETURN
   45 IF(TT.LT.TIOT) GO TO 48
      CALL DATOUT(0,TT)
      WRITE(6,406)
      TIOT=TIOT+TIOD
   48 CALL TAPOUT(TT)
      SIOT=SIOT+SIOD
   50 CONTINUE
C
C          LOOP ON TIME STEP
C
      IJUMP1=0
      IJUMP=0
      LOOP=0
   60 CONTINUE
C
      ITER=0
      LOOP=LOOP+1
      TTEMP=TT
      CTPX1=CTPX
      CV1=CV
C
C LOOK FOR JUMP CONDITION AND CALCULATE PROPER TIME,LOAD,AND VELOCITY
C
      CALL VELOC(TT,TTV,0,LV)
      CALL LOAD(TT,TTP,0,LL)
C  LOAD AND VELOCITY JUMP       OR    VELOCITY JUMP ONLY
      IF(LV.EQ.1) GO TO 61
C  LOAD JUMP ONLY
      IF(LV.EQ.0 .AND. LL.EQ.1) GO TO 64
C  NO JUMP
      TT=TTV
      IF(TTV.GT.TTP) TT=TTP
      DELT=TT-TTEMP
      DELEP=DELT/DT1
```

```
      LOOP=LOOP-1
      IF(DELEP.LT.TOLER2) GO TO 60
      LOOP=LOOP+1
      IJUMP=0
      CALL VELOC(TT,TTV,1,LV)
      IF((LOOP.EQ.1 .AND. ICOND.NE.1) .OR. CV.NE.CV1) GO TO 601
      GO TO 70
  601 WRITE(6,406)
      CALL PRECRCK(CV,NUMMAT,TT,IL)
      IF(IL.EQ.1) RETURN
      GO TO 70
   61 CALL VELOC(TT,TTV,1,LV)
      WRITE(6,406)
      CALL PRECRCK(CV,NUMMAT,TT,IL)
      IF(IL.EQ.1) RETURN
      CALL SINGEL
      CALL FORMK(2)
      CALL FORMK(3)
      CALL LOAD(TT,TTP,1,LL)
      GO TO 312
   64 IF(LOOP.EQ.1) GO TO 644
      CALL LOAD(TT,TTP,1,LL)
      GO TO 312
  644 IF(ICOND.EQ.1) GO TO 6444
      WRITE(6,406)
      CALL PRECRCK(CV,NUMMAT,TT,IL)
      IF(IL.EQ.1) RETURN
      IF(KJR.EQ.1) GO TO 6444
      CALL SINGEL
 6444 CALL FORMK(2)
      CALL FORMK(3)
      CALL LOAD(TT,TTP,1,LL)
      GO TO 312

C  END OF LOOK FOR JUMP CONDITION AND FINDING PROPER TIME

C       UPDATE CRACK TIP LOCATION

   70 CALL POSIT(TT,ILP,0)
      IF(ILP.EQ.1) RETURN
      BET1=DELT*DELT*BETA
      BET2=DELT*BETA
      BET3=(.5-BETA)/BETA
      BET4=(.5-BETA)*DELT*DELT
      BET5=(.25-BETA)*DELT**3
C
      IF(CTPX.NE.CTPX1) GO TO 71
      IF(CV.NE.CV1) GO TO 71
      IF(LOOP.EQ.1 .AND. KJR.EQ.1) GO TO 72
      IF(LOOP.EQ.1) GO TO 71
      GO TO 73
   71 CALL SINGEL
   72 CALL FORMK(2)
      CALL FORMK(3)
   73 CALL FORMK(1)
C
C         UPDATE SURFACE TRACTIONS
C
   75 CALL LOAD(TT,TTP,1,LL)
C         MODIFY EXTERNAL FORCE VECTOR FOR ASYMMETRIC STIFFNESS AND
C
      DO 82 I=1,NF
      EM(I)=0.
   82 CONTINUE
      IF(LOOP.EQ.1 .AND. ICOND.EQ.0) GO TO 120
```

```
      DO 110 I=1,NQ
      II=IF(I)+NDEL
      DO 105 J=1,NQ
      JJ=IF(J)
      EM(II)=EM(II)-(V1S(I,J)*V(JJ)+AK4S(I,J)*U(JJ))
  105 CONTINUE
      EM(II)=EM(II)*BET1
  110 CONTINUE
  120 CONTINUE
C
      MM=0
      REWIND 14
  140 CONTINUE
      MM=MM+1
      NB1=(MM-1)*NBAND+1
      NB2=MM*NBAND
      DO 150 I=NB1,NB2
      EMAT(I)=0.
  150 CONTINUE
      IF(MM.EQ.1) NB1=NB1+NDEL
      IF(MM.EQ.LASTB) NB2=NEQ+NDEL
      K=NBAND+1
      READ(14)  ((MS(I,J),I=K,NBD2),J=1,NBG)
      CALL MATPROD(NB1,NB2,MM,1)
      IF(MM.EQ.LASTB) GO TO 162
      DO 160 J=1,NBG
      DO 160 I=1,NBAND
      MS(I,J)=MS(I+NBAND,J)
  160 CONTINUE
      GO TO 140
  162 CONTINUE
C
C          END LOOP ON BLOCKS
C
      IF(CV .EQ. 0.)GO TO 182
      DO 180 I=1,NQ
      II=IF(I)+NDEL
      DO 180 J=1,NQ
      JJ=IF(J)
      EMAT(II)=EMAT(II)+VSS(I,J)*(U(JJ)*DELT*.5+V(JJ)*BET4+A(JJ)*BET5)
  180 CONTINUE
  182 CONTINUE
      DO 190 N=1,NEQ
      II=N+NDEL
      EMAT(II)=EMAT(II)+BET1*ELOAD(II)
  190 CONTINUE
  192 CONTINUE
      DO 195 N=1,NF
      E(N)=EMAT(N)+EM(N)
  195 CONTINUE
C
C          SOLVE SYSTEM OF EQUATIONS
C
      IF(ITER.EQ.0) CALL PAIR(0,0)
      IF(ITER.GT.0) CALL PAIR(0,1)
      CALL ESOLVE(0,1)
      IF(CV .EQ. 0.)GO TO 200
      IF(ITER.GE.NITER) GO TO 200
      GO TO 220
 200  DO 210 I=1,NEQ
      U(I)=US(I)
      V(I)=VS(I)
      A(I)=AS(I)
 210  CONTINUE
      GO TO 320
```

```
  220  ITER=ITER+1
       WRITE(6,2001) ITER,TT,CV,CTPX,CTPY,PF
       WRITE(6,408) SIF1,SIF2
C
C          CALCULATE ERROR VECTOR
C
       DO 230 I=1,NF
       EM(I)=0.
  230  CONTINUE
       DO 305 I=1,NQ
       II=IF(I)+NDEL
       DO 300 J=1,NQ
       JJ=IF(J)
       EM(II)=EM(II)-(V1S(I,J)*VS(JJ)+AK4S(I,J)*US(JJ))
  300  CONTINUE
       EM(II)=EM(II)*BET1
  305  CONTINUE
       GO TO 192
C
C  THIS PORTION SOLVES FOR A JUMP CONDITION
C
  312  CONTINUE
       DELT=TT-TTEMP
       REWIND 15
       IJUMP1=1
       IJUMP=1
       MM=0
       DO 3121 I=1,NEQ
       US(I)=U(I)
 3121  CONTINUE
  313  CONTINUE
       MM=MM+1
       NB1=(MM-1)*NBAND+1
       NB2=MM*NBAND
       DO 314 I=NB1,NB2
  314  EMAT(I)=0.
       IF(MM.EQ.1) NB1=NB1+NDEL
       IF(MM.EQ.LASTB) NB2=NEQ+NDEL
       K=NBAND+1
       READ(15) ((SK(I,J),I=K,NBD2),J=1,NBG)
       CALL MATPROD(NB1,NB2,MM,2)
       IF(MM.EQ.LASTB) GO TO 316
       DO 315 J=1,NBG
       DO 315 I=1,NBAND
       SK(I,J)=SK(I+NBAND,J)
  315  CONTINUE
       GO TO 313
  316  CONTINUE
C
       IF(CV .EQ. 0.)GO TO 318
       IF(LOOP.EQ.1 .AND. ICOND.EQ.0)GO TO 318
       DO 317 I=1,NQ
       II=IF(I)+NDEL
       DO 317 J=1,NQ
       JJ=IF(J)
       EMAT(II)=EMAT(II)+(V1S(I,J)+VSS(I,J))*V(JJ)+AK4S(I,J)*U(JJ)
  317  CONTINUE
  318  CONTINUE
       DO 319 I=1,NF
       E(I)=ELOAD(I)-EMAT(I)
  319  CONTINUE
       CALL PAIR(1,0)
       CALL ESOLVE(1,1)
       DO 1210 I=1,NEQ
```

```
      A(I)=US(I)
 1210 CONTINUE
      GO TO 321
C
C  END OF SOLUTION FOR A JUMP CONDITION
C

C
C          ASSIGN DISPLACEMENTS,VELOCITIES AND ACCELLERATIONS
C
  320 IJUMP1=0
  321 CONTINUE
      CALL SECOND(TMACH1)
C
C          PRINT SOLUTION
C
      IF(TT.LT.TIO2 .AND. TMACH1.LT.TMACH) GO TO 340
      CALL DATOUT(1,TT)
      CALL TAPOUT(TT)
      RETURN
  340 IF(TT.LT.TIOT) GO TO 350
      CALL DATOUT(1,TT)
      TIOT=TIOT+TIOD
      WRITE(6,406)
  350 IF(TT.LT.SIOT) GO TO 355
      CALL TAPOUT(TT)
      SIOT=SIOT+SIOD
  355 CONTINUE
      GO TO 60
C
C          END LOOP ON TIME STEP
C
  400 FORMAT(5X,1P10E12.4)
  406 FORMAT(1H1)
  408 FORMAT(/,40X,6H   K1=,1PE12.4,6H   K2=,1PE12.4)
 2000 FORMAT(///,40X,37HSTATIC SOLUTION FOR INITIAL CONDITION)
 2001 FORMAT(1H1,//,10X,47HTHE STRESS INTENSITY FACTORS BEFORE ITERATION
     1 #,I3,////,10X,12HCURRENT TIME,10X,18HCRACK-TIP VELOCITY,10X,
     220HCRACK-TIP POSITION-R,10X,20HCRACK-TIP POSITION-Z,10X,4HLOAD,/,
     310X,1PE13.4,12X,1PE13.4,16X,1PE13.4,16X,1PE13.4,9X,1PE13.4)
C
      END
```

```
      SUBROUTINE MATPROD(N1,N2,MM,LL)
C
      COMMON/SHIFT/NSBL,NDEL,LASTB,NF,NBRED,NELTYP,ISK,RCODE
      COMMON/BK1/NUMMAT,NUMNP,NUMEL,NUMPC,NUMLP, IHED(8),NTAPE,NEQ,ICOND
     1,ISTAT,TMACH, IHED1(8),ISYMT
      COMMON/BK11/DELT,DT1,DT3,BETA,BET1,BET2,BET3,BET4,BET5,NBAND,NBD2
      COMMON/STORE/B(192,96)
      COMMON/OLDDISP/U(600),V(600),A(600)
C
      COMMON/SSS/S1S(36,20),S1SD(36,20),S1SDD(36,20)
      COMMON/MASS/AMRR(36,36),AMRDR(36,36),AM22(2,2),AMR2(36,2)
     1,AMRD2(36,2),AMRDD2(36,2)
      COMMON/WK/WKAREA(1404)
      COMMON/LON/S2(36,36),S4(36,36),S2D(36,36),S2DD(36,36)
      DIMENSION E(800),ELOAD(800),EM(800),EMAT(800)
      DIMENSION US(600),VS(600),AS(600)
      DIMENSION EX1(2160),EX3(1404),EX5(5184)
      EQUIVALENCE(E(1),EX1(1))
      EQUIVALENCE(ELOAD(1),EX1(1001))
      EQUIVALENCE(EMAT(1),EX3(1))
      EQUIVALENCE(US(1),EX5(1))
      EQUIVALENCE(VS(1),EX5(1001))
      EQUIVALENCE(AS(1),EX5(2001))
      EQUIVALENCE(EM(1),EX5(3001))
      EQUIVALENCE(S1S(1,1),EX1(1))
      EQUIVALENCE(WKAREA(1),EX3(1))
      EQUIVALENCE(S2(1,1),EX5(1))
C
      NBA=NBRED
      IF(MM.EQ.NSBL) NBA=NBAND
      NMAX=2*NBA-1
      DO 100 I=N1,N2
      IGF=I-NBA-NDEL
      IF(IGF.LT.0) IGF=0
      L=I+1-NDEL
      IF(L.GT.NBA+1) L=NBA+1
      DO 90 J=1,NMAX
      IGF=IGF+1
      K=IGF+NDEL
      IF(K.GT.I) GO TO 60
      L=L-1
      GO TO 70
   60 CONTINUE
      K=I
      L=L+1
   70 CONTINUE
      IF(L.GT.NBA) GO TO 100
      IF(IGF.GT.NEQ) GO TO 100
      KK=K-(MM-2)*NBAND
      IF(LL.EQ.2) GO TO 80
      EMAT(I)=EMAT(I)+B(KK,L)*(U(IGF)+DELT*V(IGF)+BET4*A(IGF))
      GO TO 90
   80 CONTINUE
      EMAT(I)=EMAT(I)+B(KK,L)*US(IGF)
   90 CONTINUE
  100 CONTINUE
      RETURN
C
      END
```

```
      SUBROUTINE FORMK(L)
C
      COMMON/BK1/NUMMAT,NUMNP,NUMEL,NUMPC,NUMLP, IHED(8),NTAPE,NEQ,ICOND
     1,ISTAT,TMACH, IHED1(8),ISYMT
      COMMON/DIM/NA,NAA,NBT,NB,NQ,NR,NINT,NINT2,IALL,NITER,SINCOD
      COMMON/BK11/DELT,DT1,DT3,BETA,BET1,BET2,BET3,BET4,BET5,NBAND,NBD2
      COMMON/TIP/NCR1,NCR2,NCR3,NCR4,NELX,CTPX,CTPY,SIF1,SIF2
      COMMON/ELM/ESTIFM(8,8,50),ELMASS(8,8,50),VSS(20,20)
      COMMON/MMAT/SMS(20,20),V1S(20,20),AK4S(20,20)
      COMMON/KMAT/SKS(20,20)
      COMMON/SHIFT/NSBL,NDEL,LASTB,NF,NBRED,NELTYP,ISK,RCODE
      COMMON/BK10/R(300),Z(300),CODE(300),IX(6,250)
      COMMON/MISL/IF(20)
      COMMON/STORE/SK(192,96)
      COMMON/VEL/CV,NUMCV,CVH(2,20)
      COMMON/RHHO/RHO(6),RODUM
      DIMENSION IXX(100)
C
C   L=1 CALCULATES K EFFEECTIVE MATRIX AND WRITES THE RESULT ON TAPE11
C   L=2 CALCULATES MASS MATRIX AND WRITES THE RESULT ON TAPE14
C   L=3 CALCULATES K MATRIX AND WRITES THE RESULT ON TAPE15
C   L=4 CALCULATES K MATRIX, WHICH IS THE SAME AS K EFFECTIVE FOR STATIC
C   CASE, BUT WRITES THE RESULTS ON TAPE11, WHICH IS FOR K EFFECTIVE.
C
      REWIND 11
      REWIND 14
      REWIND 15
      NBG=NBAND
      DO 10 J=1,NBG
      DO 10 I=1,NBD2
   10 SK(I,J)=0.
      MM=0
      ISING=0
      NBAL2=0
   20 CONTINUE
      NBAL1=NBAL2
C
C          LOOP ON BLOCKS
C
      MM=MM+1
      NBAL2=((MM+1)*NBAND-NDEL)/2
      KS=0
      DO 30 I=1,NUMEL
      IN=IX(1,I)
      DO 25 J=2,4
      IF(IX(J,I).GT.IN) IN=IX(J,I)
   25 CONTINUE
      IF(IN.GT.NBAL1 .AND. IN.LE.NBAL2)GO TO 27
      GO TO 30
   27 KS=KS+1
      IXX(KS)=I
   30 CONTINUE
      DO 230 IP=1,KS
      NE=IXX(IP)
      IF(NE.EQ.NCR1) GO TO 33
      IF(NE.EQ.NCR2) GO TO 33
      IF(NE.EQ.NCR3) GO TO 33
      IF(NE.EQ.NCR4) GO TO 33
      GO TO 120
   33 IF(ISING .EQ. 0) GO TO 50
      GO TO 120
C
C          STORE SPECIAL ELEMENT MATRICES
C
```

```
   50 CONTINUE
      ISING=1
      DO 110 I=1,NQ
      NROWB=IF(I)-(MM-1)*NBAND+NDEL
      DO 110 J=1,NQ
      NCOL=IF(J)-(MM-1)*NBAND-NROWB+1+NDEL
      IF(NCOL) 100,100,70
   70 CONTINUE
      IF(L.EQ.2) GO TO 80
      IF(L.GE.3) GO TO 90
      IF(CV .NE. 0.)GO TO 75
      SK(NROWB,NCOL)=SK(NROWB,NCOL)+(SKS(I,J)*BET1+SMS(I,J))
      GO TO 100
   75 SK(NROWB,NCOL)=SK(NROWB,NCOL)+(SKS(I,J)*BET1+VSS(I,J)*DELT*.5
     1+SMS(I,J))
      GO TO 100
   80 CONTINUE
      SK(NROWB,NCOL)=SK(NROWB,NCOL)+SMS(I,J)
      GO TO 100
   90 SK(NROWB,NCOL)=SK(NROWB,NCOL)+SKS(I,J)
  100 CONTINUE
  110 CONTINUE
      GO TO 220
  120 CONTINUE
      IF(NE.EQ.NCR1) GO TO 220
      IF(NE.EQ.NCR2) GO TO 220
      IF(NE.EQ.NCR3) GO TO 220
      IF(NE.EQ.NCR4) GO TO 220
      IF(ISK.EQ.1) GO TO 150
      IF(ICOND.EQ.2 .AND. RCODE.EQ.0.) GO TO 140
      IF(ICOND.EQ.2 .AND. IALL .EQ. 1) GO TO 140
      CALL STIFEL
      IF(ICOND.EQ.2) GO TO 140
      IF(ISTAT.LE.0) CALL MASSEL
  140 CONTINUE
      ISK=1
  150 CONTINUE
      II=IX(5,NE)
C
C          STORE SK
C
      DO 210 JJ=1,4
      NN=IX(JJ,NE)
      NROWB=(NN-1)*2-(MM-1)*NBAND+NDEL
      DO 210 J=1,2
      NROWB=NROWB+1
      I=(JJ-1)*2+J
      DO 200 KK=1,4
      NN=IX(KK,NE)
      NCOLB=(NN-1)*2+NDEL
      DO 190 K=1,2
      M=(KK-1)*2+K
      NCOL=NCOLB+K+1-NROWB-(MM-1)*NBAND
      IF(NCOL) 190,190,160
  160 CONTINUE
      IF(L.EQ.2) GO TO 170
      IF(L.GE.3) GO TO 180
      SK(NROWB,NCOL)=SK(NROWB,NCOL)+ESTIFM(I,M,II)*BET1+ELMASS(I,M,II)
      GO TO 190
  170 CONTINUE
      SK(NROWB,NCOL)=SK(NROWB,NCOL)+ELMASS(I,M,II)
      GO TO 190
  180 CONTINUE
      SK(NROWB,NCOL)=SK(NROWB,NCOL)+ESTIFM(I,M,II)
```

```
  190 CONTINUE
  200 CONTINUE
  210 CONTINUE
  220 CONTINUE
  230 CONTINUE
      IF(L.EQ.2) GO TO 240
      IF(L.EQ.3) GO TO 250
      WRITE(11) ((SK(I,J),I=1,NBAND),J=1,NBG)
      GO TO 260
  240 WRITE(14) ((SK(I,J),I=1,NBAND),J=1,NBG)
      GO TO 260
  250 WRITE(15) ((SK(I,J),I=1,NBAND),J=1,NBG)
  260 CONTINUE
      IF(LASTB.EQ.1) RETURN
C
C         SHIFT UP ONE BLOCK
C
      DO 330 I=1,NBAND
      K=I+NBAND
      DO 330 J=1,NBG
      SK(I,J)=SK(K,J)
  330 SK(K,J)=0.
      IF(MM.EQ.LASTB-1) GO TO 360
C
C         RETURN FOR NEXT BLOCK
C
      GO TO 20
  360 IF(L.EQ.2) GO TO 370
      IF(L.EQ.3) GO TO 375
      WRITE(11) ((SK(I,J),I=1,NBAND),J=1,NBG)
      GO TO 380
  370 WRITE(14) ((SK(I,J),I=1,NBAND),J=1,NBG)
      GO TO 380
  375 WRITE(15) ((SK(I,J),I=1,NBAND),J=1,NBG)
  380 CONTINUE
      RETURN
C
      END
```

```
      SUBROUTINE ESOLVE(L,LP)
C
      COMMON/PAIR1/NPAIR,LPAIR(3,40)
      COMMON/BK1/NUMMAT,NUMNP,NUMEL,NUMPC,NUMLP, IHED(8),NTAPE,NEQ,ICOND
     1,ISTAT,TMACH, IHED1(8),ISYMT
      COMMON/BK11/DELT,DT1,DT3,BETA,BET1,BET2,BET3,BET4,BET5,NBAND,NBD2
      COMMON/IPAR/PJ1,PJ2,PJ3,PJ4,DXY1,DXY2
      COMMON/SUMAN/A1(3,3,6),A1I(3,3,6),ADET(6),ASIZE(6),CL
      COMMON/BOUND/NFIX(200),NBC(200),NBB
      COMMON/SHIFT/NSBL,NDEL,LASTB,NF,NBRED,NELTYP,ISK,RCODE
      COMMON /C13/SI(12),SIX(12),SIXX(12)
     1           ,DIS(8),DISX(8),DISXX(8)
      COMMON /C10/LAM(20)
      COMMON/MAIN/CORD(10,2)
      COMMON/STORE/A(192,96)
      COMMON/BK10/R(300),Z(300),CODE(300),IX(6,250)
      COMMON/TIP/NCR1,NCR2,NCR3,NCR4,NELX,CTPX,CTPY,SIF1,SIF2
      COMMON/INTGR/PT(10),WG(10),PT2(2),WG2(2),PEI
      COMMON/VEL/CV,NUMCV,CVH(2,20)
      COMMON/DIM/NA,NAA,NBT,NB,NQ,NR,NINT,NINT2,IALL,NITER,SINCOD
      COMMON/OLDDISP/U(600),V(600),AA(600)
      COMMON/AB/P1S(36,20),P1SD(36,20),P1SDD(36,20)
     1 ,AT(2,20),ATD(2,20),ATDD(2,20)
      COMMON/MISL/IF(20)
      COMMON/DISP/UU(2,36),UUD(2,36),UUDD(2,36),UUR(2,2)
C
      COMMON/SSS/S1S(36,20),S1SD(36,20),S1SDD(36,20)
      COMMON/MASS/AMRR(36,36),AMRDR(36,36),AM22(2,2),AMR2(36,2)
     1,AMRD2(36,2),AMRDD2(36,2)
      COMMON/WK/WKAREA(1404)
      COMMON/LON/S2(36,36),S4(36,36),S2D(36,36),S2DD(36,36)
      DIMENSION E(800),ELOAD(800),EM(800),EMAT(800)
      DIMENSION US(600),VS(600),AS(600)
      DIMENSION EX1(2160),EX3(1404),EX5(5184)
      EQUIVALENCE(E(1),EX1(1))
      EQUIVALENCE(ELOAD(1),EX1(1001))
      EQUIVALENCE(EMAT(1),EX3(1))
      EQUIVALENCE(US(1),EX5(1))
      EQUIVALENCE(VS(1),EX5(1001))
      EQUIVALENCE(AS(1),EX5(2001))
      EQUIVALENCE(EM(1),EX5(3001))
      EQUIVALENCE(S1S(1,1),EX1(1))
      EQUIVALENCE(WKAREA(1),EX3(1))
      EQUIVALENCE(S2(1,1),EX5(1))
      COMMON/STRESS/RX(3,36),RXD(3,36),RXDD(3,36)
      DIMENSION B(192)
      DIMENSION B1(324)
      EQUIVALENCE(B(1),B1(1))
      EQUIVALENCE(B1(1),RX(1,1))
      DIMENSION BETA1DD(36)
      DIMENSION BETA2(2),BETA2DD(2)
      EQUIVALENCE(BETA1DD(1),B1(285))
      EQUIVALENCE(BETA2(1),B1(321))
      EQUIVALENCE(BETA2DD(1),B1(323))
      COMMON/BET/BETA1(36),BETA1D(36),BETA2D(2)
C
      COMPLEX LAM,SI,SIX,SIXX,DIS,DISX,DISXX
      COMPLEX FACTX1,FACTX2
C
      NBG=NBAND
      NEQDEL=NEQ+NDEL
      DO 10 J=1,NBG
      DO 10 I=1,NBAND
   10 A(I,J)=0.
```

```
      DO 20 I=1,NBAND
   20 B(I)=0.
      IF(L.EQ.0)REWIND 21
      IF(L.EQ.1)REWIND 24
      REWIND 12
      NBLOC=0
   30 CONTINUE
      NBLOC=NBLOC+1
      NBA=NBRED
      IF(NBLOC.EQ.NSBL) NBA=NBAND
      IF(NBLOC-LASTB-1) 40,140,40
   40 K=NBAND+1
      IF(L.EQ.0) READ(21) ((A(I,J),I=K,NBD2),J=1,NBG)
      IF(L.EQ.1) READ(24) ((A(I,J),I=K,NBD2),J=1,NBG)
      II=(NBLOC-2)*NBAND
      DO 50 I=K,NBD2
      B(I)=B(I+II)
   50 CONTINUE
C
C         INSERT BC
C
      NB1=NBAND*(NBLOC-1)+1
      NB2=NBAND*NBLOC
      DO 100 I=1,NBB
      IF(NFIX(I).EQ.0) GO TO 100
      KK1=NFIX(I)/10
      KK2=NFIX(I)-10
      K=2*NBC(I)-KK1+NDEL
      IF(K.LT.NB1.OR.K.GT.NB2) GO TO 100
   60 NES=K-(NBLOC-2)*NBAND
      A(NES,1)=1.
      B(NES)=0.
      DO 70 L1=2,NBG
   70 A(NES,L1)=0.
      DO 80 L1=2,NBAND
      LL=NES-L1+1
   80 A(LL,L1)=0.
      IF(KK2) 100,100,90
   90 K=K+1
      KK2=0
      GO TO 60
  100 CONTINUE
      IF(NDEL.EQ.0) GO TO 130
      IF(NBLOC.NE.1) GO TO 140
      DO 120 I=1,NDEL
  120 A(I+NBAND,1)=1.
      GO TO 200
  130 IF(NBLOC.EQ.1) GO TO 200
  140 CONTINUE
      NBA=NBRED
      IF(NBLOC.EQ.NSBL+1) NBA=NBAND
      DO 180 I=1,NBAND
      PIVOT=A(I,1)
      IF(PIVOT.EQ.0.) GO TO 190
      DO 160 J=2,NBA
      C=A(I,J)/PIVOT
      K=J+I-1
      IF(II+K .GT. NEQDEL) GO TO 170
      DO 150 III=J,NBA
  150 A(K,III-J+1)=A(K,III-J+1)-C*A(I,III)
      A(I,J)=C
  160 B(K)=B(K)-C*B(I)
  170 B(I)=B(I)/PIVOT
  180 CONTINUE
  190 CONTINUE
```

```
      IF(NBLOC.EQ.LASTB+1) GO TO 230
      WRITE(12) ((A(I,J),I=1,NBAND),J=1,NBG)
      WRITE(12) (B(I),I=1,NBAND)
  200 CONTINUE
      DO 220 I=1,NBAND
      K=I+NBAND
      DO 210 J=1,NBG
      A(I,J)=A(K,J)
  210 CONTINUE
      B(I)=B(K)
  220 B(K)=0.
      GO TO 30
  230 CONTINUE
      K=NBAND-(NF-NEQDEL)
  280 CONTINUE
      NBA=NBRED
      IF(NBLOC.EQ.NSBL+1) NBA=NBAND
      DO 300 III=1,K
      JJJ=K-III+1
      DO 300 J=2,NBA
  300 B(JJJ)=B(JJJ)-A(JJJ,J)*B(JJJ+J-1)
      JJ=NBAND*(NBLOC-2)
      III=1
      IF(JJ.EQ.0) III=NDEL+1
      DO 310 I=III,K
      J=JJ+I-NDEL
  310 US(J)=B(I)
      NBLOC=NBLOC-1
      IF(NBLOC.EQ.1) GO TO 330
      DO 320 I=1,NBAND
  320 B(I+NBAND)=B(I)
      BACKSPACE 12
      BACKSPACE 12
      READ(12) ((A(I,J),I=1,NBAND),J=1,NBG)
      READ(12) (B(I),I=1,NBAND)
      BACKSPACE 12
      BACKSPACE 12
      K=NBAND
      GO TO 280
  330 CONTINUE
C    EQUATE DISPLACEMENTS FOR DOUBLE NODES
      DO 340 I=1,NPAIR
      IF(LPAIR(3,I) .LE. 0) GO TO 340
      US(LPAIR(1,I))=US(LPAIR(2,I))
 340  CONTINUE

      IF(L.EQ.1) GO TO 410

      IF(LP.EQ.0) GO TO 4020
      DO 4010 N=1,NEQ
      AN=AA(N)
      AS(N)=(US(N)-U(N))/BET1-V(N)/BET2-AN*BET3
      VS(N)=V(N)+(AN+AS(N))*DELT/2.
 4010 CONTINUE
      GO TO 4040
 4020 CONTINUE
      DO 4030 I=1,NEQ
      AS(I)=0.
      VS(I)=0.
 4030 CONTINUE
 4040 CONTINUE
C    SOLV FOR CENTER POINT DISPLACEMENT,VELOCITY,AND ACCELERATION
C    OF THE SINGULAR ELEMENT
C CALC DISPLACEMENT COEFFICIENTS FOR THE SINGULAR ELEMENT BETA=B*Q ,ETC.
      DO 350 K=1,NB
```

```
      BETA1(K)=0.
      DO 350 J=1,NQ
      BETA1(K)=BETA1(K)+P1S(K,J)*US(IF(J))
  350 CONTINUE
      DO 355 K=1,NR
      BETA2(K)=0.
      DO 355 J=1,NQ
      BETA2(K)=BETA2(K)+AT(K,J)*US(IF(J))
  355 CONTINUE
      IF(CV.EQ.0.) GO TO 4100
      DO 4060 I=1,NB
      BETA1D(I)=0.
      DO 4060 J=1,NQ
      BETA1D(I)=BETA1D(I)+P1SD(I,J)*US(IF(J))+P1S(I,J)*VS(IF(J))
 4060 CONTINUE
      DO 4070 I=1,NR
      BETA2D(I)=0.
      DO 4070 J=1,NQ
      BETA2D(I)=BETA2D(I)+ATD(I,J)*US(IF(J))+AT(I,J)*VS(IF(J))
 4070 CONTINUE
      DO 4080 I=1,NB
      BETA1DD(I)=0.
      DO 4080 J=1,NQ
      BETA1DD(I)=BETA1DD(I)+P1SDD(I,J)*US(IF(J))+2.*P1SD(I,J)*VS(IF(J))
     1+P1S(I,J)*AS(IF(J))
 4080 CONTINUE
      DO 4090 I=1,NR
      BETA2DD(I)=0.
      DO 4090 J=1,NQ
      BETA2DD(I)=BETA2DD(I)+ATDD(I,J)*US(IF(J))+2.*ATD(I,J)*VS(IF(J))
     1+AT(I,J)*AS(IF(J))
 4090 CONTINUE
      GO TO 4200
 4100 CONTINUE
      DO 4160 I=1,NB
      BETA1D(I)=0.
      DO 4160 J=1,NQ
      BETA1D(I)=BETA1D(I)+P1S(I,J)*VS(IF(J))
 4160 CONTINUE
      DO 4170 I=1,NR
      BETA2D(I)=0.
      DO 4170 J=1,NQ
      BETA2D(I)=BETA2D(I)+AT(I,J)*VS(IF(J))
 4170 CONTINUE
      DO 4180 I=1,NB
      BETA1DD(I)=0.
      DO 4180 J=1,NQ
      BETA1DD(I)=BETA1DD(I)+P1S(I,J)*AS(IF(J))
 4180 CONTINUE
      DO 4190 I=1,NR
      BETA2DD(I)=0.
      DO 4190 J=1,NQ
      BETA2DD(I)=BETA2DD(I)+AT(I,J)*AS(IF(J))
 4190 CONTINUE
 4200 CONTINUE
C  CALC DISPLACEMENTS OF THE INTERNAL NODES OF THE SINGULAR ELEMENT
      NN=IX(3,NCR2)*2
      X1=2.*(R(NN/2)-CORD(1,1))/(CORD(5,1)-CORD(1,1))-1.
      IF(CTPX.GT.R(NN/2)) GO TO 370
      CX=(2.*CTPX-CORD(1,1)-CORD(5,1))/(CORD(5,1)-CORD(1,1))
      IF(X1-CX .LT. 1.E-10) X1=1.E-10+CX
      Y1=0.
      CALL TRANS(X1,Y1,N,2,1)
```

```
      CALL FUNCTS(X1,Y1,6)
      IF(CV.EQ.0.) GO TO 362
      DO 360 K=1,2
      US(NN-2+K)=0.
      VS(NN-2+K)=0.
      AS(NN-2+K)=0.
      DO 360 J=1,NB
      US(NN-2+K)=US(NN-2+K)+UU(K,J)*BETA1(J)
      VS(NN-2+K)=VS(NN-2+K)+UUD(K,J)*BETA1(J)+UU(K,J)*BETA1D(J)
      AS(NN-2+K)=AS(NN-2+K)+UUDD(K,J)*BETA1(J)+2.*UUD(K,J)*BETA1D(J)
     1+UU(K,J)*BETA1DD(J)
  360 CONTINUE
      GO TO 364
  362 DO 363 K=1,2
      US(NN-2+K)=0.
      VS(NN-2+K)=0.
      AS(NN-2+K)=0.
      DO 363 J=1,NB
      US(NN-2+K)=US(NN-2+K)+UU(K,J)*BETA1(J)
      VS(NN-2+K)=VS(NN-2+K)+UU(K,J)*BETA1D(J)
      AS(NN-2+K)=AS(NN-2+K)+UU(K,J)*BETA1DD(J)
  363 CONTINUE
  364 DO 365 K=1,2
      DO 365 J=1,NR
      US(NN-2+K)=US(NN-2+K)+UUR(K,J)*BETA2(J)
      VS(NN-2+K)=VS(NN-2+K)+UUR(K,J)*BETA2D(J)
      AS(NN-2+K)=AS(NN-2+K)+UUR(K,J)*BETA2DD(J)
  365 CONTINUE
      US(IX(2,NCR1)*2-1)=US(NN-1)
      US(IX(2,NCR1)*2)=US(NN)
      VS(IX(2,NCR1)*2-1)=VS(NN-1)
      VS(IX(2,NCR1)*2)=VS(NN)
      AS(IX(2,NCR1)*2-1)=AS(NN-1)
      AS(IX(2,NCR1)*2)=AS(NN)
      GO TO 400
  370 CONTINUE
      DO 390 I=1,2
      IF(I.EQ.2) GO TO 375
      NN=IX(3,NCR2)*2
      Y1=-1.E-10
      CALL TRANS(X1,Y1,N,2,1)
      CALL FUNCTS(X1,Y1,6)
      GO TO 377
  375 NN=IX(2,NCR1)*2
      Y1=1.E-10
      CALL TRANS(X1,Y1,N,2,1)
      CALL FUNCTS(X1,Y1,6)
  377 IF(CV.EQ.0.) GO TO 382
      DO 380 K=1,2
      US(NN-2+K)=0.
      VS(NN-2+K)=0.
      AS(NN-2+K)=0.
      DO 380 J=1,NB
      US(NN-2+K)=US(NN-2+K)+UU(K,J)*BETA1(J)
      VS(NN-2+K)=VS(NN-2+K)+UUD(K,J)*BETA1(J)+UU(K,J)*BETA1D(J)
      AS(NN-2+K)=AS(NN-2+K)+UUDD(K,J)*BETA1(J)+2.*UUD(K,J)*BETA1D(J)
     1+UU(K,J)*BETA1DD(J)
  380 CONTINUE
      GO TO 384
  382 DO 383 K=1,2
      US(NN-2+K)=0.
      VS(NN-2+K)=0.
      AS(NN-2+K)=0.
      DO 383 J=1,NB
      US(NN-2+K)=US(NN-2+K)+UU(K,J)*BETA1(J)
```

```
      VS(NN-2+K)=VS(NN-2+K)+UU(K,J)*BETA1D(J)
      AS(NN-2+K)=AS(NN-2+K)+UU(K,J)*BETA1DD(J)
  383 CONTINUE
  384 DO 385 K=1,2
      DO 385 J=1,NR
      US(NN-2+K)=US(NN-2+K)+UUR(K,J)*BETA2(J)
      VS(NN-2+K)=VS(NN-2+K)+UUR(K,J)*BETA2D(J)
      AS(NN-2+K)=AS(NN-2+K)+UUR(K,J)*BETA2DD(J)
  385 CONTINUE
  390 CONTINUE
  400 CONTINUE
C  CALC STRESS INTENSITY FACTORS (SQUARE ELEMENT ONLY)
      CX=(2.*CTPX-CORD(1,1)-CORD(5,1))/(CORD(5,1)-CORD(1,1))
      X1=CX
      Y1=0.
      CALL TRANS(X1,Y1,N,2,1)
      CALL COEFF(1,1)
      FACTX2=LAM(1)*ALOG(PJ1)
      FACTX2=CEXP(FACTX2)
      FACTX1=(SI(5)+SI(6)+SI(7)+SI(8))*FACTX2
      FACTX2=(SI(9)+SI(10)+SI(11)+SI(12))*FACTX2
      SIF1=(REAL(FACTX1)*BETA1(1)-AIMAG(FACTX1)*BETA1(2))*SQRT(2.*PEI)
      SIF2=(REAL(FACTX2)*BETA1(1)-AIMAG(FACTX2)*BETA1(2))*SQRT(2.*PEI)
C
      RETURN
C
C     SOLV FOR CENTER POINT ACCELERATION OF SINGULAR ELEMENT FOR JUMP
C CALC DISPLACEMENT COEFFICIENTS FOR THE SINGULAR ELEMENT BETA=B*Q ,ETC.
  410 CONTINUE
      DO 420 I=1,NB
      BETA1(I)=0.
      DO 420 J=1,NQ
      BETA1(I)=BETA1(I)+P1S(I,J)*U(IF(J))
  420 CONTINUE
      DO 430 I=1,NR
      BETA2(I)=0.
      DO 430 J=1,NQ
      BETA2(I)=BETA2(I)+AT(I,J)*U(IF(J))
  430 CONTINUE
      IF(CV.EQ.0.) GO TO 4800
      DO 440 I=1,NB
      BETA1D(I)=0.
      DO 440 J=1,NQ
      BETA1D(I)=BETA1D(I)+P1SD(I,J)*U(IF(J))+P1S(I,J)*V(IF(J))
  440 CONTINUE
      DO 450 I=1,NR
      BETA2D(I)=0.
      DO 450 J=1,NQ
      BETA2D(I)=BETA2D(I)+ATD(I,J)*U(IF(J))+AT(I,J)*V(IF(J))
  450 CONTINUE
      DO 460 I=1,NB
      BETA1DD(I)=0.
      DO 460 J=1,NQ
      BETA1DD(I)=BETA1DD(I)+P1SDD(I,J)*U(IF(J))+2.*P1SD(I,J)*V(IF(J))
     1+P1S(I,J)*US(IF(J))
  460 CONTINUE
      DO 470 I=1,NR
      BETA2DD(I)=0.
      DO 470 J=1,NQ
      BETA2DD(I)=BETA2DD(I)+ATDD(I,J)*U(IF(J))+2.*ATD(I,J)*V(IF(J))
     1+AT(I,J)*US(IF(J))
  470 CONTINUE
      GO TO 4900
 4800 CONTINUE
      DO 4840 I=1,NB
```

```
      BETA1D(I)=Ø.
      DO 484Ø J=1,NQ
      BETA1D(I)=BETA1D(I)+P1S(I,J)*V(IF(J))
 484Ø CONTINUE
      DO 485Ø I=1,NR
      BETA2D(I)=Ø.
      DO 485Ø J=1,NQ
      BETA2D(I)=BETA2D(I)+AT(I,J)*V(IF(J))
 485Ø CONTINUE
      DO 486Ø I=1,NB
      BETA1DD(I)=Ø.
      DO 486Ø J=1,NQ
      BETA1DD(I)=BETA1DD(I)+P1S(I,J)*US(IF(J))
 486Ø CONTINUE
      DO 487Ø I=1,NR
      BETA2DD(I)=Ø.
      DO 487Ø J=1,NQ
      BETA2DD(I)=BETA2DD(I)+AT(I,J)*US(IF(J))
 487Ø CONTINUE
 49ØØ CONTINUE

C  CALC ACCELERATION OF THE INTERNAL NODES OF THE SINGULAR ELEMENT
      I=IX(3,NCR2)*2-2
      X1=2.*(R(I/2+1)-CORD(1,1))/(CORD(5,1)-CORD(1,1))-1.
      IF(CTPX.GT.R(I/2+1)) GO TO 471
      CX=(2.*CTPX-CORD(1,1)-CORD(5,1))/(CORD(5,1)-CORD(1,1))
      IF(X1-CX .LT. 1.E-1Ø) X1=1.E-1Ø+CX
      Y1=Ø.
      CALL TRANS(X1,Y1,N,2,1)
      CALL FUNCTS(X1,Y1,6)
      IF(CV.EQ. Ø.) GO TO 6662
      DO 666Ø K=1,2
      JJ=I+K
      US(JJ)=Ø.
      DO 666Ø J=1,NB
      US(JJ)=US(JJ)+UUDD(K,J)*BETA1(J)+2.*UUD(K,J)*BETA1D(J)
     1+UU(K,J)*BETA1DD(J)
 666Ø CONTINUE
      GO TO 6664
 6662 DO 6663 K=1,2
      JJ=I+K
      US(JJ)=Ø.
      DO 6663 J=1,NB
      US(JJ)=US(JJ)+UU(K,J)*BETA1DD(J)
 6663 CONTINUE
 6664 DO 6665 K=1,2
      JJ=I+K
      DO 6665 J=1,NR
      US(JJ)=US(JJ)+UUR(K,J)*BETA2DD(J)
 6665 CONTINUE
      US(IX(2,NCR1)*2-1)=US(I+1)
      US(IX(2,NCR1)*2)=US(I+2)
      GO TO 51Ø
  471 DO 5ØØ M=1,2
      IF(M.EQ.2) GO TO 475
      I=2*IX(3,NCR2)-2
      Y1=-1.E-1Ø
      CALL TRANS(X1,Y1,N,2,1)
      CALL FUNCTS(X1,Y1,6)
      GO TO 478
  475 I=2*IX(2,NCR1)-2
      Y1=1.E-1Ø
      CALL TRANS(X1,Y1,N,2,1)
      CALL FUNCTS(X1,Y1,6)
  478 IF(CV.EQ.Ø.) GO TO 482
```

```
      DO 480 J=1,2
      JJ=I+J
      US(JJ)=0.
      DO 480 K=1,NB
      US(JJ)=US(JJ)+UUDD(J,K)*BETA1(K)+2.*UUD(J,K)*BETA1D(K)
     1+UU(J,K)*BETA1DD(K)
  480 CONTINUE
      GO TO 486
  482 DO 483 J=1,2
      JJ=I+J
      US(JJ)=0.
      DO 483 K=1,NB
      US(JJ)=US(JJ)+UU(J,K)*BETA1DD(K)
  483 CONTINUE
  486 DO 490 J=1,2
      JJ=I+J
      DO 490 K=1,NR
      US(JJ)=US(JJ)+UUR(J,K)*BETA2DD(K)
  490 CONTINUE
  500 CONTINUE
  510 CONTINUE
C  CALC STRESS INTENSITY FACTORS (SQUARE ELEMENT ONLY)
      CX=(2.*CTPX-CORD(1,1)-CORD(5,1))/(CORD(5,1)-CORD(1,1))
      X1=CX
      Y1=0.
      CALL TRANS(X1,Y1,N,2,1)
      CALL COEFF(1,1)
      FACTX2=LAM(1)*ALOG(PJ1)
      FACTX2=CEXP(FACTX2)
      FACTX1=(SI(5)+SI(6)+SI(7)+SI(8))*FACTX2
      FACTX2=(SI(9)+SI(10)+SI(11)+SI(12))*FACTX2
      SIF1=(REAL(FACTX1)*BETA1(1)-AIMAG(FACTX1)*BETA1(2))*SQRT(2.*PEI)
      SIF2=(REAL(FACTX2)*BETA1(1)-AIMAG(FACTX2)*BETA1(2))*SQRT(2.*PEI)
      RETURN
C
      END
```

```
      SUBROUTINE MASSEL
C
      COMMON/ELM/ESTIFM(8,8,50),ELMASS(8,8,50),VSS(20,20)
      COMMON/INTGR/PT(10),WG(10),PT2(2),WG2(2),PEI
      COMMON/DIF/DXX2,DXY8,DXXY,DYX2,DYY2,DYXY,DXX,DXY,DYX,DYY,DEM
      COMMON/SHIFT/NSBL,NDEL,LASTB,NF,NBRED,NELTYP,ISK,RCODE
      COMMON/BK10/R(300),Z(300),CODE(300),IX(6,250)
      COMMON/RHHO/RHO(6),RODUM
      COMMON/DIM/NA,NAA,NBT,NB,NQ,NR,NINT,NINT2,IALL,NITER,SINCOD
      COMMON/WK2/WKAR(720)
      COMMON/BK1/NUMMAT,NUMNP,NUMEL,NUMPC,NUMLP, IHED(8),NTAPE,NEQ,ICOND
     1,ISTAT,TMACH, IHED1(8),ISYMT
      DIMENSION U(2,8)
      EQUIVALENCE(U(1,1),WKAR(1))
C
      DO 10 K=1,NELTYP
      DO 10 J=1,8
      DO 10 I=1,8
   10 ELMASS(I,J,K)=0.
      DO 80 K=1,NELTYP
      DO 20 I=1,NUMEL
      IF(IX(5,I).EQ.K) GO TO 25
   20 CONTINUE
      GO TO 80
   25 N=I
      KK=IX(6,N)
      DO 50 L=1,NINT2
      X=PT2(L)
      DO 40 M=1,NINT2
      Y=PT2(M)
      WT=WG2(L)*WG2(M)
C
      CALL TRANS(X,Y,N,2,2)
C
      CALL FUNCTR(X,Y)
C
      WTDEMR=WT*DEM*RHO(KK)
      DO 30 J=1,8
      DO 30 I=1,8
      DO 30 KF=1,2
   30 ELMASS(I,J,K)=ELMASS(I,J,K)+U(KF,I)*U(KF,J)*WTDEMR
   40 CONTINUE
   50 CONTINUE
   80 CONTINUE
      RETURN
C
      END
```

```
      SUBROUTINE FUNCTR(X,Y)
C
      COMMON/WK2/WKAR(720)
      DIMENSION U(2,8)
      EQUIVALENCE(U(1,1),WKAR(1))
C
      DO 10 J=1,8
      DO 10 I=1,2
   10 U(I,J)=0.
      U(1,1)=.25*(1.-X)*(1.-Y)
      U(1,3)=.25*(1.+X)*(1.-Y)
      U(1,5)=.25*(1.+X)*(1.+Y)
      U(1,7)=.25*(1.-X)*(1.+Y)
      U(2,2)=U(1,1)
      U(2,4)=U(1,3)
      U(2,6)=U(1,5)
      U(2,8)=U(1,7)
      RETURN
C
      END
```

```
      SUBROUTINE STIFEL
C
      COMMON/ELM/ESTIFM(8,8,50),ELMASS(8,8,50),VSS(20,20)
      COMMON/DIM/NA,NAA,NBT,NB,NQ,NR,NINT,NINT2,IALL,NITER,SINCOD
      COMMON/SHIFT/NSBL,NDEL,LASTB,NF,NBRED,NELTYP,ISK,RCODE
      COMMON/BK10/R(300),Z(300),CODE(300),IX(6,250)
      COMMON/SUMAN/E(3,3,6),EI(3,3,6),ADET(6),ASIZE(6),CL
      COMMON/INTGR/PT(10),WG(10),PT2(2),WG2(2),PEI
      COMMON/DIF/DXX2,DXY3,DXXY,DYX2,DYY2,DYXY,DXX,DXY,DYX,DYY,DEM
      COMMON/BK1/NUMMAT,NUMNP,NUMEL,NUMPC,NUMLP, IHED(8),NTAPE,NEQ,ICOND
     1,ISTAT,TMACH, IHED1(8),ISYMT
C
      DIMENSION A(2,2),C(3,8)
      DIMENSION B(2,4),AJJ(3,8)
      DIMENSION B1(60)
      COMMON/WK2/WKAR(720)
      EQUIVALENCE(B(1,1),B1(1))
      EQUIVALENCE(AJJ(1,1),B1(9))
      EQUIVALENCE(C(1,1),B1(33))
      EQUIVALENCE(A(1,1),B1(57))
      EQUIVALENCE(B1(1),WKAR(1))
C
      NIN=NINT
      IF(IALL.GT.0) NIN=NINT2
      DO 10 K=1,NELTYP
      DO 10 J=1,8
      DO 10 I=1,8
   10 ESTIFM(I,J,K)=0.
      DO 80 K=1,NELTYP
      DO 20 I=1,NUMEL
      IF(IX(5,I).EQ.K) GO TO 25
   20 CONTINUE
      GO TO 80
   25 N=I
      KK=IX(6,N)
      DO 70 L=1,NIN
      X=PT(L)
      IF(IALL.GT.0) X=PT2(L)
      DO 60 M=1,NIN
      Y=PT(M)
      IF(IALL.GT.0) Y=PT2(M)
      WT=WG(L)*WG(M)
      IF(IALL.GT.0) WT=WG2(L)*WG2(M)
C
      CALL TRANS(X,Y,N,2,2)
C
      CALL DIFFB(X,Y)
C
      DO 40 J=1,8
      DO 40 I=1,3
      AJJ(I,J)=0.
      DO 40 KF=1,3
   40 AJJ(I,J)=AJJ(I,J)+EI(I,KF,KK)*C(KF,J)
      WTDEM=WT*DEM
      DO 50 J=1,8
      DO 50 I=1,8
      DO 50 KF=1,3
   50 ESTIFM(I,J,K)=ESTIFM(I,J,K)+C(KF,I)*AJJ(KF,J)*WTDEM
   60 CONTINUE
   70 CONTINUE
   80 CONTINUE
      RETURN
C
      END
```

```
      SUBROUTINE DIFFB(X,Y)
C
      COMMON/IPAR/PJ1,PJ2,PJ3,PJ4,DXY1,DXY2
      DIMENSION A(2,2),C(3,8)
      DIMENSION B(2,4),AJJ(3,8)
      DIMENSION B1(60)
C
      COMMON/WK2/WKAR(720)
      EQUIVALENCE(B(1,1),B1(1))
      EQUIVALENCE(AJJ(1,1),B1(9))
      EQUIVALENCE(C(1,1),B1(33))
      EQUIVALENCE(A(1,1),B1(57))
      EQUIVALENCE(B1(1),WKAR(1))
C
      A(1,1)=PJ1
      A(1,2)=PJ2
      A(2,1)=PJ3
      A(2,2)=PJ4
      C(1,1)=-.25*(1.-Y)
      C(1,2)=+.25*(1.-Y)
      C(1,3)=+.25*(1.+Y)
      C(1,4)=-.25*(1.+Y)
      C(2,1)=-.25*(1.-X)
      C(2,2)=-.25*(1.+X)
      C(2,3)=+.25*(1.+X)
      C(2,4)=+.25*(1.-X)
      DO 15 J=1,4
      DO 15 I=1,2
      B(I,J)=0.
      DO 15 K=1,2
      B(I,J)=B(I,J)+A(I,K)*C(K,J)
   15 CONTINUE
      DO 20 J=1,8
      DO 20 I=1,3
   20 C(I,J)=0.
      C(1,1)=B(1,1)
      C(1,3)=B(1,2)
      C(1,5)=B(1,3)
      C(1,7)=B(1,4)
      C(2,2)=B(2,1)
      C(2,4)=B(2,2)
      C(2,6)=B(2,3)
      C(2,8)=B(2,4)
      C(3,1)=C(2,2)
      C(3,2)=C(1,1)
      C(3,3)=C(2,4)
      C(3,4)=C(1,3)
      C(3,5)=C(2,6)
      C(3,6)=C(1,5)
      C(3,7)=C(2,8)
      C(3,8)=C(1,7)
      RETURN
C
      END
```

```
      SUBROUTINE POSIT(TT,ILP,LS)
C
      COMMON/BK11/DELT,DT1,DT3,BETA,BET1,BET2,BET3,BET4,BET5,NBAND,NBD2
      COMMON/TIP/NCR1,NCR2,NCR3,NCR4,NELX,CTPX,CTPY,SIF1,SIF2
      COMMON/VEL/CV,NUMCV,CVH(2,20)
      COMMON/POS/NUMPS,POST(2,20)
      COMMON/BK1/NUMMAT,NUMNP,NUMEL,NUMPC,NUMLP, IHED(8),NTAPE,NEQ,ICOND
     1,ISTAT,TMACH, IHED1(8),ISYMT
      COMMON/BOUND/NFIX(200),NBC(200),NBB
      COMMON/DIM/NA,NAA,NBT,NB,NQ,NR,NINT,NINT2,IALL,NITER,SINCOD
      COMMON/SHIFT/NSBL,NDEL,LASTB,NF,NBRED,NELTYP,ISK,RCODE
      COMMON/BK10/R(300),Z(300),CODE(300),IX(6,250)
      COMMON/OLDDISP/U(600),V(600),A(600)
      COMMON/MISL/IF(20)
      COMMON/PAIR1/NPAIR,LPAIR(3,40)
      COMMON/MAIN/CORD(10,2)
      COMMON/RIG/IRIG1(6),IRIG2
      COMMON/TOLR/TOLER1,TOLER2
C
C        FIND NEW CRACK TIP POSITION
C
      ILP=0
      IF(TT.GE.POST(1,1) .AND. TT.LE.POST(1,NUMPS)) GO TO 5
      WRITE(6,340)
      STOP 21
    5 DO 10 I=2,NUMPS
      IF(TT .LE. POST(1,I)) GO TO 12
   10 CONTINUE
   12 DELT1=TT-POST(1,I-1)
      D1=POST(1,I)-POST(1,I-1)
      DELD=POST(2,I)-POST(2,I-1)
      IF(D1.NE.0.) GO TO 13
      IF(DELD.NE.0.) GO TO 300
      IF(I.EQ.NUMPS) RETURN
      I=I+1
      GO TO 12
   13 CTPX=POST(2,I-1)+DELD*DELT1/D1
      IF(LS.EQ.1) RETURN
      M1=IX(4,NCR2)
      MF=IX(3,NCR4)
      DELTA=(R(MF)-R(M1))*.75
      DIST=R(M1)+DELTA
      IF(CTPX.GT.DIST) GO TO 15
      GO TO 200
   15 CONTINUE
C
C        ADVANCE CRACK TIP ELEMENTS
C
      NCR1=NCR1+NELX
      NCR2=NCR2+NELX
      NCR3=NCR3+NELX
      NCR4=NCR4+NELX
      WRITE(6,330) NCR1,NCR2,NCR3,NCR4
C CHECK FOR THE SINGULAR ELEMENT TO BE INSIDE THE MESH
      IF(NCR1.LE.0 .OR. NCR1 .GT. NUMEL) GO TO 20
      IF(NCR2.LE.0 .OR. NCR2 .GT. NUMEL) GO TO 20
      IF(NCR3.LE.0 .OR. NCR3 .GT. NUMEL) GO TO 20
      IF(NCR4.LE.0 .OR. NCR4 .GT. NUMEL) GO TO 20
C
C CHECK TO SEE IF MATERIALS IN SINGULAR ELEMENT IS CORRECT
      IF(IX(6,NCR1).NE.IX(6,NCR3)) GO TO 579
      IF(IX(6,NCR2).NE.IX(6,NCR4)) GO TO 579
C CHECK FOR THE ELEMENTS OF THE SINGULAR ELEMENT TO BE SQUARES
      XL=R(IX(2,NCR1))-R(IX(1,NCR1))
      IF(IALL .EQ.1) GO TO 2466
```

```
      DO 2465 M=1,2
      IF(M.EQ.1) I=NCR3
      IF(M.EQ.2) I=NCR4
      M1=IX(1,I)
      M2=IX(2,I)
      M3=IX(3,I)
      M4=IX(4,I)
      RR1=R(M1)
      RR2=R(M2)
      RR3=R(M3)
      RR4=R(M4)
      ZZ1=Z(M1)
      ZZ2=Z(M2)
      ZZ3=Z(M3)
      ZZ4=Z(M4)
      IF(ABS(ABS((RR2-RR1)/XL)-1.) .GT. TOLER1) GO TO 21
      IF(ABS((RR3-RR2)/XL) .GT. TOLER1) GO TO 21
      IF(ABS(ABS((RR3-RR4)/XL)-1.) .GT. TOLER1) GO TO 21
      IF(ABS((RR4-RR1)/XL) .GT. TOLER1) GO TO 21
      IF(ABS((ZZ2-ZZ1)/XL) .GT. TOLER1) GO TO 21
      IF(ABS(ABS((ZZ3-ZZ2)/XL)-1.) .GT. TOLER1) GO TO 21
      IF(ABS((ZZ3-ZZ4)/XL) .GT. TOLER1) GO TO 21
      IF(ABS(ABS((ZZ4-ZZ1)/XL)-1.) .GT. TOLER1) GO TO 21
 2465 CONTINUE
 2466 CONTINUE
C CHECK FOR THE SINGULAR ELEMENT TO BE A SQUARE(I.E. CHECK TO SEE IF THE
C ELEMENTS OF THE SINGULAR ELEMENT ARE NUMBERD CORRECTLY)
      M1=IX(1,NCR2)
      M2=IX(2,NCR4)
      M3=IX(3,NCR3)
      M4=IX(4,NCR1)
      RR1=R(M1)
      RR2=R(M2)
      RR3=R(M3)
      RR4=R(M4)
      ZZ1=Z(M1)
      ZZ2=Z(M2)
      ZZ3=Z(M3)
      ZZ4=Z(M4)
      XL2=2.*XL
      IF(ABS(ABS((RR2-RR1)/XL2)-1.) .GT. TOLER1) GO TO 21
      IF(ABS((RR3-RR2)/XL2) .GT. TOLER1) GO TO 21
      IF(ABS(ABS((RR3-RR4)/XL2)-1.) .GT. TOLER1) GO TO 21
      IF(ABS((RR4-RR1)/XL2) .GT. TOLER1) GO TO 21
      IF(ABS((ZZ2-ZZ1)/XL2) .GT. TOLER1) GO TO 21
      IF(ABS(ABS((ZZ3-ZZ2)/XL2)-1.) .GT. TOLER1) GO TO 21
      IF(ABS((ZZ3-ZZ4)/XL2) .GT. TOLER1) GO TO 21
      IF(ABS(ABS((ZZ4-ZZ1)/XL2)-1.) .GT. TOLER1) GO TO 21
      GO TO 24
   20 WRITE(6,970)
      GO TO 23
   21 WRITE(6,975)
      GO TO 23
  579 WRITE(6,935)
   23 ILP=1
      RETURN
   24 CONTINUE
      IF(2)=IX(4,NCR2)*2
      IF(4)=IX(1,NCR2)*2
      IF(6)=IX(2,NCR2)*2
      IF(8)=IX(2,NCR4)*2
      IF(10)=IX(3,NCR4)*2
      IF(12)=IX(2,NCR3)*2
      IF(14)=IX(3,NCR3)*2
      IF(16)=IX(4,NCR3)*2
```

```
      IF(18)=IX(4,NCR1)*2
      IF(20)=IX(1,NCR1)*2
      IF(1)=IF(2)-1
      IF(3)=IF(4)-1
      IF(5)=IF(6)-1
      IF(7)=IF(8)-1
      IF(9)=IF(10)-1
      IF(11)=IF(12)-1
      IF(13)=IF(14)-1
      IF(15)=IF(16)-1
      IF(17)=IF(18)-1
      IF(19)=IF(20)-1
      MAX=IF(2)
      MIN=IF(2)
      DO 25 I=4,NQ,2
      IF(IF(I).GT.MAX) MAX=IF(I)
      IF(IF(I).LT.MIN) MIN=IF(I)
 25   CONTINUE
      NSBL=(MAX-1)/NBAND+1
      NDEL=(NSBL-1)*NBAND+2-MIN
      LASTB=(NEQ+NDEL-1)/NBAND+1
      NF=LASTB*NBAND
      DO 30 I=2,NQ,2
      K=IF(I)/2
      II=I/2
      CORD(II,1)=R(K)
      CORD(II,2)=Z(K)
 30   CONTINUE
      MC1=IX(2,NCR1)
      MC2=IX(3,NCR2)
      NC1=IX(1,NCR1)
      NC2=IX(4,NCR2)
      CODE(MC1)=3.
      CODE(MC2)=3.
      CODE(NC1)=0.
      IF(IRIG1(IX(6,NCR1)).EQ.1) CODE(NC1)=3.
      CODE(NC2)=0.
      IF(IRIG1(IX(6,NCR2)).EQ.1) CODE(NC2)=3.
      DO 100 I=1,NPAIR
      N=(LPAIR(1,I)+1)/2
      IF(N.EQ.MC2) LPAIR(3,I)=0
 100  CONTINUE
      I=0
      DO 120 N=1,NUMNP
      IPHI=IFIX(CODE(N))
      IF(IPHI.EQ.0) GO TO 120
      I=I+1
      NBC(I)=N
      IF(IPHI.EQ.1) NFIX(I)=10
      IF(IPHI.EQ.2) NFIX(I)=01
      IF(IPHI.EQ.3) NFIX(I)=11
 120  CONTINUE
      NBB=I
 200  RETURN
 300  WRITE(6,320)
      STOP 4
 320  FORMAT(1H1,///,50X,10HWRONG DATA,//,40X,
     1  34HPOSITION CARDS CAN NOT HAVE JUMPS.)
 330  FORMAT(1H1,///,20X,52H*****SINGULAR ELEMENT RELOCATION AND RE-RESH
     1ING*****,//,25X,17HELEMENTS NUMBERED,I5,I5,I5,6H    AND,I5,31H    AR
     2E THE NEW SPECIAL ELEMENTS,/////)
 340  FORMAT(1H1,///,20X,71H*****CRACK-TIP POSITION IS NOT DEFINED AT TH
     1IS TIME ON INPUT CARDS*****)
 935  FORMAT (1H1,//,30X,39H*****THE SINGULAR ELEMENT IS WRONG*****,//
     1,10X,60HTHE MATERIAL TYPES IN SINGULAR ELEMENT ARE NO LONGER CORRE
```

```
     2CT)
 970  FORMAT (1H1,//,20X,91H*****THE CRACK-TIP LOCATION HAS PAST THE CEN
     1TER POINT OF THE LAST ELEMENT ON THE CRACK-LINE,/,30X,
     247HIN THE MESH AND RE-MESHING IS NOT POSSIBLE*****)
 975  FORMAT (1H1,//,30X,39H*****THE SINGULAR ELEMENT IS WRONG*****,//
     1,30X,54H*****THE SINGULAR ELEMENT IS NOT A SQUARE ANYMORE*****)
C
      END
```

```
      SUBROUTINE VELOC(TT,TTV,L,LV)
C
C          THIS SUBROUTINE COMPUTES THE CRACK TIP VELOCITY
C          FROM A GIVEN VELOCITY HISTORY
C
      COMMON/VEL/CV,NUMCV,CVH(2,20)
      COMMON/BK11/DELT,DT1,DT3,BETA,BET1,BET2,BET3,BET4,BET5,NBAND,NBD2
C  L=0 FIND APPROPRIATE TIME ONLY AND DETERMINE IF THERE IS A JUMP
C  L=1 GIVEN TIME AND JUMP CONDITION FIND VELOCITY MAGNITUDE
C  LV=0 NO JUMP
C  LV=1    JUMP

      IF(L.EQ.0) GO TO 2
      IF(L.EQ.1) GO TO 66
  2   IF(NUMCV.NE.0) GO TO 4
      TTV=TT+DT1
      LV=0
      GO TO 100
  4   IF(TT.LT.CVH(1,1    )) GO TO 6
      IF(TT.GT.CVH(1,NUMCV)) GO TO 8
      GO TO 10
  6   TTV=TT+DT1
      IF(TTV.GT.CVH(1,1)) TTV=CVH(1,1)
      LV=0
      GO TO 100
  8   TTV=TT+DT1
      LV=0
      GO TO 100
 10   DO 50 I=1,NUMCV
      J=I
      IF(I.EQ.NUMCV) GO TO 20
      IF(TT.GT.CVH(1,I)) GO TO 50
      IF(TT.EQ.CVH(1,I) .AND. TT.EQ.CVH(1,I+1)) GO TO 40
      IF(TT.EQ.CVH(1,I)) GO TO 30
 20   J=J-1
 30   TTV=TT+DT1
      IF(TTV.GT.CVH(1,J+1)) TTV=CVH(1,J+1)
      LV=0
      GO TO 100
 40   IF(CV.EQ.CVH(2,I))GO TO 42
      TTV=TT+DT1
      IF(TTV.GT.CVH(1,I+2)) TTV=CVH(1,I+2)
      LV=0
      GO TO 100
42    TTV=TT
      LV=1
      GO TO 100
 50   CONTINUE

 66   IF(NUMCV.NE.0) GO TO 67
      CV=0.
      GO TO 100
 67   IF(TT.GE.CVH(1,1) .AND. TT.LE.CVH(1,NUMCV)) GO TO 68
      CV=0.
      GO TO 100
 68   DO 70 M=2,NUMCV
      IF(TT-CVH(1,M)) 80,90,70
 70   CONTINUE
 80   DT=TT-CVH(1,M-1)
      D1=CVH(1,M)-CVH(1,M-1)
      D2=CVH(2,M)-CVH(2,M-1)
      CV=CVH(2,M-1)+DT*D2/D1
      GO TO 100
 90   IF(CVH(1,M) .EQ. CVH(1,M-1)) GO TO 92
      IF(M.EQ.NUMCV) GO TO 91
```

```
      IF(CVH(1,M) .EQ. CVH(1,M+1)) GO TO 93
91    CV=CVH(2,M)
      GO TO 100
92    IF(LV.EQ.1) CV=CVH(2,M)
      IF(LV.EQ.0) CV=CVH(2,M-1)
      GO TO 100
93    IF(LV.EQ.1) CV=CVH(2,M+1)
      IF(LV.EQ.0) CV=CVH(2,M)
100   RETURN
      END
```

```
      SUBROUTINE LOAD(TT,TTP,L,LL)
C
      COMMON/BK1/NUMMAT,NUMNP,NUMEL,NUMPC,NUMLP, IHED(8),NTAPE,NEQ,ICOND
     1,ISTAT,TMACH, IHED1(8),ISYMT
      COMMON/BK11/DELT,DT1,DT3,BETA,BET1,BET2,BET3,BET4,BET5,NBAND,NBD2
      COMMON/SHIFT/NSBL,NDEL,LASTB,NF,NBRED,NELTYP,ISX,RCODE
      COMMON/PRESS/INI(100),JNJ(100),PI(100),PJ(100),T(100),P(2,20),PF
      COMMON/BK10/R(300),Z(300),CODE(300),IX(6,250)
C
      COMMON/SSS/S1S(36,20),S1SD(36,20),S1SDD(36,20)
      COMMON/MASS/AMRR(36,36),AMRDR(36,36),AM22(2,2),AMR2(36,2)
     1,AMRD2(36,2),AMRDD2(36,2)
      COMMON/WK/WKAREA(1404)
      COMMON/LON/S2(36,36),S4(36,36),S2D(36,36),S2DD(36,36)
      DIMENSION E(800),ELOAD(800),EM(800),EMAT(800)
      DIMENSION US(600),VS(600),AS(600)
      DIMENSION EX1(2160),EX3(1404),EX5(5184)
      EQUIVALENCE(E(1),EX1(1))
      EQUIVALENCE(ELOAD(1),EX1(1001))
      EQUIVALENCE(EMAT(1),EX3(1))
      EQUIVALENCE(US(1),EX5(1))
      EQUIVALENCE(VS(1),EX5(1001))
      EQUIVALENCE(AS(1),EX5(2001))
      EQUIVALENCE(EM(1),EX5(3001))
      EQUIVALENCE(S1S(1,1),EX1(1))
      EQUIVALENCE(WKAREA(1),EX3(1))
      EQUIVALENCE(S2(1,1),EX5(1))
C
C  L=0 FIND APPROPRIATE TIME ONLY AND DETERMINE IF THERE IS A JUMP
C  L=1 GIVEN TIME AND JUMP CONDITION FIND LOAD MAGNITUDE AND LOAD MATRIX
C  L=2 GIVEN TIME AND JUMP CONDITION FIND LOAD MAGNITUDE ONLY
C  L=3 GIVEN TIME AND LOAD MAGNITUDE FIND LOAD MATRIX ONLY
C  LL=0 NO JUMP
C  LL=1     JUMP

      IF(L.EQ.0) GO TO 4
      IF(L.EQ.1) GO TO 60
      IF(L.EQ.2) GO TO 66
      IF(L.EQ.3) GO TO 60
    4 IF(TT.LT.P(1,1)) GO TO 6
      IF(TT.GT.P(1,NUMLP)) GO TO 8
      GO TO 10
    6 TTP=TT+DT1
      IF(TTP.GT.P(1,1)) TTP=P(1,1)
      LL=0
      GO TO 100
    8 TTP=TT+DT1
      LL=0
      GO TO 100
   10 DO 50 I=1,NUMLP
      J=I
      IF(I.EQ.NUMLP) GO TO 20
      IF(TT.GT.P(1,I)) GO TO 50
      IF(TT.EQ.P(1,I) .AND. TT.EQ.P(1,I+1)) GO TO 40
      IF(TT.EQ.P(1,I)) GO TO 30
   20 J=I-1
   30 TTP=TT+DT1
      IF(TTP.GT.P(1,J+1)) TTP=P(1,J+1)
      LL=0
      GO TO 100
   40 IF(PF.EQ.P(2,I)) GO TO 42
      TTP=TT+DT1
      IF(TTP.GT.P(1,I+2)) TTP=P(1,I+2)
      LL=0
      GO TO 100
```

```
   42  TTP=TT
       LL=1
       GO TO 100
   50  CONTINUE

   60  DO 65 I=1,NF
   65  ELOAD(I)=0.
       IF(L.EQ.3) GO TO 95
C
C      APPLY PRESSURE LOAD
C
   66  IF (NUMPC.NE.0) GO TO 67
       PF=0.
       GO TO 100
   67  IF(TT.GE.P(1,1) .AND. TT.LE.P(1,NUMLP)) GO TO 68
       PF=0.
       GO TO 100
   68  DO 70 M=2,NUMLP
       IF(TT-P(1,M)) 80,90,70
   70  CONTINUE
   80  DT=TT-P(1,M-1)
       D1=P(1,M)-P(1,M-1)
       D2=P(2,M)-P(2,M-1)
       PF=P(2,M-1)+DT*D2/D1
       IF(L.EQ.2) GO TO 100
       GO TO 95
   90  IF(P(1,M) .EQ. P(1,M-1)) GO TO 92
       IF(M.EQ.NUMLP) GO TO 91
       IF(P(1,M) .EQ. P(1,M+1)) GO TO 93
   91  PF=P(2,M)
       GO TO 94
   92  IF(LL.EQ.1)PF=P(2,M)
       IF(LL.EQ.0)PF=P(2,M-1)
       GO TO 94
   93  IF(LL.EQ.1)PF=P(2,M+1)
       IF(LL.EQ.0)PF=P(2,M)
   94  IF(L.EQ.2) GO TO 100
   95  IF(NUMPC.EQ.0) GO TO 100
       DO 98 N=1,NUMPC
       IF(TT.LT.T(N)) GO TO 98
       IF (T(N).LT.P(1,1) .OR. T(N).GT.P(1,NUMLP)) GO TO 98
       I=INI(N)
       J=JNJ(N)
       DR=(R(J)-R(I))/6.0
       DZ=(Z(I)-Z(J))/6.0
       Q1=PI(N)
       Q2=PJ(N)
       RX=2.*Q1+Q2
       ZX=Q1+2.*Q2
       I=I+I+NDEL
       II=I-1
       J=J+J+NDEL
       JJ=J-1
       ELOAD(II)=ELOAD(II)+PF*RX*DZ
       ELOAD(JJ)=ELOAD(JJ)+PF*ZX*DZ
       ELOAD(I)=ELOAD(I)+PF*RX*DR
       ELOAD(J)=ELOAD(J)+PF*ZX*DR
   98  CONTINUE
  100  RETURN
       END
```

```
      SUBROUTINE DATOUT(L,TT)
C
      COMMON/BK1/NUMMAT,NUMNP,NUMEL,NUMPC,NUMLP, IHED(8),NTAPE,NEQ,ICOND
     1,ISTAT,TMACH, IHED1(8),ISYMT
      COMMON/BK3/NDSOUT(16),NSTOUT(16),TIO1,TIO2,TIOD,SIOD,NUMDS,NUMST
      COMMON/DIM/NA,NAA,NBT,NB,NQ,NR,NINT,NINT2,IALL,NITER,SINCOD
      COMMON/INTGR/PT(10),WG(10),PT2(2),WG2(2),PEI
      COMMON/DIF/DXX2,DXY3,DXXY,DYX2,DYY2,DYXY,DXX,DXY,DYX,DYY,DEM
      COMMON/IPAR/PJ1,PJ2,PJ3,PJ4,DXY1,DXY2
      COMMON/TIP/NCR1,NCR2,NCR3,NCR4,NELX,CTPX,CTPY,SIF1,SIF2
      COMMON/BK10/R(300),Z(300),CODE(300),IX(6,250)
      COMMON/PRESS/INI(100),JNJ(100),PI(100),PJ(100),T(100),P(2,20),PF
      COMMON/VEL/CV,NUMCV,CVH(2,20)
      COMMON/SUMAN/E(3,3,6),EI(3,3,6),ADET(6),ASIZE(6),CL
      COMMON/SHIFT/NSBL,NDEL,LASTB,NF,NBRED,NELTYP,ISK,RCODE
      COMMON/RHHO/RHO(6),RODUM
      COMMON/RIG/IRIG1(6),IRIG2
      COMMON/OLDDISP/U(600),V(600),A(600)
      DIMENSION C(3,8),A1(2,2),B(2,4),AJJ(3,8)
      DIMENSION U1(8),E1(3),S(3)
      DIMENSION B1(74)
      COMMON/WK2/WKAR(720)
      EQUIVALENCE(B(1,1),B1(1))
      EQUIVALENCE(AJJ(1,1),B1(9))
      EQUIVALENCE(C(1,1),B1(33))
      EQUIVALENCE(A1(1,1),B1(57))
      EQUIVALENCE(U1(1),B1(61))
      EQUIVALENCE(E1(1),B1(69))
      EQUIVALENCE(S(1),B1(72))
      EQUIVALENCE(B1(1),WKAR(1))
      COMMON/STRESS/RX(3,36),RXD(3,36),RXDD(3,36)
      COMMON/BET/BETA1(36),BETA1D(36),BETA2D(2)
C
      WRITE(6,2001) TT,CV,CTPX,CTPY,PF
      WRITE(6,408) SIF1,SIF2
      WRITE (6,109)
      DO 10 M=1,NUMDS
      N=M
      IF(NUMDS.EQ.NUMNP) GO TO 5
      N=NDSOUT(M)
 5    I=2*N-1
      J=I+1
      WRITE(6,120) N,U(I),U(J),V(I),V(J),A(I),A(J),N
 10   CONTINUE
      WRITE(6,108)
      IF(L.EQ.0 .AND. ICOND.EQ.0) GO TO 20
      DO 12 I=1,NUMMAT
      IF(IRIG1(I).EQ.1) GO TO 15
 12   CONTINUE
      GO TO 20
 15   WRITE(6,510)
 20   WRITE(6,130)
      IF(L.EQ.0 .AND. ICOND.EQ.0) GO TO 25
      GO TO 30
 25   CONTINUE
      S(1)=0.
      S(2)=0.
      S(3)=0.
      DO 29 M=1,NUMST
      N=M
      IF(NUMST.EQ.NUMEL) GO TO 27
      N=NSTOUT(M)
 27   CONTINUE
      WRITE(6,140) N,(S(I),I=1,3),N
 29   CONTINUE
```

```
      GO TO 165
C
30    DO 160 M=1,NUMST
      N=M
      IF(NUMST.EQ.NUMEL) GO TO 32
      N=NSTOUT(M)
32    CONTINUE
      IF(N.EQ.NCR1) GO TO 101
      IF(N.EQ.NCR2) GO TO 102
      IF(N.EQ.NCR3) GO TO 103
      IF(N.EQ.NCR4) GO TO 104
      X=0.
      Y=0.
      CALL TRANS(X,Y,N,2,2)
      CALL DIFFB(X,Y)
      N1=IX(1,N)*2
      N2=IX(2,N)*2
      N3=IX(3,N)*2
      N4=IX(4,N)*2
      U1(1)=U(N1-1)
      U1(2)=U(N1)
      U1(3)=U(N2-1)
      U1(4)=U(N2)
      U1(5)=U(N3-1)
      U1(6)=U(N3)
      U1(7)=U(N4-1)
      U1(8)=U(N4)
      DO 50 I=1,3
      E1(I)=0.
      DO 50 J=1,8
      E1(I)=E1(I)+C(I,J)*U1(J)
  50  CONTINUE
      K=IX(6,N)
      DO 55 I=1,3
      S(I)=0.
      DO 55 J=1,3
      S(I)=S(I)+EI(I,J,K)*E1(J)
 55   CONTINUE
      GO TO 155
C
 101  X=-.5
      Y=+.5
      GO TO 105
 102  X=-.5
      Y=-.5
      GO TO 105
 103  X=+.5
      Y=+.5
      GO TO 105
 104  X=+.5
      Y=-.5
 105  CONTINUE
      CALL TRANS(X,Y,N,2,1)
      CALL FUNCTS(X,Y,8)
      DO 110 J=1,3
      S(J)=0.
      DO 110 I=1,NA
      S(J)= S(J)+RX(J,I)*BETA1(I)
110   CONTINUE
155   WRITE(6,140) N,(S(I),I=1,3),N
160   CONTINUE
165   CONTINUE
      RETURN
C
 108  FORMAT (1H1)
```

```
 1Ø9  FORMAT (            //,5X,118H NODAL POINT R-DISPLACEMENT  Z-DISPLA
     1CEMENT    R-VELOCITY       Z-VELOCITY  R-ACCELERATION  Z-ACCELERATI
     2ON  NODAL POINT)
  12Ø FORMAT(1ØX,I4,1P6E16.4,I9)
 13Ø  FORMAT (            //,2ØX,69H ELEMENT NO.   XX-STRESS       YY-STRES
     1S         XY-STRESS  ELEMENT NO.)
  14Ø FORMAT(25X,I4,1P8E16.4,I9)
  4Ø8 FORMAT(/,4ØX,6H    K1=,1PE12.4,6H    K2=,1PE12.4)
 51Ø  FORMAT(//,2ØX,2ØH*****THE STRESSES IN,
     1 6ØHTHE RIGID MATERIALS, EXEPT FOR THE SINGULAR ELEMENT, ARE SET,
     2 /,25X,37HARBITRARILY EQUAL TO ZERO AND DO NOT ,
     362HREPRESNT THE TRUE VALUES OF THE STRESSES IN THIS MATERIAL*****)
2ØØ1  FORMAT(1H1,//,1ØX,12HCURRENT TIME,1ØX,18HCRACK-TIP VELOCITY,1ØX,
     12ØHCRACK-TIP POSITION-R,1ØX,2ØHCRACK-TIP POSITION-Z,1ØX,4HLOAD,/,
     21ØX,1PE13.4,12X,1PE13.4,16X,1PE13.4,16X,1PE13.4,9X,1PE13.4)
C
      END
```

```
      SUBROUTINE TAPOUT(TT)
      COMMON/VEL/CV,NUMCV,CVH(2,20)
      COMMON/PRESS/INI(100),JNJ(100),PI(100),PJ(100),T(100),P(2,20),PF
      COMMON/BK10/ R(300),Z(300),CODE(300),IX(6,250)
      COMMON/PAIR1/NPAIR,LPAIR(3,40)
      COMMON/IPLN/IPLN(5),IPLN1(2)
      COMMON/RHHO/RHO(6),RODUM
      COMMON/SUMAN/A1(3,3,6),A1I(3,3,6),ADET(6),ASIZE(6),CL
      COMMON/SHIFT/NSBL,NDEL,LASTB,NF,NBRED,NELTYP,ISK,RCODE
      COMMON/BK11/DELT,DT1,DT3,BETA,BET1,BET2,BET3,BET4,BET5,NBAND,NBD2
      COMMON/TIP/NCR1,NCR2,NCR3,NCR4,NELX,CTPX,CTPY,SIF1,SIF2
      COMMON/DIM/NA,NAA,NBT,NB,NQ,NR,NINT,NINT2,IALL,NITER,SINCOD
      COMMON/BK1/NUMMAT,NUMNP,NUMEL,NUMPC,NUMLP, IHED(8),NTAPE,NEQ,ICOND
     1,ISTAT,TMACH, IHED1(8),ISYMT
      COMMON/OLDDISP/U(600),V(600),A(600)
      COMMON/MISL/IF(20)
      COMMON/MAIN/CORD(10,2)
      COMMON/RIG/IRIG1(6),IRIG2
C
      COMMON/AB/P1S(36,20),P1SD(36,20),P1SDD(36,20)
     1 ,AT(2,20),ATD(2,20),ATDD(2,20)
      COMMON/MMAT/AM1(20,20),V1(20,20),AK4(20,20)
      COMMON/KMAT/AK1S(20,20)
      COMMON/ELM/ESTIFM(8,8,50),ELMASS(8,8,50),VSS(20,20)
      COMMON/BET/BETA1(36),BETA1D(36),BETA2D(2)
C
      REWIND 16
      WRITE(16) NUMMAT,NELTYP,NUMNP,NEQ,NUMEL,IALL,NBAND,NBRED,IHED1
      WRITE(16) IPLN,IPLN1
      WRITE(16) (((A1(I,J,K),I=1,3),J=1,3),K=1,NUMMAT)
      WRITE(16) (((A1I(I,J,K),I=1,3),J=1,3),K=1,NUMMAT)
      WRITE(16) (RHO(K),ADET(K),ASIZE(K),K=1,NUMMAT),CL
      WRITE(16) (IRIG1(K),K=1,NUMMAT),IRIG2
      WRITE(16) NBD2,NSBL,NDEL,LASTB,NF
      WRITE(16) (IF(I),I=1,NQ)
      NQ1=NQ/2
      WRITE(16) ((CORD(I,J),I=1,NQ1),J=1,2)
      WRITE(16) TT,CV,CTPX,CTPY,PF,SIF1,SIF2,NBT,NINT,NA
      WRITE(16) NCR1,NCR2,NCR3,NCR4,NELX
      WRITE(16) (U(I),I=1,NEQ)
      WRITE(16) (V(I),I=1,NEQ)
      WRITE(16) (A(I),I=1,NEQ)
      WRITE(16) ((AK4(I,J),I=1,NQ),J=1,NQ)
      WRITE(16) ((AK1S(I,J),I=1,NQ),J=1,NQ)
      WRITE(16) ((V1(I,J),I=1,NQ),J=1,NQ)
      WRITE(16) ((VSS(I,J),I=1,NQ),J=1,NQ)
      WRITE(16) ((AM1(I,J),I=1,NQ),J=1,NQ)
      WRITE(16) (((ESTIFM(I,J,K),I=1,8),J=1,8),K=1,NELTYP)
      WRITE(16) (((ELMASS(I,J,K),I=1,8),J=1,8),K=1,NELTYP)
      WRITE(16) (BETA1(I),I=1,NA)
      WRITE(16) ((P1S(I,J),I=1,NA),J=1,NQ)
      WRITE(16) ((AT(I,J),I=1,NR),J=1,NQ)
      IF(CV.EQ.0.) GO TO 10
      WRITE(16) ((P1SD(I,J),I=1,NA),J=1,NQ)
      WRITE(16) ((P1SDD(I,J),I=1,NA),J=1,NQ)
      WRITE(16) ((ATD(I,J),I=1,NR),J=1,NQ)
      WRITE(16) ((ATDD(I,J),I=1,NR),J=1,NQ)
   10 WRITE(16) (CODE(I),R(I),Z(I),I=1,NUMNP)
      WRITE(16) ((IX(I,J),I=1,6),J=1,NUMEL)
      WRITE(16) NPAIR,((LPAIR(I,J),I=1,3),J=1,NPAIR)
      RETURN
      END
```

```
      SUBROUTINE TAPIN(TT)
      COMMON/VEL/CV,NUMCV,CVH(2,20)
      COMMON/PRESS/INI(100),JNJ(100),PI(100),PJ(100),T(100),P(2,20),PF
      COMMON/BK10/ R(300),Z(300),CODE(300),IX(6,250)
      COMMON/PAIR1/NPAIR,LPAIR(3,40)
      COMMON/IPLN/IPLN(5),IPLN1(2)
      COMMON/RHHO/RHO(6),RODUM
      COMMON/SUMAN/A1(3,3,6),A1I(3,3,6),ADET(6),ASIZE(6),CL
      COMMON/SHIFT/NSBL,NDEL,LASTB,NF,NBRED,NELTYP,ISK,RCODE
      COMMON/BK11/DELT,DT1,DT3,BETA,BET1,BET2,BET3,BET4,BET5,NBAND,NBD2
      COMMON/TIP/NCR1,NCR2,NCR3,NCR4,NELX,CTPX,CTPY,SIF1,SIF2
      COMMON/DIM/NA,NAA,NBT,NB,NQ,NR,NINT,NINT2,IALL,NITER,SINCOD
      COMMON/BK1/NUMMAT,NUMNP,NUMEL,NUMPC,NUMLP, IHED(8),NTAPE,NEQ,ICOND
     1,ISTAT,TMACH, IHED1(8),ISYMT
      COMMON/OLDDISP/U(600),V(600),A(600)
      COMMON/MISL/IF(20)
      COMMON/MAIN/CORD(10,2)
      COMMON/RIG/IRIG1(6),IRIG2
C
      COMMON/AB/P1S(36,20),P1SD(36,20),P1SDD(36,20)
     1 ,AT(2,20),ATD(2,20),ATDD(2,20)
      COMMON/MMAT/AM1(20,20),V1(20,20),AK4(20,20)
      COMMON/KMAT/AK1S(20,20)
      COMMON/ELM/ESTIFM(8,8,50),ELMASS(8,8,50),VSS(20,20)
      COMMON/BET/BETA1(36),BETA1D(36),BETA2D(2)
C
      REWIND 16
      READ(16) NUMMAT,NELTYP,NUMNP,NEQ,NUMEL,IALL,NBAND,NBRED,IHED
      READ(16) IPLN,IPLN1
      READ(16) (((A1(I,J,K),I=1,3),J=1,3),K=1,NUMMAT)
      READ(16) (((A1I(I,J,K),I=1,3),J=1,3),K=1,NUMMAT)
      READ(16) (RHO(K),ADET(K),ASIZE(K),K=1,NUMMAT),CL
      DO 5 K=1,NUMMAT
      WRITE(6,310) K
      WRITE(6,311) RHO(K)
    5 CONTINUE
      READ(16) (IRIG1(K),K=1,NUMMAT),IRIG2
      READ(16) NBD2,NSBL,NDEL,LASTB,NF
      READ(16) (IF(I),I=1,NQ)
      NQ1=NQ/2
      READ(16) ((CORD(I,J),I=1,NQ1),J=1,2)
      READ(16) TT,CV,CTPX,CTPY,PF,SIF1,SIF2,NBT1,NINTR,NA1
      READ(16) NCR1,NCR2,NCR3,NCR4,NELX
      READ(16) (U(I),I=1,NEQ)
      READ(16) (V(I),I=1,NEQ)
      READ(16) (A(I),I=1,NEQ)
      READ(16) ((AK4(I,J),I=1,NQ),J=1,NQ)
      READ(16) ((AK1S(I,J),I=1,NQ),J=1,NQ)
      READ(16) ((V1(I,J),I=1,NQ),J=1,NQ)
      READ(16) ((VSS(I,J),I=1,NQ),J=1,NQ)
      READ(16) ((AM1(I,J),I=1,NQ),J=1,NQ)
      READ(16) (((ESTIFM(I,J,K),I=1,8),J=1,8),K=1,NELTYP)
      READ(16) (((ELMASS(I,J,K),I=1,8),J=1,8),K=1,NELTYP)
      READ(16) (BETA1(I),I=1,NA1)
      READ(16) ((P1S(I,J),I=1,NA1),J=1,NQ)
      READ(16) ((AT(I,J),I=1,NR),J=1,NQ)
      IF(CV.EQ.0.) GO TO 10
      READ(16) ((P1SD(I,J),I=1,NA1),J=1,NQ)
      READ(16) ((P1SDD(I,J),I=1,NA1),J=1,NQ)
      READ(16) ((ATD(I,J),I=1,NR),J=1,NQ)
      READ(16) ((ATDD(I,J),I=1,NR),J=1,NQ)
   10 READ(16) (CODE(I),R(I),Z(I),I=1,NUMNP)
      READ(16) ((IX(I,J),I=1,6),J=1,NUMEL)
      READ(16) NPAIR,((LPAIR(I,J),I=1,3),J=1,NPAIR)
      IF(IRIG2.EQ.0) GO TO 12
```

```
      NBT1=NBT1+2
      IF(NBT.EQ.NBT1 .OR. (NBT+1).EQ.NBT1) GO TO 13
      SINCOD=1.
      GO TO 14
12    IF(NBT1.NE.NBT) SINCOD=1.
13    IF(NINTR.NE.NINT) SINCOD=1.
14    IF(NINTR.NE.NINT) RCODE=1.
      RETURN
 310 FORMAT(/,50X,11HMATERIAL # ,I2)
 311 FORMAT(/,40X,25H DENSITY------------------,1PE12.4)
     END
```

```
      SUBROUTINE SINGEL
C
      COMMON/BK15/ R(300),Z(300),CODE(300),IX(6,250)
      COMMON/RHHO/RHO(6),RODUM
      COMMON/TIP/NCR1,NCR2,NCR3,NCR4,NELX,CTPX,CTPY,SIF1,SIF2
      COMMON/MMAT/AM1(20,20),V1(20,20),AK4(20,20)
      COMMON/KMAT/AK1S(20,20)
      COMMON/ELM/ESTIFM(8,8,50),ELMASS(8,8,50),VSS(20,20)
      COMMON/VEL/CV,NUMCV,CVH(2,20)
      COMMON/DIM/NA,NAA,NBT,NB,NQ,NR,NINT,NINT2,IALL,NITER,SINCOD
      COMMON/LON/S2(36,36),S4(36,36),S2D(36,36),S2DD(36,36)
      COMMON/WK/WKAREA(1404)
      COMMON/MISL/IF(20)
C
C
      CALL LINEI
C
C     NORMALIZE S2 FOR NUMERICAL ACCURACY IN THE INVERSION
      SSS=S2(1,1)
      IF(SSS.EQ.0.) GO TO 91
      DO 12 J=1,NB
      DO 12 I=1,NB
      S2(I,J)=S2(I,J)/SSS
 12   CONTINUE
      IDIGT=3
      IDIGT1=3
      CALL LINV2F(S2,NA,NAA,S4,IDIGT1,WKAREA,IER)
      IF(IER.EQ.129 .OR. IER.EQ.131) GO TO 90
      DO 14 J=1,NB
      DO 14 I=1,NB
      S2(I,J)=S2(I,J)*SSS
      S4(I,J)=S4(I,J)/SSS
 14   CONTINUE

C
      CALL STIFK
C
      IF(RODUM.EQ.0.) GO TO 40

      CALL MAREAI
C
      CALL MASSM
C
      IF(CV .EQ. 0.) GO TO 40
      DO 20 J=1,NQ
      DO 20 I=1,NQ
      AK4(I,J)=AK4(I,J)+AK1S(I,J)
 20   CONTINUE

      DO 25 J=1,NQ
      DO 25 I=1,NQ
      AK1S(I,J)=.5*(AK4(I,J)+AK4(J,I))
      VSS(I,J)=.5*(V1(I,J)+V1(J,I))
 25   CONTINUE
      DO 30 J=1,NQ
      DO 30 I=1,NQ
      AK4(I,J)=AK4(I,J)-AK1S(I,J)
      V1(I,J)=V1(I,J)-VSS(I,J)
 30   CONTINUE

      GO TO 60
 40   CONTINUE
C FOR CV=0. "AK1S" IS AUTOMATICALLY SYMMETRIC EXEPT MAYBE FOR NUMERICAL
C TRUNCATIONS, THUS DO THE FOLLOWING TO GET A PERFECTLY SYMMETRIC "AK1S"
C AND THEN SET "AK4(I,J)=0." DOWN BELOW
```

```
      DO 45 J=1,NQ
      DO 45 I=1,NQ
      AK4(I,J)=.5*(AK1S(I,J)-AK1S(J,I))
 45   CONTINUE
      DO 5Ø J=1,NQ
      DO 5Ø I=1,NQ
      AK1S(I,J)=AK1S(I,J)-AK4(I,J)
 5Ø   CONTINUE
      IF(RODUM.EQ.Ø.) GO TO 68
      DO 55 J=1,NQ
      DO 55 I=1,NQ
      VSS(I,J)=Ø.
      V1(I,J)=Ø.
      AK4(I,J)=Ø.
55    CONTINUE
 68   CONTINUE
      RETURN
  9Ø  WRITE(6,3Ø1) IER,IDIGT,IDIGT1
      STOP 5
  91  WRITE(6,3Ø2)
      STOP 6
 3Ø1  FORMAT(/,2ØX,29H*****INVERS FAILED ON S2*****,/2ØX,9HERROR NO.,I4,
     1/2X,39HPROPOSED DECIMAL DIGITS OF ACCURACY WAS,I4,2X,51HBUT THE UW
     2-IMSL LIBRARY ROUTINE "LINV2F" FOUND ONLY,I4,2X,26HDECIMAL DIGITS
     3OF ACCURACY,
     4/2X,47HERROR NO. 34=DECIMAL DIGIT ACCURACY TEST FAILED,
     5/2X,48HERROR NO. 129=MATRIX IS ALGORITHMICALLY SINGULAR,
     6/2X,85HERROR NO. 131=MATRIX IS TOO ILL-CONDITONED FOR ITERATIVE IM
     7PROVEMENTS TO BE EFFECTIVE,
     8//,1ØX,6ØH*****TRY CHANGING MATERIAL PROPERIES OR CRACK-TIP SPEED*
     9****)
 3Ø2  FORMAT(/,2ØX,29H*****INVERS FAILED ON S2*****,13H   S2(1,1)=Ø.)
C
       END
```

```
      SUBROUTINE LINEI
C
      COMMON/RHHO/RHO(6),RODUM
      COMMON/LON/S2(36,36),S4(36,36),S2D(36,36),S2DD(36,36)
      COMMON/SSS/S1S(36,20),S1SD(36,20),S1SDD(36,20)
      COMMON/INTGR/PT(10),WG(10),PT2(2),WG2(2),PEI
      COMMON/SUMAN/A1(3,3,6),A1I(3,3,6),ADET(6),ASIZE(6),CL
      COMMON/DIF/DXX2,DXY3,DXXY,DYX2,DYY2,DYXY,DXX,DXY,DYX,DYY,DEM
      COMMON/IPAR/PJ1,PJ2,PJ3,PJ4,DXY1,DXY2
      COMMON/TIP/NCR1,NCR2,NCR3,NCR4,NELX,CTPX,CTPY,SIF1,SIF2
      COMMON/VEL/CV,NUMCV,CVH(2,20)
      COMMON/DIM/NA,NAA,NBT,NB,NQ,NR,NINT,NINT2,IALL,NITER,SINCOD
      COMMON/PHYS/ANX,ANY,A(2)
      COMMON/BK10/R1(300),Z(300),CODE(300),IX(6,250)
      COMMON/DISP/U(2,36),UD(2,36),UDD(2,36),UR(2,2)
      COMMON/STRESS/RX(3,36),RXD(3,36),RXDD(3,36)
      COMMON/RLIN/G2(2,20),P3T(2,36),P2T(2,2),P3TD(2,36),P3TDD(2,36)
     1 ,P4(36,2),P4D(36,2),P4DD(36,2)
      COMMON/MASS/AMRR(36,36),AMRDR(36,36),AM22(2,2),AMR2(36,2)
     1 ,AMRD2(36,2),AMRDD2(36,2)
      COMMON/WK/WKAREA(1404)
      COMMON/INTPOL/ALS(2,20)
      COMMON/RIG/IRIG1(6),IRIG2
      DIMENSION R(2,36),RD(2,36),RDD(2,36)
      DIMENSION R2(2,2)
      EQUIVALENCE (R(1,1),WKAREA(1))
      EQUIVALENCE (RD(1,1),WKAREA(101))
      EQUIVALENCE (RDD(1,1),WKAREA(201))
      EQUIVALENCE (R2(1,1),WKAREA(301))
      COMMON/S2RIG/S2RIG(36)
      DIMENSION RRIG(2)
      DIMENSION EPS(3,36)
      EQUIVALENCE (RRIG(1),WKAREA(401))
      EQUIVALENCE (EPS(1,1),WKAREA(501))
C
C NOTE THAT MATRIX H1 AND MATRIX P11=INTEGRAL(R1TU1) ARE THE TRANSPOSE
C OF EACH OTHER EXEPT FOR THE FORTH ROW OF MATRIX P11 WHICH CORRESPONDS
C TO RIGID BODY ROTATION
C IN THIS SUBROUTINE THE MATRIX S2 IS CALCULATED AS THE INTEGRAL OF
C R1TU1 WHERE THE ELEMENTS OF SURFACE TRACTION MATRIX R ARE ASSUMED
C TO BE DERIVED FROM THE INTERNAL STRESSES OF THE SINGULAR ELEMENT
C BUT SINCE THE TERMS CORRESPONDING TO RIGID BODY ROTATIONS DO NOT
C CONTIBUTE TO THESE TRACIONS, SOME ARBIRARY TRACTIONS CORRESPONDING
C TO A RIGID BODY ROTAION ARE CHOSEN IN ORDER TO MAKE THE MATIX
C S2=P11 NON-SINGULAR,THESE TRACIONS ARE PLACED IN THE FORTH COLUMN
C OF MATRIX R,HOWEVER THIS WILL CHANGE THE TRUE VALUES OF THE ELEMENTS
C IN THE FORTH ROW OF S2,THEREFORE WE HAVE CALCULATED THE MATRIX S2RIG
C WHICH CONTAINS THE TRUE VALUES OF THE FOTH ROW OF S2. THESE TRUE
C VALUES OF S2 ARE REPLACED WITH THE PSEDO-VALUES AT THE END OF
C THE SUROUTINE STIFX, WHERE THE TRUE VALUES OF S2=H1 ARE NEEDED TO
C CALCULATE BHB
      IF(CV .EQ. 0.)GO TO 400
      DO 3 J=1,NQ
      DO 3 I=1,NA
      S1S(I,J)=0.
      S1SD(I,J)=0.
      S1SDD(I,J)=0.
    3 CONTINUE
      DO 4 J=1,NR
      DO 4 I=1,NA
      P4(I,J)=0.
      P4D(I,J)=0.
      P4DD(I,J)=0.
    4 CONTINUE
```

```
      DO  5 J=1,NB
      DO  5 I=1,NA
      S2(I,J) =0.
      S2D(I,J)=0.
      S2DD(I,J)=0.
    5 CONTINUE
      DO  6 I=1,NA
      S2RIG(I) =0.
    6 CONTINUE
      DO 7 J=1,NQ
      DO 7 I=1,NR
    7 G2(I,J)=0.
      DO 8 I=1,NR
      DO 8 J=1,NB
      AMRD2(J,I)=0.
      AMRDD2(J,I)=0.
      P3T(I,J)=0.
      P3TD(I,J)=0.
      P3TDD(I,J)=0.
    8 CONTINUE
      DO 9 J=1,NR
      DO 9 I=1,NR
      P2T(I,J)=0.
    9 CONTINUE
      GA=.5
C
C           LOOP ON SIDES
C
      DO 300 LSIDE=1,4
      CALL NORMAL(LSIDE)
      IF(LSIDE.EQ.1) GO TO 10
      IF(LSIDE.EQ.2) GO TO 15
      IF(LSIDE.EQ.3) GO TO 20
      IF(LSIDE.EQ.4) GO TO 25
   10 CONTINUE
      Y=-1.
      R2(1,1)=0.
      R2(2,1)=0.
      R2(1,2)=0.
      R2(2,2)=1.
      RRIG(1)=1.
      RRIG(2)=0.
      GO TO 30
   15 CONTINUE
      X=1.
      R2(1,1)=1.
      R2(2,1)=0.
      R2(1,2)=0.
      R2(2,2)=0.
      RRIG(1)=0.
      RRIG(2)=1.
      GO TO 30
   20 CONTINUE
      Y=1.
      R2(1,1)=0.
      R2(2,1)=0.
      R2(1,2)=0.
      R2(2,2)=1.
      RRIG(1)=-1.
      RRIG(2)=0.
      GO TO 30
   25 CONTINUE
      X=-1.
      R2(1,1)=1.
      R2(2,1)=0.
```

```
      R2(1,2)=0.
      R2(2,2)=0.
      RRIG(1)=0.
      RRIG(2)=-1.
   30 CONTINUE
      DO 200 LH=1,2
      DO 100 L=1,NINT
      IF(LSIDE.EQ.1.AND.LH.EQ.1) GO TO 3001
      IF(LSIDE.EQ.1.AND.LH.EQ.2) GO TO 3002
      IF(LSIDE.EQ.2.AND.LH.EQ.1) GO TO 3003
      IF(LSIDE.EQ.2.AND.LH.EQ.2) GO TO 3004
      IF(LSIDE.EQ.3.AND.LH.EQ.1) GO TO 3005
      IF(LSIDE.EQ.3.AND.LH.EQ.2) GO TO 3006
      IF(LSIDE.EQ.4.AND.LH.EQ.1) GO TO 3007
      IF(LSIDE.EQ.4.AND.LH.EQ.2) GO TO 3008
 3001 X=.5*(PT(L)-1.)
      GO TO 3009
 3002 X=.5*(PT(L)+1.)
      GO TO 3009
 3003 Y=.5*(PT(L)-1.)
      GO TO 3009
 3004 Y=.5*(PT(L)+1.)
      GO TO 3009
 3005 X=.5*(PT(L)+1.)
      GO TO 3009
 3006 X=.5*(PT(L)-1.)
      GO TO 3009
 3007 Y=.5*(PT(L)+1.)
      GO TO 3009
 3008 Y=.5*(PT(L)-1.)
 3009 KS=IX(6,NCR1)
      IF(Y.LT.0.) KS=IX(6,NCR2)
      RO=RHO(KS)
      WT=WG(L)
      CALL TRANS(X,Y,N,2,1)
C SCALE TO  R2  AND  RRIG  TO COMPARE WITH THE VALUES OF OTHER ELEMENTS
C OF MATRIX R (THIS IS ONLY FOR NUMERICAL ACCURACY,ESPECIALLY FOR
C (THIS IS ONLY FOR NUMERICAL ACCURACY,ESPECIALLY FOR INVERSION PROCESS)
      R2(1,1)=R2(1,1)*PJ1
      R2(1,2)=R2(1,2)*PJ1
      R2(2,1)=R2(2,1)*PJ1
      R2(2,2)=R2(2,2)*PJ1
      RRIG(1)=RRIG(1)*PJ1
      RRIG(2)=RRIG(2)*PJ1
      SCALE=DXY1
      IF(LSIDE.EQ.2.OR.LSIDE.EQ.4) SCALE=DXY2
      DY=ANX*SCALE
      WS=WT*SCALE*GA
      WY1=-WT*DY*GA*CV
      WY=WY1*RO
      CALL FUNCTS(X,Y,2)
      DO 35 M=1,NA
      RD(1,M)=ANX*RXD(1,M)+ANY*RXD(3,M)
      RDD(1,M)=ANX*RXDD(1,M)+ANY*RXDD(3,M)
      RD(2,M)=ANX*RXD(3,M)+ANY*RXD(2,M)
      RDD(2,M)=ANX*RXDD(3,M)+ANY*RXDD(2,M)
      R(1,M)=ANX*RX(1,M)+ANY*RX(3,M)
      R(2,M)=ANX*RX(3,M)+ANY*RX(2,M)
   35 CONTINUE
C   REPLACE THE ELEMENTS OF THE FORTH COLUMN OF R WITH THE PSUDO-VALUES
C   CORRRESPONDING TO RIGID BODY ROTATION AND SAVE THE TRUE VALUES OF THE
C   FORTH COLUMN OF R IN MATRIX RRIG IN ORDER TO CALCULATE THE TRUE
C   VALUES OF THE FORTH ROW OF MATRIX S2 IN MATRIX S2RIG
      IF(IRIG2.EQ.1) GO TO 36
      IF(NBT.LT.2) GO TO 36
```

```
      RR1=R(1,4)
      RR2=R(2,4)
      R(1,4)=RRIG(1)
      R(2,4)=RRIG(2)
      RRIG(1)=RR1
      RRIG(2)=RR2
 36   CONTINUE
      DO 40 J=1,NB
      DO 40 I=1,NA
      DO 40 K=1,2
      S2(I,J)=S2(I,J)+R(K,I)*U(K,J)*WS
      S2D(I,J)=S2D(I,J)+(RD(K,I)*U(K,J)+R(K,I)*UD(K,J))*WS
      S2DD(I,J)=S2DD(I,J)+(RDD(K,I)*U(K,J)+2.*RD(K,I)*UD(K,J)+
     1                    R(K,I)*UDD(K,J))*WS
 40   CONTINUE
      IF(IRIG2.EQ.1) GO TO 46
      IF(NBT.LT.2) GO TO 46
      DO 45 I=1,NA
      DO 45 J=1,2
      S2RIG(I)=S2RIG(I)+RRIG(J)*U(J,I)*WS
 45   CONTINUE
 46   CONTINUE
      DO 50 J=1,NR
      DO 50 I=1,NA
      DO 50 K=1,2
      P4(I,J)=P4(I,J)+R(K,I)*UR(K,J)*WS
      P4D(I,J)=P4D(I,J)+RD(K,I)*UR(K,J)*WS
      P4DD(I,J)=P4DD(I,J)+RDD(K,I)*UR(K,J)*WS
 50   CONTINUE
      CALL INPOL(X,Y,LSIDE,LH)
      DO 60 J=1,NQ
      DO 60 I=1,NA
      DO 60 K=1,2
      S1S(I,J)=S1S(I,J)+R(K,I)*ALS(K,J)*WS
      S1SD(I,J)=S1SD(I,J)+RD(K,I)*ALS(K,J)*WS
      S1SDD(I,J)=S1SDD(I,J)+RDD(K,I)*ALS(K,J)*WS
 60   CONTINUE
      DO 85 J=1,NQ
      DO 85 I=1,NR
      DO 85 K=1,2
 85   G2(I,J)=G2(I,J)+R2(K,I)*ALS(K,J)*WS
      DO 90 J=1,NB
      DO 90 I=1,NR
      DO 90 K=1,2
      AMRD2(J,I)=AMRD2(J,I)+U(K,J)*UR(K,I)*WY
      AMRDD2(J,I)=AMRDD2(J,I)+UD(K,J)*UR(K,I)*WY
      P3TD(I,J)=P3TD(I,J)+R2(K,I)*UD(K,J)*WS
      P3TDD(I,J)=P3TDD(I,J)+R2(K,I)*UDD(K,J)*WS
 90   P3T(I,J)= P3T(I,J)+R2(K,I)*U(K,J)*WS
      DO 95 J=1,NR
      DO 95 I=1,NR
      DO 95 K=1,2
 95   P2T(I,J)=P2T(I,J)+R2(K,I)*UR(K,J)*WS
      DO 98 J=1,NB
      DO 98 I=1,3
      EPS(I,J)=0.
      DO 98 K=1,3
 98   EPS(I,J)=EPS(I,J)+A1(I,K,KS)*RX(K,J)
100   CONTINUE
200   CONTINUE
300   CONTINUE
      RETURN

400   CONTINUE
      DO 403 J=1,NQ
```

```
      DO 403 I=1,NA
      S1S(I,J)=0.
  403 CONTINUE
      DO 404 J=1,NB
      DO 404 I=1,NA
      P4(I,J)=0.
  404 CONTINUE
      DO  405 J=1,NB
      DO  405 I=1,NA
      S2(I,J) =0.
  405 CONTINUE
      DO  406 I=1,NA
      S2RIG(I) =0.
  406 CONTINUE
      DO 407 J=1,NQ
      DO 407 I=1,NB
  407 G2(I,J)=0.
      DO 408 J=1,NB
      DO 408 I=1,NR
      P3T(I,J)=0.
  408 CONTINUE
      DO 409 J=1,NR
      DO 409 I=1,NR
      P2T(I,J)=0.
  409 CONTINUE
      GA=.5
C
C           LOOP ON SIDES
C
      DO 700 LSIDE=1,4
      CALL NORMAL(LSIDE)
      IF(LSIDE.EQ.1) GO TO 410
      IF(LSIDE.EQ.2) GO TO 415
      IF(LSIDE.EQ.3) GO TO 420
      IF(LSIDE.EQ.4) GO TO 425
  410 CONTINUE
      Y=-1.
      R2(1,1)=0.
      R2(2,1)=0.
      R2(1,2)=0.
      R2(2,2)=1.
      RRIG(1)=1.
      RRIG(2)=0.
      GO TO 430
  415 CONTINUE
      X=1.
      R2(1,1)=1.
      R2(2,1)=0.
      R2(1,2)=0.
      R2(2,2)=0.
      RRIG(1)=0.
      RRIG(2)=1.
      GO TO 430
  420 CONTINUE
      Y=1.
      R2(1,1)=0.
      R2(2,1)=0.
      R2(1,2)=0.
      R2(2,2)=1.
      RRIG(1)=-1.
      RRIG(2)=0.
      GO TO 430
  425 CONTINUE
      X=-1.
      R2(1,1)=1.
```

```
      R2(2,1)=0.
      R2(1,2)=0.
      R2(2,2)=0.
      RRIG(1)=0.
      RRIG(2)=-1.
  430 CONTINUE
      DO 600 LH=1,2
      DO 500 L=1,NINT
      IF(LSIDE.EQ.1.AND.LH.EQ.1) GO TO 4001
      IF(LSIDE.EQ.1.AND.LH.EQ.2) GO TO 4002
      IF(LSIDE.EQ.2.AND.LH.EQ.1) GO TO 4003
      IF(LSIDE.EQ.2.AND.LH.EQ.2) GO TO 4004
      IF(LSIDE.EQ.3.AND.LH.EQ.1) GO TO 4005
      IF(LSIDE.EQ.3.AND.LH.EQ.2) GO TO 4006
      IF(LSIDE.EQ.4.AND.LH.EQ.1) GO TO 4007
      IF(LSIDE.EQ.4.AND.LH.EQ.2) GO TO 4008
 4001 X=.5*(PT(L)-1.)
      GO TO 4009
 4002 X=.5*(PT(L)+1.)
      GO TO 4009
 4003 Y=.5*(PT(L)-1.)
      GO TO 4009
 4004 Y=.5*(PT(L)+1.)
      GO TO 4009
 4005 X=.5*(PT(L)+1.)
      GO TO 4009
 4006 X=.5*(PT(L)-1.)
      GO TO 4009
 4007 Y=.5*(PT(L)+1.)
      GO TO 4009
 4008 Y=.5*(PT(L)-1.)
 4009 RO=RHO(IX(6,NCR1))
      IF(Y.LT.0.) RO=RHO(IX(6,NCR2))
      WT=WG(L)
      CALL TRANS(X,Y,N,2,1)
C SCALE TO  R2   AND  RRIG  TO COMPARE WITH THE VALUES OF OTHER ELEMENTS
C OF MATRIX R (THIS IS ONLY FOR NUMERICAL ACCURACY,ESPECIALLY FOR
C (THIS IS ONLY FOR NUMERICAL ACCURACY,ESPECIALLY FOR INVERSION PROCESS)
      R2(1,1)=R2(1,1)*PJ1
      R2(1,2)=R2(1,2)*PJ1
      R2(2,1)=R2(2,1)*PJ1
      R2(2,2)=R2(2,2)*PJ1
      RRIG(1)=RRIG(1)*PJ1
      RRIG(2)=RRIG(2)*PJ1
      SCALE=DXY1
      IF(LSIDE.EQ.2.OR.LSIDE.EQ.4) SCALE=DXY2
      WS=WT*SCALE*GA
      CALL FUNCTS(X,Y,2)
      DO 435 M=1,NA
      R(1,M)=ANX*RX(1,M)+ANY*RX(3,M)
      R(2,M)=ANX*RX(3,M)+ANY*RX(2,M)
  435 CONTINUE
C  REPLACE THE ELEMENTS OF THE FORTH COLUMN OF R WITH THE PSUDO-VALUES
C  CORRRESPONDING TO RIGID BODY ROTATION AND SAVE THE TRUE VALUES OF THE
C  FORTH COLUMN OF R IN MATRIX RRIG IN ORDER TO CALCULATE THE TRUE
C  VALUES OF THE FORTH ROW OF MATRIX S2 IN MATRIX S2RIG
      IF(IRIG2.EQ.1) GO TO 436
      IF(NBT.LT.2) GO TO 436
      RR1=R(1,4)
      RR2=R(2,4)
      R(1,4)=RRIG(1)
      R(2,4)=RRIG(2)
      RRIG(1)=RR1
      RRIG(2)=RR2
  436 CONTINUE
```

```
      DO 440 J=1,NB
      DO 440 I=1,NA
      DO 440 K=1,2
  440 S2(I,J)=S2(I,J)+R(K,I)*U(K,J)*WS
      IF(IRIG2.EQ.1) GO TO 446
      IF(NBT.LT.2) GO TO 446
      DO 445 I=1,NA
      DO 445 J=1,2
      S2RIG(I)=S2RIG(I)+RRIG(J)*U(J,I)*WS
  445 CONTINUE
  446 CONTINUE
      DO 450 J=1,NR
      DO 450 I=1,NA
      DO 450 K=1,2
      P4(I,J)=P4(I,J)+R(K,I)*UR(K,J)*WS
  450 CONTINUE
      CALL INPOL(X,Y,LSIDE,LH)
      DO 460 J=1,NQ
      DO 460 I=1,NA
      DO 460 K=1,2
      S1S(I,J)=S1S(I,J)+R(K,I)*ALS(K,J)*WS
  460 CONTINUE
      DO 485 J=1,NQ
      DO 485 I=1,NR
      DO 485 K=1,2
  485 G2(I,J)=G2(I,J)+R2(K,I)*ALS(K,J)*WS
      DO 490 J=1,NB
      DO 490 I=1,NR
      DO 490 K=1,2
  490 P3T(I,J)= P3T(I,J)+R2(K,I)*U(K,J)*WS
      DO 495 J=1,NR
      DO 495 I=1,NR
      DO 495 K=1,2
  495 P2T(I,J)=P2T(I,J)+R2(K,I)*UR(K,J)*WS
  500 CONTINUE
  600 CONTINUE
  700 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE STIFK
C
      COMMON/MASS/AMRR(36,36),AMRDR(36,36),AM22(2,2),AMR2(36,2)
     1,AMRD2(36,2),AMRDD2(36,2)
      COMMON/KMAT/AK1S(20,20)
      COMMON/LON/S2(36,36),S4(36,36),S2D(36,36),S2DD(36,36)
      COMMON/SSS/S1S(36,20),S1SD(36,20),S1SDD(36,20)
      COMMON/AB/B(36,20),BD(36,20),BDD(36,20),
     1 A(2,20),AD(2,20),ADD(2,20)
      COMMON/DIM/NA,NAA,NBT,NB,NQ,NR,NINT,NINT2,IALL,NITER,SINCOD
      COMMON/RLIN/G2(2,20),P3T(2,36),P2T(2,2),P3TD(2,36),P3TDD(2,36)
     1 ,P4(36,2),P4D(36,2),P4DD(36,2)
      COMMON/VEL/CV,NUMCV,CVH(2,20)
      COMMON/WK/WKAREA(1404)
      COMMON/WK2/WKAR(720)
      DIMENSION P4A(36,20),BD1(36,20),BDD1(36,20),DDD1(36,20)
      DIMENSION DD1(36,20),DDD2(36,20)
      DIMENSION P1G1(36,20)
      DIMENSION AJP(36,20)
      DIMENSION D(2,20),DD(2,20),DDD(2,20)
      DIMENSION CDA(2,20),ADD1(2,20)
      DIMENSION CD1(36,2),CDD1(36,2),CDD2(36,2)
      DIMENSION P1P4(36,2)
      DIMENSION C1(2,36)
      DIMENSION C(2,2),CD(2,2),CDD(2,2),CI(2,2)
      EQUIVALENCE(P4A(1,1),BD1(1,1),BDD1(1,1),DDD1(1,1),AMRR(1,1))
      EQUIVALENCE(DD1(1,1),DDD2(1,1),AMRDR(1,1))
      EQUIVALENCE(P1G1(1,1),AJP(1,1),WKAR(1))
      DIMENSION BF(720)
      EQUIVALENCE(BF(1),BDD(1,1))
      EQUIVALENCE(D(1,1),DD(1,1),DDD(1,1),BF(1))
      EQUIVALENCE(CDA(1,1),ADD1(1,1),BF(41))
      EQUIVALENCE(CD1(1,1),BF(81))
      EQUIVALENCE(CDD1(1,1),BF(161))
      EQUIVALENCE(CDD2(1,1),BF(241))
      EQUIVALENCE(P1P4(1,1),BF(321))
      EQUIVALENCE(C1(1,1),BF(401))
      EQUIVALENCE(C(1,1),BF(481))
      EQUIVALENCE(CD(1,1),BF(491))
      EQUIVALENCE(CDD(1,1),BF(501))
      EQUIVALENCE(CI(1,1),BF(511))
      COMMON/S2RIG/S2RIG(36)
      COMMON/RIG/IRIG1(6),IRIG2
C
C         A
C
      DO 5 J=1,NA
      DO 5 I=1,NR
      C1(I,J)=0.
      DO 5 K=1,NB
    5 C1(I,J)=C1(I,J)+P3T(I,K)*S4(K,J)
      DO 10 J=1,NR
      DO 10 I=1,NR
      C(I,J)=0.
      DO 10 K=1,NA
   10 C(I,J)=C(I,J)+C1(I,K)*P4(K,J)
      DO 15 J=1,NR
      DO 15 I=1,NR
      C(I,J)=C(I,J)-P2T(I,J)
   15 CONTINUE
C
C     NORMALIZE C FOR NUMERICAL ACCURACY IN THE INVERSION
      CCC=C(1,1)
      IF(CCC.EQ.0.) GO TO 321
      DO 16 J=1,NR
```

```
      DO 16 I=1,NR
      C(I,J)=C(I,J)/CCC
 16   CONTINUE
      IDIGT=8
      IDIGT1=8
      CALL LINV2F(C,NR,NR,CI,IDIGT1,WKAREA,IER)
      IF(IER.EQ.129 .OR. IER.EQ.131) GO TO 820
      DO 17 J=1,NR
      DO 17 I=1,NR
      C(I,J)=C(I,J)*CCC
      CI(I,J)=CI(I,J)/CCC
 17   CONTINUE
C
      DO 20 J=1,NQ
      DO 20 I=1,NR
      D(I,J)=0.
      DO 20 K=1,NA
   20 D(I,J)=D(I,J)+C1(I,K)*S1S(K,J)
      DO 25 J=1,NQ
      DO 25 I=1,NR
   25 D(I,J)=D(I,J)-G2(I,J)
      DO 30 J=1,NQ
      DO 30 I=1,NR
      A(I,J)=0.
      DO 30 K=1,NR
   30 A(I,J)=A(I,J)+CI(I,K)*D(K,J)
C
C         B
C
      DO 35 J=1,NQ
      DO 35 I=1,NA
      P4A(I,J)=0.
      DO 35 K=1,NR
   35 P4A(I,J)=P4A(I,J)+P4(I,K)*A(K,J)
      DO 40 J=1,NQ
      DO 40 I=1,NB
      B(I,J)=0.
      DO 40 K=1,NA
   40 B(I,J)=B(I,J)+S4(I,K)*(S1S(K,J)-P4A(K,J))
C
C         AD
C
      IF(CV.EQ.0.) GO TO 218
C
      DO 45 J=1,NR
      DO 45 I=1,NA
      P1P4(I,J)=0.
      DO 45 K=1,NA
   45 P1P4(I,J)=P1P4(I,J)+S4(I,K)*P4(K,J)
      DO 50 J=1,NR
      DO 50 I=1,NA
      CD1(I,J)=0.
      DO 49 K=1,NA
   49 CD1(I,J)=CD1(I,J)+S2D(I,K)*P1P4(K,J)
   50 CD1(I,J)=P4D(I,J)-CD1(I,J)
      DO 55 J=1,NR
      DO 55 I=1,NR
      CD(I,J)=0.
      DO 55 K=1,NA
   55 CD(I,J)=CD(I,J)+C1(I,K)*CD1(K,J)
      DO 60 J=1,NR
      DO 60 I=1,NR
      DO 60 K=1,NB
   60 CD(I,J)=CD(I,J)+P3TD(I,K)*P1P4(K,J)
      DO 65 J=1,NQ
```

```
      DO 65 I=1,NA
      P1G1(I,J)=0.
      DO 65 K=1,NA
   65 P1G1(I,J)=P1G1(I,J)+S4(I,K)*S1S(K,J)
      DO 70 J=1,NQ
      DO 70 I=1,NA
      DD1(I,J)=0.
      DO 69 K=1,NA
   69 DD1(I,J)=DD1(I,J)+S2D(I,K)*P1G1(K,J)
   70 DD1(I,J)=-DD1(I,J)+S1SD(I,J)
      DO 75 J=1,NQ
      DO 75 I=1,NR
      DD(I,J)=0.
      DO 75 K=1,NA
   75 DD(I,J)=DD(I,J)+C1(I,K)*DD1(K,J)
      DO 80 J=1,NQ
      DO 80 I=1,NR
      DO 80 K=1,NB
   80 DD(I,J)=DD(I,J)+P3TD(I,K)*P1G1(K,J)
      DO 85 J=1,NQ
      DO 85 I=1,NR
      CDA(I,J)=0.
      DO 85 K=1,NR
   85 CDA(I,J)=CDA(I,J)+CD(I,K)*A(K,J)
      DO 90 J=1,NQ
      DO 90 I=1,NR
      AD(I,J)=0.
      DO 90 K=1,NR
   90 AD(I,J)=AD(I,J)+CI(I,K)*(DD(K,J)-CDA(K,J))
C
C          BD
C
      DO 95 J=1,NQ
      DO 95 I=1,NA
      BD1(I,J)=0.
      DO 95 K=1,NR
   95 BD1(I,J)=BD1(I,J)+P4D(I,K)*A(K,J)
      DO 100 J=1,NQ
      DO 100 I=1,NA
      DO 100 K=1,NR
  100 BD1(I,J)=BD1(I,J)+P4(I,K)*AD(K,J)
      DO 105 J=1,NQ
      DO 105 I=1,NA
      DO 105 K=1,NA
  105 BD1(I,J)=BD1(I,J)+S2D(I,K)*B(K,J)
      DO 110 J=1,NQ
      DO 110 I=1,NB
      BD(I,J)=0.
      DO 110 K=1,NA
  110 BD(I,J)=BD(I,J)+S4(I,K)*(S1SD(K,J)-BD1(K,J))
C
C          ADD
C
      DO 115 J=1,NR
      DO 115 I=1,NA
      CDD1(I,J)=0.
      DO 115 K=1,NA
  115 CDD1(I,J)=CDD1(I,J)+S4(I,K)*CD1(K,J)
      DO 120 J=1,NR
      DO 120 I=1,NA
      CDD2(I,J)=0.
      DO 120 K=1,NA
  120 CDD2(I,J)=CDD2(I,J)+2.*S2D(I,K)*CDD1(K,J)
      DO 125 J=1,NR
      DO 125 I=1,NA
```

```
      DO 124 K=1,NA
  124 CDD2(I,J)=CDD2(I,J)+S2DD(I,K)*P1P4(K,J)
  125 CDD2(I,J)=-CDD2(I,J)+P4DD(I,J)
      DO 130 J=1,NR
      DO 130 I=1,NR
      CDD(I,J)=0.
      DO 130 K=1,NA
  130 CDD(I,J)=CDD(I,J)+C1(I,K)*CDD2(K,J)
      DO 135 J=1,NR
      DO 135 I=1,NR
      DO 135 K=1,NB
  135 CDD(I,J)=CDD(I,J)+2.*P3TD(I,K)*CDD1(K,J)
      DO 140 J=1,NR
      DO 140 I=1,NR
      DO 140 K=1,NB
  140 CDD(I,J)=CDD(I,J)+P3TDD(I,K)*P1P4(K,J)
      DO 145 J=1,NQ
      DO 145 I=1,NA
      DDD1(I,J)=0.
      DO 145 K=1,NA
  145 DDD1(I,J)=DDD1(I,J)+S4(I,K)*DD1(K,J)
      DO 150 J=1,NQ
      DO 150 I=1,NA
      DDD2(I,J)=0.
      DO 150 K=1,NA
  150 DDD2(I,J)=DDD2(I,J)+2.*S2D(I,K)*DDD1(K,J)
      DO 155 J=1,NQ
      DO 155 I=1,NA
      DO 155 K=1,NA
  155 DDD2(I,J)=DDD2(I,J)+S2DD(I,K)*P1G1(K,J)
      DO 160 J=1,NQ
      DO 160 I=1,NR
      DDD(I,J)=0.
      DO 160 K=1,NA
  160 DDD(I,J)=DDD(I,J)+C1(I,K)*(S1SDD(K,J)-DDD2(K,J))
      DO 165 J=1,NQ
      DO 165 I=1,NR
      DO 165 K=1,NB
  165 DDD(I,J)=DDD(I,J)+2.*P3TD(I,K)*DDD1(K,J)
      DO 170 J=1,NQ
      DO 170 I=1,NR
      DO 170 K=1,NB
  170 DDD(I,J)=DDD(I,J)+P3TDD(I,K)*P1G1(K,J)
      DO 175 J=1,NQ
      DO 175 I=1,NR
      ADD1(I,J)=0.
      DO 175 K=1,NR
  175 ADD1(I,J)=ADD1(I,J)+CDD(I,K)*A(K,J)
      DO 180 J=1,NQ
      DO 180 I=1,NR
      DO 180 K=1,NR
  180 ADD1(I,J)=ADD1(I,J)+2.*CD(I,K)*AD(K,J)
      DO 185 J=1,NQ
      DO 185 I=1,NR
      ADD(I,J)=0.
      DO 185 K=1,NR
  185 ADD(I,J)=ADD(I,J)+CI(I,K)*(DDD(K,J)-ADD1(K,J))
C
C         BDD
C
      DO 190 J=1,NQ
      DO 190 I=1,NA
      BDD1(I,J)=0.
      DO 190 K=1,NA
  190 BDD1(I,J)=BDD1(I,J)+S2DD(I,K)*B(K,J)
```

```
      DO 195 J=1,NQ
      DO 195 I=1,NA
      DO 195 K=1,NA
  195 BDD1(I,J)=BDD1(I,J)+2.*S2D(I,K)*BD(K,J)
      DO 200 J=1,NQ
      DO 200 I=1,NA
      DO 200 K=1,NR
  200 BDD1(I,J)=BDD1(I,J)+P4(I,K)*ADD(K,J)
      DO 205 J=1,NQ
      DO 205 I=1,NA
      DO 205 K=1,NR
  205 BDD1(I,J)=BDD1(I,J)+2.*P4D(I,K)*AD(K,J)
      DO 210 J=1,NQ
      DO 210 I=1,NA
      DO 210 K=1,NR
  210 BDD1(I,J)=BDD1(I,J)+P4DD(I,K)*A(K,J)
      DO 215 J=1,NQ
      DO 215 I=1,NA
      BDD(I,J)=0.
      DO 215 K=1,NA
  215 BDD(I,J)=BDD(I,J)+S4(I,K)*(S1SDD(K,J)-BDD1(K,J))
C
C         B*H*B
C
C REPLACE THE FORTH ROW OF MATRIX S2 WITH ITS TRUE VALES,REMEMBER THE
C FORTH ROW OF S2 RIGTH NOW IS FOR BOUNDARY TRACTIONS FOR RIGID BODY
C ROTATION TERMS IN ORDER TO MAKE MATIX "P" NONSINGULAR
C  SEE SUBROUTINE LINEI
  218 CONTINUE
      IF(IRIG2.EQ.1) GO TO 2191
      IF(NBT.LT.2) GO TO 2191
      DO 219 I=1,NA
      S2(4,I)=S2RIG(I)
  219 CONTINUE
 2191 CONTINUE
      DO 220 J=1,NQ
      DO 220 I=1,NB
      AJP(I,J)=0.
      DO 220 K=1,NB
  220 AJP(I,J)=AJP(I,J)+S2(K,I)*B(K,J)
      DO 225 J=1,NQ
      DO 225 I=1,NQ
      AK1S(I,J)=0.
      DO 225 K=1,NB
  225 AK1S(I,J)=AK1S(I,J)+B(K,I)*AJP(K,J)
      RETURN
  320 WRITE(6,402) IER,IDIGT,IDIGT1
      STOP 7
  321 WRITE(6,403)
      STOP 10
  402 FORMAT(/,20X,28H*****INVERS FAILED ON C*****,/20X,9HERROR NO.,I4,
     1/2X,39HPROPOSED DECIMAL DIGITS OF ACCURACY WAS,I4,2X,51HBUT THE UW
     2-IMSL LIBRARY ROUTINE "LINV2F" FOUND ONLY,I4,2X,26HDECIMAL DIGITS
     3OF ACCURACY,
     4/2X,47HERROR NO. 34=DECIMAL DIGIT ACCURACY TEST FAILED,
     5/2X,48HERROR NO. 129=MATRIX IS ALGORITHMICALLY SINGULAR,
     6/2X,85HERROR NO. 131=MATRIX IS TOO ILL-CONDITONED FOR ITERATIVE IM
     7PROVEMENTS TO BE EFFECTIVE,
     8//,10X,60H*****TRY CHANGING MATERIAL PROPERIES OR CRACK-TIP SPEED*
     9****)
  403 FORMAT(/,20X,28H*****INVERS FAILED ON C*****,12H   C(1,1)=0.)
C
      END
```

```
      SUBROUTINE MARBAI
C
      COMMON/RHHO/RHO(6),RODUM
      COMMON/TIP/NCR1,NCR2,NCR3,NCR4,NELX,CTPX,CTPY,SIF1,SIF2
      COMMON/BK10/R(300),Z(300),CODE(300),IX(6,250)
      COMMON/MASS/AMRR(36,36),AMRDR(36,36),AM22(2,2),AMR2(36,2)
     1 ,AMRD2(36,2),AMRDD2(36,2)
      COMMON/DISP/U(2,36),UD(2,36),UDD(2,36),UR(2,2)
      COMMON/DIF/DXX2,DXY3,DXXY,DYX2,DYY2,DYXY,DXX,DXY,DYX,DYY,DEM
      COMMON/INTGR/PT(10),WG(10),PT2(2),WG2(2),PEI
      COMMON/DIM/NA,NAA,NBT,NB,NQ,NR,NINT,NINT2,IALL,NITER,SINCOD
      COMMON/VEL/CV,NUMCV,CVH(2,20)
      COMMON/STRESS/RX(3,36),RXD(3,36),RXDD(3,36)
      COMMON/SUMAN/A1(3,3,6),A1I(3,3,6),ADET(6),ASIZE(6),CL
C
      IF(RODUM.EQ.0.)RETURN
      DO 5 J=1,NR
      DO 5 I=1,NR
    5 AM22(I,J)=0.
      DO 15 J=1,NR
      DO 15 I=1,NB
   15 AMR2(I,J)=0.
      DO 30 J=1,NB
      DO 30 I=1,NB
      AMRDR(I,J)=0.
      AMRR(I,J)=0.
   30 CONTINUE
      G1=.5
      G2=.5
      G=G1*G2
      DO 90 LH=1,2
      DO 85 LH1=1,2
      DO 80 L=1,NINT
      X=.5*(PT(L)-1.)
      IF(LH1.EQ.2) X=.5*(PT(L)+1.)
      DO 70 M=1,NINT
      Y=.5*(PT(M)-1.)
      IF(LH.EQ.2) Y=.5*(PT(M)+1.)
      RO=RHO(IX(6,NCR1))
      IF(Y.LT.0.) RO=RHO(IX(6,NCR2))
      WT=WG(L)*WG(M)
      CALL TRANS(X,Y,N,2,1)
      WA=WT*DEM*G*RO
      CALL FUNCTS(X,Y,4)
      DO 50 J=1,2
      DO 50 I=1,NB
      DO 50 K=1,2
      AMR2(I,J)=AMR2(I,J)+U(K,I)*UR(K,J)*WA
   50 CONTINUE
      IF(CV .EQ.0.)GO TO 51
      GO TO 55
   51 CONTINUE
      DO 52 J=1,NB
      DO 52 I=1,NB
      DO 52 K=1,2
      AMRR(I,J)=AMRR(I,J)+U(K,I)*U(K,J)*WA
   52 CONTINUE
      GO TO 62
   55 CONTINUE
      DO 60 J=1,NB
      DO 60 I=1,NB
      DO 60 K=1,2
      AMRR(I,J)=AMRR(I,J)+U(K,I)*U(K,J)*WA
      AMRDR(I,J)=AMRDR(I,J)+UD(K,I)*U(K,J)*WA
   60 CONTINUE
```

```
   62 CONTINUE
      DO 65 J=1,NR
      DO 65 I=1,NR
      DO 65 K=1,2
   65 AM22(I,J)=AM22(I,J)+UR(K,I)*UR(K,J)*WA
   70 CONTINUE
   80 CONTINUE
   85 CONTINUE
   90 CONTINUE
      RETURN
C
      END
```

```
      SUBROUTINE MASSM
C
      COMMON/MASS/AMRR(36,36),AMRDR(36,36),AM22(2,2),AMR2(36,2)
     1 ,AMRD2(36,2),AMRDD2(36,2)
      COMMON/VEL/CV,NUMCV,CVH(2,20)
      COMMON/RHHO/RHO(6),RODUM
      COMMON/BK10/R(300),Z(300),CODE(300),IX(6,250)
      COMMON/SUMAN/A1(3,3,6),A1I(3,3,6),ADET(6),ASIZE(6),CL
      COMMON/LON/S2(36,36),S4(36,36),S2D(36,36),S2DD(36,36)
      COMMON/SSS/S1S(36,20),S1SD(36,20),S1SDD(36,20)
      COMMON/MMAT/AM1(20,20),V1(20,20),AK4(20,20)
      COMMON/DIM/NA,NAA,NBT,NB,NQ,NR,NINT,NINT2,IALL,NITER,SINCOD
      COMMON/TIP/NCR1,NCR2,NCR3,NCR4,NELX,CTPX,CTPY,SIF1,SIF2
      COMMON/AB/B(36,20),BD(36,20),BDD(36,20),
     1 A(2,20),AD(2,20),ADD(2,20)
      DIMENSION AM11(36,20)
      DIMENSION AMD(36,20)
      DIMENSION V11(20,36),V12(20,2),V13(20,36),V14(20,2),V15(20,36)
      DIMENSION V16(20,2)
      DIMENSION AK41(20,36),AK42(20,2),AK43(20,36),AK44(20,2)
      DIMENSION AK45(20,36),AK46(20,2),AK47(20,2)
      DIMENSION AME(20,20)
      DIMENSION AM1R(20,20)
      DIMENSION AMC(2,20)
      EQUIVALENCE(AK41(1,1),V11(1,1),AM11(1,1),S1S(1,1))
      EQUIVALENCE(AK43(1,1),V13(1,1),AMD(1,1),S1SD(1,1))
      EQUIVALENCE(AK45(1,1),V15(1,1),S1SDD(1,1))
      EQUIVALENCE(AK42(1,1),V12(1,1))
      EQUIVALENCE(AK44(1,1),V14(1,1),S2D(1,1))
      EQUIVALENCE(AK46(1,1),V16(1,1),S2DD(1,1))
      EQUIVALENCE(AME(1,1),S4(1,1))
      EQUIVALENCE(AM1R(1,1),S4(1,16))
      EQUIVALENCE(AK47(1,1),S4(1,31))

C
C
C          M1=B*MRR*B
C
      IF(RODUM.EQ. 0.) RETURN
      DO 10 J=1,NQ
      DO 10 I=1,NB
      AM11(I,J)=0.
      DO 10 K=1,NB
   10 AM11(I,J)=AM11(I,J)+AMRR(I,K)*  B(K,J)
      DO 20 J=1,NQ
      DO 20 I=1,NQ
      AM1(I,J)=0.
      DO 20 K=1,NB
   20 AM1(I,J)=AM1(I,J)+  B(K,I)*AM11(K,J)
C
C          RIGID BODY TERMS FOR M1
C
C          M1R=A*M22*A + B*MR2*A + A*M2R*B
C
      DO 30  J=1,NQ
      DO 30  I=1,NR
      AMC(I,J)=0.
      DO 30  K=1,NR
   30 AMC(I,J)=AMC(I,J)+AM22(I,K)*A(K,J)
      DO 40  J=1,NQ
      DO 40  I=1,NQ
      AM1R(I,J)=0.
      DO 40  K=1,NR
   40 AM1R(I,J)=AM1R(I,J)+A(K,I)*AMC(K,J)
      DO 50  J=1,NQ
```

```
      DO 50  I=1,NB
      AMD(I,J)=0.
      DO 50  K=1,NR
   50 AMD(I,J)=AMD(I,J)+AMR2(I,K)*A(K,J)
      DO 60  J=1,NQ
      DO 60  I=1,NQ
      AME(I,J)=0.
      DO 60  K=1,NB
   60 AME(I,J)=AME(I,J)+  B(K,I)*AMD(K,J)
      DO 70  J=1,NQ
      DO 70  I=1,NQ
   70 AM1R(I,J)=AM1R(I,J)+AME(I,J)+AME(J,I)
      DO 80  J=1,NQ
      DO 80  I=1,NQ
   80 AM1(I,J)=AM1(I,J)+AM1R(I,J)
      IF(CV.EQ.0.0) RETURN
C
C          DAMPING AND CONVECTION TERMS
C
C          K4
C
      DO 110 J=1,NB
      DO 110 I=1,NQ
      AK41(I,J)=0.
      AK45(I,J)=0.
      DO 110 K=1,NB
      AK41(I,J)=AK41(I,J)+BDD(K,I)*AMRR(K,J)
      AK45(I,J)=AK45(I,J)+BD(K,I)*AMRDR(K,J)
  110 CONTINUE
      DO 120 J=1,NR
      DO 120 I=1,NQ
      AK42(I,J)=0.
      DO 120 K=1,NR
      AK42(I,J)=AK42(I,J)+ADD(K,I)*AM22(K,J)
  120 CONTINUE
      DO 130 J=1,NB
      DO 130 I=1,NQ
      AK43(I,J)=0.
      DO 130 K=1,NR
      AK43(I,J)=AK43(I,J)+ADD(K,I)*AMR2(J,K)
  130 CONTINUE
      DO 140 J=1,NR
      DO 140 I=1,NQ
      AK44(I,J)=0.
      AK46(I,J)=0.
      AK47(I,J)=0.
      DO 140 K=1,NB
      AK44(I,J)=AK44(I,J)+BDD(K,I)*AMR2(K,J)
      AK46(I,J)=AK46(I,J)+BD(K,I)*AMRD2(K,J)
      AK47(I,J)=AK47(I,J)+   B(K,I)*AMRDD2(K,J)
  140 CONTINUE
      DO 150 J=1,NQ
      DO 150 I=1,NQ
      AK4(I,J)=0.
      DO 150 K=1,NB
      AK4(I,J)=AK4(I,J)+(2.*AK45(J,K)+AK41(J,K)+AK43(J,K))*   B(K,I)
  150 CONTINUE
      DO 160 J=1,NQ
      DO 160 I=1,NQ
      DO 160 K=1,NR
      AK4(I,J)=AK4(I,J)+(AK42(J,K)+AK44(J,K)+2.*AK46(J,K)+AK47(J,K))
     1 *A(K,I)
  160 CONTINUE
C
C          V1
```

```
C
      DO 170 J=1,NB
      DO 170 I=1,NQ
      V11(I,J)=0.
      V15(I,J)=0.
      DO 170 K=1,NB
      V15(I,J)=V15(I,J)+   B(K,I)*AMRDR(K,J)
  170 V11(I,J)=V11(I,J)+BD(K,I)*AMRR(K,J)
      DO 180 J=1,NR
      DO 180 I=1,NQ
      V12(I,J)=0.
      DO 180 K=1,NR
  180 V12(I,J)=V12(I,J)+AD(K,I)*AM22(K,J)
      DO 190 J=1,NB
      DO 190 I=1,NQ
      V13(I,J)=0.
      DO 190 K=1,NR
  190 V13(I,J)=V13(I,J)+AD(K,I)*AMR2(J,K)
      DO 200 J=1,NR
      DO 200 I=1,NQ
      V14(I,J)=0.
      V16(I,J)=0.
      DO 200 K=1,NB
      V16(I,J)=V16(I,J)+   B(K,I)*AMRD2(K,J)
  200 V14(I,J)=V14(I,J)+BD(K,I)*AMR2(K,J)
      DO 210 I=1,NQ
      DO 210 J=1,NQ
      V1(I,J)=0.
      DO 210 K=1,NB
      V1(I,J)=V1(I,J)+2.*(V11(J,K)+V13(J,K)+V15(J,K))*B(K,I)
  210 CONTINUE
      DO 220 I=1,NQ
      DO 220 J=1,NQ
      DO 220 K=1,NR
      V1(I,J)=V1(I,J)+2.*(V12(J,K)+V14(J,K)+V16(J,K))  *A(K,I)
  220 CONTINUE
      RETURN
C
      END
```

```
      SUBROUTINE INPOL(X,Y,LSIDE,LH)
C
      COMMON/DIM/NA,NAA,NBT,NB,NQ,NR,NINT,NINT2,IALL,NITER,SINCOD
      COMMON/INTPOL/ALS(2,20)
C
      DO 10 J=1,NQ
      DO 10 I=1,2
   10 ALS(I,J)=0.
      IF(LSIDE.EQ.1) GO TO 20
      IF(LSIDE.EQ.2) GO TO 40
      IF(LSIDE.EQ.3) GO TO 60
      IF(LSIDE.EQ.4) GO TO 80
   20 CONTINUE
      IF(LH.EQ.2) GO TO 30
      ALS(1,3)=-X
      ALS(1,5)=1.+X
      ALS(2,4)=ALS(1,3)
      ALS(2,6)=ALS(1,5)
      GO TO 100
   30 ALS(1,5)=1.-X
      ALS(1,7)=X
      ALS(2,6)=ALS(1,5)
      ALS(2,8)=ALS(1,7)
      GO TO 100
   40 CONTINUE
      IF(LH.EQ.2) GO TO 50
      ALS(1,7)=-Y
      ALS(1,9)=1.+Y
      ALS(2,8)=ALS(1,7)
      ALS(2,10)=ALS(1,9)
      GO TO 100
   50 ALS(1,11)=1.-Y
      ALS(1,13)=Y
      ALS(2,12)=ALS(1,11)
      ALS(2,14)=ALS(1,13)
      GO TO 100
   60 CONTINUE
      IF(LH.EQ.2) GO TO 70
      ALS(1,13)=X
      ALS(1,15)=1.-X
      ALS(2,14)=ALS(1,13)
      ALS(2,16)=ALS(1,15)
      GO TO 100
   70 ALS(1,15)=1.+X
      ALS(1,17)=-X
      ALS(2,16)=ALS(1,15)
      ALS(2,18)=ALS(1,17)
      GO TO 100
   80 CONTINUE
      IF(LH.EQ.2) GO TO 90
      ALS(1,17)=Y
      ALS(1,19)=1.-Y
      ALS(2,18)=ALS(1,17)
      ALS(2,20)=ALS(1,19)
      GO TO 100
   90 ALS(1,1)=1.+Y
      ALS(1,3)=-Y
      ALS(2,2)=ALS(1,1)
      ALS(2,4)=ALS(1,3)
  100 RETURN
C
      END
```

```
      SUBROUTINE TRANS(X,Y,N,JC,JR)
C
      COMMON/BK10/ R(300),Z(300),CODE(300),IX(6,250)
      COMMON/DIF/DXX2,DXY3,DXXY,DYX2,DYY2,DYXY,DXX,DXY,DYX,DYY,DEM
      COMMON/IPAR/PJ1,PJ2,PJ3,PJ4,DXY1,DXY2
      COMMON/PHYS/ANX,ANY,A(2)
      COMMON/MAIN/CORD(10,2)
      COMMON/DIM/NA,NAA,NBT,NB,NQ,NR,NINT,NINT2,IALL,NITER,SINCOD
      DIMENSION D(2),G(2),P(2)
      REAL KORD(10,2)
C
      IF(JR.EQ.2) GO TO 20
      L1=2
      L2=4
      L3=7
      L4=9
      NQ1=NQ/2
      DO 10 J=1,2
      DO 10 I=1,NQ1
   10 KORD(I,J)=CORD(I,J)
      GO TO 40
   20 CONTINUE
      L1=1
      L2=2
      L3=3
      L4=4
      DO 30 I=1,4
      K=IX(I,N)
      KORD(I,1)=R(K)
   30 KORD(I,2)=Z(K)
   40 CONTINUE
      DO 50 K=1,2
      A(K)=(1.-X)*(1.-Y)*.25*KORD(L1,K)
      A(K)=A(K)+(1.+X)*(1.-Y)*.25*KORD(L2,K)
      A(K)=A(K)+(1.+X)*(1.+Y)*.25*KORD(L3,K)
      A(K)=A(K)+(1.-X)*(1.+Y)*.25*KORD(L4,K)
   50 CONTINUE
      IF(JC.EQ.1) GO TO 300
      T1   =1.-Y
      T3   =1.+Y
      DO 101 K=1,2
      G(K)=0.
      P(K)=  (KORD(L1,K)-KORD(L2,K)+KORD(L3,K)-KORD(L4,K))/4.
  101 D(K)=(-T1*KORD(L1,K)+T1*KORD(L2,K)+T3*KORD(L3,K)-T3*KORD(L4,K))/4.
      DXX2=G(1)
      DYX2=G(2)
      DXXY=P(1)
      DYXY=P(2)
      DXX =D(1)
      DYX =D(2)
      T1   = 1.-X
      T3   = 1.+X
      DO   202  K=1,2
      G(K)=0.
  202 D(K)=(-T1*KORD(L1,K)-T3*KORD(L2,K)+T3*KORD(L3,K)+T1*KORD(L4,K))/4.
      DXY3=G(1)
      DYY2=G(2)
      DXY = D(1)
      DYY = D(2)
      DEM = DXX*DYY-DYX*DXY
      PJ1 =DYY/DEM
      PJ2 =-DYX/DEM
      PJ3 =-DXY/DEM
      PJ4 = DXX/DEM
      DXY1 = SQRT(DXX*DXX+DYX*DYX)
```

```
      DXY2 = SQRT(DXY*DXY+DYY*DYY)
  300 CONTINUE
      RETURN
C
      END
```

```
      SUBROUTINE NORMAL(L)
C
      COMMON/MAIN/CORD(10,2)
      COMMON/PHYS/ANX,ANY,A(2)
C
      IF(L-1) 20,10,20
   10 CONTINUE
      N1=2
      N2=4
      GO TO 80
   20 IF(L-2) 40,30,40
   30 CONTINUE
      N1=4
      N2=7
      GO TO 80
   40 IF(L-3) 60,50,60
   50 CONTINUE
      N1=7
      N2=9
      GO TO 80
   60 CONTINUE
      N1=9
      N2=2
   80 CONTINUE
      X1=CORD(N1,1)
      X2=CORD(N2,1)
      Y1=CORD(N1,2)
      Y2=CORD(N2,2)
      B=X2-X1
      C=Y2-Y1
      AR=SQRT(B**2+C**2)
      ANX=(Y2-Y1)/AR
      ANY=(X1-X2)/AR
      RETURN
C
      END
```

```
      SUBROUTINE PAIR(L,NTER)
      COMMON/STORE/SK(192,96)
      COMMON/BK11/DELT,DT1,DT3,BETA,BET1,BET2,BET3,BET4,BET5,NBAND,NBD2
      COMMON/BK10/R(300),Z(300),CODE(300),IX(6,250)
      COMMON/SHIFT/NSBL,NDEL,LASTB,NF,NBRED,NELTYP,ISK,RCODE
      COMMON/PAIR1/NPAIR,LPAIR(3,40)
      COMMON/STRESS/RX(3,36),RXD(3,36),RXDD(3,36)
      DIMENSION B(192)
      EQUIVALENCE(B(1),RX(1,1))
C
      COMMON/SSS/S1S(36,20),S1SD(36,20),S1SDD(36,20)
      COMMON/MASS/AMRR(36,36),AMRDR(36,36),AM22(2,2),AMR2(36,2)
     1,AMRD2(36,2),AMRDD2(36,2)
      COMMON/WK/WKAREA(1404)
      COMMON/LON/S2(36,36),S4(36,36),S2D(36,36),S2DD(36,36)
      DIMENSION E(800),ELOAD(800),EM(800),EMAT(800)
      DIMENSION US(600),VS(600),AS(600)
      DIMENSION EX1(2160),EX3(1404),EX5(5184)
      EQUIVALENCE(E(1),EX1(1))
      EQUIVALENCE(ELOAD(1),EX1(1001))
      EQUIVALENCE(EMAT(1),EX3(1))
      EQUIVALENCE(US(1),EX5(1))
      EQUIVALENCE(VS(1),EX5(1001))
      EQUIVALENCE(AS(1),EX5(2001))
      EQUIVALENCE(EM(1),EX5(3001))
      EQUIVALENCE(S1S(1,1),EX1(1))
      EQUIVALENCE(WKAREA(1),EX3(1))
      EQUIVALENCE(S2(1,1),EX5(1))
C
      NBG=NBAND
      IF(NTER.EQ.1) GO TO 500
      DO 1001 J=1,NBG
      DO 1001 I=1,NBAND
      SK(I,J)=0.
 1001 CONTINUE
      DO 1002 I=1,NBAND
      B(I)=0.
 1002 CONTINUE
      IF(L.EQ.0)REWIND 11
      IF(L.EQ.1)REWIND 14
      IF(L.EQ.0)REWIND 21
      IF(L.EQ.1)REWIND 24
      NB1=1
      NB2=NBAND
      NB3=NBAND+NBAND
      NBLOC=0
    2 CONTINUE
      NBLOC=NBLOC+1
      II=(NBLOC-2)*NBAND
      IF(NBLOC-LASTB-1) 5,110,5
    5 NBA=NBRED
      IF(NBLOC.EQ.NSBL) NBA=NBAND
      K=1+NBAND
      IF(L.EQ.0) READ(11) ((SK(I,J),I=K,NBD2),J=1,NBG)
      IF(L.EQ.1) READ(14) ((SK(I,J),I=K,NBD2),J=1,NBG)
      DO 6 I=K,NBD2
      B(I)=E(II+I)
    6 CONTINUE
      DO 100 N=1,NPAIR
      IF(LPAIR(3,N).LE.0)GO TO 100
      N1=LPAIR(1,N)+NDEL-II
      N2=LPAIR(2,N)+NDEL-II
      IF(N1.LT.NB1 .OR. N1.GT.NB3) GO TO 100
      IF(N2.LT.NB1 .OR. N2.GT.NB3) GO TO 100
      IF(N1.LE.NB2 .AND. N2.LE.NB2) GO TO 100
```

```
      ND=N2-N1
      IF(ND.LT.0)GO TO 40
      ND2=2+ND
      N2P=N2+1
      DO 20 K=1,NBA
      K2=N2P-K
      KD=ND2-K
      IF(KD.LE.0) GO TO 10
      SK(K2,K)=SK(K2,K)+SK(N1,KD)
      GO TO 20
   10 KD=K-ND
      SK(K2,K)=SK(K2,K)+SK(K2,KD)
   20 CONTINUE
      NBS=NBA-ND
      DO 30 K=1,NBS
      KD=K+ND
      SK(N2,K)=SK(N2,K)+SK(N1,KD)
   30 CONTINUE
      GO TO 80
   40 ND2=2-ND
      N2P=N2+1
      N2M=N2-1
      DO 60 K=1,NBA
      KD=ND2-K
      IF(KD.LE.0) GO TO 50
      K2=N2M+K
      SK(N2,K)=SK(N2,K)+SK(K2,KD)
      GO TO 60
   50 KD=K+ND
      SK(N2,K)=SK(N2,K)+SK(N1,KD)
   60 CONTINUE
      NBS=NBA+ND
      DO 70 K=1,NBS
      K2=N2P-K
      KD=K-ND
      SK(K2,K)=SK(K2,K)+SK(K2,KD)
   70 CONTINUE
   80 B(N2)=B(N2)+B(N1)
  100 CONTINUE
      IF(NBLOC .EQ. 1) GO TO 200
  110 CONTINUE
      IF(L.EQ.0) WRITE(21) ((SK(I,J),I=1,NBAND),J=1,NBG)
      IF(L.EQ.1) WRITE(24) ((SK(I,J),I=1,NBAND),J=1,NBG)
      DO 120 I=1,NBAND
      E(I+II)=B(I)
  120 CONTINUE
      IF(NBLOC.EQ.LASTB+1) RETURN
  200 DO 220 I=1,NBAND
      K=I+NBAND
      DO 210 J=1,NBAND
      SK(I,J)=SK(K,J)
  210 CONTINUE
      B(I)=B(K)
  220 CONTINUE
      GO TO 2
C
  500 CONTINUE
      DO 4002 I=1,NBAND
      B(I)=0.
 4002 CONTINUE
      NB1=1
      NB2=NBAND
      NB3=NBAND+NBAND
      NBLOC=0
  502 CONTINUE
```

```
      NBLOC=NBLOC+1
      II=(NBLOC-2)*NBAND
      IF(NBLOC-LASTB-1) 505,610,505
  505 K=1+NBAND
      DO 506 I=K,NBD2
      B(I)=B(II+I)
  506 CONTINUE
      DO 600 N=1,NPAIR
      IF(LPAIR(3,N).LE.0)GO TO 600
      N1=LPAIR(1,N)+NDEL-II
      N2=LPAIR(2,N)+NDEL-II
      IF(N1.LT.NB1 .OR. N1.GT.NB3) GO TO 600
      IF(N2.LT.NB1 .OR. N2.GT.NB3) GO TO 600
      IF(N1.LE.NB2 .AND. N2.LE.NB2) GO TO 600
      B(N2)=B(N2)+B(N1)
  600 CONTINUE
      IF(NBLOC .EQ. 1) GO TO 700
  610 CONTINUE
      DO 620 I=1,NBAND
      B(I+II)=B(I)
  620 CONTINUE
      IF(NBLOC.EQ.LASTB+1) RETURN
  700 DO 720 I=1,NBAND
      B(I)=B(I+NBAND)
  720 CONTINUE
      GO TO 502
C
      END
```

```
      SUBROUTINE FUNCTS(X1,Y1,L)
C
      COMMON/TIP/NCR1,NCR2,NCR3,NCR4,NELX,CTPX,CTPY,SIF1,SIF2
      COMMON/MAIN/CORD(16,2)
      COMMON/VEL/CV,NUMCV,CVH(2,20)
      COMMON/DIM/NA,NAA,NBT,NB,NQ,NR,NINT,NINT2,IALL,NITER,SINCOD
      COMMON/PHYS/ANX,ANY,A(2)
      COMMON/INTGR/PT(10),WG(10),PT2(2),WG2(2),PEI
      COMMON/IPAR/PJ1,PJ2,PJ3,PJ4,DXY1,DXY2
      COMMON/RIG/IRIG1(6),IRIG2
      COMMON/DISP/U(2,36),UD(2,36),UDD(2,36),UR(2,2)
      COMMON/STRESS/S(3,36),SD(3,36),SDD(3,36)
      COMPLEX UU(2,20),UUD(2,20),UUDD(2,20)
      COMPLEX SS(3,20),SSD(3,20),SSDD(3,20)
      COMMON/WK2/WKAR(720)
      EQUIVALENCE (WKAR(1),UU(1,1))
      EQUIVALENCE (WKAR(81),UUD(1,1))
      EQUIVALENCE (WKAR(161),UUDD(1,1))
      EQUIVALENCE (WKAR(241),SS(1,1))
      EQUIVALENCE (WKAR(361),SSD(1,1))
      EQUIVALENCE (WKAR(481),SSDD(1,1))
C
      COMMON /C7/ICASE(2)
      COMMON /C10/LAM(20)
      COMMON /C12/F(8)
      COMMON /C13/SI(12),SIX(12),SIXX(12)
     1           ,DIS(8),DISX(8),DISXX(8)
      COMMON /C14/MUU(2,2)
      COMPLEX MUU,LAM,SI,SIX,SIXX,DIS,DISX,DISXX,F
      COMPLEX LM,LJ,Z,ZB,ZL4,ZBZL4,ZBL4,ZZBL4,ZL3,ZBZL3,ZBL3,ZZBL3
     1        ,ZL2,ZBZL2,ZBL2,ZZBL2,ZL1,ZBZL1,ZBL1,ZZBL1,ZL0,ZBL0
C
C        THIS SUBROUTINE CONTAINS THE MODE I AND II FUNCTIONS FOR
C        STATIONARY AND MOVIG CRACKS
C        SQUARE ELEMENT ONLY

      UR(1,1)=1.
      UR(2,1)=0.
      UR(1,2)=0.
      UR(2,2)=1.
      CX=(2.*CTPX-CORD(1,1)-CORD(5,1))/(CORD(5,1)-CORD(1,1))
      X=X1-CX
      Y=Y1
      CV1=CV
      CV2=CV*CV
      P1=PJ1
      P2=PJ1*PJ1
      P3=P2*PJ1
      P1CV1=P1*CV1
      P2CV2=P2*CV2
      P2CV1=P2*CV1
      P3CV2=P3*CV2
      II=1
      IF(Y .LT. 0.) II=2
      IF(ICASE(II) .EQ. 2) GO TO 1000
      IF(L.EQ.8) GO TO 600
      IF(CV.EQ.0. .AND. L.GE.4) GO TO 200
      IF(CV.EQ. 0.) GO TO 500
      IF(L.EQ.2) GO TO 1
      IF(L.EQ.4) GO TO 100
      IF(L.EQ.6) GO TO 300
    1 DO 10  I=1,3
      DO 10  J=1,NBT
      SS(I,J)=(0.,0.)
      SSD(I,J)=(0.,0.)
```

```
      SSDD(I,J)=(0.,0.)
 10   CONTINUE
      DO 20 I=1,2
      DO 20 J=1,NBT
      UU(I,J)=(0.,0.)
      UUD(I,J)=(0.,0.)
      UUDD(I,J)=(0.,0.)
 20   CONTINUE
      DO 70 N=1,NBT
      CALL COEFF(N,II)
      DO 60 M=1,2
      M1=M-1
      Z=X+MUU(M,II)*Y
      ZB=CONJG(Z)
      LM=LAM(N)
      LJ=LM-3.
      ZL3=LJ*CLOG(Z)
      ZL3=CEXP(ZL3)
      ZBL3=CONJG(LJ)*CLOG(Z)
      ZBL3=CONJG(CEXP(ZBL3))
      ZL2=ZL3*Z
      ZBL2=ZBL3*ZB
      ZL1=ZL2*Z
      ZBL1=ZBL2*ZB
      ZL0=ZL1*Z
      ZBL0=ZBL1*ZB
      DO 30 I=1,3
      J=(I-1)*4
      J=J+M1*2
      J1=J+1
      J2=J1+1
      SS(I,N)=SS(I,N)+(SI(J1)*ZL1+SI(J2)*ZBL1)*P1
      SSD(I,N)=SSD(I,N)-(SIX(J1)*ZL2+SIX(J2)*ZBL2)*P2CV1
      SSDD(I,N)=SSDD(I,N)+(SIXX(J1)*ZL3+SIXX(J2)*ZBL3)*P3CV2
 30   CONTINUE
      DO 40 I=1,2
      J=(I-1)*4
      J=J+M1*2
      J1=J+1
      J2=J1+1
      UU(I,N)=UU(I,N)+DIS(J1)*ZL0+DIS(J2)*ZBL0
      UUD(I,N)=UUD(I,N)-(DISX(J1)*ZL1+DISX(J2)*ZBL1)*P1CV1
      UUDD(I,N)=UUDD(I,N)+(DISXX(J1)*ZL2+DISXX(J2)*ZBL2)*P2CV2
 40   CONTINUE
 60   CONTINUE
 70   CONTINUE
      DO 75 I=1,3
      DO 75 J=1,NBT
      JJ=J*2
      JJ1=JJ-1
      S(I,JJ1)=REAL(SS(I,J))
      S(I,JJ)=-AIMAG(SS(I,J))
      SD(I,JJ1)=REAL(SSD(I,J))
      SD(I,JJ)=-AIMAG(SSD(I,J))
      SDD(I,JJ1)=REAL(SSDD(I,J))
      SDD(I,JJ)=-AIMAG(SSDD(I,J))
 75   CONTINUE
      DO 80 I=1,2
      DO 80 J=1,NBT
      JJ=J*2
      JJ1=JJ-1
      U(I,JJ1)=REAL(UU(I,J))
      U(I,JJ)=-AIMAG(UU(I,J))
      UD(I,JJ1)=REAL(UUD(I,J))
      UD(I,JJ)=-AIMAG(UUD(I,J))
```

```
      UDD(I,JJ1)=REAL(UUDD(I,J))
      UDD(I,JJ)=-AIMAG(UUDD(I,J))
   80 CONTINUE
      RETURN
  100 CONTINUE
      DO 120 I=1,2
      DO 120 J=1,NBT
      UU(I,J)=(0.,0.)
      UUD(I,J)=(0.,0.)
  120 CONTINUE
      DO 170 N=1,NBT
      CALL COEFF(N,II)
      DO 160 M=1,2
      M1=M-1
      Z=X+MUU(M,II)*Y
      ZB=CONJG(Z)
      LM=LAM(N)
      LJ=LM-1.
      ZL1=LJ*CLOG(Z)
      ZL1=CEXP(ZL1)
      ZBL1=CONJG(LJ)*CLOG(Z)
      ZBL1=CONJG(CEXP(ZBL1))
      ZL0=ZL1*Z
      ZBL0=ZBL1*ZB
      DO 140 I=1,2
      J=(I-1)*4
      J=J+M1*2
      J1=J+1
      J2=J1+1
      UU(I,N)=UU(I,N)+DIS(J1)*ZL0+DIS(J2)*ZBL0
      UUD(I,N)=UUD(I,N)-(DISX(J1)*ZL1+DISX(J2)*ZBL1)*P1CV1
  140 CONTINUE
  160 CONTINUE
  170 CONTINUE
      DO 180 I=1,2
      DO 180 J=1,NBT
      JJ=J*2
      JJ1=JJ-1
      U(I,JJ1)=REAL(UU(I,J))
      U(I,JJ)=-AIMAG(UU(I,J))
      UD(I,JJ1)=REAL(UUD(I,J))
      UD(I,JJ)=-AIMAG(UUD(I,J))
  180 CONTINUE
      RETURN
  200 CONTINUE
      DO 220 I=1,2
      DO 220 J=1,NBT
      UU(I,J)=(0.,0.)
  220 CONTINUE
      DO 270 N=1,NBT
      CALL COEFF(N,II)
      DO 260 M=1,2
      M1=M-1
      Z=X+MUU(M,II)*Y
      ZB=CONJG(Z)
      LM=LAM(N)
      LJ=LM
      ZL0=LJ*CLOG(Z)
      ZL0=CEXP(ZL0)
      ZBL0=CONJG(LJ)*CLOG(Z)
      ZBL0=CONJG(CEXP(ZBL0))
      DO 240 I=1,2
      J=(I-1)*4
      J=J+M1*2
      J1=J+1
```

```
      J2=J1+1
      UU(I,N)=UU(I,N)+DIS(J1)*ZLØ+DIS(J2)*ZBLØ
 24Ø  CONTINUE
 26Ø  CONTINUE
 27Ø  CONTINUE
      DO 28Ø I=1,2
      DO 28Ø J=1,NBT
      JJ=J*2
      JJ1=JJ-1
      U(I,JJ1)=REAL(UU(I,J))
      U(I,JJ)=-AIMAG(UU(I,J))
 28Ø  CONTINUE
      RETURN
 3ØØ  CONTINUE
      DO 32Ø I=1,2
      DO 32Ø J=1,NBT
      UU(I,J)=(Ø.,Ø.)
      UUD(I,J)=(Ø.,Ø.)
      UUDD(I,J)=(Ø.,Ø.)
 32Ø  CONTINUE
      DO 37Ø N=1,NBT
      CALL COEFF(N,II)
      DO 36Ø M=1,2
      M1=M-1
      Z=X+MUU(M,II)*Y
      ZB=CONJG(Z)
      LM=LAM(N)
      LJ=LM-2.
      ZL2=LJ*CLOG(Z)
      ZL2=CEXP(ZL2)
      ZBL2=CONJG(LJ)*CLOG(Z)
      ZBL2=CONJG(CEXP(ZBL2))
      ZL1=ZL2*Z
      ZBL1=ZBL2*ZB
      ZLØ=ZL1*Z
      ZBLØ=ZBL1*ZB
      DO 34Ø I=1,2
      J=(I-1)*4
      J=J+M1*2
      J1=J+1
      J2=J1+1
      UU(I,N)=UU(I,N)+DIS(J1)*ZLØ+DIS(J2)*ZBLØ
      UUD(I,N)=UUD(I,N)-(DISX(J1)*ZL1+DISX(J2)*ZBL1)*P1CV1
      UUDD(I,N)=UUDD(I,N)+(DISXX(J1)*ZL2+DISXX(J2)*ZBL2)*P2CV2
 34Ø  CONTINUE
 36Ø  CONTINUE
 37Ø  CONTINUE
      DO 38Ø I=1,2
      DO 38Ø J=1,NBT
      JJ=J*2
      JJ1=JJ-1
      U(I,JJ1)=REAL(UU(I,J))
      U(I,JJ)=-AIMAG(UU(I,J))
      UD(I,JJ1)=REAL(UUD(I,J))
      UD(I,JJ)=-AIMAG(UUD(I,J))
      UDD(I,JJ1)=REAL(UUDD(I,J))
      UDD(I,JJ)=-AIMAG(UUDD(I,J))
 38Ø  CONTINUE
      RETURN
 5ØØ  CONTINUE
      DO 51Ø  I=1,3
      DO 51Ø  J=1,NBT
      SS(I,J)=(Ø.,Ø.)
 51Ø  CONTINUE
      DO 52Ø I=1,2
```

```
      DO 520 J=1,NBT
      UU(I,J)=(0.,0.)
520   CONTINUE
      DO 570 N=1,NBT
      CALL COEFF(N,II)
      DO 560 M=1,2
      M1=M-1
      Z=X+MUU(M,II)*Y
      ZB=CONJG(Z)
      LM=LAM(N)
      LJ=LM-1.
      ZL1=LJ*CLOG(Z)
      ZL1=CEXP(ZL1)
      ZBL1=CONJG(LJ)*CLOG(Z)
      ZBL1=CONJG(CEXP(ZBL1))
      ZL0=ZL1*Z
      ZBL0=ZBL1*ZB
      DO 530 I=1,3
      J=(I-1)*4
      J=J+M1*2
      J1=J+1
      J2=J1+1
      SS(I,N)=SS(I,N)+(SI(J1)*ZL1+SI(J2)*ZBL1)*P1
530   CONTINUE
      DO 540 I=1,2
      J=(I-1)*4
      J=J+M1*2
      J1=J+1
      J2=J1+1
      UU(I,N)=UU(I,N)+DIS(J1)*ZL0+DIS(J2)*ZBL0
540   CONTINUE
560   CONTINUE
570   CONTINUE
      DO 575 I=1,3
      DO 575 J=1,NBT
      JJ=J*2
      JJ1=JJ-1
      S(I,JJ1)=REAL(SS(I,J))
      S(I,JJ)=-AIMAG(SS(I,J))
575   CONTINUE
      DO 580 I=1,2
      DO 580 J=1,NBT
      JJ=J*2
      JJ1=JJ-1
      U(I,JJ1)=REAL(UU(I,J))
      U(I,JJ)=-AIMAG(UU(I,J))
580   CONTINUE
      RETURN

600   CONTINUE
      DO 610  I=1,3
      DO 610  J=1,NBT
      SS(I,J)=(0.,0.)
610   CONTINUE
      DO 670 N=1,NBT
      CALL COEFF(N,II)
      DO 660 M=1,2
      M1=M-1
      Z=X+MUU(M,II)*Y
      ZB=CONJG(Z)
      LM=LAM(N)
      LJ=LM-1.
      ZL1=LJ*CLOG(Z)
      ZL1=CEXP(ZL1)
      ZBL1=CONJG(LJ)*CLOG(Z)
```

```
      ZBL1=CONJG(CEXP(ZBL1))
      ZL0=ZL1*Z
      ZBL0=ZBL1*ZB
      DO 630 I=1,3
      J=(I-1)*4
      J=J+M1*2
      J1=J+1
      J2=J1+1
      SS(I,N)=SS(I,N)+(SI(J1)*ZL1+SI(J2)*ZBL1)*P1
630   CONTINUE
660   CONTINUE
670   CONTINUE
      DO 675 I=1,3
      DO 675 J=1,NBT
      JJ=J*2
      JJ1=JJ-1
      S(I,JJ1)=REAL(SS(I,J))
      S(I,JJ)=-AIMAG(SS(I,J))
675   CONTINUE
      RETURN

1000  CONTINUE
      IF(L.EQ.8) GO TO 1600
      IF(CV.EQ.0. .AND. L.GE.4) GO TO 1200
      IF(CV.EQ. 0.) GO TO 1500
      IF(L.EQ.2) GO TO 1001
      IF(L.EQ.4) GO TO 1100
      IF(L.EQ.6) GO TO 1800
1001  Z=X+MUU(1,II)*Y
      ZB=CONJG(Z)
      DO 1070 N=1,NBT
      CALL COEFF(N,II)
      LM=LAM(N)
      LJ=LM-4.
      ZL4=LJ*CLOG(Z)
      ZL4=CEXP(ZL4)
      ZBZL4=ZB*ZL4
      ZBL4=CONJG(LJ)*CLOG(Z)
      ZBL4=CONJG(CEXP(ZBL4))
      ZZBL4=Z*ZBL4
      ZL3=ZL4*Z
      ZBZL3=ZB*ZL3
      ZBL3=ZBL4*ZB
      ZZBL3=Z*ZBL3
      ZL2=ZL3*Z
      ZBZL2=ZB*ZL2
      ZBL2=ZBL3*ZB
      ZZBL2=Z*ZBL2
      ZL1=ZL2*Z
      ZBZL1=ZB*ZL1
      ZBL1=ZBL2*ZB
      ZZBL1=Z*ZBL1
      ZL0=ZL1*Z
      ZBL0=ZBL1*ZB
      DO 1030 I=1,3
      J=(I-1)*4
      J1=J+1
      J2=J1+1
      J3=J2+1
      J4=J3+1
      SS(I,N)=(SI(J1)*ZL1+SI(J2)*ZBL1
     1        +SI(J3)*ZBZL2+SI(J4)*ZZBL2)*P1
      SSD(I,N)=-(SIX(J1)*ZL2+SIX(J2)*ZBL2
     1        +SIX(J3)*ZBZL3+SIX(J4)*ZZBL3)*P2CV1
      SSDD(I,N)=(SIXX(J1)*ZL3+SIXX(J2)*ZBL3
```

```
     1             +SIXX(J3)*ZBZL4+SIXX(J4)*ZZBL4)*P3CV2
1030   CONTINUE
       DO 1040 I=1,2
       J=(I-1)*4
       J1=J+1
       J2=J1+1
       J3=J2+1
       J4=J3+1
       UU(I,N)=DIS(J1)*ZL0+DIS(J2)*ZBL0
     1         +DIS(J3)*ZBZL1+DIS(J4)*ZZBL1
       UUD(I,N)=-(DISX(J1)*ZL1+DISX(J2)*ZBL1
     1         +DISX(J3)*ZBZL2+DISX(J4)*ZZBL2)*P1CV1
       UUDD(I,N)=(DISXX(J1)*ZL2+DISXX(J2)*ZBL2
     1         +DISXX(J3)*ZBZL3+DISXX(J4)*ZZBL3)*P2CV2
1040   CONTINUE
1070   CONTINUE
       DO 1075 I=1,3
       DO 1075 J=1,NBT
       JJ=J*2
       JJ1=JJ-1
       S(I,JJ1)=REAL(SS(I,J))
       S(I,JJ)=-AIMAG(SS(I,J))
       SD(I,JJ1)=REAL(SSD(I,J))
       SD(I,JJ)=-AIMAG(SSD(I,J))
       SDD(I,JJ1)=REAL(SSDD(I,J))
       SDD(I,JJ)=-AIMAG(SSDD(I,J))
1075   CONTINUE
       DO 1080 I=1,2
       DO 1080 J=1,NBT
       JJ=J*2
       JJ1=JJ-1
       U(I,JJ1)=REAL(UU(I,J))
       U(I,JJ)=-AIMAG(UU(I,J))
       UD(I,JJ1)=REAL(UUD(I,J))
       UD(I,JJ)=-AIMAG(UUD(I,J))
       UDD(I,JJ1)=REAL(UUDD(I,J))
       UDD(I,JJ)=-AIMAG(UUDD(I,J))
1080   CONTINUE
       RETURN
1100   CONTINUE
       Z=X+MUU(1,II)*Y
       ZB=CONJG(Z)
       DO 1170 N=1,NBT
       CALL COEFF(N,II)
       LM=LAM(N)
       LJ=LM-2.
       ZL2=LJ*CLOG(Z)
       ZL2=CEXP(ZL2)
       ZBZL2=ZB*ZL2
       ZBL2=CONJG(LJ)*CLOG(Z)
       ZBL2=CONJG(CEXP(ZBL2))
       ZZBL2=Z*ZBL2
       ZL1=ZL2*Z
       ZBZL1=ZB*ZL1
       ZBL1=ZBL2*ZB
       ZZBL1=Z*ZBL1
       ZL0=ZL1*Z
       ZBL0=ZBL1*ZB
       DO 1140 I=1,2
       J=(I-1)*4
       J1=J+1
       J2=J1+1
       J3=J2+1
       J4=J3+1
       UU(I,N)=DIS(J1)*ZL0+DIS(J2)*ZBL0
```

```
     1          +DIS(J3)*ZBZL1+DIS(J4)*ZZBL1
       UUD(I,N)=-(DISX(J1)*ZL1+DISX(J2)*ZBL1
     1          +DISX(J3)*ZBZL2+DISX(J4)*ZZBL2)*P1CV1
 1140  CONTINUE
 1170  CONTINUE
       DO 1180 I=1,2
       DO 1180 J=1,NBT
       JJ=J*2
       JJ1=JJ-1
       U(I,JJ1)=REAL(UU(I,J))
       U(I,JJ)=-AIMAG(UU(I,J))
       UD(I,JJ1)=REAL(UUD(I,J))
       UD(I,JJ)=-AIMAG(UUD(I,J))
 1180  CONTINUE
       RETURN
 1200  CONTINUE
       Z=X+MUU(1,II)*Y
       ZB=CONJG(Z)
       DO 1270 N=1,NBT
       CALL COEFF(N,II)
       LM=LAM(N)
       LJ=LM-1.
       ZL1=LJ*CLOG(Z)
       ZL1=CEXP(ZL1)
       ZBZL1=ZB*ZL1
       ZBL1=CONJG(LJ)*CLOG(Z)
       ZBL1=CONJG(CEXP(ZBL1))
       ZZBL1=Z*ZBL1
       ZL0=ZL1*Z
       ZBL0=ZBL1*ZB
       DO 1240 I=1,2
       J=(I-1)*4
       J1=J+1
       J2=J1+1
       J3=J2+1
       J4=J3+1
       UU(I,N)=DIS(J1)*ZL0+DIS(J2)*ZBL0
     1          +DIS(J3)*ZBZL1+DIS(J4)*ZZBL1
 1240  CONTINUE
 1270  CONTINUE
       DO 1280 I=1,2
       DO 1280 J=1,NBT
       JJ=J*2
       JJ1=JJ-1
       U(I,JJ1)=REAL(UU(I,J))
       U(I,JJ)=-AIMAG(UU(I,J))
 1280  CONTINUE
       RETURN
 1300  CONTINUE
       Z=X+MUU(1,II)*Y
       ZB=CONJG(Z)
       DO 1370 N=1,NBT
       CALL COEFF(N,II)
       LM=LAM(N)
       LJ=LM-3.
       ZL3=LJ*CLOG(Z)
       ZL3=CEXP(ZL3)
       ZBZL3=ZB*ZL3
       ZBL3=CONJG(LJ)*CLOG(Z)
       ZBL3=CONJG(CEXP(ZBL3))
       ZZBL3=Z*ZBL3
       ZL2=ZL3*Z
       ZBZL2=ZB*ZL2
       ZBL2=ZBL3*ZB
       ZZBL2=Z*ZBL2
```

```
      ZL1=ZL2*Z
      ZBZL1=ZB*ZL1
      ZBL1=ZBL2*ZB
      ZZBL1=Z*ZBL1
      ZL0=ZL1*Z
      ZBL0=ZBL1*ZB
      DO 1340 I=1,2
      J=(I-1)*4
      J1=J+1
      J2=J1+1
      J3=J2+1
      J4=J3+1
      UU(I,N)=DIS(J1)*ZL0+DIS(J2)*ZBL0
     1          +DIS(J3)*ZBZL1+DIS(J4)*ZZBL1
      UUD(I,N)=-(DISX(J1)*ZL1+DISX(J2)*ZBL1
     1          +DISX(J3)*ZBZL2+DISX(J4)*ZZBL2)*P1CV1
      UUDD(I,N)=(DISXX(J1)*ZL2+DISXX(J2)*ZBL2
     1          +DISXX(J3)*ZBZL3+DISXX(J4)*ZZBL3)*P2CV2
 1340 CONTINUE
 1370 CONTINUE
      DO 1380 I=1,2
      DO 1380 J=1,NBT
      JJ=J*2
      JJ1=JJ-1
      U(I,JJ1)=REAL(UU(I,J))
      U(I,JJ)=-AIMAG(UU(I,J))
      UD(I,JJ1)=REAL(UUD(I,J))
      UD(I,JJ)=-AIMAG(UUD(I,J))
      UDD(I,JJ1)=REAL(UUDD(I,J))
      UDD(I,JJ)=-AIMAG(UUDD(I,J))
 1380 CONTINUE
      RETURN
 1500 CONTINUE
      Z=X+MUU(1,II)*Y
      ZB=CONJG(Z)
      DO 1570 N=1,NBT
      CALL COEFF(N,II)
      LM=LAM(N)
      LJ=LM-2.
      ZL2=LJ*CLOG(Z)
      ZL2=CEXP(ZL2)
      ZBZL2=ZB*ZL2
      ZBL2=CONJG(LJ)*CLOG(Z)
      ZBL2=CONJG(CEXP(ZBL2))
      ZZBL2=Z*ZBL2
      ZL1=ZL2*Z
      ZBZL1=ZB*ZL1
      ZBL1=ZBL2*ZB
      ZZBL1=Z*ZBL1
      ZL0=ZL1*Z
      ZBL0=ZBL1*ZB
      DO 1530 I=1,3
      J=(I-1)*4
      J1=J+1
      J2=J1+1
      J3=J2+1
      J4=J3+1
      SS(I,N)=(SI(J1)*ZL1+SI(J2)*ZBL1
     1          +SI(J3)*ZBZL2+SI(J4)*ZZBL2)*P1
 1530 CONTINUE
      DO 1540 I=1,2
      J=(I-1)*4
      J1=J+1
      J2=J1+1
      J3=J2+1
```

```
          J4=J3+1
          UU(I,N)=DIS(J1)*ZL0+DIS(J2)*ZBL0
     1          +DIS(J3)*ZBZL1+DIS(J4)*ZZBL1
 1540     CONTINUE
 1570     CONTINUE
          DO 1575 I=1,3
          DO 1575 J=1,NBT
          JJ=J*2
          JJ1=JJ-1
          S(I,JJ1)=REAL(SS(I,J))
          S(I,JJ)=-AIMAG(SS(I,J))
 1575     CONTINUE
          DO 1580 I=1,2
          DO 1580 J=1,NBT
          JJ=J*2
          JJ1=JJ-1
          U(I,JJ1)=REAL(UU(I,J))
          U(I,JJ)=-AIMAG(UU(I,J))
 1580     CONTINUE
          RETURN
 1600     CONTINUE
          Z=X+MUU(1,II)*Y
          ZB=CONJG(Z)
          DO 1670 N=1,NBT
          CALL COEFF(N,II)
          LM=LAM(N)
          LJ=LM-2.
          ZL2=LJ*CLOG(Z)
          ZL2=CEXP(ZL2)
          ZBZL2=ZB*ZL2
          ZBL2=CONJG(LJ)*CLOG(Z)
          ZBL2=CONJG(CEXP(ZBL2))
          ZZBL2=Z*ZBL2
          ZL1=ZL2*Z
          ZBZL1=ZB*ZL1
          ZBL1=ZBL2*ZB
          ZZBL1=Z*ZBL1
          ZL0=ZL1*Z
          ZBL0=ZBL1*ZB
          DO 1630 I=1,3
          J=(I-1)*4
          J1=J+1
          J2=J1+1
          J3=J2+1
          J4=J3+1
          SS(I,N)=(SI(J1)*ZL1+SI(J2)*ZBL1
     1          +SI(J3)*ZBZL2+SI(J4)*ZZBL2)*P1
 1630     CONTINUE
 1670     CONTINUE
          DO 1675 I=1,3
          DO 1675 J=1,NBT
          JJ=J*2
          JJ1=JJ-1
          S(I,JJ1)=REAL(SS(I,J))
          S(I,JJ)=-AIMAG(SS(I,J))
 1675     CONTINUE
          RETURN
C
          END
```

```
      SUBROUTINE PRECRCK(C,NUMMAT,TT,IL)
C
      COMMON/BK10/R(300),Z(300),CODE(300),IX(6,250)
      COMMON/TIP/NCR1,NCR2,NCR3,NCR4,NELX,CTPX,CTPY,SIF1,SIF2
       COMMON /C7/ICASE(2)
       COMMON /C1/IY,P2IY,PEI2
       COMPLEX MU(2),IY,P2IY
       IY=(0.,1.)
       PEI2=2.*ACOS(-1.)
       P2IY=PEI2*IY
       WRITE(6,130) TT,C
       NMAT=2
       IF(IX(6,NCR1).EQ.IX(6,NCR2)) NMAT=1
       DO 10 II=1,NMAT
       IF(II.EQ.1) IIS=IX(6,NCR1)
       IF(II.EQ.2) IIS=IX(6,NCR2)
       CALL ROOTS(C,II,IIS,MU)
       IF(ICASE(II) .EQ. 1)CALL MULT1(MU,II,IIS)
       IF(ICASE(II) .EQ. 2)CALL MULT2(MU,II,IIS)
10     CONTINUE
       IF(NMAT .EQ. 2) GO TO 20
       CALL EQUATE
20     CALL EIGEN(IL)
       RETURN
130    FORMAT(///,30X,6HTIME =,1PE13.4,5X,20HCRACK-TIP VELOCITY =,
     1 1PE13.4,//)
       END
```

```
      SUBROUTINE ROOTS(C,II,IIS,MU)
      COMPLEX MU(2),MUU,IY,P2IY
      DIMENSION P(5),R(4),AI(4),H(6),BB(6),CC(6),ALPHA(2),BETA(2)
      COMMON/SUMAN/A(3,3,6),AJI(3,3,6),ADET(6),ASIZE(6),CL
      COMMON/RHHO/RHO(6),RODUM
      COMMON/RIG/IRIG1(6),IRIG2
      COMMON/TOLR/TOLER1,TOLER2
      COMMON /C1/IY,P2IY,PEI2
      COMMON /C3/AA(3,3)
      COMMON /C7/ICASE(2)
      COMMON /C14/MUU(2,2)
      IF(IRIG1(IIS).EQ.0) GO TO 2
      AA(1,1)=0.
      AA(2,1)=1.
      AA(3,1)=0.
      AA(1,2)=1.
      AA(2,2)=0.
      AA(3,2)=0.
      AA(1,3)=0.
      AA(2,3)=0.
      AA(3,3)=-1.
C  FOR A RIGID MATERIAL THE CHARACTERISTIC EQUATION BECOMES AN IDENTITY
C  AND THEREFORE THE ROOTS CAN BE ARBITRARILY CHOSEN TO HAVE ANY VALUES
C  HERE WE ARBITRARILY CHOOSE A MULTIPLE ROOT OF MU(1)=(0.,1.),JUST TO
C  HAVE SOME VALUE FOR THE ROOTS OF THE CHARACTERISTIC EQUATION SO THAT
C  THE CALCULATIONS CAN GO ON. ANY OTHER VALUES FOR MU(1) AND MU(2)
C  COULD HAVE BEEN CHOSEN AS WELL. THE SOLUTION WILL NOT DEPEND ON THESE
C  ROOTS AND THE SOLUTION WILL BE THE SAME REGARDLESS OF THE ROOTS
C  CHOSEN HERE.
      ICASE(II)=2
      MU(1)=(0.,1.)
      GO TO 32
 2    R1=RHO(IIS)*C*C
      A1=A(1,1,IIS)
      A2=A(2,1,IIS)
      A3=A(3,1,IIS)
      A4=A(2,2,IIS)
      A5=A(3,2,IIS)
      A6=A(3,3,IIS)
      B1=A4*A6-A5*A5
      B2=A3*A5-A2*A6
      B3=A2*A5-A3*A4
      B4=A1*A6-A3*A3
      B5=A2*A3-A1*A5
      B6=A1*A4-A2*A2
      D4=1.-(A1-A2)*R1
      IF(ABS(D4).LT.TOLER1) GO TO 34
      D1=A3*(A3-A5)*R1*R1-(1.-A6*R1)*D4
      D2=-(B5*R1+A3)*R1
      D3=-(A5+B5*R1)*R1
      DD=-(D1*D4-D2*D3)
      IF(ABS(DD).LT.TOLER1) GO TO 34
      AA(1,1)=(A2+B2*R1)*R1
      AA(2,1)=(1.-(A1+A6-B4*R1)*R1)
      AA(3,1)=(A5+B5*R1)*R1
      AA(1,2)=(1.+A2*R1)
      AA(2,2)=-A1*R1
      AA(3,2)=0.
      AA(1,3)=-(A3+A5)*R1
      AA(2,3)=2.*A3*R1
      AA(3,3)=-D4
      IF(A1.EQ.0.) GO TO 34
      P(1)=(A4-R1*(B1+B6)+R1*R1*ADET(IIS))/A1
      P(2)=-2.*(A5+R1*(B3+B5))/A1
      P(3)=(A6+2.*A2-R1*(B4+B6))/A1
```

```
      P(4)=-2.*A3/A1
      P(5)=1.
      IF(P(1).LE.0.) GO TO 34
C SOLVE FOR THE ROOTS ANATICALLY IF P(2)=P(4)=0.
      IF(ABS(P(2)).LT.TOLER1 .AND. ABS(P(4)).LT.TOLER1) GO TO 210
      GO TO 250
 210  IF(ABS(P(3)) .GT. TOLER1) GO TO 220
      MU(1)=(P(1)/4.)**.25*(1+IY)
      MU(2)=-CONJG(MU(1))
      ICASE(II)=1
      GO TO 23
 220  D=P(3)*P(3)-4.*P(1)
      IF(D.LT.0.) GO TO 230
      GO TO 240
 230  D=-P(3)+2.*SQRT(P(1))
      IF(D.LT.0.) GO TO 34
      MU(1)=(SQRT(D)+IY*SQRT(P(3)+2.*SQRT(P(1))))/2.
      IF(REAL(MU(1)).GT.TOLER1) GO TO 235
      MU(1)=IY*SQRT(P(3)/2.)
      ICASE(II)=2
      GO TO 32
 235  MU(2)=-CONJG(MU(1))
      ICASE(II)=1
      GO TO 23
 240  IF(P(3) .LT. 0.) GO TO 34
      D=SQRT(D)
      D1=(P(3)+D)/2.
      D2=(P(3)-D)/2.
      D1=SQRT(D1)
      D2=SQRT(D2)
      IF((D1-D2).LT.TOLER1*D1) GO TO 247
      MU(1)=IY*D1
      MU(2)=IY*D2
      ICASE(II)=1
      GO TO 23
 247  MU(1)=IY*(D1+D2)/2.
      ICASE(II)=2
      GO TO 32
C SOLVE FOR THE ROOTS ANATICALLY IF P(3)=(P(4)**2)/4.+2.*SQRT(P(1))
C AND P(2)=P(4)*SQRT(P(1))  WHICH GIVES ONLY ONE PAIR OF COMPLEX
C ROOTS OF MULTIPLICITY  2
C NOTE THAT THE BOEING LIBRARY ROUTINE "PROOT" (BELOW) IS NOT GOOD
C IN FINDING MULTIPLE ROOTS
250   IF(ABS(P(3)-P(4)*P(4)/4.-2.*SQRT(P(1))) .LT. TOLER1  .AND.
     1  ABS(P(2)-P(4)*SQRT(P(1))).LT.TOLER1) GO TO 260
      GO TO 270
260   D=SQRT(P(1))-P(4)*P(4)/16.
      IF(D.LT.0.) GO TO 34
      D=SQRT(D)
      MU(1)=-P(4)/4.+IY*D
      IF(ABS(REAL(MU(1))).LT.TOLER1) MU(1)=IY*D
      ICASE(II)=2
      GO TO 32
C SOLVE FOR THE ROOTS NUMERICALLY
 270  CALL PROOT(4,P,R,AI,H,BB,CC,CONV)
      IF(CONV.LT.0.) GO TO 36
      NN=0
      DO 5 N=1,4
      IF(ABS(R(N)) .LT.  TOLER1) R(N)=0.
      BET1=AI(N)
      IF(BET1 .LT. 0.)GO TO 5
      NN=NN+1
      BETA(NN)=BET1
      ALPHA(NN)=R(N)
    5 CONTINUE
```

```
      IF(NN .NE. 2)GO TO 38
      R1=ABS(ALPHA(2)-ALPHA(1))
      RR1=TOLER1*ABS(ALPHA(2))
      R2=ABS(BETA(2)-BETA(1))
      RR2=TOLER1*ABS(BETA(2))
      ICASE(II)=1
      IF(R1 .LE. RR1    .AND.    R2 .LE. RR2) ICASE(II)=2
      IF(ICASE(II) .EQ. 1)GO TO 10
      IF(ICASE(II) .EQ. 2)GO TO 30
 10   DO 12 I=1,2
      MU(I)=CMPLX(ALPHA(I),BETA(I))
 12   CONTINUE
 23   DO 25 N=1,2
      MUU(N,II)=MU(N)
 25   CONTINUE
      WRITE(6,48) IIS,MU
      GO TO 40
30    MU(1)=CMPLX((ALPHA(1)+ALPHA(2))/2.,(BETA(1)+BETA(2))/2.)
32    MUU(1,II)=MU(1)
      IF(IRIG1(IIS).EQ.1) WRITE(6,60) IIS,MU(1)
      IF(IRIG1(IIS).EQ.1) GO TO 40
      WRITE(6,50) IIS,MU(1)
      GO TO 40
 34   WRITE(6,92) IIS
      GO TO 39
 36   WRITE(6,94) IIS
      GO TO 39
 38   WRITE(6,96) IIS
 39   STOP 11
 40   RETURN
 48   FORMAT(////,10X,41HTHE CHARACTERISTIC EQUATION FOR MATERIAL#,I1,
     1 1X,13HHAS ROOTS OF:,/,30X,1H(,F16.8,1H,,F16.8,1H),5X,5HAND   ,
     21H(,F16.8,1H,,F16.8,1H))
 50   FORMAT(//,10X,41HTHE CHARACTERISTIC EQUATION FOR MATERIAL#,I1,1X,
     1 23HHAS A MULTIPLE ROOT OF:,5X,1H(,F16.8,1H,,F16.8,1H))
 60   FORMAT(//,10X,41HTHE CHARACTERISTIC EQUATION FOR MATERIAL#,I1,1X,/
     1,10X,54HHAS ARBIRARILY BEEN CHOSEN TO HAVE A MULTIPLE ROOT OF:,
     25X,1H(,F16.8,1H,,F16.8,1H))
 92   FORMAT(///,10X,13HFOR MATERIAL#,I1,/,
     1           5X,81HTHE GIVEN COMBINATION OF THE MATERIAL PROPERTIES A
     2ND THE CRACK-TIP SPEED WILL NOT,/,5X,72HRESULT IN COMPLEX CONJUGAT
     3E ROOTS FOR THE CHARACTERISTIC EQUATION.       ,/,5X,69HTRY LOWERIN
     4G THE CRACK-TIP SPEED OR CHANGING THE MATERIAL PROPERTIES.,)
 94   FORMAT(///,10X,13HFOR MATERIAL#,I1,/,
     1           5X,96HTHE BOEING LIBRARY ROUTINE 'PROOT' WAS NOT ABLE TO
     2 FIND THE ROOTS OF THE CHARACTERISTIC EQUATON.,/,5X,69HTRY LOWERIN
     3G THE CRACK-TIP SPEED OR CHANGING THE MATERIAL PROPERTIES.,)
 96   FORMAT(///,10X,13HFOR MATERIAL#,I1,/,
     1           5X,114HTHE BOEING LIBRARY ROUTINE 'PROOT' DID NOT FIND P
     2AIR(S) OF COMPLEX CONJUGATE ROOTS FOR THE CHARACTERISTIC EQUATON.,
     3                                              /,5X,69HTRY LOWERIN
     4G THE CRACK-TIP SPEED OR CHANGING THE MATERIAL PROPERTIES.,)
      END
```

```
      SUBROUTINE MULT1(MU,II,IIS)
      COMMON /C1/IY,P2IY,PEI2
      COMMON /SUMAN/A(3,3,6),AI(3,3,6),ADET(6),ASIZE(6),CL
      COMMON/RHHO/RHO(6),RODUM
      COMMON/RIG/IRIG1(6),IRIG2
      COMMON /C3/AA(3,3)
      COMMON /C4/SIG(3,2,2),DISP(2,2,2)
      COMMON /C5/K2(2,2),K2B(2,2),K(8,2),KB(8,2)
      COMPLEX MU(2),K1(8),K1B(4),P(3,2)
      COMPLEX K2,K2B,K,KB,MIS1,MIS2,IY,P2IY,SIG,DISP
      P(1,1)=(1.,0.)
      P(2,1)=MU(1)*MU(1)
      P(3,1)=MU(1)
      P(1,2)=(1.,0.)
      P(2,2)=MU(2)*MU(2)
      P(3,2)=MU(2)
      DO 20 J=1,2
      DO 20 I=1,3
      SIG(I,J,II)=0.
      DO 20 L=1,3
      SIG(I,J,II)=SIG(I,J,II)+AA(I,L)*P(L,J)
   20 CONTINUE
      DO 30 J=1,2
      DO 30 I=1,2
      DISP(I,J,II)=(0.,0.)
      DO 30 L=1,3
      DISP(I,J,II)=DISP(I,J,II)+A(I,L,IIS)*SIG(L,J,II)
   30 CONTINUE
      DO 40 I=1,2
      I2=2*I
      I1=I2-1
      MIS1=SIG(2,I,II)
      MIS2=SIG(3,I,II)*IY
      K1(I1)=(MIS1-MIS2)/2.
      MIS2=(MIS1+MIS2)/2.
      K1(I2)=CONJG(MIS2)
      DISP(2,I,II)=DISP(2,I,II)/MU(I)
      MIS1=DISP(1,I,II)
      MIS2=DISP(2,I,II)*IY
      K1(I1+4)=(MIS1+MIS2)/2.
      MIS2=(MIS1-MIS2)/2.
      K1(I2+4)=CONJG(MIS2)
   40 CONTINUE
      DO 50 I=1,4
      K1B(I)=CONJG(K1(I))
   50 CONTINUE
      R=REAL(K1(4)*K1B(4)-K1(3)*K1B(3))
      K2(1,II)=(K1(1)*K1B(3)-K1(4)*K1B(2))/R
      K2(2,II)=(K1(2)*K1B(3)-K1(4)*K1B(1))/R
      DO 55 I=1,2
      K2B(I,II)=CONJG(K2(I,II))
   55 CONTINUE
      K(1,II)=-K1(4)*K2B(2,II)
      K(2,II)=-K(1,II)
      K(3,II)=-K1(3)*K2(2,II)
      K(4,II)=-K(3,II)
      K(5,II)=K1(5)+K1(7)*K2(1,II)
      K(6,II)=K1(8)*K2B(2,II)
      K(7,II)=K1(6)+K1(8)*K2B(1,II)
      K(8,II)=K1(7)*K2(2,II)
      DO 60 I=1,8
      KB(I,II)=CONJG(K(I,II))
   60 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MULT2(MU,II,IIS)
      COMMON /C1/IY,P2IY,PEI2
      COMMON /SUMAN/A(3,3,6),AI(3,3,6),ADET(6),ASIZE(6),CL
      COMMON/RHHO/RHO(6),RODUM
      COMMON/RIG/IRIG1(6),IRIG2
      COMMON /C3/AA(3,3)
      COMMON /C4/SIG(3,2,2),DISP(2,2,2)
      COMMON /C5/K2(2,2),K2B(2,2),K(8,2),KB(8,2)
      COMPLEX MU(2),K1(8),K1B(4),P(3,2),EPS(2,2)
      COMPLEX K2,K2B,K,KB,MIS1,MIS2,MV,IY,P2IY,SIG,DISP
      MV=MU(1)
      P(1,1)=(1.,0.)
      P(2,1)=MV*MV
      P(3,1)=MV
      P(1,2)=(2.,0.)
      P(2,2)=-2*IY*MV
      P(3,2)=MV-IY
      DO 20 J=1,2
      DO 20 I=1,3
      SIG(I,J,II)=(0.,0.)
      DO 20 L=1,3
      SIG(I,J,II)=SIG(I,J,II)+AA(I,L)*P(L,J)
 20   CONTINUE
      DO 30 J=1,2
      DO 30 I=1,2
      EPS(I,J)=(0.,0.)
      DO 30 L=1,3
      EPS(I,J)=EPS(I,J)+A(I,L,IIS)*SIG(L,J,II)
 30   CONTINUE
      DISP(1,1,II)=EPS(1,1)
      DISP(1,2,II)=EPS(1,2)-EPS(1,1)
      DISP(2,1,II)=EPS(2,1)/MV
      DISP(2,2,II)=(EPS(2,2)+EPS(2,1)*IY/MV)/MV
      DO 40 I=1,2
      I2=2*I
      I1=I2-1
      MIS1=SIG(2,I,II)
      MIS2=SIG(3,I,II)*IY
      K1(I1)=(MIS1-MIS2)/2.
      MIS2=(MIS1+MIS2)/2.
      K1(I2)=CONJG(MIS2)
      MIS1=DISP(1,I,II)
      MIS2=DISP(2,I,II)*IY
      K1(I1+4)=(MIS1+MIS2)/2.
      MIS2=(MIS1-MIS2)/2.
      K1(I2+4)=CONJG(MIS2)
 40   CONTINUE
      DO 50 I=1,4
      K1B(I)=CONJG(K1(I))
 50   CONTINUE
      R=REAL(K1(2)*K1B(2)-K1(1)*K1B(1))
      K2(1,II)=(K1(3)*K1B(1)-K1(2)*K1B(4))/R
      K2(2,II)=(K1(4)*K1B(1)-K1(2)*K1B(3))/R
      DO 55 I=1,2
      K2B(I,II)=CONJG(K2(I,II))
 55   CONTINUE
      K(1,II)=-K1(2)*K2B(2,II)
      K(2,II)=-K(1,II)
      K(3,II)=-K1(1)*K2(2,II)
      K(4,II)=-K(3,II)
      K(5,II)=K1(5)*(K2(1,II)+1.)+K1(7)
      K(6,II)=K1(6)*K2B(2,II)
      K(7,II)=K1(6)*(K2B(1,II)+1.)+K1(8)
      K(8,II)=K1(5)*K2(2,II)
      DO 60 I=1,8
```

```
      KB(I,II)=CONJG(K(I,II))
60    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE EIGEN(IL)
      COMPLEX LAM
      COMPLEX B1,B2,C1,C2,C3,C4,C5,C6,C7,C8,CC
      COMPLEX K2,K2B,K,KB,IY,P2IY,F,FF
      COMMON /SUMAN/A(3,3,6),AI(3,3,6),ADET(6),ASIZE(6),CL
      COMMON/DIM/NA,NAA,NBT,NB,NQ,NR,NINT,NINT2,IALL,NITER,SINCOD
      COMMON/RIG/IRIG1(6),IRIG2
      COMMON/TOLR/TOLER1,TOLER2
      COMMON /C1/IY,P2IY,PEI2
      COMMON /C5/K2(2,2),K2B(2,2),K(8,2),KB(8,2)
      COMMON /C10/LAM(20)
      COMMON /C12/F(8)
      COMMON /C16/FF(4,2)
      COMMON /C8/H(4,4)
      COMMON /C6/Z(2)
      COMPLEX Z,H
      A11=REAL(K(1,1)*KB(1,1)-K(3,1)*KB(3,1))
      A21=REAL(K(5,1)*KB(5,1)+K(6,1)*KB(6,1)-
     1        K(7,1)*KB(7,1)-K(8,1)*KB(8,1))
      A12=REAL(K(1,2)*KB(1,2)-K(3,2)*KB(3,2))
      A22=REAL(K(5,2)*KB(5,2)+K(6,2)*KB(6,2)-
     1        K(7,2)*KB(7,2)-K(8,2)*KB(8,2))
      B1=K(5,1)*KB(6,1)-K(8,1)*KB(7,1)
      B2=K(6,2)*KB(5,2)-K(7,2)*KB(8,2)
      C1=K(1,1)*KB(3,2)-K(1,2)*KB(3,1)
      C2=K(7,2)*KB(5,1)+K(8,2)*KB(6,1)-K(7,1)*KB(5,2)-K(8,1)*KB(6,2)
      C3=K(7,2)*KB(6,1)-K(8,1)*KB(5,2)
      C4=K(8,2)*KB(5,1)-K(7,1)*KB(6,2)
      C5=K(3,2)*KB(3,1)-K(1,1)*KB(1,2)
      C6=K(5,2)*KB(5,1)-K(7,1)*KB(7,2)
      C7=K(6,2)*KB(5,1)+K(5,2)*KB(6,1)-K(7,1)*KB(8,2)-K(8,1)*KB(7,2)
      C8=K(6,2)*KB(6,1)-K(8,1)*KB(8,2)
      CC=-C1*C3+C5*C8+CONJG(C5*C6-C1*C4)
      IF(REAL(B1).EQ.0. .AND. AIMAG(B1).NE.0.) STOP 12
      IF(REAL(B2).EQ.0. .AND. AIMAG(B2).NE.0.) STOP 12
      IF(REAL(CC).EQ.0. .AND. AIMAG(CC).NE.0.) STOP 12
      IF(REAL(B1).EQ.0.) GO TO 2
      IF(ABS(AIMAG(B1)/REAL(B1)).GT.TOLER1) STOP 12
    2 IF(REAL(B2).EQ.0.) GO TO 4
      IF(ABS(AIMAG(B2)/REAL(B2)).GT.TOLER1) STOP 12
    4 IF(REAL(CC).EQ.0.) GO TO 6
      IF(ABS(AIMAG(CC)/REAL(CC)).GT.TOLER1) STOP 12
    6 CONTINUE
      R=-A11*A22-A12*A21+2.*REAL(C5*C7-C1*C2)
      Q=REAL(-A11*B2-A12*B1+CC)
      R=R/(2.*Q)
      D=R*R-1.
      IF(D.LT.-TOLER1) STOP 12
      IF(D.GE.-TOLER1 .AND. D.LT.0.) D=0.
      SIZE=R+SQRT(D)
      EPS=ALOG(SIZE)/PEI2
      IF(EPS.GE.-TOLER1 .AND. EPS.LT.0.) EPS=0.
      LAM(1)=CMPLX(.5,EPS)
      LAM(2)=(1.,0.)
      DO 50 I=1,2
      IF(I.EQ.2) GO TO 30
      IF(EPS.EQ.0.) GO TO 30
      CALL MAT(LAM(I),1,3)
      CALL MAT(LAM(I),2,3)
      IDET=0
      CALL INV33(IDET)
      IF(IDET.EQ.0) GO TO 20
      EPS=-EPS
      LAM(I)=CONJG(LAM(I))
      CALL MAT(LAM(I),1,3)
```

```
      CALL MAT(LAM(I),2,3)
      IDET=0
      CALL INV33(IDET)
      IF(IDET.EQ.0) GO TO 20
      WRITE(6,140) EPS
      WRITE(6,150)
      IL=1
      RETURN
 20   IL=0
      WRITE(6,140) EPS
      FF(1,I)=F(1)
      FF(2,I)=F(2)
      FF(3,I)=F(5)
      FF(4,I)=F(6)
      GO TO 50
   30 CONTINUE
      IF(IRIG2.EQ.1) GO TO 100
      CALL MAT(LAM(I),1,2)
      CALL MAT(LAM(I),2,2)
      IDET=0
      CALL INV22(IDET)
      IF(IDET.EQ.0) GO TO 32
      WRITE(6,150)
      IL=1
      RETURN
 32   IL=0
      FF(1,I)=F(1)
      FF(2,I)=F(2)
      FF(3,I)=F(5)
      FF(4,I)=F(6)
   50 CONTINUE
      KK=0
      IF(NBT.LE.2) RETURN
      DO 70 I=3,NBT,2
      KK=KK+1
      LAM(I)=LAM(1)+KK
      LAM(I+1)=LAM(2)+KK
 70   CONTINUE
      RETURN
 100  CONTINUE
      FF(1,2)=FF(1,1)
      FF(2,2)=FF(2,1)
      FF(3,2)=FF(3,1)
      FF(4,2)=FF(4,1)
      DO 110 I=2,NBT
      LAM(I)=1.+LAM(I-1)
 110  CONTINUE
      RETURN
 140  FORMAT(/////,30X,38HTHE ELASTIC BIMATERIAL COEFFICIENT IS:,F16.8,
     1 //////)
150   FORMAT(//,10X,105HTHE COMBINATION OF MATERIAL(S) PROPERTIES AND T
     1HE GIVEN CRACK-TIP VELOCITY PRESENTS AN EXTREME SITUATION.,/,10X,
     2102HTHE DETERMINANT OF EIGEN-FUNCTION COEFFICIENT MATRIX VANISHES
     3FOR THE EIGEN-VALUES,THEREFORE MAKING IT,/,10X,86HIMPOSSIBLE TO DE
     4TERMINE THE RELATIVE VALUES OF THE COEFFICIENTS OF THE EIGEN-FUNCT
     5ION.,/,10X,69HTRY LOWERING THE CRACK-TIP SPEED OR CHANGING THE MAT
     6ERIAL PROPERTIES.)
C
      END
```

```
      SUBROUTINE INV22(IDET)
      COMPLEX HI(2),H,F,IY,P2IY
     COMMON/BK10/R1(300),Z(300),CODE(300),IX(6,250)
     COMMON/TIP/NCR1,NCR2,NCR3,NCR4,NELX,CTPX,CTPY,SIF1,SIF2
      COMMON /SUMAN/A(3,3,6),AI(3,3,6),ADET(6),ASIZE(6),CL
      COMMON/TOLR/TOLER1,TOLER2
      COMMON /C1/IY,P2IY,PEI2
      COMMON /C8/H(4,4)
      COMMON /C12/F(8)
      R=REAL(H(3,3)*H(4,4)-H(4,3)*H(3,4))
      SA=ASIZE(IX(6,NCR2))
      SA=SA*SA*TOLER1
      IF(ABS(R) .GT. SA) GO TO 5
      IDET=1
      RETURN
5     HI(1)=H(4,4)/R
      HI(2)=-H(3,4)/R
      F(1)=(1.,0.)
      F(2)=(0.,0.)
      DO 10 I=1,2
      J=I+4
      F(J)=(0.,0.)
      DO 10 K=1,2
      F(J)=F(J)+HI(K)*H(K+2,I)
10    CONTINUE
      F(6)=CONJG(F(6))
      RETURN
      END
```

```
      SUBROUTINE INV33(IDET)
      COMMON/BK10/R(300),Z(300),CODE(300),IX(6,250)
      COMMON/TIP/NCR1,NCR2,NCR3,NCR4,NELX,CTPX,CTPY,SIF1,SIF2
      COMPLEX DETER,HH(3,3),HK(3),HI(3,3),H,F,IY,P2IY
      COMMON /SUMAN/A(3,3,6),AI(3,3,6),ADET(6),ASIZE(6),CL
      COMMON/TOLR/TOLER1,TOLER2
      COMMON /C1/IY,P2IY,PEI2
      COMMON /C8/H(4,4)
      COMMON /C12/F(8)
      EQUIVALENCE(H(1,1),HI(1,1))
      HK(1)=H(1,1)
      HK(2)=H(2,1)
      HK(3)=H(4,1)
      HH(1,1)=-H(1,2)
      HH(2,1)=-H(2,2)
      HH(3,1)=-H(4,2)
      HH(1,2)=H(1,3)
      HH(2,2)=H(2,3)
      HH(3,2)=H(4,3)
      HH(1,3)=H(1,4)
      HH(2,3)=H(2,4)
      HH(3,3)=H(4,4)
      HI(1,1)=HH(2,2)*HH(3,3)-HH(3,2)*HH(2,3)
      HI(1,2)=HH(3,2)*HH(1,3)-HH(1,2)*HH(3,3)
      HI(1,3)=HH(1,2)*HH(2,3)-HH(2,2)*HH(1,3)
      DETER=HI(1,1)*HH(1,1)+HI(1,2)*HH(2,1)+HI(1,3)*HH(3,1)
      SA=(ASIZE(IX(6,NCR1))+ASIZE(IX(6,NCR2)))/2.
      IF(CABS(DETER) .GT. SA*TOLER1) GO TO 15
      IDET=1
      RETURN
15    DO 20 I=1,3
      HI(1,I)=HI(1,I)/DETER
20    CONTINUE
      HI(2,1)=(HH(3,1)*HH(2,3)-HH(2,1)*HH(3,3))/DETER
      HI(2,2)=(HH(1,1)*HH(3,3)-HH(3,1)*HH(1,3))/DETER
      HI(2,3)=(HH(2,1)*HH(1,3)-HH(1,1)*HH(2,3))/DETER
      HI(3,1)=(HH(2,1)*HH(3,2)-HH(3,1)*HH(2,2))/DETER
      HI(3,2)=(HH(3,1)*HH(1,2)-HH(1,1)*HH(3,2))/DETER
      HI(3,3)=(HH(1,1)*HH(2,2)-HH(2,1)*HH(1,2))/DETER
      F(1)=(1.,0.)
      DO 30 I=1,3
      J=I+3
      IF(I .EQ. 1) J=2
      F(J)=(0.,0.)
      DO 30 K=1,3
      F(J)=F(J)+HI(I,K)*HK(K)
30    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MAT(X,II,IFLAG)
      COMPLEX X,Z,H,K2,K2B,K,KB,IY,P2IY
      COMMON /C1/IY,P2IY,PEI2
      COMMON /C5/K2(2,2),K2B(2,2),K(8,2),KB(8,2)
      COMMON /C6/Z(2)
      COMMON /C8/H(4,4)
      IM=1
      IF(IFLAG .EQ. 2) IM=3
      Z(1)=CEXP(P2IY*X)
      Z(2)=1/Z(1)
      DO 90 I=IM,4
      I2=2*I
      JJ=2
      III=3
      JF=I2-3
      IF(I .LE. 2) III=1
      IF(I .EQ. 1 .OR. I .EQ. 3) GO TO 10
      GO TO 20
10    JJ=1
      JF=I2+1
 20   KK=JJ
      IF(II .EQ. 2)KK=3-JJ
      JK=(II-1)*2
      JJJ=JJ+JK
      IF(IFLAG.EQ.3 .AND. III.EQ.3) GO TO 85
      H(III,JJJ)=K(I2-1,II)+K(I2,II)*Z(KK)
85    H(III+1,JJJ)=KB(JF,II)+KB(JF+1,II)*Z(KK)
90    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE COEFF(I,II)
      COMPLEX LAM,X,F,IY,P2IY,FF
      COMMON /C1/IY,P2IY,PEI2
      COMMON /C7/ICASE(2)
      COMMON /C10/LAM(20)
      COMMON /C12/F(8)
      COMMON /C16/FF(4,2)
      IF(I-I/2*2 .EQ.0) GO TO 30
      F(1)=FF(1,1)
      F(2)=FF(2,1)
      F(5)=FF(3,1)
      F(6)=FF(4,1)
      GO TO 40
30    F(1)=FF(1,2)
      F(2)=FF(2,2)
      F(5)=FF(3,2)
      F(6)=FF(4,2)
 40   CONTINUE
      X=LAM(I)
      IF(ICASE(II).EQ.1) CALL CONS1(X,II)
      IF(ICASE(II).EQ.2) CALL CONS2(X,II)
      CALL MULT(X,II)
      RETURN
      END
```

```
      SUBROUTINE CONS1(X,II)
      COMPLEX X,Z,F,K2,K2B,K,KB,IY,P2IY
      COMMON /C1/IY,P2IY,PEI2
      COMMON /C5/K2(2,2),K2B(2,2),K(8,2),KB(8,2)
      COMMON /C6/Z(2)
      COMMON /C12/F(8)
      Z(1)=CEXP(P2IY*X)
      Z(2)=1/Z(1)
      IF(II .EQ. 2) GO TO 20
      F(3)=K2(1,1)*F(1)+K2(2,1)*Z(2)*F(2)
      F(4)=K2B(2,1)*Z(1)*F(1)+K2B(1,1)*F(2)
      GO TO 100
20    F(7)=K2(1,2)*F(5)+K2(2,2)*Z(1)*F(6)
      F(8)=K2B(2,2)*Z(2)*F(5)+K2B(1,2)*F(6)
100   RETURN
      END
```

```
      SUBROUTINE CONS2(X,II)
      COMPLEX X,X1,Z,K2,K2B,K,KB,F,IY,P2IY
      COMMON /C1/IY,P2IY,PBI2
      COMMON /C5/K2(2,2),K2B(2,2),K(8,2),KB(8,2)
      COMMON /C6/Z(2)
      COMMON /C12/F(8)
      Z(1)=CEXP(P2IY*X)
      Z(2)=1/Z(1)
      X1=X-1.
      IF(II .EQ. 2) GO TO 20
      F(3)=(K2(1,1)-X1)*F(1)+K2(2,1)*Z(2)*F(2)
      F(4)=K2B(2,1)*Z(1)*F(1)+(K2B(1,1)-X1)*F(2)
      GO TO 100
20    F(7)=(K2(1,2)-X1)*F(5)+K2(2,2)*Z(1)*F(6)
      F(8)=K2B(2,2)*Z(2)*F(5)+(K2B(1,2)-X1)*F(6)
100   RETURN
      END
```

```
      SUBROUTINE MULT(X,II)
      COMPLEX X,XX1,XX1X2,XX1X2X3,MM,MM1,MM2,MM3,MIS(2),MISB(2)
      COMPLEX SIG,DISP,SI,SIX,SIXX,DIS,DISX,DISXX,F,IY,P2IY,MUU
      COMMON/DIM/NA,NAA,NBT,NB,NQ,NR,NINT,NINT2,IALL,NITER,SINCOD
      COMMON /C1/IY,P2IY,PEI2
      COMMON /C4/SIG(3,2,2),DISP(2,2,2)
      COMMON /C7/ICASE(2)
      COMMON /C12/F(8)
      COMMON /C13/SI(12),SIX(12),SIXX(12)
     1           ,DIS(8),DISX(8),DISXX(8)
      COMMON /C14/MUU(2,2)
      J=(II-1)*4
      J1=J+1
      J2=J1+1
      J3=J2+1
      J4=J3+1
      XX1=X*(X-1.)
      XX1X2=XX1*(X-2.)
      XX1X2X3=XX1X2*(X-3.)
      IF(ICASE(II) .EQ. 2) GO TO 50
      DO 10 I=1,3
      DO 5 N=1,2
      MIS(N)=SIG(I,N,II)
      MISB(N)=CONJG(MIS(N))
5     CONTINUE
      K=(I-1)*4
      K1=K+1
      K2=K1+1
      K3=K2+1
      K4=K3+1
      MM=MIS(1)*F(J1)
      SI(K1)=MM*X
      SIX(K1)=MM*XX1
      SIXX(K1)=MM*XX1X2
      MM=MISB(1)*F(J2)
      SI(K2)=MM*X
      SIX(K2)=MM*XX1
      SIXX(K2)=MM*XX1X2
      MM=MIS(2)*F(J3)
      SI(K3)=MM*X
      SIX(K3)=MM*XX1
      SIXX(K3)=MM*XX1X2
      MM=MISB(2)*F(J4)
      SI(K4)=MM*X
      SIX(K4)=MM*XX1
      SIXX(K4)=MM*XX1X2
10    CONTINUE
      DO 20 I=1,2
      DO 15 N=1,2
      MIS(N)=DISP(I,N,II)
      MISB(N)=CONJG(MIS(N))
15    CONTINUE
      K=(I-1)*4
      K1=K+1
      K2=K1+1
      K3=K2+1
      K4=K3+1
      MM=MIS(1)*F(J1)
      DIS(K1)=MM
      DISX(K1)=MM*X
      DISXX(K1)=MM*XX1
      MM=MISB(1)*F(J2)
      DIS(K2)=MM
      DISX(K2)=MM*X
      DISXX(K2)=MM*XX1
```

```
      MM=MIS(2)*F(J3)
      DIS(K3)=MM
      DISX(K3)=MM*X
      DISXX(K3)=MM*XX1
      MM=MISB(2)*F(J4)
      DIS(K4)=MM
      DISX(K4)=MM*X
      DISXX(K4)=MM*XX1
 20   CONTINUE
      RETURN

 50   CONTINUE
      DO 60 I=1,3
      DO 55 N=1,2
      MIS(N)=SIG(I,N,II)
      MISB(N)=CONJG(MIS(N))
 55   CONTINUE
      K=(I-1)*4
      K1=K+1
      K2=K1+1
      K3=K2+1
      K4=K3+1
      MM=MIS(1)*F(J3)+MIS(2)*F(J1)
      MM1=MIS(1)*F(J1)
      MM2=MM1
      SI(K1)=MM*X
      SIX(K1)=(MM+MM1)*XX1
      SIXX(K1)=(MM+MM1+MM1)*XX1X2
      MM=MISB(1)*F(J4)+MISB(2)*F(J2)
      MM1=MISB(1)*F(J2)
      MM3=MM1
      SI(K2)=MM*X
      SIX(K2)=(MM+MM1)*XX1
      SIXX(K2)=(MM+MM1+MM1)*XX1X2
      MM=MM2
      SI(K3)=MM*XX1
      SIX(K3)=MM*XX1X2
      SIXX(K3)=MM*XX1X2X3
      MM=MM3
      SI(K4)=MM*XX1
      SIX(K4)=MM*XX1X2
      SIXX(K4)=MM*XX1X2X3
 60   CONTINUE
      DO 70 I=1,2
      DO 65 N=1,2
      MIS(N)=DISP(I,N,II)
      MISB(N)=CONJG(MIS(N))
 65   CONTINUE
      K=(I-1)*4
      K1=K+1
      K2=K1+1
      K3=K2+1
      K4=K3+1
      MM=MIS(1)*F(J3)+MIS(2)*F(J1)
      MM1=MIS(1)*F(J1)
      MM2=MM1
      DIS(K1)=MM
      DISX(K1)=(MM+MM1)*X
      DISXX(K1)=(MM+MM1+MM1)*XX1
      MM=MISB(1)*F(J4)+MISB(2)*F(J2)
      MM1=MISB(1)*F(J2)
      MM3=MM1
      DIS(K2)=MM
      DISX(K2)=(MM+MM1)*X
      DISXX(K2)=(MM+MM1+MM1)*XX1
```

```
      MM=MM2
      DIS(K3)=MM*X
      DISX(K3)=MM*XX1
      DISXX(K3)=MM*XX1X2
      MM=MM3
      DIS(K4)=MM*X
      DISX(K4)=MM*XX1
      DISXX(K4)=MM*XX1X2
70    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE EQUATE
      COMMON /C4/SIG(3,2,2),DISP(2,2,2)
      COMMON /C5/K2(2,2),K2B(2,2),K(8,2),KB(8,2)
      COMMON /C7/ICASE(2)
      COMMON /C14/MUU(2,2)
      COMPLEX SIG,DISP,MUU,K2,K2B,K,KB
      ICASE(2)=ICASE(1)
      DO 10 I=1,2
      K2(I,2)=K2(I,1)
      K2B(I,2)=K2B(I,1)
 10   CONTINUE
      DO 20 J=1,2
      DO 20 I=1,3
      SIG(I,J,2)=SIG(I,J,1)
 20   CONTINUE
      N=1
      IF(ICASE(1).EQ.1) N=2
      DO 25 I=1,N
      MUU(I,2)=MUU(I,1)
 25   CONTINUE
      DO 30 J=1,2
      DO 30 I=1,2
      DISP(I,J,2)=DISP(I,J,1)
 30   CONTINUE
      DO 40 I=1,8
      K(I,2)=K(I,1)
      KB(I,2)=KB(I,1)
 40   CONTINUE
      RETURN
      END
```

VITA

| | |
|---|---|
| Name: | Mohammad Ali Aminpour |
| Date of Birth: | March 21, 1955 |
| Place of Birth: | Esfahan, Iran |
| Parents: | Mr. and Mrs. Morteza Aminpour |
| Education: | |
| June 1974: | Diploma<br>Kharazmi High School<br>Tehran, Iran |
| March 1978: | B.S. in Civil Engineering<br>University of Washington<br>Seattle, Washington |
| December 1980: | M.S. in Aeronautics and Astronautics<br>University of Washington<br>Seattle, Washington |
| March 1981: | M.S. in Civil Engineering<br>University of Washington<br>Seattle, Washington |
| March 1986: | Ph.D. in Aeronautics and Astronautics<br>University of Washington<br>Seattle, Washington |